**DR. B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY, JALANDHAR (144011), PUNJAB**

****

ITPC-227

FUNDAMENTALS OF DATABASE MANAGEMENT SYSTEM LAB FILE

**SUBMITTED BY: SAHIB SINGH**

**B.Tech IT 3rd Semester**

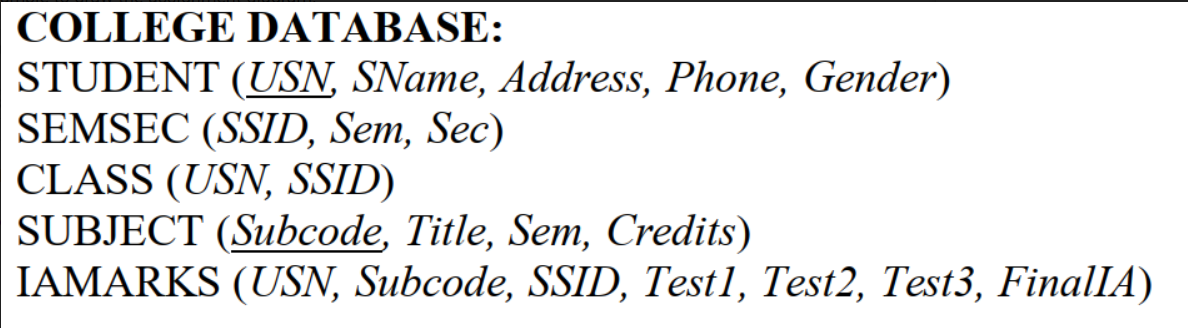
**20124086**

# INDEX

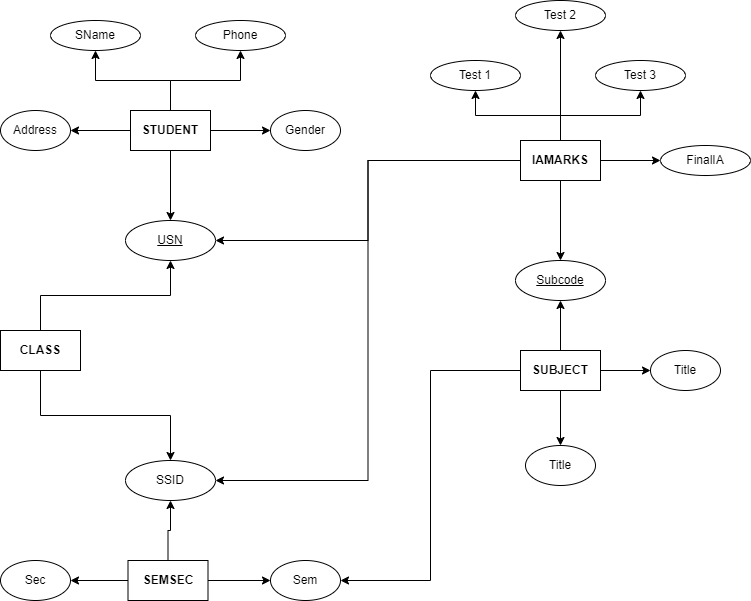
| S. No. | Programme Title | Date of Implementation | Remarks |
| --- | --- | --- | --- |
| 1 | [ASSIGNMENT 1: ER Diagram](#_3w6zkcz7urio) | 03-09-2021 |  |
| 2 | [ASSIGNMENT 2: ER Diagram](#_ed0spg7ejt80) | 10-09-2021 |  |
| 3 | [ASSIGNMENT 3: Create Database and use different variable data types](#_fxvzb94prq7t) | 20-09-2021 |  |
| 4 | [ASSIGNMENT 4: Use SELECT Statement with variations](#_tk7el9qjtp7a) | 27-11-2021 |  |
| 5 | [ASSIGNMENT 5: Implement UPDATE, DELETE, LIMIT Statements, MIX, MAX, COUNT, AVG, SUM Functions and LIKE Statement with Wildcards](#_nwsn65tvqmb2) | 28-11-2021 |  |
| 6 | [ASSIGNMENT 6: Implement IN, BETWEEN, ALIASES, GROUP BY, HAVING and EXISTS.](#_xc7sp1jjjzbf) | 28-11-2021 |  |
| 7 | [ASSIGNMENT 7: Perform JOIN Operations and UNION Clause](#_vkff3qjyvpyq) | 28-10-2021 |  |
| 8 | [ASSIGNMENT 8: Apply Primary key, Foreign Key, Unique Key, NOT NULL, Check, Default, ALTER TABLE, DROP TABLE Concepts](#_xnm5qjnwlmhy) | 09-11-2021 |  |

