

DBMS LAB ASSIG 5

• Name :HITU RAJ

• Roll no. :2005025

• Branch :CSE

-- 1. Find the average salary of each department.

CREATE TABLE employee

(

emp\_id INTEGER NOT NULL,

f\_name VARCHAR(20),

l\_name VARCHAR(20),

jobtype VARCHAR(20),

salary INTEGER,

commision INTEGER,

dept VARCHAR(20),

manager\_id INTEGER,

PRIMARY KEY(emp\_id),

doj VARCHAR(20)

);

DROP TABLE Employee;

ALTER TABLE Employee ADD COLUMN doj DATE;

INSERT INTO

employee(emp\_id,f\_name,l\_name,jobtype,salary,dept,doj)

values(1, 'arun' , 'khan' , 'manager' ,90000, 'production' ,

'1998-01-04');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,dept,doj) values(2 ,

'barun','kumar','manager',80000,'marketing','1998-02-09');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,dept,manager\_id,doj) values(3 ,

'chitra','kapoor','engineer',60000,'production',1,'1998-01-0

8');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,dept,manager\_id,doj) values(4 ,

2

'dheeraj','mishra','manager',75000,'sales',2,'2001-12-27');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,dept,manager\_id,doj) values(5 ,

'emma','dutt','engineer',55000,'production',1,'2002-03-20');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,dept,doj) values(6 ,

'floki','dutt','accountant',70000,'accounts','2000-07-16');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,dept,manager\_id,doj) values(7 ,

'dheeraj','kumar','clerk',40000,'accounts',6,'2016-07-01');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,dept,doj) values(8 ,

'saul','good','engineer',60000,'r&d','2014-09-06');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,dept,manager\_id,doj) values(9 ,

'mou','bhatt','clerk',30000,'sales',4,'2018-03-08');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,commision,dept,manager\_id,doj)

values(10 ,'sunny','deol','salesman',20000,10000,'marketing'

,2,'2001-03-31');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,dept,manager\_id,doj) values(11,

'bobby','deol','engineer',35000,'r&d',8,'2017-10-17');

INSERT INTO employee(emp\_id ,

f\_name,l\_name,jobtype,salary,commision,dept,manager\_id,doj)

values(12,

'amir','khan','salesman',15000,5000,'marketing',2,'2013-01-1

1');

SELECT dept AS "Department",AVG(salary) AS "Average\_Salary"

FROM Employee

GROUP BY dept;

-- 2

--  Find the average salary for each jobtype according

-- to each department.

SELECT jobtype,dept AS "Department",AVG(salary) AS "Average\_Salary"

FROM Employee

GROUP BY jobtype, dept

ORDER BY jobtype;

-- 3.Find the department names and their corresponding

-- average salary where the average salary is greater

-- than 40000.

SELECT dept AS "Department",AVG(salary) AS "Average\_Salary"

FROM Employee

GROUP BY dept

HAVING AVG(salary)>40000;

-- 4

--  Select the departments where the maximum salary

-- is more than 55000.

SELECT dept AS "Department"

FROM Employee

GROUP BY dept

HAVING MAX(salary)>55000;

-- 5. Find the department names and their average

-- salary where the maximum salary of the department is

-- higher than 55000.

SELECT dept AS "Department",AVG(salary) AS "Average\_Salary"

FROM Employee

GROUP BY dept

HAVING AVG(salary)>55000;

-- 6. Display the jobtypes and the total monthly salary

-- --

-- for each jobtypes as “PAYROLL”, where the total

-- payroll according to jobtypes exceeds 100000/month.

SELECT jobtype , SUM(salary)/12 AS "PAYROLL"

FROM Employee

GROUP BY jobtype

HAVING SUM(salary)/12>100000;

-- 7. Display the jobtypes and the total monthly salary

-- for each jobtypes as “PAYROLL”, where the total

-- payroll according to jobtypes exceeds 100000/month

-- and jobtype is not engineer.

SELECT jobtype , SUM(salary)/12 AS "PAYROLL"

FROM Employee

GROUP BY jobtype

HAVING SUM(salary)/12>100000 AND jobtype<>'Engineer';

-- 8. Display the jobtypes and the total monthly salary

-- for each jobtypes as “PAYROLL”, where the total

-- payroll according to jobtypes exceeds 60000/month

-- and jobtype is not engineer and sort the list in

-- ascending order of sum of salary.

