

# OVERVIEW

## I. Introduction to Database

### 1.1 What is a database ?

A database is a structured collection of data stored electronically, usually managed by a Database Management System (DBMS). It allows users to store, organize, search, update, and retrieve information efficiently. Databases are widely used in applications such as banking, e-commerce, schools, and social media to manage large amounts of data.

[Cơ sở dữ liệu là một tập hợp dữ liệu có cấu trúc, được lưu trữ bằng phương tiện điện tử và thường được quản lý bởi hệ quản trị cơ sở dữ liệu (DBMS). Nó cho phép người dùng lưu trữ, tổ chức, tìm kiếm, cập nhật và truy xuất thông tin một cách hiệu quả. Cơ sở dữ liệu được sử dụng rộng rãi trong ngân hàng, thương mại điện tử, trường học, mạng xã hội để quản lý khối lượng dữ liệu lớn.]

### 1.2 Database Management Systems (DBMS) ?

A Database Management System (DBMS) is software that helps create, manage, and use databases. It allows users to insert, update, delete, and query data while ensuring security, consistency, and backup. Examples include MySQL, Oracle, SQL Server, and PostgreSQL.

Hệ quản trị cơ sở dữ liệu (DBMS) là phần mềm giúp tao, quản lý và sử dụng cơ sở dữ liệu. Nó cho phép người dùng thêm, sửa, xóa, truy vấn dữ liệu, đồng thời đảm bảo bảo mật, tính nhất quán và sao lưu.



### 1.3 Relational Model ?

The Relational Model is a way to organize data in a database using tables (called relations). Each table has rows (tuples) representing records and columns (attributes) representing data fields. It uses keys (like primary key, foreign key) to connect tables and ensure data consistency.

Mô hình quan hệ là cách tổ chức dữ liệu trong cơ sở dữ liệu bằng các bảng (gọi là quan hệ). Mỗi bảng có hàng (bộ dữ liệu) để biểu diễn bản ghi và cột (thuộc tính) để biểu diễn trường dữ liệu. Mô hình này sử dụng khóa (như khóa chính, khóa ngoại) để liên kết các bảng và đảm bảo tính nhất quán dữ liệu.

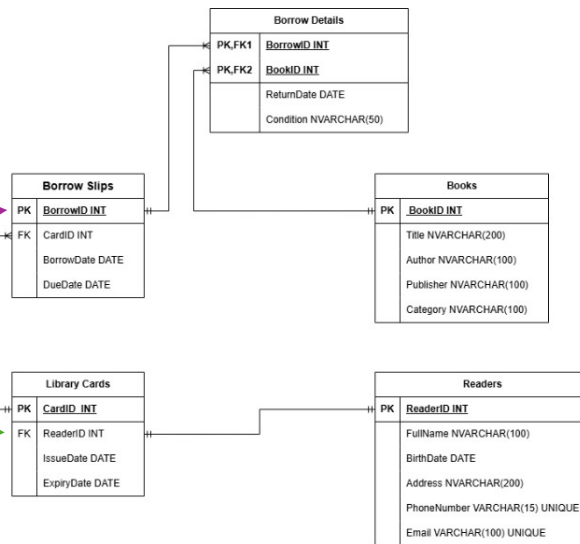
Columns (attributes)

Rows (tuples)

ReaderID	FullName	BirthDate
1	Alice Nguyen	2002-05-12
2	Carlos Diaz	2001-09-22
3	Daria Kowalska	2000-03-04
4	Ethan Smith	1999-11-30
5	Farah Hassan	2002-07-18
6	Giulia Rossi	2001-02-22

Primary key

Foreign key



## II. Relational Model Concepts

### 2.1 Tables (relations)

A table (relation) is the basic structure in the relational model where data is stored.

- A table is made up of rows , each row representing a single record.
- It has columns , each column representing a type of data (e.g., Name, Age).
- Each table usually has a primary key to uniquely identify records.

Một bảng gồm các hàng (bộ dữ liệu), mỗi hàng là một bản ghi.

Bảng có các cột (thuộc tính), mỗi cột biểu diễn một loại dữ liệu (VD: Tên, Tuổi).

Mỗi bảng thường có khóa chính để xác định duy nhất từng bản ghi.

### 2.2 Rows (tuples)

A row is a single record in a table. Each row contains values for all columns.

Example: (00000002, Phung Cao Thi Ngoc, 19920215) is one tuple in the table.

