JOINS

Joins are SQL operation which helps us in retrieving data from one or more tables that share common columns.

WHY joins.

Many a times we need to work using multiple tables at the same time assuming them to be single entity. In such cases a single SQL statement manipulates the data from the entire table.

Types of joins

JOIN

2 types

Physical join logical join

Relation between 2 tables combined data combination of data

(by using integrity constraint(pk))

Cross join

Equi join

Inner join

Outer join

Self join

Cross join

Cross join is also a cartesian join because it produces cartesian product.

Syntax

Select column\_names

From table1 cross join table1;

Or

Select column\_names

From table1, table2;

Total rows in result = Total rows in table1 \* Total rows in table2;

If table 1 has 5 records and table2 has 5 records then

5\*5 =25;

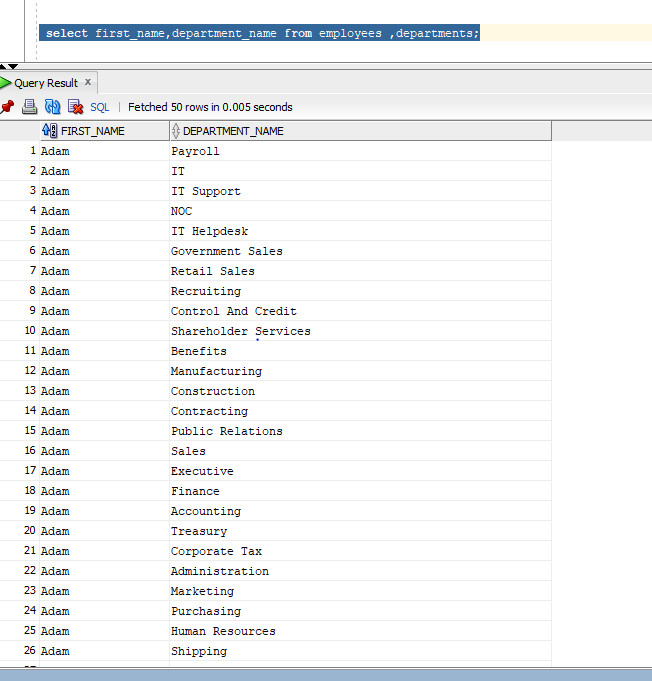
Each row in table1 is mapped with every row in second table2.

Displays all possible combinations.

Example

Select first\_name,department\_name

From employees,departments;



Equi join

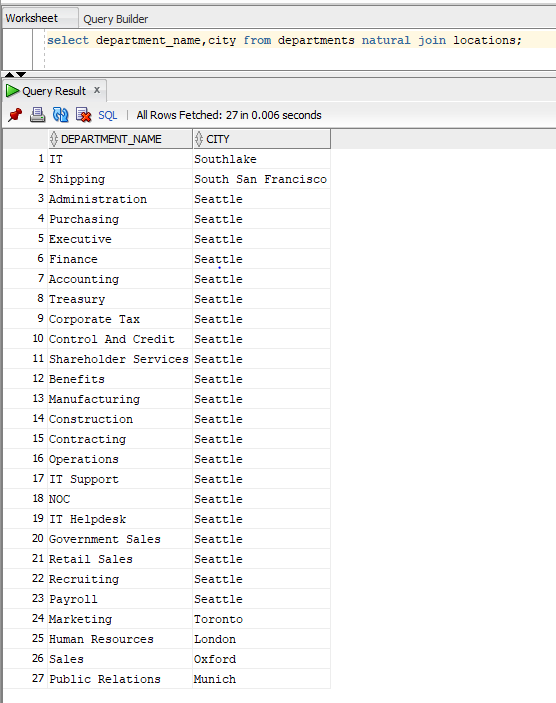
Cross join along with join condition .

Natural joins

Select department\_name,city

From departments natural join locations;

Best of natural join is that you no need to specify the join column bcoz column with same name in source and target are automatically associated with each other.

k

It returns all the rows from the department table matching with location\_id of departments table with location\_id of locations table.

Scenario 2: when there are two matching columns.

Select first\_name,department\_name

From employees natural join departments ;

Inner join

Return only those records from both the tables which satisfy the join condition.

Syntax:

Select column\_names From table1 inner join table2 On(expression) or using(column name)

Where(expression) Order by column names.

