

DBMS ASSIGNMENT

WEEK – 6

1)Group_By Clause

```
SQL> SELECT dept, SUM(salary) AS total_salary
  2  FROM employee503305
  3  GROUP BY dept;
```

DEPT	TOTAL_SALARY
Manager	70000
Designer	9000
Developer	100000
Testing	90000

```
SQL> SELECT dept, COUNT(*) AS emp_count
  2  FROM employee503305
  3  GROUP BY dept;
```

DEPT	EMP_COUNT
Manager	1
Designer	1
Developer	2
Testing	4

2)Subqueries

```
SQL> SELECT fname, lname, salary
  2  FROM employee503305
  3  WHERE salary > (
  4      SELECT AVG(salary)
  5      FROM employee503305
  6  );
```

FNAME	LNAME	SALARY
Marish	Dhanush	50000
Marish	Kumar	50000
Deepak	M	70000

```
SQL> SELECT fname, lname, salary
  2  FROM employee503305
  3  WHERE salary = (
  4      SELECT MAX(salary)
  5      FROM employee503305
  6  );
```

FNAME	LNAME	SALARY
Deepak	M	70000

3)Joins

```
SQL> SELECT e.fname, e.lname, e.dept, d.location
  2  FROM employee503305 e
  3  JOIN dept503305 d
  4  ON e.dept = d.dept;
```

FNAME	LNAME	DEPT	LOCATION
Satlas	Rohit	Testing	Chennai
Satlas	Rohit	Testing	Chennai
Ajay	Kumar	Testing	Chennai
Srini	Vasan	Testing	Chennai
Marish	Dhanush	Developer	Bangalore
Marish	Kumar	Developer	Bangalore
Deepak	M	Manager	Mumbai
Vasu	Deva	Designer	Hyderabad

8 rows selected.

```
SQL> SELECT e.fname, e.salary, d.location
  2  FROM employee503305 e
  3  JOIN dept503305 d
  4  ON e.dept = d.dept
  5  WHERE d.location = 'Chennai';
```

FNAME	SALARY	LOCATION
Satlas	20000	Chennai
Satlas	20000	Chennai
Ajay	30000	Chennai
Srini	20000	Chennai

4)Join Insert

```
SQL> INSERT INTO bonus503305 (emp_id, bonus)
  2  SELECT e.emp_id, e.salary * 0.10
  3  FROM employee503305 e
  4  JOIN dept503305 d
  5  ON e.dept = d.dept
  6  WHERE d.dept = 'Developer';
```

2 rows created.

5)Join Update

```
SQL> UPDATE employee503305 e
  2  SET salary = salary + 2000
  3  WHERE dept IN (
  4      SELECT dept
  5      FROM dept503305
  6      WHERE location = 'Chennai'
  7  );
```

4 rows updated.

6)Join Delete

```
SQL> DELETE FROM employee503305
2 WHERE dept IN (
3 SELECT dept
4 FROM dept503305
5 WHERE location = 'Hyderabad'
6 );

1 row deleted.
```

PL/SQL

Anomaly3305.sql

```
DECLARE
    v_salary employee503305.salary%TYPE;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Satlas – 2024503305');
    SELECT salary INTO v_salary
    FROM employee503305
    WHERE emp_id = 1;
    DBMS_OUTPUT.PUT_LINE('Salary = ' || v_salary);
END;
/
```

Output:

```
SQL> @D:/SQL/Anomaly3305;

PL/SQL procedure successfully completed.
```

Procedure3305.sql

```
CREATE OR REPLACE PROCEDURE get_emp_salary503305(p_id NUMBER)
IS
    v_salary employee503305.salary%TYPE;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Satlas – 2024503305');
    SELECT salary INTO v_salary
    FROM employee503305
    WHERE emp_id = p_id;
    DBMS_OUTPUT.PUT_LINE('Salary = ' || v_salary);
END;
/
EXEC get_emp_salary503305(3);
```

Output:

```
SQL> @D:/SQL/Procedure3305;

Procedure created.

Satlas 2024503305
Salary = 32000

PL/SQL procedure successfully completed.
```

Cursor3305.sql

```
DECLARE
  CURSOR emp_cursor IS
    SELECT fname, salary FROM employee503305;
  v_name employee503305.fname%TYPE;
  v_salary employee503305.salary%TYPE;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Satlas – 2024503305');
  OPEN emp_cursor;
  DBMS_OUTPUT.PUT_LINE('Name' || ' ' || 'Salary');
  LOOP
    FETCH emp_cursor INTO v_name, v_salary;
    EXIT WHEN emp_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(v_name || ' → ' || v_salary);
  END LOOP;
  CLOSE emp_cursor;
END;
/
```

Output:

```
SQL> @D:/SQL/Cursor3305;
Satlas 2024503305
Name Salary
Satlas 22000
Satlas 22000
Ajay 32000
Srini 22000
Marish 50000
Marish 50000
Deepak 70000

PL/SQL procedure successfully completed.
```

