Databases and SQL for Data Science with Python

Why SQL is Essential for Data Scientists?

SQL = The Key to Access and Prepare Your Data 🔑

- **Data Lives in Databases**
  Most real-world data is stored in relational databases (MySQL, PostgreSQL, SQL Server).

- **SQL is the Language to Talk to Data**
  Retrieve only the data you need using filtering, sorting, grouping, and joining.

- **Prepare Data at the Source**
  Clean, filter, and merge data before importing it into Python, R, or BI tools.

- **Understand Data Structure**
  Learn table relationships, data types, and data quality directly from the database.

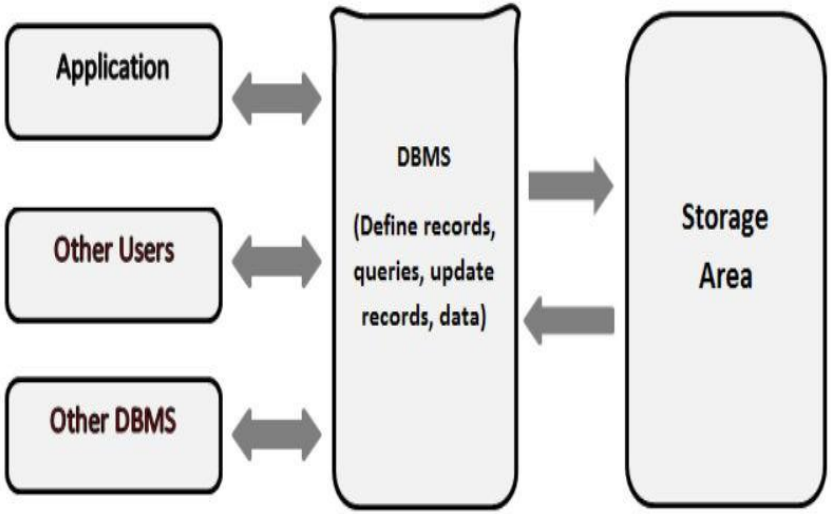
- **Highly Demanded Skill**
  Most Data Science job descriptions list SQL as a core requirement.

"Before you analyze data, you must know how to access it." **– IBM Data Science Approach**

What are Databases?

- A database is an organized collection of data, generally stored and accessed electronically from a computer system. It supports the storage and manipulation of data.

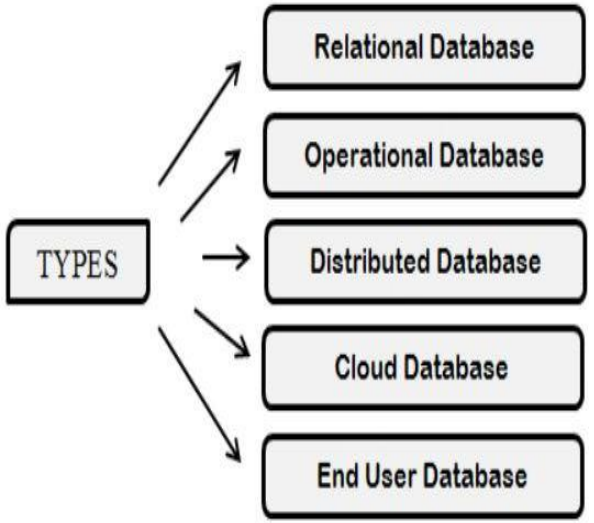
- In other words, databases are used by an organization as a method of storing, managing and retrieving information.



Types of Databases

Depending upon the usage requirements, there are following types of databases available in the market:

- Centralized database
- Distributed database
- Personal database
- End-user database
- Commercial database
- NoSQL database
- Operational database
- Relational database
- Cloud database
- Object-oriented database
- Graph database Here is a detailed article on, Types of Database Management Systems.



Advantages of using Databases

There are many advantages of databases

- Reduced data redundancy
- Reduced updating errors and increased consistency
- Greater data integrity and independence from application programs
- Improved data access to users through the use of host and query languages
- Improved data security
- Reduced data entry, storage, and retrieval costs

Disadvantages of using Databases

There are many disadvantages of databases

- Although databases allow businesses to store and access data efficiently, they also have certain disadvantages
- Complexity
- Cost
- Security
- Compatibility

Some examples of Databases

Some of the most popular databases are

- Oracle Database
- Sybase
- MySQL
- IBM db2

What is SQL?

- SQL (Structured Query Language): Is used to perform operations on the records stored in the database, such as updating records, inserting records, deleting records, creating and modifying database tables, views, etc.

- **SQL is not a database system, but it is a query language.**

Database Schema

A schema is a group of related objects in a database. There is one owner of a schema who has access to manipulate the structure of any object in the schema. A schema does not represent a person, although the schema is associated with a user that resides in the database.

Database Schema

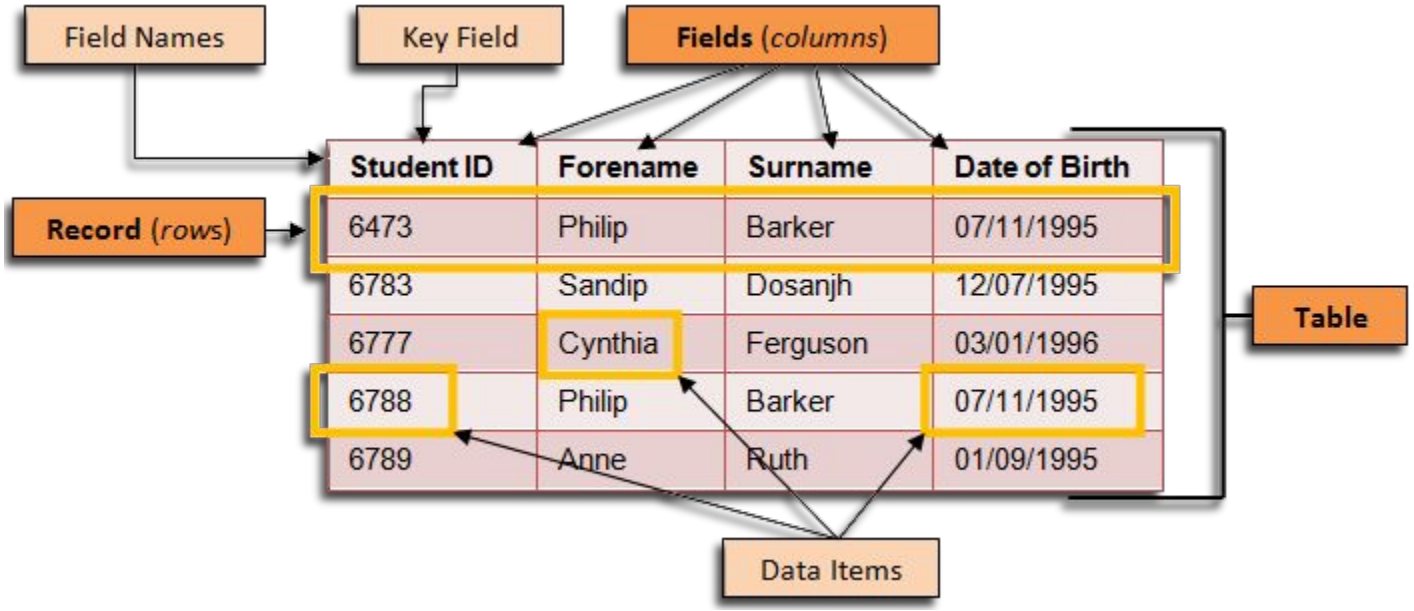
Database schema is the structure of a database, refers to the organization of data as a blueprint that demonstrates how the database is constructed.