# ASSIGNMENT 1

## Consider the following database (attached) and draw ER diagram for it.

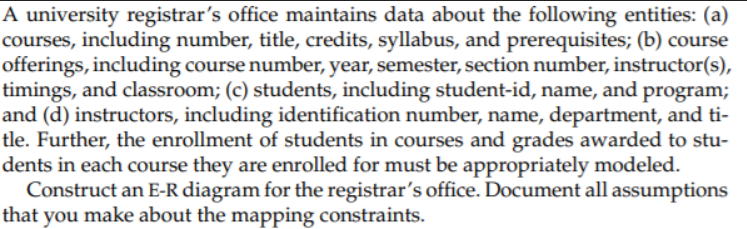


## Solution:

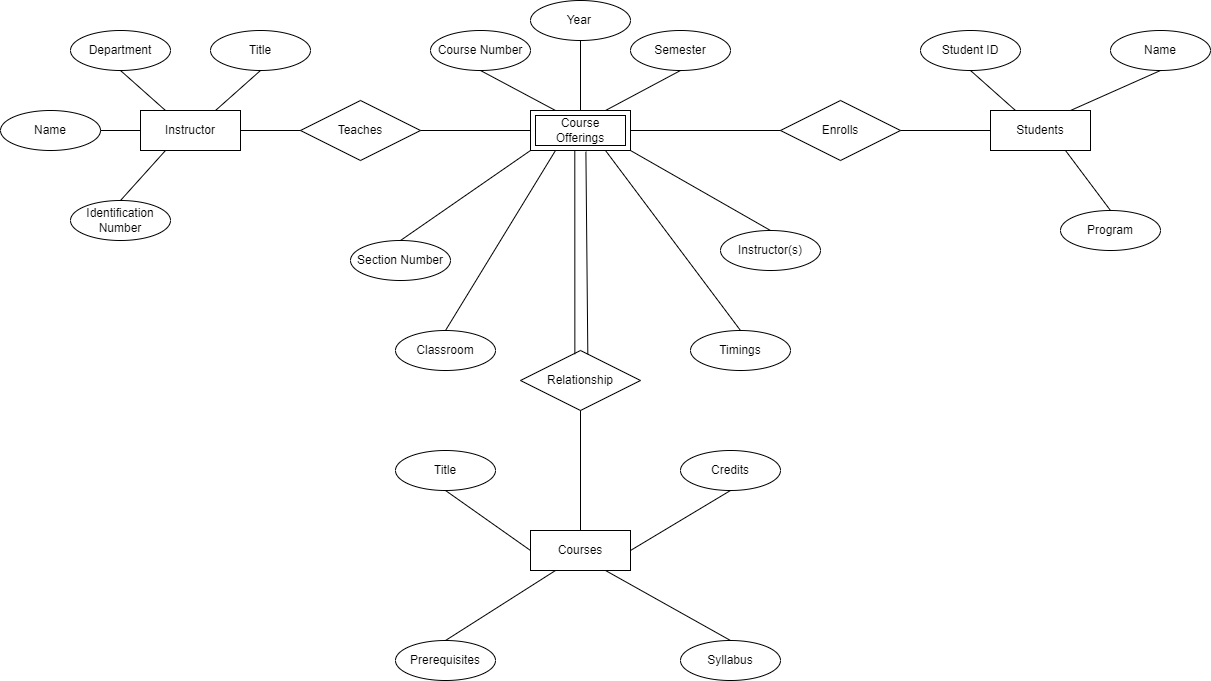
****

# ASSIGNMENT 2

## Draw ER for the following:



## Solution:

****

FOR BETTER VIEW: [Click Here](https://viewer.diagrams.net/?tags=%7B%7D&highlight=0000ff&edit=_blank&layers=1&nav=1&title=Lab%202.drawio#R3Vxbc5s4FP41mdl9qAcQ4vLYXNrtzLbNNJ1psm%2FEVmxmubgCant%2F%2FYoABo5IwBgkpXkJyCDg49M53zk64gJdhfuP1NtuPscrElwY2mp%2Fga4vDEM3bIf9y1sORYul20XDmvqr8qC64c7%2Fj5SNWtma%2BSuStA5M4zhI%2FW27cRlHEVmmrTaP0njXPuwpDtpX3XprwjXcLb2Ab%2F3hr9JN0epgrW7%2Fi%2FjrTXVlXSt%2FCb3q4LIh2XireNdoQjcX6IrGcVpshfsrEuTgVbgU53144dfjjVESpUNO2D8t76Po67f43vvHv3lMnZvd5buyl19ekJUP%2FClKUpot05iWt50eKix2Gz8ld1tvme%2Fv2Pu%2BQJebNAzYns42vcBfR2x7yW6IUNbA32F1OUJTsm80lXf8kcQhSemBHVL9apfolfTRK%2BB39cs4Qr5pvAizbPPK978%2Bdl1DxDZKlE5ADHOIffFCwmHFemIUZTuXElBDqoFmcaBdE%2FaEaZg%2Fo1LQHXEpsbNlQ2dz0H33UwaVYQXs0pePbJha6%2FT48KrgeAREFRwd3tKt2OP5T%2F7SS%2F04Yr99ycJHwps9pfh4BEgakJVnbiDJQxat3ufOl%2B1FcUTaMNE4i1Ykv4TG9sjeT%2B%2FLX%2FLth7x9gcu9633jsOtDtROxB7lv7jTOynfr0573qvNefB9JnNEl6bf6qUfXJO33p2TVUhX826UkYKz71RYZXa%2BqPPU29nNLeWQFBhbeBW%2B7eKDyrKYo6OnI0EBHxRNzHT0z5%2Fg8Z5CJFyBnkelIjIWBW9zoYUbFwpp5Dw1OdrNwPJssxdhk4jab0Eg2GcBY6ZZgNqFZTFNFpkGsaFDwJAaOZ5M%2BkE2OIDZBjwVNymDbBGTsMZISxSZe6k9km2yFbZOtlm0yHGBS9JFsQi8FRaLYxAv5K3brTFeqKTxNSzEBr%2FMK%2Fo6EJElVQ85SDjmXQ%2B6BeLQKIZOtF7Xgs35meVKqwO5dUoD3PoeFkvrXOu5kcga5xVZQtHwo%2BlQxMMXg7UiPpww%2BnqpzcH8kf6oFnw0DBunsNvgQ4o4sFQ7pEXBpRxcnD0KDg%2FC7H%2FrROlELOdtoI4fkk2%2BuiGNUwHFazPviC%2BmPODpE4vefj78Ol%2FZB32UPd5%2Bjf%2BnfmStKJGIdBLD2SJGIHdARJM7MItEwVaKTKDa5irEJkgDGnUPZZOmaXDbNFsCOjF%2FtiQPY11jSG8BWYMxNJws4rdH5EMto8xLBNN3cdOJn8WTSSVeJS5ogLkH1PTYbYmttLhmCsyEGnw2R6OhOy8ed4ekctTydPZVuslDPDMLcdOJzQ1I9nUqmSZJqGu3moJjngre5ucRny64CL0loHIc8qaSmaKEgkB4OIz6XJXUcnjJxJ2AouoJUwlQzcNCsI%2BgfZh6KqCO1l2Z52Q6fmJJSmwOkvfxCO8Rn8tQrT9R11YrDEJ%2FGu6XxmnqKWXxdc8HQls84PmVVjlHW%2BOlaLfxcrBrxJo6p30b%2BuOJMr8dEgsQr5MXocjpdf72juR0mr13PolNNDWec%2FjqRhreE%2BgyDfLifSzFDNYoBn2eOLbLTYYEV19PMJDNVE%2FlTp5Vf5Uk%2FoZAYQjH1aS%2BAPodZvMGc0lwD9iW42s7ktX5RH6WE1IdVI1i6cjB5pV8uDFFJcMEUhelKx42X%2BleUrPyOkFJurQNQEgog16H0D0HgPWaKQ2dJL7Ax%2BZnYW0oo%2BZn5CYNHMfzgoFUAv7nCJKUL%2B82hkqOi1%2Bw5flBDbY3VsJbb09HcYkOl6ccWmWZcw1Yphn46iZoyAnTCYwtt4JQRdgTTaeLpx3PopAujE1KMThaIsPFo6wQ7Ek2nibM4nXTSBlonUByhz8enoUlBUXyyoZMaPY3W19HMfKqyAs1QkXjLTYfuTDbeNt%2BkmzjMdT263DZScmVrI0snpX4eruTqWsJhdMhTuBp1um%2BCTLyK%2BU0siR8qTiv2zb5S0IIlbe6iKlQ%2FebEgiCANwfoU88md359QFVFUqY5DeCp5Cg2W4DIKzKe8biIaB8HbtP%2FOS2X18uz%2FTMs2xgxXYfp%2F8IQIFjQh4oBvheHxsyEwMBVcHY0nXrjxJviEO%2BJJqcXRcJlmVTJ%2FsvyHnzAQHE5iPtn17RmaOEo2%2FvZN%2BgAMfDPqmh0R6wPmygGZJmqNW9fpyQSxHVhH0QjnTYzaI1qzUM%2BYHl%2BacVa5bMXb2fNGUE%2BMrf7BIM7npuPnHugi8kaqTZLgoR%2BsEZaGhOv94KzY8PV%2BYAZ4vrTRa%2BO064M1X5%2BemDno%2Bj5A5TzIPv%2F0R8ju0M9tf25iVnH2%2BHyYLslngBoOodXanQjP9vEAFpA0RuxCM9zBo9bSJ3EPXV7oN3IZmrPAbuOvPeI1faFbWv3njDQAr17F1PDCbFwEPMRo48B2649SF4fXn%2FZGN%2F8D)

