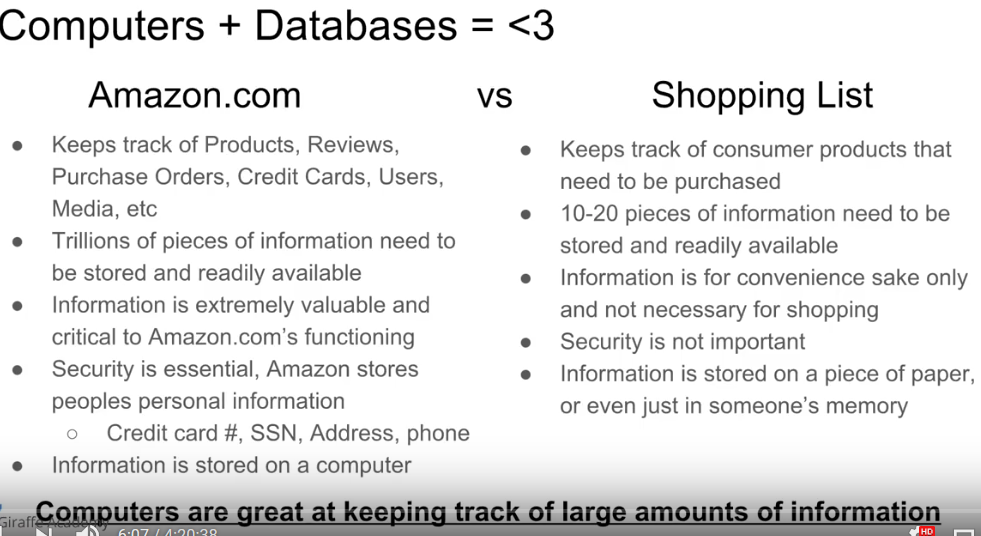
**What is database? Collection of releated information that can be stored in different ways**

Any collection of releated information

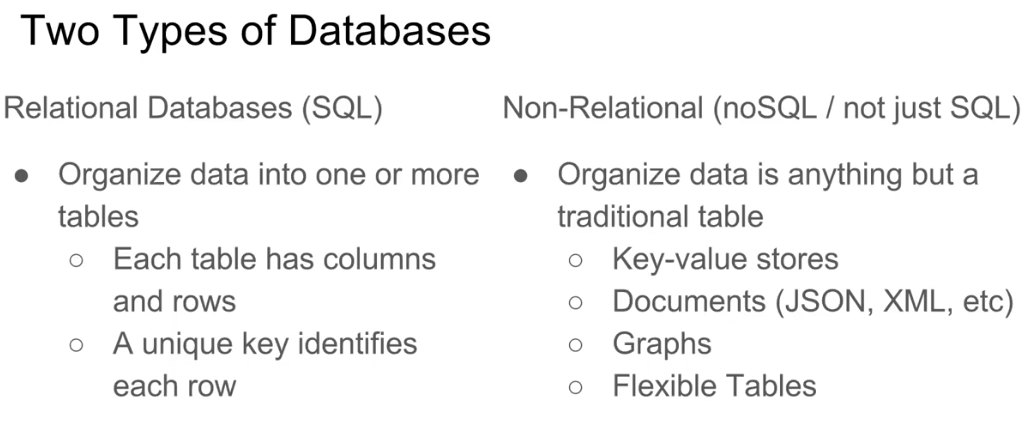
* Phone book
* Shopping list
* Todo list
* Your 5 best friends
* Fb user base

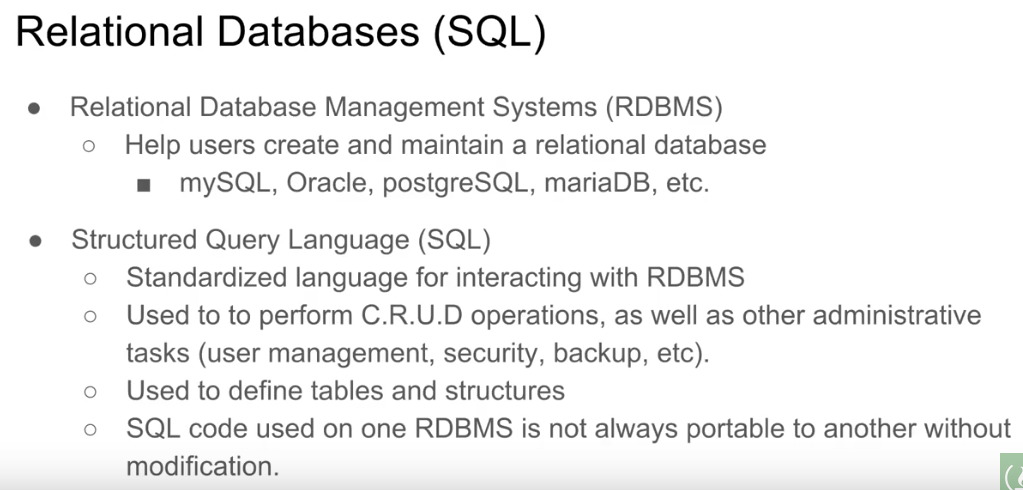
Databasses can be stored in different ways