So, a database schema describes how the data may relate to other tables or other data models. However, the schema does not actually contain data.

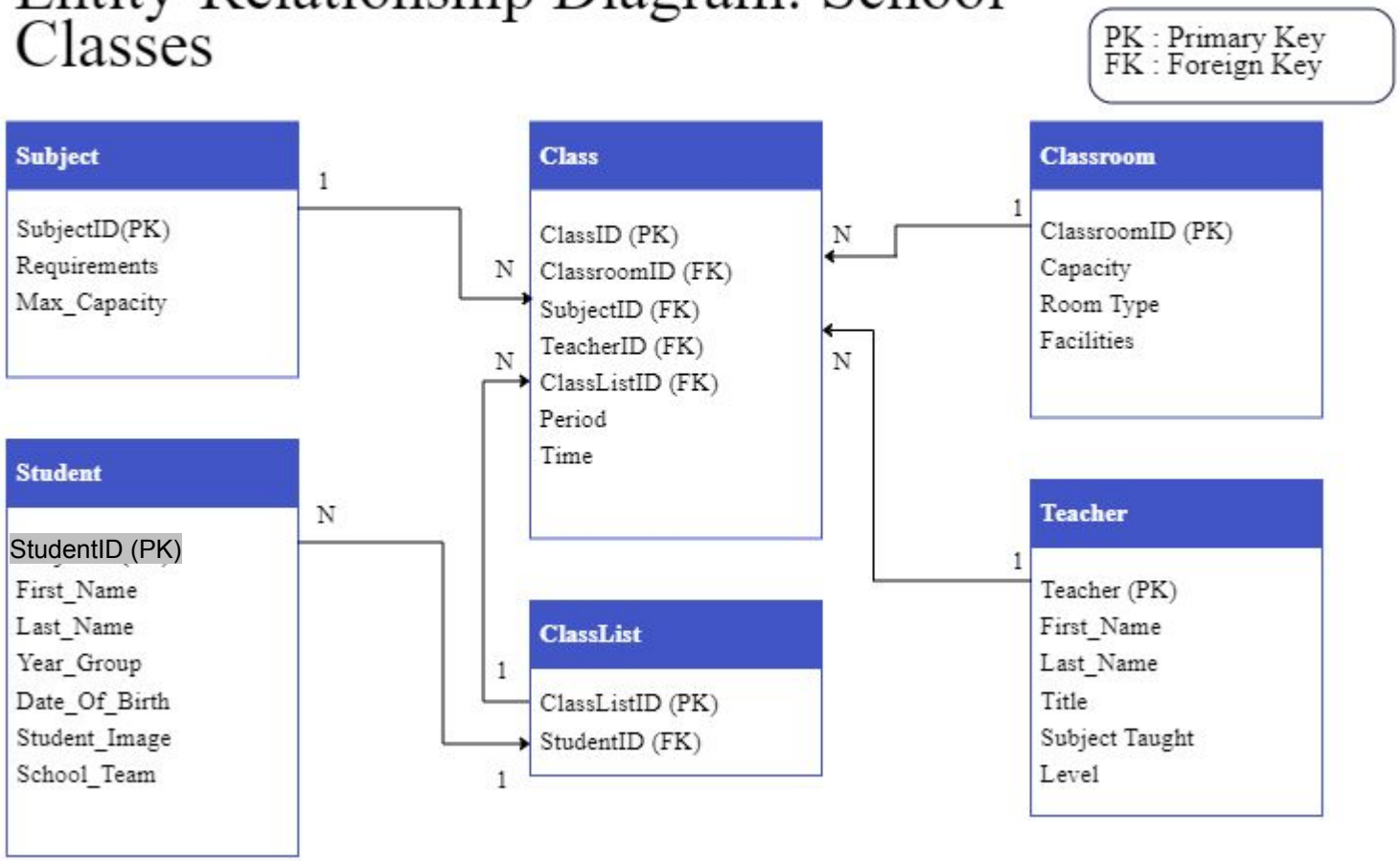
Database instance

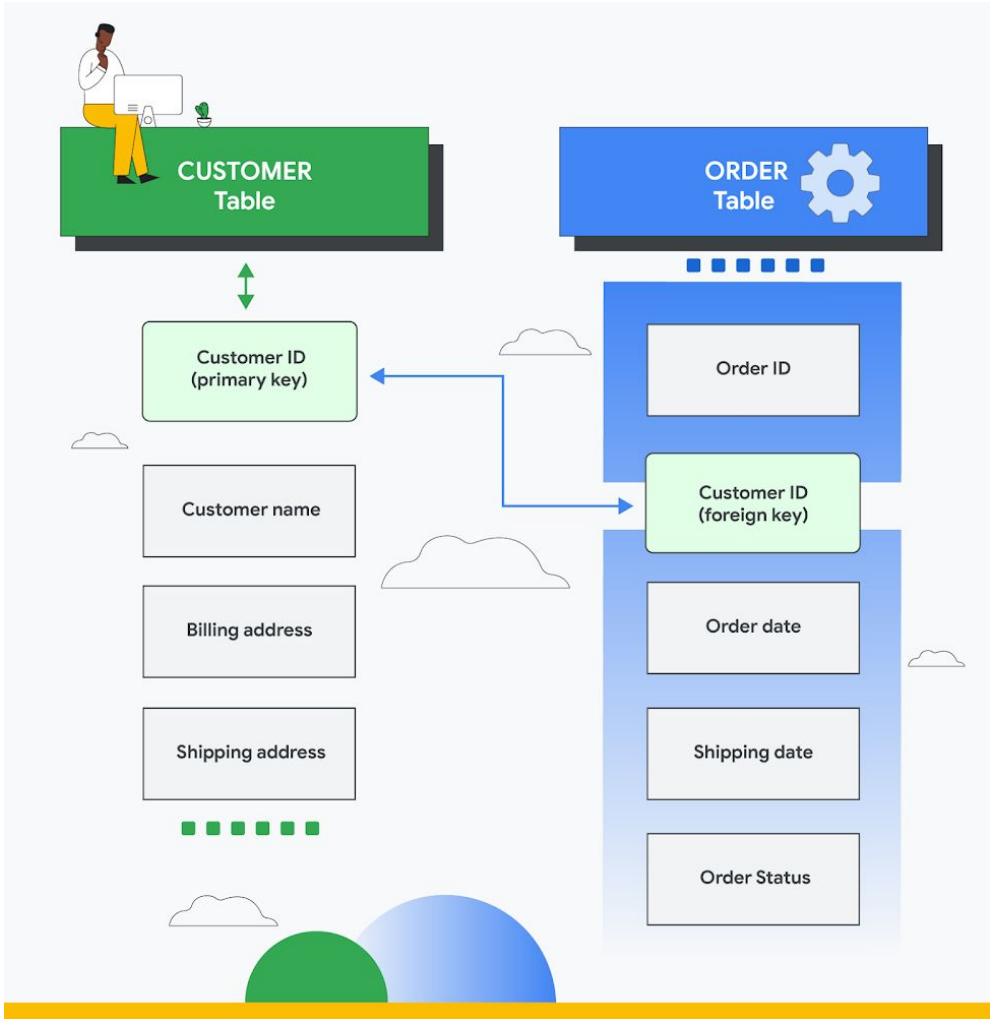
