

Database Design and Query

Section 2 : Designing Database

- Fundamental Principles
 - Database Terminology
 - Data Types
- Database Normalization
- Design Process
- Map Out Structure

Database Terminology

- Collection of information organized in tables
 - **Tables are also “relations”**
- Tables are constructed and associated to each other through shared fields – “common” fields
 - **Fields are also “columns” or “attributes”**
- A set of attributes comprises a record
 - **Records are also “rows” or “tuples”**
- Tables are related through common fields designated as primary and foreign keys
- Allow us to find, update, and delete data quickly, and help to ensure accuracy

Primary Key and Foreign Key Fields

- **Primary Key**

- Primary key fields must be unique and cannot contain a null value.
- Each table should have a primary key field.
- Concatenated keys: using more than one field as a primary key field.

- **Foreign Key** : Fields in a table that refer to the primary key in another table

- The data in this field must exactly match data contained in the primary key field.

Data Types

- A database table contains multiple columns with specific data types such as numeric or string. MySQL provides more data types other than just numeric or string. Each data type in MySQL can be determined by the following characteristics:
 - The kind of values it represents.
 - The space that takes up and whether the values is a fixed-length or variable length.
 - The values of the data type can be indexed or not.
 - How MySQL compares the values of a specific data type.

Numeric Data Types

- You can find all SQL standard numeric types in MySQL including exact number data type and approximate numeric data types including integer, fixed-point and floating point. In addition, MySQL also supports BIT data type for storing bit field values. Numeric types can be signed or unsigned except the BIT type.

Numeric Types	Description
TINYINT	A very small integer
SMALLINT	A small integer
MEDIUMINT	A medium-sized integer
INT	A standard integer
BIGINT	A large integer
DECIMAL	A fixed-point number
FLOAT	A single-precision floating point number
DOUBLE	A double-precision floating point number
BIT	A bit field

String Data Types

- In MySQL, a string can hold anything from plain text to binary data such as images and files. The string can be compared and searched based on pattern matching by using the LIKE operator, regular expression, and full-text search.

String Types	Description
CHAR	A fixed-length nonbinary (character) string
VARCHAR	A variable-length non-binary string
BINARY	A fixed-length binary string
VARBINARY	A variable-length binary string
TINYBLOB	A very small BLOB (binary large object)
BLOB	A small BLOB
MEDIUMBLOB	A medium-sized BLOB
LOB	A large BLOB
TINYTEXT	A very small non-binary string
TEXT	A small non-binary string
MEDIUMTEXT	A medium-sized non-binary string
LONGTEXT	A large non-binary string
ENUM	An enumeration; each column value may be assigned one enumeration member
SET	A set; each column value may be assigned zero or more set members

Date and Time Data Types

- MySQL provides types for date and time as well as a combination of date and time. In addition, MySQL supports timestamp data type for tracking the changes of a row in a table. If you just want to store the year without date and month, you can use YEAR data type.

Date and Time Types	Description
DATE	A date value in 'CCYY-MM-DD' format
TIME	A time value in 'hh:mm:ss' format
DATETIME	A date and time value in 'CCYY-MM-DD hh:mm:ss' format
TIMESTAMP	A timestamp value in 'CCYY-MM-DD hh:mm:ss' format
YEAR	A year value in CCYY or YY format

Spatial Data Types

- MySQL supports many spatial data types that contain various kinds of geometrical and geographical values as shown in the following table.

Spatial Data Types	Description
GEOMETRY	A spatial value of any type
POINT	A point (a pair of X-Y coordinates)
LINESTRING	A curve (one or more POINT values)
POLYGON	A polygon
GEOMETRYCOLLECTION	A collection of GEOMETRY values
MULTILINESTRING	A collection of LINESTRING values
MULTIPOINT	A collection of POINT values
MULTIPOLYGON	A collection of POLYGON values

Relational Model Criteria

- Each column value must be a **single value** only.
- All value for e given column must be of the **same data type**.
- Each column name **must be unique**.
- The **order** of columns is **insignificant**.
- No two rows in a relation can be **identical**.

Functional Dependencies

A **Functional Dependencies** describe a relation between columns within **single relation**.

- A column is dependent on another if one value can be used to determine the value of another.
- Example :
 - **productName** is functional dependent on **id** because **id** can be used to uniquely determine the value of **productName**.
 - This **id** is called **Primary Key**.

