

ĐỀ CƯƠNG ÔN TẬP

CƠ SỞ DỮ LIỆU

Dành cho sinh viên đại học khối kinh tế, tài chính, công nghệ tài chính

Databases – a bilingual study guide

Mô hình ER · Mô hình quan hệ · Đại số quan hệ
Phụ thuộc hàm · Chuẩn hóa · SQL · Sao lưu

Bài tập có lời giải · Tự luận có dàn ý · Trắc nghiệm có đáp án
SQL chạy thật trên PostgreSQL

Song ngữ Việt – Anh · Vietnamese – English

Phiên bản 1.0 – Tháng 10 năm 2026

<https://tailieuso.vn>

MỤC LỤC

CONTENTS

Chương 1. Tổng quan về cơ sở dữ liệu – Chapter 1. Introduction to databases	7
1.1. Dữ liệu, thông tin và tri thức – 1.1. Data, information and knowledge.....	7
1.2. Trước khi có CSDL: lưu dữ liệu bằng tệp rời – 1.2. Before databases: keeping data in separate files	9
1.3. Cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu và hệ cơ sở dữ liệu – 1.3. Database, database management system and database system.....	14
1.4. Đặc trưng và lợi ích của CSDL – 1.4. Features and benefits of a database	17
1.5. Chức năng của HQTCSDDL – 1.5. What a DBMS does.....	22
1.6. Phân loại HQTCSDDL – 1.6. Classifying DBMSs.....	25
1.7. Người dùng CSDL – 1.7. Database users	29
1.8. Ứng dụng của CSDL, nhất là trong tài chính, ngân hàng và ví điện tử – 1.8. Uses of databases, especially in finance, banking and e-wallets.....	31
1.9. Dữ liệu cá nhân và trách nhiệm của người giữ CSDL – 1.9. Personal data and the duties of whoever keeps a database	33
1.10. Bài mẫu dạng đề thi – 1.10. Worked examples in exam style	36
1.11. Bảng thuật ngữ – 1.11. Glossary	38
1.12. Bài tập vận dụng có lời giải – Applied exercises with solutions.....	40
1.13. Câu hỏi tự luận và dàn ý trả lời – Essay questions and answer outlines	43
1.14. Câu hỏi trắc nghiệm – Multiple-choice questions.....	45
Chương 2. Mô hình dữ liệu và mô hình quan hệ – Chapter 2. Data models and the relational model	48
2.1. Mô hình dữ liệu là gì – 2.1. What a data model is.....	48
2.2. Ba mức mô hình: khái niệm, logic, vật lý – 2.2. Three levels of modelling: conceptual, logical, physical	50
2.3. Các loại mô hình dữ liệu – 2.3. Kinds of data model.....	52
2.4. Kiến trúc ba lược đồ và tính độc lập dữ liệu – 2.4. The three-schema architecture and data independence.....	56
2.5. Mô hình quan hệ: quan hệ, bộ, thuộc tính, miền – 2.5. The relational model: relation, tuple, attribute, domain	60
2.6. Tính chất của quan hệ và giá trị NULL – 2.6. Properties of a relation and the NULL value	64
2.7. Khóa – 2.7. Keys	68
2.8. Bài mẫu dạng đề thi – 2.8. Worked exercises in exam style	74
2.9. Đọc lược đồ quan hệ của CSDL ví Trường Tiền – 2.9. Reading the relational schema of the Trường Tiền wallet database	78
2.10. Bảng thuật ngữ – 2.10. Glossary	80
2.11. Bài tập vận dụng có lời giải – Applied exercises with solutions.....	82
2.12. Câu hỏi tự luận và dàn ý trả lời – Essay questions and answer outlines	84

2.13. Câu hỏi trắc nghiệm – <i>Multiple-choice questions</i>	86
Chương 3. Mô hình thực thể kết hợp và thiết kế khái niệm – Chapter 3. The entity–relationship model and conceptual design	88
3.1. Vòng đời phát triển cơ sở dữ liệu – <i>3.1. The database development life cycle</i>	88
3.2. Thu thập yêu cầu và viết mô tả nghiệp vụ – <i>3.2. Gathering requirements and writing the business description</i>	91
3.3. Thực thể và thuộc tính – <i>3.3. Entities and attributes</i>	93
3.4. Mỗi kết hợp và bậc của mỗi kết hợp – <i>3.4. Relationships and their degree</i>	97
3.5. Bản số và ràng buộc tham gia – <i>3.5. Cardinality and participation</i>	100
3.6. Thực thể yếu và thuộc tính phân biệt – <i>3.6. Weak entities and the discriminator</i>	105
3.7. Tổng quát hóa và chuyên biệt hóa – <i>3.7. Generalisation and specialisation</i>	107
3.8. Hai cách ký hiệu: Chen và chân chim – <i>3.8. Two notations: Chen and crow's foot</i>	109
3.9. Bảy bước vẽ ERD từ mô tả nghiệp vụ – <i>3.9. Seven steps to draw an ERD from a business description</i>	112
3.10. Nhận biết lỗi thiết kế ERD – <i>3.10. Spotting design faults in an ERD</i>	119
3.11. Ba bài ERD luyện tập có lời giải – <i>3.11. Three practice ERD problems with solutions</i>	120
3.12. Bài tập vận dụng có lời giải – <i>Applied exercises with solutions</i>	126
3.13. Câu hỏi tự luận và dàn ý trả lời – <i>Essay questions and answer outlines</i>	129
3.14. Câu hỏi trắc nghiệm – <i>Multiple-choice questions</i>	131
Chương 4. Chuyển mô hình ER sang mô hình quan hệ và ràng buộc toàn vẹn – Chapter 4. Mapping ER to relations and integrity constraints.....	133
4.1. Quy trình chuyển và bảng cho tập thực thể mạnh – <i>4.1. The mapping procedure and tables for strong entity sets</i>	133
4.2. Thuộc tính đa trị và thuộc tính dẫn xuất – <i>4.2. Multivalued and derived attributes</i> .136	
4.3. Mỗi kết hợp 1–n và 1–1: thêm khóa ngoại – <i>4.3. One-to-many and one-to-one relationships: adding a foreign key</i>	140
4.4. Mỗi kết hợp n–n và mỗi kết hợp có thuộc tính: bảng trung gian – <i>4.4. Many-to-many relationships and relationships with attributes: the junction table</i>	144
4.5. Mỗi kết hợp ba ngôi và mỗi kết hợp đệ quy – <i>4.5. Ternary and recursive relationships</i>	146
4.6. Thực thể yếu và tổng quát hóa – <i>4.6. Weak entities and generalisation</i>	149
4.7. Bài mẫu: từ sơ đồ ER của ví Trường Tiền đến chín lược đồ quan hệ – <i>4.7. Worked case: from the Trường Tiền wallet ER diagram to nine relation schemas</i>	153
4.8. Ràng buộc toàn vẹn: khái niệm và các loại – <i>4.8. Integrity constraints: the idea and the kinds</i>	158
4.9. Hành động khi xóa hoặc sửa dòng được tham chiếu – <i>4.9. What happens when a referenced row is deleted or updated</i>	161
4.10. Đặc tả ràng buộc toàn vẹn kiểu giáo trình Việt Nam – <i>4.10. Specifying integrity constraints in the Vietnamese textbook style</i>	164
4.11. Bảng so sánh tổng hợp và thuật ngữ – <i>4.11. Summary comparison and glossary</i> .168	
4.12. Bài tập vận dụng có lời giải – <i>Applied exercises with solutions</i>	170
4.13. Câu hỏi tự luận và dàn ý trả lời – <i>Essay questions and answer outlines</i>	173

4.14. Câu hỏi trắc nghiệm – <i>Multiple-choice questions</i>	174
Chương 5. Đại số quan hệ – Chapter 5. Relational algebra	177
5.1. Ngôn ngữ truy vấn hình thức và cách đọc biểu thức – 5.1. <i>A formal query language and how to read an expression</i>	177
5.2. Phép chọn σ và phép chiếu π – 5.2. <i>Selection σ and projection π</i>	180
5.3. Phép đổi tên ρ và phép gán – 5.3. <i>Renaming ρ and assignment</i>	184
5.4. Phép hợp $\cup$, giao $\cap$ và hiệu – 5.4. <i>Union $\cup$, intersection $\cap$ and difference</i> –	187
5.5. Tích Descartes và các phép kết – 5.5. <i>The Cartesian product and joins</i>	190
5.6. Kết ngoài và nửa kết – 5.6. <i>Outer joins and the semijoin</i>	196
5.7. Phép chia $\div$: tìm đối tượng thỏa MỌI điều kiện – 5.7. <i>Division $\div$: finding items that meet EVERY condition</i>	200
5.8. Gom nhóm và hàm tổng hợp γ – 5.8. <i>Grouping and aggregate functions γ</i>	204
5.9. Biểu thức lồng, cây biểu thức và tối ưu truy vấn – 5.9. <i>Nested expressions, expression trees and query optimisation</i>	207
5.10. Bài mẫu dạng đề thi – 5.10. <i>Worked examples in exam style</i>	211
5.11. Bảng thuật ngữ – 5.11. <i>Glossary</i>	217
5.12. Bài tập vận dụng có lời giải – <i>Applied exercises with solutions</i>	219
5.13. Câu hỏi tự luận và dàn ý trả lời – <i>Essay questions and answer outlines</i>	222
5.14. Câu hỏi trắc nghiệm – <i>Multiple-choice questions</i>	223
Chương 6. Phụ thuộc hàm, bao đóng và khóa – Chapter 6. Functional dependencies, closures and keys	226
6.1. Bảng phẳng từ Excel và vì sao dữ liệu lặp gây rắc rối – 6.1. <i>A flat sheet from Excel and why repeated data causes trouble</i>	226
6.2. Phụ thuộc hàm: biết cột này thì suy ra cột kia – 6.2. <i>Functional dependency: know one column, work out another</i>	228
6.3. Bốn loại phụ thuộc hàm cần phân biệt – 6.3. <i>Four kinds of functional dependency to tell apart</i>	232
6.4. Hệ tiên đề Armstrong và các luật suy diễn – 6.4. <i>Armstrong's axioms and the inference rules</i>	234
6.5. Bao đóng: tìm mọi thứ một nhóm cột xác định được – 6.5. <i>Closure: finding everything a group of columns determines</i>	237
6.6. Siêu khóa và khóa: cột nào nhận ra một dòng – 6.6. <i>Superkeys and keys: which columns pick out a row</i>	242
6.7. Tìm tất cả các khóa – 6.7. <i>Finding all the keys</i>	244
6.8. Phủ tối thiểu: viết gọn tập phụ thuộc mà không đổi nghĩa – 6.8. <i>Minimal cover: shortening a set of dependencies without changing its meaning</i>	248
6.9. Áp dụng vào CSDL Trường Tiền – 6.9. <i>Applying it to the Trường Tiền database</i> ..	253
6.10. Năm dạng bài và cách giải – 6.10. <i>Five types of exercise and how to solve them</i> ..	256
6.11. Bảng thuật ngữ – 6.11. <i>Glossary</i>	257
6.12. Bài tập vận dụng có lời giải – <i>Applied exercises with solutions</i>	259
6.13. Câu hỏi tự luận và dàn ý trả lời – <i>Essay questions and answer outlines</i>	261
6.14. Câu hỏi trắc nghiệm – <i>Multiple-choice questions</i>	263

Chương 7. Chuẩn hóa cơ sở dữ liệu – Chapter 7. Database normalisation	265
7.1. Bảng phẳng và ba dị thường – 7.1. The flat table and three anomalies	265
7.2. Ôn nhanh: phụ thuộc hàm, bao đóng và khóa – 7.2. A quick review: functional dependencies, closures and keys	269
7.3. Dạng chuẩn 1: mỗi ô một giá trị – 7.3. First normal form: one value per cell.....	274
7.4. Dạng chuẩn 2 và dạng chuẩn 3 – 7.4. Second and third normal form.....	276
7.5. Dạng chuẩn Boyce–Codd (BCNF) – 7.5. Boyce–Codd normal form (BCNF).....	280
7.6. Xác định dạng chuẩn cao nhất của một lược đồ – 7.6. Finding the highest normal form of a schema	283
7.7. Phân rã: giữ thông tin và giữ phụ thuộc – 7.7. Decomposition: keeping information and keeping dependencies	286
7.8. Thuật toán tổng hợp 3NF – 7.8. The 3NF synthesis algorithm.....	292
7.9. Thuật toán phân rã BCNF – 7.9. The BCNF decomposition algorithm.....	296
7.10. Dạng chuẩn 4, dạng chuẩn 5 và phi chuẩn hóa – 7.10. Fourth and fifth normal form, and denormalisation.....	299
7.11. Bảng thuật ngữ – 7.11. Glossary	301
7.12. Bài tập vận dụng có lời giải – Applied exercises with solutions.....	303
7.13. Câu hỏi tự luận và dàn ý trả lời – Essay questions and answer outlines	306
7.14. Câu hỏi trắc nghiệm – Multiple-choice questions.....	308
Chương 8. SQL: định nghĩa và cập nhật dữ liệu – Chapter 8. SQL: data definition and data manipulation.....	310
8.1. SQL là gì và các nhóm lệnh – 8.1. What SQL is and its groups of statements.....	310
8.2. Cài PostgreSQL, pgAdmin và làm quen psql – 8.2. Installing PostgreSQL and pgAdmin, and getting to know psql	313
8.3. Tạo cơ sở dữ liệu và chọn kiểu dữ liệu – 8.3. Creating a database and choosing data types.....	317
8.4. CREATE TABLE và các ràng buộc – 8.4. CREATE TABLE and constraints	320
8.5. Sửa bảng: ALTER TABLE, DROP TABLE và TRUNCATE – 8.5. Changing tables: ALTER TABLE, DROP TABLE and TRUNCATE	327
8.6. INSERT: thêm dòng vào bảng – 8.6. INSERT: adding rows to a table	329
8.7. UPDATE và DELETE: sửa và xóa dòng – 8.7. UPDATE and DELETE: changing and removing rows	333
8.8. CREATE VIEW: khung nhìn – 8.8. CREATE VIEW: views	340
8.9. CREATE INDEX: chỉ mục và đọc EXPLAIN – 8.9. CREATE INDEX: indexes and reading EXPLAIN	343
8.10. Dựng lại toàn bộ CSDL Trường Tiền bằng script – 8.10. Rebuilding the whole Trường Tiền database from a script.....	347
8.11. Bảng thuật ngữ – 8.11. Glossary	352
8.12. Bài tập vận dụng có lời giải – Applied exercises with solutions.....	354
8.13. Câu hỏi tự luận và dàn ý trả lời – Essay questions and answer outlines	357
8.14. Câu hỏi trắc nghiệm – Multiple-choice questions.....	359
Chương 9. SQL: truy vấn dữ liệu – Chapter 9. SQL: querying data.....	362

9.1. Câu truy vấn đầu tiên: SELECT và FROM – 9.1. The first query: SELECT and FROM.....	362
9.2. Lọc dòng bằng WHERE – 9.2. Filtering rows with WHERE.....	364
9.3. Bỏ dòng trùng, sắp xếp và cắt bớt kết quả – 9.3. Removing duplicates, sorting and trimming the result	368
9.4. Biểu thức, hàm chuỗi, hàm ngày giờ và CASE – 9.4. Expressions, string functions, date-time functions and CASE.....	370
9.5. Hàm tổng hợp, GROUP BY, HAVING và thứ tự thực hiện câu SELECT – 9.5. Aggregate functions, GROUP BY, HAVING and the order in which SELECT runs.....	373
9.6. Kết nối nhiều bảng: JOIN – 9.6. Joining tables: JOIN.....	377
9.7. Truy vấn con và phép chia – 9.7. Subqueries and division.....	383
9.8. UNION, INTERSECT, EXCEPT, CTE và hàm cửa sổ – 9.8. UNION, INTERSECT, EXCEPT, CTEs and window functions	389
9.9. Quy trình năm bước từ đề bài đến câu truy vấn – 9.9. A five-step routine from problem to query	393
9.10. Mười lăm truy vấn nghiệp vụ của ví Trường Tiền – 9.10. Fifteen business queries on the Trường Tiền wallet	394
9.11. Bảng thuật ngữ – 9.11. Glossary	401
9.12. Bài tập vận dụng có lời giải – Applied exercises with solutions.....	403
9.13. Câu hỏi tự luận và dàn ý trả lời – Essay questions and answer outlines	405
9.14. Câu hỏi trắc nghiệm – Multiple-choice questions.....	407
Chương 10. Bảo mật, giao tác, sao lưu và phục hồi – Chapter 10. Security, transactions, backup and recovery.....	409
10.1. An toàn cơ sở dữ liệu: ba lớp bảo vệ – 10.1. Database safety: three layers of protection	409
10.2. Tài khoản, vai trò và mật khẩu – 10.2. Accounts, roles and passwords	412
10.3. GRANT và REVOKE: cấp và thu quyền – 10.3. GRANT and REVOKE: giving and withdrawing permissions.....	415
10.4. Kịch bản phân quyền cho Trường Tiền – 10.4. A permission plan for Trường Tiền	419
10.5. Che cột nhạy cảm, dữ liệu cá nhân và nhật ký kiểm toán – 10.5. Hiding sensitive columns, personal data and the audit log	423
10.6. Giao tác và bốn tính chất ACID – 10.6. Transactions and the four ACID properties	427
10.7. Nhiều người cùng ghi: bốn hiện tượng và mức cô lập – 10.7. Many writers at once: four problems and isolation levels	432
10.8. Khóa, SELECT ... FOR UPDATE và khóa chết – 10.8. Locks, SELECT ... FOR UPDATE and deadlocks.....	437
10.9. Sao lưu và phục hồi bằng pg_dump – 10.9. Backup and restore with pg_dump.....	440
10.10. Kế hoạch sao lưu, nhật ký ghi trước và người quản trị CSDL – 10.10. The backup plan, the write-ahead log and the database administrator	445
10.11. Bảng thuật ngữ – 10.11. Glossary	449
10.12. Bài tập vận dụng có lời giải – Applied exercises with solutions.....	451
10.13. Câu hỏi tự luận và dàn ý trả lời – Essay questions and answer outlines	454

10.14. Câu hỏi trắc nghiệm – <i>Multiple-choice questions</i>	456
Phụ lục A. Đáp án câu hỏi trắc nghiệm – <i>Appendix A. Answer key</i>.....	459
Phụ lục B. Tóm tắt cú pháp SQL và ký hiệu – <i>Appendix B. SQL syntax and notation summary</i>.....	464
B.1. Cú pháp SQL của PostgreSQL – <i>SQL syntax in PostgreSQL</i>	464
B.2. Ký hiệu và công thức theo chương – <i>Notation and formulas by chapter</i>	465
Phụ lục C. Thuật ngữ Việt – Anh – <i>Appendix C. Vietnamese–English glossary</i>	471
Phụ lục D. Lược đồ CSDL ví Trường Tiền – <i>Appendix D. Schema of the Trường Tiền wallet database</i>.....	479
D.1. Mô tả bảng và cột – <i>Tables and columns</i>	479
D.2. Dữ liệu từng bảng – <i>Table data</i>	483
Phụ lục E. Văn bản pháp luật liên quan – <i>Appendix E. Related legislation</i>	491
Tài liệu tham khảo – <i>References</i>.....	492

Chương 1. Tổng quan về cơ sở dữ liệu – Chapter 1. Introduction to databases

Mục tiêu chương – Chapter objectives

- Phân biệt dữ liệu, thông tin và tri thức bằng ví dụ từ sao kê ví điện tử.
Tell data, information and knowledge apart using examples from an e-wallet statement.
- Giải thích vì sao lưu dữ liệu bằng các tệp rời (như bảng Excel của từng phòng) gây ra dư thừa, sai lệch và rủi ro, và cơ sở dữ liệu (CSDL, database) giải quyết từng vấn đề ra sao.
Explain why keeping data in separate files (such as each department's own spreadsheet) causes duplication, mismatches and risk, and how a database solves each problem.
- Phân biệt CSDL, hệ quản trị cơ sở dữ liệu (HQTCSDL, DBMS) và hệ cơ sở dữ liệu; kể được năm thành phần của một hệ cơ sở dữ liệu.
Distinguish a database, a database management system (DBMS) and a database system, and name the five parts of a database system.
- Nêu đặc trưng, lợi ích, chức năng và cách phân loại HQTCSDL; xác định các nhóm người dùng CSDL và việc của từng nhóm.
State the features, benefits, functions and classification of DBMSs, and identify the groups of database users and what each does.
- Nắm các quy định về dịch vụ ví điện tử và về bảo vệ dữ liệu cá nhân mà người giữ CSDL của một công ty tài chính phải tuân theo; tự xem được từ điển dữ liệu của một CSDL bằng các lệnh đơn giản của SQL (Structured Query Language, ngôn ngữ truy vấn có cấu trúc).
Know the rules on e-wallet services and on personal data protection that the keeper of a financial company's database must follow, and look at a database's data dictionary with simple commands in SQL (Structured Query Language).

1.1. Dữ liệu, thông tin và tri thức – 1.1. Data, information and knowledge

Bạn mở ứng dụng ví và xem sao kê. Dòng đầu tiên của tháng 9/2026 trên ví V0001 ghi: ngày 01/09/2026, 08:15, nạp tiền, 2.000.000 đ. Dòng này là một sự việc đã xảy ra, được ghi lại đúng như nó diễn ra. Mẫu ghi như vậy gọi là dữ liệu (data).

You open a wallet app and look at the statement. The first line for September 2026 on wallet V0001 reads: 1 September 2026, 08:15, top-up, VND 2,000,000. That line records something that happened, written down just as it happened. A record like this is called data.

Một dòng riêng lẻ chưa trả lời được câu hỏi nào. Giả sử bạn gom 40 dòng của tháng 9, bỏ những dòng thất bại, rồi đếm và cộng các dòng nạp tiền. Bạn có một câu có nghĩa: tháng 9 có 11 lượt nạp thành công, tổng 17.500.000 đ. Câu này là thông tin (information): dữ liệu đã được xử lý và đặt vào ngữ cảnh để trả lời một câu hỏi.

A single line answers no question. Suppose you gather the 40 lines for September, drop the failed ones, then count and add up the top-ups. You get a sentence that means something: September has 11 successful top-ups, VND 17,500,000 in total. That sentence is information: data that has been processed and placed in context to answer a question.

Thông tin chưa nói bạn nên làm gì. Khi bạn đọc nhiều thông tin cùng nhau và hiểu ra một quy luật để quyết định, bạn có tri thức (knowledge). Ví dụ: ví Trường Tiền thu 3.300 đ cho mỗi lần rút, dù rút ít hay nhiều. Từ đó bạn hiểu rằng rút năm lần nhỏ tốn phí gấp năm lần rút một lần, nên gom lại rút một lần thì rẻ hơn.

Information does not yet tell you what to do. When you read several pieces of information together and see a pattern you can decide by, you have knowledge. For example, the Trường Tiền wallet charges VND 3,300 per withdrawal, large or small. From that you see that five small withdrawals cost five times the fee of one, so gathering them into a single withdrawal is cheaper.

Cấp Level	Là gì What it is	Ví dụ ở ví Trường Tiền Example at Trường Tiền	Câu hỏi giúp nhận ra Question that spots it
Dữ liệu Data	Sự việc hoặc giá trị được ghi lại, chưa giải thích. <i>A fact or value that has been recorded, not yet explained.</i>	V0001; nạp; 2.000.000; 01/09/2026 08:15 <i>V0001; top-up; 2,000,000; 01/09/2026 08:15</i>	Mẫu này đã trả lời câu hỏi nào chưa? Chưa. <i>Has this item answered any question yet? No.</i>
Thông tin Information	Dữ liệu đã được xử lý, có ngữ cảnh, trả lời một câu hỏi. <i>Data that has been processed, has context and answers a question.</i>	Tháng 9 có 11 lượt nạp thành công, tổng 17.500.000 đ. <i>September has 11 successful top-ups, VND 17,500,000 in total.</i>	Nó cho tôi biết điều gì về thực tế? Một con số hay kết luận cụ thể. <i>What does it tell me about reality? A specific figure or conclusion.</i>
Tri thức Knowledge	Hiểu biết rút ra từ nhiều thông tin, dùng để chọn cách làm. <i>Understanding drawn from several pieces of information and used to choose what to do.</i>	Phí rút tính theo lần, nên gom rút một lần thì rẻ hơn. <i>The fee is per withdrawal, so one larger withdrawal is cheaper.</i>	Tôi dùng nó để làm gì? Để quyết định hành động. <i>What do I use it for? To decide what to do.</i>

PHẢI THUỘC / MEMORISE – Dữ liệu là sự việc được ghi lại. Thông tin là dữ liệu đã xử lý để trả lời một câu hỏi. Tri thức là hiểu biết từ thông tin, dùng để quyết định.

Data is a fact that has been recorded. Information is data processed to answer a question. Knowledge is understanding built from information and used to decide.

Cách phân biệt ba cấp – *How to tell the three levels apart*

1. Đọc mẫu cần xét. Hỏi: đây là một sự việc hay giá trị đơn lẻ, chưa ai đếm, cộng hay so sánh? Nếu đúng, đó là dữ liệu.
Read the item. Ask: is this a single fact or value that nobody has counted, added or compared? If so, it is data.
2. Nếu mẫu là kết quả của việc đếm, cộng, lọc hay so sánh nhiều dữ liệu và trả lời một câu hỏi cụ thể, đó là thông tin.
If the item is the result of counting, adding, filtering or comparing several pieces of data and answers a specific question, it is information.

3. Nếu mẫu là một quy tắc hay nhận định dùng để chọn cách làm, vượt khỏi việc mô tả, đó là tri thức.

If the item is a rule or judgement used to choose what to do, going beyond description, it is knowledge.

VÍ DỤ / EXAMPLE – Xét ba mẫu: (a) "KH001"; (b) "Có 3 khách hạng Vàng"; (c) "Phí rút tính theo lần nên gom rút một lần thì rẻ hơn". Mẫu (a) chỉ là một mã, chưa trả lời gì: dữ liệu. Mẫu (b) là kết quả của việc đếm khách theo hạng: thông tin. Mẫu (c) là quy tắc để quyết định cách rút tiền: tri thức. Nghĩa là cùng một ví, ba cấp khác nhau nằm ở mức độ xử lý và mục đích dùng.

Take three items: (a) "KH001"; (b) "There are 3 Gold customers"; (c) "The fee is per withdrawal, so one larger withdrawal is cheaper". Item (a) is just a code and answers nothing: data. Item (b) results from counting customers by tier: information. Item (c) is a rule for deciding how to withdraw: knowledge. So the same wallet yields three levels, which differ in how far the item has been processed and what it is used for.

Dữ liệu cũng khác nhau về hình dạng. Dữ liệu có cấu trúc xếp vào dòng và cột cố định, như bảng sao kê. Dữ liệu không có cấu trúc thì không có khuôn như vậy: ảnh chụp giấy tờ khi xác minh tài khoản, tin nhắn khách gửi cho bộ phận hỗ trợ. Phần lớn sách này nói về dữ liệu có cấu trúc, vì đó là thứ CSDL quan hệ lưu tốt và là nền để học các lệnh truy vấn ở những chương sau.