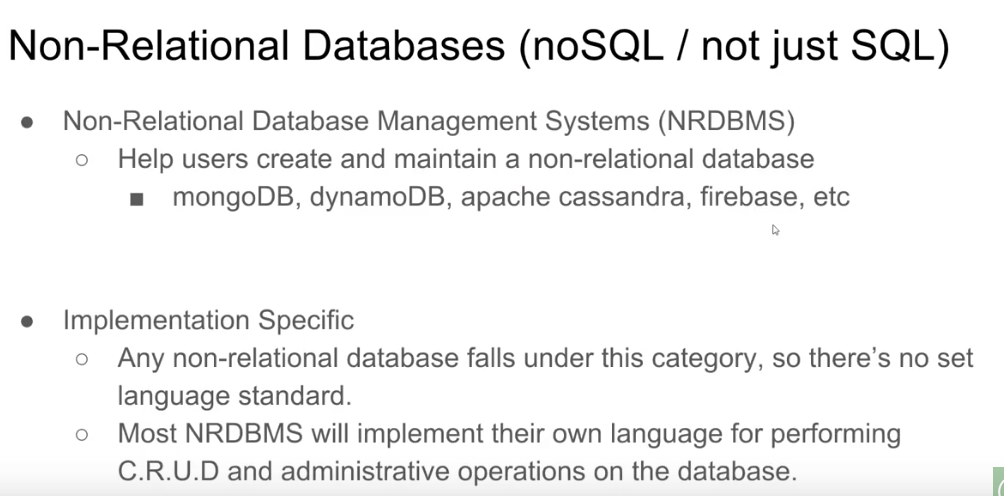
* On paper
* In your mind
* On a computer
* This powerpoint
* Comments section