Database instance is a snapshot of data in a database at a single moment in time . It contains all the properties that the schema describes as data values.

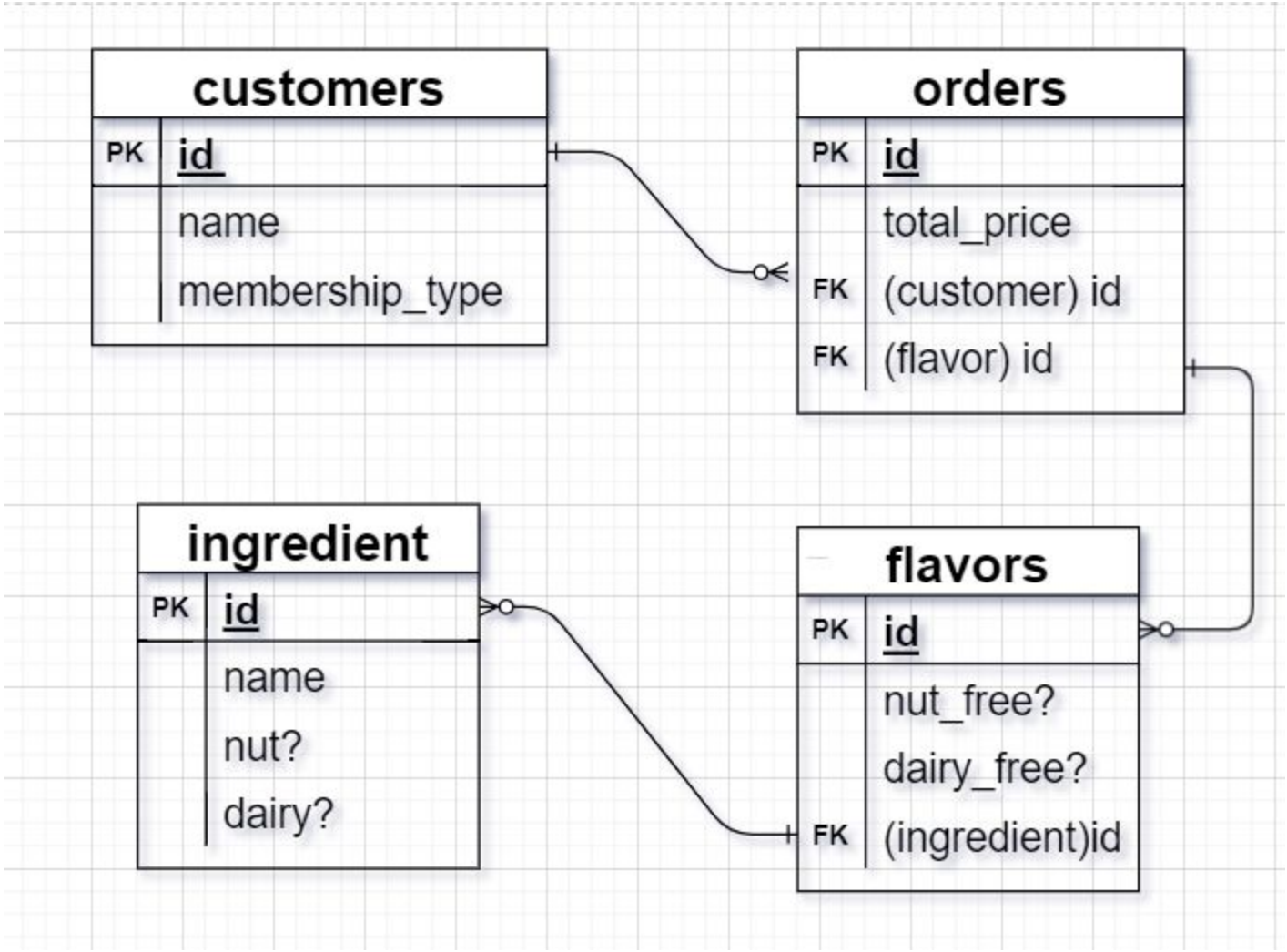
Since database instances are just a snapshot at a given moment, they're likely to change over time, unlike database schemas.