## 2.3 Columns (attributes)

A column is a field in a table that stores one type of data (e.g., Name, Date of Birth).

Cột (thuộc tính) là một trường trong bảng, lưu cùng một loại dữ liệu (VD: Họ tên, Ngày sinh).

## 2.4 Primary keys

A Primary Key is a column (or set of columns) that uniquely identifies each row.

Khóa chính là một cột (hoặc nhóm cột) dùng để phân biệt duy nhất mỗi bản ghi.

## 2.5 Foreign keys

A Foreign Key is a column that creates a link between two tables, referencing the Primary Key of another table.

Khóa ngoại là một cột dùng để liên kết hai bảng, tham chiếu đến khóa chính của bảng khác.

ReaderID is the **Primary Key**. Each reader has a unique ID.



ReaderID	FullName	BirthDate
1	Alice Nguyen	2002-05-12
2	Carlos Diaz	2001-09-22
3	Daria Kowalska	2000-03-04
4	Ethan Smith	1999-11-30
5	Farah Hassan	2002-07-18
6	Giulia Rossi	2001-02-22

No two rows can have the same value here. **FullName** stores the name of the reader.

Different rows may have different or even the same names. **BirthDate** stores the date of birth.

## III. Structured Query Language (SQL)

Structured Query Language (SQL) is a standard programming language used to communicate with and manage databases. It allows users to create structures, insert and update data, and retrieve information efficiently.

SQL (Ngôn ngữ truy vấn có cấu trúc) là một ngôn ngữ lập trình chuẩn dùng để giao tiếp và quản lý cơ sở dữ liệu. Nó cho phép người dùng tạo cấu trúc, chèn và cập nhật dữ liệu, cũng như truy xuất thông tin một cách hiệu quả.

### 3.1 DML (Data Manipulation Language)

DML is a part of SQL used to work with the data inside tables. It includes commands such as INSERT, UPDATE, DELETE, and SELECT. These commands help users add, modify, remove, or retrieve records.

DML là một phần của SQL dùng để làm việc với dữ liệu trong bảng. Nó bao gồm các lệnh như INSERT, UPDATE, DELETE và SELECT. Các lệnh này giúp người dùng thêm, sửa, xóa hoặc truy xuất bản ghi.

```
INSERT INTO Readers (FullName, BirthDate, Address, PhoneNumber, Email)
VALUES
(N'Alice Nguyen', '2002-05-12', N'Hanoi, Vietnam', '8405349123456789', 'alice.nguyen@uni.edu'),
(N'Carlos Diaz', '2001-09-22', N'Madrid, Spain', '3491234567890123', 'carlos.diaz@uni.edu'),
(N'Daria Kowalska', '2000-03-04', N'Krakow, Poland', '4812345678901234', 'daria.k@uni.edu'),
(N'Ethan Smith', '1999-11-30', N'New York, USA', '1123456789012345', 'ethan.smith@uni.edu'),
(N'Farah Hassan', '2002-07-18', N'Casablanca, Morocco', '2123456789012345', 'farah.hassan@uni.edu'),
(N'Giulia Rossi', '2001-02-22', N'Rome, Italy', '3901234567890123', 'giulia.rossi@uni.edu'),
(N'Hiro Tanaka', '2000-12-09', N'Tokyo, Japan', '8131234567890123', 'hiro.tanaka@uni.edu'),
(N'Isabella Santos', '2002-10-14', N'Sao Paulo, Brazil', '5511234567890123', 'isabella.santos@uni.edu'),
(N'Jakub Nowak', '1999-06-25', N'Warsaw, Poland', '4856789012345678', 'jakub.nowak@uni.edu'),
(N'Linh Tran', '2001-08-05', N'Ho Chi Minh City, Vietnam', '8481234567890123', 'linh.tran@uni.edu');
```

### 3.2 DDL (Data Definition Language)

DDL is used to *define and manage the structure* of the database. It includes commands like *CREATE*, *ALTER*, and *DROP*, which are used to create, change, or delete tables and other database objects.

