**Question - 02:**

Firstly create the Database and add data using these SQL Queries and adding data into tables:

| use master GO CREATE DATABASE [Workforce\_Management\_System]; GO USE [Workforce\_Management\_System];  -- Create Tables with their Schema CREATE TABLE [Jobs] (  [JOB\_ID] VARCHAR(100) PRIMARY KEY NOT NULL,  JOB\_TITLE VARCHAR(100) NOT NULL,  MIN\_SALARY INT NOT NULL,  MAX\_SALARY INT NOT NULL );  CREATE TABLE [Locations] (  [LOCATION\_ID] INT PRIMARY KEY NOT NULL,  STREET\_ADDRESS VARCHAR(100) NOT NULL,  POSTAL\_CODE VARCHAR(100) DEFAULT NULL,  CITY VARCHAR(100) NOT NULL,  STATE\_PROVINCE VARCHAR(100) DEFAULT NULL,  COUNTRY\_ID CHAR(2) NOT NULL );  CREATE TABLE [Departments] (  [DEPARTMENT\_ID] INT PRIMARY KEY NOT NULL,  DEPARTMENT\_NAME VARCHAR(100) NOT NULL,  MANAGER\_ID INT NOT NULL,  LOCATION\_ID INT NOT NULL  FOREIGN KEY (LOCATION\_ID) REFERENCES Locations(LOCATION\_ID) );  CREATE TABLE [Employee] (  [EMPLOYEE\_ID] INT PRIMARY KEY NOT NULL,  FIRST\_NAME VARCHAR(100) NOT NULL,  LAST\_NAME VARCHAR(100) NOT NULL,  EMAIL VARCHAR(100) NOT NULL,  PHONE\_NUMBER VARCHAR(100) NOT NULL,  HIRE\_DATE DATE NOT NULL,  JOB\_ID VARCHAR(100) NOT NULL,  SALARY INT NOT NULL,  COMMISION\_PCT INT NOT NULL DEFAULT 0,  MANAGER\_ID INT NOT NULL,  DEPARTMENT\_ID INT NOT NULL  FOREIGN KEY (JOB\_ID) REFERENCES Jobs(JOB\_ID),  FOREIGN KEY (DEPARTMENT\_ID) REFERENCES Departments(DEPARTMENT\_ID) );  -- Entering Values into the Tables INSERT INTO Jobs(JOB\_ID, JOB\_TITLE, MIN\_SALARY, MAX\_SALARY) VALUES ('AD\_PRES', 'President', 20000, 40000), ('AD\_VP', 'Administration Vice President', 15000, 30000), ('AD\_ASST', 'Administration Assistant', 3000, 6000), ('FI\_MGR', 'Finance Manager', 8200, 16000), ('FI\_ACCOUNT', 'Accountant', 4200, 9000), ('AC\_MGR', 'Accounting Manager', 8200, 16000), ('AC\_ACCOUNT', 'Public Accountant', 4200, 9000), ('SA\_MAN', 'Sales Manager', 10000, 20000), ('SA\_REP', 'Sales Representative', 6000, 12000), ('PU\_MAN', 'Purchasing Manager', 8000, 15000), ('PU\_CLERK', 'Purchasing Clerk', 2500, 5500), ('ST\_MAN', 'Stock Manager', 5500, 8500), ('ST\_CLERK', 'Stock Clerk', 2000, 5000), ('SH\_CLERK', 'Shipping Clerk', 2500, 5500), ('IT\_PROG', 'Programmer', 4000, 10000), ('MK\_MAN', 'Marketing Manager', 9000, 15000), ('MK\_REP', 'Marketing Representative', 4000, 9000), ('HR\_REP', 'Human Resources Representative', 4000, 9000), ('PR\_REP', 'Public Relations Representative', 4500, 10500);  INSERT INTO