

# Assignment V

Name :- Bibek Chand Sah

Roll :- 22054029

Branch :- CSE

Section :- CSE-05

1. Find the average salary of each department.

→ SELECT dept, AVG(salary) AS average\_salary FROM employee GROUP BY dept;

```
SQL> SELECT dept, AVG(salary) AS average_salary FROM employee GROUP BY dept;
```

DEPT	AVERAGE_SALARY
accounts	55000
production	68333.3333
sales	75000
r&d	47500
marketing	50000
null	12000
admin	28000

7 rows selected.

2. Find the average salary for each jobtype according to each department.

→ SELECT dept, job\_types, AVG(salary) AS average\_salary FROM employee GROUP BY dept, job\_types;

```
SQL> SELECT dept, job_types, AVG(salary) AS average_salary FROM employee GROUP BY dept, job_types;
```

DEPT	JOB_TYPES	AVERAGE_SALARY
production	engineer	57500
r&d	engineer	47500
admin	engineer	28000
accounts	accountant	70000
production	manager	90000
marketing	manager	80000
accounts	clerk	40000
sales	manager	75000
marketing	salesman	20000
null	clerk	12000

10 rows selected.

3. Find the department names and their corresponding average salary where the average salary is greater than 40000.

→ SELECT dept, AVG(salary) AS average\_salary FROM employee GROUP BY dept HAVING AVG(salary) > 40000;

```
DEPT      AVERAGE_SALARY
-----
accounts      55000
production    68333.3333
sales          75000
r&d           47500
marketing      50000
```

4. Select the departments where the maximum salary is more than 55000.

➔ SELECT dept FROM employee GROUP BY dept HAVING MAX(salary) > 55000;

```
SQL> SELECT dept FROM employee GROUP BY dept HAVING MAX(salary) > 55000;

DEPT
-----
accounts
production
sales
r&d
marketing
```

5. Find the department names and their average salary where the maximum salary of the department is higher than 55000.

➔ SELECT dept, AVG(salary) AS average\_salary FROM employee GROUP BY dept HAVING MAX(salary) > 55000;

```
SQL> SELECT dept, AVG(salary) AS average_salary FROM employee GROUP BY dept HAVING MAX(salary) > 55000;

DEPT      AVERAGE_SALARY
-----
accounts          55000
production    68333.3333
sales           75000
r&d            47500
marketing        50000
```

6. Display the job\_types and the total monthly salary for each jobtypes as “PAYROLL”, where the total payroll according to jobtypes exceeds 100000/month.

➔ SELECT job\_types, SUM(salary) AS total\_salary FROM employee GROUP BY job\_types HAVING SUM(salary) > 100000;

```
SQL> SELECT job_types, SUM(salary) AS total_salary FROM employee GROUP BY job_types HAVING SUM(salary) > 100000;

JOB_TYPES  TOTAL_SALARY
-----
engineer      238000
manager      245000
```

7. Display the job\_types 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 job\_types, SUM(salary) AS total\_salary FROM employee WHERE job\_types != 'engineer' GROUP BY job\_types HAVING SUM(salary) > 100000;

```
SQL> SELECT job_types, SUM(salary) AS total_salary FROM employee WHERE job_types != 'engineer' GROUP BY job_types HAVING SUM(salary) > 100000;

JOB_TYPES  TOTAL_SALARY
-----
manager    245000
```

8. Display the job\_types 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 job\_types, SUM(salary) AS total\_salary FROM employee WHERE job\_types != 'engineer' GROUP BY job\_types HAVING SUM(salary) > 60000 ORDER BY SUM(salary) ASC;

```
SQL> SELECT job_types, SUM(salary) AS total_salary FROM employee WHERE job_types != 'engineer' GROUP BY job_types HAVING SUM(salary) > 60000 ORDER BY SUM(salary) ASC;

JOB_TYPES  TOTAL_SALARY
-----
accountant  70000
manager    245000
```

