

DBMS LAB ASSIG 7

• Name :HITU RAJ

• Roll no. :2005025

• Branch :CSE

-- Q1. Create the following table and insert the values.

-- Using the above Job\_History table and the Employee

-- table (of assignment 2) write SQL statements for the

-- following queries.

CREATE TABLE Job\_History

(

emp\_id INT(6),

startdate DATE,

End\_Date DATE,

Job\_Type VARCHAR(20),

D\_Name VARCHAR(15)

);

INSERT INTO Job\_History Values(1,'1998-01-04','2001-06-30',

'engineer','production');

INSERT INTO Job\_History Values(2,'1998-02-09','2002-02-28',

'salesman','sales');

INSERT INTO Job\_History Values(1,'2001-07-01','2010-12-31',

'manager','rnd');

INSERT INTO Job\_History Values(4,'2001-12-27','2016-09-19',

'sales\_executive','marketing');

INSERT INTO Job\_History Values(2,'2002-03-01','2015-03-30',

'sales\_executive','marketing');

INSERT INTO Job\_History Values(2,'2016-04-01','2017-12-16',

'manager','sales');

INSERT INTO Job\_History Values(4,'2016-09-20','2017-12-16',

'asst. manager','sales');

INSERT INTO Job\_History Values(6,'2000-07-16','2006-11-30',

'clerk','accounts');

INSERT INTO Job\_History Values(5,'2002-03-20','2011-08-12',

'engineer','rnd');

INSERT INTO Job\_History Values(1,'2011-01-01','2012-01-31',

'engineer','production');

SELECT \* FROM Job\_History;

CREATE TABLE Employee(

emp\_id INT,

f\_name VARCHAR(50),

l\_name VARCHAR(50),

job\_type VARCHAR(50),

salary INT,

commision INT,

dept VARCHAR(50),

manager\_id INT,

DOJ DATE);

INSERT INTO Employee (emp\_id,f\_name,l\_name,job\_type,salary,dept,DOJ)

VALUES (1,'arun','khan','manager',90000,'production','1998-01-04');

INSERT INTO Employee (emp\_id,f\_name,l\_name,job\_type,salary,dept,DOJ)

VALUES (2,'barun','kumar','manager',80000,'marketing','1998-02-09');

INSERT INTO Employee (emp\_id,f\_name,l\_name,job\_type,salary,dept,manager\_id,DOJ)

VALUES (3,'chitra','kapoor','engineer',60000,'production',1,'1998-01-08');

INSERT INTO Employee (emp\_id,f\_name,l\_name,job\_type,salary,dept,manager\_id,DOJ)

VALUES (4,'dhirag','mishra','manager',75000,'sales',2,'2001-12-27');

INSERT INTO Employee (emp\_id,f\_name,l\_name,job\_type,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,job\_type,salary,dept,DOJ)

VALUES (6,'floki','dutt','accountant',70000,'accounts','2000-07-16');

INSERT INTO Employee (emp\_id,f\_name,l\_name,job\_type,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,job\_type,salary,dept,DOJ)

VALUES (8,'saul','good','engineer',60000,'rnd','2014-09-06');

INSERT INTO Employee (emp\_id,f\_name,l\_name,job\_type,salary,dept,manager\_id,DOJ)

VALUES (9,'mou','bhat','clerk',30000,'sales',4,'2018-03-08');

INSERT INTO Employee (emp\_id,f\_name,l\_name,job\_type,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,job\_type,salary,dept,manager\_id,DOJ)

VALUES (11,'bobby','deol','engineer',35000,'rnd',8,'2017-10-17');

INSERT INTO Employee (emp\_id,f\_name,l\_name,job\_type,salary,commision,dept,manager\_id,DOJ)

VALUES (12,'amir','khan','salesman',15000,5000,'marketing',2,'2013-01-11');

SELECT \* FROM Employee;

-- Q2. Display the emp\_id along with every job\_type

-- they have worked (including their current job\_type).

-- (use union/union all)

SELECT EMP\_ID, DEPT FROM EMPLOYEE UNION SELECT Emp\_id,

Job\_Type FROM Job\_History;

-- Q3. Display the emp\_id, d\_name, and job\_types

-- current and previous (if any) of all employees.(use

-- union/union all)

SELECT EMP\_ID, JOB\_TYPE, DEPT FROM EMPLOYEE UNION SELECT

Emp\_id, Job\_Type, D\_Name FROM Job\_History;

-- Q4. Display the emp\_id and the job\_type of

-- employees who currently have a job title that they

-- held previously.(use intersect)

SELECT EMP\_ID, JOB\_TYPE FROM EMPLOYEE INTERSECT SELECT

Emp\_id, Job\_Type FROM Job\_History;

-- Q5. Find the employees who have not changed their

-- job for once.(use minus)

SELECT EMP\_ID, JOB\_TYPE FROM EMPLOYEE MINUS SELECT Emp\_id,

Job\_Type FROM Job\_History;

-- Q6. Find the employees who earn more than Chitra.