Select first\_name,department\_name From employees e join departments d

On(e.department\_id=d.department\_id);

Right outer join

Returns each and every record from the source table and returns only those values from the target table that fulfill the join condition.

Syntax

Select column\_names

From target\_Table right outer join source\_Table

On(Expression);

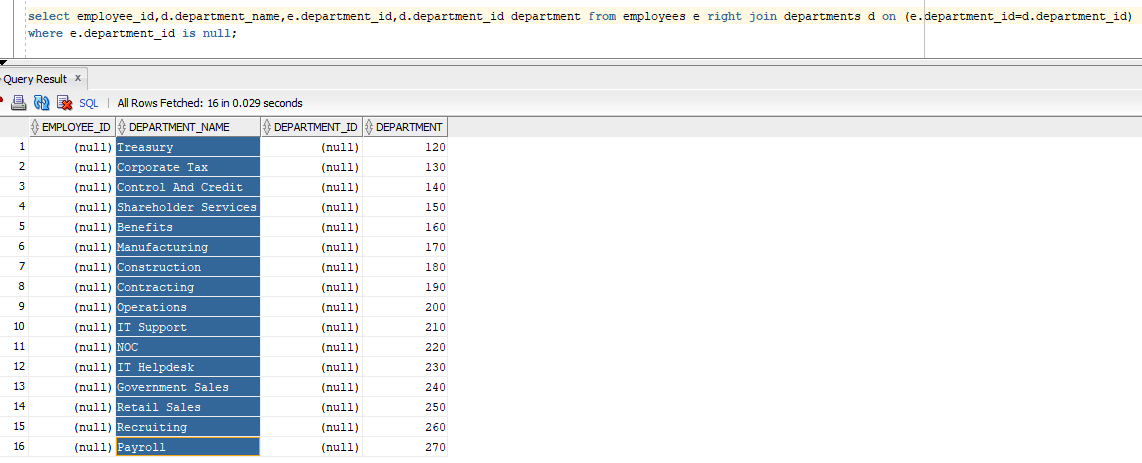
Example

To display deparment name where no employees are not working.

select first\_name,d.department\_name,e.department\_id,d.department\_id department from employees e right join departments d on (e.department\_id=d.department\_id)

where e.department\_id is null;

column participating in on clause must have same name or different name but with same datatype.



Left outer join

Returns each and every record from the source table and returns only those values from the target table that fulfill the join condition.

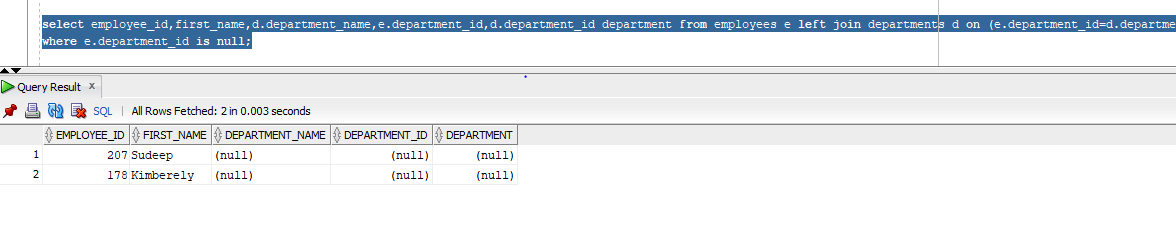
Select column\_names From source\_Table right outer join target\_Table On(Expression);

Example

Display employees names not assigned to any department.

select employee\_id,first\_name,d.department\_name,e.department\_id,d.department\_id department from employees e left join departments d on (e.department\_id=d.department\_id)

where e.department\_id is null;