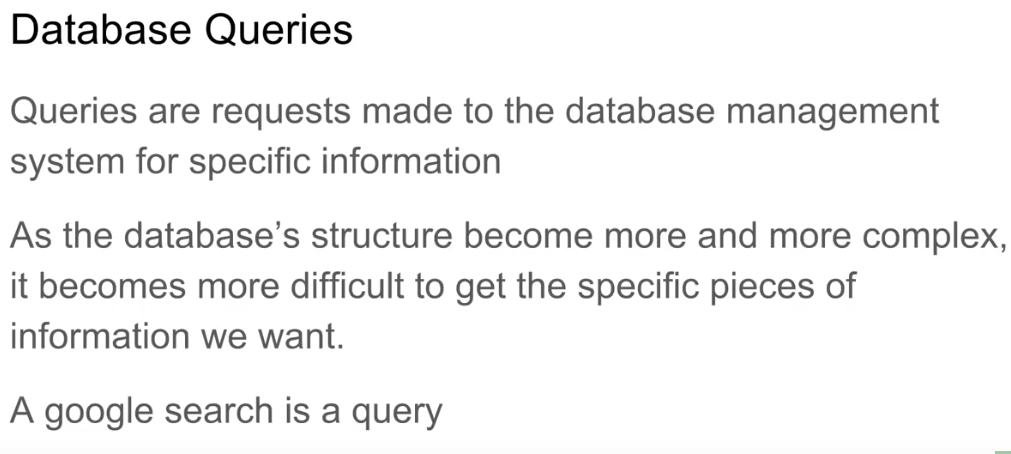


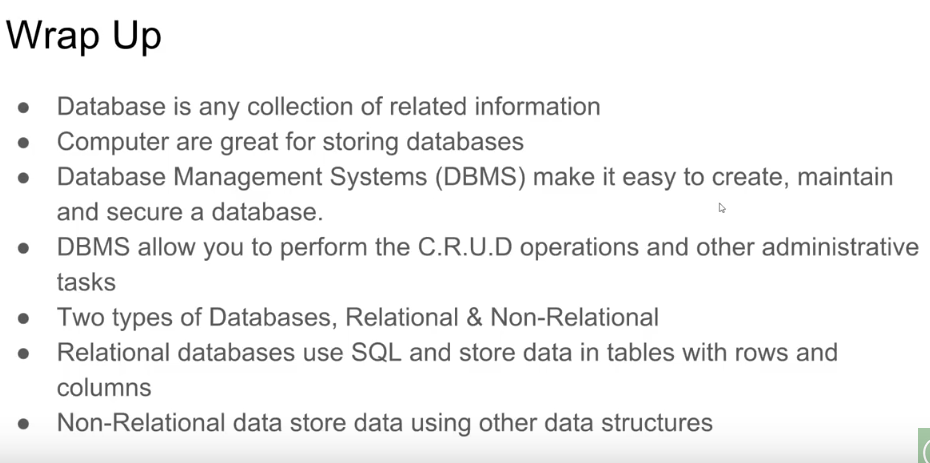












Query is a request that you make to a database

**Table and Keys**

Surrogate key –type of primary key assigned to people like emp\_id : 1001

Natural key — social security key that has a mapping in real world to people unlike surrogate key.

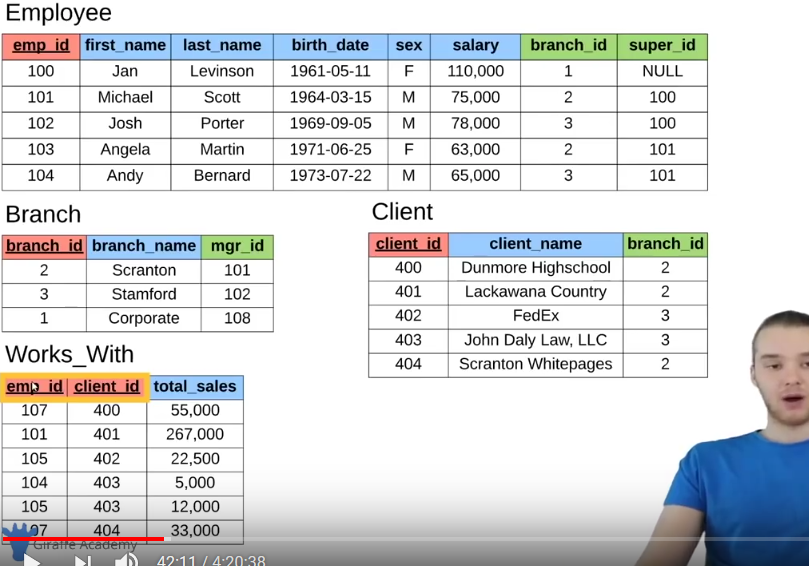
Foreign key – that will link to other table—it stores the primary key of other table

A table can have more than foreign key.

