**Ex. No: 3 AGGREGATE FUNCTIONS**

**Date:**

# AIM:

To write SQL queries to perform aggregate functions.

# AGGREGATE FUNCTIONS

* 1. Count
  2. Max
  3. Min
  4. Sum
  5. Avg

***// 1. COUNT - displays total number of rows***

SELECT COUNT(examid) AS STUDENTS\_REGISTERED FROM exam;

## Output:

**STUDENTS\_REGISTERED**

3

***// 2. MAX - displays the maximum value*** SELECT MAX(average) AS RANK\_1 FROM exam; **Output:**

**RANK\_1**

90.8

***// 3. MIN - displays the minimum value***

SELECT MIN(average) AS LAST\_RANK FROM exam;

## Output:

**LAST\_RANK**

87.6

***// 4. SUM - displays the total value***

SELECT department, SUM(total) AS SUM\_DEPARTMENT FROM exam GROUP BY department;

## Output:

|  |  |
| --- | --- |
| **DEPARTMENT** | **SUM\_DEPARTMENT** |
| IT 1346 | |

***// 5. AVG - displays the average value***

SELECT department, AVG(total) AS AVERAGE FROM exam GROUP BY department;

**Output:**

|  |  |
| --- | --- |
| **DEPARTMENT** | **AVERAGE** |
| IT 448.666666666666666666666666666666666667 | |

# RESULT:

Thus the SQL queries using aggregate functions were successfully executed and verified.

**Ex. No: 4 SQL JOINS**

**Date:**

# AIM:

To write SQL queries using joins to combine two or more tables.

# JOIN TYPES

1. INNER JOIN
2. OUTER JOIN
   1. LEFT OUTER JOIN
   2. RIGHT OUTER JOIN
   3. FULL OUTER JOIN
3. SELF JOIN
4. EQUI JOIN
5. SELF JOIN

## Table Creation Query:

CREATE TABLE itstudent (

);

## Output:

studentID NUMBER PRIMARY KEY, sname VARCHAR(30),

department CHAR(5), sem NUMBER

## Query:

*Table Created*

CREATE TABLE placement (

PlacementID NUMBER PRIMARY KEY,

StudentID NUMBER references itstudent(studentid), department CHAR(5),

Company VARCHAR2(30), salary NUMBER

);

## Output:

*Table Created*

## Insert into the table Query:

INSERT INTO itstudent VALUES(101,'reema', 'IT',5);

## Output:

*1 row(s) inserted.*

## Query:

INSERT INTO itstudent VALUES(102,'reenu', 'IT',3);

## Output:

*1 row(s) inserted*

## Query:

INSERT INTO itstudent VALUES(103,'sheela', 'CSE',3);

## Output:

*1 row(s) inserted*

## Query:

INSERT INTO itstudent VALUES(104,'nirmal', 'IT',3);

## Output:

*1 row(s) inserted*

## Query:

INSERT INTO itstudent VALUES(105,'eshwar', 'CSE',5);

## Output:

*1 row(s) inserted*

## Query:

INSERT INTO placement VALUES(1, 104, 'IT', 'infosys', 25000);

## Output:

*1 row(s) inserted.*

## Query:

INSERT INTO placement VALUES(2, 105, 'CSE', 'Wipro', 22000);

## Output:

*1 row(s) inserted*

## Query:

INSERT INTO placement VALUES(4, 102, 'IT', 'infosys', 25000);

## Output:

*1 row(s) inserted*

## Query:

INSERT INTO placement VALUES(5, 103, 'CSE', 'infosys', 25000);

## Output:

*1 row(s) inserted*

## Select the table: Query:

SELECT \* FROM itstudent;

## Output:

|  |  |  |  |
| --- | --- | --- | --- |
| **STUDENTID** | **SNAME** | **DEPARTMENT** | **SEM** |
| 101 | rupesh | IT | 5 |
| 102 | bala | IT | 3 |
| 103 | sheela | CSE | 3 |
| 104 | nirmal | IT | 3 |
| 105 | eshwar | CSE | 5 |

**Query:**

SELECT \* FROM placement;

**Output:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PLACEMENTID** | **STUDENTID** | **DEPARTMENT** | **COMPANY** | **SALARY** |
| 1 | 104 | IT | infosys | 25000 |
| 2 | 105 | CSE | Wipro | 22000 |
| 3 | 102 | IT | infosys | 25000 |
| 4 | 103 | CSE | infosys | 25000 |

# INNER JOIN

## Query:

SELECT \*

FROM itstudent INNER JOIN Placement

ON itstudent.studentID=placement.StudentID;

## Output:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STUD** |  | | | | | | | |
| **ENTI** | **SNA** | **DEPART** | **SE** | **PLACEM** | **STUDE** | **DEPART** | **COMP** | **SAL** |
| **D** | **ME** | **MENT** | **M** | **ENTID** | **NTID** | **MENT** | **ANY** | **ARY** |

1. nirm

al

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IT | 3 | 1 | 104 | IT |
| CSE | 5 | 2 | 105 | CSE |
| IT | 3 | 4 | 102 | IT |

infosy s

250

00

1. eshw ar

Wipro 220

00

1. bala

infosy s

250

00

1. sheel

a

CSE 3 5 103 CSE infosy

250

00

## Query:

s

SELECT itstudent.studentID, itstudent.sname,placement.company, placement.salary FROM itstudent

INNER JOIN placement

ON itstudent.studentID=placement.studentID;

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STUDENTID** | **SNAME** | **COMPANY** | **SALARY** |
| 104 | nirmal | infosys | 25000 |
| 105 | eshwar | Wipro | 22000 |
| 102 | bala | infosys | 25000 |
| 103 | sheela | infosys | 25000 |