Locations(LOCATION\_ID, STREET\_ADDRESS, POSTAL\_CODE, CITY, STATE\_PROVINCE, COUNTRY\_ID) VALUES (1000, '1297 Via Cola di Rie', '00989', 'Roma', NULL, 'IT'), (1100, '93091 Calle della Testa', '10934', 'Venice', NULL, 'IT'), (1200, '2017 Shinjuku-ku', '1689', 'Tokyo', 'Tokyo Perfecture', 'JP'), (1300, '9450 Kamiya-cho', '6823', 'Hiroshima', NULL, 'JP'), (1400, '2014 Jabberwocky Rd', '26192', 'Southlake', 'Texas', 'US'), (1500, '2011 Interiors Blvd', '99236', 'South San Francisco', 'California', 'US'), (1600, '2007 Zagora St', '50090', 'South Brunswick', 'New Jersey', 'US'), (1700, '2004 Charade Rd', '98199', 'Seattle', 'Washington', 'US'), (1800, '147 Spadina Ave', 'M5V 2L7', 'Toronto', 'Ontario', 'CA'), (1900, '6092 Boxwood St', 'YSW 9T2', 'Whitehorse', 'Yukon', 'CA'), (2000, '40-5-12 Laogianggen', '190518', 'Beijing', NULL, 'CN'), (2100, '1298 Vileparle (E)', '490231', 'Mumbai', 'Maharashtra', 'IN'), (2200, '12-98 Victoria Street', '2901', 'Sydney', 'New South Wales', 'AU'), (2300, '198 Clementi North', '540198', 'Singapore', NULL, 'SG'), (2400, '8204 Arthur St', NULL, 'London', NULL, 'UK'), (2500, 'Magdalen Centre, The Oxford Science Park', 'OX9 9ZB', 'Oxford', 'Oxford', 'UK'), (2600, '9702 Chester Road', '9629850293', 'Stretford', 'Manchester', 'UK'), (2700, 'Schwanthalerstr. 7031', '80925', 'Munich', 'Bavaria', 'DE'), (2800, 'Rua Frei Caneca 1360', '01307-002', 'Sao Paulo', 'Sao Paulo', 'BR'), (2900, '20 Rue des Corps-Saints', '1730', 'Geneva', 'Geneve', 'CH'), (3000, 'Murtenstrasse 921', '3095', 'Bern', 'BE', 'CH'), (3100, 'Pieter Breughelstraat 837', '3029SK', 'Utrecht', 'Utrecht', 'NL'), (3200, 'Mariano Escobedo 9991', '11932', 'Mexico City', 'Distrito Federal Mexico', 'MX');  INSERT INTO Departments(DEPARTMENT\_ID, DEPARTMENT\_NAME, MANAGER\_ID, LOCATION\_ID) VALUES (10, 'Administration', 200, 1700), (20, 'Marketing', 201, 1800), (30, 'Purchasing', 114, 1700), (40, 'Human Resources', 203, 2400), (50, 'Shipping', 121, 1500), (60, 'IT', 103, 1400), (70, 'Public Relations', 204, 2700), (80, 'Sales', 145, 2500), (90, 'Executive', 100, 1700), (100, 'Finance', 108, 1700), (110, 'Accounting', 205, 1700), (120, 'Treasury', 0, 1700), (130, 'Corporate Tax', 0, 1700), (140, 'Control And Credit', 0, 1700), (150, 'Shareholder Services', 0, 1700), (160, 'Benefits', 0, 1700);  INSERT INTO Employee(EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISION\_PCT, MANAGER\_ID, DEPARTMENT\_ID) VALUES (100, 'Steven', 'King', 'SKING', '515.123.4567', '1987-06-17', 'AD\_PRES', 24000, 