

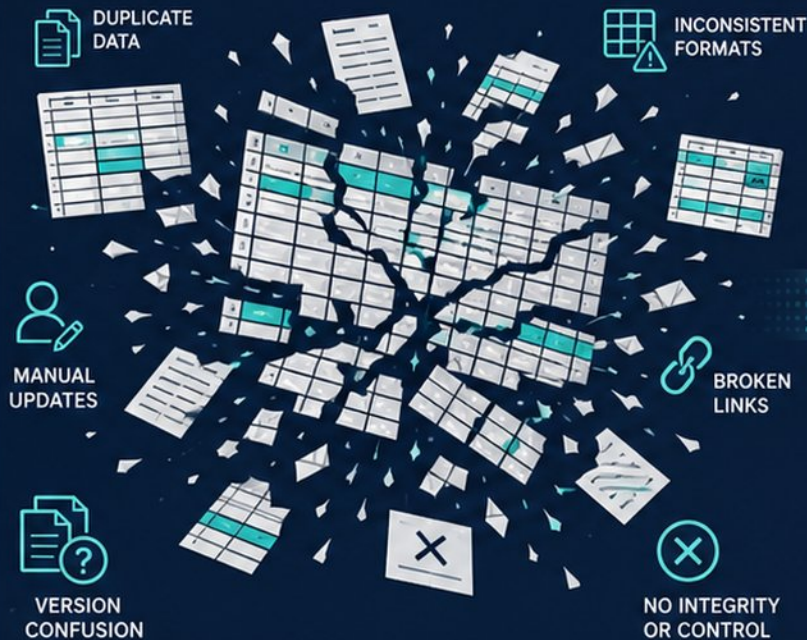
Introduction to Relational Databases

- A relational database organizes data into linked tables, remaining the dominant model for structured data in 2026.
- Powers e-commerce, banking, content management, and SaaS applications.
- Evolved from Edgar F. Codd's 1970 relational model to modern RDBMS.
- Relational databases hold roughly 60% of the cloud database market; MySQL, PostgreSQL, and Oracle lead adoption.
- This course builds from core table concepts to practical querying and real-world application.

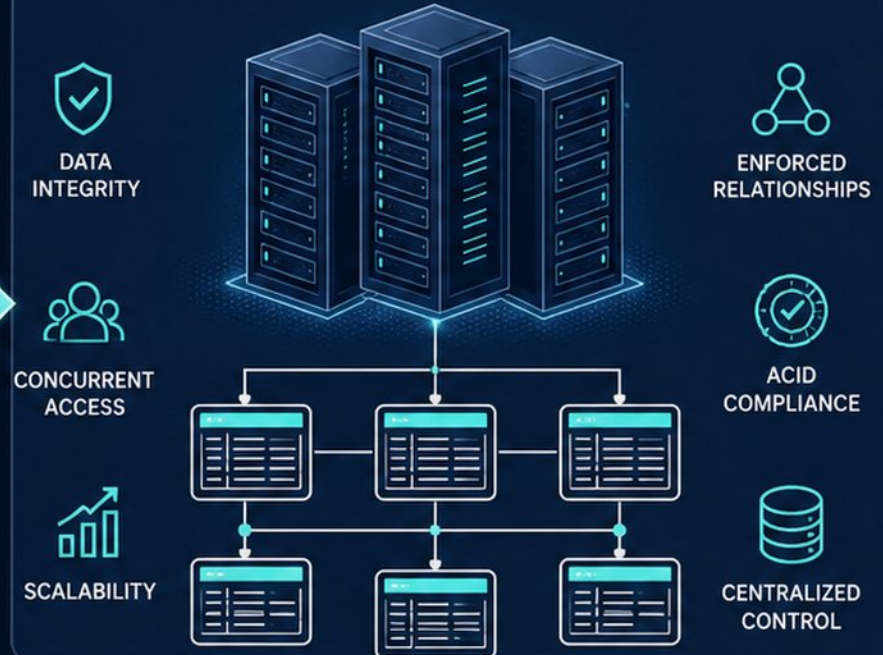


Why Not Just a Spreadsheet?

SPREADSHEET CHAOS FRAGILE. DUPLICATED. UNRELIABLE.



RELATIONAL SYSTEM ORDER STRUCTURED. INTEGRATED. RELIABLE.