# LEFT OUTER JOIN

## Query:

SELECT \*

FROM itstudent

LEFT OUTER JOIN placement

ON itstudent.studentID=placement.studentID;

## Output:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STUDE** | **SNA** | **DEPART** | **SE** | **PLACEM** | **STUDE** | **DEPART** | **COMP** | **SAL** |
| **NTID** | **ME** | **MENT** | **M** | **ENTID** | **NTID** | **MENT** | **ANY** | **ARY** |

nir mal

104

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IT | 3 | 1 | 104 | IT |
| CSE | 5 | 2 | 105 | CSE |
| IT | 3 | 4 | 102 | IT |
| CSE | 3 | 5 | 103 | CSE |
| IT | 5 | - | - | - |

infosy s

2500

0

esh war

105

1. bala
2. shee

la

101 rupe sh

Wipro 2200

0

|  |  |
| --- | --- |
| infosy | 2500 |
| s | 0 |
| infosy | 2500 |
| s | 0 |
| - | - |

# RIGHT OUTER JOIN

## Query:

SELECT \*

FROM itstudent

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RIGHT OUTER JOIN placement  ON itstudent.studentID=placement.studentID;  **Output:** | | | | | | | | | |  |
|  | **STUDE NTID** | **SNA ME** | **DEPART MENT** | **SE M** | **PLACEM ENTID** | **STUDE NTID** | **DEPART MENT** | **COMP ANY** | **SAL ARY** |  |
| 102 | bala | IT | 3 | 4 | 102 | IT | infosy s | 2500  0 |  |
| 103 | shee la | CSE | 3 | 5 | 103 | CSE | infosy s | 2500  0 |  |
| 104 | nir mal | IT | 3 | 1 | 104 | IT | infosy s | 2500  0 |  |
| 105 | esh war | CSE | 5 | 2 | 105 | CSE | Wipro | 2200  0 |  |
| - | - | - | - | 3 | 204 | MECH | HYUN DAI | 3000  0 |  |
| **7. FULL OUTER JOIN**  **Query:**  SELECT \*  FROM itstudent  FULL OUTER JOIN placement  ON itstudent.studentID=placement.studentID;  **Output:** | | | | | | | | | |  |
|  | **STUDE NTID** | **SNA ME** | **DEPART MENT** | **SE M** | **PLACEM ENTID** | **STUDE NTID** | **DEPART MENT** | **COMP ANY** | **SAL ARY** |  |
| 104 | nir mal | IT | 3 | 1 | 104 | IT | infosy s | 2500  0 |  |
| 105 | esh war | CSE | 5 | 2 | 105 | CSE | Wipro | 2200  0 |  |
| - | - | - | - | 3 | 204 | MECH | HYUN DAI | 3000  0 |  |
| 102 | bala | IT | 3 | 4 | 102 | IT | infosy s | 2500  0 |  |
| 103 | shee la | CSE | 3 | 5 | 103 | CSE | infosy s | 2500  0 |  |
| 101 | rupe sh | IT | 5 | - | - | - | - | - |  |
| **8. EQUI JOIN**  **Query:**  SELECT \* FROM itstudent, placement WHERE | | | | | | | | | |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| itstudent.studentID=placement.studentID;  **Output:** | | | | | | | | | |
|  | **STUD** |  |  |  |  |  |  |  |  |
| **ENTI** | **SNA** | **DEPART** | **SE** | **PLACEM** | **STUDE** | **DEPART** | **COMP** | **SAL** |
| **D** | **ME** | **MENT** | **M** | **ENTID** | **NTID** | **MENT** | **ANY** | **ARY** |
| 104 | nir | IT | 3 | 1 | 104 | IT | infosy | 2500 |
| mal | s | 0 |
| 105 | esh | CSE | 5 | 2 | 105 | CSE | Wipro | 2200 |
| war | 0 |
| 102 | bala | IT | 3 | 4 | 102 | IT | infosy | 2500 |
| s | 0 |
| 103 | shee | CSE | 3 | 5 | 103 | CSE | infosy | 2500 |
| la | s | 0 |
| **9. SELF JOIN**  Returns rows by comparing the values of the same table.  **Query:**  CREATE TABLE employee (  empid NUMBER, empname VARCHAR2(25), reportingid NUMBER  );  **Output:**  *Table Created*  **Query:**  INSERT INTO employee VALUES(1, 'Principal', null);  **Output:**  *1 row(s) inserted.*  **Query:**  INSERT INTO employee VALUES(2, 'HOD', 1);  **Output:**  *1 row(s) inserted.*  **Query:**  INSERT INTO employee VALUES(3, 'PO', 1);  **Output:**  *1 row(s) inserted.*  **Query:** | | | | | | | | | |

INSERT INTO employee VALUES(4, 'Staff', 2);

## Output: Query: Output: Query: Output:

|  |  |  |
| --- | --- | --- |
| **EMPID** | **EMPNAME** | **REPORTINGID** |
| 1 | Principal | - |
| 2 | HOD | 1 |
| 3 | PO | 1 |
| 4 | Staff | 2 |
| 5 | Non Teaching Staff | 2 |

*1 row(s) inserted.*

INSERT INTO employee VALUES(5, 'Non Teaching Staff', 2);

*1 row(s) inserted.*

SELECT \* FROM employee;

## Query:

SELECT e1.empid, e1.empname, e2.empname AS HeadName FROM employee e1, employee e2

WHERE e1.reportingid=e2.empid;

**Output:**

|  |  |  |
| --- | --- | --- |
| **EMPID** | **EMPNAME** | **HEADNAME** |
| 3 | PO | Principal |
| 2 | HOD | Principal |
| 5 | Non Teaching Staff | HOD |
| 4 | Staff | HOD |

# RESULT:

Thus the SQL queries using SQL joins were successfully executed and verified.