Field Names
Key Field
Fields (columns)
Record (rows)
Table
Data Items
Student ID
Forename
Surname
Date of Birth
6473
Philip
Barker
07/11/1995
6783
Sandip
Dosanjh
12/07/1995
6777
Cynthia
Ferguson
03/01/1996
6788
Philip
Barker
07/11/1995
6789
Anne
Ruth
01/09/1995

Entity-Relationship Diagram: School Classes




CUSTOMER Table
ORDER Table
Customer ID (primary key)
Customer name
Billing address
Shipping address
Order ID
Customer ID (foreign key)
Order date
Shipping date
Order Status


customers
PK
id
name
membership_type
orders
PK
id
total_price
FK
(customer) id
FK
(flavor) id
ingredient
PK
id
name
nut?
dairy?
flavors
PK
id
nut_free?
dairy_free?
FK
(ingredient)id

Database Constraints

- Primary Key (Not Null + Unique)
- Not Null
- Unique Key
- Referential Integrity (Foreign key FK)
- Check

Data types

A data type determines the type of data that can be stored in a database column. The most commonly used data types are:

1. Alphanumeric: data types used to store characters, numbers, special characters, or nearly any combination.
2. Numeric
3. Date and Time

MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 214748-3647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

- **Primary key**:

  - A field in a table that uniquely identifies each rows in a database table.

  - Primary keys must contain unique values.

  - A table can have only one Primary keys.

```
--define primary key
CREATE TABLE students(
 student id int PRIMARY KEY,
 name varchar(20));
```

- **Foreign key:**

  - Reference a column in another table to define the relationship between two tables.

  - The relationship between 2 tables matches the Primary Key in one of the tables with a Foreign Key in the second table.

```
--define primary key
CREATE TABLE teachers(
 id int PRIMARY KEY,
 teache_name varchar(30),
);
```

```
CREATE TABLE classes(
 id integer PRIMARY KEY,
 teacher_id int REFERENCES teachers(id), --foreign key is created
 category varchar(40),
);
```

Database Transaction

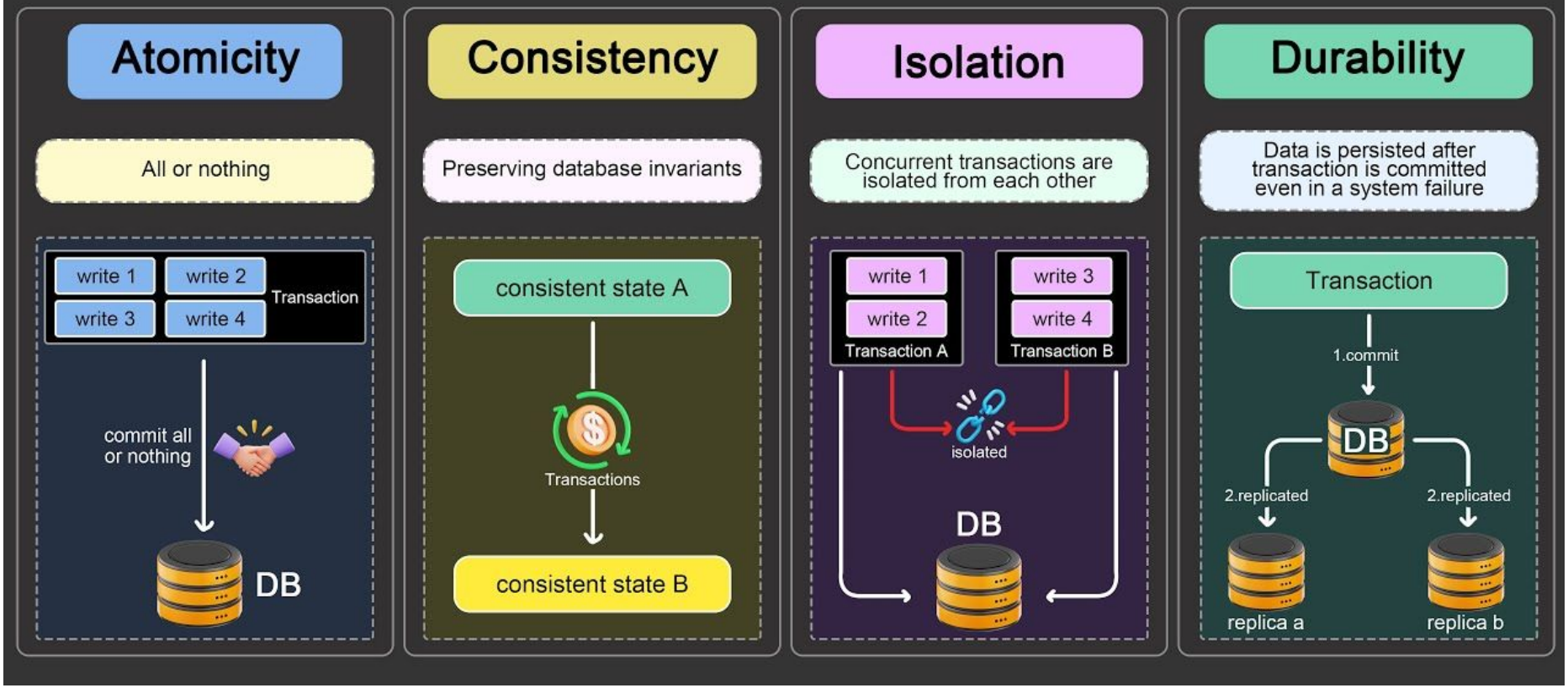
- A transaction is an executing program that forms a logical unit of database actions.
- It includes one or more database access operations such as insert, delete and update.
- The database operations that form a transaction can either be embedded within an application program or they can be specified interactively via a high-level query language such as SQL.

Database Transaction Properties

- Transactions should possess several properties, often called the ACID properties:

1. Atomicity
2. Consistency
3. Isolation
4. Durability

What does ACID Really Mean?

