RELATIONAL DB- MYSQL, ORACLE (data stored in tables)

Non relational- mongo db

Sql- structured query language.

Rational db understand sql

Crud operation- create, read, update, delete

—---------------------------------------------------------------------------------------

To create db- CREATE DATABASE bd\_name; OR create database temp2;

To drop db- DROP DATABASE db\_name;

CREATE DATABASE college;

USE college; // use the db

// Creating table:

//CREATE TABLE table\_name(column\_name data\_type constraint)

CREATE TABLE student(

Id INT PRIMARY KEY,

name VARCAR(50),

age INT NOT NULL);

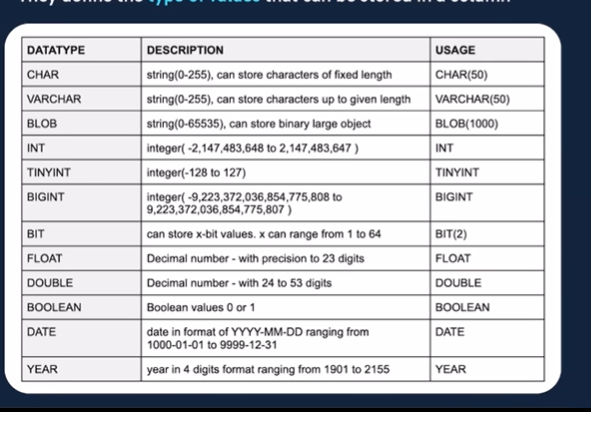
// insert data in table, PRIMARY KEY -UNIQUE

INSERT INTO student VALUES(1, “ZAIMA”, 26)

// PRINT FILL TABLE

SELECT \* FROM student;

// datatypes



Varcar- takes needed space, char- reserves the fixed space

BIT(1)- 0,1 BIT(2)-00, 01,10,11

BOOLEAN needs TINYINT-0 FLASE, 1 TRUE

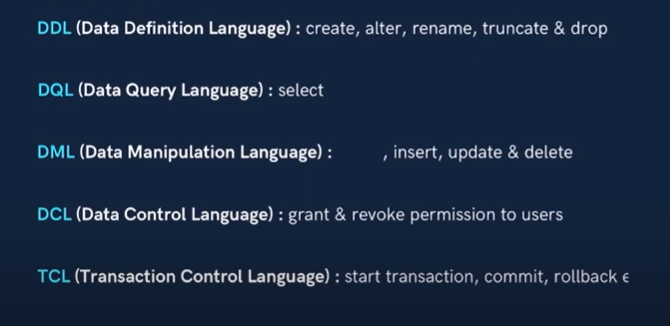
// SIGNED AND UNSIGNED

SIGNED- BOTH NEG AND POS NUMERIC

Unsigned- age, salary(positive values)

TINYINT UNSIGNED (0 TO 255)

// SQL COMMAND

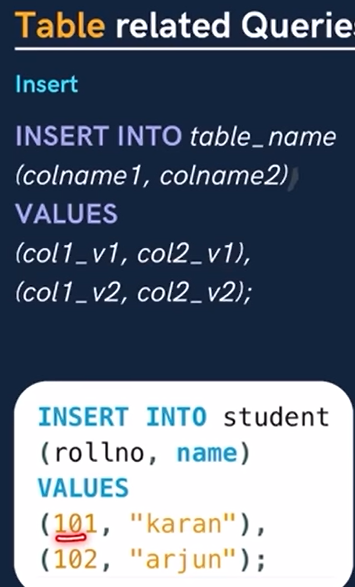


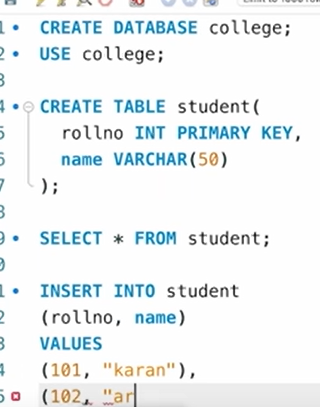
CREATE DATABASE OF NOT EXIST db\_name; // if db not already present