Common Design Mistakes

- Tables with **too many** or with fields that **do not relate** to each other.
- Too many tables with **similar data**.
- Repeated rows.
- Using **comma separated values** or multiple values in a single row.
- Poor naming conventions.
- Poor or no planning.
- Non-normalized data.

Database Normalization

- Normalization is the process of organizing the fields and tables of relational database to minimize **redundancy** and **dependency**.
- This can involve dividing larger tables into smaller tables and **defining relationships** between them.
- The objective is to isolate data so that actions in a field can be made in one table and then propagated through the rest of the needed tables using **properly defined relationships**.

First Normal Form (1NF)

- There are no repeating or duplicate fields.
- Each row contains only a single value.
- Each record is unique.
 - Identified by primary key.

Example (1NF)

Item	Color	Price	Tax
T-Shirt	Red, Blue	60.000	6.000
Polo	Red, Yellow	75.000	7.500
Sweatshirt	Green, Black	75.000	7.500
Polo	Red, Yellow	75.000	7.500

- Table is not in first normal form because:
 - Multiple items in color field
 - Duplicate records / no primary key

Example (1NF)

Id [PK]	Item	Color	Price	Tax
1	T-Shirt	Red	60.000	6.000
2	T-Shirt	Blue	60.000	6.000
3	Polo	Red	75.000	7.500
4	Polo	Yellow	75.000	7.500
5	Sweatshirt	Green	75.000	7.500
6	Sweatshirt	Black	75.000	7.500

- Table is now in first normal form

Second normal form (2NF)

- Should be in 1NF.
- All non-key fields depend on all components of the primary key.
 - Guaranteed when primary key is a single field
- No partial dependencies.

Example (2NF)

Id [PK]	Item	Color	Price	Tax
1	T-Shirt	Red	60.000	6.000
2	T-Shirt	Blue	60.000	6.000
3	Polo	Red	75.000	7.500
4	Polo	Yellow	75.000	7.500
5	Sweatshirt	Green	75.000	7.500
6	Sweatshirt	Black	75.000	7.500

- Table is not in second normal form because:
 - price and tax depend on item, but not color

Example (2NF)

Id [PK]	Item	Price	Tax
1	T-Shirt	60.000	6.000
2	Polo	75.000	7.500
3	Sweatshirt	75.000	7.500

ItemId [FK]	Item	Color
1	T-Shirt	Red
1	T-Shirt	Blue
2	Polo	Red
2	Polo	Yellow
3	Sweatshirt	Green
3	Sweatshirt	Black

- Tables are now in second normal form

Third Normal Form (3NF)

- Should be in 2NF.
- Every non-prime attribute of table must dependent on primary key.

Example (3NF)

Id [PK]	Item	Price	Tax
1	T-Shirt	60.000	6.000
2	Polo	75.000	7.500
3	Sweatshirt	75.000	7.500

ItemId [FK]	Item	Color
1	T-Shirt	Red
1	T-Shirt	Blue
2	Polo	Red
2	Polo	Yellow
3	Sweatshirt	Green
3	Sweatshirt	Black

- Tables are not in third normal form because:
 - tax depends on price, not item

Example (3NF)

Id [PK]	Item	PriceId [FK]
1	T-Shirt	1
2	Polo	2
3	Sweatshirt	2

Id [PK]	Price	Tax
1	60.000	6.000
2	75.000	7.500

ItemId [FK]	Item	Color
1	T-Shirt	Red
1	T-Shirt	Blue
2	Polo	Red
2	Polo	Yellow
3	Sweatshirt	Green
3	Sweatshirt	Black

- Tables are now in third normal form

Exercise

Student Name	Assignment 1	Assignment 2
Michael Suyama	Article Summary	Poetry Analysis
Nancy Davolio	Article Summary	Reaction Paper
David Buchanan	Article Summary	Poetry Analysis

- Table is not in normal form

Exercise

Assignment ID	Assignment Name
1	Article Summary
2	Poetry Analysis
3	Reaction Paper

Student ID	First Name	Last Name
1	Michael	Suyama
2	Nancy	Davolio
3	David	Buchanan

Student ID	Assignment ID
1	1
1	2
2	1
2	3
3	1
3	2

- Tables are now in third normal form