**Data Query Language or Data Retrieve Language**

* The SQL Select Query does not store any data itself. It simply displays the data that is stored in database tables.
* The SELECT statement’s main purpose is to retrieve the data from the database table and return it in a tabular structure.
* The SELECT clause is executed after the FROM clause and any optional WHERE, GROUP BY, and HAVING clauses if present.
* The SELECT is used to retrieve all rows from a table at a time or can be used to retrieve a specific row from a table by using the “WHERE” condition.

CREATE TABLE Employee (

Id INT PRIMARY KEY,

Name VARCHAR(15),

Department VARCHAR(10),

Salary NUMBER(8, 2),

Gender VARCHAR(10),

Age INT,

City VARCHAR(10)

);

INSERT INTO Employee (Id, Name, Department, Salary, Gender, Age, City) VALUES (1001, 'John', 'IT', 35000, 'Male', 25, 'London');

INSERT INTO Employee (Id, Name, Department, Salary, Gender, Age, City) VALUES (1002, 'Smith', 'HR', 45000, 'Female', 27, 'Mumbai');

INSERT INTO Employee (Id, Name, Department, Salary, Gender, Age, City) VALUES (1003, 'James', 'Finance', 50000, 'Male', 28, 'Delhi');

INSERT INTO Employee (Id, Name, Department, Salary, Gender, Age, City) VALUES (1004, 'Mike', 'Finance', 50000, 'Male', 28, 'London');

INSERT INTO Employee (Id, Name, Department, Salary, Gender, Age, City) VALUES (1005, 'Linda', 'HR', 75000, 'Female', 26, 'Mumbai');

INSERT INTO Employee (Id, Name, Department, Salary, Gender, Age, City) VALUES (1006, 'Anurag', 'IT', 35000, 'Male', 25, 'London');

INSERT INTO Employee (Id, Name, Department, Salary, Gender, Age, City) VALUES (1007, 'Priyanla', 'HR', 45000, 'Female', 27, 'Mumbai');

INSERT INTO Employee (Id, Name, Department, Salary, Gender, Age, City) VALUES (1008, 'Sambit', 'IT', 50000, 'Male', 28, 'London');

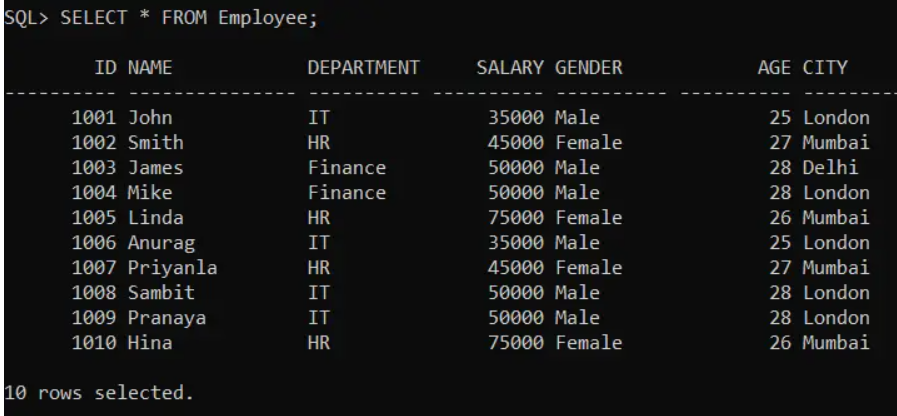
INSERT INTO Employee (Id, Name, Department, Salary, Gender, Age, City) VALUES (1009, 'Pranaya', 'IT', 50000, 'Male', 28, 'London');

INSERT INTO Employee (Id, Name, Department, Salary, Gender, Age, City) VALUES (1010, 'Hina', 'HR', 75000, 'Female', 26, 'Mumbai');

###### **Syntax1: SELECT All Columns**

If you want to select all the columns of a table or view then you can use “\*” as shown in the below syntax.  
**SELECT \* FROM Table\_Name;**

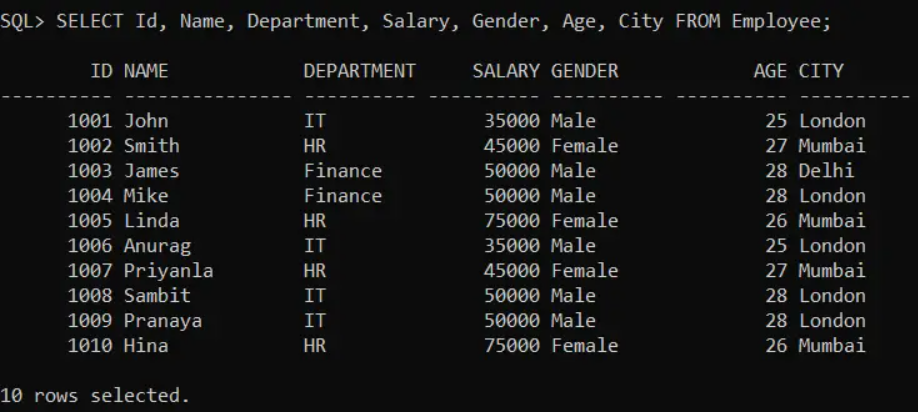
**Use \* as follows**  
**SELECT \* FROM Employee;**  
**Output:**

