ERDs and Normalization

11 February 2025 14:30

Using ERDs

ERDs visually represent a database's structure, offering key advantages:

- **Identifies Issues Early**: Helps spot tables without relationships or illogical connections, preventing SQL design flaws.
- **Enhances Team Understanding**: Provides clarity on table fields and relationships.
- **Speeds Up SQL Writing**: Offers a condensed view, making multi-table queries easier.

Creating an ERD ensures a well-structured, efficient database design.

Components of ERDs

- **Entities** → Represent tables (e.g., Students, Orders). **Drawn as rectangles**.
- **Attributes** → Represent columns in a table. **Drawn as ellipses (ovals)**.
  - **Simple** (Name, Age)
  - **Composite** (Full Name → First + Last Name)
  - **Derived** (Age from DOB)
  - **Multi-valued** (Phone Numbers)
- **Relationships** → Define how entities are related. **Drawn as diamonds**.
  - Example: STUDENT **enrolls in** COURSE
- **Cardinality** → Defines how many instances relate to another.
  - **1:1** (One person ↔ One passport)
  - **1:M** (One customer → Many orders)
  - **M:N** (Students ↔ Courses)

NORMALIZATION :

Normalization is the process of organizing data in a database to reduce redundancy and improve integrity. It follows a series of steps called **normal forms (NF)**.

1NF (First Normal Form) - Eliminate Duplicates

Rule:

- Each column must have **atomic** (indivisible) values.
- Each row must be unique.

Example (Before 1NF - Repeating Groups):

OrderID	Customer	Products
1	Alice	Apple, Banana
2	Bob	Orange

After 1NF (Separate Rows for Each Product):

OrderID	Customer	Product
1	Alice	Apple
1	Alice	Banana
2	Bob	Orange

2NF (Second Normal Form) - Remove Partial Dependencies

Rule:

- Must be in **1NF**.
- Every **non-key** column must depend on the **whole** primary key, not just part of it.

Example (Before 2NF - Partial Dependency on OrderID):

OrderID	Product	ProductPrice	Customer
1	Apple	10	Alice
1	Banana	5	Alice

- Issue: `ProductPrice` depends only on `Product`, not `OrderID`.

After 2NF (Split into Separate Tables):

✅ Orders Table

OrderID	Customer
1	Alice

✅ Products Table

Product	ProductPrice
Apple	10
Banana	5

✅ OrderDetails Table

OrderID	Product
1	Apple
1	Banana

3NF (Third Normal Form) - Remove Transitive Dependencies
Rule:

- Must be in **2NF**.
- Non-key attributes should depend **only** on the primary key.

Example (Before 3NF - Transitive Dependency):

OrderID	Customer	CustomerAddress
1	Alice	NY, USA

- Issue: `CustomerAddress` depends on `Customer`, not `OrderID`.

After 3NF (Separate Customers Table):

✅ Orders Table

OrderID	CustomerID
1	101

✅ Customers Table

CustomerID	Customer	CustomerAddress
101	Alice	NY, USA

BCNF (Boyce-Codd Normal Form) - More Strict Than 3NF
Rule:

- Must be in **3NF**.
- Every determinant must be a **candidate key**.

Example (Before BCNF - Multi-Keys Issue):

Professor	Course	Department
Dr. A	DBMS	CS
Dr. B	Math	Math

- **Issue**: `Course → Department`, but `Professor → Course`, causing dependency issues.

After BCNF (Split Tables to Remove Dependency):

✅ Professor Table

Professor	Course
Dr. A	DBMS
Dr. B	Math

✅ Course Table

Course	Department
DBMS	CS
Math	Math

4NF (Fourth Normal Form) - Remove Multi-Valued Dependencies

Rule:

- Must be in **BCNF**.
- One row should not contain **two independent** multi-valued facts.

Example (Before 4NF - Multi-Valued Dependency):

Student	Course	Hobby
Alice	Math	Chess
Alice	Math	Music

- **Issue**: `Course` and `Hobby` are **independent** but stored together.

After 4NF (Separate into Two Tables):

✅ Student_Course Table

Student	Course
Alice	Math

✅ Student_Hobby Table

Student	Hobby
Alice	Chess
Alice	Music

Normal Form	Purpose	Fixes
1NF	Eliminate duplicate columns, ensure atomicity	Split repeating groups into separate rows
2NF	Remove partial dependencies	Create separate tables for independent attributes
3NF	Remove transitive dependencies	Ensure non-key columns depend only on the primary key
BCNF	Stronger than 3NF, eliminate non-key dependencies	Ensure every determinant is a candidate key
4NF	Remove multi-valued dependencies	Separate independent multi-valued attributes

MYSQL workbench installation was done !