DROP DATABASE IF EXISTS db\_name; /// only drop if the db exist

SHOW DATABASES; //shows all db of server

SHOW TABLE; // only shows the tables from USE DATABASE



 // SELECT \* FROM STUNET PRINTS THE TABLE

Primary key(pk)- unique for a row(1 pk unique and not null)

Foreign key- a col/ set of col that refers to a primary key of diff table(multiple fk)



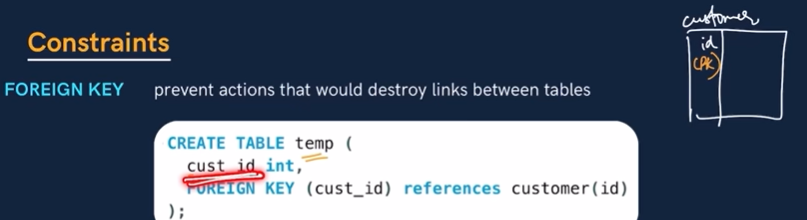
Constraints- NOT NULL, UNIQUE, PRIMARY KEY

Col1 int NOT NULL

Col2 int UNIQUE

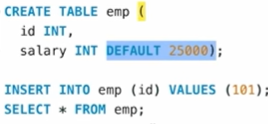
Id int PRIMARY KEY //primary key is unique or null

CREATE TABLE temp1( id INT, name VARCAR(50), age INT, city VARCAR(20), PRIMARY KEY(id, name)); // name id combo will be unique



Default constraint

Salary INT DEFAULT 2500



CREATE TABLE city(

Id INT PRIMARY KEY, city VARCHAR(50), age INT, CONSTRAINT age\_check CHECK (age>=18 and city=’Delhi’));

////////////////////////////////////////////////////////// practice /////////////////////////////////////////////////////////

CREATE DATABASE college;

USE college;

CREATE TABLE student(rollno INT PRIMARY KEY, name VARCHAR(50), marks INT NOT NULL, grade VARCHAR(1), city VARCHAR(20));

INSERT INTO student( rollno, name, marks, grade, city) VALUES ( 101, ‘ami’, 78, ‘c’, ‘dhaka’),

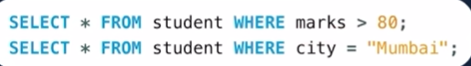
(102, ‘pixi’, 90, ‘a’, ‘ctg’));

SELECT name, marks FROM students

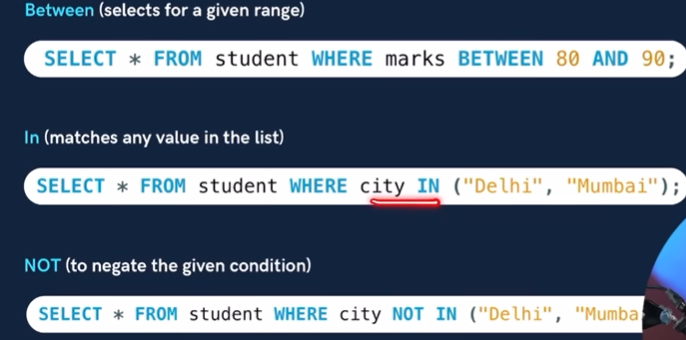
SELECT DISTINCT city FROM student; // all unique cities

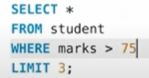
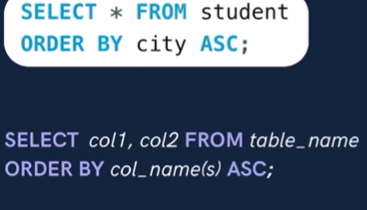
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

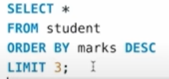
///////\*\*\*\*\*\*\*\*\*\*\* WHERE Clause (for conditionaning for rows)







Top 3 marks 

Aggregate functions- count(), max(), min(), sum(), avg()