# ASSIGNMENT 3

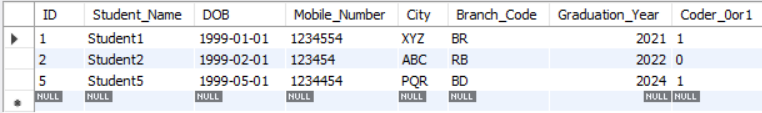
## Creating A Database:

| CREATE DATABASE Students; |
| --- |

## Creating a Table, Inserting values in it and Finally Selecting all Data from Database:

| USE Students; CREATE TABLE student ( ID INT NOT NULL PRIMARY KEY, Student\_Name VARCHAR(30), DOB DATE, Mobile\_Number MEDIUMINT(11), City TEXT(10), Branch\_Code CHAR(3), Graduation\_Year YEAR, Coder\_0or1 BOOL ); INSERT INTO student VALUES (1 , 'Student1', '1999-01-01', 1234554, 'XYZ', 'BR', 2021 , 1); INSERT INTO student VALUES (2 ,'Student2', '1999-02-01', 123454, 'ABC', 'RB', 2022 , 0); INSERT INTO student VALUES (5 ,'Student5', '1999-05-01', 1234454, 'PQR', 'BD', 2024 , 1); SELECT \* FROM student; |
| --- |

## Result:



# ASSIGNMENT 4

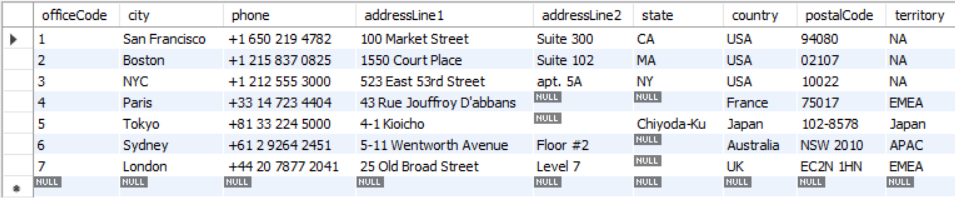
## DATABASE USED

| USE classicmodels; |
| --- |

## \* Statement

| SELECT \* FROM offices; |
| --- |

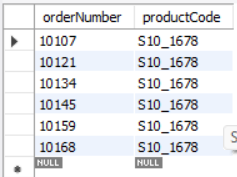
## RESULT:



## 'Attribute'

| SELECT orderNumber, productCode FROM orderdetails LIMIT 6; |
| --- |

## RESULT:

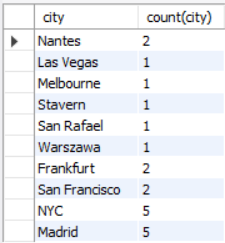


## DISTINCT

| SELECT DISTINCT city , count(city)  FROM customers  GROUP BY city LIMIT 10; |
| --- |

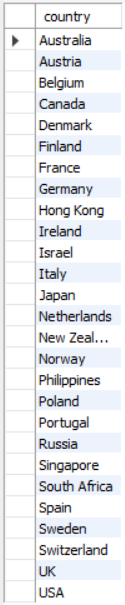
## 

## RESULT:



| SELECT DISTINCT country FROM customers  ORDER BY country; |
| --- |

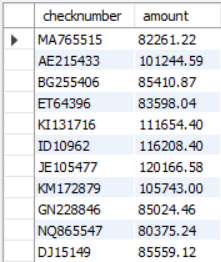
## RESULT:



## WHERE Clause

| SELECT checknumber, amount  FROM payments  WHERE amount > 80000.00 LIMIT 10; |
| --- |

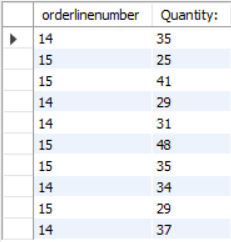
## RESULT:



## OR Operation

| SELECT orderlinenumber, (quantityordered) "Quantity:"  FROM orderdetails  WHERE orderlinenumber = 15 OR orderlineNumber = 14 LIMIT 10; |
| --- |

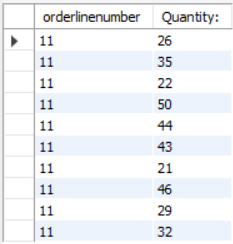
## RESULT:



## AND Operation

| SELECT orderlinenumber, (quantityordered) "Quantity:"  FROM orderdetails  WHERE orderlinenumber < 12 AND orderlineNumber > 10 LIMIT 10; |
| --- |

## RESULT:



## NOT Operation

| SELECT customerName, city  FROM customers  WHERE city NOT IN ('norway' , 'nantes' , 'frankfurt') LIMIT 10; |
| --- |