9. Display the job\_types 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 job\_types, SUM(salary) AS total\_salary FROM employee WHERE job\_types != 'engineer' GROUP BY job\_types HAVING SUM(salary) > 50000 ORDER BY SUM(salary) DESC;

```
SQL> SELECT job_types, SUM(salary) AS total_salary FROM employee WHERE job_types != 'engineer' GROUP BY job_types HAVING SUM(salary) > 50000 ORDER BY SUM(salary) DESC;

JOB_TYPES  TOTAL_SALARY
-----
manager    245000
accountant  70000
clerk       52000
```

10. Find the maximum average salary according to departments.

➔ SELECT DEPT, AVG\_SALARY AS MAX\_AVG\_SALARY FROM (SELECT DEPT, AVG(SALARY) AS AVG\_SALARY FROM EMPLOYEE GROUP BY DEPT) Subquery WHERE AVG\_SALARY = (SELECT MAX(AVG\_SALARY) FROM (SELECT AVG(SALARY) AS AVG\_SALARY FROM EMPLOYEE GROUP BY DEPT));

```
SQL> SELECT DEPT, AVG_SALARY AS MAX_AVG_SALARY FROM (SELECT DEPT, AVG(SALARY) AS AVG_SALARY FROM EMPLOYEE GROUP BY DEPT) Subquery WHERE AVG_SALARY = (SELECT MAX(AVG_SALARY) FROM (SELECT AVG(SALARY) AS AVG_SALARY FROM EMPLOYEE GROUP BY DEPT));

DEPT      MAX_AVG_SALARY
-----
sales      75000
```

11. Find the minimum average salary according to jobtypes.

➔ SELECT MIN(AVG(SALARY)) AS MIN\_AVG\_SALARY FROM EMPLOYEE GROUP BY JOB\_TYPES;

```
SQL> SELECT MIN(AVG(SALARY)) AS MIN_AVG_SALARY FROM EMPLOYEE GROUP BY JOB_TYPES;

MIN_AVG_SALARY
-----
          20000
```

12. Find the employee name and date of joining who are working in delhi.

➔ SELECT e.FIRST\_NAME || ' ' || e.LAST\_NAME AS EMPLOYEE\_NAME, e.DOJ FROM EMPLOYEE e JOIN DEPARTMENT d ON e.DEPT = d.D\_NAME WHERE d.D\_LOC = 'delhi';

```
SQL> SELECT e.FIRST_NAME || ' ' || e.LAST_NAME AS EMPLOYEE_NAME, e.DOJ FROM EMPLOYEE e JOIN DEPARTMENT d ON e.DEPT = d.D_NAME WHERE d.D_LOC = 'delhi';

EMPLOYEE_NAME      DOJ
-----
Floki dutt         16-JUL-00
dheeraj kumar      01-JUL-16
saul good          06-SEP-14
bobby deol         17-OCT-17
```

13. Create the table 'Emp\_Address' for storing the permanent address of the employees and insert the values.

EMP_ID	CITY	DISTRICT	STATE
1	Suri	Birbhum	WB
3	Kolkata	Kolkata	WB
4	Bhubaneswar	Khurda	Odisha
5	Noida	GB Nagar	UP
6	Secunderabad	Hyderabad	Telangana
7	Derhadun	Derhadun	Uttarakhand
8	Asansol	Burdwan	WB
9	Siliguri	Darjeeling	WB
10	Kolkata	Kolkata	WB
11	New delhi	New delhi	Delhi

➔ CREATE TABLE Emp\_Address (emp\_id INT, city VARCHAR(15), district VARCHAR(15), state VARCHAR(15));

INSERT INTO Emp\_Address VALUES (&emp\_id, '&city', '&district', '&state');

```
SQL> INSERT INTO Emp_Address VALUES (&emp_id, '&city', '&district', '&state')
2 ;
Enter value for emp_id: 1
Enter value for city: suri
Enter value for district: birbhum
Enter value for state: wb
old 1: INSERT INTO Emp_Address VALUES (&emp_id, '&city', '&district', '&state')
new 1: INSERT INTO Emp_Address VALUES (1, 'suri', 'birbhum', 'wb')

1 row created.
```