DDL được dùng để *định nghĩa và quản lý cấu trúc* của cơ sở dữ liệu. Nó bao gồm các lệnh như *CREATE*, *ALTER* và *DROP*, được sử dụng để tạo, thay đổi hoặc xóa bảng và các đối tượng khác trong cơ sở dữ liệu.

```
CREATE TABLE Readers (
    ReaderID INT IDENTITY(1,1) PRIMARY KEY,
    FullName NVARCHAR(100) NOT NULL,
    BirthDate DATE NOT NULL,
    Address NVARCHAR(200),
    PhoneNumber VARCHAR(15) UNIQUE,
    Email VARCHAR(100) UNIQUE
);
```

### 3.3 Querying Data

Querying is the process of asking the database for specific information using the *SELECT* statement. Queries can include conditions (*WHERE*), sorting (*ORDER BY*), and grouping (GROUP BY) to filter and organize the results.

Truy vấn dữ liệu là quá trình yêu cầu cơ sở dữ liệu cung cấp thông tin cụ thể bằng cách sử dụng câu lệnh SELECT. Truy vấn có thể bao gồm điều kiện (WHERE), sắp xếp (ORDER BY) và nhóm (GROUP BY) để lọc và tổ chức kết quả.

```
SELECT ReaderID, FullName, Address, PhoneNumber, Email
FROM Readers
WHERE Address LIKE '%Poland%';
```

ReaderID	FullName	Address	PhoneNumber	Email
3	Daria Kowalska	Krakow, Poland	48123456789	daria.k@uni.edu
9	Jakub Nowak	Warsaw, Poland	48501234567	jakub.nowak@uni.edu

3.4 Joins

Joins are used to combine rows from two or more tables based on related columns. Common types include *INNER JOIN*, *LEFT JOIN*, *RIGHT JOIN*, and *FULL JOIN*. Joins help users retrieve meaningful data across multiple tables.

Join được dùng để kết hợp các hàng từ hai hoặc nhiều bảng dựa trên các cột có liên quan. Các loại join phổ biến gồm INNER JOIN, LEFT JOIN, RIGHT JOIN và FULL JOIN. Join giúp người dùng truy xuất dữ liệu có ý nghĩa từ nhiều bảng khác nhau.

```
SELECT r.FullName, b.Title, bs.DueDate
FROM Readers r
      JOIN LibraryCards lc ON r.ReaderID = lc.ReaderID
      JOIN BorrowSlips bs ON lc.CardID = bs.CardID
      JOIN BorrowDetails bd ON bs.BorrowID = bd.BorrowID
      JOIN Books b ON bd.BookID = b.BookID
WHERE bd.ReturnDate IS NULL
      AND bs.DueDate < CAST(GETDATE() AS DATE);
```

FullName	Title	DueDate
Alice Nguyen	Data Structures and Algorithms	2024-01-24
Carlos Diaz	Artificial Intelligence: A Modern Approach	2024-01-29
Daria Kowalska	Machine Learning Basics	2024-02-03
Daria Kowalska	Linear Algebra and Its Applications	2024-02-03
Farah Hassan	Computer Networks	2024-02-24
Farah Hassan	Statistics for Data Science	2024-02-24

Students who have borrowed books that have not been returned

IV. Database Design

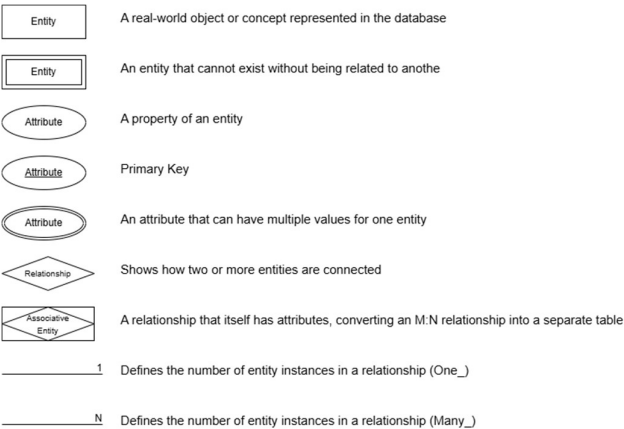
Database design is *the process of organizing data and defining its structure* to ensure efficiency, consistency, and accuracy. A good design makes databases easier to manage and reduces redundancy.