Insert_Trig3305.sql

```
CREATE OR REPLACE TRIGGER trg_insert503305
BEFORE INSERT ON employee503305
FOR EACH ROW
BEGIN
    DBMS_OUTPUT.PUT_LINE('Satlas – 2024503305');
    DBMS_OUTPUT.PUT_LINE('New employee inserted');
END;
/
```

Output:

```
SQL> @D:/SQL/Insert_Trig3305;

Trigger created.

SQL> INSERT INTO EMPLOYEE503305 VALUES(
  2  'Ram',
  3  'Kumar',
  4  11,
  5  15000,
  6  'Assistant'
  7  );
Satlas  2024503305
New employee inserted

1 row created.
```

Update_Trig3305.sql

```
CREATE OR REPLACE TRIGGER trg_update503305
BEFORE UPDATE ON employee503305
FOR EACH ROW
BEGIN
    DBMS_OUTPUT.PUT_LINE('Satlas 2024503305');
    DBMS_OUTPUT.PUT_LINE('Salary updated');
END;
/
```

Output:

```
SQL> @D:/SQL/Update_Trig3305;

Trigger created.

SQL> Update Employee503305 set Salary=20000 Where Emp_id=2;
Satlas  2024503305
Salary updated

1 row updated.
```

Delete_Trig3305.sql

```
CREATE OR REPLACE TRIGGER trg_delete503305
BEFORE DELETE ON employee503305
FOR EACH ROW
BEGIN
    DBMS_OUTPUT.PUT_LINE('Satlas 2024503305');
    DBMS_OUTPUT.PUT_LINE('Employee deleted');
END;
/
```

Output:

```
SQL> @D:/SQL/Delete_Trig3305;

Trigger created.

SQL> DELETE FROM EMPLOYEE503305 WHERE Emp_id=11;
Satlas 2024503305
Employee deleted

1 row deleted.
```

IF_Statement3305.sql

```
DECLARE
    v_salary NUMBER := 30000;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Satlas 2024503305');

    IF v_salary > 25000 THEN
        DBMS_OUTPUT.PUT_LINE('High Salary');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Low Salary');
    END IF;
END;
/
```

Output:

```
SQL> @D:/SQL/IF_Statement3305;
Satlas 2024503305
High Salary

PL/SQL procedure successfully completed.
```

LOOP_Statement3305.sql;

```
DECLARE
    i NUMBER := 1;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Satlas 2024503305');

    LOOP
        DBMS_OUTPUT.PUT_LINE('Value: ' || i);
        i := i + 1;
        EXIT WHEN i > 5;
    END LOOP;
```

```
END LOOP;  
END;  
/
```

Output:

```
SQL> @D:/SQL/LOOP_Statement3305;  
Satlas  2024503305  
Value: 1  
Value: 2  
Value: 3  
Value: 4  
Value: 5  
  
PL/SQL procedure successfully completed.
```

[WHILE_Statement3305.sql](#)

```
DECLARE  
  i NUMBER := 1;  
BEGIN  
  DBMS_OUTPUT.PUT_LINE('Satlas 2024503305');  
  
  WHILE i <= 5 LOOP  
    DBMS_OUTPUT.PUT_LINE('Value: ' || i);  
    i := i + 1;  
  END LOOP;  
END;  
/
```

Output:

```
SQL> @D:/SQL/WHILE_Statement3305;  
Satlas  2024503305  
Value: 1  
Value: 2  
Value: 3  
Value: 4  
Value: 5  
  
PL/SQL procedure successfully completed.
```

[FOR_Statement3305.sql](#)

```
BEGIN  
  DBMS_OUTPUT.PUT_LINE('Satlas 2024503305');  
  
  FOR i IN 1..5 LOOP  
    DBMS_OUTPUT.PUT_LINE('Value: ' || i);  
  END LOOP;  
END;  
/
```

Output:

```
SQL> @D:/SQL/FOR_Statement3305;
Satlas  2024503305
Value: 1
Value: 2
Value: 3
Value: 4
Value: 5

PL/SQL procedure successfully completed.
```

CASE_Statement3305.sql

```
DECLARE
  v_dept VARCHAR2(20) := 'Manager';
BEGIN
  DBMS_OUTPUT.PUT_LINE('Satlas 2024503305');

  CASE v_dept
    WHEN 'Manager' THEN DBMS_OUTPUT.PUT_LINE('Manager Dept');
    WHEN 'Developer' THEN DBMS_OUTPUT.PUT_LINE('Developer Dept');
    ELSE DBMS_OUTPUT.PUT_LINE('Other Dept');
  END CASE;
END;
/
```

Output:

```
SQL> @D:/SQL/CASE_Statement3305;
Satlas  2024503305
Manager Dept

PL/SQL procedure successfully completed.
```