- ✓ Duplication and inconsistency plague spreadsheet-based data.
- ✓ Flat files lack data integrity, concurrent access, and scalability.
- ✓ Relational systems enforce structure, ACID compliance, and relationships.
- ✓ Choose relational over NoSQL when data consistency and relationships matter.
- ✓ SQL is learnable and databases benefit projects of any size.

The Building Blocks: Tables, Rows, and Columns



- Tables represent real-world entities like Customers, Orders, Products



- Each row is a single, unique record or instance



- Columns define attributes with specific data types



- Every cell holds one atomic value, never a list



- Think of a spreadsheet grid with enforced rules

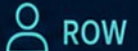
EXAMPLE TABLE: Customers

	CustomerID	Name	Email	SignupDate	Status
	INT	VARCHAR(100)	VARCHAR(255)	DATE	VARCHAR(20)
1	1001	Alice Johnson	alice@example.com	2024-01-15	Active
2	1002	Bob Smith	bob@example.com	2024-02-20	Active
3	1003	Carol Davis	carol@example.com	2024-03-05	Inactive
4	1004	David Lee	david@example.com	2024-03-18	Active
5	1005	Eva Brown	eva@example.com	2024-04-01	Active



TABLE

A collection of related records about one entity



ROW

A single record (instance) in the table



COLUMN

An attribute that defines a type of data



CELL

The intersection of a row and column; holds one atomic value

»»» Guaranteeing Uniqueness with Primary Keys



- Duplicate records cause chaos; primary keys enforce row uniqueness.



- A primary key is a column that uniquely identifies each row.



- Natural keys (email) exist; surrogate keys (auto-ID) are simpler.



- Choose keys that are unique, stable, and never change over time.



- Primary keys enable precise lookups and link tables together.



Connecting Tables: Relationships and Foreign Keys



- One-to-Many: a single Customer can place many Orders



- Foreign keys link rows across tables as cross-references



- Many-to-Many relationships require a junction table



- Splitting data across tables reduces redundancy and errors



- Referential integrity keeps all relationships valid



Entity-Relationship Thinking



- Translate business needs into entities and their relationships



- Read ERDs: boxes for entities, lines for relationships, crow's foot for multiplicity



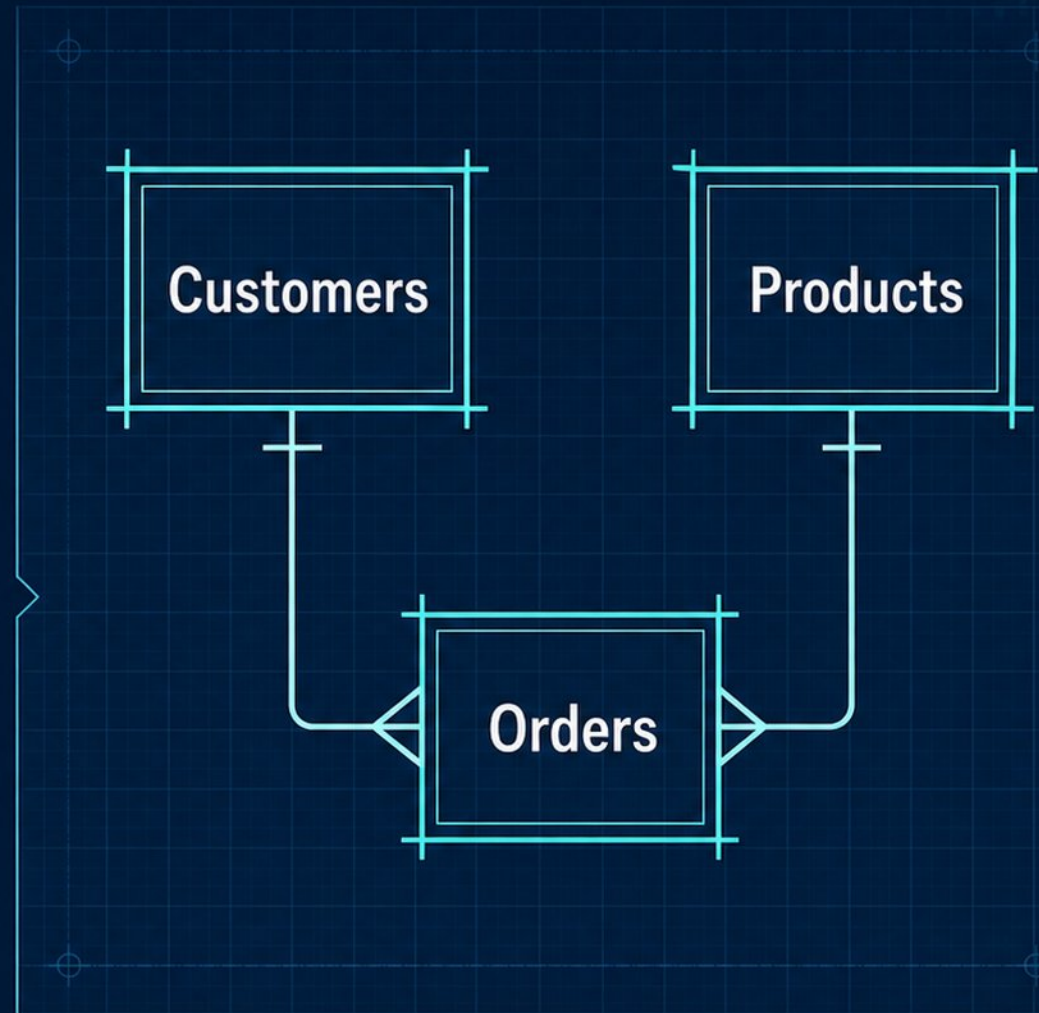
- Example: design an e-commerce schema (Customers, Orders, Products)



