NUGGET 4 : COMBINING DATA

Persistent University

PERSISTENT

Partners in Innovation

KEY LEARNING POINTS

- Self Join
- Outer Join (Oracle Proprietary Syntax)
- Outer Join (SQL:1999 complaint Syntax)

SAMPLE DATA

PERSISTENT

Table Name : Employee

EmployeeId	FirseName	LastName	Email	PhoneNumber	HireDate	JobId	Salary	CommitionPct	ManagerId	DepartmentId
1	John	Demn	JohnD@yahoo.com	9898780979	1/10/2001	IT_PROF	70000	0.5	NULL	10
2	Ken	Dale	kendaleD@gmail.com	7877787655	4/1/2001	SALES_HEAD	50000	NULL	NULL	10
3	James	Walton	JW@yahoo.com	5787887888	1/1/2001	IT_REP	30000	0.2	1	20
4	robin	sngal	robin@gmail.com	4990988839	5/1/2001	SALES_REP	40000	0.3	2	20
5	ajay	ghosala	ghosala@hotmail.com	9809888898	6/10/2002	SALES_REP	30000	0.4	2	20
6	John	Reddies	John@gmail.com	6878900989	6/10/2003	M_per	50000	NULL	NULL	NULL

Table Name : Department

DEPARTMENT ID	DEPARTMENTNAME	MANAGERID	LOCATIONID
10	Sales	1	1
20	IT	2	2
30	Marketing	(null)	1

Table Name : Location

LOCATIONID	CITY
1	Pune
2	Mumbai

Table Name : Job_History

EMPLOYEEID	START_DATE	END_DATE	JOBID	DEPARTMENTID
1	10-Jan-01	31-Dec-01	IT_PROF	10
1	1-Jan-02	31-Aug-05	IT_REP	10
2	1-Apr-01	31-Jan-02	SALES_REP	20
2	1-Feb-02	31-Jan-05	SALES_PERS	10

JOINING TABLES

PERSISTENT

- **Definition:**

  When data from more than one table in database is required, a JOIN condition is used. Rows in one table can be joined to rows in another table according to common values existing in corresponding column, usually, primary and foreign key columns.

- **Syntax:**

```
SELECT table1.column, table2.column
From table1, table2
Where table1.column1=table2.column2;
    -- WHERE clause contains the condition that joins
        the tables together.
```

JOINING TABLES

PERSISTENT

Guidelines:

- When writing the SELECT statement that joins the tables, precede the column name with table name for clarity and enhance database access.
- If same column name appears in more than one table, column name must be prefixed with table name.
- To join n tables, you need minimum of n-1 join conditions. This rule may not apply if your table has composite primary key.

There are various types of JOINS available

> Inner Join/ Equi Join

> Cartesian Product/Cross Join

> Natural Join

> Self Join

> Outer Join

Note : Inner Join, Natural Join and Cartesian product details are already covered in Basic SQL Queries course.

TABLE ALIASES

- Qualifying column names with table names can be very time consuming; particularly if table names are lengthy. User can use table aliases instead of table names.
- As column alias gives a column another name, table alias gives table another name.
- Table aliases helps to keep SQL code smaller and thereby saving the memory.
- Table aliases are specified as : From Employee Emp, department Dept
- Guidelines:
  - Can be up to 30 characters in length; but shorter is better.
  - If the table alias is used for a particular table name in FROM clause then that table alias must be substituted for the table name throughout the SELECT statement.
  - Table aliases should be meaningful.
  - Table alias is valid only for the current SELECT statement.

SELF JOIN

PERSISTENT

- Joining a **Table to itself**:

  Sometimes User needs to join table to itself.

  For example: To find employee's manager name, user need to join Employee table to itself. It is called as Self Join.

- To Find Robin's manager:
  - Find details of Robin in Employee table
  - Find the manager number of Robin in employee table. Robin's manager id is 2.
  - Find the name of the employee with employee id 2. 2 is the employee id of Ken. So Ken is Robin's manager.

  Thus, in this process, we need to look into same table twice.

- Example:

```
Select e.employeeid, e.firstname||' '||e.lastname employeename,
m.employeeid managerId, m.firstname ||' '|| m.lastname manager_name
from employee e, employee m
where e.managerid= m.employeeid;
```

EMPLOYEEID	EMPLOYEE NAME	MANAGERID	MANAGER_NAME
3	James Walton	1	John Demn
5	ajay ghosala	2	Ken Dale
4	robin sngal	2	Ken Dale

OUTER JOIN: ORACLE PROPRIETARY SYNTAX

- If a particular row does not satisfy the Join condition, it will not appear in the query result. Thus, to include such rows in the result, user can use outer join condition.
- Outer Join operator is (+) : plus sign enclosed in parenthesis.
- This (+) operator has the effect of creating one or more null rows, to which one or more rows from non deficient table can be joined.
- **Syntax**:

  SELECT table1.column, table2.column

  From table1, table2

  Where table1.column1=table2.column2(+);

- (+) Operator can be placed on any side of Join condition; but not on the both sides.
- A condition involving an Outer join can not use the IN operator or be linked with another condition using OR operator.