## RESULT:

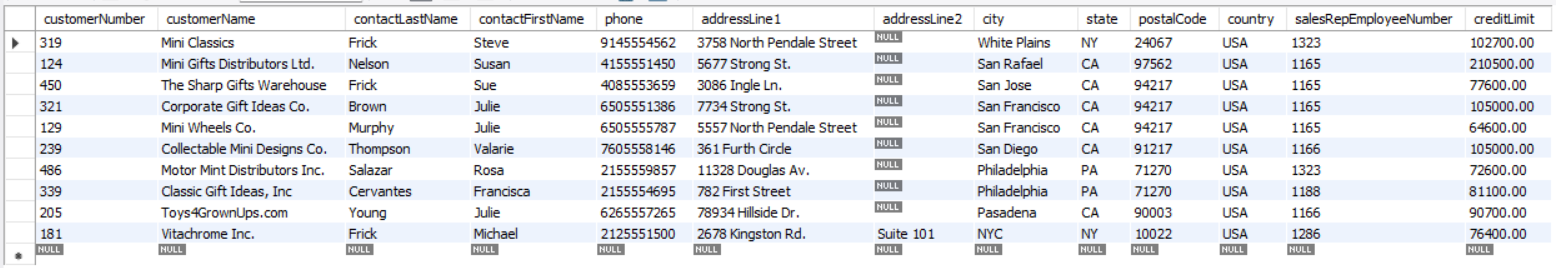


## 

## ORDER BY

| SELECT \* FROM customers  WHERE country = 'USA'  ORDER BY CITY DESC; |
| --- |

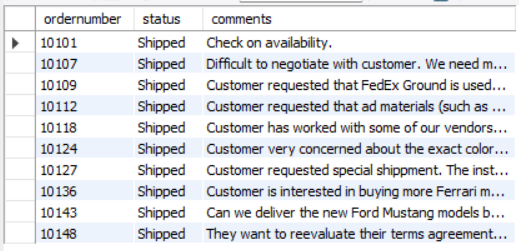
## RESULT:



## NULL Values

| SELECT ordernumber ,status ,comments  FROM orders  WHERE comments IS NOT NULL AND STATUS = 'shipped' LIMIT 10; |
| --- |

## RESULT:



| SELECT \* FROM orders  WHERE shippeddate IS NULL LIMIT 10; |
| --- |

## 

## RESULT:

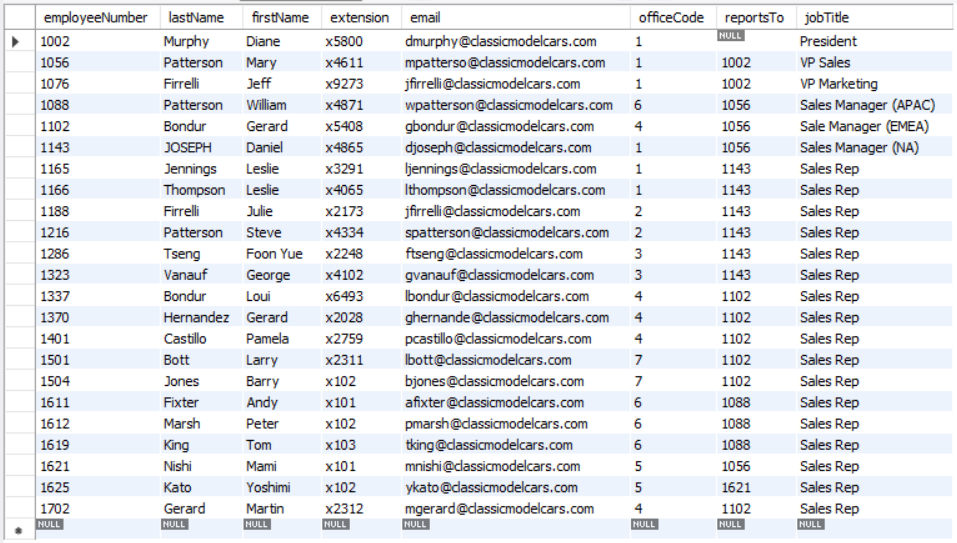
## 

# ASSIGNMENT 5

## UPDATE STATEMENT

| UPDATE employees  SET lastName = 'JOSEPH'  WHERE employeeNumber = 1143;  SELECT \* FROM employees; |
| --- |

## RESULT:

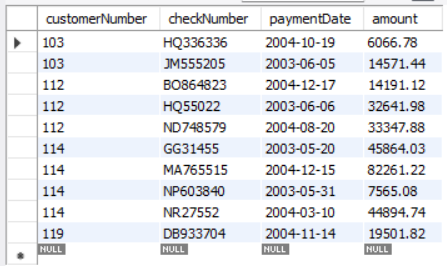


## DELETE STATEMENT

| DELETE FROM payments  WHERE customerNumber IN (103, 141) AND AMOUNT < 5000; SELECT \* FROM payments LIMIT 10; |
| --- |

## 

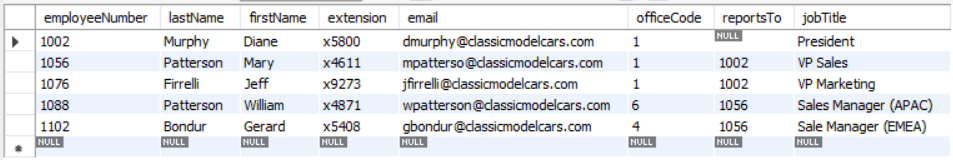
## RESULT:



## LIMIT

| SELECT \* FROM employees  LIMIT 5; |
| --- |

## RESULT:



## MIN() AND MAX() FUNCTIONS

| SELECT MAX(amount) , MIN(amount) FROM payments; |
| --- |

## RESULT:



## 

## COUNT(), SUM(), AVG() FUNCTIONS

| SELECT  SUM(amount) AS "Total Amount", COUNT(customerNumber) AS "Total Customers", AVG(amount) AS "Average Amount Per Customer"  FROM payments; |
| --- |