\*\*\*

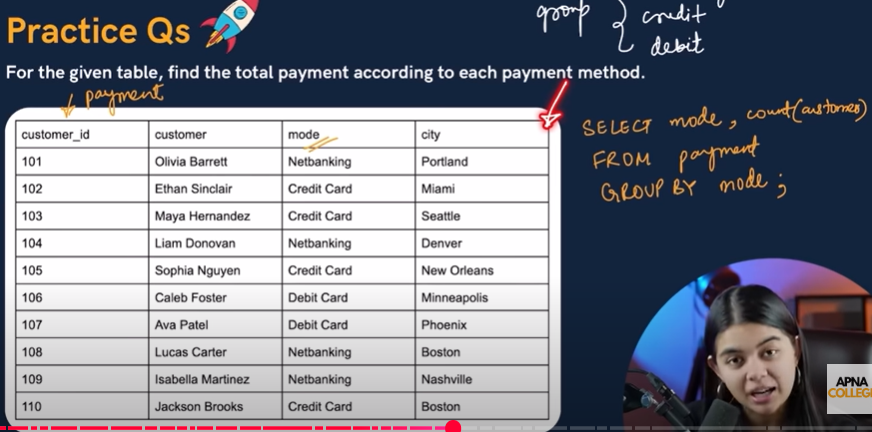
SELECT max(marks) FROM student;

///////////////////////////////////////// group by ///////////////////////////////////

SELECT city, count(name) FROM student GROUP BY city;

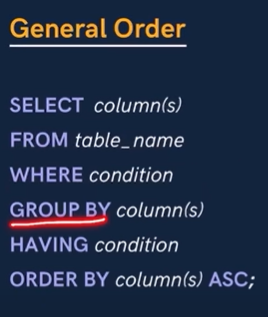
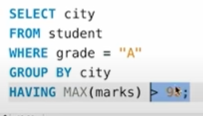
**SELECT city, name count(rollno) FROM student GROUP BY city, name;**

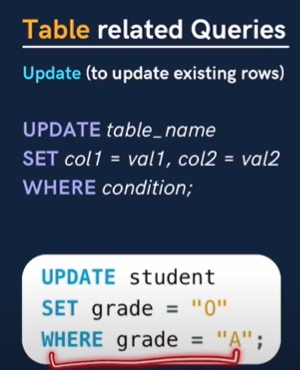
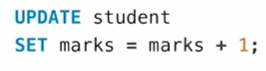
SELECT city, avg(marks) from student group by city order by avg(marks);

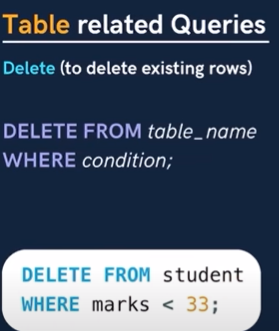


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* having clause- conditions on groups\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

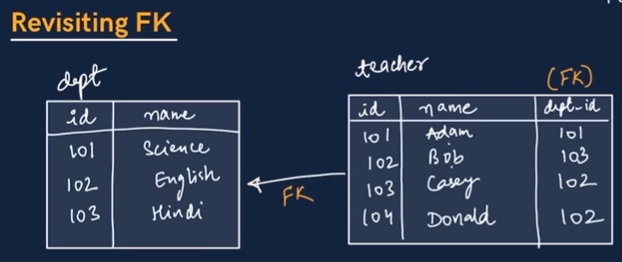
SELECT city, count(rollno) From student group by city HAVING max(marks>90);

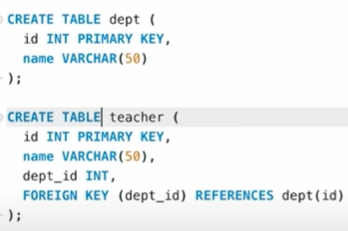
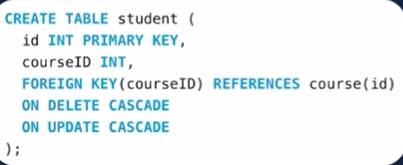
  grade A in all cols but 93 only on group by