- Junction tables resolve many-to-many: Students-Courses, Actors-Movies



- Build the mental model first, then write SQL



The Language of Data: SQL Basics



“SQL is the universal language for relational databases, standardized since 1986”



“The four core operations: SELECT, INSERT, UPDATE, DELETE”



“Basic SELECT syntax: choose columns, specify the table”



“Filtering with WHERE: narrow rows based on conditions”



“Runnable examples: find customers by city, list recent orders”

SELECT...
FROM...
WHERE



Common Beginner Mistakes and How to Avoid Them



- Using **SELECT *** everywhere — fetch only the columns you need



- Forgetting the **WHERE** clause in **UPDATE** or **DELETE** — a dangerous oversight



- Using **=** instead of **IS NULL** to check for missing values



- Misusing **GROUP BY** — every non-aggregated column must appear in **GROUP BY**



- Ignoring query readability — formatting and aliases matter



- SQL execution order:
FROM → **WHERE** → **GROUP BY** →
HAVING → **SELECT** → **ORDER BY**



Your First JOIN: Combining Tables



- Data for a single report often lives in separate tables



- INNER JOIN returns only rows that match in both tables



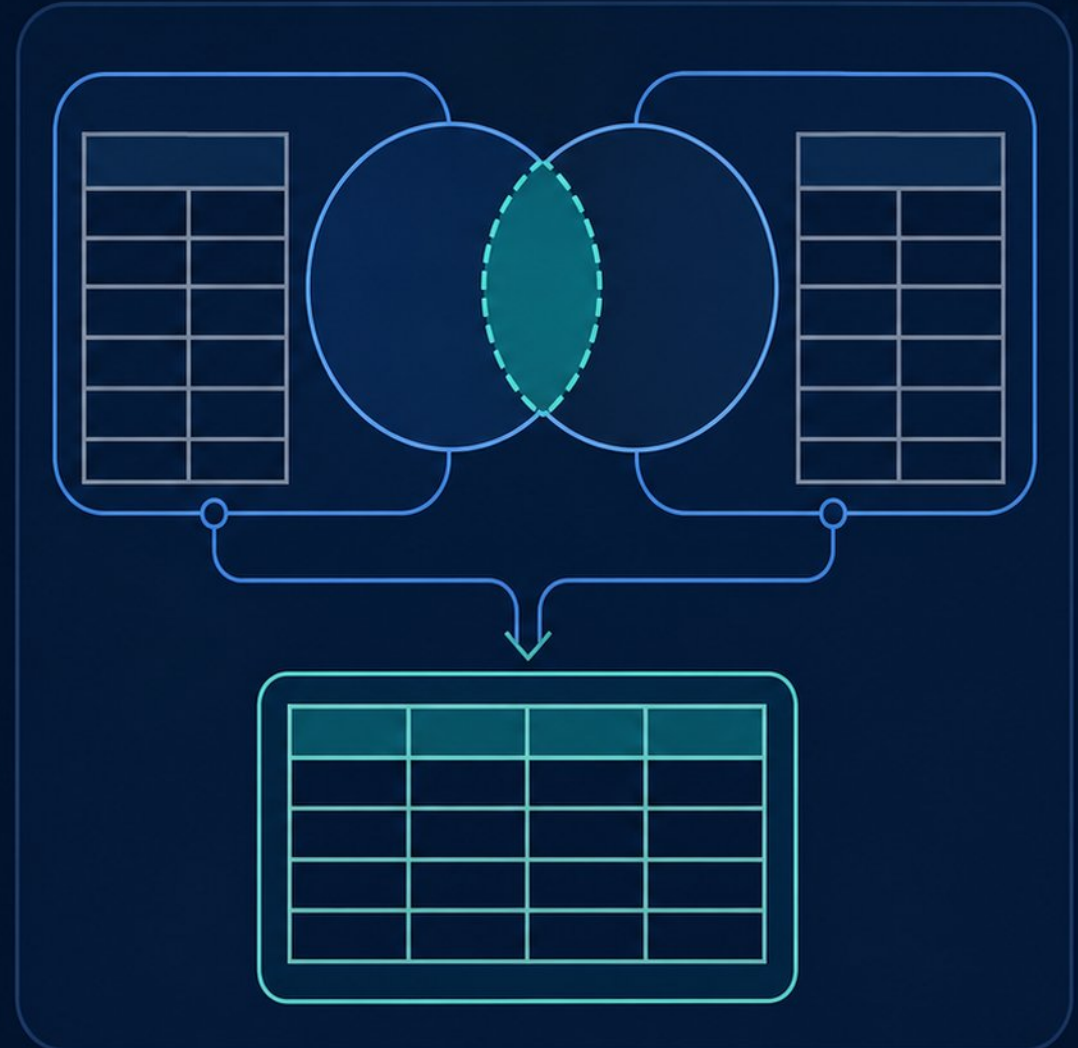
- Visualize JOINS as Venn diagram intersections