0, 0, 90), (101, 'Neena', 'Kochhar', 'NKOCHHAR', '515.123.4568', '1987-06-18', 'AD\_VP', 17000, 0, 100, 90), (102, 'Lex', 'De Haan', 'LDEHAAN', '515.123.4569', '1987-06-19', 'AD\_VP', 17000, 0, 100, 90), (103, 'Alexander', 'Hunold', 'AHUNOLD', '590.423.4567', '1987-06-20', 'IT\_PROG', 9000, 0, 102, 60), (104, 'Bruce', 'Ernst', 'BERNST', '590.423.4568', '1987-06-21', 'IT\_PROG', 6000, 0, 103, 60), (105, 'David', 'Austin', 'DAUSTIN', '590.423.4569', '1987-06-22', 'IT\_PROG', 4800, 0, 103, 60), (106, 'Valli', 'Pataballa', 'VPATABAL', '590.423.4560', '1987-06-23', 'IT\_PROG', 4800, 0, 103, 60), (107, 'Diana', 'Lorentz', 'DLORENTZ', '590.423.5567', '1987-06-24', 'IT\_PROG', 4200, 0, 103, 60), (108, 'Nancy', 'Greenberg', 'NGREENBE', '515.124.4569', '1987-06-25', 'FI\_MGR', 12000, 0, 101, 100), (109, 'Daniel', 'Faviet', 'DFAVIET', '515.124.4169', '1987-06-26', 'FI\_ACCOUNT', 9000, 0, 108, 100), (110, 'John', 'Chen', 'JCHEN', '515.124.4269', '1987-06-27', 'FI\_ACCOUNT', 8200, 0, 108, 100), (111, 'Ismael', 'Sciarra', 'ISCIARRA', '515.124.4369', '1987-06-28', 'FI\_ACCOUNT', 7700, 0, 108, 100), (112, 'Jose Manuel', 'Urman', 'JMURMAN', '515.124.4469', '1987-06-29', 'FI\_ACCOUNT', 7800, 0, 108, 100), (113, 'Luis', 'Popp', 'LPOPP', '515.124.4567', '1987-06-30', 'FI\_ACCOUNT', 6900, 0, 108, 100), (114, 'Den', 'Raphaely', 'DRAPHEAL', '515.127.4561', '1987-07-01', 'PU\_MAN', 11000, 0, 100, 30), (115, 'Alexander', 'Khoo', 'AKHOO', '515.127.4562', '1987-07-02', 'PU\_CLERK', 3100, 0, 114, 30), (116, 'Shelli', 'Baida', 'SBAIDA', '515.127.4563', '1987-07-03', 'PU\_CLERK', 2900, 0, 114, 30), (117, 'Sigal', 'Tobias', 'STOBIAS', '515.127.4564', '1987-07-04', 'PU\_CLERK', 2800, 0, 114, 30), (118, 'Guy', 'Himuro', 'GHIMURO', '515.127.4565', '1987-07-05', 'PU\_CLERK', 2600, 0, 114, 30), (119, 'Karen', 'Colmenares', 'KCOLMENA', '515.127.4566', '1987-07-06', 'PU\_CLERK', 2500, 0, 114, 30), (120, 'Matthew', 'Weiss', 'MWEISS', '650.123.1234', '1987-07-07', 'ST\_MAN', 8000, 0, 100, 50), (121, 'Adam', 'Fripp', 'AFRIPP', '650.123.2234', '1987-07-08', 'ST\_MAN', 8200, 0, 100, 50), (122, 'Payam', 'Kaufling', 'PKAUFLIN', '650.123.3234', '1987-07-09', 'ST\_MAN', 7900, 0, 100, 50), (123, 'Shanta', 'Vollman', 'SVOLLMAN', '650.123.4234', '1987-07-10', 'ST\_MAN', 6500, 0, 100, 50), (124, 'Kevin', 'Mourgos', 'KMOURGOS', '650.123.5234', '1987-07-11', 'ST\_MAN', 5800, 0, 100, 50), (125, 'Julia', 'Nayer', 'JNAYER', '650.124.1214', '1987-07-12', 'ST\_CLERK', 3200, 0, 120, 50); |
| --- |