Data also differs in shape. Structured data fits fixed rows and columns, like a statement table. Unstructured data has no such mould: a photo of an ID taken when an account is verified, or a message a customer sends to the support team. Most of this book deals with structured data, which relational databases store well and which is the basis for the query commands in later chapters.

LỖI HAY GẶP / COMMON MISTAKE – Cho rằng chữ là thông tin còn số là dữ liệu. Ranh giới nằm ở chỗ mẫu đã được xử lý để trả lời câu hỏi hay chưa: "Huế" trong cột tỉnh thành là dữ liệu, còn "có 1 khách ở Huế" là thông tin.

Thinking that words are information and numbers are data. The line falls where an item has been processed to answer a question: "Huế" in a province column is data, while "there is 1 customer in Huế" is information.

LỖI HAY GẶP / COMMON MISTAKE – Nói rằng bảng sao kê chứa thông tin. Nói chính xác, bảng sao kê giữ dữ liệu; thông tin xuất hiện khi dữ liệu được đếm, cộng, so sánh và đặt vào ngữ cảnh.

Saying that a statement table contains information. Strictly, it holds data; information appears when the data is counted, added, compared and placed in context.

1.2. Trước khi có CSDL: lưu dữ liệu bằng tệp rời – 1.2. Before databases: keeping data in separate files

Công ty Trường Tiền ngày đầu hoạt động chỉ có ba phòng và ba bảng Excel. Phòng Vận hành giữ tệp ví tiền, phòng Kế toán giữ tệp sao kê giao dịch, phòng Chăm sóc khách hàng (CSKH) giữ tệp khách hàng. Mỗi phòng có một chương trình nhỏ riêng để đọc tệp của mình. Cách làm này gọi là hệ thống tệp (file-based system): dữ liệu nằm trong các tệp rời,

chương trình nào lo tệp này. Cách làm thay thế là cơ sở dữ liệu (CSDL), một kho dữ liệu dùng chung có phần mềm quản lý; mục 1.3 nói rõ.

On its first day the Trường Tiền company has just three departments and three Excel sheets. Operations keeps the wallet file, Accounting keeps the transaction statement file, and Customer Care (CSKH) keeps the customer file. Each department has a small program of its own to read its file. This way of working is called a file-based system: data sits in separate files, and each program looks after its own file. The alternative is a database, a shared store of data run by management software; section 1.3 spells it out.

Tệp File	Phòng giữ Kept by	Các cột Columns
Ví tiền Wallets	Vận hành Operations	Mã ví, họ tên khách, số điện thoại Wallet code, customer name, phone number
Sao kê Statement	Kế toán Accounting	Mã giao dịch, họ tên khách, số điện thoại, số tiền Transaction code, customer name, phone number, amount
Khách hàng Customers	CSKH Customer Care	Họ tên khách, số điện thoại, hạng thành viên Customer name, phone number, membership tier

Ngay từ bảng này bạn đã thấy họ tên và số điện thoại của cùng một khách nằm ở cả ba tệp. Đoạn lệnh sau dựng lại ba tệp bằng ba bảng rời trong PostgreSQL (phần mềm quản lý dữ liệu ta dùng xuyên suốt sách), rồi đổi số điện thoại của khách Trần Minh Khôi trong tệp CSKH, vì phòng này vừa nhận được số mới. Các lệnh viết bằng SQL; bạn chưa cần hiểu từng lệnh, Chương 8 sẽ dạy. Hãy nhìn kết quả ở cuối.

Even in this table you can see that one customer's name and phone number appear in all three files. The next block rebuilds the three files as three separate tables in PostgreSQL (the data management software we use throughout the book), then changes the phone number of the customer Trần Minh Khôi in the CSKH file, because that team has just received a new number. The commands are written in SQL; you do not need to understand each one yet, as Chapter 8 teaches them. Look at the result at the end.

```
CREATE TABLE TepViTien (MaVi text, HoTen text, SDT text);
CREATE TABLE TepSaoKe (MaGD int, HoTen text, SDT text, SoTien int);
CREATE TABLE TepCSKH (HoTen text, SDT text, Hang text);
INSERT INTO TepViTien VALUES ('V0001', 'Trần Minh Khôi', '0905550101');
INSERT INTO TepSaoKe VALUES (1, 'Trần Minh Khôi', '0905550101', 2000000),
(6, 'Trần Minh Khôi', '0905550101', 500000);
INSERT INTO TepCSKH VALUES ('Trần Minh Khôi', '0905550101', 'Vàng');
UPDATE TepCSKH SET SDT = '0905550199' WHERE HoTen = 'Trần Minh Khôi';
SELECT 'ViTien' AS tep, SDT FROM TepViTien
UNION ALL SELECT 'SaoKe', SDT FROM TepSaoKe
UNION ALL SELECT 'CSKH', SDT FROM TepCSKH;
```

Kết quả chạy / Output

```
CREATE TABLE
CREATE TABLE
CREATE TABLE
INSERT 0 1
INSERT 0 2
INSERT 0 1
UPDATE 1
 tep | sdt
-----+-----
```

ViTien		0905550101
SaoKe		0905550101
SaoKe		0905550101
CSKH		0905550199
(4 rows)		

Cùng một khách nhưng giờ có hai số điện thoại: tệp CSKH ghi số mới, ba dòng còn lại (một ở tệp ví, hai ở tệp sao kê) vẫn ghi số cũ. Không ai báo lỗi, vì mỗi tệp tự nó vẫn hợp lệ. Khi Kế toán cần gọi khách xác nhận một giao dịch, họ sẽ gọi sai số.

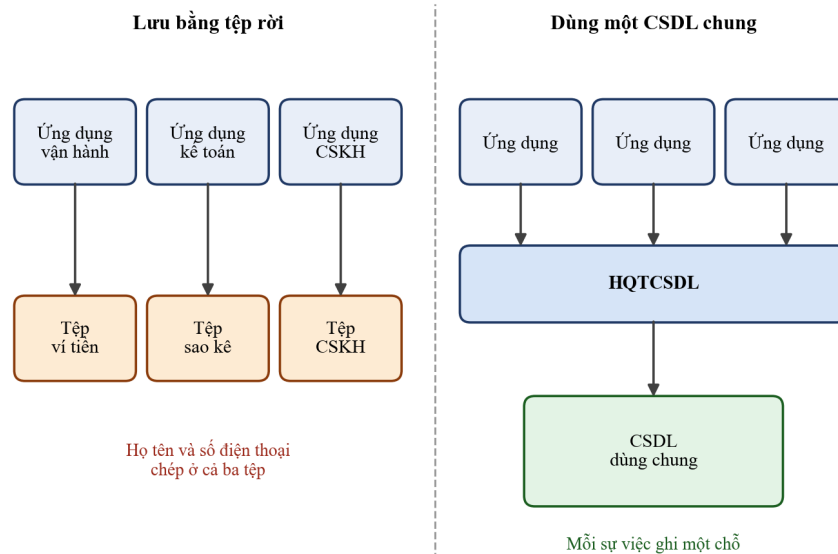
The same customer now has two phone numbers: the CSKH file holds the new one, and the other three lines (one in the wallet file, two in the statement file) still hold the old one. Nobody sees an error, because each file is valid on its own. When Accounting needs to ring the customer to confirm a transaction, it will ring the wrong number.

Sai lệch này chỉ là một trong nhiều vấn đề của hệ thống tệp. Bảng sau liệt kê từng vấn đề, đặt cạnh chuyện xảy ra ở ba tệp của Trường Tiền.

This mismatch is only one of the problems of a file-based system. The next table lists each problem beside what happens in Trường Tiền's three files.

Vấn đề <i>Problem</i>	Chuyện xảy ra ở ba tệp <i>What happens in the three files</i>	Hậu quả <i>Consequence</i>
Dư thừa dữ liệu (data redundancy) <i>Data redundancy</i>	Họ tên, số điện thoại của một khách được chép ở cả ba tệp. <i>One customer's name and phone number are copied into all three files.</i>	Tổn chỗ, và phải sửa ở ba nơi mỗi khi có thay đổi. <i>It wastes space, and every change has to be made in three places.</i>
Không nhất quán (inconsistency) <i>Inconsistency</i>	Sửa số điện thoại ở tệp CSKH mà quên hai tệp kia. <i>The phone number is changed in the CSKH file and the other two are forgotten.</i>	Một khách có hai số điện thoại; không biết số nào đúng. <i>One customer has two phone numbers, and nobody knows which is right.</i>
Khó truy xuất (data isolation) <i>Data isolation</i>	Muốn biết khách hạng Vàng nạp bao nhiêu phải ghép tay tệp CSKH với tệp sao kê. <i>To find how much Gold customers topped up, someone must match the CSKH file and the statement file by hand.</i>	Mỗi câu hỏi mới cần một chương trình mới hoặc một buổi làm tay. <i>Every new question needs a new program or an afternoon of manual work.</i>
Khó giữ ràng buộc (integrity) <i>Integrity</i>	Quy tắc "số tiền phải lớn hơn 0" nằm trong chương trình của Kế toán; ai sửa tệp bằng Excel thì không bị kiểm. <i>The rule "the amount must be above 0" lives inside the Accounting program; anyone editing the file in Excel escapes the check.</i>	Số tiền âm hay mã ví không tồn tại lọt vào dữ liệu. <i>Negative amounts or wallet codes that do not exist slip into the data.</i>

Vấn đề <i>Problem</i>	Chuyện xảy ra ở ba tệp <i>What happens in the three files</i>	Hậu quả <i>Consequence</i>
Bảo mật yếu (security) <i>Weak security</i>	Tệp chỉ có hai mức: cho mở hoặc không cho mở. Cho Kế toán xem tệp sao kê là cho xem luôn số điện thoại. <i>A file has two settings: openable or not. Letting Accounting open the statement file also shows them the phone numbers.</i>	Không thể cho xem một phần cột hay một phần dòng. <i>You cannot grant access to only some columns or rows.</i>
Khó chia sẻ: truy cập đồng thời (concurrency) <i>Hard to share: concurrent access</i>	Hai người cùng mở một tệp Excel: người thứ hai chỉ được đọc, hoặc cả hai lưu và bản lưu sau đè bản lưu trước. <i>Two people open one Excel file: the second can only read, or both save and the later save overwrites the earlier one.</i>	Làm việc chờ nhau, hoặc mất thay đổi của một người. <i>People wait for each other, or one person's changes are lost.</i>
Phụ thuộc cấu trúc tệp <i>Dependence on file layout</i>	Thêm một cột vào tệp ví thì chương trình đọc tệp phải sửa theo. <i>Adding a column to the wallet file means the program that reads it must be changed.</i>	Mỗi thay đổi nhỏ kéo theo sửa nhiều chương trình. <i>Each small change drags in edits to several programs.</i>



Hình 1.1. Hai cách tổ chức dữ liệu. Bên trái, mỗi ứng dụng có tệp riêng và họ tên, số điện thoại bị chép ở cả ba tệp. Bên phải, cả ba ứng dụng đi qua một phần mềm quản lý chung (HQTCSDL, giải thích ở mục 1.3) để dùng một CSDL, mỗi sự việc ghi một chỗ.

Figure 1.1. Two ways to organise data. On the left, each application has its own file and names and phone numbers are copied into all three. On the right, all three applications go through one piece of management software (the DBMS, explained in section 1.3) and share one database, with each fact stored once.

Cách nhận ra nhược điểm trong một tình huống – *How to spot the weaknesses in a situation*

1. Liệt kê dữ liệu của tình huống ở những nơi nào: tệp nào, phòng nào, chương trình nào.
List where the situation keeps its data: which files, which departments, which programs.
2. Gạch dưới mọi thông tin xuất hiện ở từ hai nơi trở lên. Đó là dư thừa; nếu hai nơi ghi khác nhau thì đã sai lệch.
Underline every piece of information that appears in two or more places. That is redundancy; if the places disagree, there is already a mismatch.
3. Với từng câu hỏi của người dùng, kiểm xem trả lời có phải ghép tay nhiều tệp không (khó truy xuất), và ai đang giữ các quy tắc kiểm tra dữ liệu (ràng buộc).
For each user question, check whether answering means matching several files by hand (isolation), and who holds the rules that check the data (integrity).
4. Hỏi tiếp: có thể cho người này xem tệp mà giấu bớt cột không, và hai người sửa cùng lúc thì chuyện gì xảy ra (bảo mật, truy cập đồng thời).
Then ask: can one person be shown the file with some columns hidden, and what happens when two people edit at once (security, concurrency)?

VÍ DỤ / EXAMPLE – Một quán cà phê có tệp thực đơn do bếp giữ và tệp hóa đơn do thu ngân giữ; cả hai đều ghi giá "cà phê sữa". Bước 1: dữ liệu ở hai tệp. Bước 2: giá cà phê sữa nằm ở cả hai nơi, đây là dư thừa; khi quán tăng giá mà thu ngân quên sửa, hai tệp ghi hai giá, đây là sai lệch. Bước 3: muốn biết món nào bán chạy phải ghép hai tệp bằng tay. Bước 4: ai mở được tệp hóa đơn cũng thấy hết doanh thu. Kết luận: quán đang gặp dư thừa, không nhất quán, khó truy xuất và bảo mật yếu.

A café has a menu file kept by the kitchen and a bill file kept by the cashier; both record the price of a "milk coffee". Step 1: data in two files. Step 2: the price sits in both places, which is redundancy; when the café raises the price and the cashier forgets to change it, the files hold two prices, which is a mismatch. Step 3: to learn which dish sells best, someone must match the two files by hand. Step 4: anyone who can open the bill file sees all the takings. Conclusion: the café suffers from redundancy, inconsistency, poor retrieval and weak security.

Excel vẫn là công cụ tốt cho một người với vài trăm dòng. Vấn đề chỉ xuất hiện khi nhiều người, nhiều phòng cùng dựa vào dữ liệu và dữ liệu lớn dần.

Excel is still a good tool for one person with a few hundred rows. The problems appear only when many people and departments depend on the same data and the data keeps growing.

LỖI HAY GẶP / COMMON MISTAKE – Nhầm dư thừa với sao lưu. Sao lưu là bản chép có chủ đích, cất riêng để cứu dữ liệu khi hỏng. Dư thừa là cùng một sự việc sống ở nhiều nơi cùng lúc và phải được sửa đồng thời.

Mixing up redundancy with backup. A backup is a deliberate copy kept apart to rescue data after a failure. Redundancy is one fact living in several places at once, all of which must be changed together.

LỖI HAY GẶP / COMMON MISTAKE – Cho rằng chỉ cần nhắc mọi người "nhớ sửa cả ba tệp" là đủ. Quy tắc dựa vào trí nhớ sẽ thủng; cần hệ thống giữ mỗi sự việc ở một chỗ.

Believing it is enough to remind everyone to "change all three files". A rule that relies on memory will leak; the system needs to keep each fact in one place.

1.3. Cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu và hệ cơ sở dữ liệu – 1.3. Database, database management system and database system

Hãy nghĩ đến danh bạ trong điện thoại. Mỗi người chỉ có một mục, dữ liệu về họ (tên, số, ảnh) nằm cạnh nhau, và mọi ứng dụng gọi điện, nhắn tin đều đọc cùng danh bạ đó. Cơ sở dữ liệu (CSDL, database) là phiên bản đầy đủ của ý tưởng ấy: một tập dữ liệu có tổ chức, liên quan với nhau, phản ánh một phần của thế giới thật, được lưu lại để nhiều người và nhiều chương trình cùng dùng.

Think of the contacts list in your phone. Each person has a single entry, the data about them (name, number, photo) sits together, and every calling or messaging app reads the same list. A database is the full-size version of that idea: an organised set of related data that reflects part of the real world and is kept so that many people and programs can use it together.

CSDL của Trường Tiền phản ánh việc vận hành ví: ai là khách hàng, mỗi người có những ví nào, ví liên kết với ngân hàng nào, ví đã làm những giao dịch gì. Dữ liệu được tổ chức theo ba cấp. Một trường (field) là một ô, ví dụ họ tên của một khách. Một bản ghi (record) là một dòng gồm các trường nói về cùng một đối tượng, ví dụ cả hàng về khách KH001. Một bảng (table) là tập hợp các bản ghi có cùng các trường, ví dụ bảng khách hàng. Trong sách này ta nói "cột" thay cho trường và "dòng" thay cho bản ghi.

The Trường Tiền database reflects how the wallet runs: who the customers are, which wallets each owns, which banks a wallet is linked to, and which transactions a wallet has made. The data is organised in three levels. A field is one cell, for example a customer's name. A record is a row of fields about the same thing, for example the whole line about customer KH001. A table is a set of records with the same fields, for example the customer table. In this book we say "column" for field and "row" for record.

Muốn xem CSDL Trường Tiền có những bảng nào, ta dùng psql, chương trình dòng lệnh đi kèm PostgreSQL. Lệnh \dt (viết tắt của "describe tables") liệt kê các bảng. Tên bảng và tên cột sẽ hiện bằng chữ thường; lý do được giải thích ở Chương 8.

To see which tables the Trường Tiền database holds, we use psql, the command-line program that comes with PostgreSQL. The command \dt (short for "describe tables") lists the tables. Table and column names will appear in lower case; Chapter 8 explains why.

```
\dt
```

Kết quả chạy / Output

List of relations			
Schema	Name	Type	Owner
public	cuahang	table	postgres
public	giaodich	table	postgres
public	khachhang	table	postgres
public	khoanvay	table	postgres
public	kytrano	table	postgres
public	lienket	table	postgres
public	nganhang	table	postgres
public	nhanvien	table	postgres
public	vi	table	postgres

(9 rows)

Kết quả có 9 bảng. Cột Name là tên bảng, cột Owner là tài khoản sở hữu. Bây giờ ta thử đọc dữ liệu thật: lệnh SELECT lấy các dòng ra, FROM chỉ bảng, WHERE giữ lại những dòng thỏa điều kiện. Lệnh thứ hai đếm số dòng của bảng khách hàng. Mỗi lệnh hỏi dữ liệu kiểu này gọi là một truy vấn (query).

The result shows 9 tables. The Name column is the table name and the Owner column is the account that owns it. Now let us read real data: SELECT picks rows out, FROM names the table, and WHERE keeps only the rows that meet a condition. The second command counts the rows of the customer table. Each command that asks the data a question like this is called a query.

```
SELECT MaKH, HoTen, TinhThanh FROM KhachHang WHERE TinhThanh = 'Huế';
SELECT COUNT(*) AS soKhach FROM KhachHang;
```

Kết quả chạy / Output

```

 makh |      hoten      | tinhthanh
-----+-----+-----
 KH001 | Trần Minh Khôi | Huế
(1 row)

 sokhach
-----
      12
(1 row)
```

Trong 12 khách hàng có đúng 1 người ở Huế, là KH001. Từ nay mọi con số về CSDL Trường Tiền trong sách đều lấy từ kết quả chạy thật như vậy.

Of the 12 customers exactly 1 lives in Huế, KH001. From here on every figure about the Trường Tiền database in this book comes from a real run like this.

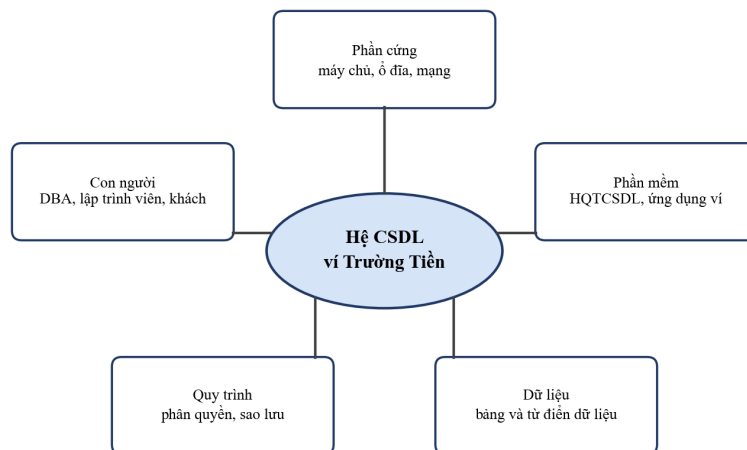
Còn ai lo việc cất, tìm và bảo vệ các bảng đó? Đó là hệ quản trị cơ sở dữ liệu (HQTCSDL, database management system, DBMS): phần mềm đứng giữa người dùng, chương trình và dữ liệu. Mọi yêu cầu đọc hay ghi đều đi qua nó. Hãy hình dung thủ thư của một thư viện lớn: bạn không tự vào kho lấy sách, bạn nói với thủ thư, thủ thư tìm, kiểm tra và đưa sách cho bạn. HQTCSDL là người thủ thư đó. PostgreSQL, MySQL, SQL Server, Oracle và SQLite là các HQTCSDL.

Who looks after storing, finding and protecting those tables? The database management system (DBMS): software that stands between users, programs and the data. Every request to read or write goes through it. Picture the librarian of a large library: you do not walk into the stacks yourself, you tell the librarian, who finds the book, checks your card and hands it over. The DBMS is that librarian. PostgreSQL, MySQL, SQL Server, Oracle and SQLite are DBMSs.

Một HQTCSDL đứng một mình chưa làm được gì. Để ví Trường Tiền chạy, cần thêm máy chủ, ứng dụng, quy định vận hành và con người. Tất cả gộp lại gọi là hệ cơ sở dữ liệu (database system) và có năm thành phần.

A DBMS standing alone does nothing. For the Trường Tiền wallet to run, you also need servers, applications, working rules and people. All of these together are called a database system, and it has five parts.

Thành phần <i>Part</i>	Là gì <i>What it is</i>	Ở ví Trường Tiền <i>At Trường Tiền</i>
Phần cứng <i>Hardware</i>	Máy móc chạy hệ thống: máy chủ, ổ đĩa, thiết bị mạng. <i>The machines the system runs on: servers, disks, network equipment.</i>	Máy chủ đặt dữ liệu và đường truyền tới điện thoại của khách. <i>The server holding the data and the connection to customers' phones.</i>
Phần mềm <i>Software</i>	HQTCSDL cùng các chương trình ứng dụng dùng nó. <i>The DBMS and the application programs that use it.</i>	PostgreSQL và ứng dụng ví trên điện thoại. <i>PostgreSQL and the wallet app on the phone.</i>
Dữ liệu <i>Data</i>	CSDL và phần mô tả cấu trúc của chính nó (từ điển dữ liệu). <i>The database and the description of its own structure (the data dictionary).</i>	Chín bảng khách hàng, ví, giao dịch và các bảng còn lại. <i>The nine tables of customers, wallets, transactions and the rest.</i>
Quy trình <i>Procedures</i>	Quy định về cách dùng, cấp quyền, sao lưu, xử lý sự cố. <i>Rules on use, granting access, backup and handling incidents.</i>	Ai được xem số điện thoại, bao lâu sao lưu một lần. <i>Who may see phone numbers, and how often backups run.</i>
Con người <i>People</i>	Những người thiết kế, quản trị, lập trình và sử dụng hệ thống. <i>The people who design, administer, program and use the system.</i>	Người quản trị, lập trình viên, nhân viên CSKH và khách hàng. <i>The administrator, programmers, CSKH staff and customers.</i>



Hình 1.2. Năm thành phần của hệ CSDL. Elip ở giữa là hệ thống ví Trường Tiền; năm ô xung quanh là phần cứng, phần mềm, dữ liệu, quy trình và con người, mỗi ô nối vào hệ thống bằng một đường thẳng.

Figure 1.2. The five parts of a database system. The ellipse in the middle is the Trường Tiền wallet system; the five boxes around it are hardware, software, data, procedures and people, each joined to the system by a straight line.

Khái niệm <i>Concept</i>	Là gì <i>What it is</i>	Ví dụ <i>Example</i>	Dễ nhầm ở đâu <i>Where people slip</i>
CSDL <i>Database</i>	Tập dữ liệu có tổ chức, dùng chung. <i>An organised set of shared data.</i>	Chín bảng của ví Trường Tiền. <i>The nine tables of the Trường Tiền wallet.</i>	Là dữ liệu, không phải phần mềm. <i>It is data, not software.</i>
HQTCSDL <i>DBMS</i>	Phần mềm quản lý và bảo vệ CSDL. <i>Software that manages and protects the database.</i>	PostgreSQL. <i>PostgreSQL.</i>	Cài đặt được, nhưng chưa có dữ liệu của bạn. <i>You can install it, but it does not yet hold your data.</i>
Hệ CSDL <i>Database system</i>	CSDL, HQTCSDL, ứng dụng, phần cứng, quy trình và con người. <i>The database, DBMS, applications, hardware, procedures and people.</i>	Toàn bộ hệ thống ví của công ty. <i>The company's whole wallet system.</i>	Rộng hơn hai khái niệm trên. <i>It is wider than the other two.</i>

PHẢI THUỘC / MEMORISE – CSDL là tập dữ liệu có tổ chức, dùng chung. HQTCSDL là phần mềm quản lý CSDL. Hệ CSDL gồm CSDL, HQTCSDL, ứng dụng, phần cứng, quy trình và con người.

A database is an organised set of shared data. A DBMS is software that manages it. A database system is the database, DBMS, applications, hardware, procedures and people together.

LỖI HAY GẶP / COMMON MISTAKE – Gọi PostgreSQL là "một cơ sở dữ liệu". PostgreSQL là HQTCSDL; cơ sở dữ liệu là thứ nó quản lý, ví dụ CSDL Trường Tiền.

Calling PostgreSQL "a database". PostgreSQL is a DBMS; the database is what it manages, such as the Trường Tiền database.

LỖI HAY GẶP / COMMON MISTAKE – Nghĩ rằng một bảng Excel nhiều sheet đã là hệ CSDL. Excel là bảng tính; nó có thể chứa dữ liệu có tổ chức nhưng thiếu phần giữ ràng buộc, phân quyền theo cột và xử lý nhiều người cùng ghi.

Thinking a multi-sheet Excel workbook is already a database system. Excel is a spreadsheet; it can hold organised data but lacks constraint checking, column-level permissions and handling of many writers at once.

1.4. Đặc trưng và lợi ích của CSDL – 1.4. Features and benefits of a database

Mục 1.2 liệt kê các vấn đề của tệp rời. CSDL có tám đặc trưng, và mỗi đặc trưng là câu trả lời cho một hay nhiều vấn đề đó. Ta đi lần lượt, đặc trưng nào làm thử được thì chạy lệnh thật.

Section 1.2 listed the problems of separate files. A database has eight features, and each answers one or more of those problems. We take them in turn, and run real commands where a feature can be tried.

Đặc trưng 1. Tự mô tả – Feature 1. Self-describing

Ngoài dữ liệu, CSDL giữ cả bản mô tả về chính nó: bảng nào có cột gì, kiểu gì, quy tắc gì. Phần mô tả đó gọi là siêu dữ liệu (metadata, nghĩa là dữ liệu về dữ liệu) và nơi cất nó gọi là từ điển dữ liệu (data dictionary). Nhờ vậy ai mở CSDL cũng đọc được cấu trúc mà không cần hỏi người viết ra nó. Lệnh \d kèm tên bảng cho ta xem phần mô tả của bảng đó.