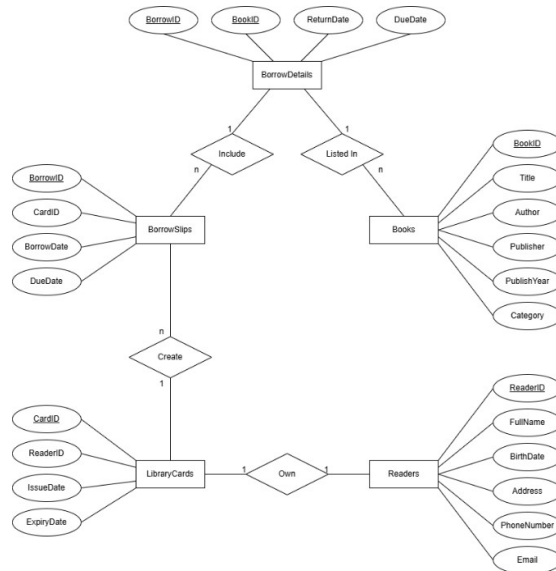
Thiết kế cơ sở dữ liệu là *quá trình tổ chức dữ liệu và xác định cấu trúc* của nó để đảm bảo hiệu quả, tính nhất quán và độ chính xác. Một thiết kế tốt giúp cơ sở dữ liệu dễ quản lý hơn và giảm sự dư thừa.

4.1 Entity-Relationship (ER) Modeling

ER modeling is a method used to visually *represent entities (objects)* and the *relationships* between them. It helps in planning how data should be structured before building the database. Common components are entities, attributes, and relationships.

Mô hình ER là phương pháp được dùng để *biểu diễn trực quan các thực thể (đối tượng)* và *mối quan hệ* giữa chúng. Nó giúp lập kế hoạch về cách dữ liệu nên được tổ chức trước khi xây dựng cơ sở dữ liệu. Các thành phần chính bao gồm thực thể, thuộc tính và mối quan hệ.





## 4.2 Normalization

Normalization is *the process of organizing data* into tables to *minimize redundancy* and *improve data* integrity. It divides large tables into smaller ones and defines relationships between them. Normal forms (1NF, 2NF, 3NF, etc.) guide this process.

Chuẩn hóa là quá trình tổ chức dữ liệu thành các bảng nhằm *giảm thiểu dư thừa* và *cải thiện tính toàn vẹn dữ liệu*. Nó chia các bảng lớn thành các bảng nhỏ hơn và xác định mối quan hệ giữa chúng. Các dạng chuẩn (1NF, 2NF, 3NF, ...) định hướng quá trình này.

## 4.3 Keys and constraints

Keys and constraints are *rules* that ensure *the accuracy and consistency* of data. Keys (such as primary keys and foreign keys) uniquely identify records and connect tables, while constraints (like NOT NULL, UNIQUE, CHECK) control what kind of data can be stored.

Khóa và ràng buộc là các *quy tắc đảm bảo tính chính xác và nhất quán của dữ liệu*. Các khóa (như khóa chính và khóa ngoại) dùng để xác định duy nhất bản ghi và liên kết các bảng, trong khi các ràng buộc (như NOT NULL, UNIQUE, CHECK) kiểm soát loại dữ liệu được lưu trữ.

```

ALTER TABLE Books
ADD CONSTRAINT CK_PublishYear CHECK (PublishYear >= 1900 AND PublishYear <= YEAR(GETDATE()))

ALTER TABLE LibraryCards
ADD CONSTRAINT CK_Card_Expiry CHECK (ExpiryDate IS NULL OR ExpiryDate >= IssueDate)

ALTER TABLE BorrowSlips
ADD CONSTRAINT CK_DueDate CHECK (DueDate >= BorrowDate)
  
```

## V. Logical Database Design

Logical database design focuses on *translating real-world requirements* into a *detailed blueprint* of how data will be stored, organized, and connected in the database. It ensures that the database structure supports applications and user needs effectively.

Thiết kế cơ sở dữ liệu logic tập trung vào việc *chuyển các yêu cầu thực tế thành một bản thiết kế chi tiết* về cách dữ liệu sẽ được lưu trữ, tổ chức và liên kết trong cơ sở dữ liệu. Nó đảm bảo cấu trúc cơ sở dữ liệu hỗ trợ hiệu quả cho các ứng dụng và nhu cầu của người dùng.

### 5.1 Logical Data Model

A logical data model *represents the structure of the data, including entities, attributes, and relationships, without considering the physical implementation*. It is a step between the conceptual model (*ER model*) and the physical database design.

Mô hình dữ liệu logic *biểu diễn cấu trúc dữ liệu, bao gồm các thực thể, thuộc tính và mối quan hệ, mà không xét đến việc triển khai vật lý*. Đây là bước trung gian giữa mô hình khái niệm (*mô hình ER*) và thiết kế cơ sở dữ liệu vật lý.