****

###### **Option2: Specify all the Columns in the SELECT Class as follows:**

**SELECT Id, Name, Department, Salary, Gender, Age, City FROM Employee;**

###### **Output:**

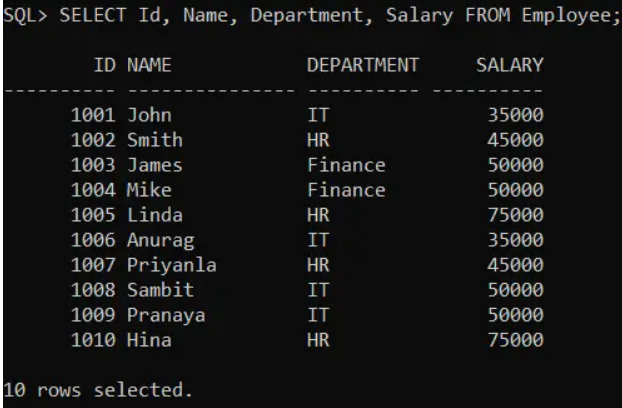


###### **Syntax2: SELECT Specific Columns**

If you want to select specific columns, then you need to specify such column names before the FROM Clause as shown in the below syntax.  
**SELECT Column\_List FROM Table\_Name;**

##### **Retrieve Specific Columns from Employee table**

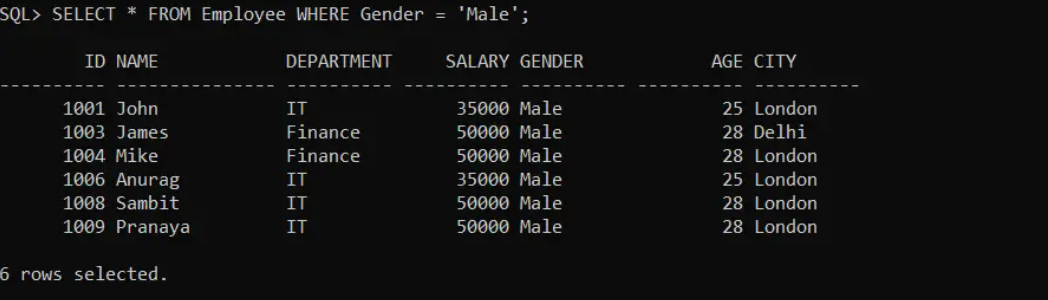
**SELECT Id, Name, Department, Salary FROM Employee;**



###### **Syntax3: SELECT All Columns with Conditions**

If you want to select All Columns with WHERE (or any other) conditions then you need to use the select statement as shown in the below syntax.  
**SELECT \* FROM tablename  [WHERE condition];**

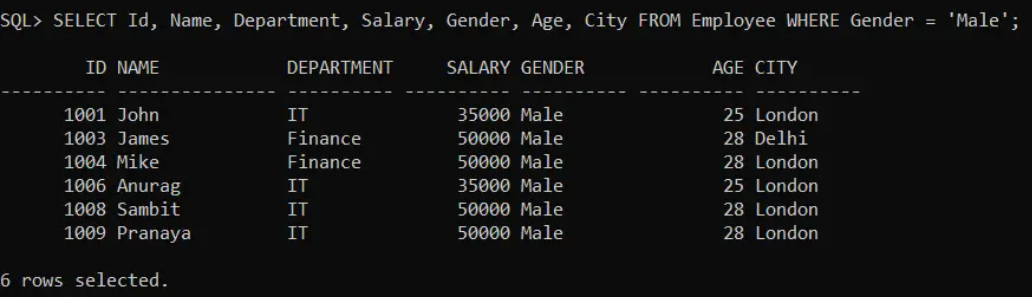
**SELECT \* FROM Employee WHERE Gender = ‘Male’;**

****

###### **Syntax4: SELECT Specified Columns with Conditions**

If you want to select specific Columns with WHERE (or any other) conditions then you need to use the select statement as shown in the below syntax. Here, you need to specify the column list that you want to retrieve.  
**SELECT Column\_List FROM tables [WHERE conditions];**