-- (use single-row subquery)

SELECT EMP\_ID FROM(SELECT EMP\_ID FROM EMPLOYEE WHERE

SALARY>60000);

-- Q7. Find the employees details who have the same

-- job\_type as of emp\_id 7. (use single-row subquery)

SELECT \* FROM (SELECT \* FROM EMPLOYEE WHERE JOB\_TYPE =

'clerk');

-- Q8.Display the employee names whose job is the same

-- as employee 3 and earn more than employee 7. (use

-- single-row subquery)

SELECT F\_NAME FROM (SELECT F\_NAME FROM EMPLOYEE WHERE

JOB\_TYPE='engineer' AND SALARY>40000);

-- Q9. Display the employees earning less than the

-- average salary. (use single-row subquery)

SELECT F\_NAME, SALARY FROM EMPLOYEE WHERE SALARY < (SELECT

AVG(SALARY) FROM EMPLOYEE);

-- Q10. Find the job\_type with the lowest average

salary. (use single-row subquery)

SELECT JOB\_TYPE FROM (SELECT JOB\_TYPE, AVG(SALARY) FROM

EMPLOYEE GROUP BY JOB\_TYPE HAVING AVG(SALARY) = (SELECT

MIN(AVG(SALARY)) FROM EMPLOYEE GROUP BY(JOB\_TYPE)));

-- Q11. Display all the department names whose minimum

-- salary is greater than the minimum salary of the Sales

-- department.

SELECT DEPT, MIN(SALARY) FROM EMPLOYEE GROUP BY (DEPT)

HAVING MIN(SALARY) < (SELECT MIN(SALARY) FROM EMPLOYEE GROUP

BY (DEPT) HAVING DEPT='sales');

-- Q12. Select the employee names, department and

-- salary who are the lowest earners of their

-- corresponding department (use multi-row subquery).

SELECT F\_NAME, DEPT, SALARY FROM EMPLOYEE WHERE SALARY

IN(SELECT MIN(SALARY) FROM EMPLOYEE GROUP BY (DEPT));

-- Q13. Find the highest earners of each job\_type.(use

-- multi-row subquery).

SELECT F\_NAME, JOB\_TYPE, SALARY FROM EMPLOYEE WHERE SALARY

IN(SELECT MAX(SALARY) FROM EMPLOYEE GROUP BY(JOB\_TYPE));

-- Q14. Display the employees who are not engineers and

-- earn less than any engineer.(use multi-row subquery).

SELECT F\_NAME, JOB\_TYPE, DEPT, SALARY FROM EMPLOYEE WHERE

JOB\_TYPE != 'engineer' AND SALARY < ANY(SELECT SALARY FROM

EMPLOYEE WHERE JOB\_TYPE = 'engineer');

-- Q15. Display the employees who are not clerks but

-- earn more than all clerks.(use multi-row subquery).

SELECT F\_NAME, JOB\_TYPE, DEPT, SALARY FROM EMPLOYEE WHERE

JOB\_TYPE != 'clerk' AND SALARY > ALL(SELECT SALARY FROM

EMPLOYEE WHERE JOB\_TYPE = 'clerk');

-- Q16. Display the top 5 highest earning employees.

SELECT F\_NAME, SALARY FROM (SELECT F\_NAME, SALARY FROM

EMPLOYEE ORDER BY SALARY DESC) WHERE ROWNUM<=5;

-- Q17. Display the name and department of the top 2

-- highest paid managers.

SELECT F\_NAME, L\_NAME, DEPT, SALARY FROM (SELECT F\_NAME,

L\_NAME, DEPT, SALARY FROM EMPLOYEE WHERE JOB\_TYPE =

'engineer' ORDER BY SALARY DESC) WHERE ROWNUM<=2;

-- Q18. Update the salary of the employees working as

-- managers to the average salary of all the employees.

UPDATE EMPLOYEE SET SALARY = (SELECT AVG(SALARY) FROM

EMPLOYEE) WHERE JOB\_TYPE='manager';

OUTPUTS:-

