### 5.2 Tables and Relationship

In logical design, data is *organized into tables*, and relationships are defined using *primary keys and foreign keys*. This step ensures that entities from the conceptual model are properly connected in the database.

Trong thiết kế logic, dữ liệu được *tổ chức thành các bảng*, và mối quan hệ được xác định bằng *khóa chính và khóa ngoại*. Bước này đảm bảo các thực thể từ mô hình khái niệm được liên kết đúng trong cơ sở dữ liệu.

### 5.3 Attributes and Data Types

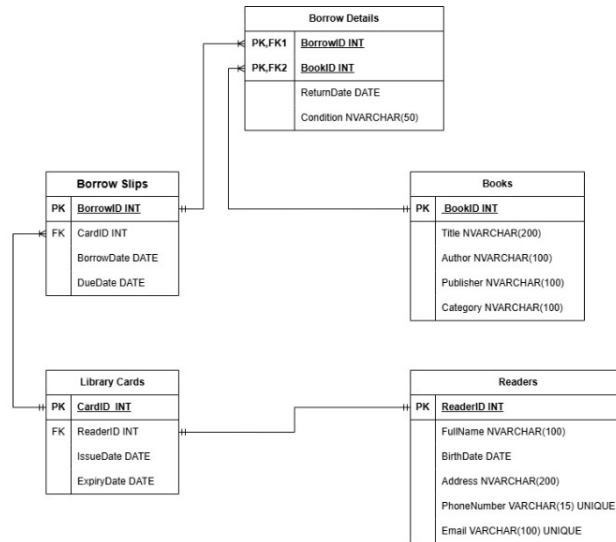
Attributes represent *the properties of entities*, and *each attribute is assigned a data type* (e.g., INTEGER, VARCHAR, DATE) that specifies what kind of values it can hold. Choosing the right data type improves efficiency and accuracy.

Thuộc tính biểu diễn các *đặc điểm của thực thể*, và *mỗi thuộc tính được gán một kiểu dữ liệu* (ví dụ: INTEGER, VARCHAR, DATE) để xác định loại giá trị có thể chứa. Việc chọn đúng kiểu dữ liệu giúp nâng cao hiệu quả và độ chính xác.

### 5.4 Normalization

Normalization in *logical design* ensures that data is stored in *the most efficient way* by eliminating redundancy and dependency issues. It follows formal rules like 1NF, 2NF, 3NF, and BCNF.

Chuẩn hóa trong *thiết kế logic* đảm bảo dữ liệu được *lưu trữ hiệu quả nhất* bằng cách loại bỏ sự dư thừa và các vấn đề phụ thuộc. Nó tuân theo các quy tắc chính thức như 1NF, 2NF, 3NF và BCNF.



## 5.5 Indexes

Indexes are special database objects that speed up data retrieval by creating quick access paths to rows in a table. Common types are single-column indexes and composite indexes.

Chỉ mục là các đối tượng đặc biệt trong cơ sở dữ liệu giúp tăng tốc truy xuất dữ liệu bằng cách tạo đường dẫn truy cập nhanh đến các hàng trong bảng. Các loại phổ biến là chỉ mục một cột và chỉ mục tổng hợp.

## VI. Data Integrity and Constraints

Data integrity ensures the *accuracy, consistency, and reliability* of data in a database. Constraints are rules applied to tables and columns to enforce data integrity and prevent invalid or inconsistent data.

Toàn vẹn dữ liệu đảm bảo tính *chính xác, nhất quán và độ tin cậy* của dữ liệu trong cơ sở dữ liệu. Ràng buộc là các quy tắc được áp dụng cho bảng và cột để duy trì toàn vẹn dữ liệu và ngăn chặn dữ liệu không hợp lệ hoặc không nhất quán.

### 6.1 Referential Integrity

Referential integrity is a rule that ensures *relationships between tables remain consistent*. For example, a foreign key in one table must match a primary key in another table, or be NULL. This prevents orphan records.

Toàn vẹn tham chiếu là một quy tắc *đảm bảo mối quan hệ giữa các bảng luôn nhất quán*. Ví dụ, một khóa ngoại trong bảng này phải khớp với một khóa chính trong bảng khác, hoặc có thể là NULL. Điều này ngăn chặn việc xuất hiện các bản ghi “mò côi”.