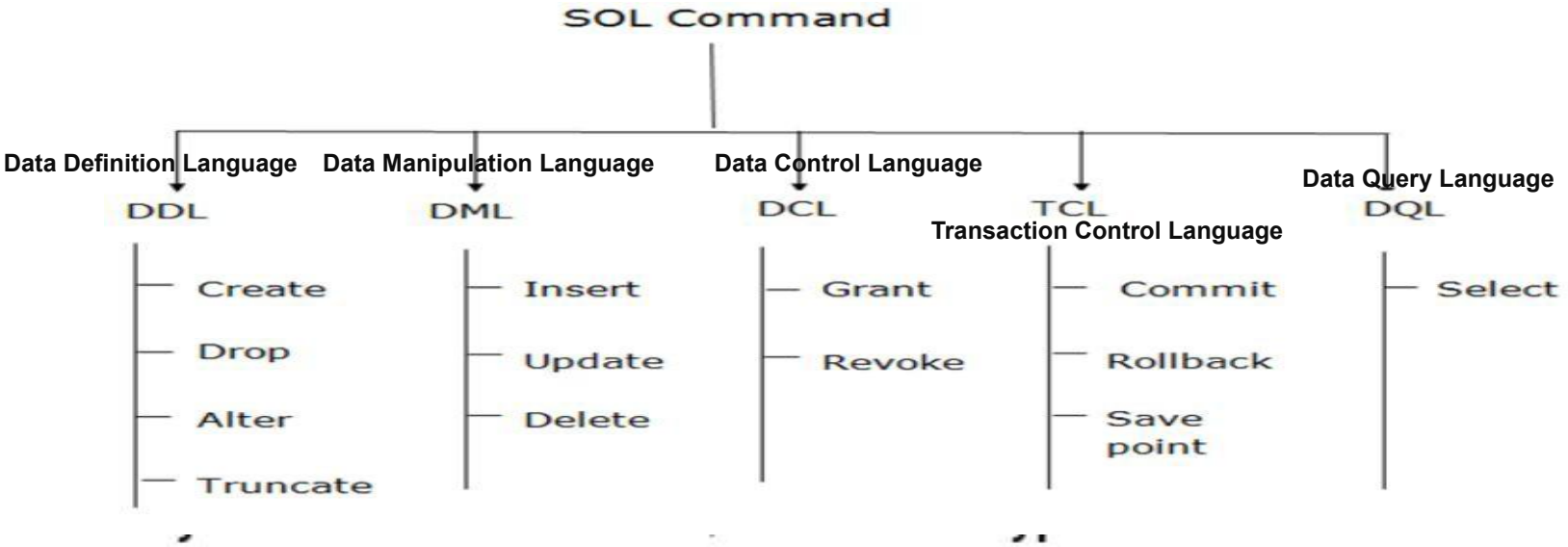


SQL Commands

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.

- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

Types of SQL Commands



Type	Primary Function	Examples
DDL	Defines database structure	`CREATE`, `ALTER`, `DROP`, `TRUNCATE`
DML	Manipulates data in tables	`INSERT`, `UPDATE`, `DELETE`
DCL	Manages permissions and roles	`GRANT`, `REVOKE`
TCL	Controls database transactions	`COMMIT`, `ROLLBACK`, `SAVEPOINT`
DQL	Retrieves data from the database	`SELECT`

Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.

- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

Data Definition Language (DDL)

- CREATE It is used to create a new table in the database

```
CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,....]);
```

Example

```
CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VARCHAR2(100), DOB DATE);
```

Data Definition Language (DDL)

- DROP: It is used to delete both the structure and record stored in the table.

```
DROP TABLE table_name;
```

Example

.

```
DROP TABLE EMPLOYEE;
```

Data Definition Language (DDL)

- ALTER: It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

```
ALTER TABLE table_name ADD column_name COLUMN-definition;
```

Example

.

```
ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));
ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));
```

Data Definition Language (DDL)

- TRUNCATE: It is used to delete all the rows from the table and free the space containing the table.

```
TRUNCATE TABLE table_name;
```

Example

```
TRUNCATE TABLE EMPLOYEE;
```

Data Definition Language (DDL)

- TRUNCATE: It is used to delete all the rows from the table and free the space containing the table.

```
TRUNCATE TABLE table_name;
```

Example

```
TRUNCATE TABLE EMPLOYEE;
```

Data Manipulation Language (DML)

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

•Here are some commands that come under DML:

- INSERT
- UPDATE
- DELETE

Data Manipulation Language (DML)

- INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table

```
INSERT INTO TABLE_NAME
VALUES (value1, value2, value3, .... valueN);
```

Example:

.

```
INSERT INTO javatpoint (Author, Subject) VALUES ("Sonoo", "DBMS");
```

Data Manipulation Language (DML)

- UPDATE: This command is used to update or modify the value of a column in the table.

```
UPDATE table_name SET [column_name1= value1,...column_nameN = valueN] [WHERE CONDITION]
```

Example:
.

```
UPDATE students
SET User_Name = 'Sonoo'
WHERE Student_Id = '3'
```

Data Manipulation Language (DML)

- DELETE: It is used to remove one or more row from a table.

```
DELETE FROM table_name [WHERE condition];
```

Example:

.

```
DELETE FROM javatpoint
WHERE Author="Sonoo";
```

Data Control Language (DCL)

- DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL:

- Grant
- Revoke

Data Control Language (DCL)

- Grant: It is used to give user access privileges to a database.

Example:

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

Data Control Language (DCL)

- Revoke: It is used to take back permissions from the user.

Example:

```
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

Transaction Control Language(TCL)

- TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.
- These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

Transaction Control Language(TCL)

- Commit: Commit command is used to save all the transactions to the database.

```
COMMIT;
```

Example

```
DELETE FROM CUSTOMERS
WHERE AGE = 25;
COMMIT;
```

Transaction Control Language(TCL)

- Rollback: Rollback command is used to undo transactions that have not already been saved to the database.

```
ROLLBACK;
```

Example

```
DELETE FROM CUSTOMERS
WHERE AGE = 25;
ROLLBACK;
```

Transaction Control Language(TCL)

- SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Example