## RESULT:



## LIKE

| SELECT firstName, lastName FROM employees WHERE firstName LIKE 's%'; SELECT firstName, lastName FROM employees WHERE firstName LIKE '%e'; SELECT firstName, lastName FROM employees WHERE firstName LIKE 's\_%'; SELECT firstName, lastName FROM employees WHERE firstName LIKE '% %'; |
| --- |

## RESULT:



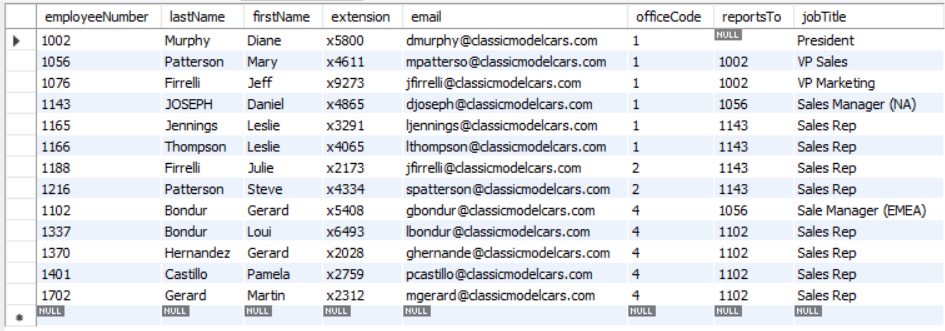
## 

# ASSIGNMENT 6

## IN Operator

| SELECT \* FROM employees WHERE officeCode IN (1, 2, 4) ORDER BY officeCode; |
| --- |

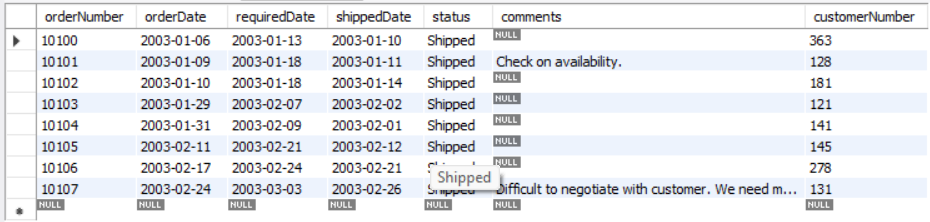
## RESULT:



## BETWEEN Operator

| SELECT \* FROM orders WHERE orderDate BETWEEN '2003-01-01' AND '2003-02-28' ORDER BY orderDate; |
| --- |

## RESULT:

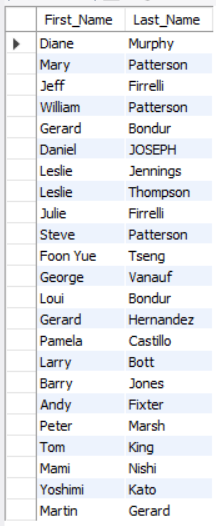


## 

## ALIASES

| SELECT firstName AS First\_Name, lastName AS Last\_Name  FROM employees; |
| --- |

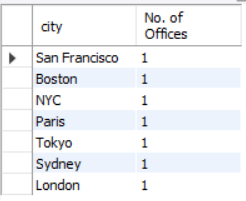
## RESULT:



## GROUP BY Statement

| SELECT city, COUNT(city) AS 'No. of Offices' FROM classicmodels.offices GROUP BY city; |
| --- |

## RESULT:

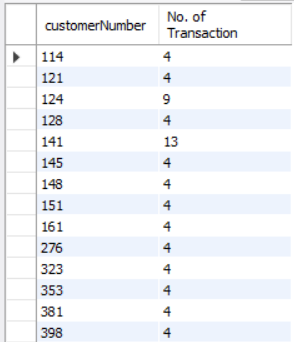


## 

## HAVING Clause

| SELECT customerNumber, COUNT(customerNumber) AS 'No. of Transaction' FROM payments GROUP BY customerNumber HAVING COUNT(customerNumber) > 3 ORDER BY customerNumber; |
| --- |

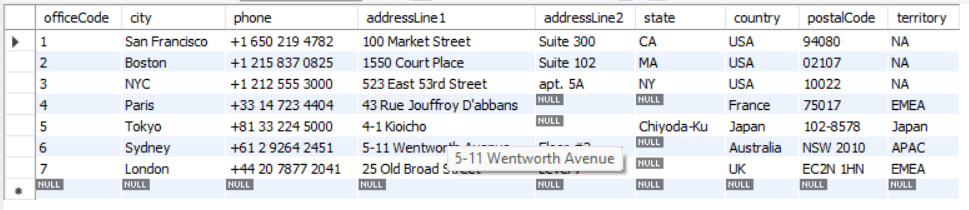
## RESULT:



## EXISTS

| SELECT \* FROM offices WHERE EXISTS (SELECT \* FROM employees WHERE officeCode IN (1, 2, 4) ORDER BY officeCode); |
| --- |