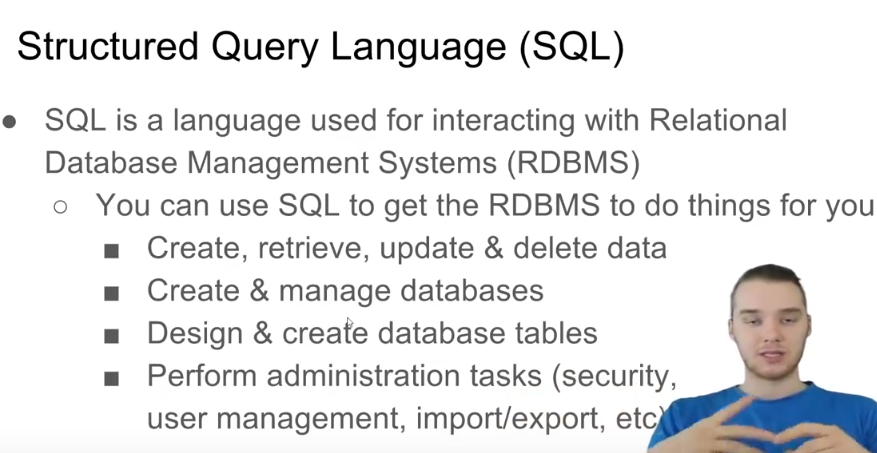


Composite keyb -> 2 column combined can also be primary key

Eg in works with table empid and clientid is the primary key

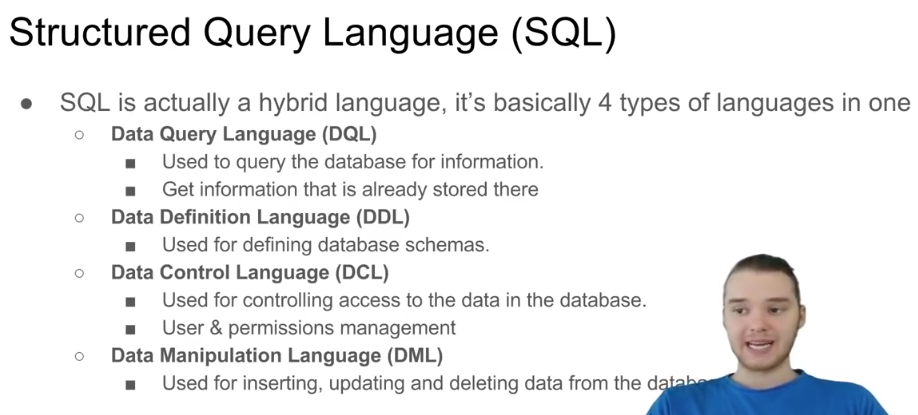


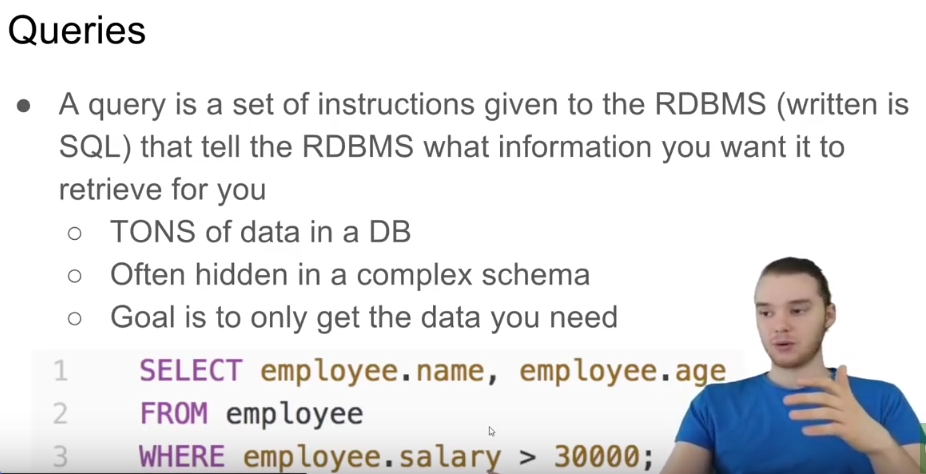
**SQL Basics**

****

SQL implementations vary between systems

* Not all RDMS follow the sql standard to a ‘T’
* The concepts are the same but the implementation may vary





**MySql installation in windows**

1 > Download sql

From <https://dev.mysql.com/downloads/file/?id=479861> and install it by going to custom install select

* install mysql server 8.0.12
* mysql shell 8.0.12

Set password

Do default installation then

2 > PopSql–text editor from *https://popsql.io/*

Now open mysql command line

Type password

Create a db -> create database giraffe;

New db is created and we can use this db in popSql to do operations.

**>Open popsql**

Define database schema to populate our db

**Different types of datatypes**

* INT ----Whole numbers
* DECIMAL(M,N) ----decimal numbers – exact value
* VARCHAR(l) ----string of text of length l
* BLOB ----Binary large objects, stores large data
* DATE ----‘yyyy-mm-dd’
* TIMESTAMP ----‘yyyy-mm-dd HH:MM:SS’ –used or recording time

examples

Decimal(10,4) = 987654.3219

Varchar(5) = max no of charcters in string would be 5

Blob structure that can store large file like image

======================================================================

**CREATING And MODIFYING table**

CREATE TABLE student (

student\_id INT PRIMARY KEY,

name VARCHAR(20),

major VARCHAR(20)

);

Or

CREATE TABLE student (

student\_id INT,

name VARCHAR(20),

major VARCHAR(20), PRIMARY KEY( student\_id )

);

DESCRIBE student;

DROP TABLE student;

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

ALTER TABLE student DROP COLUMN gpa;