- Use table aliases and column prefixes for clarity



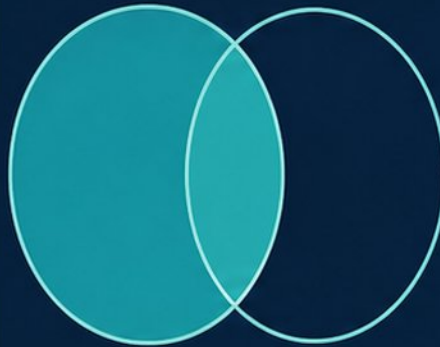
- Example: listing each order with its customer's name



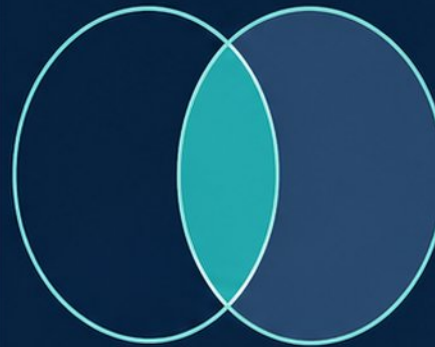
Beyond INNER JOIN: LEFT, RIGHT, and Self Joins

- LEFT JOIN keeps all left table rows, even without a match
- RIGHT JOIN is rarely used; flip table order instead
- SELF JOIN compares a table to itself, like employees-managers
- Always specify a JOIN condition to avoid Cartesian products
- Choose join type based on which side must retain all rows