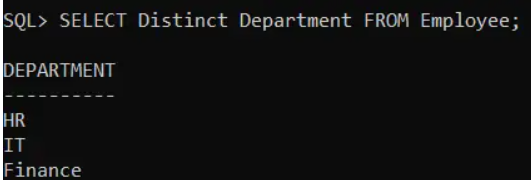
**SELECT Id, Name, Department, Salary, Gender, Age, City FROM Employee WHERE Gender = ‘Male’;**

****

##### **SELECT DISTINCT Statement in Oracle**

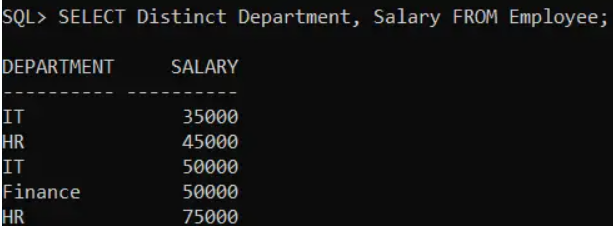
The SELECT DISTINCT SQL statement in Oracle is used to return only the distinct or different values from a table column. In a database table, a column may contain duplicate or similar values.

**SELECT Distinct Department FROM Employee;**



We can also combine 2 or more columns to get distinct values.

**SELECT Distinct Department, Salary FROM Employee;**



##### **Alias Names in Oracle:**

It is nothing but an alternate (or) temporary name. Users can create alias names on two levels in a database.

###### **I) Column Level:**

At this level, we are creating alias names on columns. The syntax is given below.  
<column name> <column alias name>  
**Example: Department Dept**

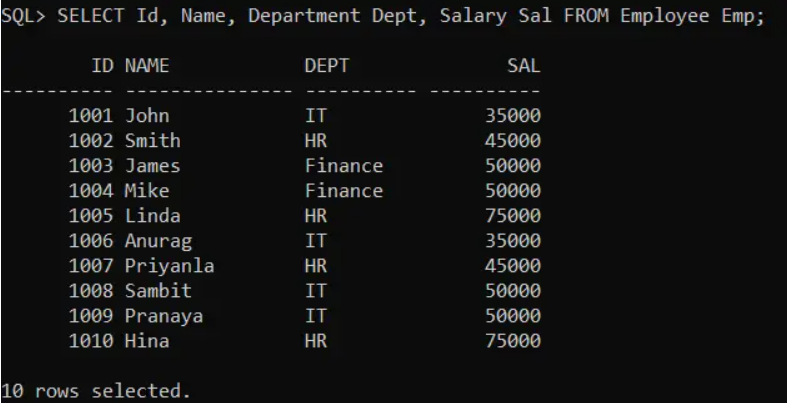
###### **II) Table Level:**

In this level, we create alias names on the table. The syntax is given below.  
<table name> <table alias name>  
**Example: Employee Emp**

###### **Syntax to combined column + table level alias names by using “select” query:**

**Select <column name1> <column name1 alias name>, <column name2> <column name2 alias name> From <Table Name> <Table Alias Name>;**

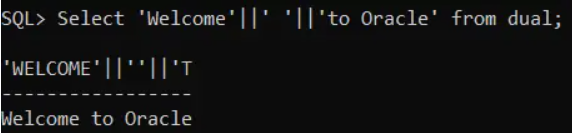
**SELECT Id, Name, Department Dept, Salary Sal FROM Employee Emp;**



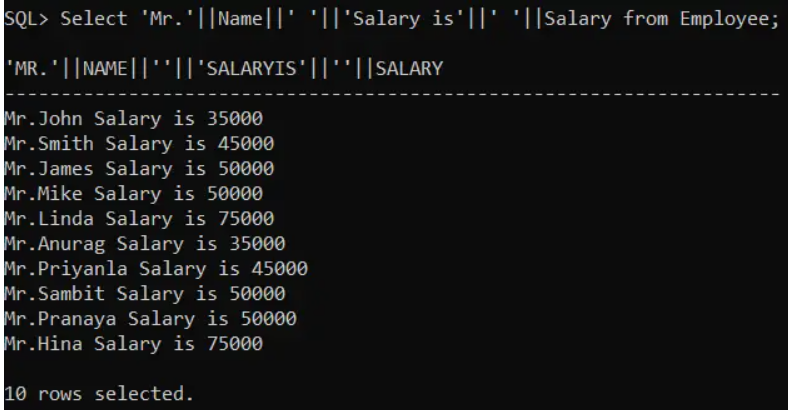
##### **Concatenation Operator (||) in Oracle:**

This operator is used to join two string values (or) two expressions in a select query.