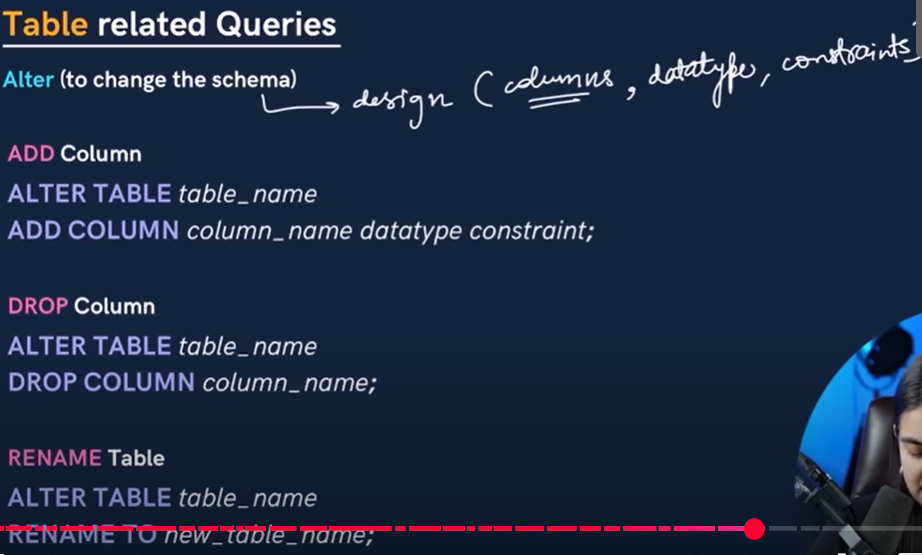
  update all marks by 1

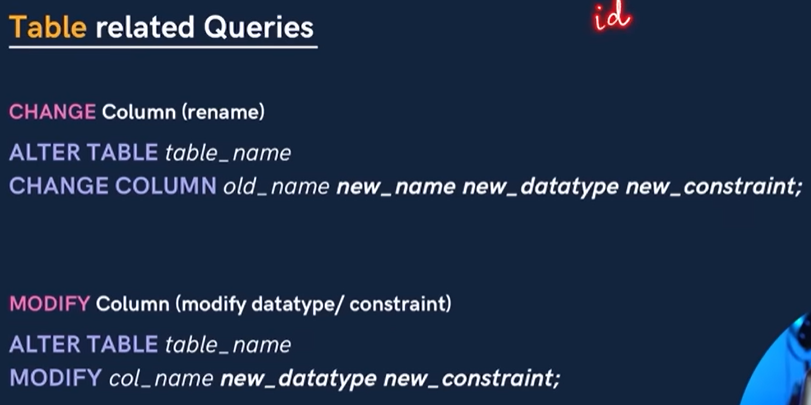
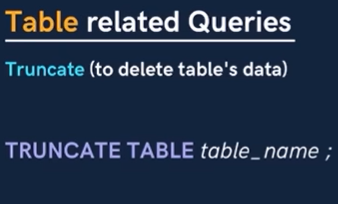


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* FK \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