14. Display the employee name with their home city and the city they work in.

➔ `SELECT e.FIRST_NAME || ' ' || e.LAST_NAME AS Employee_Name, a_home.City AS Home_City, d.D_LOC AS Working_City FROM EMPLOYEE e JOIN EMP_ADDRESS a_home ON e.EMP_ID = a_home.EMP_ID JOIN DEPARTMENT d ON e.DEPT = d.D_NAME;`

15. Create the following Job\_Grades table.

Grade	Lowest_Sal	Highest_Sal
A	10000	24999
B	25000	49999
C	50000	100000

➔ `CREATE TABLE JOB_GRADES (GRADE VARCHAR2(5), LOWEST_SAL NUMBER(10), HIGHEST_SAL NUMBER(10));`

`INSERT INTO JOB_GRADES VALUES('A',10000,24999);`  
`INSERT INTO JOB_GRADES VALUES('B',25000,49999);`  
`INSERT INTO JOB_GRADES VALUES('C',50000,100000);`

```
SQL> CREATE TABLE JOB_GRADES (GRADE VARCHAR2(5), LOWEST_SAL NUMBER(10), HIGHEST_SAL NUMBER(10));
Table created.

SQL> INSERT INTO JOB_GRADES VALUES('A',10000,24999);
1 row created.
```

16. Display the employee names along with their salary and job\_grade.

➔ `SELECT e.FIRST_NAME || ' ' || e.LAST_NAME AS EMPLOYEE_NAME, e.SALARY, j.GRADE FROM EMPLOYEE e JOIN JOB_GRADES j ON e.SALARY BETWEEN j.LOWEST_SAL AND j.HIGHEST_SAL;`

```
SQL> SELECT e.FIRST_NAME || ' ' || e.LAST_NAME AS EMPLOYEE_NAME, e.SALARY, j.GRADE FROM EMPLOYEE e JOIN JOB_GRADES j ON
e.SALARY BETWEEN j.LOWEST_SAL AND j.HIGHEST_SAL;

EMPLOYEE_NAME      SALARY GRADE
-----
sunny deol         20000 A
anandi patel       12000 A
dheeraj kumar      40000 B
bobby deol         35000 B
anand patil        28000 B
arun khan          90000 C
barun kumar        80000 C
chitra Kapoor      60000 C
dheeraj mishra     75000 C
emma dutt          55000 C
floki dutt         70000 C
saul good          60000 C

12 rows selected.
```

17. Display the employees name along with their manager's name. (use SELF JOIN)

➔ SELECT e1.FIRST\_NAME AS Employee\_Name, e2.FIRST\_NAME AS Manager\_Name FROM  
EMPLOYEE e1 JOIN EMPLOYEE e2 ON e1.MANAGER\_ID = e2.EMPLOYEE\_ID;

```
SQL> SELECT e1.FIRST_NAME AS Employee_Name, e2.FIRST_NAME AS Manager_Name FROM EMPLOYEE e1 JOIN EMPLOYEE e2 ON e1.MANAGER_ID = e2.EMPLOYEE_ID;

EMPLOYEE_N  MANAGER_NA
-----
anandi      arun
anand       arun
sunny       arun
emma        arun
dheeraj     arun
chitra      arun
dheeraj     floki
bobby       saul

8 rows selected.
```

18. Display emp\_id, f\_name, d\_loc, and hod\_id (using natural join).

➔ SELECT EMPLOYEE.EMPLOYEE\_ID, EMPLOYEE.FIRST\_NAME, DEPARTMENT.D\_LOC,  
DEPARTMENT.HOD\_ID FROM EMPLOYEE NATURAL JOIN DEPARTMENT;

```
SQL> SELECT EMPLOYEE.EMPLOYEE_ID, EMPLOYEE.FIRST_NAME, DEPARTMENT.D_LOC, DEPARTMENT.HOD_ID FROM EMPLOYEE NATURAL JOIN DEPARTMENT;

EMPLOYEE_ID FIRST_NAME D_LOC      HOD_ID
-----
1 arun      kol        4
2 barun     kol        4
3 chitra    kol        4
4 dheeraj   kol        4
5 emma      kol        4
6 floki     kol        4
7 dheeraj   kol        4
8 saul      kol        4
10 sunny    kol        4
11 bobby    kol        4
13 anand    kol        4
14 anandi   kol        4
1 arun      delhi      6
```

