

**Experiment no: - 3**

**Experiment Name: - Ret rival Commands-II**

**Aim: -** Performing practical by using multiple tables and group functions.

**Resource required: -** Oracle 9i - iSQLplus

**Theory: -**

- **DISPLAYING DATA FROM MULTIPLE TABLES:**

**Cartesian Products:** It is formed when:

- a join condition is omitted
- a join condition is invalid
- All rows in the first table are joined to all rows in the second table.

**Syntax:** Joining tables

SELECT table1.column, table2.column

FROM table1, table2

WHERE table1.column1= table2.column2;

- write the join condition in the WHERE clause.

1. Retrieving record with **Equijoin/simple join/ inner join:**

e.g.: SELECT employees.employee\_id, employees.last\_name,  
employees.department\_id, departments.department\_id,  
departments.location\_id  
FROM employees, departments  
WHERE employees.department\_id = departments.department\_id;

EMPLOYEE_ID	LAST_NAME	DEPARTMENT-ID	DEPARTMENT-ID	LOCATION_ID
200	Whalen	10	10	1700
201	Hartstein	20	20	1800
202	Fay	30	30	1500

2. Retrieving record with **Non-Equijoins:** contain something other than equality operator.

e.g.: SELECT e.last\_name, e.salary, j.grade\_level  
FROM employees e, job\_grades j  
WHERE e.salary  
BETWEEN j.lowest\_sal AND j.highest\_sal;

LAST_NAME	SALARY	GRA
Matros	2500	A
Vargas	2600	A
Lorentz	4200	B

3. **Outer join:** the missing rows can be returned if an outer join operator (+) is used in join condition.

**Syntax:** SELECT table1.column, table2.column  
FROM table1, table2  
WHERE table1.column (+) = table2.column;

e.g.: SELECT e.last\_name, e.department\_id, d.department\_name  
FROM employees e, departments d  
WHERE e.department\_id(+) = d.department\_id;

LAST_NAME	DEPARTMENT-ID	DEPARTMENT_NAME
Whalen	10	Administration
Matos	50	Shipping
		Contracting

>> LEFT OUTER JOIN

e.g.: SELECT e.last\_name, e.department\_id, and d.department\_name  
FROM employees e  
LEFT OUTER JOIN departments d  
ON (e. department\_id = d.department\_id);  
- ON clause used to join specify column.

>> RIGHT OUTER JOIN

e.g.: SELECT e.last\_name, e.department\_id, d.department\_name  
FROM employees e  
RIGHT OUTER JOIN departments d  
ON (e.department\_id = d.department\_id);

>> FULL OUTER JOIN

e.g.: SELECT e.last\_name, e.department\_id, and d.department\_name  
FROM employees e  
FULL OUTER JOIN departments d  
ON (e. department\_id = d.department\_id);

4. **Cross joins:** produces the cross product of two tables.

e.g.: SELECT last\_name, department\_name  
FROM employees  
CROSS JOIN departments;

LAST_NAME	DEPARTMENT_NAME
King	Administration
Kochhar	Administration
De Haan	Administration

5. **Natural Joins:** select record from tables that have equal values in all matched columns.

e.g.: SELECT department\_id, department\_name  
FROM departments  
NATURAL JOIN locations;

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
60	IT	1400	Southlake
50	Shipping	1500	Seattle
90	Executive	1700	Seattle

- **AGGREGATING DATA USING GROUP FUNCTIONS:**

**Group Functions:** It operate on sets of rows to give one result per group.

**Syntax:** SELECT [column,] group\_function (column), ...  
FROM table  
[WHERE condition]  
[GROUP BY column]  
[ORDER BY column];

- Types of Group Functions:

1. AVG, COUNT, MAX, MIN, STDDEV, SUM, and VARIANCE

e.g.: SELECT AVG (salary), MAX (salary), MIN (salary), and SUM (salary)  
FROM employees  
WHERE job\_id LIKE ‘, %REP%’;

AVG (SALARY)	MAX (SALARY)	MIN (SALARY)	SUM (SALARY)
8150	11000	6000	32600

2. COUNT (\*): returns the number of rows in a table.

e.g.: SELECT COUNT (\*)  
FROM employees  
WHERE department\_id = 50;