## RESULT:



# ASSIGNMENT 7

## DATABASE USED:

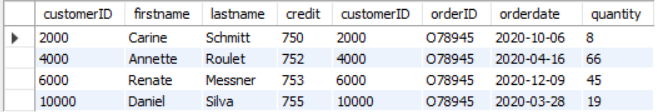
| CREATE DATABASE shop;  USE shop;  CREATE TABLE store( customerID int , firstname varchar(20), lastname varchar(30), credit int   );  CREATE TABLE order\_details( customerID int , orderID varchar(25) , orderdate date, quantity int );   INSERT INTO store VALUES(2000,"Carine","Schmitt",750); INSERT INTO store VALUES(3000,"Helen","Bennett",751); INSERT INTO store VALUES(4000,"Annette","Roulet",752); INSERT INTO store VALUES(6000,"Renate","Messner",753); INSERT INTO store VALUES(8000,"Paolo","Accorti",754); INSERT INTO store VALUES(10000,"Daniel","Silva",755); INSERT INTO store VALUES(11000,"Giovanni","Rovelli",756); INSERT INTO store VALUES(12000,"Adrian","Huxley",757);    INSERT INTO order\_details VALUES(2000 ,"O78945",'2020-10-06',8 ); INSERT INTO order\_details VALUES(4000 ,"O78945",'2020-04-16',66 ); INSERT INTO order\_details VALUES(6000 ,"O78945",'2020-12-09',45 ); INSERT INTO order\_details VALUES(10000 ,"O78945",'2020-03-28',19 ); INSERT INTO order\_details VALUES(23000 ,"O78945",'2020-06-13',12 ); |
| --- |

## JOINS:

## INNER JOIN:

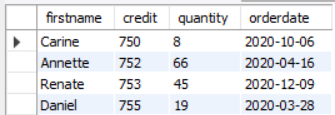
| USE shop;  SELECT \* FROM store  INNER JOIN order\_details ON store.customerID = order\_details.customerID; |
| --- |

## RESULT:



| USE shop;  SELECT   store.firstname,  store.credit,  order\_details.quantity,  order\_details.orderdate  FROM order\_details INNER JOIN store ON order\_details.customerID=store.customerID ; |
| --- |

## RESULT:

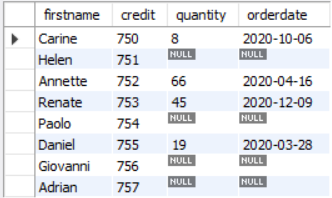


## 

## LEFT JOIN:

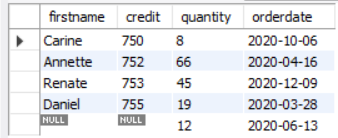
| USE shop;  SELECT  store.firstname,  store.credit,  order\_details.quantity,  order\_details.orderdate FROM store LEFT JOIN order\_details ON store.customerID=order\_details.customerID ; |
| --- |

## RESULT:



| USE shop;  SELECT  store.firstname,  store.credit,  order\_details.quantity,  order\_details.orderdate FROM order\_details LEFT JOIN store ON order\_details.customerID=store.customerID ; |
| --- |

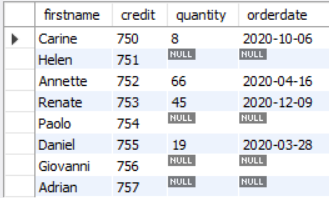
## RESULT:



## RIGHT JOIN:

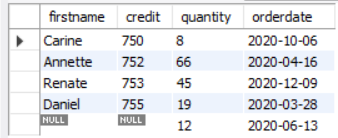
| USE shop;  SELECT   store.firstname,  store.credit,  order\_details.quantity,  order\_details.orderdate FROM order\_details RIGHT JOIN store ON order\_details.customerID=store.customerID ; |
| --- |

## RESULT:



| USE shop;  SELECT  store.firstname,  store.credit,  order\_details.quantity,  order\_details.orderdate FROM store RIGHT JOIN order\_details ON store.customerID=order\_details.customerID ; |
| --- |

## RESULT:

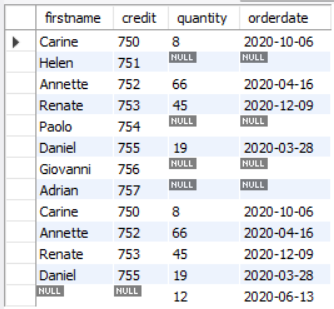


## OUTER JOIN:

MySQL does not have any keyword for OUTER JOIN. So for OUTER JOIN we can use UNION of LEFT and RIGHT JOIN.

| USE shop;  SELECT   store.firstname,  store.credit,  order\_details.quantity,  order\_details.orderdate  FROM store LEFT JOIN order\_details ON order\_details.customerID=store.customerID  UNION ALL SELECT   store.firstname,  store.credit,  order\_details.quantity,  order\_details.orderdate  FROM store RIGHT JOIN order\_details ON order\_details.customerID=store.customerID ; |
| --- |

