Relational Database Design and Normalization

> The E-commerce schema was designed to meet the **Third Normal Form (3NF)** to ensure data integrity, minimize redundancy, and optimize transactional operations (OLTP).

First Normal Form (1NF)

Achieved by ensuring all columns hold **atomic values** (e.g., separating a customer's full address into StreetAddress, City, State). There are no repeating groups.

Second Normal Form (2NF)

Achieved by ensuring all non-key attributes in tables with **Composite Keys** depend on the entire key.

> **Example:** In the `OrderDetail` table (PK: `(OrderID, ProductID)`), the `UnitPrice` is dependent on the entire key (it's the price at the time of the order). However, attributes like `ProductName` were moved out to the `Product` table because they only depend on `ProductID` (a partial key).

Third Normal Form (3NF)

Achieved by eliminating **transitive dependencies** (where a non-key attribute depends on another non-key attribute).

> **Example:** In the `Product` table, storing `CategoryName` would be transitively dependent on `CategoryID`. This was corrected by creating the separate `Category` table, linked back by `CategoryID`.

📊 SQL Performance Analysis (Query Justification)

This section addresses the **Integration Report (PDF)** under *Performance Analysis of Complex Queries*.

Our goal was to use advanced features for efficiency and functionality:

SQL Feature Used	Justification and Performance
Window Functions (Q2)	Used to perform ranking (`RANK()`, `ROW_NUMBER()`) without requiring `PARTITION BY CategoryName` pass over the data.
Window Functions (Q2)	Used to perform ranking (`RANK()`, `ROW_NUMBER()`) without requiring

Window Functions (Q2)	Used to perform ranking (`RANK()` , `ROW_NUMBER()`) without requiring `PARTITION BY CategoryName` pass over the data.
Common Table Expressions (CTEs) (Q2)	CTEs (`WITH ProductSales AS (...)`) improve readability and maintainability of complex queries. They define temporary, reusable named result sets, making the multi-step ranking logic easier to debug and optimize.
Transactional Operations (Q4)	Essential for maintaining ACID properties (Atomicity, Consistency, Isolation, Durability). The `ROLLBACK` logic demonstrated guarantees that a multi-step business process (like inventory deduction) is treated as a single unit; if the stock constraint fails, no changes are applied, preventing corrupted inventory data.
Indexing (Implicit)	Primary Keys and Foreign Keys automatically create indexes, ensuring fast lookups and joins between `Product` and `OrderDetail` , and efficient data retrieval in the frequently queried `Category` table.

Key Benefits of This Design

- **Data Integrity:** Normalization ensures consistent and accurate data across all tables
- **Reduced Redundancy:** Eliminates duplicate data storage, saving space and preventing inconsistencies
- **Optimized Performance:** Strategic use of indexes and efficient query patterns improve response times
- **Maintainability:** Clear table structure and relationships make the database easier to maintain and extend
- **Scalability:** Normalized design supports growth in data volume and complexity