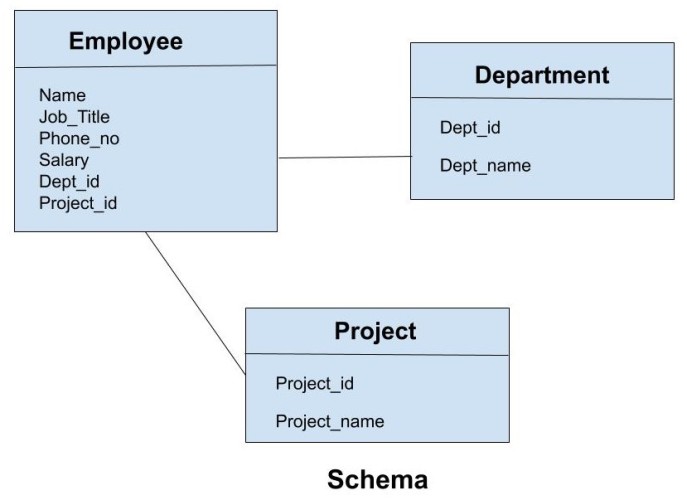
**Database Basics**

Database : A database is an organized collection of structured information, or data, typically stored electronically in a computer system.

Database Schema : A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated.

Database Scehma

**Database management system (DBMS)**

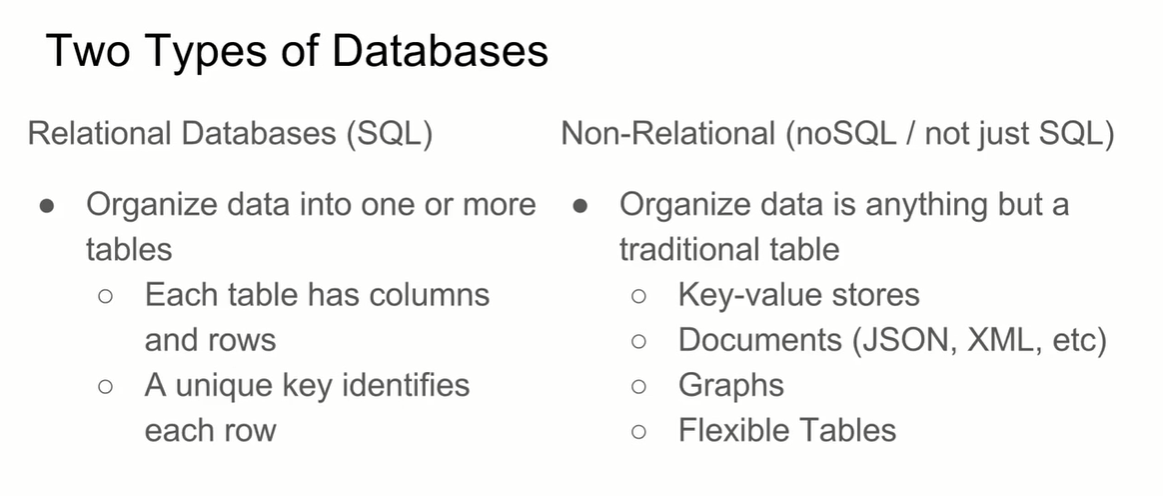
**Relational Database Management System (RDMS)**