SELECT jobtype , SUM(salary)/12 AS "PAYROLL"

FROM Employee

GROUP BY jobtype

HAVING SUM(salary)/12>60000 AND jobtype<>'Engineer'

ORDER BY SUM(salary) ASC;

-- 9. Display the jobtypes and the total monthly salary

-- for each jobtypes as “PAYROLL”, where the total

-- payroll according to jobtypes exceeds 50000/month

-- and jobtype is not engineer and sort the list in

-- descending order of sum of salary.

SELECT jobtype , SUM(salary)/12 AS "PAYROLL"

FROM Employee GROUP BY jobtype

HAVING SUM(salary)/12>50000 AND jobtype<>'Engineer'

ORDER BY SUM(salary) DESC;

-- 10.

-- Find the maximum average salary according to

-- departments.

SELECT dept AS "Department", MAX(salary) AS "Maximum Average

Salary"

FROM (

SELECT dept,AVG(salary) AS salary

FROM Employee

GROUP BY dept

) AS E;

-- 11. Find the minimum average salary according to

-- jobtypes.

SELECT dept AS "Department", MIN(salary) AS "Maximum Average

Salary"

FROM (

SELECT dept,AVG(salary) AS salary

FROM Employee

GROUP BY dept

) AS E;

-- 12. Find the employee name and date of joining who

-- are working in delhi.

CREATE TABLE department

(

d\_name VARCHAR(20),

d\_loc VARCHAR(20),

hod\_id INTEGER NOT NULL

);

INSERT INTO department VALUES('sales','Kolkata','4');

INSERT INTO department VALUES('accounts','New Delhi','6');

INSERT INTO department VALUES('production','Kolkata','1');

INSERT INTO department VALUES('marketing','Kolkata','2');

INSERT INTO department VALUES('r&d','New Delhi','8');

SELECT CONCAT(f\_name," ",l\_name) AS "Employee name", doj

FROM Employee

CROSS JOIN (

SELECT d\_name,d\_loc

FROM department

WHERE d\_loc='New Delhi'

) AS E2

ON Employee.dept=E2.d\_name;

-- 13. Create the table ‘Emp\_Address’ for storing the

-- permanent address of the employees and insert the

-- values.

CREATE TABLE Emp\_Address

(

emp\_id INT NOT NULL,

city VARCHAR(30),

district VARCHAR(30),

state VARCHAR(30)

);

INSERT INTO Emp\_Address VALUES(1,'Suri','Birbhum','WB');

INSERT INTO Emp\_Address VALUES(3,'Kolkata','Kolkata','WB');

INSERT INTO Emp\_Address

VALUES(4,'Bhubaneswar','Khurda','Odisha');

INSERT INTO Emp\_Address VALUES(5,'Noida','GB Nagar','UP');

INSERT INTO Emp\_Address

VALUES(6,'Secunderabad','Hydrabad','Telengana');

INSERT INTO Emp\_Address

VALUES(7,'Derhadun','Derhadun','Uttarakhand');

INSERT INTO Emp\_Address VALUES(8,'Asansol','Burdwan','WB');

INSERT INTO Emp\_Address

VALUES(9,'Siliguri','Darjeeling','WB');

INSERT INTO Emp\_Address VALUES(10,'Kolkata','Kolkata','WB');

INSERT INTO Emp\_Address VALUES(11,'New Delhi','New

Delhi','Delhi');

-- 14. Display the employee name with their home city

-- and the city they work in.

SELECT CONCAT(f\_name," ",l\_name) AS "Employee name", city AS

"Home city",d\_loc AS "Work city"

FROM Employee

CROSS JOIN (

SELECT d\_name,d\_loc

FROM department

) AS E2 ON Employee.dept=E2.d\_name

CROSS JOIN

emp\_address ON Employee.emp\_id=emp\_address.emp\_id;

-- 15. Create the following Job\_Grades table.

CREATE TABLE Job\_Grades

(

grade CHAR(1),

lowest\_sal INT NOT NULL,

highest\_sal INT

);

INSERT INTO Job\_Grades VALUES('A','10000','24999');

INSERT INTO Job\_Grades VALUES('B','25000','49999');

INSERT INTO Job\_Grades VALUES('C','50000','100000');

SELECT emp\_id,max(l\_name) AS "last name" FROM Employee UNION

SELECT emp\_id,MIN(l\_name) AS "last name" FROM Employee ;

-- 16. Display the employee names along with their

-- salary and job\_grade.