19. Display the employees f\_name, city and state in which they live (using natural join).

➔ SELECT EMPLOYEE.FIRST\_NAME, EMP\_ADDRESS.CITY, EMP\_ADDRESS.STATE FROM  
EMPLOYEE NATURAL JOIN EMP\_ADDRESS;

```
SQL> SELECT EMPLOYEE.FIRST_NAME, EMP_ADDRESS.CITY, EMP_ADDRESS.STATE FROM EMPLOYEE NATURAL JOIN EMP_ADDRESS;

FIRST_NAME CITY      STATE
-----
arun      suri      wb
barun     suri      wb
chitra    suri      wb
dheeraj   suri      wb
emma      suri      wb
floki     suri      wb
dheeraj   suri      wb
saul      suri      wb
sunny     suri      wb
bobby     suri      wb
anand     suri      wb
anandi    suri      wb
arun      kolkata  wb
barun     kolkata  wb
```

20. Display the employees emp\_id, f\_name, d\_loc, hod\_id using inner join.

➔ SELECT EMPLOYEE.EMPLOYEE\_ID, EMPLOYEE.FIRST\_NAME, DEPARTMENT.D\_LOC, DEPARTMENT.HOD\_ID FROM EMPLOYEE INNER JOIN DEPARTMENT ON EMPLOYEE.DEPT = DEPARTMENT.D\_NAME;

```
SQL> SELECT EMPLOYEE.EMPLOYEE_ID, EMPLOYEE.FIRST_NAME, DEPARTMENT.D_LOC, DEPARTMENT.HOD_ID FROM EMPLOYEE INNER JOIN DEPARTMENT ON EMPLOYEE.DEPT = DEPARTMENT.D_NAME;
```

EMPLOYEE_ID	FIRST_NAME	D_LOC	HOD_ID
1	arun	kol	1
3	chitra	kol	1
4	dheeraj	kol	4
5	emma	kol	1
6	floki	delhi	6
7	dheeraj	delhi	6
8	saul	delhi	8
11	bobby	delhi	8
13	anand	mumbai	5

9 rows selected.

21. Display the employees f\_name, city and state in which they live (using inner join).

➔ SELECT EMPLOYEE.FIRST\_NAME, EMP\_ADDRESS.CITY, EMP\_ADDRESS.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 EMPLOYEE.FIRST\_NAME, EMP\_ADDRESS.CITY, EMP\_ADDRESS.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.

EMP_ID	F_NAME	L_NAME	JOB_TYPE	SALARY	COMMISION	D_NAME	MANAGER_ID	DOJ
20	alex		engineer	28000	2000		1	31-JAN-17
21	priya	patel	clerk	12000	500		1	01-APR-17

➔ INSERT INTO EMPLOYEE VALUES(20,'ALEX',NULL,'ENGINEER',28000,2000,NULL,1,TO\_DATE('31-JAN-2017','DD-MM-YYYY'));

INSERT INTO EMPLOYEE VALUES(21,'PRIYA','PATEL','CLERK',12000,500,NULL,1,TO\_DATE('01-APR-2017','DD-MM-YYYY'));

```
SQL> INSERT INTO EMPLOYEE VALUES(20, 'ALEX', NULL, 'ENGINEER', 28000, 2000, NULL, 1, TO_DATE('31-JAN-2017', 'DD-MM-YYYY'));
1 row created.

SQL> INSERT INTO EMPLOYEE VALUES(21, 'PRIYA', 'PATEL', 'CLERK', 12000, 500, NULL, 1, TO_DATE('01-APR-2017', 'DD-MM-YYYY'));
1 row created.
```

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



D_NAME	D_LOC	HOD_ID
Training	Mumbai	1
Placement	Mumbai	1

➔ INSERT INTO DEPARTMENT VALUES('TRAINING','MUMBAI',1);