**C.R.U.D**

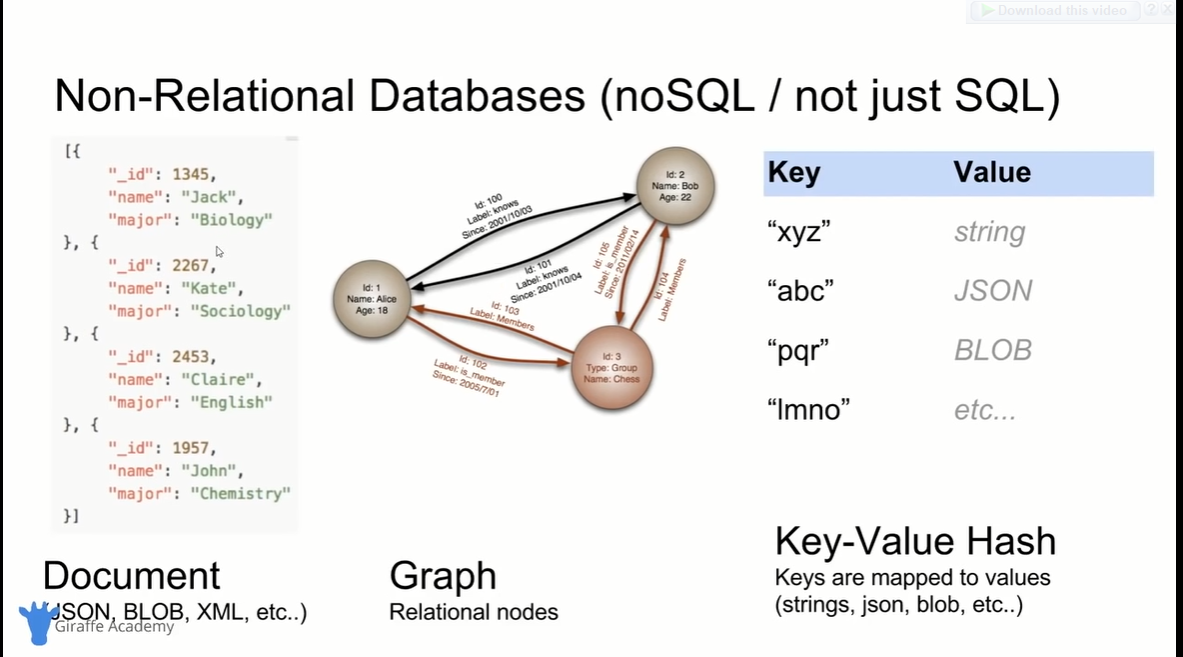
Create Read/Retrieve Update Delete

**Databases are Mainly two types**

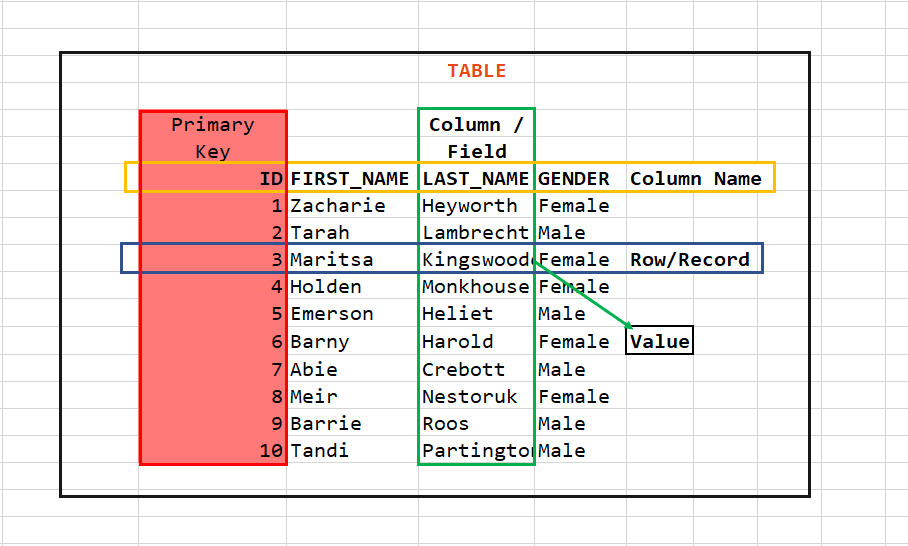
1. Relational Database (SQL) (mysql, mariadb, Oracle, postgresql)
2. Non Relational Database (No SQL) (Mongodb, firebase, dynamodb)



Non Relational Database



Database all that I need to know :