SELECT CONCAT(f\_name," ",l\_name) AS "Employee name" ,

salary , grade AS "Job Grade"

FROM employee

CROSS JOIN

job\_grades WHERE salary BETWEEN lowest\_sal AND highest\_sal;

-- 17. Display the employees name along with their

-- manager’s name. (use SELF JOIN)

SELECT A.f\_name AS "Employee name",B.f\_name AS "Manager

name"

FROM Employee A,Employee B

WHERE A.dept=B.dept AND B.jobtype='Manager';

-- 18. Display emp\_id, f\_name, d\_loc, and hod\_id (using

-- natural join).

SELECT emp\_id,f\_name,l\_name,d\_loc,hod\_id FROM Employee

NATURAL JOIN department

WHERE Employee.dept = department.d\_name;

-- 19. Display the employees f\_name, city and state in

-- which they live (using natural join).

SELECT f\_name,city,state FROM Employee

NATURAL JOIN emp\_address

WHERE Employee.emp\_id = emp\_address.emp\_id;

-- 20. Display the employees emp\_id, f\_name, d\_loc,

-- hod\_id using inner join.

SELECT emp\_id,f\_name,d\_loc,hod\_id FROM Employee

INNER JOIN department

ON Employee.dept = department.d\_name;

-- 21. Display the employees f\_name, city and state in

-- which they live (using inner join).

SELECT f\_name,city,state FROM Employee

INNER JOIN emp\_address

ON Employee.emp\_id = emp\_address.emp\_id;

-- 22. Display the employees f\_name, city and state in

-- which they live (using join keyword).

SELECT f\_name,city,state FROM Employee

JOIN emp\_address

ON Employee.emp\_id = emp\_address.emp\_id;

-- 23. Insert the following two rows in the employee

-- table without inserting any value in the department

-- field.

INSERT INTO Employee (emp\_id,f\_name, jobtype,

salary,commision, manager\_id, doj) VALUES (20,'Alex',

'Engineer', 28000, 2000, 1, '2017/01/31');

INSERT INTO Employee (emp\_id,f\_name,l\_name, jobtype,

salary,commision, manager\_id, doj) VALUES

(21,'Priya','Patel', 'Clerk', 12000, 500, 1, '2017/04/01');

-- 24. Insert the following two rows into the department

-- table.

INSERT INTO department VALUES ('Training','Mumbai',1);

INSERT INTO department VALUES ('Placement','Mumbai',1);

-- 25.Display the employees f\_name, city and state in

-- which they live after joining employee and

-- employee\_address table using left outer join.

SELECT f\_name,city,state FROM employee

LEFT OUTER JOIN emp\_address

ON employee.emp\_id = emp\_address.emp\_id;

-- 26. Display the employees f\_name and their work

-- location after joining employee and department table

-- using left join.

SELECT f\_name, d\_loc FROM Employee

LEFT JOIN department ON Employee.dept = department.d\_name;

-- 27. Display the employees f\_name and their work

-- location after joining employee and department table

-- using right join.

SELECT f\_name, d\_loc FROM Employee

RIGHT JOIN department ON Employee.dept = department.d\_name;

-- 28. Display the employees f\_name and their work

-- location after joining employee and department table

-- using full join/full outer join.

SELECT f\_name, d\_loc FROM Employee

LEFT JOIN department ON Employee.dept = department.d\_name

UNION

SELECT f\_name, d\_loc FROM Employee

RIGHT JOIN department ON Employee.dept = department.d\_name;

-- 29. Find the employees who are working in their home

-- city.

SELECT Employee.\*

FROM Employee

CROSS JOIN department

ON Employee.dept=department.d\_name

CROSS JOIN emp\_address

ON Employee.emp\_id=emp\_address.emp\_id

WHERE emp\_address.city=department.d\_loc;

-- 30. Find the job type having the minimum average

-- salary according to jobtypes.

SELECT jobtype , MIN(salary) AS "Minimum Salary" FROM

Employee GROUP BY jobtype;

SELECT d\_name,d\_loc,hod\_id FROM department;

SELECT emp\_id,city,district,state FROM emp\_address;

SELECT emp\_id,city,district,state FROM emp\_address;

OUTPUTS:-



