## 6.2 Constraints

Constraints are rules that define what kind of data can be stored in a table. Common constraints include:

NOT NULL: ensures a column cannot have empty values.

UNIQUE: ensures all values in a column are different.

PRIMARY KEY: uniquely identifies each record.

FOREIGN KEY: ensures consistency between related tables.

CHECK: ensures values meet a specific condition.

Ràng buộc là các quy tắc xác định loại dữ liệu có thể được lưu trong bảng

NOT NULL: đảm bảo một cột không được để trống.

UNIQUE: đảm bảo tất cả giá trị trong một cột đều khác nhau.

PRIMARY KEY: xác định duy nhất mỗi bản ghi.

FOREIGN KEY: đảm bảo tính nhất quán giữa các bảng liên quan.

CHECK: đảm bảo giá trị thỏa mãn một điều kiện nhất định.

```
ALTER TABLE Books
ADD CONSTRAINT CK_PublishYear CHECK (PublishYear >= 1900 AND PublishYear <= YEAR(GETDATE()))

ALTER TABLE LibraryCards
ADD CONSTRAINT CK_Card_Expiry CHECK (ExpiryDate IS NULL OR ExpiryDate >= IssueDate)

ALTER TABLE BorrowSlips
ADD CONSTRAINT CK_DueDate CHECK (DueDate >= BorrowDate)
```

## VII. Indexes and Performance Optimization

### 7.1 Indexes

An index is a special database object that provides *quick access* to rows in a table, similar to an *index in a book*. While indexes improve query performance, they also consume storage and may slow down data insertion or updates.

Chỉ mục là một đối tượng đặc biệt trong cơ sở dữ liệu cung cấp khả năng *truy cập nhanh* đến các hàng trong bảng, giống như *mục lục trong một cuốn sách*. Mặc dù chỉ mục giúp cải thiện hiệu suất truy vấn, nhưng chúng cũng tiêu tốn bộ nhớ và có thể làm chậm việc thêm hoặc cập nhật dữ liệu.

```
CREATE INDEX IX_BorrowSlips_Card ON BorrowSlips(CardID);
CREATE INDEX IX_BorrowDetails_Book ON BorrowDetails(BookID);
CREATE INDEX IX_Books_Category ON Books(Category);
```

## 7.2 Query Optimization

Query optimization is the process of improving the efficiency of SQL queries so they run faster and use fewer resources. Techniques include using indexes, rewriting queries, avoiding unnecessary columns, and analyzing query execution plans.

Tối ưu hóa truy vấn là quá trình cải thiện hiệu suất của các truy vấn SQL để chúng chạy nhanh hơn và sử dụng ít tài nguyên hơn. Các kỹ thuật bao gồm sử dụng chỉ mục, viết lại truy vấn, tránh lấy cột không cần thiết và phân tích kế hoạch thực thi truy vấn.

## VIII. Transactions and ACID Properties

### 8.1 Transactions

A transaction groups multiple operations (*such as insert, update, delete*) into one logical unit. Either all operations succeed, or none of them take effect.

Một giao dịch gom nhiều thao tác (*như thêm, cập nhật, xóa*) thành một đơn vị logic. Hoặc tất cả thao tác đều thành công, hoặc không thao tác nào có hiệu lực.

### 8.2 ACID Properties

ACID stands for Atomicity, Consistency, Isolation, and Durability. These properties guarantee reliable processing of transactions:

**Atomicity:** all operations in a transaction are completed, or none are applied.

**Consistency:** a transaction brings the database from one valid state to another.

**Isolation:** concurrent transactions do not interfere with each other.

**Durability:** once a transaction is committed, the changes remain even after a system crash.

ACID gồm Tính nguyên tử, Tính nhất quán, Tính độc lập và Tính bền vững. Những tính chất này đảm bảo việc xử lý giao dịch đáng tin cậy:

Nguyên tử (Atomicity): tất cả thao tác trong giao dịch đều hoàn tất, hoặc không thao tác nào được áp dụng.

Nhất quán (Consistency): giao dịch đưa cơ sở dữ liệu từ một trạng thái hợp lệ sang trạng thái hợp lệ khác.

Độc lập (Isolation): các giao dịch đồng thời không can thiệp lẫn nhau.

Bền vững (Durability): một khi giao dịch được xác nhận, các thay đổi sẽ tồn tại ngay cả khi hệ thống gặp sự cố.