**INSERTING DATA in TABLE**

INSERT INTO student VALUES( 1, ‘JACK’ ,’Biology’);

INSERT INTO student VALUES( 2, ‘KATE’ ,’Sociology’);

If we want to set only 2 values in the table we specify the column name in which our value go *student( student\_id, name).*

INSERT INTO student(student\_id, name) VALUES( 3, ‘CLAIRE’);

SELECT \* FROM student; //to print thetable

**CONSTRAINT**

* NOT NULL
* UNIQUE
* DEFAULT ‘undecided ’

Eg :-

CREATE TABLE student {

Student INT,

name VARCHAR(20) **NOT NULL**,

major VARCHAR(20) **UNIQUE** ***DEFAULT ‘undecided’***, PRIMARY KEY(student\_id )   
};

* Now if we want to have a unique primary key automatically increased also

CREATE TABLE student {

Student INT **AUTOINCREMENT**,

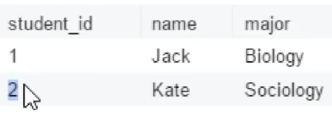
name VARCHAR(20),

major VARCHAR(20), PRIMARY KEY( student\_id )   
};

INSERT INTO student(name, major) VALUES(‘JOHN’,”MATHS”);

INSERT INTO student(name, major) VALUES(‘KATE’,”MATHS”);

Output :-



-==========================================================================

**UPDATE AND DELETE rows in a table**

#Comparison operators

* = equals
* <> not equals
* > greater than
* < less than
* >= greater than or equal
* <= less than or equal
* AND / OR

UPDATE student

SET major = ‘Bio’

**WHERE major = ‘Biology’; WHERE student\_id = 4;**

**WHERE major = ‘Bio’ or major =’Chemstry’;**

If WHERE is not specified this query will update all the 5 rows.

DELETE student

WHERE student\_id =1;

**BASIC QUERIES**

SELECT name,major

FROM student

ORDER BY name DESC; //by default **order by** display item in **asc**

SELECT name,major

FROM student

ORDER BY student\_id DESC

LIMIT 2;

SELECT \*

FROM student

WHERE major=’Biology’

**IN**

SELECT \*

FROM student

WHERE name IN (‘Claire’, ‘Kate’, ‘Jack’);

**#COMPANY DB INTRO**

TASK -> create a company database



CREATE TABLE employee (

emp\_id INT PRIMARY KEY,

first\_name VARCHAR(40),

last\_name VARCHAR(40),

birth\_day DATE,

sex VARCHAR(1),

salary INT,

super\_id INT,

branch\_id INT

);

CREATE TABLE branch (

branch\_id INT PRIMARY KEY,

branch\_name VARCHAR(40),

mgr\_id INT,

mgr\_start\_date DATE,

FOREIGN KEY(mgr\_id) REFERENCES employee(emp\_id) ON DELETE SET NULL

);

***Now we have declare the tables we have to alter the foreign key in the employee table***

ALTER TABLE employee

ADD FOREIGN KEY(branch\_id)

REFERENCES branch(branch\_id)

ON DELETE SET NULL;

ALTER TABLE employee

ADD FOREIGN KEY(super\_id)

REFERENCES employee(emp\_id)

ON DELETE SET NULL;

***While setting the foreign key we also need to specify the references from which table we tooked the foreign key .***

CREATE TABLE client (

client\_id INT PRIMARY KEY,

client\_name VARCHAR(40),

branch\_id INT,

FOREIGN KEY(branch\_id) REFERENCES branch(branch\_id) ON DELETE SET NULL

);

CREATE TABLE works\_with (

emp\_id INT,

client\_id INT,

total\_sales INT,

PRIMARY KEY(emp\_id, client\_id),

FOREIGN KEY(emp\_id) REFERENCES employee(emp\_id) ON DELETE CASCADE,

FOREIGN KEY(client\_id) REFERENCES client(client\_id) ON DELETE CASCADE

);

CREATE TABLE branch\_supplier (

branch\_id INT,

supplier\_name VARCHAR(40),

supply\_type VARCHAR(40),

PRIMARY KEY(branch\_id, supplier\_name),

FOREIGN KEY(branch\_id) REFERENCES branch(branch\_id) ON DELETE CASCADE

);

***Now inserting the data in table***

-- Corporate

INSERT INTO employee VALUES(100, 'David', 'Wallace', '1967-11-17', 'M', 250000, NULL, NULL);

