

1

Database Design

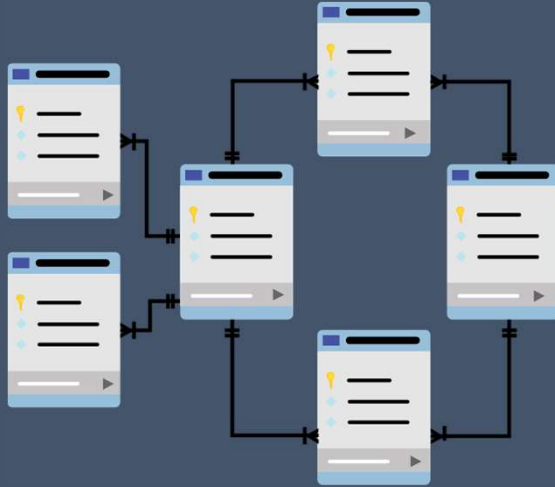
Module 6 - Concepts

1. · Database design fundamentals
2. · Identifying tables & assigning columns · Relationships between tables
3. · Primary keys and foreign keys
4. · Data types in MySQL

2

Designing Your Database – Where to Start?

1. Design your tables.
2. Design your columns.
3. Design your data types.
4. Create a schema.



3

Step 1: Design Your Tables

- Create a model of **real-world entities**
- What are the components of your model?
- Create a table for each entity category
- Each row represents a single unique entity

http5112-database: stock_items	
id	int(255)
item	varchar(255)
price	int(255)
inventory	varchar(255)
category	varchar(255)

http5112-database: employees	
id	varchar(255)
last_name	varchar(255)
first_name	varchar(255)
sin	int(10)
phone	varchar(255)
role	varchar(255)

http5112-database: sales	
id	int(255)
date	date
item	varchar(255)
employee	int(255)

4

Entities

Each table is about an object, person or thing

Customers

Books

Teachers

Stock_items

Borrowers

Students

Sales

Appointments

Classes

Patients

Doctors

5

Pet Store Tables

- employees (primary)
- stock_items (primary)
- sales (bridging)
- What else?
- suppliers
- customers

6

Step 2: Design Your Columns

- What questions do you need to answer?
- What attributes of the entity will answer these questions?
- What is the unique identifier for this table (primary key)?
- Does this table need to refer to other tables (foreign keys)?
- Only one value per column.

7

Stock_items

id	item	price	inventory	category
Primary key for this table.	Name of stock item.	Retail price of item.	How many currently in stock.	Animal group this item is for.
1001	Luxury dog bed	95	2	Canine
1002	Super size chew toy	15	45	Canine
1003	Water dish	87	11	Feline
1004	Fish bowl starter kit	16	22	Piscine
1005	Luxury cat bed	89	0	Feline
1006	Just as good dog bed	74	1	Canine
1007	Deluxe Aquarium	12	10	Piscine
1008	Terrarium	231	24	Murine
1009	Cat stand	55	6	Feline
1010	Premium Terrarium	524	2	Murine
1011	Water dish	10	88	Canine

Employees table

id	last_name	first_name	sin	phone	role
Primary key for this table.			Social Insurance Number		
110	Maracle	James	123456789	4165551234	Manager
111	Marquis	Henry	987654321	4165551235	Sales
112	Kaur	Jas	456123789	4165551236	Sales
113	Neema	Hank	654987321	4165551237	Assistant
114	Said	Farud	258741369	4165551238	Sales
115	Fitzpatrick	Henry	753951842	4165551239	Stock

Sales table

id	date	item	employee
		FK for stock_items	FK for employees
21000	2021-06-09	1002	111
21016	2021-06-13	1002	111
21006	2021-06-11	1002	111
21027	2021-06-17	1003	114
21013	2021-06-13	1003	112
21001	2021-06-09	1003	111
21002	2021-06-09	1004	111
21014	2021-06-15	1004	112
21032	2021-06-18	1004	111

Primary key
Foreign key

Stock_items x employees

8

Step 3: Design Your Data Types

- MySQL has set data types that it will accept.
- Only one data type per column. You will specify the type and (usually) the size.
- 3 categories: numeric, string, and date/time.

9

Numeric Data Types

- **INT(x)** for integers. In parentheses, specify how many characters your integer can have:
 - `id INT(3)` `inventory INT(4)` `serial INT(6)`
- **DEC(x, y)** for decimals. In parentheses, specify how many characters your number can have (x), and how many decimal places (y):
 - `price DEC(5,2)` `expenses DEC(7,2)`
 - Also: FLOAT and DOUBLE for scientific values
- **BOOL** for Booleans. This stores 1 (true) or 0 (false).

10

String Data Types

- **CHAR(x)** for fixed-length strings. In parentheses, specify how many characters your string can have (max 255):
 - `phone CHAR(10)` `sin CHAR(9)`
- **VARCHAR(x)** for strings of unknown length. In parentheses, specify how many characters your string can have (max 255):
 - `name VARCHAR(25)` `product VARCHAR(15)`
- **TEXT** for strings longer than 255 characters
 - `description TEXT`

11

Variations

- **INT and TEXT** have variations that limit the size that can be stored.
 - `Int` = -2147483648 to 214748364
 - `MEDIUMINT`
 - `SMALLINT`
 - `TINYINT` = -128 to 127
- **BEWARE: Excess data will not be stored.**
- How much space do you need?
- Like a buffet: take what you need; eat what you take.

12

Date/Time Data Types

- **DATE** for storing a date value. Format: YYYY-MM-DD
 - `created DATE` `due_date DATE`
- **TIME** for storing a time value. Format: HH:MM:SS
 - `class_start TIME` `first_name VARCHAR(30)`
 `product VARCHAR(15)`
- **DATETIME** for storing both.
 - Format: YYYY-MM-DD HH:MM:SS
- **TIMESTAMP** is DATETIME but generated by the SQL server.

13

Step 4: Create a Schema

- A schema is a blueprint for a database
- It can also contain the instructions for how to build it
- It does not contain data, it is simply the recipe for your DB
- Start with paper and pencil

14

Summary

When creating a structure for your data, decide on
For a database, how many tables to include
For each table, How many columns each table will have
For each column, what kind of data will be stored

15

References

- Thomson, L., & Welling, L. (2003). *MySQL tutorial*. Sams.
 - Chapter 3: Database Design Crash Course (pp. 27-32)
 - Chapter 4: Creating Databases, Tables and Indexes (pp. 51-54)
- W3Schools MySQL Data Types reference:
https://www.w3schools.com/sql/sql_datatypes.asp

16

Assignment 6 & Quiz 2

- Due Friday 11:59PM
- Quiz will not be available past the due date