Oracle

Table of Contents

[1. Oracle Join 3](#_Toc510095203)

[1.1 Physical Join 3](#_Toc510095204)

[1.2 Combined Data 3](#_Toc510095205)

[1.3 Cross join 3](#_Toc510095206)

[1.4 Equi join 4](#_Toc510095207)

[1.5 Natural Join 4](#_Toc510095208)

[1.6 Inner Join 5](#_Toc510095209)

[1.7 Outer Join 5](#_Toc510095210)

[1.8 Left Outer Join 5](#_Toc510095211)

[1.9 Left Outer Join 6](#_Toc510095212)

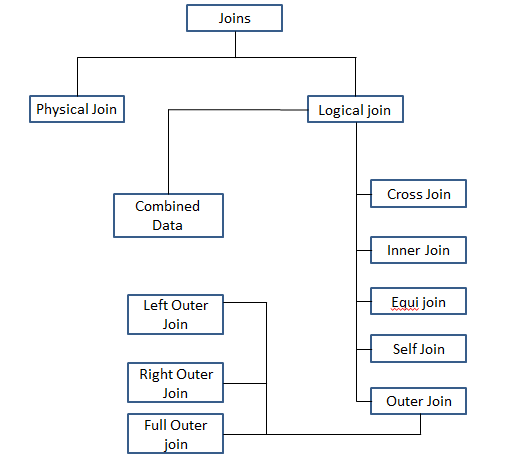
[1.10 Full Outer Join 6](#_Toc510095213)

[1.11 Self-Join 6](#_Toc510095214)

[2. Triggers 6](#_Toc510095215)

# Oracle Join

Join operations are used to retrieve data from multiple tables.



## Physical Join

This join can be accomplished by establishing relation between the required tables through primary and foreign key relationship.

## Combined Data

Combined data can be retrieved from multiple tables through set operations.

## Cross join

The outcome of the cross join will be same as the Cartesian product. The ideal scenario to use cross join is when we have the relation between the tables is many to one. For example if we are to map specific promotion with a given set of service in such scenario cross join will be provide the required result.

Example if we have two sets A1 = {a, b} and B1 = {10, 20} the Cartesian product of A1 \* B1 will be A1 \* B1 = {(a, 10), (a, 20), (b, 10), (b, 20)}

**Query Syntax 1:** *select \* from table\_1 cross join table\_2 where Criteria;*

**Query Syntax 2:** *select \* from table\_1 , table\_2 where Criteria;*

**Example:** Select \* from Customer\_Equipment\_Tracking cet cross join Tap\_Infce.Tracked\_Equipment\_Reference ter where b1\_number = 'b1aada87';

## Equi join

It performs a join based on the equality operator or matching columns between the associated tables. Only an equal operator can be used for Equi Join operation. Which differentiates it from the rest of the join types.

**Note:** Always prefer writing join conditions proceeding with filter conditions reason being, Say we have 100 records. First the filter will be applied and on the remaining records joining will be performed.

Example : Select \* from customer\_equipment\_Tracking cet , Tap\_Infce.Tacked\_Equipment\_Reference ter where b1\_number = 'b1aada87' and cet.oui = ter.oui;

## Natural Join

It is a type of equi join which implicitly performs joins based on the columns that are available in both the associated tables with the same name. As we don’t have control over the columns being compared at times the desired result may not be available using natural join.

Example: Consider the below mentioned table descriptions.

Table 1: Product\_subscription

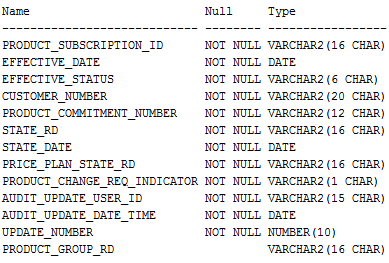
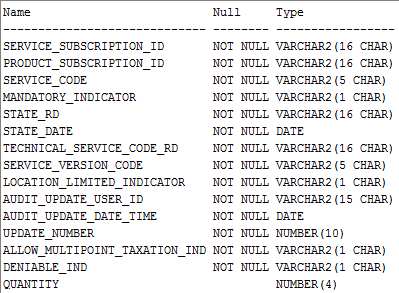


Table 2: Service\_subscription



The column that is common among both the tables is the Product\_subscription\_id and State\_Rd. Upon using natural join the outcome of the query would match both the tables based on the common columns and the provide the results where the value is some in the associated tables.

**Example:** select Product\_Subscription\_id from Product\_Subscription ps natural join Service\_Subscription ss where ps.Product\_Subscription\_Id = '0000000030677565';

## Inner Join

This join selects rows from both the tables as long as there is a match between the columns. It is almost same as Equi join except the fact that the join condition may contain operators like >, <.