**Database Query** : Queries are the requests to the DBMS for specific information. (A google search is query)

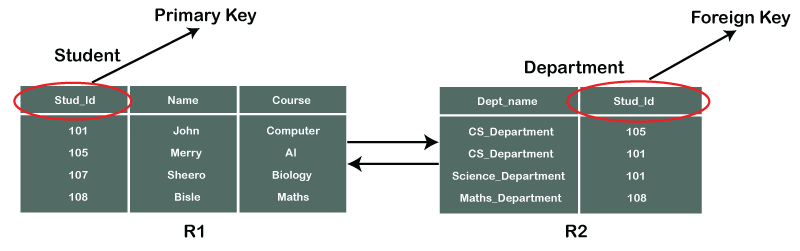
**Database Keys**

**Primary Key** : A primary key is the column or columns that contain values that uniquely identify each row in a table.

**Surrogate Key :** Surrogate Key is that has no mapping to anything in the real world. (id number)

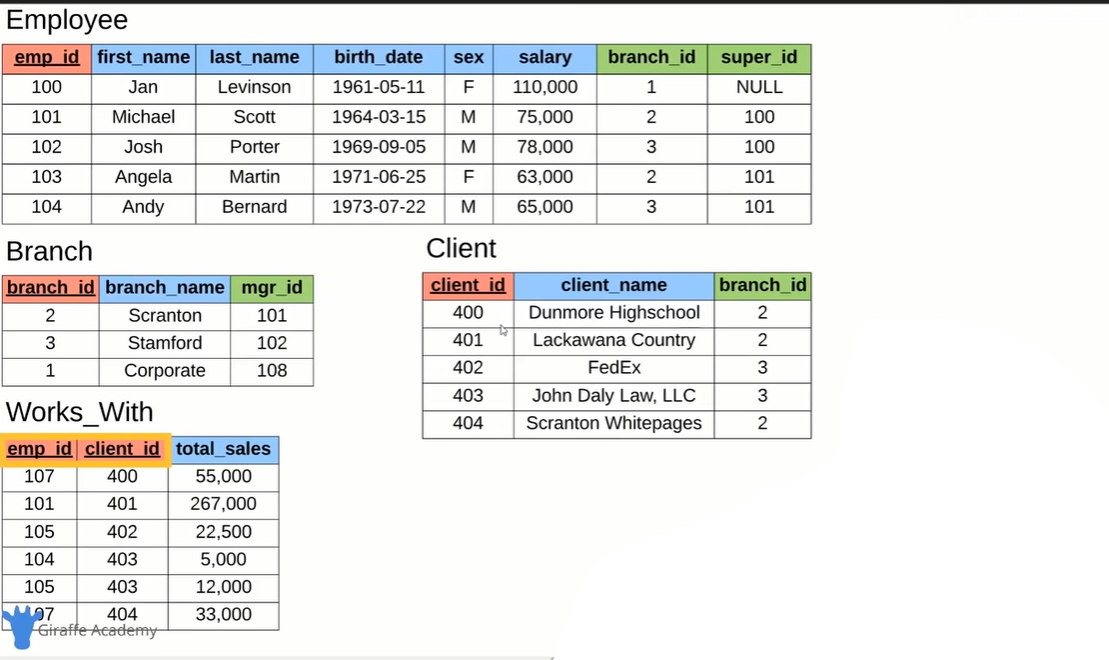
**Natural Key :** Surrogate Key is that has mapping to the real world entity. (Roll Number)

**\*Foreign Key :** A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables.



**Composite Key :** This is a key that need two or more attributes. That two or more attributes work like a primary key.

**All Kinds of Keys :**

****

**SQL(Structured Query Language)**

SQL : SQL is a language to operate databases. SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

* SQL can execute queries against a database
* SQL can retrieve data from a database
* SQL can insert records in a database
* SQL can update records in a database
* SQL can delete records from a database
* SQL can create new databases
* SQL can create new tables in a database
* SQL can create stored procedures in a database
* SQL can create views in a database
* SQL can set permissions on tables, procedures, and view

**There are 4 types of SQL**

Data Definition Language (DDL)

This includes CREATE (tables, views, objects, etc.), ALTER and DROP (delete).

Data Manipulation Language (DML)

SELECT, INSERT, UPDATE, DELETE of records within tables.

Data Control Language (DCL)

GRANT and/or REVOKE user privileges, etc.

Data Query Language (DQL)

Used to query the database information

**Query :** A query is a set of instruction given to the RDBMS that tell the RDBMS what information you want it to retrieve.

A query is a request for data or information from a database table or combination of tables. This data may be generated as results returned by Structured Query Language (SQL) or as pictorials, graphs or complex results, e.g., trend analyses from data-mining tools.

**Installing MYSQL**

Install mysql server

Install Mysql workbench (For Graphical use) Or POPSQL

**Connect mysql to cmd**

Cd “C:\Program Files\MySQL\MySQL Server 8.0\bin”

Mysql –u root -p

**Database commands**

Creating a database

Create Database “Database Name”

Show created databases

show databases;

Deleting database

Drop database “Database Name”

**Most used datatypes in SQL**

1. INT
2. DECIMAL(Total Number of digit, number of digit after decimal point)
3. VARCHAR(number of char)
4. BLOB[Binary large Object]
5. DATE
6. TIMESTAMP
7. TEXT

**Table Operations**

//creating table

CREATE TABLE student(

student\_id INT auto\_increment primary key,

name VARCHAR(10),

major VARCHAR(30) not null

);

// adding single column

ALTER TABLE student ADD gpa DECIMAL (3,2);

//delete single column

ALTER TABLE student DROP COLUMN gpa;