```
SAVEPOINT SAVEPOINT_NAME;
```

MySQL VS PostgreSQL VS SQL Server

Core Idea

SQL basics are the same across systems

- Core CRUD commands (`SELECT`, `INSERT`, `UPDATE`, `DELETE`) work almost the same in MySQL, PostgreSQL, and SQL Server.
- Learning SQL in one system makes it easy to move to another.

Strengths of Each System

- **MySQL**: Simple, widely used, great for learning and small-to-medium projects.
- **PostgreSQL**: Rich advanced features (JSONB, Arrays, GIS), strong for complex data analysis.
- **SQL Server**: Strong integration with Microsoft tools, advanced stored procedures, business analytics support.

SQL is one language, but each system has its own accent.

Master the basics first—then adapting to another system is just learning a few new words.

Differences Are in the Details and if u need

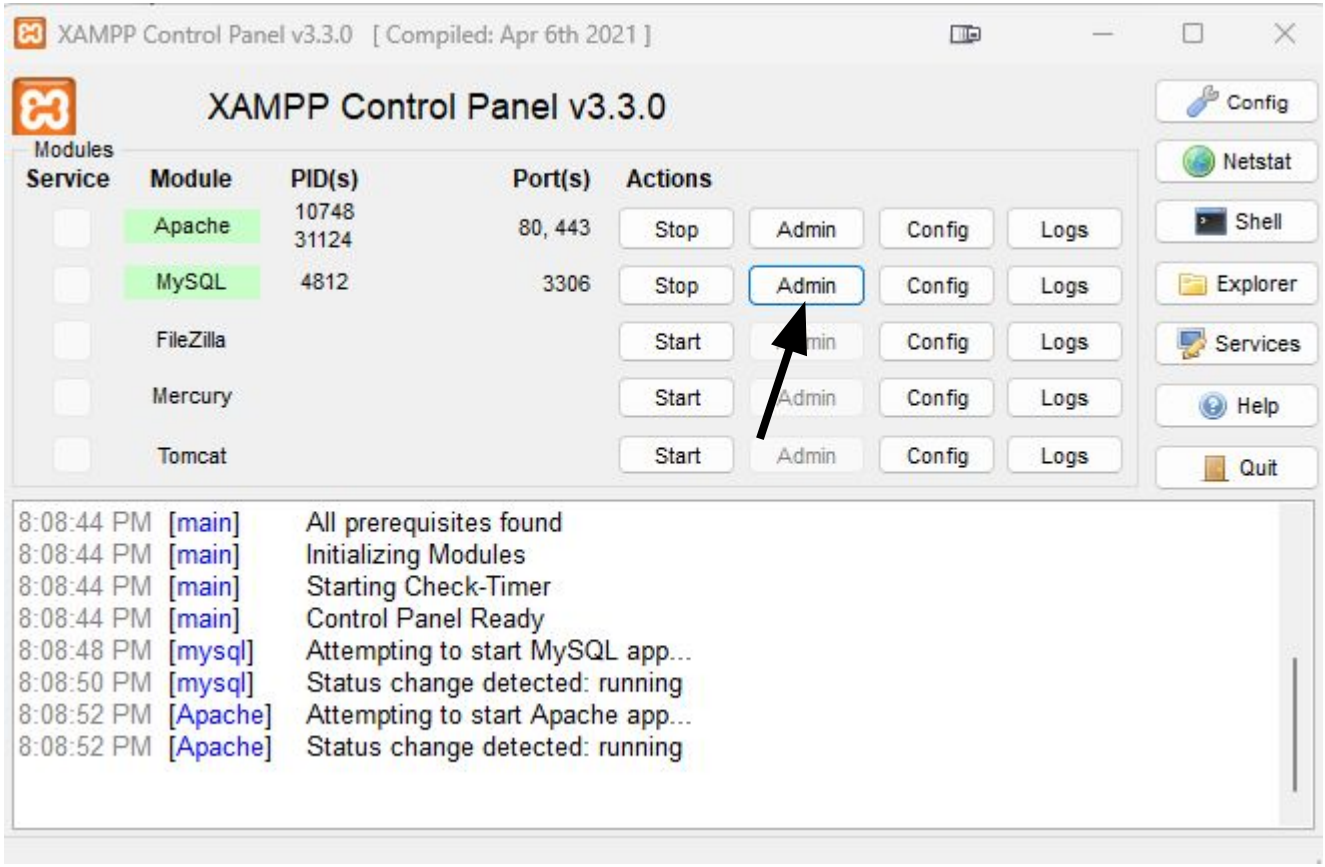
Main differences appear in:

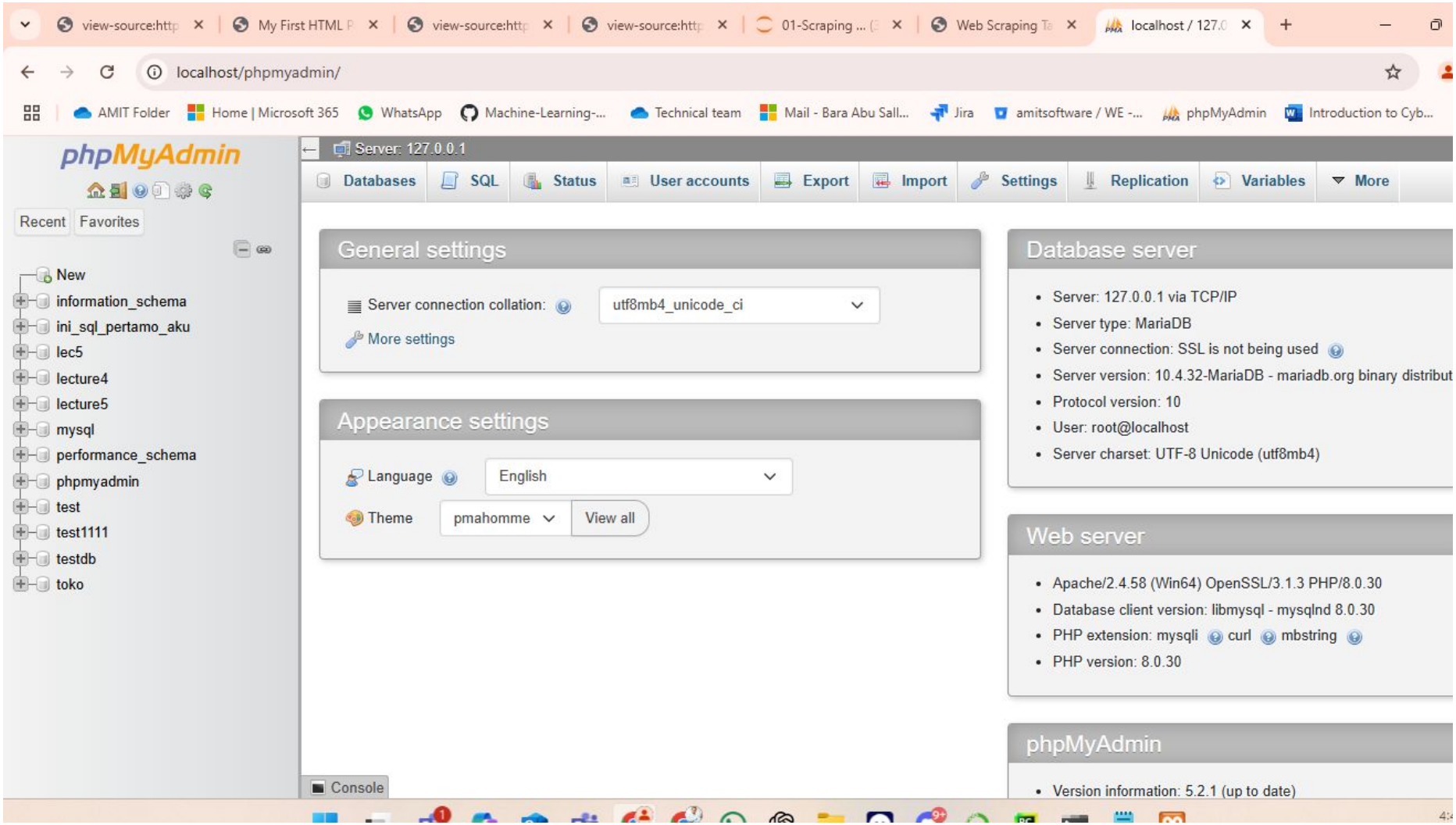
- **Built-in Functions**:
  - Text length → `LENGTH` (MySQL), `CHAR_LENGTH` (PostgreSQL), `LEN` (SQL Server)
  - Dates → `NOW()` (MySQL/PostgreSQL), `GETDATE()` (SQL Server)
- **Pagination**:
  - `LIMIT` in MySQL/PostgreSQL
  - `TOP` or `OFFSET…FETCH` in SQL Server
- **Data Types**: Boolean, JSON, Arrays
- **Auto Increment IDs**:
  - MySQL → `AUTO_INCREMENT`
  - PostgreSQL → `SERIAL` or `GENERATED AS IDENTITY`
  - SQL Server → `IDENTITY(1,1)`