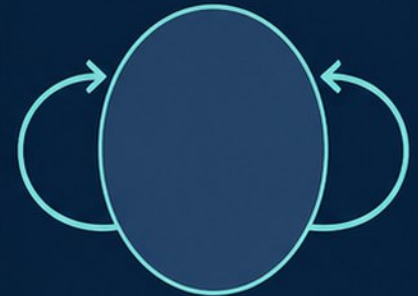
LEFT JOIN



RIGHT JOIN



SELF JOIN



Sorting, Grouping, and Aggregating Data



- **ORDER BY:** arrange results ascending or descending



- **GROUP BY:** collapse rows into summary groups



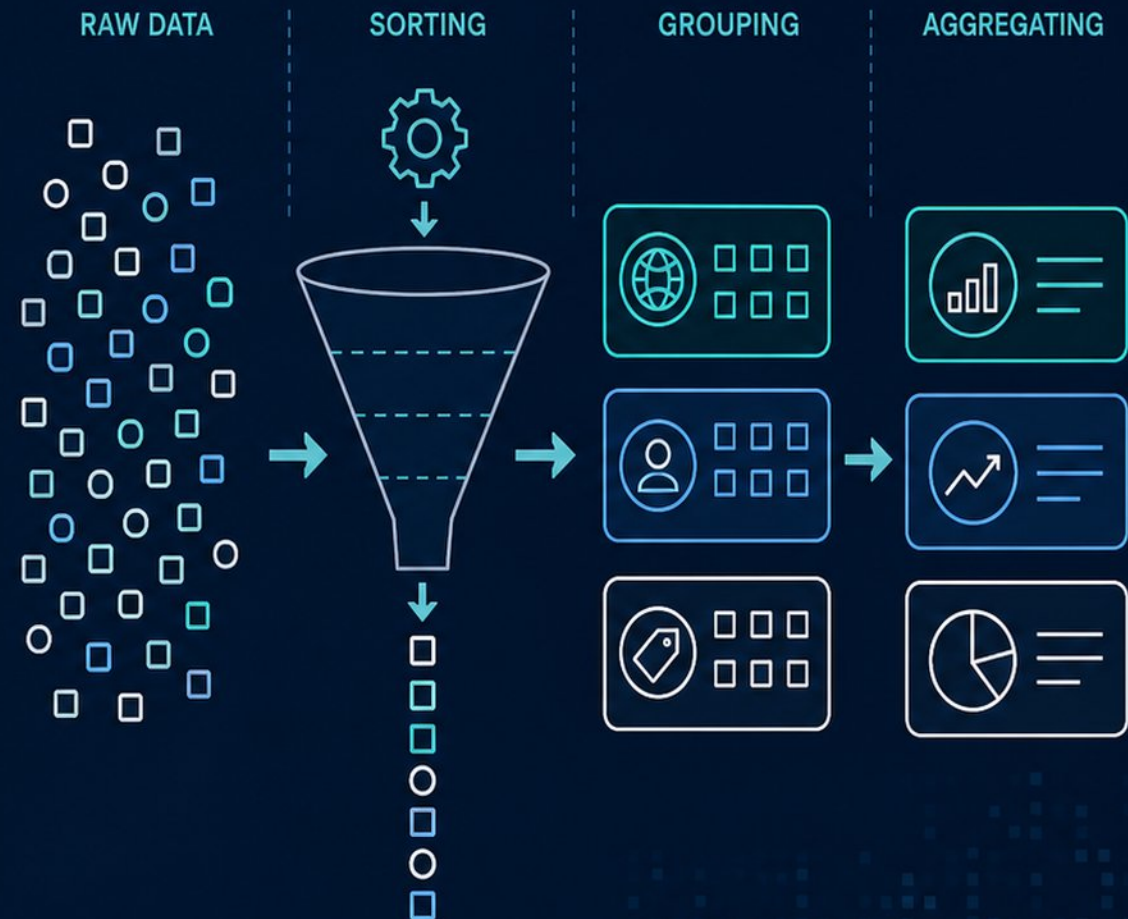
- **Aggregate functions:** COUNT, SUM, AVG, MAX, MIN



- **HAVING** filters groups; **WHERE** filters rows before grouping



- **Use cases:** total sales per region, average order value, customer order counts



Keeping Data Safe:

Normalization and Constraints



- Redundancy causes insert, update, and delete anomalies



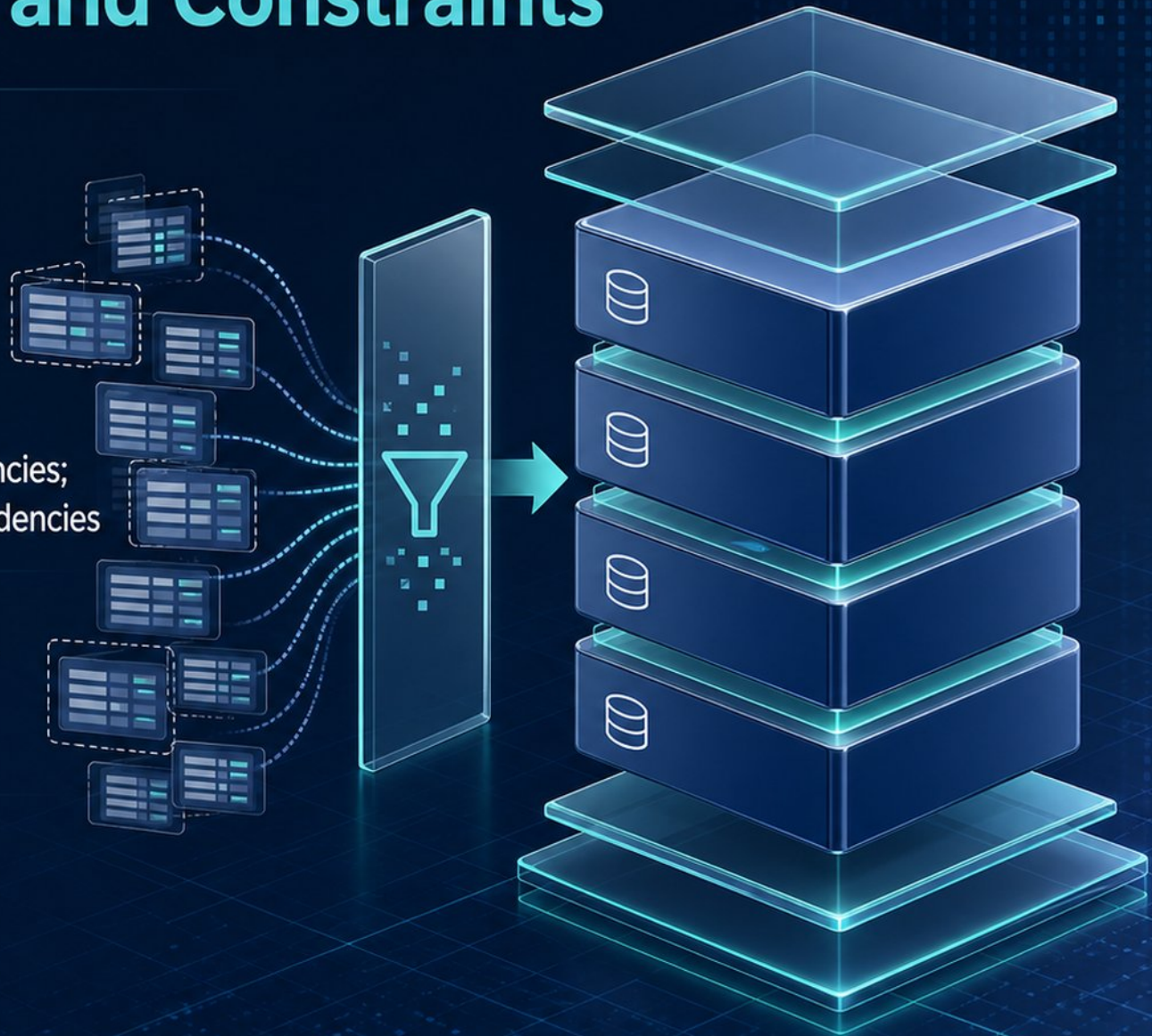
- 1NF: Atomic values;
2NF: No partial dependencies;
3NF: No transitive dependencies



- Constraints: NOT NULL, UNIQUE, CHECK, and DEFAULT guard data



- Normalize for integrity, denormalize selectively for read speed



The Database as an Application Backend



- Database sits behind the app server – stores, retrieves, and manages all persistent data
- SQL queries power reports, dashboards, and business intelligence tools
- Client-server model: applications send SQL, database returns results
- Top RDBMS in 2026: PostgreSQL, MySQL, SQLite, and cloud-managed services (AWS RDS, Azure SQL)
- PostgreSQL leads 55–65% of net-new open-source relational projects worldwide



Choosing the Right Database for Your Project

FIELD GUIDE



DB-Engines Jul 2026: Oracle, MySQL, SQL Server, PostgreSQL lead rankings



PostgreSQL now the default for new enterprise apps; MySQL dominates web/CMS



SQLite for embedded/mobile; cloud services: Amazon RDS, Azure SQL, Google Cloud SQL



Key factors: data structure, scale, team skills, budget, and compliance needs



Hands-On Practice: Your SQL Sandbox

- Free SQL playgrounds: SQL Fiddle, DB Fiddle, and FWD Tools' PostgreSQL sandbox
- Pre-loaded sample databases: employees, departments, e-commerce, and more
- Write queries and run them instantly—100% browser-based, no install needed
- Structured lessons guide you from basic SELECT to JOINS and GROUP BY
- No account required for most platforms; start experimenting immediately



Next Steps and Learning Resources



- **Recap journey:** tables, keys, SQL, JOINS, and normalization



- **Next topics:** intermediate SQL, database design, performance tuning



- **Key resources:** PostgreSQL/MySQL docs, 'Learning SQL' by Alan Beaulieu



- **Communities:** Stack Overflow, DB-Engines, Reddit's r/SQL



- **Mindset shift:** from learning SQL to thinking in sets and relationships



PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/fba1c414/public