OUTER JOIN: ORACLE PROPRIETARY SYNTAX: EXAMPLE

- Example:

```
Select E.employeeid, E.lastname, D.departmentid, D.departmentname
From employee E, department D
Where E.departmentid(+) =D.departmentid;
```

EMPLOYEEID	LASTNAME	DEPARTMENTID	DEPARTMENTNAME
1	Demn	10	Sales
2	Dale	20	IT
3	Walton	10	Sales
4	sngal	20	IT
5	ghosala	20	IT
(null)	(null)	30	Marketing

-- Output contains all the Departments whether or not it contains any Employee.

OUTER JOIN: ANSI SQL : 1999 SYNTAX

- Oracle 9i and further versions also supports SQL:1999 JOIN syntax.
- **Syntax**:

  Select table1.column, table2.column

  From table1 LEFT/FULL/RIGHT OUTER JOIN table2

  ON (table1.column=table2.column);

- LEFT Outer Join retrieves all the rows from the table which is at the Left Hand Side(LHS) , even if there is no match in right table. It is same as

  ..From table1, table2 Where table1.column=table2.column(+)

- RIGHT Outer Join retrieves all the rows from the table which is at the Right Hand Side (RHS), even if there is no match in left table. It is same as

  ..From table1, table2 Where table1.column(+)=table2.column

OUTER JOIN: ANSI SQL : 1999 SYNTAX

- FULL Outer Join retrieves all the rows; matched as well as unmatched from both the tables. It is same as

  select table1.column, table2.column

  From table1, table2 Where table1.column(+)=table2.column

  UNION

  select table1.column, table2.column

  From table1, table2 Where table1.column=table2.column(+)

- Left Outer Join Example:

  Select E.employeeid, E.lastname, D.departmentid, D.departmentname

  from employee E Left outer join department D

  on E.departmentid =D.departmentid;

EMPLOYEEID	LASTNAME	DEPARTMENTID	DEPARTMENTNAME
3	Walton	10	Sales
1	Demn	10	Sales
5	ghosala	20	IT
4	sngal	20	IT
2	Dale	20	IT
6	Reddies	null	null

OUTER JOIN: ANSI SQL : 1999 SYNTAX

- Right Outer Join Example:

```
Select E.employeeid, E.lastname, D.departmentid, D.departmentname
From employee E  Right outer join department D
On E.departmentid =D.departmentid;
```

EMPLOYEEID	LASTNAME	DEPARTMENTID	DEPARTMENTNAME
1	Demn	10	Sales
2	Dale	20	IT
3	Walton	10	Sales
4	sngal	20	IT
5	ghosala	20	IT
(null)	(null)	30	Marketing

- Full Outer Join Example:

```
Select E.employeeid, E.lastname, D.departmentid, D.departmentname
From employee E  FULL outer join
        department D
On E.departmentid = D.departmentid;
```

EMPLOYEEID	LASTNAME	DEPARTMENTID	DEPARTMENTNAME
1	Demn	10	Sales
2	Dale	20	IT
6	Reddies	null	null
3	Walton	10	Sales
4	sngal	20	IT
5	ghosala	20	IT
null	null	30	Marketing

REFERENCE MATERIAL

- Sites:

http://docs.oracle.com/cd/B19306_01/server.102/b14200/queries006.htm

http://www.techonthenet.com/sql/joins.php

http://psoug.org/definition/JOIN.htm

http://beginner-sql-tutorial.com/sql-joins.htm

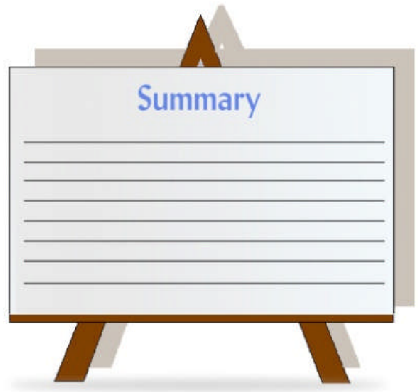
SESSION 4 SUMMARY

PERSISTENT

With this we have come to an end of our fourth session where we discussed concept of Self join and Outer join with examples.

At the end of Nugget 4, we see that you are now able to answer following questions:

- Explain “Types of Joins”
- Explain “Concept of Self Join with example”
- Explain “OUTER Join syntax with examples”



THANK YOU!!!!!!!!!!!!!!! 

PERSISTENT

Partners in Innovation

Persistent University