What is a Database? A database is a collection of related data that: Reflects some aspect of the real world, Is logically coherent and meaningful, Is designed, built, and populated for a specific purpose

What is a DBMS? A DBMS (Database Management System) is a collection of programs that enables users to create and maintain a database, Define data types, structures, and constraints of the data, Store data on some medium under control of the DBMS, Query and update the database, Ensure security Protects data against malfunction and unauthorized access

Why do we use a DBMS instead of just writing code to store data? Efficient data storage & retrieval, Concurrency control, Data security, Automatic backup & recovery

Database vs. DBMS: A database is just a container of data, while a DBMS is the brain that organizes and manages it.

Many Kinds of DBMSs: Graph databases, Spatial databases, Document stores, Key-Value stores

What is a Relational Database? Based on the relational model (tables/relations). Developed over 45+ years, widely used. Supports various applications. Structure: Tables: Rows (tuples), Columns (attributes). Constraints: Ensure data integrity (e.g., primary keys, foreign keys). Referential Integrity: Links between tables enforce consistency.

What is a Primary Key? A Primary Key (PK) is a unique identifier for each record in a table.

What is a Foreign Key? A Foreign Key (FK) is a column (or set of columns) that links two tables together by referring to a Primary Key (PK) in another table.

What is Referential Integrity? Referential Integrity ensures that a Foreign Key (FK) must always reference a valid Primary Key (PK) in another table.

Feature of Relational DBMS: Data Independence, Declarative Languages, Reliable Transactions (ACID), High Performance, Authentication and authorization, Backup and recovery

Database Design Process: Requirements Collection & Analysis, Conceptual Database Design, Logical Design, Physical Design, System Implementation & Tuning

Components of a Relational Database: Set of Relations, Tables that store the data. Integrity Constraints, Rules that ensure data consistency (PK, FK). A database is in a valid state if it satisfies all integrity constraints.

Components of a Relation: Schema, Describes the structure of a relation (table name, attributes, domains) Example: STUDENT (Name, SSN, Phone, Dorm, Age, GPA). State, the current data stored in a relation (rows and columns).

Value Structure in Tuples: Follow First Normal Form (1NF) to ensure atomic values (No Composite, or Multi-valued)

Schema-Based Constraints: Domain Constraint, Key Constraint, Entity Integrity, Referential Integrity

Data Operations and Constraint Violations: Data operations include: 1. Insert: Adding new rows, which may violate(可能违反四个 Schema-Based Constraints). 2. Delete: If a deleted tuple is referenced by a foreign key, options include: Reject deletion, Cascade deletion, Set default or NULL. Update: Ensure updated values adhere to constraints

Relational Algebra Basics: RA is the formal basis for SQL, allowing users to retrieve data using a set of well-defined operations. (主要特征) Results in a new relation that can be queried further. Forms the foundation for query optimization.

Set Operations: UNION: Combine two result sets (no duplicates). UNION ALL (include duplicates). INTERSECT: Common rows in both sets. EXCEPT: Rows in one set but not the other.

Where: =, <> ‘U2’, < or >, <= or >=, LIKE ‘%Example%’, IN ‘Canada’, IS NULL, IS NOT NULL, BETWEEN 1 and 2

What does ORDER BY do in SQL? Used to sort the query results based on 1 or more cols. ASC A-Z, DESC Z-A

What is a SQL JOIN? A JOIN in SQL is used to combine data from two or more tables based on a related column (usually a Primary Key & Foreign Key relationship). INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN

INNER JOIN vs. LEFT JOIN: INNER JOIN, Returns only matching rows from both tables. LEFT JOIN Returns ALL rows from the “left” table, even if there’s no match. Inner 只返回匹配的，Left 返回左，没有匹配则 Null

Aggregate Functions: MAX, MIN, SUM, AVG, COUNT

What does GROUP BY do in SQL? GROUP BY is used to group rows with the same values in a specific column and apply aggregate functions (like COUNT(), SUM(), AVG(), etc.) to each group.

What is the difference between DISTINCT and GROUP BY in SQL? DISTINCT → Removes duplicate rows GROUP BY → Groups rows and allows aggregate functions

What is the difference between WHERE and HAVING in SQL? WHERE → Filters rows before grouping; HAVING → Filters groups after aggregation

Subqueries: A subquery is a query nested inside another query, Common uses: In SELECT, FROM, or WHERE clauses Subqueries can return single or multiple values

What is DML? Data Manipulation Language is used to 增删改查 data. DML Commands: **SELECT** → Retrieves rows.

Inserting Rows: INSERT INTO STUDENT (Name, SSN, GPA) VALUES ('Alice', '123-45-6789', 3.8);

Updating Rows: UPDATE STUDENT SET GPA = 4.0 WHERE Name = 'Alice';

Deleting Rows: DELETE FROM STUDENT WHERE GPA < 2.0;

What is DDL? Data Definition Language is used to define and modify database structures. DDL Commands, **Schema**: CREATE TABLE → Creates a new table. ALTER TABLE → Modifies an existing table. DROP TABLE → Deletes a table. **Transaction**: BEGIN: Start a transaction. COMMIT: Save all changes. ROLLBACK: Undo all changes.

