**EPX no - 6**

**Aim: To study various types of JOINS.**

**INNER JOIN:**

An 'inner join' is the most common join operation used in applications and can be regarded as the default join-type. Inner join creates a new result table by combining column values of two tables (A and B) based upon the join-predicate. The query compares each row of A with each row of B to find all pairs of rows which satisfy the join-predicate. When the join-predicate is satisfied, column values for each matched pair of rows of A and B are combined into a result row. The result of the join can be defined as the outcome of first taking the Cartesian product (or Cross join) of all records in the tables (combining every record in table A with every record in table B) and then returning all records which satisfy the join predicate.

**SELECT \*FROM employee INNER JOIN department**

**ON employee.DepartmentID = department.DepartmentID;**

**EQUI JOIN:**

An equi-join is a specific type of comparator-based join, that uses only equality comparisons in the join-predicate. Using other comparison operators (such as <) disqualifies a join as an equi-join. The query shown above has already provided an example of an equi-join:

**SELECT \***

**FROM employee JOIN department**

**ON employee.DepartmentID = department.DepartmentID;**

**OUTER JOIN:**

An outer join does not require each record in the two joined tables to have a matching record. The joined table retains each record—even if no other matching record exists. Outer joins subdivide further into left outer joins, right outer joins, and full outer joins, depending on which table's rows are retained (left, right, or both).

**LEFT OUTER JOIN:**

The result of a left outer join (or simply left join) for tables A and B always contains all records of the "left" table (A), even if the join-condition does not find any matching record in the "right" table (B). This means that if the ON clause matches 0 (zero) records in B (for a given record in A), the join will still return a row in the result (for that record)—but with NULL in each column from B. A left outer join returns all the values from an inner join plus all values in the left table that do not match to the right table.

Example of a left outer join (the OUTER keyword is optional), with the additional result row (compared with the inner join) italicized:

**SELECT \***

**FROM employee LEFT OUTER JOIN department**

**ON employee.DepartmentID = department.DepartmentID;**

**RIGHT OUTER JOIN:**

A right outer join (or right join) closely resembles a left outer join, except with the treatment of the tables reversed. Every row from the "right" table (B) will

appear in the joined table at least once. If no matching row from the "left" table (A) exists, NULL will appear in columns from A for those records that have no

match in B.

A right outer join returns all the values from the right table and matched values from the left table (NULL in the case of no matching join predicate).

An example of a right outer join (the OUTER keyword is optional), with the additional result row italicized:

**SELECT \***

**FROM employee RIGHT OUTER JOIN department**

**ON employee.DepartmentID = department.DepartmentID;**

**FULL OUTER JOIN:**

Conceptually, a full outer join combines the effect of applying both left and right outer joins. Where records in the FULL OUTER JOINed tables do not match,

the result set will have NULL values for every column of the table that lacks a matching row. For those records that do match, a single row will be produced in

the result set (containing fields populated from both tables).

For example, this allows us to see each employee who is in a department and each department that has an employee, but also see each employee who is not part

of a department and each department which doesn't have an employee.

Example of a full outer join (the OUTER keyword is optional):

**SELECT \***

**FROM employee FULL OUTER JOIN department**

**ON employee.DepartmentID = department.DepartmentID;**

**CONCLUSION:**

A SQL join clause combines records from two or more tables in a database. It creates a set that can besaved as a table or used as it is. A JOIN is a means for combining fields from two tables by using values common to each. ANSI standard SQL specifies four types of JOIN: INNER, OUTER, LEFT, and RIGHT. As a special case, a table (base table, view, or joined table) can JOIN to itself in a self-join.Aprogrammerwrites a JOIN statement to identify the records for joining. If the evaluated predicate is true,the combined record is then produced in the expected format, a record set or a temporary table.

**PROGRAM:**

SQL\*Plus: Release 11.1.0.6.0 - Production on Sat Aug 17 00:00:31 2013

Copyright (c) 1982, 2007, Oracle. All rights reserved.

Enter user-name: system

Enter password:

Connected to:

Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production

With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> create table customerKhushi (CustIdvarchar(6)primary key, Custnamevarchar

(15), CCityvarchar(10), CCountryvarchar(10));

Table created.