3. DISTINCT: returns the number of distinct non-null values.

e.g.: SELECT COUNT (DISTINCT department\_id)  
FROM employees;

Others function with group functions:

- NVL function: forces group functions to include null values.

e.g.: SELECT AVG (NVL (commission\_pct, 0))  
FROM employees;

- Creating groups of data using **GROUP BY** clause:

**Syntax:** SELECT column, group\_function (column)  
From table  
[WHERE condition]  
[GROUP BY group\_by\_expression]  
[ORDER BY column];

e.g.: SELECT AVG (salary)  
FROM employees

GROUP BY department\_id;

- The **HAVING** Clause: to restrict groups Syntax:  
SELECT column,  
group\_function  
From  
table  
[WH  
ERE  
condi  
tion]  
[GROUP BY  
group\_  
by\_expression]  
[HAVING  
group\_condition]  
[ORDER BY column];

e.g.: SELECT department\_id,  
MAX (salary) FROM  
employees  
GROUP BY  
department\_id  
HAVING MAX  
(salary)>10000;

- Displays department numbers and maximum salaries for those departments whose maximum salary is greater than \$10,000.

- Nesting Group Functions:  
e.g.: SELECT MAX (AVG (salary))  
FROM employees  
GROUP BY department\_id;

### **Conclusion:**

You should have learned how to use joins to display data from multiple tables and group functions.

## LAB ASSIGNMENT -3

1. Write a query to display the employee last name, department name, location ID, and city of all employees who earn a commission

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area displays an SQL Worksheet with the following query:

```
1 SELECT e.last_name, d.department_name, d.location_id, l.city
2 FROM employees e, departments d, locations l
3 WHERE e.department_id = d.department_id
4 AND
5 d.location_id = l.location_id
6 AND e.commission_pct IS NOT NULL;
7
```

Below the query, the results are displayed in a table:

LAST_NAME	DEPARTMENT_NAME	LOCATION_ID	CITY
Zlotkey	Sales	2500	Oxford
Abel	Sales	2500	Oxford
Taylor	Sales	2500	Oxford

A "Download CSV" button is located below the table. The footer indicates: "2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with ❤️ using Oracle APEX - Privacy - Terms of Use".

2. Create a unique listing of all jobs that are in department 80. Include the location of the department in the output.

The screenshot shows the Oracle Live SQL interface. The left sidebar contains navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area displays an SQL Worksheet with the following query:

```
1 SELECT DISTINCT job_id, location_id
2 FROM employees, departments
3 WHERE employees.department_id = departments.department_id
4 AND employees.department_id = 80;
5
```

Below the query, the results are displayed in a table:

JOB_ID	LOCATION_ID
SA_MAN	2500
SA_REP	2500

A "Download CSV" button is located below the table. Below the table, it says "2 rows selected.". The footer indicates: "2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with ❤️ using Oracle APEX - Privacy - Terms of Use".

- Display the employee last name and department name for all employees who have an a (lowercase) in their last names.

The screenshot shows the Live SQL interface. The left sidebar contains navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains the following SQL query:

```
1 SELECT last_name, department_name
2 FROM employees, departments
3 WHERE employees.department_id = departments.department_id
4 AND last_name LIKE 'a%';
5
```

Below the query editor, the results are displayed in a table:

LAST_NAME	DEPARTMENT_NAME
Kochhar	Executive
De Haan	Executive
Rajs	Shipping
Davies	Shipping

The footer of the interface indicates: 2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with using Oracle APEX - Privacy - Terms of Use.

- Create a query to display the name and hire date of any employee hired after employee Davies.

