**All Table code :**

-- Create Employees table

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(255),

LastName VARCHAR(255),

Branch NCHAR(5),

Age NUMERIC(2, 0),

NIC NCHAR(15),

Scale NUMERIC(1, 0),

Salary INT,

JoiningDate DATE,

OvertimeHours FLOAT,

JobTitle VARCHAR(255),

DepartmentId INT

);

-- Insert data for the Employees table

INSERT INTO Employees (EmployeeID, FirstName, LastName, Branch, Age, NIC, Scale, Salary, JoiningDate, OvertimeHours, JobTitle, DepartmentId)

VALUES (1, 'Noman', 'Ijaz', 'NY', 30, '123456789012345', 1, 65000, TO\_DATE('2022-05-10', 'YYYY-MM-DD'), 5.5, 'Manager', 1);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Branch, Age, NIC, Scale, Salary, JoiningDate, OvertimeHours, JobTitle, DepartmentId)

VALUES (2, 'Ali', 'ahmad', 'NY', 30, '123456789012345', 1, 75000, TO\_DATE('2022-05-10', 'YYYY-MM-DD'), 5.5, 'Manager', 1);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Branch, Age, NIC, Scale, Salary, JoiningDate, OvertimeHours, JobTitle, DepartmentId)

VALUES (3, 'Hamza', 'khan', 'NY', 30, '123456789012345', 1, 85000, TO\_DATE('2022-05-10', 'YYYY-MM-DD'), 5.5, 'Manager', 1);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Branch, Age, NIC, Scale, Salary, JoiningDate, OvertimeHours, JobTitle, DepartmentId)

VALUES (4, 'Bial', 'khan', 'NY', 30, '123456789012345', 1, 95000, TO\_DATE('2022-05-10', 'YYYY-MM-DD'), 5.5, 'Manager', 1);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Branch, Age, NIC, Scale, Salary, JoiningDate, OvertimeHours, JobTitle, DepartmentId)

VALUES (5, 'Faraz', 'maqsood', 'NY', 30, '123456789012345', 1, 175000, TO\_DATE('2022-05-10', 'YYYY-MM-DD'), 5.5, 'Manager', 1);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Branch, Age, NIC, Scale, Salary, JoiningDate, OvertimeHours, JobTitle, DepartmentId)

VALUES (6, 'sameer', 'khan', 'NY', 30, '123456789012345', 1, 5000, TO\_DATE('2022-05-10', 'YYYY-MM-DD'), 5.5, 'Manager', 1);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Branch, Age, NIC, Scale, Salary, JoiningDate, OvertimeHours, JobTitle, DepartmentId)

VALUES (7, 'Hammad', 'Ijaz', 'NY', 30, '123456789012345', 1, 705000, TO\_DATE('2022-05-10', 'YYYY-MM-DD'), 5.5, 'Manager', 1);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Branch, Age, NIC, Scale, Salary, JoiningDate, OvertimeHours, JobTitle, DepartmentId)

VALUES (8, 'usman', 'khan', 'NY', 30, '123456789012345', 1, 750000, TO\_DATE('2022-05-10', 'YYYY-MM-DD'), 5.5, 'Manager', 1);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Branch, Age, NIC, Scale, Salary, JoiningDate, OvertimeHours, JobTitle, DepartmentId)

VALUES (9, 'khurram', 'Riaz', 'NY', 30, '123456789012345', 1, 75000, TO\_DATE('2022-05-10', 'YYYY-MM-DD'), 5.5, 'Manager', 1);

INSERT INTO Employees (EmployeeID, FirstName, LastName, Branch, Age, NIC, Scale, Salary, JoiningDate, OvertimeHours, JobTitle, DepartmentId)

VALUES (10, 'raheel', 'shareef', 'NY', 30, '123456789012345', 1, 55000, TO\_DATE('2022-05-10', 'YYYY-MM-DD'), 5.5, 'Manager', 1);