SQL>desccustomerKhushi

Name Null? Type

----------------------------------------- -------- ----------------------------

CUSTID NOT NULL VARCHAR2(6)

CUSTNAME VARCHAR2(15)

CCITY VARCHAR2(10)

CCOUNTRY VARCHAR2(10)

SQL> insert into customerKhushivalues(123456,'Khushboo','Mumbai','India');

1 row created.

SQL> insert into customerKhushivalues(21256,'Mansi','Delhi','India');

1 row created.

SQL> insert into customerKhushivalues(245456,'Hetal','Hong kong','China');

1 row created.

SQL> select \* from customerKhushi;

CUSTID CUSTNAME CCITY CCOUNTRY

------ --------------- ---------- ----------

123456 Khushboo Mumbai India

21256 Mansi Delhi India

245456 Hetal Hong kong China

SQL> insert into customerKhushivalues(223256,'Mahalaxmi','Sindhu','russia');

1 row created.

SQL> insert into customerKhushivalues(22216,'Rexy','Hiroshima','Japan');

1 row created.

SQL> select \* from customerKhushi;

CUSTID CUSTNAME CCITY CCOUNTRY

------ --------------- ---------- ----------

123456 Khushboo Mumbai India

21256 Mansi Delhi India

245456 Hetal Hong kong China

223256 MahalaxmiSindhurussia

22216 Rexy Hiroshima Japan

SQL> create table orderKhushi (OderIdvarchar(10), CustIdvarchar(6),foreign key

(CustId) references customerKhushi(custId), OrderDate date);

Table created.

SQL>descorderKhushi;

Name Null? Type

----------------------------------------- -------- ----------------------------

ODERID VARCHAR2(10)

CUSTID VARCHAR2(6)

ORDERDATE DATE

SQL> insert into orderKhushi values (12345,123456,'22-May -2012');

1 row created.

SQL> insert into orderKhushivalues(25515,21256,'12-Sep-1995');

1 row created.

SQL> insert into orderKhushivalues(21025,22216,'04-Aug-2010');

1 row created.

SQL> insert into orderKhushivalues(12005,223256,'10-March-2013');

1 row created.

SQL> insert into orderKhushivalues(47825,245456,'10-Jan-2012');

1 row created.

SQL> select\* from orderKhushi;

ODERID CUSTID ORDERDATE

---------- ------ ---------

12345 123456 22-MAY-12

25515 21256 12-SEP-95

21025 22216 04-AUG-10

12005 223256 10-MAR-13

47825 245456 10-JAN-12

SQL> select orderKhushi.oderId,customerKhushi.custName,orderKhushi.orderDate fro

morderKhushi INNER JOIN customerKhushi on orderKhushi.custId=customerKhushi.cus

tId;

ODERID CUSTNAME ORDERDATE

---------- --------------- ---------

12345 Khushboo 22-MAY-12

25515 Mansi 12-SEP-95

47825 Hetal 10-JAN-12

12005 Mahalaxmi 10-MAR-13

21025 Rexy 04-AUG-10

SQL> select customerKhushi.custName,orderKhushi.oderId from customerKhushi LEFT

JOIN orderKhushi on orderKhushi.custId=customerKhushi.custId;

CUSTNAME ODERID

--------------- ----------

Khushboo 12345

Mansi 25515

Rexy 21025

Mahalaxmi 12005

Hetal 47825

SQL> select customerKhushi.custName,orderKhushi.oderId from customerKhushi RIGHT

JOIN orderKhushi on orderKhushi.custId=customerKhushi.custId;

CUSTNAME ODERID

--------------- ----------

Khushboo 12345

Mansi 25515

Hetal 47825

Mahalaxmi 12005

Rexy 21025

SQL> select customerKhushi.custName,orderKhushi.oderId from customerKhushi FULL

OUTER JOIN orderKhushi on orderKhushi.custId=customerKhushi.custId;

CUSTNAME ODERID

--------------- ----------

Khushboo 12345

Mansi 25515

Rexy 21025

Mahalaxmi 12005

Hetal 47825