//adding multiple column

ALTER TABLE student ADD gpa DECIMAL(3,2), ADD roll int;

//deleting multiple column

AlTER TABLE student DROP COLUMN gpa, DROP COLUMN roll;

// describing table

DESCRIBE student;

//delete table

DROP TABLE student;

**Constraints**

NOT NULL; AUTO\_INCREMENT; UNIQUE; PRIMARY KEY;

DEFAULT “value”;

**Update and Delete**

UPDATE TableName

SET ColumnName = "Value", ColumnName = "Value"

WHERE ColumnName = Records value;

UPDATE student

SET name = "Workless", major = "Mechanical"

WHERE student\_id = 6;

DELETE FROM “”tableName”

Where ColumnName = “Values” AND ColumnName = Values;

**Basic Query**

SELECT ColumnName, ColumnName, …

FROM DatabaseName.TableName/TableName, TableName

WHERE ColumnName (=,<,>,=<,=>,!=,&&,||) Values

ORDER BY ColumnName DESC/ASC

LIMIT IntegerValue;

SELECT \*

FROM student

WHERE Depertment != "CMT" AND Depertment != "CSE"

ORDER BY Result DESC

LIMIT 5;

SELECT Distinct “ColumnName"

FROM “TableName”;

SELECT Distinct Branch\_id

FROM Employee;

SELECT ColumnName AS NewColumnName, ColumnName AS NewColumnName

FROM “TableName”;

SELECT FirstName AS ForeName, LastName AS SurName

FROM Employee;

**FOREIGN KEY**

FOREIGN KEY (ColumnName) REFERENCES TableName(ColumnName) ON DELETE SET NULL;

FOREIGN KEY (Mgr\_id) REFERENCES Employee(Emp\_id) ON DELETE SET NULL;

**FUNCTIONS**

# Counting the emp\_id number

SELECT COUNT(Emp\_id)

FROM Employee;

SELECT COUNT(Emp\_id)

FROM Employee

WHERE gender = "M";

SELECT AVG(Salary)

FROM Employee;

# Adding the salary of emlpoyee

SELECT SUM(Salary)

FROM Employee

WHERE gender = "F" AND Birth\_date > "1971-01-01";

SELECT COUNT(Gender) AS NumberOfGender, Gender AS GenderName

FROM Employee

GROUP BY Gender;

SELECT SUM(Total\_sales) AS Total\_Sale, Emp\_id

FROM Works\_with

GROUP BY Emp\_id;

**WILDCARDS**

**WILDCARDS** : A wildcard character is used to substitute one or more characters in a string.

Wildcard characters are used with the [LIKE](https://www.w3schools.com/sql/sql_like.asp) operator. The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

SELECT \*

FROM “TableName”

WHERE “ColumnName” LIKE ‘Sequence’;

There are 2 types of Sequence

1. % => Any numbers of Characters (%ex -> any char then **ex**)
2. \_ => One Character (-ex one char then **ex**)

SELECT \*

FROM Client

WHERE Client\_Name LIKE '%Ex';

SELECT \*

FROM Employee

WHERE Birth\_date LIKE '\_\_\_\_-02%';

**UNIONS**

The UNION operator is used to combine the result-set of two or more SELECT statements.

* Every SELECT statement within UNION must have the same number of columns
* The columns must also have similar data types
* The columns in every SELECT statement must also be in the same order

SELECT First\_Name

FROM Employee

UNION

SELECT Branch\_Name

FROM Branch;

**JOINS**

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

Here are the different types of the JOINs in SQL:

* (INNER) JOIN: Returns records that have matching values in both tables
* LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
* RIGHT (OUTER) 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

SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate

FROM Orders

INNER JOIN Customers

ON Orders.CustomerID=Customers.CustomerID;

-- Find the client Name, Branch Name, Branch Manager Name

SELECT client\_name, Branch.Branch\_Name, Employee.First\_Name AS Manager

FROM Client

JOIN Branch

ON Client.Branch\_id = Branch.Branch\_id

JOIN Employee

ON Branch.Mgr\_id = Employee.Emp\_id;

**NESTED QUERIES/ SUBQUERY**

SELECT Employee.First\_Name, Employee.Last\_name

FROM Employee

WHERE Employee.Emp\_id IN (

SELECT Emp\_id

FROM works\_with

WHERE Total\_sales > 30000

);

SELECT Client.Client\_Name

FROM client

WHERE Branch\_id = (

SELECT Branch.Branch\_id

FROM Branch

WHERE Branch.Mgr\_id = 102

);