Besides data, the database keeps a description of itself: which tables have which columns, of what type, under what rules. That description is called metadata (data about data), and the place that holds it is the data dictionary. So anyone who opens the database can read its structure without asking whoever built it. The command \d followed by a table name shows the description of that table.

```
\d lienket
```

Kết quả chạy / Output

```

Table "public.lienket"
Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
mavi   | character(5)           |           | not null |
manh   | character(4)           |           | not null |
sotaikhoan | character varying(20) |           | not null |
ngaylienket | date                  |           | not null |
Indexes:
  "pklienket" PRIMARY KEY, btree (mavi, manh)
Foreign-key constraints:
  "fklknganhang" FOREIGN KEY (manh) REFERENCES nganhang(manh)
  "fklkvi" FOREIGN KEY (mavi) REFERENCES vi(mavi)

```

Cách đọc kết quả: phần đầu liệt kê từng cột với kiểu dữ liệu (character(4) là chuỗi đúng 4 ký tự, date là ngày) và cột có được để trống hay không (not null là không). Phần Indexes cho biết khóa chính, tức tập cột nhận diện duy nhất mỗi dòng; ở bảng này là cặp mã ví và mã ngân hàng. Phần Foreign-key constraints cho biết hai cột này tham chiếu sang bảng ví và bảng ngân hàng. Bạn sẽ học các khái niệm ấy ở Chương 2 và 4. Điều cần thấy lúc này: câu trả lời nằm sẵn trong CSDL.

How to read the result: the top part lists each column with its data type (character(4) is a string of exactly 4 characters, date is a date) and whether it may be left empty (not null means it may not). The Indexes part shows the primary key, the set of columns that identifies each row uniquely; in this table it is the pair of wallet code and bank code. The Foreign-key constraints part shows that these two columns refer to the wallet table and the bank table. You will meet these ideas in Chapters 2 and 4. What to see now: the answer is stored in the database itself.

Ta cũng hỏi từ điển dữ liệu bằng SELECT thường: mỗi bảng có bao nhiêu cột.

We can also question the data dictionary with an ordinary SELECT: how many columns each table has.

```

SELECT table_name AS bang, COUNT(*) AS soCot
FROM information_schema.columns
WHERE table_schema = 'public'
GROUP BY table_name
ORDER BY table_name;

```

Kết quả chạy / Output

bang	socot
cuahang	4
giaodich	9
khachhang	8
khoanvay	8
kytrano	6
lienket	4
nganhang	2
nhanvien	5
vi	7
(9 rows)	

Bảng giao dịch có 9 cột, bảng khách hàng 8 cột, bảng ngân hàng chỉ 2 cột. Số liệu này không do ai gõ vào; HQTCSDL tự biết vì nó giữ cấu trúc của chính mình.

The transaction table has 9 columns, the customer table 8 and the bank table just 2. Nobody typed these figures in; the DBMS knows them because it holds its own structure.

Đặc trưng 2. Cách ly giữa chương trình và dữ liệu – Feature 2. Insulation between programs and data

Với tệp rời, cấu trúc tệp nằm trong chương trình; thêm một cột là sửa chương trình. Với CSDL, cấu trúc nằm trong từ điển dữ liệu, ngoài chương trình. Tính chất này gọi là tính độc lập giữa chương trình và dữ liệu (program-data independence). Thử xem: ta thêm cột Website vào bảng ngân hàng, rồi chạy lại đúng câu truy vấn cũ.

With separate files, the file layout lives inside the program, so adding a column means changing the program. With a database, the layout lives in the data dictionary, outside the program. This property is called program-data independence. See it at work: we add a Website column to the bank table, then run the old query again unchanged.

```
ALTER TABLE NganHang ADD COLUMN Website varchar(40);
SELECT MaNH, TenNH FROM NganHang ORDER BY MaNH;
```

Kết quả chạy / Output

manh	tennh
NH01	Ngân hàng Đông Ba
NH02	Ngân hàng Kim Long
NH03	Ngân hàng Phú Xuân
NH04	Ngân hàng Hương Bình
(4 rows)	

Câu SELECT không nhắc gì đến cột mới và vẫn cho kết quả cũ, đủ 4 ngân hàng. Chương trình cũ không phải sửa.

The SELECT says nothing about the new column and still gives the same result, all 4 banks. The old program does not have to be changed.

Đặc trưng 3. Nhiều khung nhìn – Feature 3. Multiple views

Mỗi nhóm người cần thấy một phần khác nhau của dữ liệu. Khung nhìn (view) là một "cửa sổ" định sẵn trên CSDL, chỉ cho thấy những cột và dòng nhóm đó cần. Khung nhìn không chép dữ liệu; nó lưu câu hỏi, và mỗi lần mở lại chạy câu hỏi ấy trên dữ liệu hiện

có. Ta tạo khung nhìn cho nhân viên lễ tân: có mã, họ tên, tỉnh, hạng; không có số điện thoại.

Each group of people needs to see a different part of the data. A view is a ready-made "window" onto the database that shows only the columns and rows that group needs. A view does not copy data; it stores a question, and each time it is opened it runs that question on the current data. Here we create a view for front-desk staff: code, name, province and tier, but no phone number.

```
CREATE VIEW KhachLeTan AS
SELECT MaKH, HoTen, TinhThanh, Hang FROM KhachHang;
SELECT * FROM KhachLeTan WHERE Hang = 'Vàng';
```

Kết quả chạy / Output

```
CREATE VIEW
makh | hoten | tinhthanh | hang
-----+-----+-----+-----
KH001 | Trần Minh Khôi | Huế | Vàng
KH003 | Phạm Quốc Bảo | Hà Nội | Vàng
KH009 | Đinh Công Phúc | Hải Phòng | Vàng
(3 rows)
```

Có 3 khách hạng Vàng, và người dùng khung nhìn này không thấy cột số điện thoại. Chương 8 dạy tạo khung nhìn, Chương 10 dùng nó để che dữ liệu nhạy cảm.

There are 3 Gold customers, and a user of this view does not see the phone number column. Chapter 8 teaches how to create views, and Chapter 10 uses them to hide sensitive data.

Năm đặc trưng còn lại – The remaining five features

Ba đặc trưng đầu liên quan đến cách dữ liệu được tổ chức và trình bày. Năm đặc trưng sau nói về an toàn và độ tin cậy; mỗi cái được dạy kỹ ở chương sau, ở đây chỉ cần hiểu ý.

The first three features concern how data is organised and presented. The next five are about safety and reliability; each is taught in detail in a later chapter, so here you only need the idea.

Đặc trưng Feature	Ý nghĩa Meaning	Ví dụ ở Trường Tiền Example at Trường Tiền	Chương Chapter
4. Chia sẻ và nhiều người dùng đồng thời <i>4. Sharing and many users at once</i>	Nhiều người cùng đọc, cùng ghi mà dữ liệu không bị hỏng, nhờ cơ chế kiểm soát đồng thời. <i>Many people read and write together without the data being damaged, thanks to concurrency control.</i>	Hai khách cùng chuyển tiền vào một ví trong cùng giây; số dư cuối vẫn đúng. <i>Two customers send money to one wallet in the same second; the final balance is still right.</i>	10
5. Kiểm soát dư thừa <i>5. Controlled redundancy</i>	Mỗi sự việc ghi một chỗ; nếu có lặp thì lặp có chủ đích và được kiểm soát. <i>Each fact is stored once; any repetition is deliberate and controlled.</i>	Số điện thoại của khách chỉ nằm trong bảng khách hàng. Cột số dư của ví là dữ liệu tính được từ giao dịch nhưng lưu sẵn để tra nhanh: một chỗ lặp có chủ đích. <i>A customer's phone number sits only in the customer table. The</i>	6, 7

Đặc trưng <i>Feature</i>	Ý nghĩa <i>Meaning</i>	Ví dụ ở Trường Tiền <i>Example at Trường Tiền</i>	Chương <i>Chapter</i>
		<i>wallet balance column can be worked out from transactions but is kept ready for fast lookup: a deliberate repetition.</i>	
6. Ràng buộc toàn vẹn <i>6. Integrity constraints</i>	Quy tắc kiểm tra dữ liệu nằm trong CSDL, mọi chương trình đều bị kiểm. <i>Rules that check the data live in the database, so every program is checked.</i>	Số dư ví không âm; giao dịch phải thuộc một ví có thật. <i>A wallet balance cannot be negative; a transaction must belong to a real wallet.</i>	2, 4
7. Bảo mật, phân quyền <i>7. Security and access rights</i>	Mỗi tài khoản chỉ làm được những việc được cấp. <i>Each account can do only what it has been granted.</i>	Phòng Kế toán xem được giao dịch nhưng không xem được số điện thoại. <i>Accounting can see transactions but not phone numbers.</i>	10
8. Sao lưu và phục hồi <i>8. Backup and recovery</i>	Có cách chép dữ liệu ra nơi khác và đưa nó trở lại sau sự cố. <i>There is a way to copy the data elsewhere and bring it back after a failure.</i>	Ổ đĩa máy chủ hỏng; công ty dựng lại CSDL từ bản sao lưu gần nhất. <i>The server disk fails; the company rebuilds the database from the latest backup.</i>	10

Đặc trưng 4 cần giải thích thêm một chút. Kiểm soát đồng thời quyết định cái gì xảy ra khi hai yêu cầu chạm vào cùng một dòng: HQTCSĐL xếp chúng vào hàng và cho từng yêu cầu chạy trọn vẹn thay vì để chúng giẫm lên nhau. Chương 10 sẽ dùng ví dụ chuyển tiền giữa hai ví để cho thấy việc đó.

Feature 4 needs a little more explaining. Concurrency control decides what happens when two requests touch the same row: the DBMS lines them up and lets each run in full instead of letting them trample each other. Chapter 10 uses a transfer between two wallets to show this.

Ghép lại, mỗi vấn đề ở mục 1.2 đã có lời đáp:

Putting it together, every problem in section 1.2 now has an answer:

Vấn đề của tệp rời <i>Problem of separate files</i>	Cách CSDL giải quyết <i>How a database solves it</i>
Dư thừa, không nhất quán <i>Redundancy and inconsistency</i>	Mỗi sự việc ghi một chỗ (đặc trưng 5). <i>Each fact is stored once (feature 5).</i>
Khó truy xuất <i>Data isolation</i>	Mọi dữ liệu liên quan nằm trong một CSDL; một câu SQL hỏi được nhiều bảng (đặc trưng 1, 2). <i>All related data is in one database, and a single SQL statement can ask across tables (features 1, 2).</i>

Vấn đề của tệp rời <i>Problem of separate files</i>	Cách CSDL giải quyết <i>How a database solves it</i>
Khó giữ ràng buộc <i>Integrity</i>	Quy tắc nằm trong CSDL (đặc trưng 6). <i>Rules live in the database (feature 6).</i>
Bảo mật yếu <i>Weak security</i>	Phân quyền theo bảng, cột, khung nhìn (đặc trưng 3, 7). <i>Permissions by table, column and view (features 3, 7).</i>
Khó chia sẻ, truy cập đồng thời <i>Hard to share, concurrent access</i>	Kiểm soát đồng thời (đặc trưng 4). <i>Concurrency control (feature 4).</i>
Phụ thuộc cấu trúc tệp <i>Dependence on file layout</i>	Cách ly chương trình và dữ liệu (đặc trưng 2). <i>Program-data insulation (feature 2).</i>

LỖI HAY GẶP / COMMON MISTAKE – Cho rằng dùng CSDL là hết dư thừa. CSDL kiểm soát dư thừa chứ không bỏ hẳn; cột số dư lưu sẵn là ví dụ về một chỗ lặp có chủ đích, và nó phải được giữ khớp với giao dịch.

Believing that using a database ends redundancy. A database controls redundancy rather than abolishing it; the stored balance column is a deliberate repetition and must be kept in step with the transactions.

LỖI HAY GẶP / COMMON MISTAKE – Nghĩ khung nhìn là một bản sao dữ liệu. Khung nhìn chỉ lưu câu hỏi; sửa dữ liệu gốc thì khung nhìn thấy ngay thay đổi.

Thinking a view is a copy of the data. A view stores only a question; change the underlying data and the view shows the change at once.

1.5. Chức năng của HQTCSĐL – 1.5. What a DBMS does

Nếu coi HQTCSĐL là người thủ thư của kho dữ liệu, bảng sau liệt kê công việc của người đó. Ngôn ngữ để ra lệnh cho HQTCSĐL thường là SQL, đã gặp ở mục 1.2. SQL chia lệnh thành ba nhóm: DDL (Data Definition Language) dùng để dựng và sửa cấu trúc, DML (Data Manipulation Language) dùng để thêm, sửa, xóa, đọc dữ liệu, và DCL (Data Control Language) dùng để cấp hoặc thu hồi quyền.

If the DBMS is the librarian of the data store, the next table lists the librarian's jobs. The language for giving orders to a DBMS is usually SQL, which we met in section 1.2. SQL sorts its commands into three groups: DDL (Data Definition Language) builds and changes the structure, DML (Data Manipulation Language) adds, changes, deletes and reads data, and DCL (Data Control Language) grants or withdraws permissions.

Chức năng <i>Function</i>	Việc HQTCSĐL làm <i>What the DBMS does</i>	Ví dụ ở Trường Tiền <i>Example at Trường Tiền</i>
Định nghĩa dữ liệu <i>Defining data</i>	Cho phép khai báo bảng, cột, kiểu dữ liệu, quy tắc (DDL). <i>Lets you declare tables, columns, data types and rules (DDL).</i>	Tạo bảng ví với quy tắc số dư không âm. <i>Create the wallet table with a no-negative-balance rule.</i>

Chức năng <i>Function</i>	Việc HQTCSDL làm <i>What the DBMS does</i>	Ví dụ ở Trường Tiền <i>Example at Trường Tiền</i>
Lưu trữ và quản lý cách lưu <i>Storing data</i>	Quyết định dữ liệu nằm ở đâu trên đĩa, giữ chỉ mục (giống mục lục cuối sách) để tìm nhanh. <i>Decides where data sits on disk and keeps indexes (like the index at the back of a book) for quick search.</i>	Tìm một giao dịch trong hàng triệu dòng mà không đọc hết. <i>Find one transaction among millions of rows without reading them all.</i>
Truy vấn <i>Querying</i>	Trả lời câu hỏi: lọc, sắp xếp, ghép bảng, tính tổng, tính toán (DML). <i>Answers questions: filtering, sorting, joining tables, totals, calculations (DML).</i>	Tổng tiền nạp của tháng 9. <i>Total top-ups for September.</i>
Cập nhật <i>Updating</i>	Thêm, sửa, xóa dữ liệu (DML). <i>Adds, changes and deletes data (DML).</i>	Ghi một giao dịch mới, đổi số điện thoại khách. <i>Record a new transaction, change a customer's phone number.</i>
Giữ ràng buộc <i>Enforcing constraints</i>	Từ chối dữ liệu vi phạm quy tắc. <i>Rejects data that breaks the rules.</i>	Chặn rút quá số dư. <i>Block a withdrawal larger than the balance.</i>
Bảo mật <i>Security</i>	Xác thực tài khoản, cấp và thu quyền (DCL). <i>Verifies accounts and grants or withdraws permissions (DCL).</i>	Cho Kế toán đọc bảng giao dịch, không cho sửa. <i>Let Accounting read the transaction table but not change it.</i>
Kiểm soát đồng thời và giao tác <i>Concurrency and transactions</i>	Cho nhiều người dùng chung, bảo đảm một giao tác, tức một chuỗi thao tác coi như một khối, hoặc xong cả chuỗi hoặc không gì cả. <i>Lets many users share data and ensures a transaction, a chain of operations treated as one block, either completes in full or not at all.</i>	Chuyển tiền: trừ ví này và cộng ví kia cùng thành công hoặc cùng hủy. <i>A transfer: debiting one wallet and crediting the other both succeed or both are cancelled.</i>
Sao lưu và phục hồi <i>Backup and recovery</i>	Chép dữ liệu ra nơi khác, đưa nó trở lại sau sự cố. <i>Copies data elsewhere and restores it after a failure.</i>	Phục hồi bảng giao dịch bị xóa nhầm. <i>Restore a transaction table deleted by mistake.</i>
Duy trì từ điển dữ liệu <i>Maintaining the data dictionary</i>	Ghi và cập nhật siêu dữ liệu mỗi khi cấu trúc đổi. <i>Records and updates metadata whenever the structure changes.</i>	Thêm cột mới thì từ điển tự biết. <i>When a column is added, the dictionary knows.</i>

Chức năng truy vấn đáng xem kỹ, vì nó làm trong một lệnh điều mà với tệp rời phải làm bằng tay: lọc, nhóm, đếm, cộng. Lệnh sau tính, với các giao dịch thành công, mỗi loại giao dịch có bao nhiêu lượt và tổng bao nhiêu tiền.

The query function deserves a close look, because in one command it does what separate files force you to do by hand: filter, group, count, add. The next command works out, for successful transactions, how many there are of each type and how much money they total.

```
SELECT LoaiGD, COUNT(*) AS soGD, SUM(SoTien) AS tong
FROM GiaoDich
WHERE TrangThai = 'Thành công'
GROUP BY LoaiGD
ORDER BY LoaiGD;
```

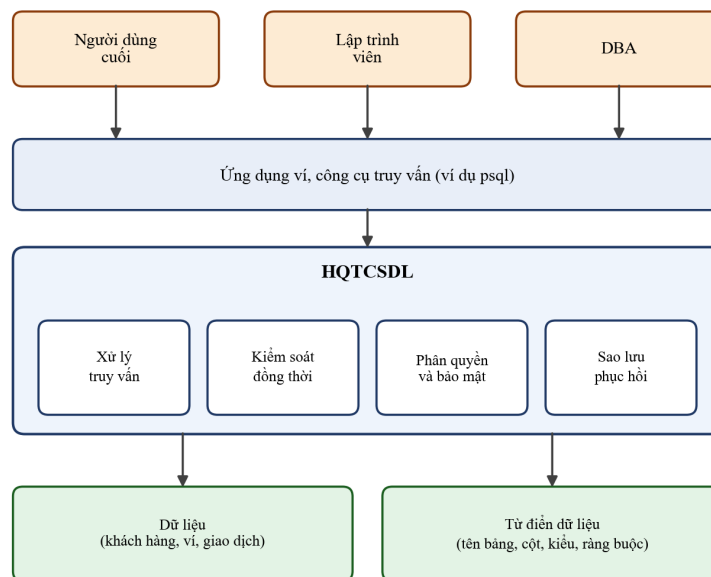
Kết quả chạy / Output

loaigd	sogd	tong
Chuyển	6	4900000
Nạp	11	17500000
Rút	5	6800000
Thanh toán	15	9020000

(4 rows)

Đọc bảng kết quả: có 11 lượt nạp thành công với tổng 17.500.000 đ, chính là con số ta dùng làm "thông tin" ở mục 1.1. Có 5 lượt rút thành công, 6.800.000 đ. Mỗi lượt rút bị thu 3.300 đ, nên 5 lượt thu tổng 16.500 đ phí.

Reading the result: there are 11 successful top-ups totalling VND 17,500,000, the very figure we used as "information" in section 1.1. There are 5 successful withdrawals, VND 6,800,000. Each withdrawal is charged VND 3,300, so the 5 withdrawals bring in VND 16,500 in fees.



Hình 1.3. Đường đi của một yêu cầu. Người dùng, lập trình viên và DBA làm việc qua ứng dụng hoặc công cụ truy vấn; mọi lệnh đi vào HQTCSDL, ở đó được xử lý, kiểm quyền, kiểm soát đồng thời rồi mới chạm tới dữ liệu và từ điển dữ liệu ở dưới cùng.

Figure 1.3. The path of a request. Users, programmers and the DBA work through an application or query tool; every command enters the DBMS, where it is processed, checked for permission and for concurrency before reaching the data and the data dictionary at the bottom.

Hành trình của một câu lệnh – *The journey of one command*

1. Ứng dụng hoặc người dùng gửi một câu lệnh SQL tới HQTCSĐL.
An application or user sends an SQL command to the DBMS.
2. HQTCSĐL kiểm tra tài khoản gửi lệnh có quyền làm việc này không. Không có quyền thì lệnh bị từ chối ngay.
The DBMS checks whether the sending account has the right to do this. Without the right, the command is refused at once.
3. HQTCSĐL tra từ điển dữ liệu để biết bảng, cột nào có thật, kiểu gì, và chọn cách tìm dữ liệu nhanh nhất.
The DBMS looks in the data dictionary to learn which tables and columns exist and of what type, and picks the quickest way to find the data.
4. HQTCSĐL đọc hoặc ghi dữ liệu, kiểm các ràng buộc và giữ chỗ cho những yêu cầu khác đang chạy cùng lúc.
The DBMS reads or writes the data, checks the constraints and sets the right locks against other requests running at the same time.
5. HQTCSĐL trả kết quả hoặc thông báo lỗi về cho nơi gửi lệnh.
The DBMS returns the result or an error message to whoever sent the command.

VÍ DỤ / EXAMPLE – Nhân viên CSKH gõ lệnh đếm khách hàng Vàng. Bước 2: tài khoản CSKH có quyền đọc bảng khách hàng nên được qua. Bước 3: HQTCSĐL thấy bảng khách hàng có cột hang. Bước 4: đọc 12 dòng, giữ 3 dòng hạng Vàng. Bước 5: trả số 3. Nghĩa là mỗi lệnh đi qua cùng một chuỗi, và người gõ lệnh chỉ thấy kết quả cuối.

A CSKH employee types a command to count Gold customers. Step 2: the CSKH account may read the customer table, so it passes. Step 3: the DBMS sees that the customer table has a column hang. Step 4: it reads 12 rows and keeps the 3 Gold ones. Step 5: it returns 3. So every command goes through the same chain, and the person typing sees only the final result.

LỖI HAY GẶP / COMMON MISTAKE – Nghĩ SQL là một HQTCSĐL. SQL là ngôn ngữ để ra lệnh; PostgreSQL, MySQL, SQL Server là các HQTCSĐL hiểu ngôn ngữ ấy.

Thinking SQL is a DBMS. SQL is a language for giving orders; PostgreSQL, MySQL and SQL Server are DBMSs that understand it.

1.6. Phân loại HQTCSĐL – *1.6. Classifying DBMSs*

Có hàng chục HQTCSĐL, và chúng khác nhau ở nhiều điểm. Để chọn đúng, người ta xếp chúng theo bốn tiêu chí. Mỗi tiêu chí trả lời một câu hỏi về bài toán của bạn.

There are dozens of DBMSs, and they differ in many ways. To choose well, people sort them by four criteria. Each criterion answers a question about your problem.

Tiêu chí 1. Theo mô hình dữ liệu – *Criterion 1. By data model*

Mô hình dữ liệu (data model) là cách quy ước để tổ chức dữ liệu và nối các phần dữ liệu với nhau. Mô hình quan hệ xếp dữ liệu vào bảng gồm dòng và cột, các bảng nối với nhau qua giá trị chung. Đây là mô hình phổ biến nhất và là mô hình sách này dạy; PostgreSQL, MySQL, SQL Server, Oracle và SQLite đều theo mô hình này. Trước nó có mô hình

phân cấp (dữ liệu xếp thành cây) và mô hình mạng (dữ liệu nối thành lưới); hai mô hình này còn trong một số hệ thống cũ. Sau nó có mô hình hướng đối tượng và nhóm NoSQL, như MongoDB (lưu từng "tài liệu" gồm các cặp tên và giá trị) hay Redis (lưu cặp khóa và giá trị). Chương 2 so sánh các mô hình kỹ hơn.

A data model is an agreed way to organise data and to link pieces of data together. The relational model arranges data into tables of rows and columns, with tables linked through shared values. It is the most widespread model and the one this book teaches; PostgreSQL, MySQL, SQL Server, Oracle and SQLite all follow it. Before it came the hierarchical model (data arranged as a tree) and the network model (data joined as a web); both survive in some older systems. After it came the object-oriented model and the NoSQL group, such as MongoDB (which stores whole "documents" made of name and value pairs) or Redis (which stores pairs of key and value). Chapter 2 compares the models in more depth.

Tiêu chí 2. Theo số người dùng – Criterion 2. By number of users

HQTCSDL một người dùng (single-user) phục vụ một người tại một thời điểm, như cơ sở dữ liệu nhỏ nằm trong một ứng dụng cài trên máy cá nhân; SQLite thường được dùng theo kiểu này. HQTCSDL nhiều người dùng (multiuser) phục vụ nhiều người cùng lúc và phải có kiểm soát đồng thời; ví điện tử luôn thuộc loại này.

A single-user DBMS serves one person at a time, such as a small database inside an application installed on a personal computer; SQLite is often used this way. A multiuser DBMS serves many people at once and needs concurrency control; an e-wallet always belongs to this type.

Tiêu chí 3. Theo nơi đặt dữ liệu – Criterion 3. By where the data sits

Hệ tập trung (centralised) đặt HQTCSDL và dữ liệu ở một nơi; nhiều chương trình và người dùng ở xa kết nối tới. Hệ phân tán (distributed) rải dữ liệu ra nhiều nơi nối với nhau bằng mạng. Nếu mọi nơi dùng cùng một HQTCSDL thì hệ phân tán là đồng nhất (homogeneous); nếu mỗi nơi dùng một loại HQTCSDL và có thêm lớp phần mềm chung để trao đổi thì hệ không đồng nhất (heterogeneous). Giả sử Trường Tiền đặt một máy chủ ở Huế và một ở Hà Nội, cả hai chạy PostgreSQL và đồng bộ với nhau: đó là hệ phân tán đồng nhất.

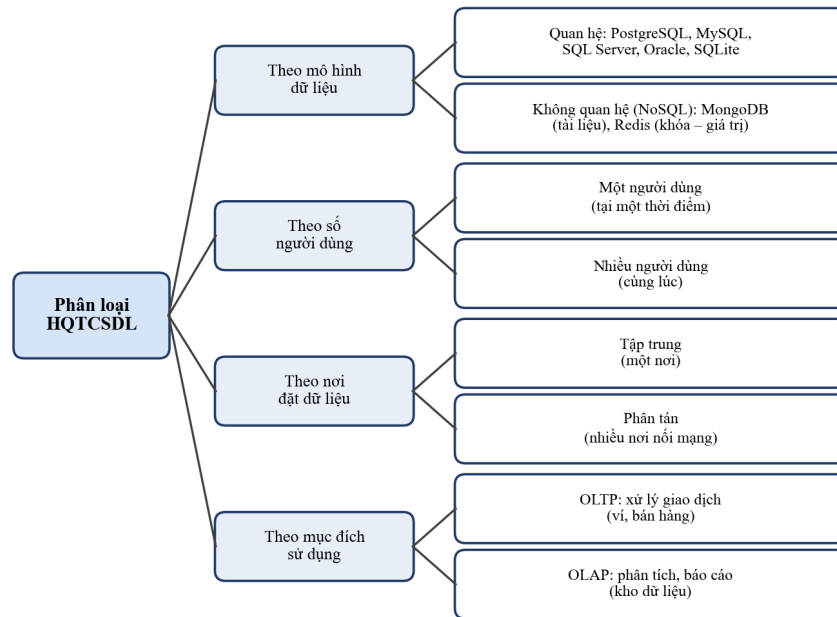
A centralised system puts the DBMS and the data in one place; many programs and users connect to it from afar. A distributed system spreads the data over several places joined by a network. If every place runs the same DBMS, the distributed system is homogeneous; if each place runs a different DBMS and an extra layer of common software handles the exchange, it is heterogeneous. Suppose Trường Tiền kept one server in Huế and another in Hà Nội, both running PostgreSQL and synchronised: that would be a homogeneous distributed system.