- **Advanced Features**: Full-Text Search, Window Functions, Upsert.

Task XAMPP & MySQL

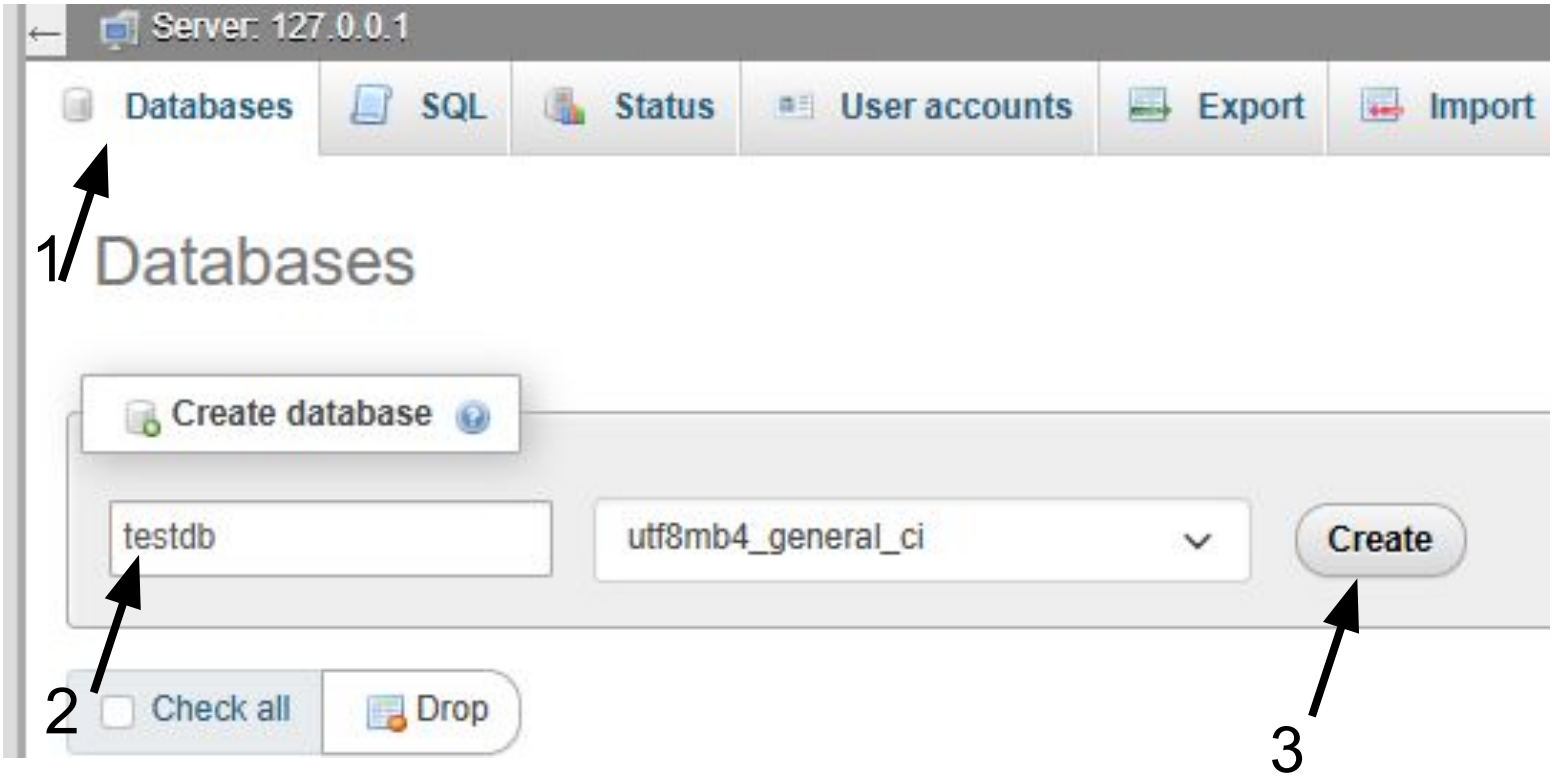
1. Install XAMPP
   a. https://www.apachefriends.org/download.html
2. Run Apache and MySQL
3. Create Database (YourName_db)
   a. Create 2 Table (customers,orders)
   b. Insert 3 customers (You, Baraa, Ali)
   c. Insert 6 orders (3 for you, 2 for Baraa,1 for Ali)

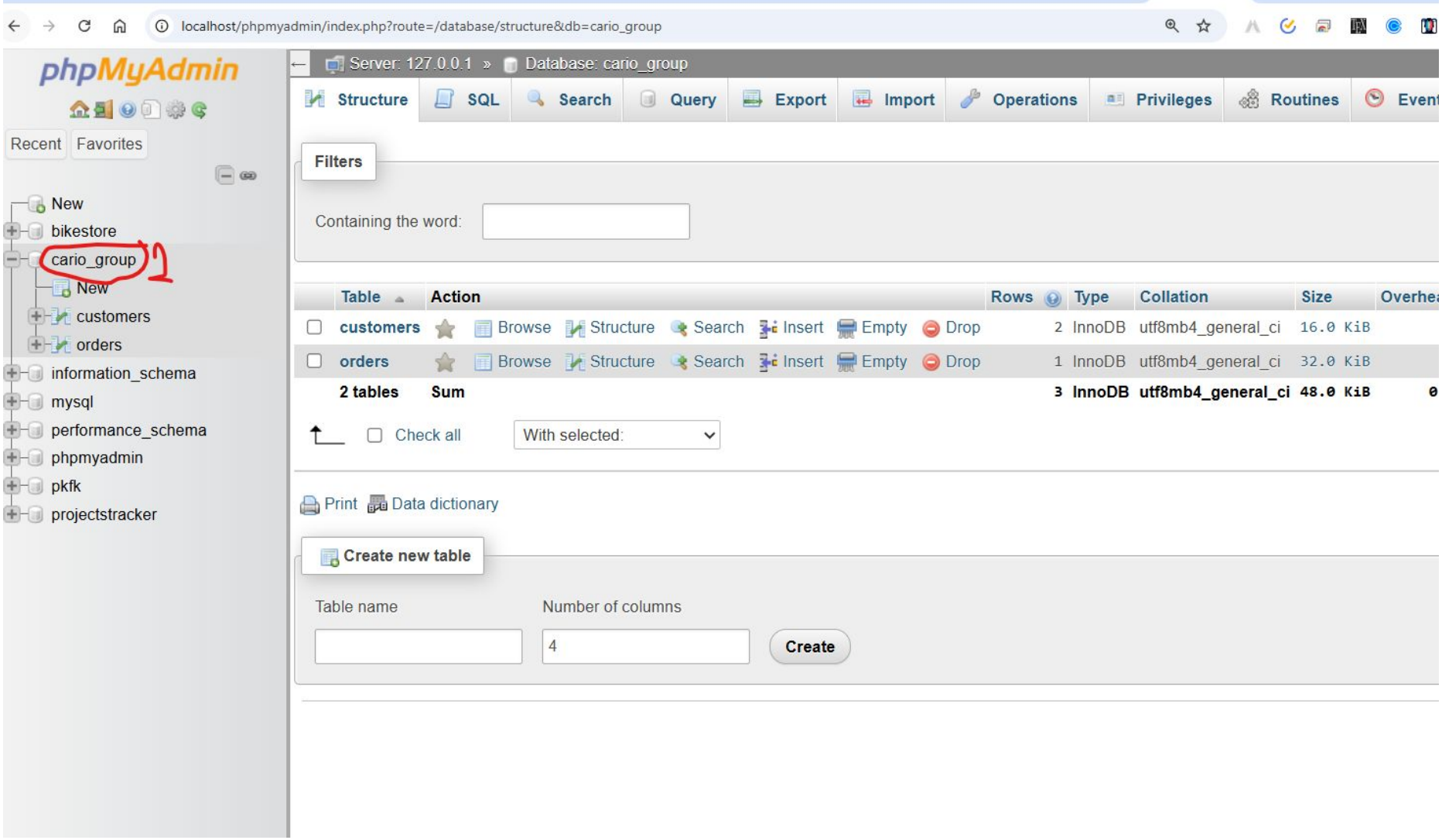
- **Run Apache and MySQL**




localhost/phpmyadmin/
phpMyAdmin
Recent
Favorites
New
information_schema
ini_sql_pertamo_aku
lec5
lecture4
lecture5
mysql
performance_schema
phpmyadmin
test
test1111
testdb
toko
Server: 127.0.0.1
Databases
SQL
Status
User accounts
Export
Import
Settings
Replication
Variables
More
General settings
Server connection collation:
utf8mb4_unicode_ci
More settings
Appearance settings
Language
English
Theme
pmahomme
View all
Database server
Server: 127.0.0.1 via TCP/IP
Server type: MariaDB
Server connection: SSL is not being used
Server version: 10.4.32-MariaDB - mariadb.org binary distribut
Protocol version: 10
User: root@localhost
Server charset: UTF-8 Unicode (utf8mb4)
Web server
Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.0.30
Database client version: libmysql - mysqlnd 8.0.30
PHP extension: mysqli curl mbstring
PHP version: 8.0.30
phpMyAdmin
Version information: 5.2.1 (up to date)
Console