* **SQL Queries:**

**Q1: find the name (first\_name, last\_name) and the salary of the employees who have a higher salary than the employee whose last\_name = 'Bull'.**

| SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM Employee WHERE SALARY > (SELECT SALARY FROM Employee WHERE LAST\_NAME = 'Bull'); |
| --- |

**Q2: find the name (first\_name, last\_name) of all employees who work in the IT department.**

| SELECT FIRST\_NAME, LAST\_NAME FROM Employee WHERE DEPARTMENT\_ID = (SELECT DEPARTMENT\_ID FROM Departments WHERE DEPARTMENT\_NAME = 'IT'); |
| --- |

**Q3: find the name (first\_name, last\_name) of the employees who have a manager and worked in a USA based department.**

| SELECT DISTINCT FIRST\_NAME, LAST\_NAME FROM Employee WHERE MANAGER\_ID IS NOT NULL AND DEPARTMENT\_ID IN (SELECT DEPARTMENT\_ID FROM Departments WHERE LOCATION\_ID IN (SELECT LOCATION\_ID FROM Locations WHERE COUNTRY\_ID = 'US')); |
| --- |

**Q4: find those employees who earn more than the average salary. Return employee ID, first name, last name.**

| SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME FROM Employee WHERE SALARY > (SELECT AVG(SALARY) FROM Employee); |
| --- |

**Q5: find those employees whose department is located at ‘Toronto’. Return first name, last name, employee ID, job ID.**

| SELECT e.FIRST\_NAME, e.LAST\_NAME, e.EMPLOYEE\_ID, e.JOB\_ID FROM Employee e JOIN Departments d ON e.DEPARTMENT\_ID = d.DEPARTMENT\_ID JOIN Locations l ON d.LOCATION\_ID = l.LOCATION\_ID WHERE l.CITY = 'Toronto'; |
| --- |

**Q6: find those employees who report to that manager whose first name is 'Payam'. Return first name, last name, employee ID and salary.**

| SELECT FIRST\_NAME, LAST\_NAME, EMPLOYEE\_ID, SALARY FROM Employee WHERE MANAGER\_ID IN (SELECT EMPLOYEE\_ID FROM Employee WHERE FIRST\_NAME = 'Payam'); |
| --- |

**Q7: find all those departments where at least one employee is employed. Return department name.**

| SELECT DEPARTMENT\_NAME FROM Departments WHERE DEPARTMENT\_ID IN (SELECT DEPARTMENT\_ID FROM Employee); |
| --- |

**Q8: find those employees who do not work in the departments where managers’ IDs are between 100 and 200 (Begin and end values are included.). Return all the fields of the employees.**

| SELECT \* FROM Employee WHERE DEPARTMENT\_ID NOT IN (  SELECT DEPARTMENT\_ID  FROM Departments  WHERE MANAGER\_ID BETWEEN 100 AND 200 ); |
| --- |

**Q9: From the following table, find those employees whose salary matches the lowest salary of any of the departments. Return first name, last name and department ID.**

| SELECT FIRST\_NAME, LAST\_NAME, DEPARTMENT\_ID FROM Employee WHERE SALARY = (SELECT MIN(SALARY) FROM Employee); |
| --- |

**Q10: find the name (first\_name, last\_name) of the employees who are managers.**

| SELECT FIRST\_NAME, LAST\_NAME FROM Employee WHERE EMPLOYEE\_ID IN (SELECT MANAGER\_ID FROM Departments); |
| --- |

**Q11: find those employees whose salary is lower than that of employees whose job title is 'MK\_MAN'. Exclude employees of Job title 'MK\_MAN'. Return employee ID, first name, last name, job ID.**

| SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, JOB\_ID FROM Employee WHERE SALARY < (  SELECT SALARY  FROM Employee  WHERE JOB\_ID = 'MK\_MAN' ) AND JOB\_ID != 'MK\_MAN'; |
| --- |

**Q12: Find the name (first\_name, last\_name), and salary of the employees whose salary is greater than the average salary.**

| SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM Employee WHERE SALARY > (  SELECT AVG(SALARY)  FROM Employee ); |
| --- |

**Q13: Find the name (first\_name, last\_name), and salary of the employees whose salary is equal to the minimum salary for their job grade.**

| SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM Employee WHERE SALARY = (  SELECT MIN\_SALARY  FROM Jobs  WHERE JOB\_ID = Employee.JOB\_ID ); |
| --- |