Authorization: Manage user permissions with GRANT and REVOKE.

Issues with Accessing SQL via API: Impedance mismatch, DBMS abstraction layer, Cursors, Injection attacks

Prepared Query Sequence: Generate parameterized SQL, Bind values to SQL parameters, Execute, Get Results

Authentication 认证: Verifying who you are (密码生物识别令牌). 长密码，限制登录次数，2Factor Authentication

Authorization 授权: Determining what actions you can perform (e.g., access levels).

常见攻击及防御措: SQL Injection: Exploit improperly sanitized input to manipulate SQL queries. Prevention: Use parameterized queries or prepared statements. Sanitize user input and avoid dynamic SQL when possible. Denial of Service (DoS): Overload resources. Prevention: Set resource limits. Monitor traffic and implement patching policies. Brute-Force Attacks: Prevention: Limit login attempts. Use CAPTCHA or lockouts after multiple failed attempts. Internal Threats: Up to 80% of breaches originate internally. Regular audits, patching, and isolated database environments.

Encryption | 加密: Symmetric Encryption: One key for encryption and decryption. Asymmetric Encryption: Public key encrypts, private key decrypts. Hashing: One-way transformation with no decryption (e.g., SHA-256, bcrypt). Encrypt sensitive data (e.g., database files, backups, communications).

Password Security | 密码安全: ISSUE: Plaintext storage leads to vulnerabilities if compromised. Weak or common passwords are susceptible to dictionary attacks. Best Practices: Hash Passwords: Use salted hashes (e.g., bcrypt, PBKDF2). Add Salting: Unique random values for each user to prevent precomputed attacks. Honeywords: Store fake passwords to detect breaches.

A database is composed of? Tables (Relations) → Store structured data, Rows (Tuples) → Represent individual records, Columns (Attributes) → Define data fields, Schemas → Define the structure and constraints of data, Indexes → Speed up data retrieval, Constraints → Rules to maintain data integrity (e.g., Primary Key, Foreign Key)

A table schema is composed of? Table Name → The name of the table, Columns (Attributes) → Define the data fields, Data Types → Specify the type of data in each column (e.g., INTEGER, VARCHAR, DATE), Constraints → Rules that ensure data integrity, Default Values → Specifies a default value for a column, Indexes: Optimize the query performance

Each [schema component] has a ____ of valid _____ values? Domain, Data

What is the difference between a set vs. bag of tuples? Set of Tuples → No duplicate rows (tuples), Bag (Multiset) of Tuples → Allows duplicate rows **In what context does each apply?** Set-based model → Used in theoretical relational algebra. Ensures data integrity by preventing duplicates. Bag-based model → Used in SQL implementations. Bags are more practical in real-world databases since duplicates may occur naturally (e.g., in query results).

What kind of constraints that can be defined in the schema? Primary Key (PK): Uniquely identifies each row, Foreign Key (FK) : Links to another table, NOT NULL: Ensures a column cannot have NULL values, UNIQUE: Ensures all values in a column are different, CHECK: Ensures values meet specific conditions (e.g., Age > 0), Default

What is a superkey vs. a key? Superkey: A set of attributes that uniquely identifies a row (may have extra attributes). Candidate Key (Key): A minimal Superkey (no extra attributes). **How Do You Identify a Primary Key?** A primary key is a minimal superkey chosen to uniquely identify rows. It must be unique and not null. Chosen based on simplicity, stability, and business logic. **What happens to other superkeys?** Other superkeys remain valid but are not used as the primary key. They can enforce uniqueness with a unique constraint, if needed.

How do foreign keys fit in? A foreign key (FK) is a column or set of columns in one table that references the primary key (PK) of another table. It establishes a relationship between two tables and ensures referential integrity in the database.

What is a transaction? A sequence of database operations including retrieval and update(s).

What are the ACID guarantees? Atomicity: All or nothing. Consistency: Valid data remains valid. Isolation: Transactions don't interfere. Durability: Once committed, changes are permanent.

Examples for Each Constraint | 每种约束的示例:

Primary Key (主键): CREATE TABLE student (id INT PRIMARY KEY, name VARCHAR(50));

Foreign Key (外键): CREATE TABLE enrollment (student_id INT, course_id INT, FOREIGN KEY (student_id) REFERENCES student(id));

Unique (唯一约束): CREATE TABLE user (email VARCHAR(100) UNIQUE,username VARCHAR(50));

Not Null (非空约束): CREATE TABLE product (id INT PRIMARY KEY, name VARCHAR(100) NOT NULL);

Check (查约束): CREATE TABLE employee (id INT PRIMARY KEY, salary DECIMAL(10, 2), CHECK (salary > 0));

Default (默认值): CREATE TABLE orders (id INT PRIMARY KEY, status VARCHAR(20) DEFAULT 'pending');