**Example: Select ‘Welcome’||’ ‘||’to Oracle’ from dual;**



**Example: Select ‘Mr.’||Name||’ ‘||’Salary is’||’ ‘||Salary from Employee;**



##### **copy data from one table to another table in Oracle**

before that create the table structure

CREATE TABLE TempEmployee (

Id INT PRIMARY KEY,

Name VARCHAR(15),

Department VARCHAR(10),

Salary NUMBER(8, 2),

Gender VARCHAR(10),

Age INT,

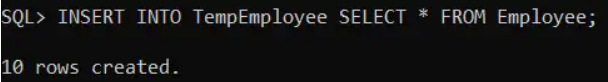
City VARCHAR(10)

);

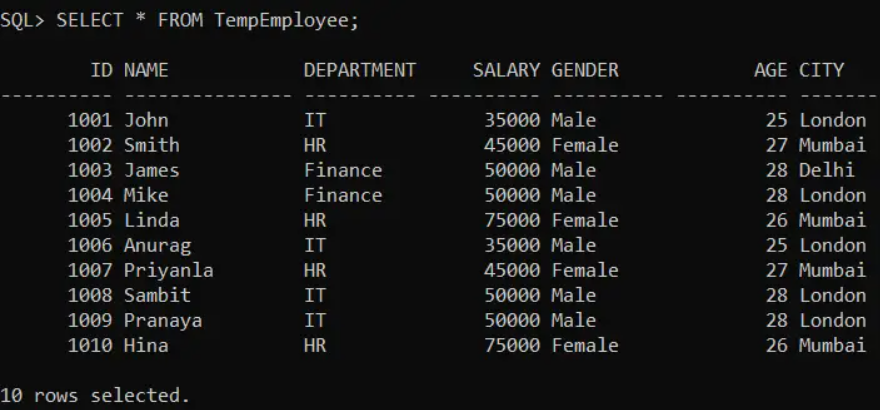
Rules for copying one table to another table :

1. No of columns in both tables must be same
2. Order of columns and data types of columns in both tables must match.

**INSERT INTO <DESTINATION TABLE NAME> SELECT \* FROM <SOURCE TABLE NAME>;**

**Eg: INSERT INTO TempEmployee SELECT \* FROM Employee;** ****

**SELECT \* FROM TempEmployee;**

****

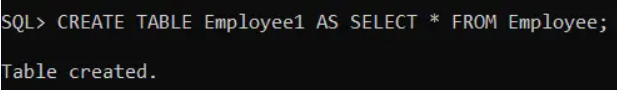
##### **Example2:**

Suppose you want to create a new table with the same structure and same data as the existing table.

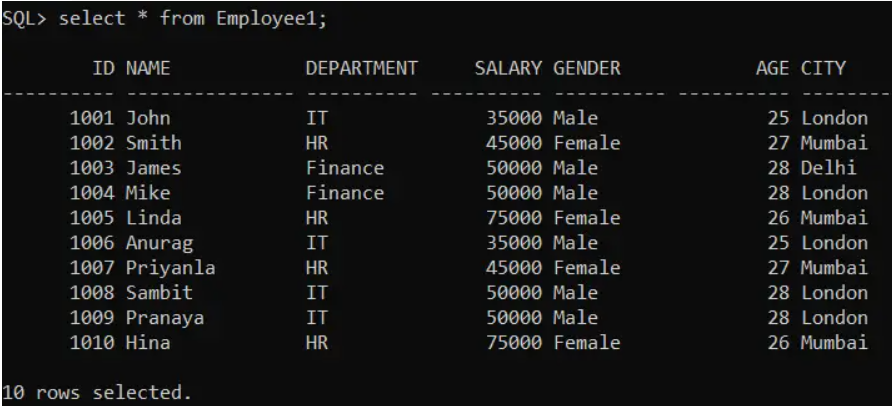
**CREATE TABLE Employee1 AS SELECT \* FROM Employee;**

**(or)**

**CREATE TABLE Employee1 AS SELECT \* FROM Employee WHERE 1 = 1;**

****

**SELECT \* FROM Employee1;**

****

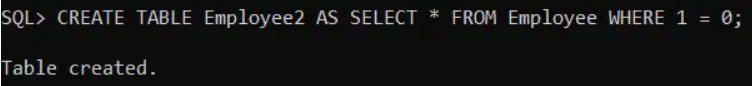
##### **Example3:**

If you want to create a new table with the same structure as an existing table but without data.

**CREATE TABLE Employee2 AS SELECT \* FROM Employee WHERE 1 = 0;**

**(OR)**

**CREATE TABLE Employee2 AS SELECT \* FROM Employee WHERE 1 = 2;**

****

Example 4 :

If you want to create a new table from old table only with specific columns

**CREATE TABLE old table AS SELECT COLUMN LIST FROM old table;**

**CREATE TABLE Employee3 AS SELECT COLUMN LIST FROM Employee;**

Example 5:

If you want to create a new table from old table only with specific rows

**CREATE TABLE Employee4 AS SELECT \* FROM Employee WHERE CONDITION;**