Create Database (testDB)




localhost/phpmyadmin/index.php?route=/database/structure&db=cario_group
phpMyAdmin
Recent
Favorites
New
bikestore
cario_group
New
customers
orders
information_schema
mysql
performance_schema
phpmyadmin
pkfk
projectstracker
Server: 127.0.0.1 » Database: cario_group
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Event
Filters
Containing the word:
Table
Action
Rows
Type
Collation
Size
customers
Browse
Structure
Search
Insert
Empty
Drop
2
InnoDB
utf8mb4_general_ci
16.0 KiB
orders
Browse
Structure
Search
Insert
Empty
Drop
1
InnoDB
utf8mb4_general_ci
32.0 KiB
2 tables
Sum
3
InnoDB
utf8mb4_general_ci
48.0 KiB
0
Check all
With selected:
Print
Data dictionary
Create new table
Table name
Number of columns
4
Create

Run SQL query/queries on database baraa_db:

```sql
CREATE TABLE cust(
    cust_id int AUTO_INCREMENT PRIMARY KEY,
    name varchar(20),
    email varchar(100) UNIQUE,
    phone varchar(50) NOT NULL);

cREATE TABLE orders(
    order_id int AUTO_INCREMENT PRIMARY KEY,
    order_date DATE NOT NULL,
    amount DECIMAL(10,2) NOT null,
    custumerId INT,
    FOREIGN KEY (custumerId)  REFERENCES cust(cust_id)

)
```

insert into customers

Run SQL query/queries on table baraa_db.cust:

```
1 INSERT INTO cust (name,email,phone) VALUES
("Ali","ali@gmail.com","0108088808"),
("omer","omer@gmail.com","01088754857");
```

insert into orders