The screenshot shows the Live SQL interface with the same sidebar as the previous image. The main area is titled 'SQL Worksheet' and contains the following SQL query:

```
1 SELECT e.last_name, e.hire_date
2 FROM employees e, employees davies
3 WHERE davies.last_name = 'Davies'
4 AND davies.hire_date < e.hire_date
5
6
```

Below the query editor, the results are displayed in a table:

LAST_NAME	HIRE_DATE
Lorentz	07-FEB-99
Mourgos	16-NOV-99
Matos	15-MAR-98
Vargas	09-JUL-98
Zlotkey	09-JUL-98
Abel	09-JUL-98
Taylor	09-JUL-98
Grant	04-MAY-99
Fay	17-AUG-97

The footer of the interface indicates: 2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with using Oracle APEX - Privacy - Terms of Use.

5. Display the names and hire date for all employees who were hired before their managers, along with their manager's names and hire dates. Label the column Employee, EMP, Hired, Manager, and Mgr Hired, respectively.

The screenshot shows the Live SQL interface. The SQL Worksheet contains the following query:

```
1 SELECT w.last_name, w.hire_date, m.last_name, m.hire_date
2 FROM employees w, employees m
3 WHERE w.manager_id = m.employee_id
4 AND w.hire_date < m.hire_date;
5
```

The query results are displayed in a table with 4 columns: LAST\_NAME, HIRE\_DATE, LAST\_NAME, HIRE\_DATE. There are 6 rows of data.

LAST_NAME	HIRE_DATE	LAST_NAME	HIRE_DATE
Whalen	17-SEP-87	Kochhar	21-SEP-89
Hunold	03-JAN-90	De Haan	13-JAN-93
Rajs	17-OCT-95	Moungos	16-NOV-99
Davies	29-JAN-97	Moungos	16-NOV-99
Matos	15-MAR-98	Moungos	16-NOV-99
Vargas	09-JUL-98	Moungos	16-NOV-99

Download CSV

6 rows selected.

2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

6. Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

The screenshot shows the Live SQL interface. The SQL Worksheet contains the following query:

```
1 SELECT ROUND(MAX(salary),0) "Maximum",
2 ROUND(MIN(salary),0) "Minimum",
3 ROUND(SUM(salary),0) "Sum",
4 ROUND(AVG(salary),0) "Average"
5 FROM employees;
6
7
```

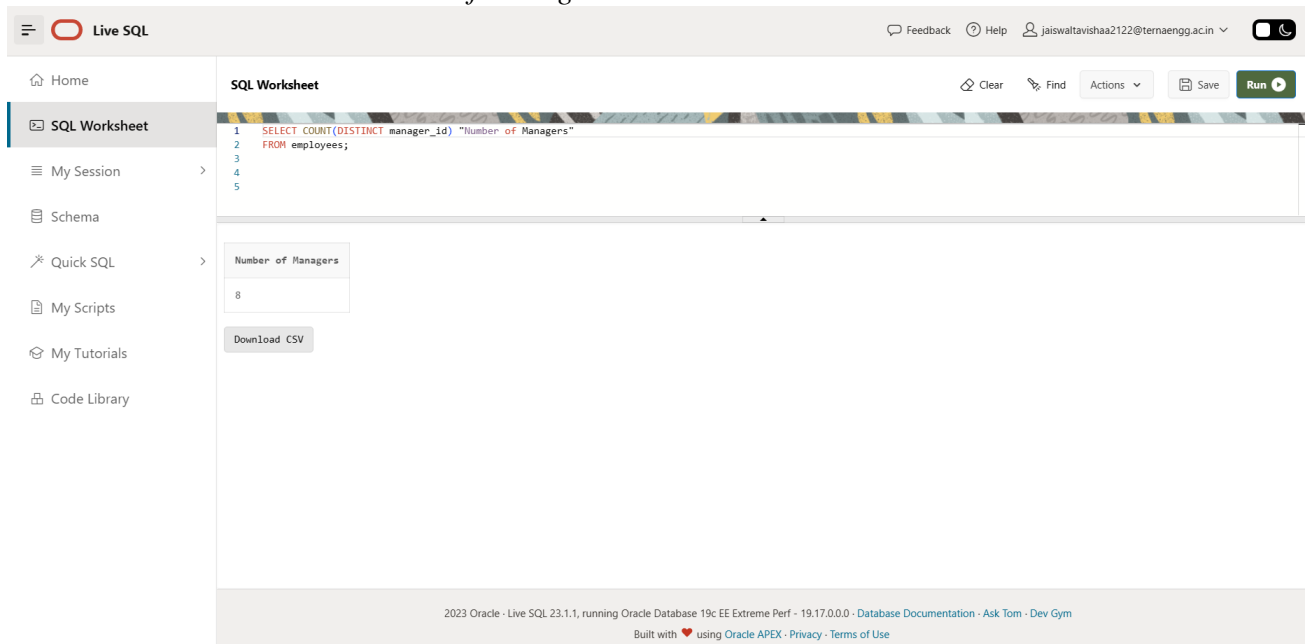
The query results are displayed in a table with 4 columns: Maximum, Minimum, Sum, Average. There is 1 row of data.