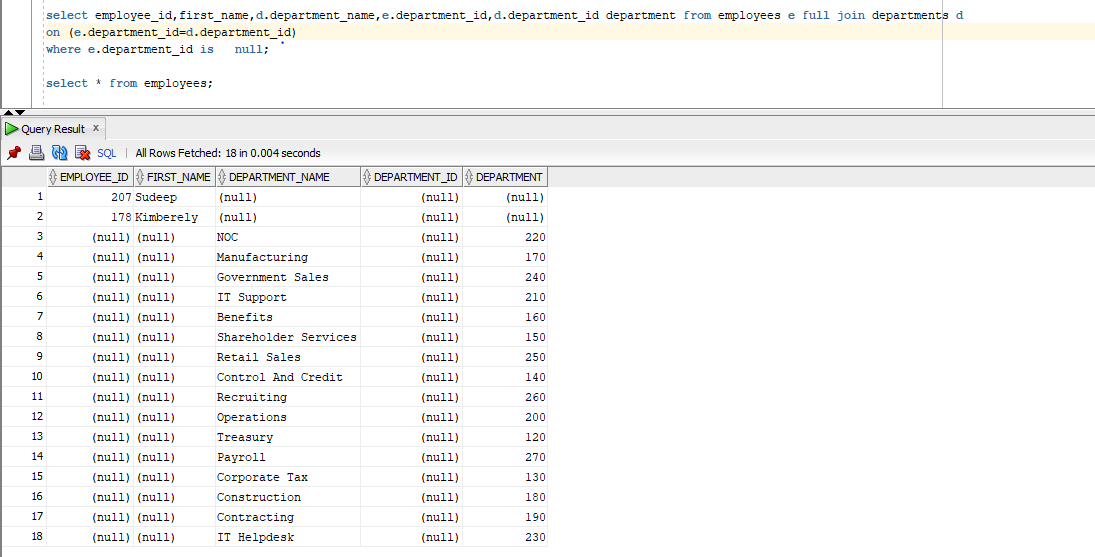


Full outer join

It will returns the both matching and non matching records from both the tables.

select employee\_id,first\_name,d.department\_name,e.department\_id,d.department\_id department from employees e full join departments d on (e.department\_id=d.department\_id)

where e.department\_id is null;



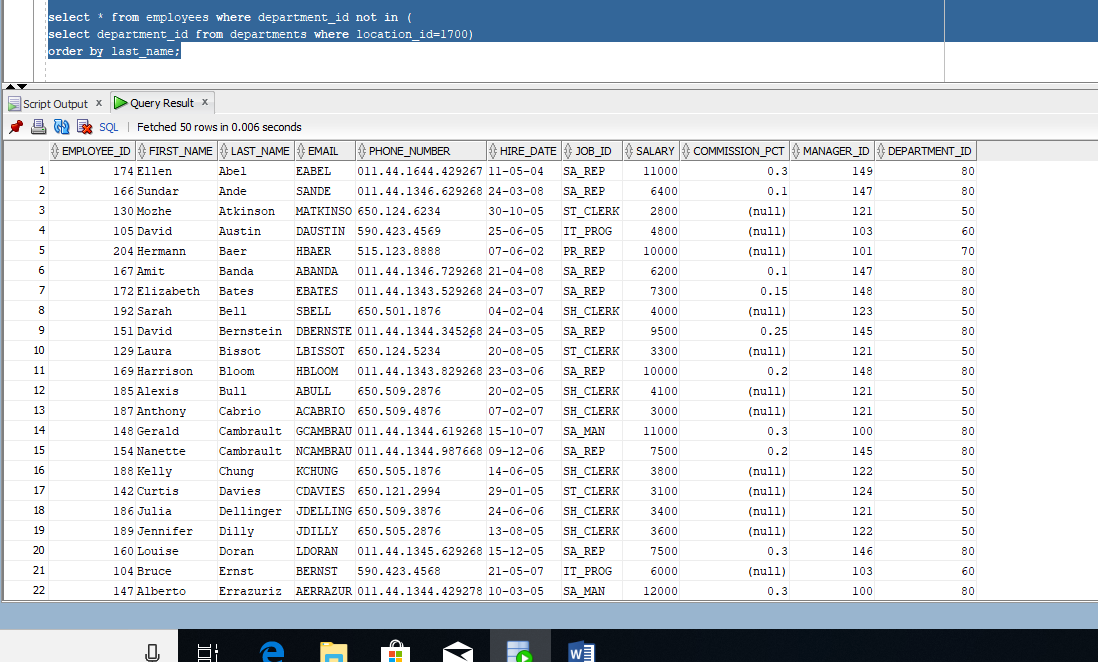
Antijoin

An antijoin returns rows from the left side of the predicate for which there are no corresponding rows on the right side of the predicate. It returns rows that fail to match(not in) the subquery on the right side.

select \* from employees where department\_id not in (

select department\_id from departments where location\_id=1700)

order by last\_name;



Semijoin

A semijoin can be used, for example to find all the department with atleast one employee whose salary s greater then 2500;

select \* from departments d

where exists

(select 1 from employees e where d.department\_id=e.department\_id and e.salary>2500);