**Q14: Find the name (first\_name, last\_name), and salary of the employees who earns more than the average salary and works in any of the IT departments.**

| SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM Employee WHERE SALARY > (  SELECT AVG(SALARY)  FROM Employee ) AND DEPARTMENT\_ID IN (  SELECT DEPARTMENT\_ID  FROM Departments  WHERE DEPARTMENT\_NAME = 'IT' ); |
| --- |

**Q15: Find the name (first\_name, last\_name), and salary of the employees who earns more than the earning of Mr. Bell.**

| SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM Employee WHERE SALARY > (  SELECT SALARY  FROM Employee  WHERE LAST\_NAME = 'Bell' ); |
| --- |

**Q16: .Find the name (first\_name, last\_name), and salary of the employees who earn the same salary as the minimum salary for all departments.**

| SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM Employee WHERE SALARY = (  SELECT MIN(SALARY)  FROM Employee ); |
| --- |

**Q17: Find the name (first\_name, last\_name), and salary of the employees whose salary is greater than the average salary of all departments.**

| SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM Employee WHERE SALARY > (  SELECT AVG(SALARY)  FROM Employee ); |
| --- |

**Q18: Find the 3rd maximum salary in the employees table.**

| SELECT DISTINCT SALARY FROM Employee ORDER BY SALARY DESC OFFSET 2 ROWS FETCH NEXT 1 ROWS ONLY; |
| --- |

* **Using Relational Algebra:**

**Q1: find the name (first\_name, last\_name) and the salary of the employees who have a higher salary than the employee whose last\_name = 'Bull'.**



**Q2: find the name (first\_name, last\_name) of all employees who work in the IT department.**



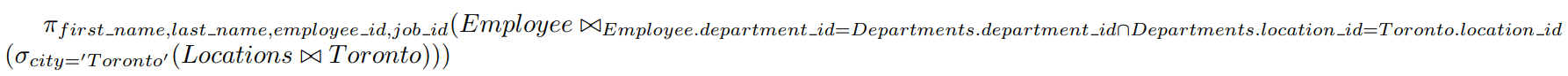
**Q3: find the name (first\_name, last\_name) of the employees who have a manager and worked in a USA based department.**



**Q4: find those employees who earn more than the average salary. Return employee ID, first name, last name.**



**Q5: find those employees whose department is located at ‘Toronto’. Return first name, last name, employee ID, job ID.**



**Q6: find those employees who report to that manager whose first name is 'Payam'. Return first name, last name, employee ID and salary.**



**Q7: find all those departments where at least one employee is employed. Return department name.**



**Q8: find those employees who do not work in the departments where managers’ IDs are between 100 and 200 (Begin and end values are included.). Return all the fields of the employees.**



It can also be written as:



**Q9: From the following table, find those employees whose salary matches the lowest salary of any of the departments. Return first name, last name and department ID.**



**Q10: find the name (first\_name, last\_name) of the employees who are managers.**



**Q11: find those employees whose salary is lower than that of employees whose job title is 'MK\_MAN'. Exclude employees of Job title 'MK\_MAN'. Return employee ID, first name, last name, job ID.**



**Q12: Find the name (first\_name, last\_name), and salary of the employees whose salary is greater than the average salary.**



**Q13: Find the name (first\_name, last\_name), and salary of the employees whose salary is equal to the minimum salary for their job grade.**



**Q14: Find the name (first\_name, last\_name), and salary of the employees who earns more than the average salary and works in any of the IT departments.**







**Q15: Find the name (first\_name, last\_name), and salary of the employees who earns more than the earning of Mr. Bell.**

****

**Q16: .Find the name (first\_name, last\_name), and salary of the employees who earn the same salary as the minimum salary for all departments.**

****

**Q17: Find the name (first\_name, last\_name), and salary of the employees whose salary is greater than the average salary of all departments.**



**Q18: Find the 3rd maximum salary in the employees table.**



**Note:** The Relational Algebra equations written in more than 1 different lines are the same query. But, it was a long query. So, I divided it into multiple lines. Moreover, please check the Workforce\_Management\_System.sql file which is submitted on Google Classroom and contains all SQL Queries for Question 02.