Tiêu chí 4. Theo mục đích sử dụng – Criterion 4. By purpose

Hệ OLTP (online transaction processing, xử lý giao dịch trực tuyến) phục vụ công việc hằng ngày: ghi từng giao dịch nhỏ, nhanh, nhiều lần, như mỗi lần nạp hay chuyển tiền. Hệ OLAP (online analytical processing, xử lý phân tích trực tuyến) phục vụ phân tích: đọc khối dữ liệu lớn, tổng hợp qua nhiều năm để làm báo cáo, thường nằm trong kho dữ liệu (data warehouse), nơi gom dữ liệu từ nhiều hệ thống để phân tích. Ví Trường Tiền cần hệ OLTP để chạy ví và hệ OLAP để làm báo cáo quý cho ban giám đốc.

An OLTP system (online transaction processing) serves daily work: recording many small transactions, quickly, over and over, such as each top-up or transfer. An OLAP system (online analytical processing) serves analysis: reading large volumes of data and summarising across years for reports, usually sitting in a data warehouse, a place that gathers data from many systems for analysis. The Trường Tiền wallet needs an OLTP system to run the wallet and an OLAP system to produce the quarterly report for the board.

Tiêu chí <i>Criterion</i>	Các loại <i>Types</i>	Ví dụ <i>Examples</i>	Ví Trường Tiền chọn <i>Trường Tiền's choice</i>
Mô hình dữ liệu <i>Data model</i>	Quan hệ; phân cấp; mạng; hướng đối tượng; NoSQL (tài liệu, khóa và giá trị) <i>Relational; hierarchical; network; object-oriented; NoSQL (document, key-value)</i>	PostgreSQL, MySQL, SQL Server, Oracle, SQLite (quan hệ); MongoDB (tài liệu); Redis (khóa và giá trị) <i>PostgreSQL, MySQL, SQL Server, Oracle, SQLite (relational); MongoDB (document); Redis (key-value)</i>	Quan hệ <i>Relational</i>
Số người dùng <i>Number of users</i>	Một người dùng; nhiều người dùng <i>Single-user; multiuser</i>	SQLite thường dùng kiểu một người; PostgreSQL nhiều người <i>SQLite is often used single-user; PostgreSQL is multiuser</i>	Nhiều người dùng <i>Multiuser</i>
Nơi đặt dữ liệu <i>Where the data sits</i>	Tập trung; phân tán (đồng nhất, không đồng nhất) <i>Centralised; distributed (homogeneous, heterogeneous)</i>	Máy chủ duy nhất; hai máy chủ cùng PostgreSQL <i>A single server; two servers both running PostgreSQL</i>	Tập trung <i>Centralised</i>
Mục đích <i>Purpose</i>	OLTP; OLAP <i>OLTP; OLAP</i>	Chạy ví hằng ngày; kho dữ liệu cho báo cáo <i>Running the wallet daily; a data warehouse for reports</i>	OLTP cho ví <i>OLTP for the wallet</i>



Hình 1.4. Cây phân loại HQTCSDL. Từ ô gốc bên trái tỏa ra bốn tiêu chí, mỗi tiêu chí có hai nhóm ở cột bên phải kèm ví dụ.

Figure 1.4. The classification tree of DBMSs. Four criteria branch from the root box on the left, and each has two groups in the right-hand column with examples.

Cách chọn loại HQTCSDL cho một bài toán – *How to choose the type of DBMS for a problem*

1. Xác định dữ liệu có dạng bảng với các cột cố định không. Nếu có và các thực thể liên quan chặt chẽ với nhau, chọn mô hình quan hệ.
Decide whether the data fits tables with fixed columns. If it does and the things in it are closely related, choose the relational model.
2. Đếm số người hoặc chương trình ghi dữ liệu cùng lúc. Một thì có thể dùng hệ một người dùng; nhiều thì bắt buộc nhiều người dùng.
Count the people or programs writing at the same time. One may do with a single-user system; several require a multiuser one.
3. Hỏi dữ liệu có phải đặt ở nhiều nơi vì lý do địa lý, pháp lý hay chịu tải không. Không thì chọn tập trung cho đơn giản.
Ask whether the data must sit in several places for reasons of geography, law or load. If not, choose centralised for simplicity.
4. Hỏi công việc chính là ghi nhiều giao dịch nhỏ (OLTP) hay đọc khối lớn để tổng hợp (OLAP).
Ask whether the main job is writing many small transactions (OLTP) or reading large blocks to summarise (OLAP).

VÍ DỤ / EXAMPLE – Một thư viện trường có một máy ở quầy, hai thủ thư cùng mượn trả sách, dữ liệu gồm sách, bạn đọc, phiếu mượn. Bước 1: dữ liệu dạng bảng, có liên hệ chặt: quan hệ. Bước 2: hai thủ thư ghi cùng lúc: nhiều người dùng. Bước 3: một máy duy nhất: tập trung. Bước 4: ghi phiếu mượn từng cái: OLTP. Nghĩa là thư viện cần một HQTCSDL quan

hệ, nhiều người dùng, tập trung, hướng OLTP, chẳng hạn PostgreSQL.

A school library has one computer at the desk, two librarians lending and returning books together, and data about books, readers and loans. Step 1: tabular data with close links, so relational. Step 2: two librarians write at once, so multiuser. Step 3: a single machine, so centralised. Step 4: loans recorded one by one, so OLTP. So the library needs a relational, multiuser, centralised DBMS aimed at OLTP, such as PostgreSQL.

LỖI HAY GẶP / COMMON MISTAKE – Cho rằng NoSQL là loại không dùng SQL và luôn tốt hơn quan hệ. Tên này thường được đọc là "Not Only SQL" (nhiều hơn SQL); mỗi loại hợp với một kiểu bài toán, và chọn theo bài toán chứ không theo tên.

Assuming NoSQL means no SQL and is always better than relational. The name is usually read as "not only SQL"; each type suits a kind of problem, so choose by problem and not by name.

LỖI HAY GẶP / COMMON MISTAKE – Xếp MySQL, Oracle hay MongoDB vào mục "cơ sở dữ liệu" khi làm bài phân loại. Đó là các HQTCSDL; phân loại áp dụng cho HQTCSDL.

Listing MySQL, Oracle or MongoDB as "databases" in a classification answer. They are DBMSs, and the classification applies to DBMSs.

LỖI HAY GẶP / COMMON MISTAKE – Hiểu "tập trung" là chỉ có một người dùng. Tập trung nói về nơi đặt dữ liệu, một hệ tập trung vẫn có thể phục vụ hàng nghìn người cùng lúc.

Reading "centralised" as having only one user. Centralised describes where the data sits; a centralised system can still serve thousands of people at once.

1.7. Người dùng CSDL – 1.7. Database users

Một CSDL có nhiều kiểu người làm việc quanh nó, mỗi kiểu một việc. Bảng sau chia họ thành bốn nhóm và gắn với công ty Trường Tiền.

A database has several kinds of people working around it, each with a job of their own. The next table sorts them into four groups and ties them to the Trường Tiền company.

Nhóm Group	Việc chính Main job	Làm việc với CSDL bằng gì Works with the database through	Ở Trường Tiền At Trường Tiền
Quản trị CSDL (DBA) <i>Database administrator (DBA)</i>	Cài đặt, cấp quyền, giám sát, sao lưu, phục hồi, chỉnh để chạy nhanh. <i>Installs, grants access, monitors, backs up, recovers and tunes for speed.</i>	Công cụ quản trị, lệnh DCL, lệnh sao lưu. <i>Admin tools, DCL commands, backup commands.</i>	Một người hoặc một nhóm giữ máy chủ PostgreSQL. <i>A person or team looking after the PostgreSQL server.</i>
Người thiết kế CSDL <i>Database designer</i>	Quyết định lưu dữ liệu gì, chia bảng ra sao, các bảng liên hệ thế nào. <i>Decides what data to keep, how to split tables and how tables relate.</i>	Mô hình và sơ đồ thiết kế (Chương 3 đến 7). <i>Models and design diagrams (Chapters 3 to 7).</i>	Người vẽ sơ đồ chín bảng của ví. <i>The person who drew the diagram of the wallet's nine tables.</i>

Nhóm <i>Group</i>	Việc chính <i>Main job</i>	Làm việc với CSDL bằng gì <i>Works with the database through</i>	Ở Trường Tiền <i>At Trường Tiền</i>
Lập trình viên ứng dụng <i>Application programmer</i>	Viết chương trình đọc và ghi CSDL. <i>Writes programs that read and write the database.</i>	SQL và ngôn ngữ lập trình. <i>SQL and a programming language.</i>	Người viết ứng dụng ví trên điện thoại. <i>The person who wrote the wallet app for phones.</i>
Người dùng cuối <i>End user</i>	Dùng dữ liệu cho công việc: tra cứu, nhập, lấy báo cáo. <i>Uses the data for work: looking up, entering, getting reports.</i>	Ứng dụng có sẵn; người thạo hơn tự viết truy vấn. <i>Ready-made applications; more skilled users write their own queries.</i>	Khách hàng dùng ứng dụng ví, nhân viên CSKH tra khách, chuyên viên phân tích viết truy vấn. <i>Customers using the wallet app, CSKH staff looking up customers, an analyst writing queries.</i>

Nhóm người dùng cuối có hai kiểu. Người dùng ứng dụng (application user) chỉ bấm nút trên chương trình có sẵn, như khách chuyển tiền trên ứng dụng ví. Người dùng thạo (sophisticated user) tự gõ truy vấn hoặc dùng công cụ phân tích, như chuyên viên cần một báo cáo chưa ứng dụng nào làm sẵn.

End users come in two kinds. An application user only presses buttons in a ready-made program, like a customer sending money in the wallet app. A sophisticated user types queries or uses analysis tools, like an analyst who needs a report that no application has yet built.

DBA có quyền lớn nhất, nên cũng chịu trách nhiệm lớn nhất: quyết định ai được xem gì, làm gì, và bảo đảm khi sự cố xảy ra, dữ liệu còn nguyên. Chương 10 dạy các việc đó bằng lệnh thật.

The DBA holds the greatest powers and therefore the greatest responsibility: deciding who may see and do what, and making sure the data survives an incident. Chapter 10 teaches these jobs with real commands.

Cách xác định một người thuộc nhóm nào – *How to tell which group a person belongs to*

1. Hỏi: người này quyết định hoặc thay đổi cấu trúc CSDL (bảng, cột)? Nếu có, hỏi tiếp là thiết kế ban đầu hay quản trị hằng ngày.
Ask: does this person decide or change the structure of the database (tables, columns)? If so, ask whether it is the original design or daily administration.
2. Hỏi: người này viết chương trình để người khác dùng CSDL? Nếu có, là lập trình viên ứng dụng.
Ask: does this person write programs for others to use the database? If so, an application programmer.
3. Còn lại là người dùng cuối. Hỏi họ bấm nút trên chương trình có sẵn (người dùng ứng dụng) hay tự viết truy vấn (người dùng thạo).

Everyone else is an end user. Ask whether they press buttons in a ready-made program (application user) or write their own queries (sophisticated user).

VÍ DỤ / EXAMPLE – Bốn người ở Trường Tiền: (a) chị Lan cấp quyền cho nhân viên mới và chạy sao lưu mỗi đêm; (b) anh Nam vẽ sơ đồ bảng cho dịch vụ vay trả góp; (c) em Hà viết chức năng nạp tiền cho ứng dụng; (d) anh Sơn mở ứng dụng tra cứu khách để trả lời điện thoại. Theo ba bước: chị Lan quản trị hằng ngày, thuộc nhóm DBA. Anh Nam quyết định cấu trúc ban đầu, là người thiết kế. Em Hà viết chương trình cho người khác dùng CSDL, là lập trình viên ứng dụng. Anh Sơn bấm nút trên chương trình có sẵn, là người dùng ứng dụng.
Four people at Trường Tiền: (a) Lan grants access to a new employee and runs the backup each night; (b) Nam draws the table diagram for the instalment loan service; (c) Hà writes the top-up function for the app; (d) Sơn opens the customer lookup application to answer a phone call. By the three steps: Lan administers daily, so she is the DBA. Nam fixes the original structure, so he is the designer. Hà writes a program for others to use the database, so she is an application programmer. Sơn presses buttons in a ready-made program, so he is an application user.

LỖI HAY GẶP / COMMON MISTAKE – Cho rằng DBA là người nhập dữ liệu. DBA quản trị hệ thống; việc nhập dữ liệu thuộc người dùng cuối hoặc chương trình.
Believing the DBA is the person who enters data. The DBA administers the system; data entry belongs to end users or programs.

LỖI HAY GẶP / COMMON MISTAKE – Cấp cho lập trình viên hay người dùng quyền của DBA cho tiện. Mỗi người chỉ nên có quyền vừa đủ cho việc của mình (nguyên tắc quyền tối thiểu).
Giving programmers or users DBA rights for convenience. Each person should have just enough rights for their own work (the principle of least privilege).

1.8. Ứng dụng của CSDL, nhất là trong tài chính, ngân hàng và ví điện tử – 1.8. Uses of databases, especially in finance, banking and e-wallets

Gần như mọi hệ thống bạn dùng hằng ngày đều có CSDL phía sau. Bảng sau điếm qua vài lĩnh vực.

Almost every system you use daily has a database behind it. The next table runs through a few fields.

Lĩnh vực <i>Field</i>	CSDL giữ gì <i>What the database holds</i>	Việc nó làm được <i>What it makes possible</i>
Ngân hàng, tài chính <i>Banking and finance</i>	Khách hàng, tài khoản, giao dịch, khoản vay. <i>Customers, accounts, transactions, loans.</i>	Tra số dư, chuyển khoản, tính lãi, so khớp sổ sách. <i>Checking balances, transfers, interest, matching the books.</i>
Ví điện tử <i>E-wallets</i>	Khách hàng, ví, liên kết ngân hàng, giao dịch. <i>Customers, wallets, bank links, transactions.</i>	Nạp, rút, chuyển, thanh toán tại cửa hàng. <i>Top-ups, withdrawals, transfers, payments at shops.</i>

Lĩnh vực <i>Field</i>	CSDL giữ gì <i>What the database holds</i>	Việc nó làm được <i>What it makes possible</i>
Thương mại điện tử, bán lẻ <i>E-commerce and retail</i>	Sản phẩm, tồn kho, đơn hàng, khách mua. <i>Products, stock, orders, buyers.</i>	Đặt hàng, theo dõi giao hàng, gợi ý sản phẩm. <i>Ordering, tracking deliveries, suggesting products.</i>
Giáo dục <i>Education</i>	Sinh viên, môn học, điểm, lịch thi. <i>Students, courses, grades, exam timetables.</i>	Đăng ký học phần, tra điểm. <i>Course enrolment, checking grades.</i>
Y tế <i>Healthcare</i>	Bệnh nhân, lần khám, đơn thuốc. <i>Patients, visits, prescriptions.</i>	Tra hồ sơ khám, đặt lịch. <i>Looking up records, booking appointments.</i>
Giao thông, du lịch <i>Transport and travel</i>	Chuyến, ghế, vé, hành khách. <i>Trips, seats, tickets, passengers.</i>	Đặt chỗ mà không bán trùng ghế. <i>Booking without selling the same seat twice.</i>
Viễn thông <i>Telecommunications</i>	Thuê bao, cuộc gọi, gói cước. <i>Subscribers, calls, tariff plans.</i>	Tính cước, báo hóa đơn. <i>Charging and billing.</i>
Cơ quan nhà nước <i>Government</i>	Hồ sơ dân cư, đất đai, doanh nghiệp, thuế. <i>Records on residents, land, businesses, tax.</i>	Cấp giấy tờ, tra cứu, quản lý. <i>Issuing papers, lookup, administration.</i>

Trong tài chính, CSDL gánh việc nặng hơn nơi khác vì đối tượng nó giữ là tiền. Mỗi đồng phải truy vết được từ lúc vào đến lúc ra. Hai yêu cầu đến cùng giây không được làm sai số dư. Một sự cố không được làm mất giao dịch đã báo thành công. Dữ liệu của khách phải được giấu khỏi người không có quyền. Đó là các đặc trưng ở mục 1.4: kiểm soát đồng thời, ràng buộc, phân quyền, sao lưu.

In finance the database carries a heavier load than elsewhere, because what it holds is money. Every dong must be traceable from entry to exit. Two requests in the same second must not corrupt a balance. A failure must not lose a transaction that has been reported successful. Customer data must be hidden from anyone without the right. These are the features of section 1.4: concurrency control, constraints, permissions and backup.

Pháp luật Việt Nam định nghĩa dịch vụ ví điện tử ở Nghị định 52/2024/NĐ-CP về thanh toán không dùng tiền mặt, Điều 3 khoản 16: đó là dịch vụ do ngân hàng, chi nhánh ngân hàng nước ngoài hoặc tổ chức cung ứng dịch vụ trung gian thanh toán cung cấp cho khách hàng để nạp tiền vào ví điện tử, rút tiền ra khỏi ví điện tử và thực hiện giao dịch thanh toán. Cùng Điều 3, khoản 10 xếp ví điện tử vào các phương tiện thanh toán không dùng tiền mặt, bên cạnh séc, thẻ ngân hàng và các phương tiện khác.

Vietnamese law defines the e-wallet service in Decree 52/2024/ND-CP on non-cash payments, Article 3, clause 16: it is a service that a bank, a foreign bank branch or a payment intermediary provides to customers to top up an e-wallet, withdraw money from an e-wallet and carry out payment transactions. In the same Article 3, clause 10 lists the e-wallet among the non-cash payment instruments, alongside cheques, bank cards and others.

Ba việc trong định nghĩa ứng với ba trong bốn giá trị của cột *LoaiGD* trong bảng giao dịch của Trường Tiền: Nạp, Rút, Thanh toán. Bảng này còn ghi thêm loại Chuyển, giữa hai ví. CSDL ta học mô phỏng thứ mà một dịch vụ có thật phải lưu và bảo vệ.

*The three jobs in the definition match three of the four values of the *LoaiGD* column in the Trường Tiền transaction table: Nạp (top-up), Rút (withdrawal) and Thanh toán (payment). The table also records a fourth type, Chuyển (transfer), between two wallets. The database we study models what a real service has to store and protect.*

LỖI HAY GẶP / COMMON MISTAKE – Cho rằng bảng nào có tiền thì chỉ cần cột số dư là đủ. Số dư là kết quả; cái cần lưu để truy vết là từng giao dịch, từ đó mới kiểm lại được số dư.

Believing that any money table needs only a balance column. The balance is a result; what must be stored for tracing is each transaction, from which the balance can be checked.

1.9. Dữ liệu cá nhân và trách nhiệm của người giữ CSDL – 1.9. Personal data and the duties of whoever keeps a database

Bảng khách hàng của Trường Tiền có họ tên, ngày sinh, số điện thoại, tỉnh thành. Bảng giao dịch cho biết ví nào chuyển tiền cho ví nào, lúc nào. Những dữ liệu này xác định hoặc giúp xác định một con người cụ thể, nên là dữ liệu cá nhân theo Luật Bảo vệ dữ liệu cá nhân số 91/2025/QH15, Điều 2 khoản 1. Luật có hiệu lực từ ngày 01/01/2026 (Điều 38 khoản 1). Cũng theo Điều 2 khoản 1, dữ liệu sau khi khử nhận dạng, tức đã bỏ hoặc đổi sao cho không còn chỉ ra được một người, thì không còn là dữ liệu cá nhân.

The Trường Tiền customer table holds names, dates of birth, phone numbers and provinces. The transaction table shows which wallet sent money to which wallet, and when. This data identifies, or helps identify, a particular person, so it is personal data under the Law on Personal Data Protection No. 91/2025/QH15, Article 2, clause 1. The law has taken effect since 1 January 2026 (Article 38, clause 1). Under the same clause, data that has been de-identified, meaning altered or stripped so that it no longer points to a person, stops being personal data.

Luật gọi người mà dữ liệu nói về là chủ thể dữ liệu cá nhân (Điều 2 khoản 5): ở đây là khách hàng. Bên quyết định mục đích và phương tiện xử lý là bên kiểm soát dữ liệu cá nhân (khoản 7): ở đây là công ty Trường Tiền. Bên làm việc xử lý theo yêu cầu của bên kiểm soát thông qua hợp đồng là bên xử lý dữ liệu cá nhân (khoản 8); nếu công ty thuê một đơn vị khác vận hành máy chủ, đơn vị đó có thể là bên xử lý. Xử lý (khoản 6) gồm thu thập, phân tích, tổng hợp, mã hóa (biến dữ liệu thành dạng không đọc được nếu thiếu khóa giải), chỉnh sửa, xóa, hủy, cung cấp và nhiều hoạt động khác; mỗi lệnh thêm, sửa, xóa hay tổng hợp dữ liệu khách hàng đều là một dạng xử lý.

The law calls the person the data is about the data subject (Article 2, clause 5): here, the customer. The party that decides the purpose and means of processing is the personal data controller (clause 7): here, the Trường Tiền company. A party that does the processing on the controller's request under a contract is the personal data processor (clause 8); if the company hires another firm to run its servers, that firm may be a processor. Processing (clause 6) covers collecting, analysing, compiling, encrypting (turning data into a form that cannot be read

without the key), editing, deleting, destroying, providing and other activities; every command that adds, changes, deletes or summarises customer data is a form of processing.

Bảng sau nối các quy định với việc cụ thể người giữ CSDL phải làm.

The next table links the rules to what whoever keeps the database has to do.

Việc phải làm <i>What must be done</i>	Căn cứ <i>Basis</i>	Làm ở CSDL Trường Tiền thế nào <i>How it is done in the Trường Tiền database</i>
Chỉ thu thập, xử lý đúng phạm vi và mục đích cụ thể, rõ ràng. <i>Collect and process only within a specific, clear scope and purpose.</i>	Luật 91/2025/QH15, Điều 3 khoản 2 <i>Law 91/2025/QH15, Article 3, clause 2</i>	Bảng khách hàng chỉ giữ cột cần để vận hành ví; không thêm cột chưa có mục đích. <i>The customer table keeps only the columns needed to run the wallet; no column is added without a purpose.</i>
Giữ dữ liệu chính xác, sửa khi cần, lưu trong thời gian phù hợp với mục đích. <i>Keep data accurate, correct it when needed and store it for a period fitting the purpose.</i>	Điều 3 khoản 3 <i>Article 3, clause 3</i>	Khách đổi số điện thoại thì sửa ở một chỗ duy nhất (đặc trưng 5); đặt thời hạn lưu cho dữ liệu cũ. <i>When a customer changes phone number, it is corrected in one place only (feature 5); a retention period is set for old data.</i>
Áp dụng biện pháp quản lý và kỹ thuật phù hợp, rà soát và cập nhật khi cần. <i>Apply suitable managerial and technical measures, reviewing and updating them as needed.</i>	Điều 37 khoản 1 điểm c <i>Article 37, clause 1, point c</i>	Phân quyền theo vai trò, khung nhìn che số điện thoại, sao lưu định kỳ. <i>Role-based permissions, a view that hides phone numbers, regular backups.</i>
Thực hiện quy định về dữ liệu cá nhân nhạy cảm (nhóm dữ liệu cá nhân mà luật quy định riêng) và tiêu chuẩn an toàn, bảo mật trong tài chính, ngân hàng. <i>Follow the rules on sensitive personal data (a group of personal data that the law regulates separately) and the safety and security standards in finance and banking.</i>	Điều 27 khoản 1 điểm a <i>Article 27, clause 1, point a</i>	Rà soát định kỳ bảng nào chứa dữ liệu nhạy cảm và ai được truy cập. <i>Review regularly which tables hold sensitive data and who can access them.</i>
Thông báo cho chủ thể khi lộ hoặc mất thông tin về tài khoản ngân hàng, tài chính, tín dụng. <i>Notify the data subject when information about bank, financial or credit accounts is leaked or lost.</i>	Điều 27 khoản 1 điểm d <i>Article 27, clause 1, point d</i>	Có quy trình xác định khách nào bị ảnh hưởng bằng truy vấn rồi báo cho họ. <i>A procedure that identifies the affected customers by query and informs them.</i>

Việc phải làm <i>What must be done</i>	Căn cứ <i>Basis</i>	Làm ở CSDL Trường Tiền thế nào <i>How it is done in the Trường Tiền database</i>
Không mua, bán dữ liệu cá nhân (trừ khi luật có quy định khác); không chiếm đoạt, cố ý làm lộ hay làm mất. <i>Not buy or sell personal data (unless a law provides otherwise); not seize, deliberately leak or lose it.</i>	Điều 7 khoản 6 và 7 <i>Article 7, clauses 6 and 7</i>	Cấm xuất danh sách khách ra ngoài công ty để bán hoặc trao đổi. <i>No exporting customer lists outside the company to sell or trade.</i>

Chủ thể dữ liệu cũng có quyền (Điều 4 khoản 1): được biết về hoạt động xử lý, đồng ý hoặc không đồng ý, xem và chỉnh sửa dữ liệu của mình, yêu cầu cung cấp, xóa hoặc hạn chế xử lý. Muốn đáp ứng yêu cầu của một khách, người giữ CSDL phải tìm được đúng các dòng của khách đó trong mọi bảng. Đây là lý do mã khách hàng (MaKH) có mặt ở nhiều bảng và là khóa để lần theo.

The data subject has rights too (Article 4, clause 1): to be told about the processing, to agree or not, to view and correct their data, and to ask for it to be provided, deleted or restricted. To meet a customer's request, whoever keeps the database must find exactly that customer's rows in every table. That is why the customer code (MaKH) appears in several tables and serves as the key for tracing.

Cách rà một cột có nên giữ và cho ai xem – *How to review whether to keep a column and who may see it*

- Hỏi cột này có xác định hoặc giúp xác định một người không. Nếu có, nó là dữ liệu cá nhân.
Ask whether the column identifies, or helps identify, a person. If so, it is personal data.
- Hỏi cột này phục vụ mục đích cụ thể nào của dịch vụ. Không nêu được mục đích thì không nên thu thập.
Ask which specific purpose of the service the column serves. If you cannot name one, do not collect it.
- Liệt kê các nhóm người dùng và hỏi từng nhóm có cần cột này cho việc của họ không. Nhóm không cần thì không cấp quyền xem.
List the groups of users and ask whether each needs the column for its work. A group that does not need it gets no right to see it.