CREATE TABLE Department (

DepartmentID NUMBER PRIMARY KEY,

DepartmentName VARCHAR2(50)

);

-- Sample data for Departments table

INSERT INTO Department (DepartmentID, DepartmentName)

VALUES (1, 'HR');

INSERT INTO Department (DepartmentID, DepartmentName)

VALUES (2, 'Sales');

INSERT INTO Department (DepartmentID, DepartmentName)

VALUES (3, 'Finance');

-- Order table

CREATE TABLE Orders (

Order\_ID Int PRIMARY KEY,

Customer\_ID Int,

Order\_Date Date

);

-- Sample data for Orders table

INSERT INTO Orders (Order\_ID, Customer\_ID, Order\_Date)

VALUES

(101, 1, '2023-01-15'),

(102, 2, '2023-02-20'),

(103, 3, '2023-03-25'),

-- Add more orders as needed

;

CREATE TABLE Products (

Product\_ID Int PRIMARY KEY,

Product\_Name VARCHAR(50),

Unit\_Price float

);

-- Sample data for Products table

INSERT INTO Products (Product\_ID, Product\_Name, Unit\_Price)

VALUES (201, 'Product A', 10.99);

INSERT INTO Products (Product\_ID, Product\_Name, Unit\_Price)

VALUES (202, 'Product B', 12.99);

INSERT INTO Products (Product\_ID, Product\_Name, Unit\_Price)

VALUES (203, 'Product C', 13.99);

INSERT INTO Products (Product\_ID, Product\_Name, Unit\_Price)

VALUES (204, 'Product D', 14.99);

CREATE TABLE Customers (

Customer\_ID Int PRIMARY KEY,

Customer\_Name VARCHAR(55),

Sales\_Representative\_ID Int

);

-- Sample data for Customers table

INSERT INTO Customers (Customer\_ID, Customer\_Name, Sales\_Representative\_ID)

VALUES

(1, 'Customer X', 1),

(2, 'Customer Y', 2),

(3, 'Customer Z', 1),

;

CREATE TABLE OrderDetails (

Order\_ID Int,

Product\_ID Int,

Quantity Int

);

-- Sample data for OrderDetails table

INSERT INTO OrderDetails (Order\_ID, Product\_ID, Quantity)

VALUES (101, 201, 5);

INSERT INTO OrderDetails (Order\_ID, Product\_ID, Quantity)

VALUES (102, 202, 15);

INSERT INTO OrderDetails (Order\_ID, Product\_ID, Quantity)

VALUES (103, 203, 25);

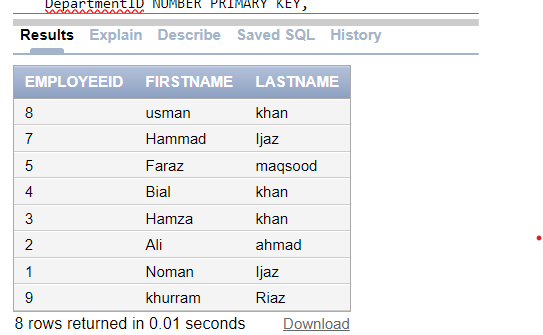
Task #1:

SELECT EmployeeID, FirstName, LastName

FROM Employees

WHERE Salary > 70000

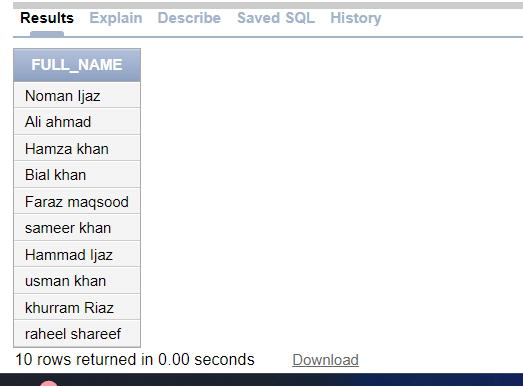
ORDER BY Salary DESC;