Maximum	Minimum	Sum	Average
42000	2500	213340	10667

Download CSV

2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

7. Determine the number of managers without listing them. Label the column Number of Managers. *Hint: Use the MANAGER\_ID column to determine the number of managers.*



The screenshot shows the Live SQL interface. The SQL Worksheet contains the following query:

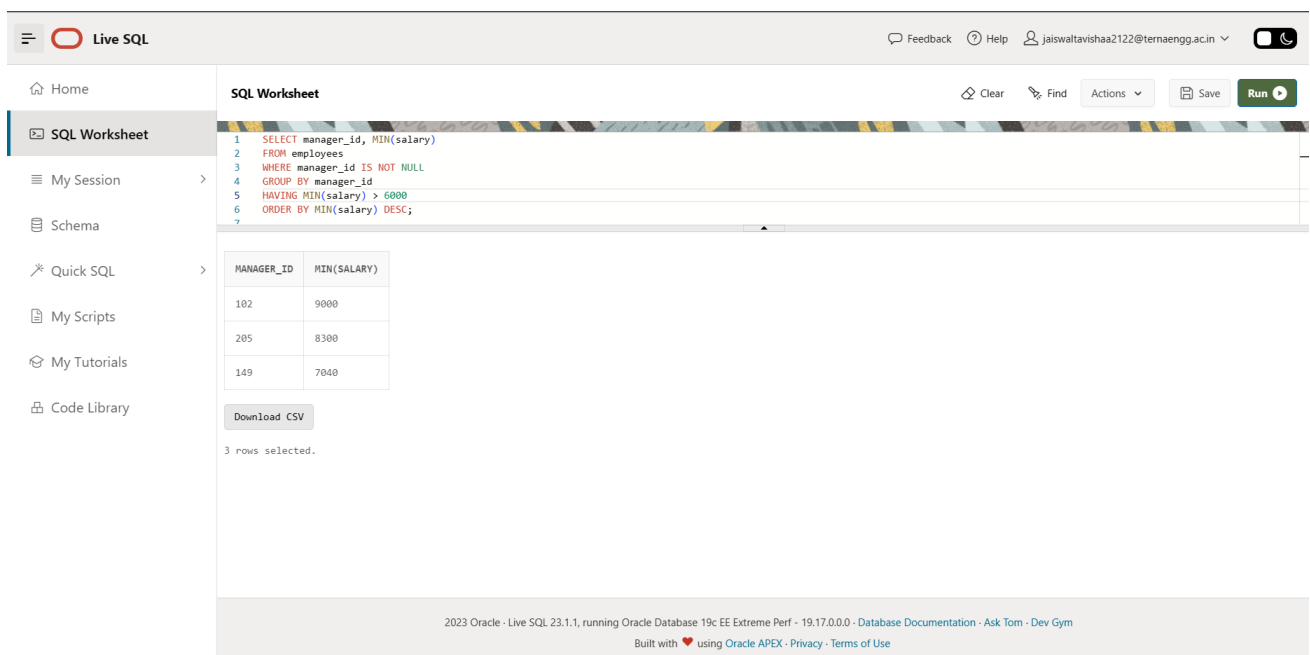
```
1 SELECT COUNT(DISTINCT manager_id) "Number of Managers"
2 FROM employees;
3
4
5
```

The result table displays the following data:

Number of Managers
8

Below the table is a "Download CSV" button. The footer indicates: "2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with using Oracle APEX - Privacy - Terms of Use".