VÍ DỤ / EXAMPLE – Xét cột SoDienThoai của bảng KhachHang. Bước 1: số điện thoại giúp xác định một người, nên là dữ liệu cá nhân. Bước 2: nó phục vụ xác minh và liên lạc về giao dịch, nên được giữ. Bước 3: nhân viên CSKH cần để liên lạc; nhân viên lễ tân chỉ cần biết khách là ai và hạng gì, nên không cần. Cách làm: cho lễ tân dùng khung nhìn KhachLeTan ở mục 1.4, vốn không có cột số điện thoại. Nghĩa là quyền xem được cắt theo nhu cầu của từng nhóm, chứ không để một bảng đầy đủ cho mọi người.

Take the SoDienThoai column of the KhachHang table. Step 1: a phone number helps identify a person, so it is personal data. Step 2: it serves verification and contact about transactions, so it is kept. Step 3: CSKH staff need it to make contact; front-desk staff only need to know who the customer

is and the tier, so they do not. The method: give the front desk the KhachLeTan view from section 1.4, which has no phone column. So viewing rights are cut to each group's need rather than leaving a full table open to everyone.

LỖI HAY GẶP / COMMON MISTAKE – Cho rằng dữ liệu trong bài tập thì không cần giữ kỷ luật. Thói quen giữ dữ liệu khách hàng theo nguyên tắc phải được tập từ bài tập; với dữ liệu thật, gửi nguyên bảng khách hàng qua tin nhắn hay thư để nhờ giúp có thể là xử lý dữ liệu cá nhân trái quy định.

Assuming that exercise data needs no discipline. The habit of handling customer data by the rules should be built in exercises; with real data, sending a whole customer table by message or email to ask for help may be processing personal data against the rules.

LỖI HAY GẶP / COMMON MISTAKE – Nghĩ rằng chỉ cần xóa họ tên là hết dữ liệu cá nhân. Nếu số điện thoại hay mã khách vẫn chỉ ra được một người thì dữ liệu vẫn là dữ liệu cá nhân; chỉ khi đã khử nhận dạng đúng nghĩa, không còn xác định được ai thì mới thôi là dữ liệu cá nhân.

Thinking that deleting the name is enough to remove personal data. If a phone number or customer code still points to a person, the data is still personal data; only data properly de-identified, no longer pointing to anyone, stops being personal data.

1.10. Bài mẫu dạng đề thi – 1.10. Worked examples in exam style

Bài mẫu 1. Xếp dữ liệu, thông tin, tri thức – Worked example 1. Sorting data, information and knowledge

Đề: với CSDL Trường Tiền, xếp bốn mẫu sau vào dữ liệu, thông tin hay tri thức và giải thích. (a) "0978550104"; (b) "Tháng 9 có 3 giao dịch thất bại"; (c) "Trong 15 ví có 1 ví đang tạm khóa"; (d) "Muốn đỡ mất phí, đừng rút nhiều lần nhỏ".

Exam question: for the Trường Tiền database, place each of the four items in data, information or knowledge and explain. (a) "0978550104"; (b) "September has 3 failed transactions"; (c) "Of 15 wallets, 1 is temporarily locked"; (d) "To save on fees, do not make many small withdrawals".

1. Mẫu (a) là một chuỗi số đứng một mình, ghi trong cột số điện thoại. Chưa ai đếm hay so sánh gì: dữ liệu.
Item (a) is a string of digits standing alone, recorded in the phone number column. Nobody has counted or compared anything: data.
2. Mẫu (b) là kết quả đếm những giao dịch có trạng thái thất bại trong tháng 9 và trả lời câu hỏi "có bao nhiêu giao dịch thất bại": thông tin.
Item (b) results from counting the transactions with a failed status in September and answers "how many transactions failed": information.
3. Mẫu (c) là kết quả đếm ví theo trạng thái trên toàn bộ 15 ví: thông tin.
Item (c) results from counting wallets by status across all 15 wallets: information.
4. Mẫu (d) là lời khuyên rút ra từ việc phí tính theo lần rút, dùng để chọn cách làm: tri thức.
Item (d) is advice drawn from the fact that the fee is charged per withdrawal, and is used to choose what to do: knowledge.

Đáp án: (a) dữ liệu, (b) thông tin, (c) thông tin, (d) tri thức.

Answer: (a) data, (b) information, (c) information, (d) knowledge.

Bài mẫu 2. Tìm nhược điểm của hệ thống tệp – *Worked example 2. Finding the weaknesses of a file-based system*

Đề: một trung tâm ngoại ngữ có tệp "Học viên" do lễ tân giữ (họ tên, số điện thoại, lớp) và tệp "Học phí" do kế toán giữ (họ tên, số điện thoại, số tiền đã đóng). Giáo viên được cho mở tệp Học phí để xem danh sách lớp. Tháng này học viên Mai đổi số điện thoại, lễ tân đã sửa. Hãy chỉ ra các nhược điểm và đề xuất cách khắc phục bằng CSDL.

Exam question: a language centre has a "Students" file kept by reception (name, phone number, class) and a "Fees" file kept by accounting (name, phone number, amount paid). Teachers are allowed to open the Fees file to see class lists. This month the student Mai changed her phone number and reception corrected it. Point out the weaknesses and propose how a database would fix them.

1. Dữ liệu nằm ở hai tệp, do hai bộ phận giữ.

The data sits in two files, kept by two departments.

2. Họ tên và số điện thoại xuất hiện ở cả hai tệp: dư thừa. Lễ tân chỉ sửa tệp Học viên nên tệp Học phí còn số cũ: không nhất quán.

Names and phone numbers appear in both files: redundancy. Reception corrected only the Students file, so the Fees file keeps the old number: inconsistency.

3. Muốn biết lớp nào còn nhiều học viên chưa đóng đủ phí phải ghép hai tệp bằng tay: khó truy xuất.

To find which class has many students who have not paid in full, someone must match the two files by hand: poor retrieval.

4. Giáo viên mở tệp Học phí thì thấy cả số tiền mỗi học viên đã đóng: bảo mật yếu, vì tệp chỉ có mở hoặc không mở.

Teachers who open the Fees file see how much each student has paid: weak security, because a file is either open or closed.

5. Cách khắc phục: dùng một CSDL với bảng học viên chứa số điện thoại một lần, bảng học phí tham chiếu đến học viên bằng mã. Tạo khung nhìn danh sách lớp không có cột tiền cho giáo viên.

Fix: use one database with a student table holding the phone number once, and a fee table that refers to students by code. Create a class-list view without the money column for teachers.

Kết luận: trung tâm gặp dư thừa, không nhất quán, khó truy xuất và bảo mật yếu; CSDL xử lý bằng đặc trưng kiểm soát dư thừa và khung nhìn kèm phân quyền.

Conclusion: the centre suffers from redundancy, inconsistency, poor retrieval and weak security; a database handles them through controlled redundancy and views with permissions.

Bài mẫu 3. Chọn loại HQTCSĐL và nhóm người dùng – *Worked example 3. Choosing the DBMS type and the user groups*

Đề: một chuỗi ba cửa hàng điện thoại có máy tính tiền ở mỗi cửa hàng, nhiều nhân viên bán cùng lúc, số liệu về một máy chủ đặt tại trụ sở. Cuối tháng giám đốc xem báo cáo

doanh thu tổng hợp của ba năm. Phân loại hệ phục vụ bán hàng và hệ phục vụ báo cáo theo bốn tiêu chí, rồi nêu nhóm người dùng trong ba người: kế toán xem báo cáo, thu ngân quét hàng, người cài và sao lưu máy chủ.

Exam question: a chain of three phone shops has a till in each shop, several staff selling at once, and data going to one server at head office. At month end the manager reads a revenue report covering three years. Classify the selling system and the reporting system by the four criteria, then name the user group of three people: the accountant reading the report, the cashier scanning goods, and the person who installs and backs up the server.

1. Mô hình dữ liệu: hàng hóa, đơn bán, khách đều dạng bảng và liên quan chặt chẽ nên chọn quan hệ, cho cả hai hệ.

Data model: goods, sales and customers are tabular and closely related, so choose relational, for both systems.

2. Số người dùng: nhiều nhân viên bán cùng lúc, nên nhiều người dùng, cho cả hai hệ.

Users: several staff sell at once, so multiuser, for both systems.

3. Nơi đặt dữ liệu: tất cả về một máy chủ tại trụ sở, nên tập trung.

Where data sits: everything goes to one server at head office, so centralised.

4. Mục đích: hệ bán hàng ghi từng giao dịch nhỏ nên OLTP. Hệ báo cáo đọc khối dữ liệu ba năm để tổng hợp nên OLAP.

Purpose: the selling system records small transactions one by one, so OLTP. The reporting system reads three years of data to summarise, so OLAP.

5. Nhóm người dùng: kế toán xem báo cáo bằng chương trình có sẵn là người dùng ứng dụng; thu ngân quét hàng cũng là người dùng ứng dụng; người cài và sao lưu máy chủ là DBA.

User groups: the accountant reading the report in a ready-made program is an application user; the cashier scanning goods is also an application user; the person installing and backing up the server is the DBA.

Đáp án: hệ bán hàng là quan hệ, nhiều người dùng, tập trung, OLTP; hệ báo cáo là quan hệ, nhiều người dùng, tập trung, OLAP.

Answer: the selling system is relational, multiuser, centralised, OLTP; the reporting system is relational, multiuser, centralised, OLAP.

1.11. Bảng thuật ngữ – 1.11. Glossary

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Dữ liệu	data	Sự việc hoặc giá trị được ghi lại, chưa giải thích. <i>A fact or value that has been recorded, not yet explained.</i>
Thông tin	information	Dữ liệu đã xử lý, có ngữ cảnh, trả lời một câu hỏi. <i>Data that has been processed, has context and answers a question.</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Tri thức	knowledge	Hiểu biết rút ra từ thông tin, dùng để quyết định. <i>Understanding drawn from information and used to decide.</i>
Cơ sở dữ liệu (CSDL)	database	Tập dữ liệu có tổ chức, liên quan nhau, dùng chung. <i>An organised set of related data shared by many users and programs.</i>
Hệ quản trị cơ sở dữ liệu (HQTCSDL)	database management system (DBMS)	Phần mềm quản lý, bảo vệ và truy vấn CSDL. <i>Software that manages, protects and queries a database.</i>
Hệ cơ sở dữ liệu	database system	CSDL cùng HQTCSDL, ứng dụng, phần cứng, quy trình và con người. <i>The database with the DBMS, applications, hardware, procedures and people.</i>
Hệ thống tệp	file-based system	Cách lưu dữ liệu trong các tệp rời, chương trình nào lo tệp nấy. <i>Keeping data in separate files, each program looking after its own.</i>
Dư thừa dữ liệu	data redundancy	Một sự việc được ghi ở nhiều nơi cùng lúc. <i>One fact recorded in several places at once.</i>
Siêu dữ liệu	metadata	Dữ liệu mô tả cấu trúc của dữ liệu khác. <i>Data that describes the structure of other data.</i>
Từ điển dữ liệu	data dictionary	Nơi HQTCSDL cất siêu dữ liệu. <i>The place where the DBMS keeps metadata.</i>
Độc lập chương trình và dữ liệu	program-data independence	Đổi cấu trúc dữ liệu mà không phải sửa chương trình. <i>Changing the data structure without changing the programs.</i>
Khung nhìn	view	Cửa sổ định sẵn cho thấy một phần dữ liệu, lưu câu hỏi chứ không chép dữ liệu. <i>A ready-made window onto part of the data; it stores a question, not a copy.</i>
Ràng buộc toàn vẹn	integrity constraint	Quy tắc CSDL dùng để từ chối dữ liệu sai. <i>A rule the database uses to reject wrong data.</i>
SQL	Structured Query Language	Ngôn ngữ ra lệnh cho HQTCSDL quan hệ. <i>The language for giving orders to a relational DBMS.</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
DDL	Data Definition Language	Nhóm lệnh SQL dựng và sửa cấu trúc. <i>The group of SQL commands that builds and changes structure.</i>
DML	Data Manipulation Language	Nhóm lệnh SQL thêm, sửa, xóa, đọc dữ liệu. <i>The group of SQL commands that adds, changes, deletes and reads data.</i>
DCL	Data Control Language	Nhóm lệnh SQL cấp và thu hồi quyền. <i>The group of SQL commands that grants and withdraws permissions.</i>
Mô hình dữ liệu	data model	Cách quy ước tổ chức và nối dữ liệu, ví dụ mô hình quan hệ. <i>An agreed way to organise and link data, such as the relational model.</i>
OLTP	online transaction processing	Hệ xử lý nhiều giao dịch nhỏ hằng ngày. <i>A system that handles many small daily transactions.</i>
OLAP	online analytical processing	Hệ đọc khối dữ liệu lớn để phân tích, báo cáo. <i>A system that reads large blocks of data for analysis and reports.</i>
Quản trị CSDL (DBA)	database administrator (DBA)	Người cài đặt, cấp quyền, giám sát, sao lưu và phục hồi. <i>The person who installs, grants access, monitors, backs up and recovers.</i>
CSDL phân tán	distributed database	Dữ liệu rải ra nhiều nơi nối bằng mạng. <i>Data spread over several places joined by a network.</i>
Dữ liệu cá nhân	personal data	Dữ liệu xác định hoặc giúp xác định một con người cụ thể. <i>Data that identifies or helps identify a particular person.</i>
Bên kiểm soát dữ liệu cá nhân	personal data controller	Bên quyết định mục đích và phương tiện xử lý dữ liệu cá nhân. <i>The party that decides the purpose and means of processing personal data.</i>

1.12. Bài tập vận dụng có lời giải – *Applied exercises with solutions*

Bài 1. Phòng khám nha khoa ba tệp – *Exercise 1. A dental clinic with three files*

Phòng khám nha khoa Ánh Dương dùng ba tệp Excel: lễ tân giữ tệp lịch hẹn (họ tên, số điện thoại, giờ hẹn), bác sĩ giữ tệp hồ sơ điều trị (họ tên, số điện thoại, ghi chú điều trị),

kế toán giữ tệp thu tiền (họ tên, số điện thoại, số tiền). Lễ tân được mở cả ba tệp để tiện hỏi han. Khi bệnh nhân Hạnh đổi số điện thoại, lễ tân chỉ sửa tệp lịch hẹn. Hãy chỉ ra các nhược điểm của cách làm này và đề xuất cách dùng CSDL để khắc phục.

The Ánh Dương dental clinic uses three Excel files: reception keeps the appointment file (name, phone number, appointment time), the dentist keeps the treatment file (name, phone number, treatment notes) and accounting keeps the payment file (name, phone number, amount). Reception may open all three files for convenience. When the patient Hạnh changes her phone number, reception corrects only the appointment file. Point out the weaknesses of this arrangement and propose how a database would fix them.

1. Dữ liệu nằm ở ba tệp do ba người giữ. Họ tên và số điện thoại xuất hiện ở cả ba: dư thừa dữ liệu.
The data sits in three files kept by three people. Names and phone numbers appear in all three: data redundancy.
2. Lễ tân chỉ sửa tệp lịch hẹn nên hai tệp kia còn số cũ: không nhất quán. Bác sĩ hoặc kế toán gọi số cũ sẽ không liên lạc được.
Reception corrected only the appointment file, so the other two keep the old number: inconsistency. The dentist or accountant calling the old number will not get through.
3. Lễ tân mở được cả tệp hồ sơ điều trị và tệp thu tiền, tức xem được thông tin điều trị và số tiền không thuộc việc của mình: bảo mật yếu.
Reception can open the treatment file and the payment file, so it sees treatment details and amounts that are not its business: weak security.
4. Muốn biết bệnh nhân nào có hẹn tuần này mà chưa đóng tiền phải ghép tay hai tệp: khó truy xuất.
To find which patients have an appointment this week but have not paid, someone must match two files by hand: poor retrieval.
5. Cách khắc phục: một CSDL, bảng bệnh nhân giữ họ tên, số điện thoại đúng một lần; bảng lịch hẹn, bảng hồ sơ và bảng thu tiền tham chiếu bệnh nhân bằng mã. Sửa số điện thoại một chỗ là cả hệ thống cùng thấy.
Fix: one database, with a patient table holding name and phone number exactly once, and appointment, treatment and payment tables referring to the patient by code. Correct the phone number in one place and the whole system sees it.
6. Cho lễ tân dùng khung nhìn chỉ có họ tên, số điện thoại và giờ hẹn, không có ghi chú điều trị và số tiền. Phân quyền để kế toán chỉ đọc bảng thu tiền.
Give reception a view with only name, phone number and appointment time, without treatment notes or amounts. Set permissions so accounting reads only the payment table.

Bài 2. Đặt phòng ký túc xá – Exercise 2. Booking dormitory rooms

Trường có ứng dụng đặt phòng ký túc xá. Dữ liệu gồm sinh viên, phòng, lượt đăng ký. Đầu kỳ, hàng nghìn sinh viên đặt phòng cùng lúc; mọi dữ liệu về một máy chủ đặt tại trường. Cuối năm, ban quản lý đọc báo cáo tỉ lệ lấp đầy phòng của bốn kỳ gần nhất. Hãy phân loại hệ phục vụ đặt phòng và hệ phục vụ báo cáo theo bốn tiêu chí, rồi xếp bốn người sau vào nhóm người dùng: sinh viên đặt phòng trên ứng dụng; nhân viên viết chức

năng đăng ký phòng; người cấp quyền và sao lưu máy chủ; cán bộ ban quản lý tự viết truy vấn để lấy báo cáo.

The university has an application for booking dormitory rooms. The data covers students, rooms and bookings. At the start of term thousands of students book at once; all data goes to one server at the university. At year end the management board reads a report of the occupancy rate over the last four terms. Classify the booking system and the reporting system by the four criteria, then place these four people in user groups: a student booking in the app; a staff member who writes the room booking function; the person who grants access and backs up the server; a board officer who writes queries to get the report.

1. Mô hình dữ liệu: sinh viên, phòng, lượt đăng ký đều dạng bảng và nối với nhau nên chọn quan hệ, cho cả hai hệ.
Data model: students, rooms and bookings are tabular and linked, so choose relational, for both systems.
2. Số người dùng: hàng nghìn sinh viên đặt cùng lúc nên nhiều người dùng, cần kiểm soát đồng thời để không xếp hai sinh viên vào một chỗ.
Users: thousands of students book at once, so multiuser, with concurrency control so that two students are not given the same place.
3. Nơi đặt dữ liệu: tất cả về một máy chủ tại trường nên tập trung.
Where the data sits: everything goes to one server at the university, so centralised.
4. Mục đích: hệ đặt phòng ghi từng lượt đăng ký nhỏ nên OLTP. Hệ báo cáo đọc dữ liệu bốn kỳ để tổng hợp tỉ lệ lấp đầy nên OLAP.
Purpose: the booking system records small bookings one by one, so OLTP. The reporting system reads four terms of data to summarise occupancy, so OLAP.
5. Nhóm người dùng: sinh viên đặt phòng bằng ứng dụng có sẵn là người dùng ứng dụng; nhân viên viết chức năng là lập trình viên ứng dụng; người cấp quyền và sao lưu là DBA; cán bộ tự viết truy vấn là người dùng thạo.
User groups: the student booking with a ready-made app is an application user; the staff member who writes the function is an application programmer; the person granting access and backing up is the DBA; the officer who writes queries is a sophisticated user.

Bài 3. Đọc truy vấn trên CSDL Trường Tiền – *Exercise 3. Reading queries on the Trường Tiền database*

Cho CSDL Trường Tiền. Hãy viết lệnh và nêu kết quả cho ba yêu cầu: (a) tìm ví đang ở trạng thái Tạm khóa; (b) đếm số ví thuộc loại Tiết kiệm; (c) tính tổng số tiền của các giao dịch Nạp thành công. Gợi ý cột: ví có MaVi, LoaiVi, TrangThai; giao dịch có LoaiGD, SoTien, TrangThai.

Given the Trường Tiền database, write the command and state the result for three requests: (a) find the wallet whose status is Tạm khóa (temporarily locked); (b) count the wallets of type Tiết kiệm (savings); (c) total the amounts of successful top-up (Nạp) transactions. Hint on columns: wallets have MaVi, LoaiVi, TrangThai; transactions have LoaiGD, SoTien, TrangThai.

1. Yêu cầu (a) lấy một cột từ bảng ví với điều kiện về trạng thái: "SELECT MaVi FROM Vi WHERE TrangThai = 'Tạm khóa';" Kết quả: 1 dòng, ví V0010.

Request (a) takes one column from the wallet table with a condition on status: "SELECT MaVi FROM Vi WHERE TrangThai = 'Tạm khóa';" Result: 1 row, wallet V0010.

2. Yêu cầu (b) cần đếm dòng thỏa điều kiện loại ví: "SELECT COUNT(1) FROM Vi WHERE LoaiVi = 'Tiết kiệm';" Kết quả: 4.

Request (b) needs a count of rows meeting a condition on wallet type: "SELECT COUNT(1) FROM Vi WHERE LoaiVi = 'Tiết kiệm';" Result: 4.

3. Yêu cầu (c) cần cộng cột số tiền với hai điều kiện, loại Nạp và trạng thái Thành công: "SELECT SUM(SoTien) FROM GiaoDich WHERE LoaiGD = 'Nạp' AND TrangThai = 'Thành công';" Kết quả: 17.500.000.

Request (c) needs the amount column added up under two conditions, type Nạp and status Thành công: "SELECT SUM(SoTien) FROM GiaoDich WHERE LoaiGD = 'Nạp' AND TrangThai = 'Thành công';" Result: 17,500,000.

4. Đọc kết quả: trong 15 ví có đúng 1 ví tạm khóa và 4 ví tiết kiệm; các lượt nạp thành công của tháng 9 cộng lại được 17.500.000 đ.

Reading the results: of 15 wallets exactly 1 is temporarily locked and 4 are savings wallets; the successful top-ups of September add up to VND 17,500,000.

1.13. Câu hỏi tự luận và dàn ý trả lời – Essay questions and answer outlines

Câu 1. Phân biệt dữ liệu, thông tin và tri thức. Lấy ví dụ từ sao kê ví điện tử cho từng cấp.

Question 1. Distinguish data, information and knowledge. Give an example of each level from an e-wallet statement.

- Dữ liệu: sự việc hoặc giá trị được ghi lại, chưa giải thích; ví dụ một dòng sao kê (ví, loại giao dịch, số tiền, thời điểm).
Data: a fact or value that has been recorded, not yet explained; for example one statement line (wallet, transaction type, amount, time).
- Thông tin: dữ liệu đã xử lý, có ngữ cảnh, trả lời một câu hỏi; ví dụ số lượt và tổng tiền nạp thành công trong tháng.
Information: data that has been processed, has context and answers a question; for example the count and total of successful top-ups in the month.
- Tri thức: hiểu biết rút ra từ nhiều thông tin, dùng để quyết định; ví dụ phí rút tính theo lần nên gom rút một lần thì rẻ hơn.
Knowledge: understanding drawn from several pieces of information, used to decide; for example the fee is per withdrawal, so one larger withdrawal is cheaper.
- Mối liên hệ: dữ liệu được xử lý thành thông tin, nhiều thông tin được hiểu thành tri thức.
The link: data is processed into information, and several pieces of information are understood as knowledge.
- Câu hỏi giúp phân biệt: mẫu này đã trả lời câu hỏi nào chưa, và có dùng để chọn hành động không.
Questions that tell them apart: has this item answered a question yet, and is it used to choose an action.

Câu 2. Trình bày các nhược điểm của hệ thống tệp và cho biết CSDL khắc phục từng nhược điểm bằng đặc trưng nào.

Question 2. *Present the weaknesses of a file-based system and say which feature of a database remedies each.*

- **Dư thừa và không nhất quán:** cùng một dữ kiện ở nhiều tệp, sửa một nơi bỏ sót nơi khác; CSDL ghi mỗi sự việc một chỗ (kiểm soát dư thừa).
Redundancy and inconsistency: the same fact in several files, corrected in one place and missed in another; a database stores each fact once (controlled redundancy).
- **Khó truy xuất:** mỗi câu hỏi mới phải ghép tệp bằng tay; CSDL cho một câu truy vấn hỏi nhiều bảng.
Poor retrieval: each new question means matching files by hand; a database lets one query ask across tables.
- **Khó giữ ràng buộc:** quy tắc nằm trong từng chương trình; CSDL giữ quy tắc trong chính nó.
Poor integrity: rules sit inside each program; a database holds the rules itself.
- **Bảo mật yếu:** tệp chỉ có mở hoặc không mở; CSDL phân quyền theo bảng, cột, khung nhìn.
Weak security: a file is open or closed; a database grants permissions by table, column and view.
- **Truy cập đồng thời:** hai người sửa một tệp thì chờ nhau hoặc đè nhau; CSDL có kiểm soát đồng thời.
Concurrent access: two people editing a file wait for each other or overwrite each other; a database has concurrency control.
- **Phụ thuộc cấu trúc tệp:** đổi cột phải sửa chương trình; CSDL cách ly chương trình và dữ liệu nhờ từ điển dữ liệu.
Dependence on file layout: changing a column means changing programs; a database insulates programs from data through the data dictionary.

Câu 3. Một công ty ví điện tử giữ CSDL khách hàng. Nêu các trách nhiệm của công ty theo Luật Bảo vệ dữ liệu cá nhân số 91/2025/QH15 và việc cụ thể phải làm ở CSDL.

Question 3. *An e-wallet company keeps a customer database. State the company's duties under the Law on Personal Data Protection No. 91/2025/QH15 and what has to be done in the database.*

- **Xác định dữ liệu cá nhân (Điều 2 khoản 1):** họ tên, ngày sinh, số điện thoại giúp xác định một người; công ty là bên kiểm soát dữ liệu cá nhân (khoản 7), khách hàng là chủ thể (khoản 5).
Identify personal data (Article 2, clause 1): name, date of birth and phone number help identify a person; the company is the controller (clause 7) and the customer is the data subject (clause 5).
- **Chỉ thu thập, xử lý đúng phạm vi và mục đích cụ thể, rõ ràng (Điều 3 khoản 2):** chỉ giữ các cột cần cho vận hành ví.

Collect and process only within a specific, clear scope and purpose (Article 3, clause 2): keep only the columns the wallet needs.

- Giữ dữ liệu chính xác, sửa khi cần, lưu trong thời gian phù hợp (Điều 3 khoản 3): sửa một chỗ duy nhất, đặt thời hạn lưu.
Keep data accurate, correct it when needed, store it for a fitting period (Article 3, clause 3): correct in one place only and set a retention period.
- Áp dụng biện pháp quản lý và kỹ thuật phù hợp (Điều 37 khoản 1 điểm c): phân quyền, khung nhìn che cột, sao lưu.
Apply suitable managerial and technical measures (Article 37, clause 1, point c): permissions, views that hide columns, backups.
- Trong tài chính, ngân hàng: thông báo cho chủ thẻ khi lộ hoặc mất thông tin tài khoản (Điều 27 khoản 1 điểm d).
In finance and banking: notify the data subject when account information is leaked or lost (Article 27, clause 1, point d).
- Không mua, bán dữ liệu cá nhân, không chiếm đoạt hay cố ý làm lộ (Điều 7 khoản 6 và 7).
Do not buy or sell personal data, nor seize or deliberately leak it (Article 7, clauses 6 and 7).