## RESULT:

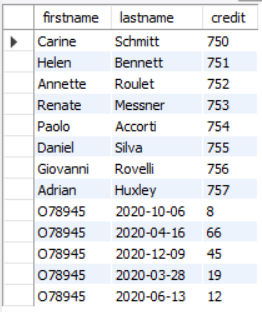


## 

## UNION:

| USE shop;  SELECT firstname, lastname , credit FROM store  UNION SELECT orderid, orderdate , quantity FROM order\_details; |
| --- |

## RESULT:



# ASSIGNMENT 8

## Create Database:

| CREATE DATABASE Departments; |
| --- |

## Database Backup:

We can not take Backup of Databases in MySQL with Queries.

For other Database Service there is a way that is given below:

| BACKUP DATABASE databasename TO DISK = 'filepath'; |
| --- |

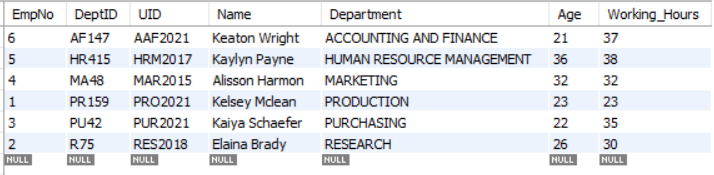
Changing Above Query for our Database:

| BACKUP DATABASE Departments TO DISK = 'C:\Storage\1. NITJ\LECTURES\LABS\FDBMS\Assignment 8'; |
| --- |

## Create Table and Apply Primary, Foreign, Unique Key, Not Null, Check, Default, Index Concepts:

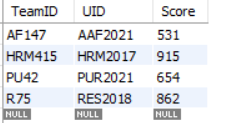
| CREATE TABLE Employee(  EmpNo INT NOT NULL UNIQUE AUTO\_INCREMENT,  DeptID VARCHAR(5) UNIQUE NOT NULL,  UID VARCHAR(10) PRIMARY KEY ,  Name VARCHAR(35),  Department VARCHAR(35) UNIQUE,  Age INT CHECK ( Age > 18 AND Age < 40 ),  Working\_Hours INT DEFAULT '15' );   INSERT INTO Employee (DeptID, UID, Name, Department, Age, Working\_Hours) VALUES("PR159", "PRO2021", "Kelsey Mclean" ,"PRODUCTION", 23, 23);  INSERT INTO Employee (DeptID, UID, Name, Department, Age, Working\_Hours) VALUES("R75", "RES2018", "Elaina Brady", "RESEARCH", 26, 30);  INSERT INTO Employee (DeptID, UID, Name, Department, Age, Working\_Hours) VALUES("PU42", "PUR2021", "Kaiya Schaefer", "PURCHASING" , 22, 35);  INSERT INTO Employee (DeptID, UID, Name, Department, Age, Working\_Hours) VALUES("MA48", "MAR2015", "Alisson Harmon", "MARKETING" , 32, 32);  INSERT INTO Employee (DeptID, UID, Name, Department, Age, Working\_Hours) VALUES("HR415", "HRM2017", "Kaylyn Payne", "HUMAN RESOURCE MANAGEMENT", 36, 38);  INSERT INTO Employee (DeptID, UID, Name, Department, Age, Working\_Hours) VALUES("AF147", "AAF2021", "Keaton Wright", "ACCOUNTING AND FINANCE", 21, 37 );   SELECT \* FROM Employee; |
| --- |

## RESULT:



| CREATE TABLE Employee\_Scores(  TeamID VARCHAR(10) PRIMARY KEY ,  UID VARCHAR(10),  Score VARCHAR(5) NOT NULL, FOREIGN KEY (UID) REFERENCES Employee(UID));   INSERT INTO Employee\_Scores (TeamID, UID, Score) VALUES("R75", "RES2018", "862");  INSERT INTO Employee\_Scores (TeamID, UID, Score) VALUES("PU42", "PUR2021", "654");  INSERT INTO Employee\_Scores (TeamID, UID, Score) VALUES("HRM415", "HRM2017", "915");  INSERT INTO Employee\_Scores (TeamID, UID, Score) VALUES("AF147", "AAF2021", "531");  SELECT \* FROM Employee\_Scores;   CREATE INDEX SkillScore ON Employee\_Scores(TeamID, Score); |
| --- |

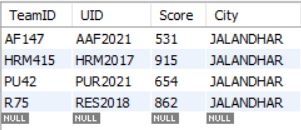
## RESULT:



## Alter Table:

| ALTER TABLE Employee\_Scores MODIFY COLUMN Score INT; ALTER TABLE Employee\_Scores ADD City VARCHAR(20) DEFAULT 'JALANDHAR';  SELECT \* FROM Employee\_Scores; |
| --- |

## RESULT:



## 

## Drop the Table which was used for Foreign Key:

| DROP TABLE Employee\_Scores; |
| --- |

## Drop Database:

| DROP DATABASE Departments; |
| --- |