8. Display the manager number and the salary of the lowest paid employee for that manager. Exclude anyone whose manager is not know. Exclude any group where the minimum salary is \$6,000 or less. Sort the output in descending order of salary.



The screenshot shows the Live SQL interface. The SQL Worksheet contains the following query:

```
1 SELECT manager_id, MIN(salary)
2 FROM employees
3 WHERE manager_id IS NOT NULL
4 GROUP BY manager_id
5 HAVING MIN(salary) > 6000
6 ORDER BY MIN(salary) DESC;
7
```

The result table displays the following data:

MANAGER_ID	MIN(SALARY)
102	9000
205	8300
149	7040

Below the table is a "Download CSV" button. The text "3 rows selected." is displayed. The footer indicates: "2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with using Oracle APEX - Privacy - Terms of Use".



9. Write a query to display each department's name, location, number of employees, and the average salary for all employees in that department. Label the columns Name, Location, Number of People, and Salary, respectively. Round the average salary to two decimal places.

Live SQL

Feedback Help jaiswalthavishaa2122@ternaengg.ac.in

Home SQL Worksheet Clear Find Actions Save Run

```

1 select d.department_name "NAME", d.location_id "LOCATIONS", count(*) "NUMBER OF PEOPLE", round(avg(e.salary), 2) "SAL"
2 from departments d, employees e
3 where e.department_id = d.department_id group by d.department_name, d.location_id;

```

NAME	LOCATIONS	NUMBER OF PEOPLE	SAL
Accounting	1700	2	10150
Sales	2500	3	10033.33
IT	1400	3	19000
Administration	1700	1	4400
Executive	1700	3	19333.33
Shipping	1500	5	3500
Marketing	1800	2	9500

Download CSV

7 rows selected.

2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with using Oracle APEX - Privacy - Terms of Use

10. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.

Live SQL

Feedback Help jaiswalthavishaa2122@ternaengg.ac.in

Home SQL Worksheet Clear Find Actions Save Run

```

1 SELECT job_id "Job",
2 SUM(DECODE(department_id, 20, salary)) "Dept 20",
3 SUM(DECODE(department_id, 50, salary)) "Dept 50",
4 SUM(DECODE(department_id, 80, salary)) "Dept 80",
5 SUM(DECODE(department_id, 90, salary)) "Dept 90",
6 SUM(salary) "Total" FROM employees
7 GROUP BY job_id;

```

Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
AD_VP	-	-	-	34000	34000
AC_MGR	-	-	-	-	12000
ST_CLERK	-	11700	-	-	11700
AD_ASST	-	-	-	-	4400
IT_PROG	-	-	-	-	57000
SA_MAN	-	-	10500	-	10500
AC_ACCOUNT	-	-	-	-	8300
ST_MAN	-	5800	-	-	5800
AD_PRES	-	-	-	24000	24000

2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with using Oracle APEX - Privacy - Terms of Use

Live SQL

FeedbackHelpjaiswalthavishaa2122@ternaengg.ac.in

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

ClearFindActionsSaveRun

```
1 SELECT job_id "Job",
2 SUM(DECODE(department_id , 20, salary)) "Dept 20",
3 SUM(DECODE(department_id , 50, salary)) "Dept 50",
4 SUM(DECODE(department_id , 80, salary)) "Dept 80",
5 SUM(DECODE(department_id , 90, salary)) "Dept 90",
6 SUM(salary) "Total"FROM employees
7 GROUP BY job_id;
```

IT_PROG	-	-	-	-	57000
SA_MAN	-	-	10500	-	10500
AC_ACCOUNT	-	-	-	-	8300
ST_MAN	-	5800	-	-	5800
AD_PREP	-	-	-	24000	24000
MK_MAN	13000	-	-	-	13000
SA_REP	-	-	19600	-	26640
MK_REP	6000	-	-	-	6000

Download CSV

12 rows selected.

2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym  
Built with using Oracle APEX - Privacy - Terms of Use