1.14. Câu hỏi trắc nghiệm – Multiple-choice questions

1. Mẫu nào sau đây là thông tin (không phải dữ liệu đơn lẻ)?

Which of the following is information (and not a single piece of data)?

- A. Mã ví V0001 ghi ở dòng sao kê đầu tiên / *The wallet code V0001 on the first statement line*
- B. Số tiền 2.000.000 đ ghi ở dòng sao kê đầu tiên / *The amount VND 2,000,000 on the first statement line*
- C. Ngày giờ 01/09/2026 08:15 ghi ở dòng sao kê đầu tiên / *The date and time 01/09/2026 08:15 on the first statement line*
- D. Tháng 9 có 11 lượt nạp thành công, tổng 17.500.000 đ / *September has 11 successful top-ups, VND 17,500,000 in total*

2. Lễ tân sửa số điện thoại của một khách ở tệp A nhưng tệp B vẫn ghi số cũ. Nhược điểm này của hệ thống tệp là gì?

Reception corrects a customer's phone number in file A but file B still holds the old one. What is this weakness of a file-based system?

- A. Mất tính độc lập giữa chương trình và dữ liệu / *Loss of independence between programs and data*
- B. Bảo mật yếu vì thiếu phân quyền cho từng cột / *Weak security from missing column-level permissions*
- C. Khó truy xuất vì dữ liệu nằm rải rác nhiều tệp / *Poor retrieval because data is scattered across files*
- D. Không nhất quán vì một dữ kiện bị ghi ở nhiều nơi / *Inconsistency because one fact is recorded in several places*

3. Phát biểu nào đúng về hệ quản trị cơ sở dữ liệu (HQTCSDL)?

Which statement about a database management system (DBMS) is correct?

- A.** Là tập dữ liệu có tổ chức dùng chung của một công ty / *It is the organised set of shared data of a company*
- B.** Là phần mềm đứng giữa người dùng và dữ liệu, nhận mọi yêu cầu đọc ghi / *It is software standing between users and data, receiving every request to read or write*
- C.** Là toàn bộ máy chủ, ứng dụng, quy trình và con người của hệ thống / *It is the whole of the servers, applications, procedures and people of the system*
- D.** Là ngôn ngữ dùng để ra lệnh thêm, sửa, xóa dữ liệu / *It is the language used to order data to be added, changed or deleted*

4. Sau khi thêm cột Website vào bảng ngân hàng, câu "SELECT MaNH, TenNH FROM NganHang" vẫn trả đủ 4 dòng như cũ. Hiện tượng này minh họa đặc trưng nào?

After a Website column is added to the bank table, the query "SELECT MaNH, TenNH FROM NganHang" still returns the same 4 rows. Which feature does this illustrate?

- A.** Kiểm soát đồng thời nhiều người dùng / *Concurrency control for many users*
- B.** Bảo mật theo phân quyền cho từng người / *Security through per-person permissions*
- C.** Cách ly giữa chương trình và dữ liệu / *Insulation between programs and data*
- D.** Sao lưu và phục hồi sau sự cố / *Backup and recovery after a failure*

5. Khung nhìn KháchLeTan chỉ chọn bốn cột MaKH, HoTen, TinhThanh, Hang của bảng KháchHang (8 cột). Mục đích chính của khung nhìn này là gì?

The view KháchLeTan selects only four columns, MaKH, HoTen, TinhThanh and Hang, from the KháchHang table (8 columns). What is the main purpose of this view?

- A.** Cho lễ tân thấy phần dữ liệu họ cần và giấu số điện thoại / *To show the front desk the data it needs and hide the phone number*
- B.** Lập một bản sao riêng của bảng để dùng làm bản sao lưu / *To make a separate copy of the table for use as a backup*
- C.** Tăng thêm số cột của bảng khách hàng lên bốn cột nữa / *To add four more columns to the customer table*
- D.** Xóa các cột còn lại ra khỏi bảng khách hàng gốc / *To delete the remaining columns from the original customer table*

6. Cuối quý, công ty đọc khối dữ liệu giao dịch của nhiều năm để tổng hợp báo cáo và ít khi ghi mới. Loại hệ theo mục đích phù hợp nhất là gì?
At quarter end, a company reads transaction data covering many years to compile a report and seldom writes anything new. Which type of system by purpose fits best?
- A. Hệ OLTP, phục vụ ghi giao dịch nhỏ hằng ngày / *An OLTP system, serving small daily transactions*
 - B. Hệ một người dùng, mỗi lúc phục vụ một người / *A single-user system, serving one person at a time*
 - C. Hệ OLAP, phục vụ phân tích và làm báo cáo / *An OLAP system, serving analysis and reports*
 - D. Hệ phân tán đồng nhất, cùng loại phần mềm ở mọi nơi / *A homogeneous distributed system, the same software everywhere*
7. Người cài đặt máy chủ, cấp quyền cho nhân viên, giám sát và sao lưu CSDL hằng đêm thuộc nhóm nào?
Which group does the person belong to who installs the server, grants access to staff, monitors and backs up the database nightly?
- A. Người thiết kế CSDL / *Database designer*
 - B. Lập trình viên ứng dụng / *Application programmer*
 - C. Quản trị CSDL (DBA) / *Database administrator (DBA)*
 - D. Người dùng cuối thạo / *Sophisticated end user*
8. Ví Trường Tiền thu phí 3.300 đ cho mỗi lần rút. Trong tháng 9 có 5 lượt rút thành công. Tổng phí rút thu được là bao nhiêu?
The Trường Tiền wallet charges VND 3,300 per withdrawal. In September there were 5 successful withdrawals. What is the total withdrawal fee collected?
- A. 16.500 đ / *VND 16,500*
 - B. 6.800.000 đ / *VND 6,800,000*
 - C. 3.300 đ / *VND 3,300*
 - D. 33.000 đ / *VND 33,000*

Chương 2. Mô hình dữ liệu và mô hình quan hệ – Chapter 2. Data models and the relational model

Mục tiêu chương – Chapter objectives

- Giải thích mô hình dữ liệu là gì và phân biệt ba mức mô hình: khái niệm, logic, vật lý.
Explain what a data model is and tell the three levels of modelling apart: conceptual, logical and physical.
- So sánh các loại mô hình (phân cấp, mạng, quan hệ, hướng đối tượng, NoSQL) và chọn được mô hình hợp với một bài toán.
Compare the kinds of model (hierarchical, network, relational, object-oriented, NoSQL) and pick the one that fits a problem.
- Mô tả kiến trúc ba lược đồ ANSI/SPARC và hai kiểu độc lập dữ liệu, kèm ví dụ chạy được trên PostgreSQL.
Describe the ANSI/SPARC three-schema architecture and the two kinds of data independence, with examples that run on PostgreSQL.
- Đọc một bảng thành quan hệ: chỉ ra bộ, thuộc tính, miền, bậc, lực lượng; nêu tính chất của quan hệ và cách xử lý giá trị NULL.
Read a table as a relation: name the tuples, attributes, domains, degree and cardinality; state the properties of a relation and how NULL behaves.
- Phân biệt siêu khóa, khóa ứng viên, khóa chính, khóa thay thế, khóa ngoại và tự tìm khóa cho một bảng từ quy tắc nghiệp vụ.
Tell superkey, candidate key, primary key, alternate key and foreign key apart, and find the keys of a table from its business rules.

2.1. Mô hình dữ liệu là gì – 2.1. What a data model is

Bạn mở một quán trà sữa. Trước khi mua phần mềm, bạn lấy giấy vẽ ra: có khách, có đơn, có món; mỗi đơn gồm nhiều món; mỗi món có một giá. Tờ giấy ấy chưa chạy được, nhưng ai nhìn vào cũng hiểu quán cần ghi nhớ những gì. Mô hình dữ liệu (data model) là cách vẽ và cách nói đã chuẩn hóa cho tờ giấy đó.

You open a bubble tea shop. Before buying any software you sketch it on paper: customers, orders and drinks, each order holding several drinks, each drink with a price. The sketch cannot run, yet anyone who reads it knows what the shop has to remember. A data model is a standard way of drawing and talking about a sketch like that.

Nói chính xác hơn, mô hình dữ liệu là một bộ khái niệm để mô tả ba thứ về dữ liệu. Thứ nhất là cấu trúc: dữ liệu gồm những loại gì và nối với nhau ra sao. Thứ hai là ràng buộc (constraint): điều kiện dữ liệu phải thỏa, chẳng hạn số dư ví không được âm. Thứ ba là thao tác (operation): những việc được phép làm, như thêm, tìm, sửa, xóa. Bảng sau gắn ba thứ ấy vào ví điện tử Trường Tiền.

More exactly, a data model is a set of concepts for describing three things about data. The first is structure: what kinds of data there are and how they connect. The second is constraints: conditions the data must meet, such as a wallet balance never dropping under zero. The third is

operations: what you are allowed to do, such as adding, finding, changing and deleting. The table that follows ties the three to the Trường Tiền e-wallet.

Thành phần <i>Component</i>	Trả lời câu hỏi <i>Question it answers</i>	Ví dụ ở ví Trường Tiền <i>Example in the Trường Tiền wallet</i>
Cấu trúc <i>Structure</i>	Dữ liệu gồm những gì, nối nhau thế nào? <i>What does the data consist of, and how is it linked?</i>	Một khách hàng có nhiều ví; một ví có nhiều giao dịch. <i>A customer has several wallets; a wallet has many transactions.</i>
Ràng buộc <i>Constraints</i>	Dữ liệu nào bị coi là sai? <i>Which data counts as wrong?</i>	Số dư không âm; mỗi mã ví chỉ xuất hiện một lần. <i>The balance is never negative; each wallet code appears once.</i>
Thao tác <i>Operations</i>	Được làm gì với dữ liệu? <i>What may be done with the data?</i>	Thêm giao dịch mới; tìm giao dịch của một ví; khóa ví. <i>Add a new transaction; find the transactions of a wallet; lock a wallet.</i>

Mô hình dữ liệu cần thiết vì mỗi người nhìn dữ liệu một kiểu. Nhân viên tín dụng quan tâm khoản vay, kế toán quan tâm phí, khách hàng quan tâm số dư. Một mô hình chung giúp mọi người đồng ý "hệ thống này nhớ những gì" trước khi ai đó viết dòng lệnh đầu tiên. Sửa một chỗ sai trên giấy dễ hơn nhiều so với sửa khi hệ thống đã chứa hàng triệu giao dịch.

A data model is needed because everyone looks at data differently. A credit officer cares about loans, an accountant about fees, a customer about the balance. A shared model lets everybody agree on "what this system remembers" before anyone writes the first line of code. Correcting a mistake on paper is far easier than correcting it once the system holds millions of transactions.

Từ ý tưởng đến hệ thống chạy được, người thiết kế đi qua ba bước. Hệ quản trị cơ sở dữ liệu (HQTCSDL, DBMS – database management system) là phần mềm quản lý kho dữ liệu, ví dụ PostgreSQL; bước thứ hai và thứ ba gắn với nó.

Between an idea and a running system, the designer goes through three steps. A database management system (DBMS, in Vietnamese HQTCSDL) is the software that looks after the stored data, for example PostgreSQL; the second and third steps depend on it.

- 1. Mô hình hóa dữ liệu (data modelling):** liệt kê đối tượng cần nhớ, mối liên hệ và điều kiện của chúng, chưa nghĩ tới phần mềm nào.
Data modelling: list the objects to remember, how they relate and the conditions on them, without yet thinking of any software.
- 2. Thiết kế logic:** viết lại kết quả bước 1 theo cách tổ chức của loại HQTCSDL đã chọn; nếu chọn kiểu bảng thì kết quả là các bảng và cột.
Logical design: rewrite the result of step 1 in the way the chosen kind of DBMS organises data; for a table-based system the result is tables and columns.
- 3. Thiết kế vật lý:** quyết định dữ liệu nằm ở đâu trên đĩa và tạo thêm cấu trúc phụ để tìm nhanh.

Physical design: decide where the data sits on disk and add extra structures so that it can be found quickly.

LỖI HAY GẶP / COMMON MISTAKE – Coi mô hình dữ liệu là một phần mềm hay một câu lệnh. Nó là bản mô tả bằng khái niệm và hình vẽ; phần mềm chỉ dùng lại bản mô tả đó.
Treating a data model as a program or a command. It is a description made of concepts and drawings; software merely follows it.

PHẢI THUỘC / MEMORISE – Mô hình dữ liệu = bộ khái niệm mô tả cấu trúc, ràng buộc và thao tác trên dữ liệu.
A data model = a set of concepts that describes the structure of data, the constraints on it and the operations on it.

2.2. Ba mức mô hình: khái niệm, logic, vật lý – 2.2. Three levels of modelling: conceptual, logical, physical

Muốn nấu một nồi canh chua, trước hết bạn nghĩ tới món: cần cá, rau, me. Rồi bạn viết công thức: bao nhiêu gram, cho gì trước, cho gì sau. Cuối cùng là chuyện trong bếp: nồi nào, lửa to hay nhỏ, đun bao lâu. Thiết kế cơ sở dữ liệu (CSDL, database) cũng đi từ ý chung đến chi tiết như vậy, qua ba mức.

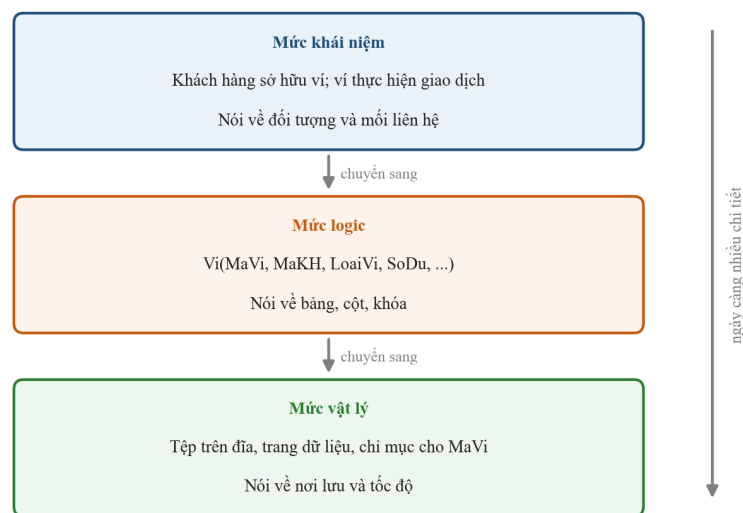
To cook a pot of sour soup you first think of the dish: fish, vegetables, tamarind. Then you write the recipe: how many grams, what goes in first and what next. Last comes the kitchen itself: which pot, high or low heat, how long. Designing a database follows the same path from general idea to detail, in three levels.

Trước khi đọc bảng, hãy nhớ một từ. Lược đồ (schema) là bản mô tả cấu trúc của CSDL ở một mức nào đó. Nó giống mẫu biểu của một bản sao kê ngân hàng: mẫu biểu ít đổi, còn con số điền vào đổi từng ngày.

One word first. A schema is the description of a database's structure at some level. It resembles the layout of a bank statement: the layout rarely changes, while the figures filled in change every day.

Mức Level	Câu hỏi Question	Mô tả bằng gì Described with	Ví dụ ở ví Trường Tiền Example in the wallet
Khái niệm (conceptual) <i>Conceptual</i>	Hệ thống cần nhớ những gì? <i>What must the system remember?</i>	Sơ đồ thực thể – kết hợp (ERD, entity–relationship diagram), học ở Chương 3; chưa phụ thuộc phần mềm. <i>An entity–relationship diagram (ERD), met in Chapter 3; independent of any software.</i>	Khách hàng sở hữu ví; ví thực hiện giao dịch. <i>A customer owns wallets; a wallet makes transactions.</i>
Logic (logical) <i>Logical</i>	Dữ liệu được tổ chức theo kiểu nào? <i>In what style is the data organised?</i>	Với hệ quan hệ: bảng, cột, khóa, ràng buộc; phụ thuộc kiểu mô hình, chưa phụ thuộc cách lưu. <i>For a relational system: tables, keys, constraints; dependent on the model type, not on the storage method.</i>	Vi(MaVi, MaKH, LoaiVi, SoDu, ...) với MaVi là khóa chính. <i>Vi(MaVi, MaKH, LoaiVi, SoDu, ...) with MaVi as primary key.</i>

Mức <i>Level</i>	Câu hỏi <i>Question</i>	Mô tả bằng gì <i>Described with</i>	Ví dụ ở ví Trường Tiền <i>Example in the wallet</i>
		<i>columns, keys and constraints; tied to the style of model, not yet to storage.</i>	<i>primary key.</i>
Vật lý (physical) <i>Physical</i>	Dữ liệu nằm ở đâu trên đĩa, tìm nhanh bằng cách nào? <i>Where does the data sit on disk, and how is it found quickly?</i>	Tệp, trang dữ liệu, chỉ mục, nén, mã hóa; phụ thuộc từng HQTCSĐL và phần cứng. <i>Files, data pages, indexes, compression and encryption; depends on the DBMS and the hardware.</i>	Một chỉ mục trên cột MaVi của bảng GiaoDich để tìm giao dịch của ví nhanh hơn. <i>An index on column MaVi of GiaoDich so that a wallet's transactions are found faster.</i>



Hình 2.1. Ba mức mô hình của ví Trường Tiền, đọc từ trên xuống. Ô xanh dương là ý chung, ô cam là bảng và cột, ô xanh lá là chuyện lưu trên đĩa; mũi tên bên phải chỉ chi tiết tăng dần.

Figure 2.1. The three levels of modelling for the Trường Tiền wallet, read from the top. The blue box is the general idea, the orange box is tables and columns, the green box is storage on disk; the arrow on the right shows detail increasing.

Mỗi mức có người làm riêng. Người phân tích và thiết kế lo mức khái niệm và logic. Quản trị viên CSDL (DBA, database administrator) lo mức vật lý: chọn chỉ mục, theo dõi dung lượng, sao lưu. Chương 8 và Chương 10 quay lại việc của DBA.

Each level has its own people. Analysts and designers handle the conceptual and logical levels. The database administrator (DBA) handles the physical level: choosing indexes, watching disk space, taking backups. Chapters 8 and 10 come back to the DBA's work.

Bài mẫu: xếp một câu mô tả vào đúng mức – Worked example: placing a statement at the right level

Đề: xếp mỗi câu sau vào mức khái niệm, logic hay vật lý. (a) Một khách hàng có thể mở nhiều ví. (b) Bảng Vi có cột SoDu kiểu numeric(15,0). (c) Tạo chỉ mục cho cột ThoiDien của bảng GiaoDich. (d) Dữ liệu trên ổ đĩa được mã hóa.

Task: place each statement at the conceptual, logical or physical level. (a) A customer may open several wallets. (b) Table Vi has a column SoDu of type numeric(15,0). (c) Create an index on column ThoiDiem of table GiaoDich. (d) The data on the disk is encrypted.

1. Hỏi: câu này nói về "cái gì tồn tại và liên hệ ra sao", hay về "bảng và cột", hay về "lưu trên đĩa"? Câu (a) nói về khách hàng và ví nên thuộc mức khái niệm.

Ask: does the statement say "what exists and how it relates", or "tables and columns", or "storage on disk"? Statement (a) speaks of customers and wallets, so it is conceptual.

2. Câu (b) nhắc tới bảng, cột và kiểu dữ liệu, tức cách tổ chức trong hệ quan hệ: mức logic.

Statement (b) mentions a table, a column and a data type, which is how a relational system organises data: logical.

3. Câu (c) nói tới chỉ mục, cấu trúc phụ chỉ để tìm nhanh; câu (d) nói tới mã hóa trên đĩa. Cả hai thuộc mức vật lý.

Statement (c) concerns an index, an extra structure that only speeds up searching; statement (d) concerns encryption on disk. Both are physical.

Kết quả: (a) khái niệm; (b) logic; (c) vật lý; (d) vật lý.

Result: (a) conceptual; (b) logical; (c) physical; (d) physical.

LỖI HAY GẶP / COMMON MISTAKE – Cho rằng mức khái niệm đã phải có bảng và cột. Mức đó mới nói đối tượng và mối liên hệ; bảng và cột chỉ xuất hiện ở mức logic.

Believing the conceptual level already needs tables and columns. That level only names objects and relationships; tables and columns appear at the logical level.

2.3. Các loại mô hình dữ liệu – 2.3. Kinds of data model

Mọi nhà đều cần cất đồ, nhưng có người dùng tủ nhiều ngăn kéo, có người dùng thùng có nhãn. Mô hình logic cũng vậy: mỗi loại chọn một cách sắp dữ liệu. Ta xem lần lượt sáu cách, mỗi cách gắn với cùng một nghiệp vụ: khách hàng, ví, giao dịch và cửa hàng.

Every household needs storage, but some people use a chest of drawers and others use labelled boxes. Logical models are the same: each kind arranges data in its own way. We look at six, each tied to the same business: customers, wallets, transactions and shops.

Mô hình phân cấp (hierarchical) – Hierarchical model

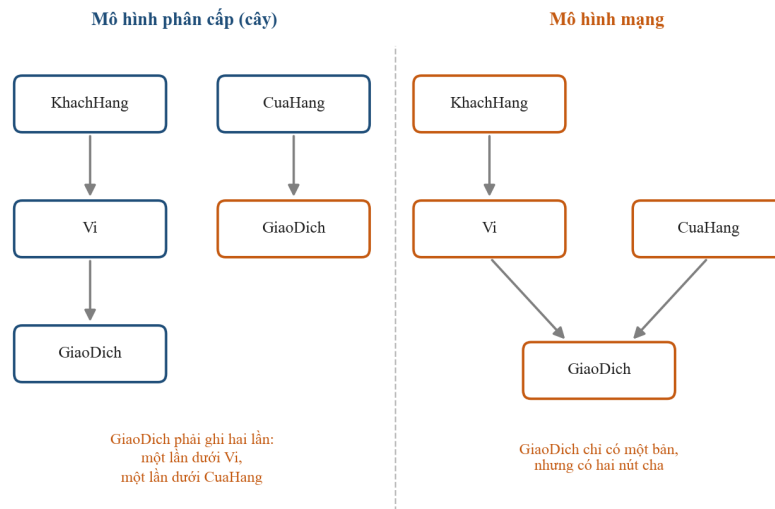
Hãy nghĩ tới thư mục trong máy tính: mỗi thư mục con nằm trong đúng một thư mục cha. Mô hình phân cấp tổ chức dữ liệu thành cây như vậy. Với ví Trường Tiền, ta có cây KháchHang, rồi Vi, rồi GiaoDich. Đi từ gốc xuống là thao tác nhanh. Khó khăn xuất hiện khi một giao dịch thanh toán còn thuộc về một cửa hàng: cây chỉ cho một cha, nên giao dịch phải được ghi thêm lần nữa dưới cửa hàng.

Think of folders on a computer: each subfolder sits inside exactly one parent folder. The hierarchical model arranges data as a tree in the same way. For the Trường Tiền wallet the tree runs KháchHang, then Vi, then GiaoDich. Going from the root downwards is quick. Trouble starts when a payment also belongs to a shop: a tree allows one parent only, so the transaction must be recorded again under the shop.

Mô hình mạng (network) – *Network model*

Mô hình mạng cho phép một nút có nhiều cha, giống mạng lưới tuyến xe buýt nơi một bến có nhiều tuyến đi qua. Giao dịch thanh toán khi đó chỉ có một bản, nối tới ví và nối tới cửa hàng. Cái giá phải trả: các nối kết là con trỏ cố định, người dùng phải tự chỉ đường (đi từ ví, theo nối kết này, tới giao dịch), và đổi cấu trúc thì phải sửa lại chương trình.

The network model lets a node have several parents, like a bus network in which one stop is served by several routes. A payment then exists once, linked to a wallet and linked to a shop. The price is that the links are fixed pointers: the user must give directions (start at the wallet, follow this link, arrive at the transaction), and changing the structure means rewriting programs.



Hình 2.2. Cùng dữ liệu, hai cách tổ chức. Bên trái, cây buộc giao dịch xuất hiện hai lần (ô cam là bản sao). Bên phải, mạng cho giao dịch hai nút cha là ví và cửa hàng.

Figure 2.2. The same data, two ways of organising it. On the left, the tree forces the transaction to appear twice (the orange box is the copy). On the right, the network gives the transaction two parent nodes, the wallet and the shop.

Mô hình quan hệ (relational) – *Relational model*

Mô hình quan hệ chỉ dùng một loại cấu trúc: bảng. Các bảng liên hệ với nhau bằng giá trị trùng nhau, không bằng con trỏ. Bảng GiaoDich có cột MaVi mang mã ví và cột MaCH mang mã cửa hàng; muốn biết giao dịch thuộc ví nào, bạn so mã. Vì thế bạn chỉ cần nói mình muốn gì, không cần chỉ đường. Đây là mô hình ta học kỹ ở các mục sau và dùng suốt sách.

The relational model uses a single kind of structure: the table. Tables are related by matching values, not by pointers. Table GiaoDich has a column MaVi holding a wallet code and a column MaCH holding a shop code; to learn which wallet a transaction belongs to, you compare codes. So you only say what you want, never how to get there. This is the model studied closely in the sections that follow and used throughout the book.

Mô hình hướng đối tượng (object-oriented) – *Object-oriented model*

Lập trình viên quen lưu "đối tượng": một đối tượng *Vi* có dữ liệu bên trong và có các hành vi như nạp tiền, rút tiền, lại có thể kế thừa từ một đối tượng tổng quát hơn. Mô hình hướng đối tượng lưu thẳng các đối tượng đó vào CSDL, để khỏi đổi qua lại giữa đối tượng và bảng. Loại hệ này ít phổ biến hơn hệ quan hệ; ngược lại, nhiều hệ quan hệ như PostgreSQL đã thêm kiểu mảng và kiểu JSON (cách ghi dữ liệu thành cặp tên và giá trị) cho nhu cầu tương tự.

Programmers are used to keeping "objects": a Vi object holds data inside and has behaviours such as depositing and withdrawing, and it may inherit from a more general object. The object-oriented model stores those objects directly in the database, so there is no switching back and forth between objects and tables. Systems of this kind are less common than relational ones; in turn, many relational systems such as PostgreSQL have added array and JSON types (JSON writes data as name and value pairs) for similar needs.