INSERT INTO DEPARTMENT VALUES('PLACEMENT','MUMBAI',1);

```
SQL> INSERT INTO DEPARTMENT VALUES('TRAINING','MUMBAI',1);

1 row created.

SQL> INSERT INTO DEPARTMENT VALUES('PLACEMENT','MUMBAI',1);

1 row created.
```

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 EMPLOYEE.FIRST\_NAME, EMP\_ADDRESS.CITY, EMP\_ADDRESS.STATE FROM EMPLOYEE LEFT 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 EMPLOYEE.FIRST\_NAME, DEPARTMENT.D\_LOC AS WORK\_LOCATION FROM EMPLOYEE LEFT JOIN DEPARTMENT ON EMPLOYEE.DEPT = DEPARTMENT.D\_NAME;

```
SQL> SELECT EMPLOYEE.FIRST_NAME, DEPARTMENT.D_LOC AS WORK_LOCATION FROM EMPLOYEE LEFT JOIN DEPARTMENT ON EMPLOYEE.DEPT = DEPARTMENT.D_NAME;

FIRST_NAME WORK_LOCAT
-----
dheeraj     kol
dheeraj     delhi
floki       delhi
emma        kol
chitra      kol
arun        kol
bobby       delhi
saul        delhi
anand       mumbai
PRIYA
ALEX
anandi
sunny
barun

14 rows selected.
```

27. Display the employees f\_name and their work location after joining employee and department table using right join.

➔ SELECT EMPLOYEE.FIRST\_NAME, DEPARTMENT.D\_LOC AS WORK\_LOCATION FROM EMPLOYEE RIGHT JOIN DEPARTMENT ON EMPLOYEE.DEPT = DEPARTMENT.D\_NAME;

```
SQL> SELECT EMPLOYEE.FIRST_NAME, DEPARTMENT.D_LOC AS WORK_LOCATION FROM EMPLOYEE RIGHT JOIN DEPARTMENT ON EMPLOYEE.DEPT = DEPARTMENT.D_NAME;
```

FIRST_NAME	WORK_LOCAT
arun	kol
chitra	kol
dheeraj	kol
emma	kol
floki	delhi
dheeraj	delhi
saul	delhi
bobby	delhi
anand	mumbai
	MUMBAI
	mumbai
	MUMBAI
	kol

13 rows selected.

28. Display the employees f\_name and their work location after joining employee and department table using full join/full outer join.

➔ SELECT EMPLOYEE.FIRST\_NAME, DEPARTMENT.D\_LOC AS WORK\_LOCATION FROM EMPLOYEE FULL JOIN DEPARTMENT ON EMPLOYEE.DEPT = DEPARTMENT.D\_NAME;

```
SQL> SELECT EMPLOYEE.FIRST_NAME, DEPARTMENT.D_LOC AS WORK_LOCATION FROM EMPLOYEE FULL JOIN DEPARTMENT ON EMPLOYEE.DEPT = DEPARTMENT.D_NAME;
```

FIRST_NAME	WORK_LOCAT
dheeraj	kol
dheeraj	delhi
floki	delhi
emma	kol
chitra	kol
arun	kol
bobby	delhi
saul	delhi
anand	mumbai
PRIYA	
ALEX	
anandi	
sunny	
barun	
	MUMBAI
	mumbai
	MUMBAI
	kol

18 rows selected.

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

➔ SELECT e.EMP\_ID, e.FIRST\_NAME, e.LAST\_NAME FROM EMPLOYEE e JOIN DEPARTMENT d ON e.DEPT = d.D\_NAME JOIN EMP\_ADDRESS ea ON e.EMP\_ID = ea.EMP\_ID WHERE d.D\_LOC = ea.CITY;

30. Find the job type having the minimum average salary according to jobtypes.

➔ `SELECT JOB_TYPE, AVG(SALARY) AS AVERAGE_SALARY FROM EMPLOYEE GROUP BY  
JOB_TYPE ORDER BY AVG(SALARY) ASC FETCH FIRST 1 ROW ONLY;`