



# SQL

Structured Query Language



# Agenda

- 1 What is Data?
- 2 Introduction to SQL
- 3 SQL Database
- 4 SQL queries
- 5 SQL database on XAMPP Server



# WHAT IS DATA?

Data is a collection of information.

# WHAT IS DATABASE?

- A database is an collection of similar information, or data, typically stored electronically in a computer system



# WHAT IS STRUCTURED DATA?

- Structured data is when data is in a standardized format, has a well-defined structure, complies to a data model, follows a persistent order, and is easily accessed by humans and programs.



# WHAT IS UNSTRUCTURED DATA?

- Unstructured data is information that is not arranged according to a preset data model or schema, and therefore cannot be stored in a traditional relational database



# What is SQL?

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute(ANSI) in 1986, and of the International Organization for Standardization(ISO) in 1987.



# Using SQL in Your Web Site

To build a web site that shows data from a database, you will need:

- An RDBMS database program(i.e. MS Access,SQL Server,MySQL)
- Server-side scripting language, like PHP or ASP
- SQL to get the data you want
- HTML and CSS to style the page

# SQL Constraints

- Unique
- Not Null
- Primary Key
- Foreign Key
- Check

# Unique

The UNIQUE constraint ensures that all values in a column are different.

- Roll no.
- Email Id
- Adhaar Card
- PRN no.
- Phone no.
- Voter Id

# Students

Roll no.	Sname	Sage
1	Akash	20
2	Parth	20
3	Akash	20
4	Riya	20

# Not Null\*

By default, a column can hold NULL values.

Roll_no.	s_name	s_age
1	Akash	20
2	Parth	20
3	Akash	20
4	Riya	20



# primary Key

## { Unique + Not Null }

- The PRIMARY KEY constraint uniquely identifies each record in a table.
- Primary keys must contain UNIQUE values, and cannot contain NULL values
- A table can have only ONE primary key



# Student

Roll_no.	s_name	s_age
1	Vaidhahi	20
2	Mandar	20
3	Harshad	20

# Foreign Key

- The FOREIGN KEY constraint is a key used to link two tables together.
- FOREIGN KEY is used to maintain **Referential Integrity**



# Foreign Key

project

Roll no.	PID	Project Name
1	1	Machine Learning
2	2	IOT
3	1	Machine Learning

student

Roll no.	Sname	Sage
1	Vaidhahi	20
2	Mandar	20
3	Harshad	20

# Check

- The CHECK constraint is used to limit the value range that can be placed in a column.
- If you define a CHECK constraint on a column it will allow only certain values for this column.

eg. Check (age>18);

# Creating Table in SQL

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ...  
);
```

# Datatypes in SQL

char(n)	<b>It is used to specify a fixed length string</b>
varchar(n)	<b>It is used to specify a variable length string</b>
BOOL	<b>It is used to specify Boolean values true and false.</b>
INT(size)	<b>It is used for the integer value.</b>
FLOAT(size, d)	<b>It is used to specify a floating point number.</b>

```
1 -- create a table
2 CREATE TABLE students (
3
4     id INTEGER PRIMARY KEY,
5     name TEXT NOT NULL,
6     gender TEXT NOT NULL
7
8 );
```

1	Ryan	20
2	Joanna	20
3	vaidahi	20
4	mandar	19
5	harshad	20

```
1 -- create a table
2 CREATE TABLE STUDENTS(
3     SID int NOT NULL PRIMARY KEY,
4     SNAME varchar2(10),
5     AGE int
6 );
7 CREATE TABLE PROJECT(
8     PID int NOT NULL PRIMARY KEY,
9     SID int FOREIGNKEY REFERENCES STUDENTS(SID),
10    PNAME varchar2(10)
11 );
12 -- insert some values
13 INSERT INTO STUDENTS VALUES (1, 'Ryan', '20');
14 INSERT INTO STUDENTS VALUES (2, 'Joanna', '20');
15 INSERT INTO STUDENTS VALUES (3, 'vaidahi', '20');
16 INSERT INTO STUDENTS VALUES (4, 'mandar', '20');
17 INSERT INTO STUDENTS VALUES (5, 'harshad', '20');
18
19 INSERT INTO PROJECT VALUES (1, '3', 'IOT');
20 INSERT INTO PROJECT VALUES (2, '1', 'AI');
21 INSERT INTO PROJECT VALUES (3, '5', 'ML');
22 INSERT INTO PROJECT VALUES (4, '2', 'WEB D');
23 INSERT INTO PROJECT VALUES (5, '4', 'APP D');
24 -- -- fetch some values
25 -- SELECT * FROM students ;
26 SELECT * FROM project ;
```

PID

SID

PNAME

1	3	IOT
2	1	AI
3	5	ML
4	2	WEB D
5	4	APP D

# Updates to Table:

- Insert:

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

- Delete:

```
DELETE FROM table_name WHERE condition;
```

- Alter table:

```
ALTER TABLE table_name  
ADD column_name datatype;
```

# SQL Arithmetic operators:

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide
%	Modulo

Operator	Description
&	Bitwise AND
	Bitwise OR

# SQL Arithmetic operators:

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to

# Order by clause

- The order by keyword is used to sort the result set in ascending or descending order.
- The order by keyword sorts the records in ascending order by default. To sort the records in descending order ,use DESC .

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

# Group by clause

- The **GROUP BY** statement groups rows that have the same values into summary rows, like "find the number of customers in each country".
- The **GROUP BY** statement is often used with aggregate functions (**COUNT()**,**MAX()**,**MIN()**,**SUM()**,**AVG()**) to group the result-set by one or more columns.

# Group by clause