NoSQL: tài liệu và khóa – giá trị – *NoSQL: documents and key-value*

NoSQL là tên gọi chung (từ "not only SQL", tức ngoài SQL còn có cách khác) cho các hệ ra đời để xử lý dữ liệu cực lớn hoặc có nhiều dạng. Ngôn ngữ truy vấn có cấu trúc SQL (Structured Query Language) là ngôn ngữ ta dùng với hệ quan hệ, xem từ Chương 8. Hai kiểu NoSQL hay gặp nhất:

NoSQL is a collective name (from "not only SQL", meaning there are ways besides SQL) for systems that appeared to handle huge or highly varied data. The Structured Query Language, SQL, is the language we use with relational systems, from Chapter 8 on. The two most common NoSQL kinds are:

- Kiểu tài liệu (document): mỗi bản ghi là một tài liệu dạng JSON, có thể chứa các phần lồng nhau và mỗi tài liệu có thể có trường khác nhau. MongoDB là một ví dụ.

Document: each record is a JSON document that may contain nested parts, and each document may have different fields. MongoDB is one example.

- Kiểu khóa – giá trị (key-value): như tủ gửi đồ đánh số, đưa số là lấy đúng món đồ, nhưng không hỏi được "những món đồ màu đỏ". Redis là một ví dụ.

Key-value: like numbered lockers, give the number and you get the item, but you cannot ask for "all the red items". Redis is one example.

Đây là cùng một khách hàng KH001 viết ở dạng tài liệu. Hai ví của khách nằm lồng trong tài liệu, không có bảng riêng.

Here is the same customer KH001 written as a document. The two wallets of the customer are nested inside it, with no table of their own.

```
{ "maKH": "KH001", "hoTen": "Trần Minh Khôi", "hang": "Vàng",
  "vi": [
    { "maVi": "V0001", "loai": "Thanh toán", "trangThai": "Hoạt động" },
    { "maVi": "V0002", "loai": "Tiết kiệm", "trangThai": "Hoạt động" }
  ] }
```

Bảng sau đặt sáu mô hình cạnh nhau. Cột "Liên kết bằng" cho thấy khác biệt gốc: con trỏ hay giá trị.

The table that follows puts the six models side by side. The "Linked by" column shows the root difference: pointers or values.

Mô hình <i>Model</i>	Cách tổ chức <i>Organisation</i>	Liên kết bằng <i>Linked by</i>	Hợp với <i>Suits</i>	Hạn chế <i>Limitation</i>
Phân cấp <i>Hierarchical</i>	Cây, mỗi nút một cha <i>Tree, one parent per node</i>	Nói cha – con <i>Parent–child links</i>	Dữ liệu vốn là cây, như thư mục. <i>Data that is naturally a tree, such as folders.</i>	Khó biểu diễn nút có nhiều cha. <i>Hard to show a node with several parents.</i>
Mạng <i>Network</i>	Đồ thị, nút nhiều cha <i>Graph, nodes with several parents</i>	Con trỏ <i>Pointers</i>	Nối kết phức tạp, cố định. <i>Complex, fixed connections.</i>	Người dùng tự chỉ đường; khó đổi cấu trúc. <i>The user gives directions; structure is hard to change.</i>
Quan hệ <i>Relational</i>	Các bảng <i>Tables</i>	Giá trị trùng nhau <i>Matching values</i>	Dữ liệu giao dịch cần ràng buộc chặt, như ví và ngân hàng. <i>Transaction data that needs strict constraints, such as wallets and banks.</i>	Bảng cứng nhắc khi mỗi bản ghi một khác. <i>Tables are rigid when every record differs.</i>
Hướng đối tượng <i>Object-oriented</i>	Đối tượng có hành vi <i>Objects with behaviours</i>	Tham chiếu đối tượng <i>Object references</i>	Ứng dụng viết bằng ngôn ngữ hướng đối tượng. <i>Applications written in an object-oriented language.</i>	Ít phổ biến, ít công cụ. <i>Less common, fewer tools.</i>
Tài liệu <i>Document</i>	Tài liệu JSON lồng nhau <i>Nested JSON documents</i>	Lồng hoặc mã tham chiếu <i>Nesting or reference codes</i>	Hồ sơ mỗi bản một kiểu. <i>Records that each look different.</i>	Ràng buộc giữa các tài liệu yếu. <i>Weak constraints between documents.</i>
Khóa – giá trị <i>Key–value</i>	Cặp khóa và giá trị <i>Key and value pairs</i>	Không có <i>None</i>	Phiên đăng nhập, bộ nhớ đệm. <i>Login sessions, caches.</i>	Chỉ tra theo khóa. <i>Lookup by key only.</i>

Bài mẫu: chọn mô hình cho từng bài toán – *Worked example: choosing a model for each problem*

Đề: chọn mô hình hợp nhất cho (a) sổ cái giao dịch của ví, (b) phiên đăng nhập tạm của ứng dụng ví, (c) hồ sơ khảo sát khách hàng mà mỗi người trả lời một bộ câu hỏi khác nhau.

Task: choose the best-fitting model for (a) the wallet's transaction ledger, (b) the app's temporary login sessions, (c) customer survey files in which each person answers a different set of questions.

1. Với mỗi bài toán, hỏi hai câu: dữ liệu có cần ràng buộc chặt không, và có cần tìm theo nhiều tiêu chí không?
For each problem ask two questions: does the data need strict constraints, and must it be searched on many criteria?
2. (a) cần chặt (số dư không âm, mỗi giao dịch phải thuộc một ví có thật) và cần tìm theo nhiều tiêu chí: mô hình quan hệ.
(a) needs strict constraints (no negative balance, every transaction belongs to a real wallet) and many search criteria: relational.
3. (b) chỉ tra theo mã phiên, mất thì đăng nhập lại: khóa – giá trị. (c) mỗi hồ sơ một dạng, ít ràng buộc: tài liệu.
(b) is looked up by session code only and is simply recreated by logging in again: key–value. (c) has a different shape per file and few constraints: document.

LỖI HAY GẶP / COMMON MISTAKE – Chọn mô hình theo mắt thay vì theo bài toán, hoặc cho rằng NoSQL luôn nhanh hơn. Dữ liệu tiền bạc cần ràng buộc chặt chẽ, nên mô hình quan hệ vẫn là lựa chọn quen thuộc.

Choosing a model by fashion instead of by the problem, or assuming NoSQL is always faster. Money data needs strict constraints, which is why the relational model remains the usual choice.

2.4. Kiến trúc ba lược đồ và tính độc lập dữ liệu – 2.4. The three-schema architecture and data independence

Cùng một tòa nhà, người bán hàng nhìn thấy sảnh và quầy, người bảo vệ nhìn thấy cổng và camera, còn kỹ sư xây dựng nhìn thấy móng và cột. Từ giữa những năm 1970, nhóm SPARC thuộc Viện Tiêu chuẩn Quốc gia Hoa Kỳ (ANSI) đề xuất chia CSDL thành ba mức nhìn tương tự, gọi là kiến trúc ba lược đồ ANSI/SPARC.

In one building the shop assistant sees the lobby and counters, the guard sees the gate and cameras, and the structural engineer sees the foundations and columns. From the mid-1970s the SPARC group of the American National Standards Institute (ANSI) proposed splitting a database into three similar ways of seeing it, known as the ANSI/SPARC three-schema architecture.

- Lược đồ ngoài (external schema): phần dữ liệu mà một nhóm người dùng nhìn thấy, gọi là khung nhìn (view). Mỗi nhóm có một lược đồ ngoài riêng và nhiều nhóm có thể cùng tồn tại.
External schema: the part of the data one group of users sees, called a view. Each group has its own external schema, and there can be many.
- Lược đồ khái niệm (conceptual schema): mô tả toàn bộ CSDL cho cả tổ chức, gồm các thực thể, mối liên hệ và ràng buộc, chưa nói tới cách lưu. Chỉ có một lược đồ khái niệm.
Conceptual schema: describes the whole database for the whole organisation, including entities, relationships and constraints, but not how it is stored. There is only one.

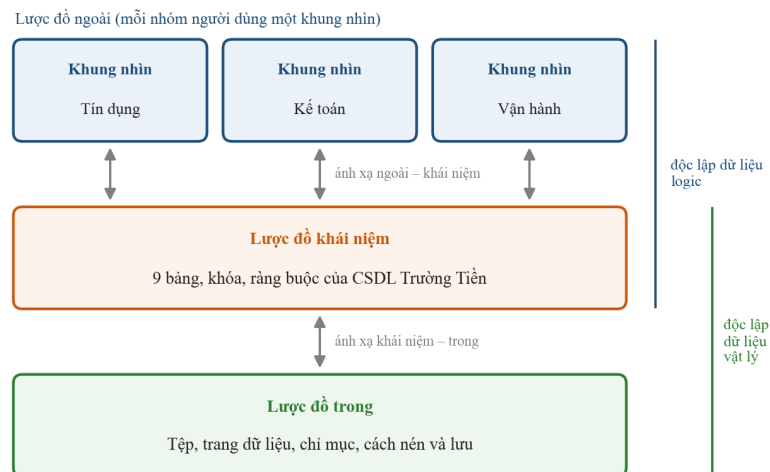
- **Lược đồ trong (internal schema):** cách dữ liệu được lưu trên đĩa, gồm tệp, chỉ mục, nén. Cũng chỉ có một.

Internal schema: how the data is stored on disk, covering files, indexes and compression.

There is also only one.

Giữa các mức có hai ánh xạ (mapping): HQTCSDL tự dịch một câu hỏi viết trên khung nhìn thành câu hỏi trên lược đồ khái niệm, rồi thành thao tác trên tệp. Hình 2.3 vẽ kiến trúc này cho ba phòng của công ty Trường Tiền.

Between the levels sit two mappings: the DBMS translates a question written against a view into a question on the conceptual schema, and then into operations on files. Figure 2.3 draws this architecture for three departments of the Trường Tiền company.



Hình 2.3. Ba lược đồ ANSI/SPARC. Hàng trên là ba khung nhìn của ba phòng ban; ở giữa là lược đồ khái niệm duy nhất; dưới cùng là lược đồ trong. Dấu ngoặc bên phải chỉ chỗ nào đổi mà mức trên không phải sửa.

Figure 2.3. The three ANSI/SPARC schemas. The top row holds the views of three departments, the middle is the single conceptual schema, the bottom is the internal schema. The brackets on the right show where a change leaves the level above untouched.

Hai bộ tên này dễ lẫn. Ba mức mô hình ở mục 2.2 là ba giai đoạn thiết kế. Ba lược đồ ở đây là ba mức nhìn trong một hệ đã chạy. Với hệ quan hệ, lược đồ khái niệm chính là tập các bảng của mức logic, lược đồ trong ứng với mức vật lý, còn lược đồ ngoài là các khung nhìn.

The two sets of names are easy to mix up. The three levels of modelling in section 2.2 are three stages of design. The three schemas here are three ways of seeing a system that is already running. In a relational system the conceptual schema is the set of tables of the logical level, the internal schema corresponds to the physical level, and the external schemas are views.

Ba mức mô hình (thiết kế) <i>Three modelling levels (design)</i>	Ba lược đồ ANSI/SPARC (vận hành) <i>ANSI/SPARC schemas (operation)</i>	Trong ví Trường Tiền <i>In the wallet</i>
Khái niệm, rồi logic <i>Conceptual, then logical</i>	Lược đồ khái niệm <i>Conceptual schema</i>	9 bảng cùng khóa và ràng buộc. <i>The 9 tables with their keys and constraints.</i>
Không có mức tương ứng <i>No matching level</i>	Lược đồ ngoài <i>External schema</i>	Khung nhìn cho phòng Tín dụng, Kế toán, Vận hành. <i>Views for the Credit, Accounting and Operations departments.</i>
Vật lý <i>Physical</i>	Lược đồ trong <i>Internal schema</i>	Tập dữ liệu và chỉ mục của PostgreSQL. <i>PostgreSQL data files and indexes.</i>

Nhờ ánh xạ, ta có hai kiểu độc lập dữ liệu (data independence): chương trình và khung nhìn không phải sửa khi mức dưới thay đổi.

Thanks to the mappings we get two kinds of data independence: programs and views need no change when a lower level changes.

- Độc lập dữ liệu logic (logical data independence): đổi lược đồ khái niệm, chẳng hạn thêm một cột, mà không phải sửa lược đồ ngoài hay chương trình đang dùng.
Logical data independence: change the conceptual schema, for example by adding a column, without having to change the external schemas or the programs that use them.
- Độc lập dữ liệu vật lý (physical data independence): đổi lược đồ trong, chẳng hạn thêm chỉ mục hay chuyển tệp sang ổ khác, mà lược đồ khái niệm giữ nguyên.
Physical data independence: change the internal schema, for example by adding an index or moving a file to another drive, while the conceptual schema stays the same.

Đoạn lệnh sau thử cả hai. Ta tạo một khung nhìn cho phòng Tín dụng, thêm một cột vào bảng gốc, rồi tạo một chỉ mục. Sau mỗi thay đổi, cùng một câu hỏi vẫn trả cùng một kết quả. Ba lệnh CREATE VIEW, ALTER TABLE, CREATE INDEX sẽ được giải thích kỹ ở Chương 8; ở đây chỉ cần xem kết quả.

The next script tries both. We create a view for the Credit department, add a column to the base table, then create an index. After each change the same question still gives the same answer. The commands CREATE VIEW, ALTER TABLE and CREATE INDEX are explained in Chapter 8; here only the results matter.

```
CREATE VIEW VTinDung AS
  SELECT MaKH, HoTen, Hang FROM KhachHang;
SELECT COUNT(*) AS khách_vàng FROM VTinDung WHERE Hang = 'Vàng';
ALTER TABLE KhachHang ADD COLUMN Email varchar(60);
SELECT COUNT(*) AS khách_vàng FROM VTinDung WHERE Hang = 'Vàng';
CREATE INDEX idxHang ON KhachHang (Hang);
SELECT COUNT(*) AS khách_vàng FROM VTinDung WHERE Hang = 'Vàng';
```

Kết quả chạy / Output

```
CREATE VIEW
khach_vang
-----
          3
(1 row)

ALTER TABLE
khach_vang
-----
          3
(1 row)

CREATE INDEX
khach_vang
-----
          3
(1 row)
```

Cả ba lần đều ra 3 khách hạng Vàng. Cột Email mới và chỉ mục mới không làm khung nhìn hỏng. Giới hạn: nếu bạn bỏ chính cột mà khung nhìn đang dùng, PostgreSQL từ chối vì có khung nhìn phụ thuộc, nên độc lập logic không phải tuyệt đối.

All three runs return 3 Gold-rank customers. The new column Email and the new index do not break the view. A limit: if you drop a column the view itself uses, PostgreSQL refuses because a view depends on it, so logical independence is not absolute.

Bài mẫu: thay đổi nào ảnh hưởng mức nào – *Worked example: which change touches which level*

Đề: với mỗi thay đổi, cho biết nó đổi lược đồ nào và khung nhìn của phòng Tín dụng có phải sửa không. (a) Thêm cột Email vào KháchHang. (b) Chuyển bảng GiaoDich sang ổ đĩa nhanh hơn. (c) Bỏ cột Hang khỏi KháchHang.

Task: for each change, say which schema it alters and whether the Credit department's view must be edited. (a) Add column Email to KháchHang. (b) Move table GiaoDich to a faster disk. (c) Drop column Hang from KháchHang.

- Với từng thay đổi, hỏi nó nói về cấu trúc bảng hay về chỗ lưu. (a) và (c) đổi cấu trúc bảng nên đổi lược đồ khái niệm; (b) chỉ đổi chỗ lưu nên đổi lược đồ trong.
For each change ask whether it concerns table structure or storage. (a) and (c) alter table structure and so the conceptual schema; (b) alters only storage and so the internal schema.
- Hỏi tiếp: khung nhìn VTinDung có dùng phần bị đổi không? Nó dùng MaKH, HoTen, Hang. (a) không đụng tới ba cột này; (b) không đụng tới cấu trúc; (c) bỏ đúng cột Hang.

Next ask whether the view VTinDung uses the part that changed. It uses MaKH, HoTen and Hang. (a) touches none of the three; (b) touches no structure; (c) removes exactly Hang.

Kết quả: (a) lược đồ khái niệm, khung nhìn không phải sửa. (b) lược đồ trong, khung nhìn không phải sửa. (c) lược đồ khái niệm, khung nhìn phải sửa vì mất cột Hang.

Result: (a) conceptual schema, the view needs no edit. (b) internal schema, the view needs no edit. (c) conceptual schema, the view must be edited because column Hang is gone.

LỖI HAY GẶP / COMMON MISTAKE – Nhầm lược đồ với dữ liệu. Lược đồ là khung ít đổi (tên bảng, cột, kiểu); dữ liệu là các dòng đổi từng giây. Thêm một giao dịch không làm đổi lược đồ.

Confusing schema with data. The schema is the rarely changing frame (table names, columns, types); the data is the rows that change every second. Adding a transaction does not change the schema.

PHẢI THUỘC / MEMORISE – Ba lược đồ: ngoài (khung nhìn, nhiều cái), khái niệm (toàn CSDL, một cái), trong (cách lưu, một cái). Độc lập logic: đổi khái niệm không sửa ngoài. Độc lập vật lý: đổi trong không sửa khái niệm.

Three schemas: external (views, many), conceptual (the whole database, one), internal (storage, one). Logical independence: changing the conceptual schema leaves the external ones alone. Physical independence: changing the internal schema leaves the conceptual one alone.

2.5. Mô hình quan hệ: quan hệ, bộ, thuộc tính, miền – 2.5. The relational model: relation, tuple, attribute, domain

Năm 1970, nhà toán học E. F. Codd công bố mô hình quan hệ (relational model). Ý của ông giản dị: mọi dữ liệu đều có thể đặt vào các bảng, và bảng có thể được xử lý bằng toán tập hợp và logic. Nhờ nền toán đó, ta chứng minh được nhiều điều về truy vấn và thiết kế. Hầu hết hệ CSDL thương mại hiện nay, trong đó có PostgreSQL, đi theo mô hình này.

In 1970 the mathematician E. F. Codd published the relational model. His idea was simple: all data can be put into tables, and tables can be handled with set theory and logic. That mathematical base lets us prove many facts about queries and design. Most commercial database systems today, PostgreSQL among them, follow this model.

Mỗi khái niệm của mô hình có hai tên: tên toán học và tên đời thường. Bảng sau ghép chúng lại. Từ đây về sau sách dùng cả hai, nên hãy học thuộc cột đầu.

Each concept of the model has two names: a mathematical one and an everyday one. The table that follows pairs them. The book uses both from here on, so learn the first column well.

Thuật ngữ mô hình <i>Model term</i>	Tên thường gặp <i>Everyday name</i>	Nghĩa <i>Meaning</i>
Quan hệ (relation) <i>Relation</i>	Bảng (table), tệp <i>Table, file</i>	Một bảng có tên, ví dụ KháchHang. <i>A named table, for example KháchHang.</i>
Bộ (tuple) <i>Tuple</i>	Dòng (row), bản ghi <i>Row, record</i>	Một dòng dữ liệu, ví dụ thông tin một khách hàng. <i>One row of data, for example the details of one customer.</i>
Thuộc tính (attribute) <i>Attribute</i>	Cột (column), trường <i>Column, field</i>	Một đặc điểm cần ghi, ví dụ HoTen. <i>One feature to record, for example HoTen.</i>

Thuật ngữ mô hình <i>Model term</i>	Tên thường gặp <i>Everyday name</i>	Nghĩa <i>Meaning</i>
Miền (domain) <i>Domain</i>	Kiểu dữ liệu và giá trị hợp lệ <i>Data type and allowed values</i>	Tập giá trị một thuộc tính được nhận. <i>The set of values an attribute may take.</i>
Bậc (degree) <i>Degree</i>	Số cột <i>Number of columns</i>	Số thuộc tính của quan hệ. <i>How many attributes the relation has.</i>
Lực lượng (cardinality) <i>Cardinality</i>	Số dòng <i>Number of rows</i>	Số bộ hiện có trong quan hệ. <i>How many tuples the relation holds at the moment.</i>

Miền cần nói kỹ hơn. Miền là tập các giá trị mà một thuộc tính được phép nhận. Miền của NgàySinh là các ngày hợp lệ; miền của GiớiTinh chỉ gồm hai giá trị Nam và Nữ. Trong PostgreSQL, miền được diễn tả bằng kiểu dữ liệu cộng với điều kiện kiểm tra. Bảng sau liệt kê bốn nhóm miền hay gặp.

Domains deserve a closer look. A domain is the set of values an attribute is allowed to take. The domain of NgàySinh is the valid dates; the domain of GiớiTinh holds just two values, Nam and Nữ. In PostgreSQL a domain is expressed by a data type plus a check condition. The table that follows lists four common groups of domain.

Nhóm miền <i>Kind of domain</i>	Ví dụ thuộc tính <i>Example attribute</i>	Kiểu trong PostgreSQL <i>PostgreSQL type</i>
Số <i>Number</i>	SoDu, SoTien: số tiền đồng <i>SoDu, SoTien: amounts in dong</i>	integer, numeric
Chuỗi ký tự <i>Text</i>	HoTen, TinhThanh <i>HoTen, TinhThanh</i>	varchar, char, text
Ngày giờ <i>Date and time</i>	NgàySinh, ThoiDiem <i>NgàySinh, ThoiDiem</i>	date, timestamp
Đúng hoặc sai <i>True or false</i>	Cờ như "đã xác minh danh tính" <i>A flag such as "identity verified"</i>	boolean

Một quan hệ là tập con của tích Descartes (Cartesian product) các miền. Tích Descartes nghe đáng sợ nhưng ý đơn giản: đó là mọi cách ghép một giá trị của miền này với một giá trị của miền kia. Ví dụ miền thứ nhất có hai tỉnh Huế, Đà Nẵng; miền thứ hai có hai ngành Ăn uống, Bán lẻ. Có $2 \times 2 = 4$ cách ghép. Một quan hệ chỉ giữ lại vài cách ghép thật sự xảy ra, chẳng hạn Huế đi với Ăn uống.

A relation is a subset of the Cartesian product of its domains. A Cartesian product sounds frightening but the idea is simple: it is every way of pairing a value from one domain with a value from another. Say the first domain holds two provinces, Huế and Đà Nẵng, and the second holds two trades, Ăn uống and Bán lẻ. There are $2 \times 2 = 4$ pairings. A relation keeps only the pairings that really occur, for example Huế with Ăn uống.

$$R(A_1, A_2, \dots, A_n) \quad (2.1)$$

$$r \subseteq \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n) \quad (2.2)$$

Công thức (2.1) viết khung của quan hệ R có n thuộc tính A1 đến An. Công thức (2.2) nói tập các dòng hiện có r của R là tập con của tích các miền, trong đó $\text{dom}(A)$ là miền của thuộc tính A. Đọc: "r chỉ chứa những bộ có thật".

Formula (2.1) writes the frame of a relation R with n attributes A1 to An. Formula (2.2) says the set r of rows R currently holds is a subset of the product of the domains, where $\text{dom}(A)$ is the domain of attribute A. Read it as "r holds only the tuples that really exist".

Đọc một bảng thành quan hệ – Reading a table as a relation

1. Lấy tên bảng làm tên quan hệ, ví dụ CuaHang.
Take the table's name as the relation's name, for example CuaHang.
2. Đếm số cột: đó là bậc. Gọi tên từng cột: đó là các thuộc tính.
Count the columns: that is the degree. Name each column: those are the attributes.
3. Với mỗi thuộc tính, ghi miền: kiểu dữ liệu và các giá trị cho phép.
For each attribute write the domain: the data type and the allowed values.
4. Đếm số dòng: đó là lực lượng. Mỗi dòng là một bộ.
Count the rows: that is the cardinality. Each row is a tuple.
5. Viết lược đồ dạng tên quan hệ và danh sách thuộc tính trong ngoặc.
Write the schema as the relation name followed by the attribute list in brackets.

Làm mẫu với bảng cửa hàng của ví. Hai truy vấn sau đếm số dòng rồi liệt kê các cột; bạn sẽ gặp lại cú pháp SELECT ở Chương 9, nay chỉ cần đọc kết quả. Tiêu đề cột trong kết quả psql hiện bằng chữ thường vì PostgreSQL đổi tên không đặt trong ngoặc kép sang chữ thường.

Here is a worked example with the shop table of the wallet. These two queries count the rows and then list the columns; you will meet SELECT properly in Chapter 9, so for now just read the output. Column headings in psql appear in lower case because PostgreSQL folds unquoted names to lower case.

```
SELECT COUNT(*) AS luc_luong FROM CuaHang;
SELECT column_name AS cot, data_type AS kieu
FROM information_schema.columns
WHERE table_name = 'cuahang'
ORDER BY ordinal_position;
```

Kết quả chạy / Output

```
luc_luong
-----
          6
(1 row)

   cot   |      kieu
-----+-----
mach     | character
tench    | character varying
nganhhang | character varying
tinhthanh | character varying
(4 rows)
```

Kết quả: lực lượng 6, bốn cột nên bậc 4. Lược đồ là CuaHang(MaCH, TenCH, NganhHang, TinhThanh). Miền của NganhHang chỉ gồm năm giá trị Ăn uống, Bán lẻ, Giáo dục, Y tế, Di chuyển. Hình 2.4 vẽ các phần này lên chính bảng.

Result: cardinality 6, four columns so degree 4. The schema is CuaHang(MaCH, TenCH, NganhHang, TinhThanh). The domain of NganhHang has only five values: Ăn uống, Bán lẻ, Giáo dục, Y tế, Di chuyển. Figure 2.4 marks these parts on the table itself.

bậc = 4 (bốn thuộc tính)

MaCH	TenCH	NganhHang	TinhThanh
CH01	Quán cơm hến Bến Ngự	Ăn uống	Huế
CH02	Tạp hóa Minh Anh	Bán lẻ	Đà Nẵng
CH03	Trung tâm Anh ngữ Sao Mai	Giáo dục	Huế
CH04	Nhà thuốc An Tâm	Y tế	Cần Thơ
CH05	Dịch vụ xe Hải Âu	Di chuyển	Hà Nội
CH06	Nhà sách Tri Thức	Bán lẻ	TP. Hồ Chí Minh

lực lượng = 6 (sáu bộ)

♦ bộ: một dòng

thuộc tính NganhHang, miền: Ăn uống, Bán lẻ, Giáo dục, Y tế, Di chuyển

Hình 2.4. Quan hệ CuaHang viết dưới dạng bảng. Dòng tiêu đề ghi các thuộc tính; mỗi dòng bên dưới là một bộ (dòng nền cam là bộ thứ ba). Dấu ngoặc phía trên ghi bậc 4, dấu ngoặc bên trái ghi lực lượng 6.

Figure 2.4. The relation CuaHang drawn as a table. The heading row holds the attributes; each row under it is a tuple (the shaded row is the third tuple). The bracket above gives degree 4 and the bracket on the left gives cardinality 6.

Còn một cặp khái niệm nữa: lược đồ quan hệ và thể hiện quan hệ. Lược đồ là khung, giống mẫu sao kê. Thể hiện (instance) là nội dung tại một thời điểm, giống sao kê của hôm nay. Bảng NganhHang luôn có lược đồ NganhHang(MaNH, TenNH); hôm nay nó có 4 bộ, mai thêm một ngân hàng thì có 5 bộ, nhưng lược đồ không đổi.