INSERT INTO branch VALUES(1, 'Corporate', 100, '2006-02-09');

UPDATE employee

SET branch\_id = 1

WHERE emp\_id = 100;

INSERT INTO employee VALUES(101, 'Jan', 'Levinson', '1961-05-11', 'F', 110000, 100, 1);

-- Scranton

INSERT INTO employee VALUES(102, 'Michael', 'Scott', '1964-03-15', 'M', 75000, 100, NULL);

INSERT INTO branch VALUES(2, 'Scranton', 102, '1992-04-06');

UPDATE employee

SET branch\_id = 2

WHERE emp\_id = 102;

INSERT INTO employee VALUES(103, 'Angela', 'Martin', '1971-06-25', 'F', 63000, 102, 2);

INSERT INTO employee VALUES(104, 'Kelly', 'Kapoor', '1980-02-05', 'F', 55000, 102, 2);

INSERT INTO employee VALUES(105, 'Stanley', 'Hudson', '1958-02-19', 'M', 69000, 102, 2);

-- Stamford

INSERT INTO employee VALUES(106, 'Josh', 'Porter', '1969-09-05', 'M', 78000, 100, NULL);

INSERT INTO branch VALUES(3, 'Stamford', 106, '1998-02-13');

UPDATE employee

SET branch\_id = 3

WHERE emp\_id = 106;

INSERT INTO employee VALUES(107, 'Andy', 'Bernard', '1973-07-22', 'M', 65000, 106, 3);

INSERT INTO employee VALUES(108, 'Jim', 'Halpert', '1978-10-01', 'M', 71000, 106, 3);

-- BRANCH SUPPLIER

INSERT INTO branch\_supplier VALUES(2, 'Hammer Mill', 'Paper');

INSERT INTO branch\_supplier VALUES(2, 'Uni-ball', 'Writing Utensils');

INSERT INTO branch\_supplier VALUES(3, 'Patriot Paper', 'Paper');

INSERT INTO branch\_supplier VALUES(2, 'J.T. Forms & Labels', 'Custom Forms');

INSERT INTO branch\_supplier VALUES(3, 'Uni-ball', 'Writing Utensils');

INSERT INTO branch\_supplier VALUES(3, 'Hammer Mill', 'Paper');

INSERT INTO branch\_supplier VALUES(3, 'Stamford Lables', 'Custom Forms');

-- CLIENT

INSERT INTO client VALUES(400, 'Dunmore Highschool', 2);

INSERT INTO client VALUES(401, 'Lackawana Country', 2);

INSERT INTO client VALUES(402, 'FedEx', 3);

INSERT INTO client VALUES(403, 'John Daly Law, LLC', 3);

INSERT INTO client VALUES(404, 'Scranton Whitepages', 2);

INSERT INTO client VALUES(405, 'Times Newspaper', 3);

INSERT INTO client VALUES(406, 'FedEx', 2);

-- WORKS\_WITH

INSERT INTO works\_with VALUES(105, 400, 55000);

INSERT INTO works\_with VALUES(102, 401, 267000);

INSERT INTO works\_with VALUES(108, 402, 22500);

INSERT INTO works\_with VALUES(107, 403, 5000);

INSERT INTO works\_with VALUES(108, 403, 12000);

INSERT INTO works\_with VALUES(105, 404, 33000);

INSERT INTO works\_with VALUES(107, 405, 26000);

INSERT INTO works\_with VALUES(102, 406, 15000);

INSERT INTO works\_with VALUES(105, 406, 130000);

==================================================================================

**# MORE BASIC QUERIES**

Select \* from employee;

Select \* from clients;

SELECT \* FROM employee ORDER BY salary;

SELECT \* FROM employee ORDER BY sex, first\_name, last\_name;

SELECT \* FROM employee ORDER BY emp\_id LIMIT 5;

SELECT first\_name, last\_name FROM employee;

* **TO DISPLAY THE SELECTED QUERY WITH DIFFERENT COLUMN NAME WE USE** AS **KEYWORD.**

SELECT first\_name AS forename, last\_name AS surname FROM employee;

* **Find out all the different genders.**

SELECT DISTINCT sex FROM employee;

* **Find the number of employees**

SELECT COUNT(emp\_id) FROM employee;

* Find the number of females born after 1970

SELECT COUNT(emp\_id) FROM employee WHERE SEX= ‘F’ AND birth\_date > ‘1970-01-01’;