### 8.3 Transactions Control

Transaction control statements are commands used to manage transactions:

**COMMIT:** saves all changes made by the transaction.

**ROLLBACK:** undoes changes if there is an error.

**SAVEPOINT:** sets a point within a transaction to roll back partially

Các lệnh điều khiển giao dịch được dùng để quản lý giao dịch

COMMIT: lưu tất cả thay đổi do giao dịch thực hiện.

ROLLBACK: hoàn tác các thay đổi nếu có lỗi.

SAVEPOINT: đặt một điểm trong giao dịch để có thể hoàn tác một phần.

# IX. Normalization

## 9.1 First Normal Form (1NF)

A table is in 1NF if all its attributes contain only atomic (indivisible) values and each record is unique.

Một bảng đạt chuẩn 1NF nếu tất cả các thuộc tính chỉ chứa các giá trị nguyên tử (không thể chia nhỏ) và mỗi bản ghi là duy nhất.

## 9.2 Second Normal Form (2NF)

A table is in 2NF if it is in 1NF and all non-key attributes depend entirely on the primary key, not just part of it.

Một bảng đạt chuẩn 2NF nếu nó ở 1NF và tất cả các thuộc tính không phải khóa đều phụ thuộc hoàn toàn vào khóa chính, chứ không chỉ một phần của khóa.

## 9.3 Thirs Normal Form (3NF)

A table is in 3NF if it is in 2NF and all its attributes depend only on the primary key, not on other non-key attributes (no transitive dependency).

Một bảng đạt chuẩn 3NF nếu nó ở 2NF và tất cả các thuộc tính chỉ phụ thuộc vào khóa chính, chứ không phụ thuộc vào các thuộc tính không phải khóa khác (không có phụ thuộc bắc cầu).

Raw data

BorrowID	ReaderName	PhoneNumbers	BookTitle	Author	BorrowDate	DueDate
1	Alice Nguyen	{+84812345678, +8487654321}	Introduction to Databases	Elmasri	2024-01-10	2024-01-24
1	Alice Nguyen	{+84812345678, +8487654321}	Data Structures and Algorithms	Lafore	2024-01-10	2024-01-24
2	Carlos Diaz	{+34912345678}	Artificial Intelligence: A Modern Approach	Russell	2024-01-15	2024-01-29

After using 1NF form

BorrowID	ReaderName	PhoneNumber	BookTitle	Author	BorrowDate	DueDate
1	Alice Nguyen	+84812345678	Introduction to Databases	Elmasri	2024-01-10	2024-01-24
1	Alice Nguyen	+8487654321	Data Structures and Algorithms	Lafore	2024-01-10	2024-01-24
2	Carlos Diaz	+34912345678	Artificial Intelligence...	Russell	2024-01-15	2024-01-29

After using 2NF form

**BorrowSlips**

BorrowID	ReaderID	BorrowDate	DueDate
1	1	2024-01-10	2024-01-24
2	2	2024-01-15	2024-01-29

**BorrowDetails**

BorrowID	BookID	ReturnDate	Condition
1	1	NULL	On Loan
1	2	NULL	On Loan
2	3	NULL	On Loan

After using 3NF form

**Departments**

DepartmentID	DepartmentName
CS	Computer Science
AI	Artificial Intelligence

**Readers**

ReaderID	FullName	DepartmentID
1	Alice Nguyen	CS
2	Carlos Diaz	AI

## X. Relational Algebra and Operations

### 10.1 Basic Operations

Basic relational algebra operations include:

Selection ( $\sigma$ ): retrieves rows that meet a condition.

Projection ( $\pi$ ): selects specific columns.

Union ( $\cup$ ): combines rows from two tables without duplicates.

Set Difference ( $-$ ): finds rows in one table but not in another.

Cartesian Product ( $\times$ ): combines all rows from two tables.

Join: combines rows based on related columns.

Các phép toán cơ bản của đại số quan hệ gồm:

Chọn ( $\sigma$ ): lấy các hàng thỏa mãn một điều kiện.

Chiếu ( $\pi$ ): chọn các cột cụ thể.

Hợp ( $\cup$ ): gộp các hàng từ hai bảng, không trùng lặp.

Hiệu ( $-$ ): tìm các hàng có trong một bảng nhưng không có trong bảng khác.

Tích Descartes ( $\times$ ): kết hợp tất cả hàng từ hai bảng.

Kết (Join): kết hợp các hàng dựa trên các cột liên quan.