```
SELECT column_name(s)
      FROM table_name
      WHERE condition
      GROUP BY column_name(s)
      ORDER BY column_name(s);
```

# Having Clause:

The Having clause was added to the SQL because WHERE keyword cannot be used with aggregate functions.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

# JOIN statement:

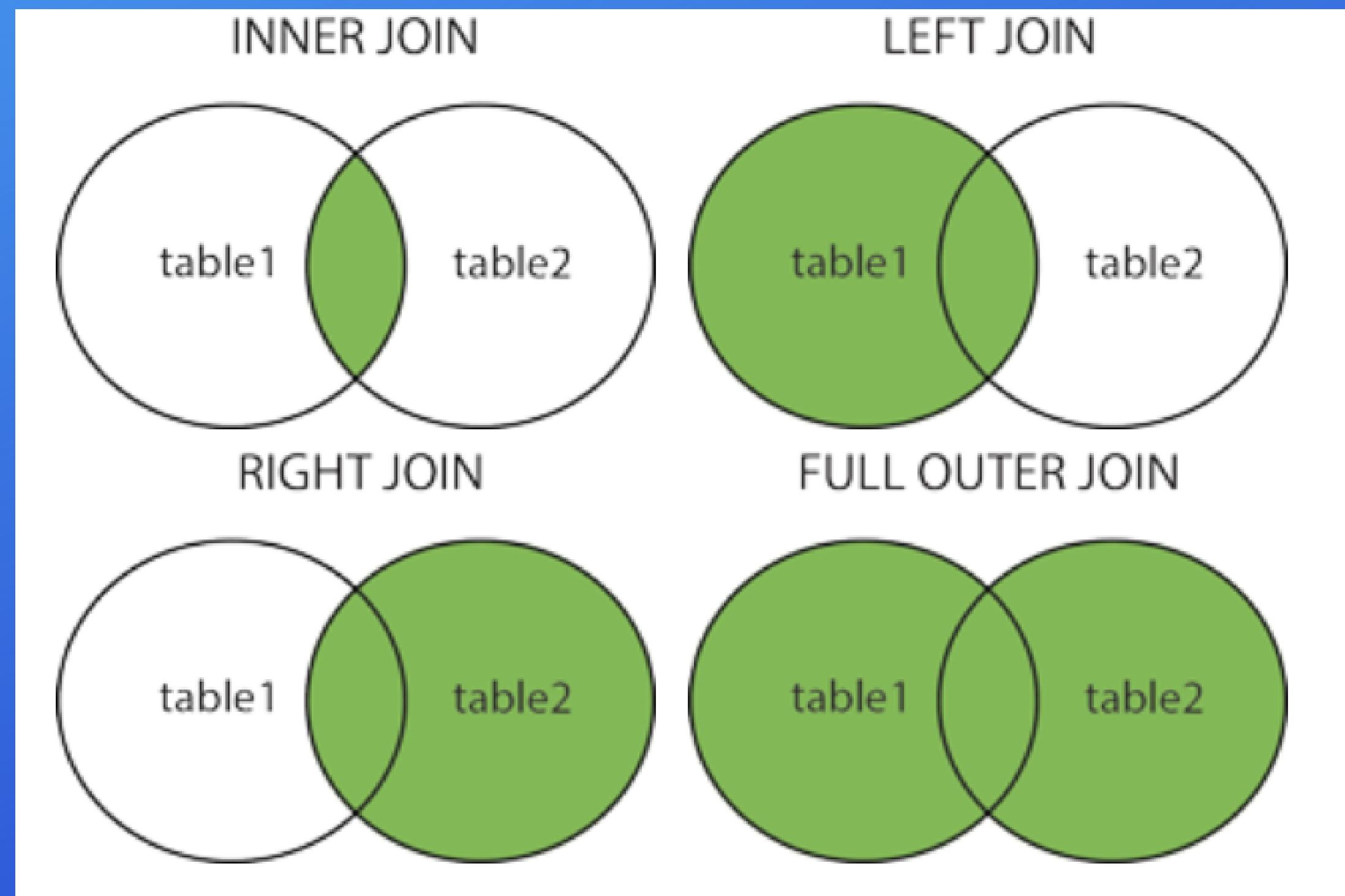
The JOIN clause is used to combine rows from two or more tables,based on a related column between them.

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

- Inner Join: Returns record that have matching values in both tables.
- Left Join: Returns all records from the left table, and the matched records from the right table
- Right Join: Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

# JOIN statement:

## Different types of JOIN statements:



# MAX() functions:

- The MAX() function returns the maximum value of the selected column.

```
SELECT MAX(column_name)  
FROM table_name  
WHERE condition;
```

# MIN() function:

The MIN() function returns the smallest value of the selected column.

```
SELECT MIN(column_name)  
FROM table_name  
WHERE condition;
```

# COUNT() functions:

The COUNT() function returns the number of rows that matches a specified criterion.

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```



# AVG() functions:

The AVG() function returns the average of a numeric column.

```
SELECT AVG(column_name)
FROM tableName
WHERE condition;
```

# SUM() functions:

The SUM() function returns the total sum of a numeric column.

```
SELECT SUM(column_name)  
FROM table_name  
WHERE condition;
```

# SELECT DISTINCT:

The SELECT DISTINCT returns only distinct values.

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

Thank  
you!