Task #2:

SELECT first\_name || ' ' || last\_name AS full\_name

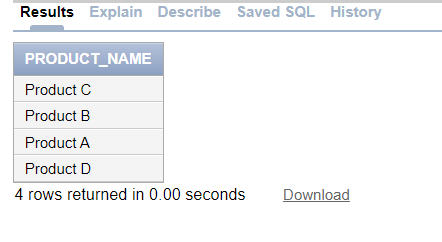
FROM Employees;



Task #3:

SELECT DISTINCT Product\_Name

FROM Products;



Task #4:

SELECT Order\_Id, Customer\_Id, Order\_Date

FROM Orders

WHERE Order\_date > TO\_DATE('2023-01-01', 'YYYY-MM-DD');

Task #5:

SELECT \*

FROM Customers

WHERE City = 'New York' OR City = 'Los Angeles';



**Reason :**Because “City ” is not part of Customers table.Customers have no attribute of “City”

Task #6:

SELECT

P.Product\_Name,

SUM(OD.Quantity) AS Total\_Units\_Sold,

SUM(OD.Quantity \* P.Unit\_Price) AS Total\_Revenue

FROM

Products P

JOIN

OrderDetails OD ON P.Product\_ID = OD.Product\_ID

JOIN

Orders O ON OD.Order\_ID = O.Order\_ID

WHERE

TO\_CHAR(O.Order\_Date, 'YYYY') = '2023'

GROUP BY

P.Product\_Name

ORDER BY

Total\_Revenue DESC;



Task #7:

SELECT

D.DepartmentName,

COUNT(E.EmployeeID) AS TotalEmployees,

AVG(E.Salary) AS AverageSalary

FROM

Department D

JOIN

Employees E ON D.DepartmentID = E.DepartmentID

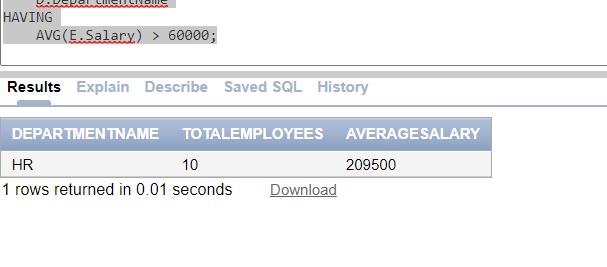
GROUP BY

D.DepartmentName

HAVING

AVG(E.Salary) > 60000;

+



Task #8:

SELECT

C.Customer\_Name,

COUNT(O.Order\_ID) AS TotalOrders

FROM

Customers C

JOIN

Orders O ON C.Customer\_ID = O.Customer\_ID

GROUP BY

C.Customer\_Name

HAVING

COUNT(O.Order\_ID) > 3;

Output:

Task #9:

SELECT

EmployeeID,

FirstName,

LastName,

CASE

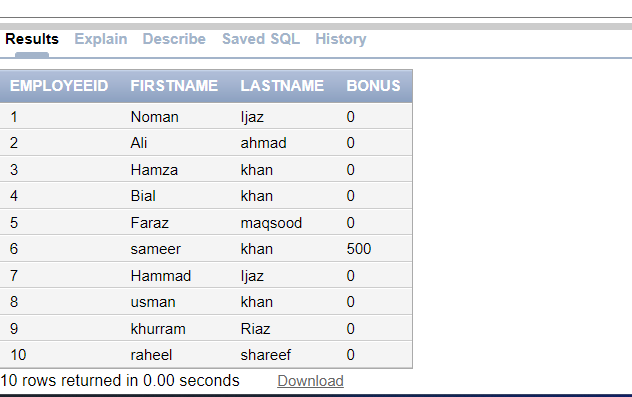
WHEN Salary < 50000 THEN Salary \* 0.10

ELSE 0

END AS Bonus

FROM

Employees;



Task #10:

SELECT \*

FROM Employees

ORDER BY (Salary + (CASE WHEN Salary < 50000 THEN Salary \* 0.10 ELSE 0 END)) DESC

FETCH FIRST 5 ROWS ONLY;

# Task #11:

SELECT \*

FROM (

SELECT

EmployeeID,

FirstName,

LastName,

(Salary + (CASE WHEN Salary < 50000 THEN Salary \* 0.10 ELSE 0 END)) AS CombinedEarnings

FROM

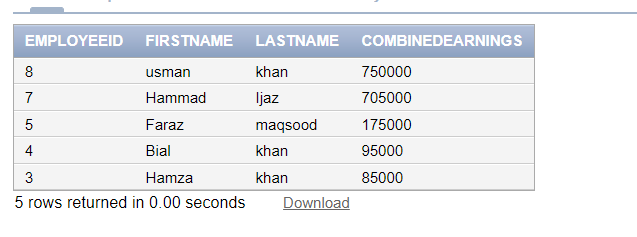
Employees

ORDER BY

CombinedEarnings DESC

)

WHERE ROWNUM <= 5;



# Task #12:

SELECT

product\_id,

SUM(Quantity) AS TotalQuantitySold

FROM

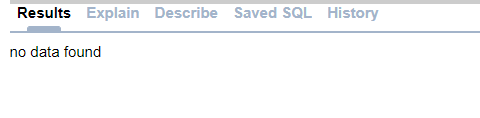
OrderDetails

GROUP BY

Product\_Id

HAVING

SUM(Quantity) > 50;



# Task #13:

SELECT

C.Customer\_Name,

SUM(O.order\_amount) AS TotalOrderAmount

FROM

Customers C

JOIN

Orders O ON C.Customer\_Id = O.Customer\_Id

WHERE

TO\_CHAR(O.Order\_Date, 'YYYY') = '2023'

GROUP BY

C.Customer\_Name

ORDER BY

TotalOrderAmount DESC

FETCH FIRST 1 ROW ONLY;

# Task #14

SELECT location, AVG(order\_count) AS avg\_orders

FROM (

SELECT c.location, COUNT(o.order\_id) AS order\_count

FROM Customers c

LEFT JOIN Orders o ON c.customer\_id = o.customer\_id

GROUP BY c.location

)

GROUP BY location;

# Task #15

SELECT employee\_id, first\_name, last\_name, hire\_date

FROM employees

WHERE EXTRACT(YEAR FROM hire\_date) = 2022

ORDER BY hire\_date ASC;

# Task #16

SELECT product\_name, unit\_price

FROM Products

WHERE unit\_price < (SELECT AVG(unit\_price) FROM Products);

# Task #17

SELECT \*

FROM Customers

WHERE customer\_id IN (SELECT DISTINCT customer\_id FROM Orders WHERE EXTRACT(YEAR FROM order\_date) = 2023)

ORDER BY customer\_name;

# Task #18

SELECT o.order\_id, o.customer\_id, o.order\_date

FROM Orders o

JOIN Customers c ON o.customer\_id = c.customer\_id

WHERE c.sales\_rep\_id IN (SELECT employee\_id FROM employees WHERE department\_id = (SELECT department\_id FROM departments WHERE department\_name = 'Marketing'))

ORDER BY o.order\_date ASC;

# task #19

SELECT e.employee\_id, e.first\_name, e.last\_name, COUNT(o.order\_id) AS total\_orders\_placed

FROM employees e

LEFT JOIN Orders o ON e.employee\_id = o.sales\_rep\_id

GROUP BY e.employee\_id, e.first\_name, e.last\_name

HAVING COUNT(o.order\_id) >= 10;

# task #20

OrderDetails tables to compute this, and display the category name along with the total revenue.

SELECT p.Category, SUM(od.quantity \* p.unit\_price) AS total\_revenue

FROM Products p

JOIN OrderDetails od ON p.ProductId = od.ProductId

GROUP BY p.Category