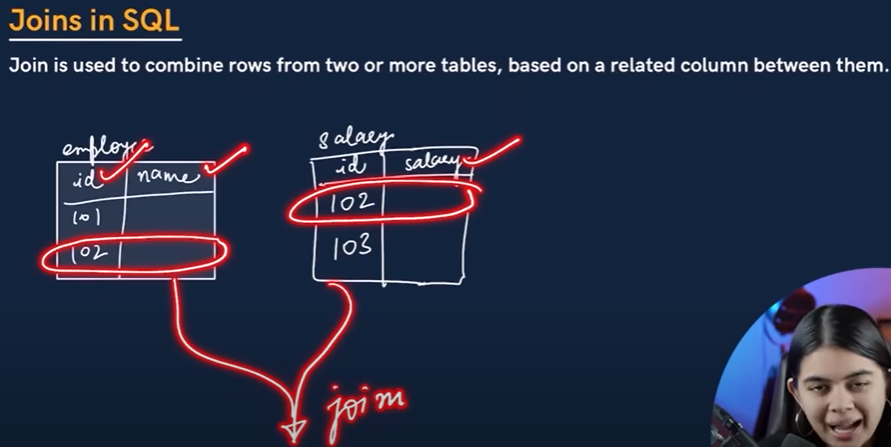


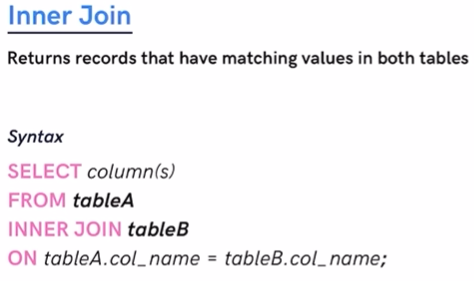
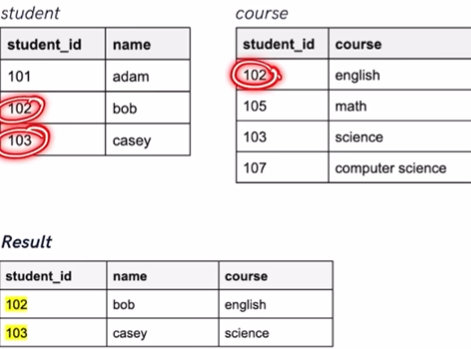
 

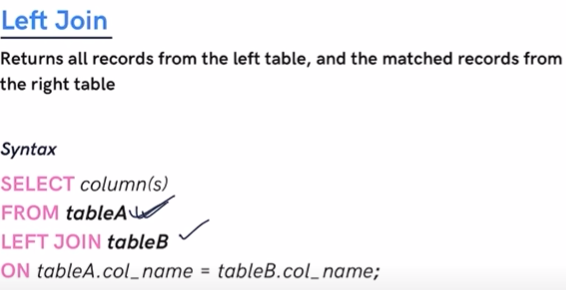
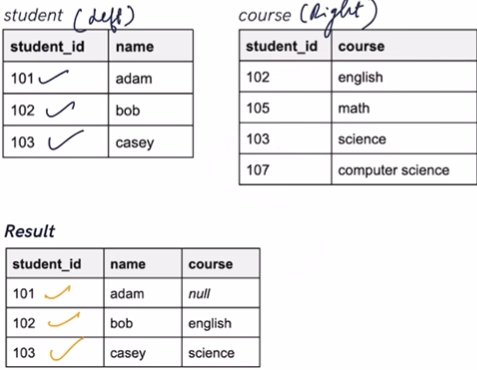


   Delete all the data of table (not the table)

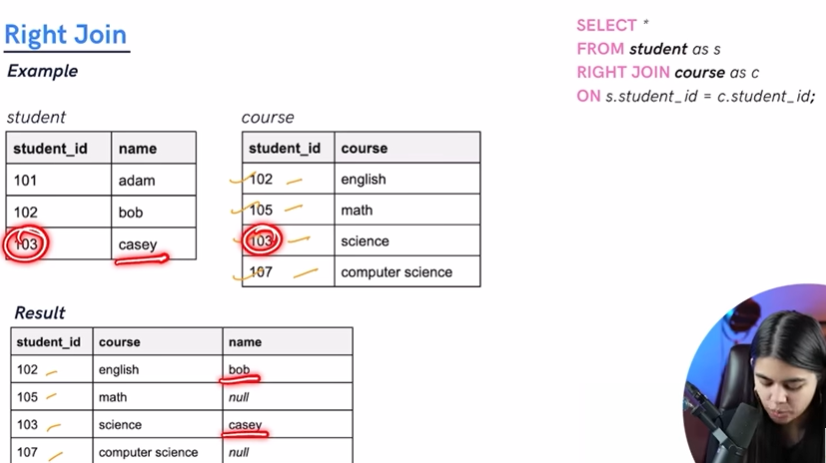
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* joins in sql \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



Same for right



Full join