One more pair of ideas: relation schema and relation instance. The schema is the frame, like the layout of a statement. The instance is the content at one moment, like today's statement. Table NganhHang always has the schema NganhHang(MaNH, TenNH); today it has 4 tuples, tomorrow after one more bank is added it has 5, but the schema is unchanged.

Tiêu chí <i>Criterion</i>	Lược đồ quan hệ <i>Relation schema</i>	Thể hiện quan hệ <i>Relation instance</i>
Là gì <i>What it is</i>	Tên và danh sách thuộc tính cùng miền. <i>The name and the attributes with their domains.</i>	Tập bộ có mặt tại một thời điểm. <i>The set of tuples present at one moment.</i>
Đổi khi nào <i>When it changes</i>	Hiếm, khi sửa cấu trúc bảng. <i>Rarely, when the table structure is altered.</i>	Liên tục, mỗi lần thêm, sửa, xóa dòng. <i>Constantly, whenever a row is added, changed or deleted.</i>
Ví dụ <i>Example</i>	NganHang(MaNH, TenNH) <i>NganHang(MaNH, TenNH)</i>	{(NH01, Ngân hàng Đông Ba), ...} gồm 4 bộ. <i>{(NH01, Ngân hàng Đông Ba), ...}, 4 tuples.</i>

LỖI HAY GẶP / COMMON MISTAKE – Lẫn bậc với lực lượng. Bậc đếm cột, lực lượng đếm dòng. Chữ "lực lượng" ở đây cũng không phải "bản số" của mối kết hợp (1–1, 1–n) học ở Chương 3.

Mixing up degree and cardinality. Degree counts columns, cardinality counts rows. The word "cardinality" here is also not the "cardinality ratio" of a relationship (1–1, 1–n) met in Chapter 3.

PHẢI THUỘC / MEMORISE – Quan hệ = bảng; bộ = dòng; thuộc tính = cột; bậc = số cột; lực lượng = số dòng. Lược đồ là khung ít đổi, thể hiện là nội dung đổi liên tục.

Relation = table; tuple = row; attribute = column; degree = number of columns; cardinality = number of rows. The schema is the rarely changing frame, the instance is the content that keeps changing.

2.6. Tính chất của quan hệ và giá trị NULL – 2.6. Properties of a relation and the NULL value

Không phải bảng Excel nào cũng là quan hệ. Bảng điểm có hai dòng giống hệt nhau, ô điện thoại chứa hai số cách nhau dấu phẩy, ô trộn ghi "vắng" lẫn điểm số: với Excel thì ổn, với CSDL thì không. Một quan hệ phải thỏa bốn tính chất sau.

Not every Excel sheet is a relation. A gradebook with two identical rows, a phone cell holding two numbers separated by a comma, a column mixing the word "absent" with marks: fine for Excel, not for a database. A relation must have the following four properties.

Tính chất <i>Property</i>	Nghĩa là <i>What it means</i>	Excel hay vi phạm thế nào <i>How Excel sheets often break it</i>
1. Không có hai bộ trùng nhau <i>1. No two identical tuples</i>	Quan hệ là một tập hợp, mỗi phần tử xuất hiện một lần. <i>A relation is a set, and each element appears once.</i>	Nhập một dòng hai lần. <i>Typing a row twice.</i>

Tính chất <i>Property</i>	Nghĩa là <i>What it means</i>	Excel hay vi phạm thế nào <i>How Excel sheets often break it</i>
2. Thứ tự không quan trọng <i>2. Order does not matter</i>	Đổi chỗ các dòng hoặc các cột, quan hệ vẫn là chính nó. <i>Swap rows or columns and it is still the same relation.</i>	Dùng "dòng thứ 5" để chỉ một người. <i>Using "row 5" to point at a person.</i>
3. Giá trị nguyên tố <i>3. Atomic values</i>	Mỗi ô một giá trị đơn, không chia nhỏ thêm theo nhu cầu của hệ thống. <i>Each cell holds one value that the system never needs to split further.</i>	Ô "0901111222, 0902222333". <i>A cell reading "0901111222, 0902222333".</i>
4. Mỗi cột một miền, tên cột khác nhau <i>4. One domain per column, distinct names</i>	Mọi ô trong cột cùng loại; không có hai cột trùng tên. <i>All cells in a column are of one kind; no two columns share a name.</i>	Cột Điểm chứa cả số lẫn chữ "vắng". <i>A Mark column holding both numbers and the word "absent".</i>

Tính chất 3 cần hiểu đúng. "Nguyên tố" tùy theo cách hệ thống dùng dữ liệu. Nếu hệ thống không bao giờ cần tìm theo tên riêng, cột HoTen coi là nguyên tố. Nếu cần xếp theo tên riêng, nên tách thành họ đệm và tên. Chương 7 quay lại tính chất này dưới tên dạng chuẩn 1.

Property 3 needs care. "Atomic" depends on how the system uses the data. If the system never needs to search by given name, the column HoTen counts as atomic. If it must sort by given name, split it into family name and given name. Chapter 7 returns to this property under the name first normal form.

Hai tính chất đầu có thể thấy ngay bằng PostgreSQL. Thứ nhất, hai câu hỏi sau chỉ khác thứ tự dòng (ORDER BY là mệnh đề sắp xếp) nhưng cho cùng một tập bộ. Thứ hai, thử nhập thêm một ngân hàng mang mã NH01 đã có; HQTCSDL từ chối.

The first two properties can be seen at once in PostgreSQL. First, the two queries that follow differ only in row order (ORDER BY is the sorting clause) yet return the same set of tuples. Second, try adding a bank with the existing code NH01; the DBMS refuses.

```
SELECT MaNH FROM NganHang ORDER BY MaNH DESC;
SELECT MaNH FROM NganHang ORDER BY MaNH ASC;
```

Kết quả chạy / Output

```
manh
-----
NH04
NH03
NH02
NH01
(4 rows)

manh
-----
NH01
NH02
NH03
NH04
(4 rows)
```

```
INSERT INTO NganHang VALUES ('NH01', 'Ngân hàng Mới');
```

Kết quả chạy / Output

```
ERROR: duplicate key value violates unique constraint "pknganhang"
DETAIL: Key (manh)=(NH01) already exists.
```

Thông báo ERROR đến từ khóa chính của NganHang: mã đã tồn tại thì bị chặn, và một mã rỗng (NULL) cũng bị chặn theo cách tương tự. Mục 2.7 nói khóa chính là gì.

The ERROR message comes from the primary key of NganHang: a code that already exists is blocked, and an empty code (NULL) is blocked in a similar way. Section 2.7 explains what a primary key is.

Giá trị NULL – The NULL value

Có ô trong bảng mà chưa có gì để điền. Số 0 không đúng vì 0 là một số thật; chuỗi rỗng cũng không đúng. CSDL dùng một giá trị đặc biệt là NULL, nghĩa là "không có giá trị". NULL xuất hiện với ba lý do. Chưa biết: không rõ giá trị. Không áp dụng: thuộc tính vô nghĩa với bộ này. Chưa xảy ra: sự việc chưa đến.

Some cells have nothing to put in them yet. The number 0 is wrong because 0 is a real number; an empty string is wrong too. Databases use a special value, NULL, meaning "no value". NULL appears for three reasons. Unknown: the value is not known. Not applicable: the attribute makes no sense for this tuple. Not yet happened: the event has not arrived.

Cột trong CSDL Trường Tiền <i>Column in the Trường Tiền database</i>	NULL nghĩa là <i>NULL means</i>	Lý do <i>Reason</i>
KyTraNo.NgayTra	Kỳ này khách chưa trả. <i>The customer has not yet paid this instalment.</i>	Chưa xảy ra <i>Not yet happened</i>
GiaoDich.MaViNhan	Giao dịch Nạp, Rút, Thanh toán không có ví nhận. <i>Deposits, withdrawals and payments have no receiving wallet.</i>	Không áp dụng <i>Not applicable</i>
NhanVien.MaQuanLy	NV01 không có cấp trên. <i>NV01 has no superior.</i>	Không áp dụng <i>Not applicable</i>

Điều làm người mới bối rối: so sánh với NULL không cho đúng hay sai mà cho "không biết" (UNKNOWN). Hỏi "tuổi của An có bằng tuổi của Bình không" khi cả hai chưa khai tuổi, câu trả lời trung thực là không biết. Mệnh đề WHERE, phần lọc dòng của câu truy vấn, chỉ giữ những dòng cho kết quả đúng, nên điều kiện = NULL không bao giờ giữ dòng nào. Muốn tìm ô trống phải viết IS NULL. Hàm COUNT(cột) cũng bỏ qua các ô NULL, còn COUNT(*) đếm mọi dòng.

What puzzles newcomers: comparing with NULL gives neither true nor false but "unknown". Asked "is An's age equal to Bình's age" when neither has given an age, the honest answer is unknown. The WHERE clause, the row filter of a query, keeps only rows whose condition is true, so a condition = NULL never keeps any row. To find empty cells you must write IS NULL. The function COUNT(column) also skips NULL cells, whereas COUNT() counts every row.*

```
SELECT COUNT(*) AS tong, COUNT(NgayTra) AS da_tra FROM KyTraNo;
SELECT MaVay, KySo FROM KyTraNo WHERE NgayTra = NULL;
SELECT MaVay, KySo FROM KyTraNo WHERE NgayTra IS NULL;
```

Kết quả chạy / Output

```
tong | da_tra
-----+-----
  24 |      20
(1 row)

mavay | kyso
-----+-----
(0 rows)

mavay | kyso
-----+-----
KV03 |    3
KV03 |    4
KV05 |    5
KV05 |    6
(4 rows)
```

Đọc kết quả: bảng có 24 kỳ nhưng chỉ 20 kỳ có ngày trả, nên 4 kỳ còn lại là NULL. Điều kiện = NULL cho 0 dòng; IS NULL tìm đúng 4 kỳ chưa trả, thuộc KV03 và KV05.

Reading the output: the table has 24 instalments but only 20 have a payment date, so the other 4 are NULL. The condition = NULL returns 0 rows; IS NULL finds exactly the 4 unpaid instalments, belonging to KV03 and KV05.

Biểu thức <i>Expression</i>	Kết quả <i>Result</i>	Vì sao <i>Why</i>
NULL = NULL	UNKNOWN	Hai giá trị chưa biết chưa chắc bằng nhau. <i>Two unknown values are not necessarily equal.</i>
5 + NULL	NULL	Phép tính với giá trị chưa biết vẫn chưa biết. <i>Arithmetic on an unknown stays unknown.</i>
NULL IS NULL	TRUE	Đây là câu hỏi "ô này có trống không", có trả lời chắc chắn. <i>This asks "is the cell empty", which has a definite answer.</i>
FALSE AND UNKNOWN	FALSE	Một vế sai thì cả AND sai, bất kể vế kia. <i>One false side makes the AND false whatever the other is.</i>
TRUE AND UNKNOWN	UNKNOWN	Kết quả còn phụ thuộc vế chưa biết. <i>The result still depends on the unknown side.</i>

LỖI HAY GẶP / COMMON MISTAKE – Viết điều kiện = NULL hoặc != NULL để tìm hay loại ô trống. Cả hai không bao giờ giữ dòng nào; phải dùng IS NULL và IS NOT NULL.
Writing = NULL or != NULL to find or exclude empty cells. Neither ever keeps a row; use IS NULL and IS NOT NULL.

LỖI HAY GẶP / COMMON MISTAKE – Coi NULL bằng 0 hoặc bằng chuỗi rỗng. Số dư NULL nghĩa là chưa biết số dư, khác hẳn số dư 0 đồng.

Treating NULL as 0 or as an empty string. A NULL balance means the balance is unknown, which is quite different from a balance of 0 dong.

PHẢI THUỘC / MEMORISE – Quan hệ: không trùng bộ, thứ tự không quan trọng, giá trị nguyên tố, mỗi cột một miền. NULL = không có giá trị; so sánh với NULL cho UNKNOWN, dùng IS NULL để tìm.

A relation has no duplicate tuples, no significant order, atomic values and one domain per column. NULL = no value; comparing with NULL gives UNKNOWN, so use IS NULL to find it.

2.7. Khóa – 2.7. Keys

Trong lớp có hai bạn cùng tên Minh. Gọi "Minh ơi" thì cả hai quay lại; gọi mã số sinh viên thì chỉ một người đáp. Khóa (key) đóng vai mã số đó trong một bảng: một thuộc tính, hoặc một nhóm thuộc tính, dùng để chỉ đúng một dòng. Có sáu loại khóa, từ rộng đến hẹp, ta đi từng loại.

Two classmates share the name Minh. Call "Minh" and both turn round; call a student number and only one answers. A key plays the part of that number in a table: an attribute, or a group of attributes, that picks out exactly one row. There are several kinds of key, from broad to narrow, and we take them one by one.

Siêu khóa, khóa ứng viên, khóa chính, khóa thay thế – *Superkey, candidate key, primary key, alternate key*

- Siêu khóa (superkey): tập thuộc tính mà hai bộ khác nhau không bao giờ trùng giá trị trên nó. Thêm thuộc tính thừa vào một siêu khóa vẫn được một siêu khóa.

Superkey: a set of attributes on which two different tuples never share the same values.

Adding extra attributes to a superkey still gives a superkey.

- Khóa ứng viên (candidate key), thường gọi gọn là khóa: siêu khóa nhỏ nhất, tức bỏ đi bất kỳ thuộc tính nào là không còn là siêu khóa. Một bảng có thể có nhiều khóa ứng viên.

Candidate key, often just called a key: a minimal superkey, meaning that removing any attribute stops it being a superkey. A table may have several candidate keys.

- Khóa chính (primary key): một khóa ứng viên được chọn làm đại diện. Mỗi bảng có đúng một khóa chính, không được NULL. Trong sách, khóa chính ghi in đậm hoặc kèm cụm "khóa chính".

Primary key: the candidate key chosen as the representative. Each table has exactly one primary key, and it may never be NULL. In this book the primary key is shown in bold or tagged "primary key".

- Khóa thay thế (alternate key): các khóa ứng viên còn lại sau khi đã chọn khóa chính. Trong SQL, chúng khai báo bằng UNIQUE. Một số sách dùng chữ "khóa thay thế" cho khóa nhân tạo; sách này tách hai khái niệm.

Alternate key: the candidate keys left after the primary key has been chosen. In SQL they are declared with UNIQUE. Some books use the Vietnamese term for the surrogate key; this book keeps the two ideas apart.

$$\forall t_1, t_2 \in r: t_1 \neq t_2 \Rightarrow t_1[K] \neq t_2[K] \quad (2.3)$$

Công thức (2.3) nói K là siêu khóa của r nếu hai bộ khác nhau bất kỳ t_1, t_2 đều khác nhau trên K . Ký hiệu $t[K]$ là phần giá trị của bộ t ở các thuộc tính K .

Formula (2.3) says K is a superkey of r when any two different tuples t_1 and t_2 differ on K . The notation $t[K]$ means the part of tuple t on the attributes K .

Khóa đơn, khóa phức hợp, khóa ngoại – *Simple key, composite key, foreign key*

Khóa chỉ gồm một thuộc tính gọi là khóa đơn; gồm từ hai thuộc tính trở lên gọi là khóa phức hợp (composite key). Bảng LienKet ghi ví nào liên kết với ngân hàng nào. Một ví liên kết nhiều ngân hàng, một ngân hàng có nhiều ví, nên khóa của nó là cặp (MaVi, MaNH).

A key of one attribute is a simple key; a key of two or more attributes is a composite key. Table LienKet records which wallet is linked to which bank. One wallet links to several banks and one bank has many wallets, so its key is the pair (MaVi, MaNH).

Khóa ngoại (foreign key) là thuộc tính của bảng này dùng để chỉ về khóa chính của một bảng khác, giống số chứng minh nhân dân ghi trên phiếu đăng ký chỉ về đúng một người trong sổ hộ tịch. Giá trị khóa ngoại phải có mặt ở bảng được chỉ tới, hoặc là NULL. Đó là ràng buộc tham chiếu, Chương 4 học kỹ.

A foreign key is an attribute of one table that points at the primary key of another table, like an ID number on a registration slip pointing at one person in the civil register. A foreign key value must exist in the table pointed at, or be NULL. That is the referential constraint, studied closely in Chapter 4.

$$\forall t \in r_1: t[\text{FK}] = \text{NULL} \vee \exists s \in r_2: s[\text{PK}] = t[\text{FK}] \quad (2.4)$$

Công thức (2.4) đọc là: với mọi bộ t của r_1 , khóa ngoại FK hoặc là NULL, hoặc bằng khóa chính PK của một bộ s nào đó trong r_2 . Ba ví dụ trong CSDL Trường Tiền:

Formula (2.4) reads: for every tuple t of r_1 , the foreign key FK is either NULL or equal to the primary key PK of some tuple s in r_2 . Three examples from the Trường Tiền database:

- Vi.MaKH chỉ về KháchHang.MaKH: mỗi ví thuộc về một khách có thật.
Vi.MaKH points at KháchHang.MaKH: every wallet belongs to a real customer.
- GiaoDich có hai khóa ngoại cùng chỉ về Vi: MaVi là ví thực hiện, MaViNhan là ví nhận (có thể NULL). Tên cột không cần trùng tên khóa chính mà nó chỉ tới.
GiaoDich has two foreign keys pointing at Vi: MaVi is the paying wallet, MaViNhan the receiving wallet (which may be NULL). The column name need not match the primary key it points at.
- NhanVien.MaQuanLy chỉ về NhanVien.MaNV: khóa ngoại chỉ về chính bảng của nó, vì cấp trên cũng là một nhân viên.
NhanVien.MaQuanLy points at NhanVien.MaNV: a foreign key pointing at its own table, since a superior is also an employee.

Khóa tự nhiên và khóa nhân tạo – *Natural keys and surrogate keys*

Khóa tự nhiên (natural key) lấy từ dữ liệu có sẵn ngoài đời, như số điện thoại hay số căn cước. Khóa nhân tạo (surrogate key) là mã do hệ thống tự cấp, không mang nghĩa nghiệp

vụ, như số thứ tự MaGD của giao dịch. Mã ví V0001 và mã khách KH001 nằm ở giữa: do công ty đặt ra nhưng có cấu trúc dễ đọc.

A natural key comes from data that already exists in the real world, such as a phone number or an identity number. A surrogate key is a code issued by the system with no business meaning, such as the running number MaGD of a transaction. The wallet code V0001 and customer code KH001 sit in between: issued by the company but with a readable pattern.

Tiêu chí <i>Criterion</i>	Khóa tự nhiên <i>Natural key</i>	Khóa nhân tạo <i>Surrogate key</i>
Nguồn <i>Source</i>	Dữ liệu có sẵn ngoài đời. <i>Data that already exists in the real world.</i>	Hệ thống tự cấp. <i>Issued by the system.</i>
Ổn định <i>Stability</i>	Có thể đổi, như khách đổi số điện thoại. <i>May change, as when a customer changes phone number.</i>	Không đổi suốt đời bản ghi. <i>Never changes during the record's life.</i>
Độ dài <i>Length</i>	Có thể dài hoặc phức hợp. <i>May be long or composite.</i>	Ngắn, thường một số nguyên. <i>Short, usually one integer.</i>
Nhược điểm <i>Drawback</i>	Đổi giá trị thì phải sửa mọi bảng tham chiếu. <i>A change forces edits in every table that refers to it.</i>	Không đọc ra nghĩa; vẫn cần UNIQUE cho khóa tự nhiên. <i>Reveals no meaning; a natural key still needs UNIQUE.</i>

Cách tìm khóa từ quy tắc nghiệp vụ – *Finding keys from business rules*

Khóa là tính chất của lược đồ, do quy tắc nghiệp vụ quyết định. Nhìn vài dòng dữ liệu chưa thấy trùng thì chưa kết luận được khóa, vì dòng của ngày mai có thể trùng. Hãy đi theo các bước:

A key is a property of the schema, decided by business rules. Seeing no duplicates in a few rows proves nothing, because tomorrow's rows may duplicate. Follow these steps:

- Viết ra các quy tắc nghiệp vụ dạng "điều gì không bao giờ trùng".
Write down the business rules in the form "what never repeats".
- Thử từng thuộc tính đơn: có thể có hai bộ chung giá trị ở thuộc tính này không? Nếu có, loại.
Test each single attribute: can two tuples share a value here? If so, rule it out.
- Thử các cặp thuộc tính, rồi bộ ba, theo cùng câu hỏi, cho tới khi gặp tập luôn phân biệt được mọi bộ: đó là siêu khóa.
Test pairs, then triples, with the same question until a set always tells every tuple apart: that is a superkey.
- Bỏ lần lượt từng thuộc tính của siêu khóa; nếu vẫn còn là siêu khóa thì thuộc tính ấy thừa. Khi không bỏ được nữa, ta có khóa ứng viên.
Drop each attribute of the superkey in turn; if it is still a superkey, that attribute was surplus. When nothing more can be dropped, you have a candidate key.

5. Chọn khóa chính: ngắn, không đổi, không NULL. Các khóa ứng viên còn lại là khóa thay thế.

Choose the primary key: short, unchanging, never NULL. The remaining candidate keys are alternate keys.

Ví dụ mẫu: KháchHang – *Worked example: KháchHang*

Bảng KháchHang(MaKH, HoTen, NgaySinh, GioiTinh, SoDienThoai, TinhThanh, NgayDangKy, Hang) có 8 thuộc tính. Quy tắc nghiệp vụ: mỗi khách có một mã riêng; mỗi số điện thoại chỉ đăng ký cho một khách; tên, ngày sinh, tỉnh, hạng thì nhiều người trùng nhau. Theo bước 2, chỉ MaKH và SoDienThoai luôn phân biệt được mọi khách. Mỗi cái là một khóa ứng viên, vì chỉ có một thuộc tính nên không còn gì để bỏ.

Table KháchHang(MaKH, HoTen, NgaySinh, GioiTinh, SoDienThoai, TinhThanh, NgayDangKy, Hang) has 8 attributes. Business rules: each customer has a unique code; each phone number is registered to one customer only; names, birth dates, provinces and ranks are shared by many people. By step 2, only MaKH and SoDienThoai always tell customers apart. Each is a candidate key, and having one attribute each leaves nothing to drop.

Chọn MaKH làm khóa chính vì mã nội bộ không đổi, còn số điện thoại có thể đổi. SoDienThoai là khóa thay thế. Đếm siêu khóa, giả sử chỉ có hai khóa ứng viên này: tập chứa MaKH có $2^7 = 128$ cách chọn 7 thuộc tính còn lại; tập chứa SoDienThoai cũng 128; tập chứa cả hai có $2^6 = 64$. Số siêu khóa là $128 + 128 - 64 = 192$, vì phải trừ phần đếm hai lần.

Choose MaKH as primary key because the internal code never changes whereas a phone number can. SoDienThoai is an alternate key. Count the superkeys, assuming only these two candidate keys: sets containing MaKH have $2^7 = 128$ ways to choose from the other 7 attributes; sets containing SoDienThoai also 128; sets containing both $2^6 = 64$. The number of superkeys is $128 + 128 - 64 = 192$, because the overlap was counted twice.

Giờ thử lý thuyết với dữ liệu. Ba truy vấn đầu tìm những giá trị lặp: số điện thoại, cặp (ví, thời điểm) của giao dịch, và số giao dịch của mỗi ví. Hai truy vấn cuối nói về bảng LienKet. GROUP BY gom các dòng cùng giá trị thành nhóm, HAVING COUNT(*) > 1 giữ nhóm có từ hai dòng trở lên, LIMIT n chỉ lấy n dòng đầu.

Now test the theory against the data. The first three queries look for repeated values: phone numbers, the (wallet, time) pair of a transaction, and the number of transactions per wallet. The last two concern table LienKet. GROUP BY gathers rows with the same value into groups, HAVING COUNT() > 1 keeps groups of two or more rows, and LIMIT n takes only the first n rows.*

```
SELECT SoDienThoai, COUNT(*) AS so_lan FROM KháchHang
GROUP BY SoDienThoai HAVING COUNT(*) > 1;
SELECT MaVi, ThoiDiem, COUNT(*) AS so_lan FROM GiaoDich
GROUP BY MaVi, ThoiDiem HAVING COUNT(*) > 1;
SELECT MaVi, COUNT(*) AS so_gd FROM GiaoDich
GROUP BY MaVi ORDER BY so_gd DESC, MaVi LIMIT 3;
SELECT MaVi, COUNT(*) AS so_nh FROM LienKet
GROUP BY MaVi ORDER BY so_nh DESC, MaVi LIMIT 2;
SELECT MaNH, COUNT(*) AS so_vl FROM LienKet
GROUP BY MaNH ORDER BY so_vl DESC, MaNH LIMIT 2;
```

Kết quả chạy / Output

```
sodienthoai | so_lan
-----+-----
(0 rows)

mavi | thoidiem | so_lan
-----+-----+-----
(0 rows)

mavi | so_gd
-----+-----
V0001 |      7
V0003 |      4
V0004 |      4
(3 rows)

mavi | so_nh
-----+-----
V0001 |      4
V0003 |      2
(2 rows)

manh | so_vi
-----+-----
NH01 |      4
NH02 |      4
(2 rows)
```

Đọc kết quả. Hai truy vấn đầu trả 0 dòng: dữ liệu hiện tại không có số điện thoại trùng, cũng không có cặp (ví, thời điểm) trùng. Với số điện thoại, quy tắc nghiệp vụ bảo đảm nó là khóa. Với cặp (MaVi, ThoiDiem) thì không: quy tắc không cấm một ví có hai giao dịch cùng giây, nên đó chỉ là may mắn của dữ liệu, không phải khóa. Ví V0001 xuất hiện 7 lần trong GiaoDich nên MaVi riêng lẻ không là khóa. Ở LienKet, V0001 nối 4 ngân hàng và ngân hàng NH01 nối 4 ví, nên cả MaVi lẫn MaNH đều không đủ một mình; phải ghép cặp.

Reading the output. The first two queries return 0 rows: the current data has no repeated phone number and no repeated (wallet, time) pair. For the phone number, the business rule guarantees it is a key. For the pair (MaVi, ThoiDiem) it does not: no rule forbids two transactions of one wallet in the same second, so the absence of duplicates is luck, not a key. Wallet V0001 appears 7 times in GiaoDich, so MaVi alone is not a key. In LienKet, V0001 links 4 banks and bank NH01 links 4 wallets, so neither MaVi nor MaNH is enough alone; they must be paired.

Khi nào bảng thật sự khai báo khóa? Truy vấn sau lấy từ danh mục hệ thống của PostgreSQL các ràng buộc của bảng GiaoDich. Cột contype ghi p là khóa chính, f là khóa ngoại, c là điều kiện kiểm tra CHECK.

How does a table actually declare its keys? The next query pulls the constraints of table GiaoDich from PostgreSQL's system catalogue. Column contype shows p for primary key, f for foreign key and c for a CHECK condition.

```
SELECT conname, contype FROM pg_constraint
WHERE conrelid = 'giaodich'::regclass
ORDER BY contype, conname;
```