## Outer Join

The join retrieves the rows from both participating table which satisfies the join condition and also the one’s that does not satisfy the condition. The outer join is always used on one side of the join condition.

## Left Outer Join

The operation retrieves all the rows from the participating tables matching the join condition and also the ones that are not satisfying the condition from the first table.

## Right Outer Join

The Join fetches all the rows from the corresponding tables satisfying the join condition and also the ones that are not satisfying the condition from the second table.

## Full Outer Join

This join operation returns all the rows from both the participating table in the event if there no matching record on the counter part of the table the record will be initialized to null.

## Self-Join

This operation can be viewed as a join operation on the table itself. For example if we are to retrieve all the records from business activity table using the root activity id.

# Triggers

The following are the types of triggers supported by oracle SQL.

1. Before Insert.
2. After Insert.
3. Before update.
4. After Update.
5. Before Delete.
6. After Delete.

# Procedure creation in Oracle

Procedures in Oracle are very much similar to function with few of its own additional features. Here is the basic syntax to create a procedure.

CREATE [OR REPLACE] Procedure *Procedure\_Name* (Paramesters1…)

AS

Declaration Section

Begin

Executable Section

Exception Handling

End *Procesure\_name*

During the creation of procedure of function the parameters can be declared in 3 ways.

1. *IN -* These parameters can be referenced by a procedure or function but the value of these parameters cannot be overridden by procedure or function.
2. *OUT –* These parameters cannot be referenced but the contained values can be overridden by a method or function.
3. *IN OUT –* These parameters can be referenced as well as it can be overridden by the procedure or function.

*Example:*

*CREATE OR REPLACE PROCEDURE UpdateServiceName*

*(Customer\_Number* ***IN*** *varchar2)*

***AS***

*B1\_Number number;*

*Cursor c1 is*

*Select Customer\_Internal\_Ref\_Number from customer;*

***BEGIN***

*Open c1;*

*Fetch c1 into B1\_number;*

*If 1%notfound then*

*B1\_Number := 0000;*

*End if;*

*Close c1;*

*Exception When others then*

*Raise\_applicaton\_error (-2001, ‘An Error occurred’||SQLCODE|| ‘Error’ || SQLERRNUM);*

***End*** *UpdateServiceName;*

Syntax for calling a function from SQL query:

*select TRACKING\_EQUIPMENT\_PKG.getCurrentEquipment('','','b1vvnf37') from dual;*

Syntax for calling procedure:

*declare out varchar2(32767);*

*begin*

*TAP\_INFCE.peg\_MDM\_EQP\_REQUEST\_BUNDLER.ProcessBundling(job,out,'MDM','BUNDLER','N','N', 'N');*

*end;*

# Procedure VS Function

1. Functions must return a value but procedures may or may not return any values.
2. Procedures can have IN and IN OUT parameters but functions can have only IN parameters. Oracle compiler does not throw any error message if a parameters is declared as IN / IN OUT but the function can no longer be called through a plain SQL query if attempted the calling would result in an error.
3. Functions can be called from a procedure but procedures cannot be called from functions.

Procedures allows Select as well as DML statements(Insert / Delete /update).Whereas functions does not allow meaning if there is a DML statement within a function the same can no longer be called from a plain SQL query. **Error Message:**

*ORA-14551: cannot perform a DML operation inside a query*

*ORA-06512: at "TAP\_INFCE.TRACKING\_EQUIPMENT\_PKG", line 37*

*14551. 00000 - "cannot perform a DML operation inside a query "*

*\*Cause: DML operation like insert, update, delete or select-for-update*

*cannot be performed inside a query or under a PDML slave.*

*\*Action: Ensure that the offending DML operation is not performed or*

*use an autonomous transaction to perform the DML operation within*

*the query or PDML slave.*

1. Functions can be called from a plain SQL statement or in *where* clause or in *having* but procedures cannot be called from plain SQL statements.

# Cursor

# Bulk

# Query to find second largest salary

Select \* from (select e.\*, rank () over (order by salary desc) as rank from employee e) where rank = 2;

# Union VS Union all

Union operation displays the common records from the two tables by removing duplicates. Whereas union all does the same job without removing the duplicates that is all the redundant rows will be displayed as the result.

# Primary Key vs. Unique Key

|  |  |
| --- | --- |
| ***Primary Key*** | ***Unique Key*** |
| There can be only one primary key in a table. | There can be more than one unique key in a table. |
| It is a unique identifier for the record | Unique key can have null values |
| In Oracle Primary key cannot be null |  |