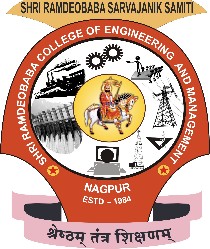
**Shri Ramdeobaba College of Engineering & Management Nagpur-13**

**Department of Computer Application**

**Session: 2023-2024**



**Submission for**

**Course Name:** Database Management Systems Lab

**Course Code:** MCP545

**Name of the Student:** Jayesh Lalit Nandanwar

**Class Roll No:** 26

**Semester:** MCA II semester

**Shift:** 2

**Batch:** 2

Under the Guidance of

Prof. Yoginee Pethe

Date of submission:

**List of Experiments**

1. Implementation of DDL commands of SQL with suitable examples
   1. Create table b) Alter table c) Truncate table d) Drop table

Implementation of DML commands of SQL with suitable examples

1. Select b) Insert c) Update d) Delete
2. Study and implementation of different types of constraints.
3. Implementation of different types of operators in SQL

a)Arithmetic Operators b) Logical Operators

c) Comparison Operators d) Set Operators

1. Study and Implementation of
   1. Aggregate functions
   2. Group By & Having clause
   3. Order by clause
2. Study and Implementation of different types of joins like cross join, natural join, inner join, and outer joins.
3. Study and Implementation of Sub queries
4. Study and Implementation of views, and synonyms.
5. Study and Implementation of PL/SQL.

**Practical 6**

**Aim:** Study and Implementation of Sub queries

**Solution:**

Employee (emp\_no, emp\_name, emp\_sal, emp\_comm, dept\_no, Job\_id, location)

Job (job\_id, job\_title, min\_sal, max\_sal)

department (dept\_no, dept\_name, location\_id).

employee\_manager (emp\_no, emp\_hiredate, mng\_no,mng\_name).

**Query:** CREATE TABLE Employee (

emp\_no INT PRIMARY KEY,

emp\_name VARCHAR(100),

emp\_sal DECIMAL(10, 2),

emp\_comm DECIMAL(10, 2),

dept\_no INT, Job\_id INT,

location VARCHAR(100)

);

CREATE TABLE Job (

job\_id INT PRIMARY KEY,

job\_title VARCHAR(100),

min\_sal DECIMAL(10, 2),

max\_sal DECIMAL(10, 2)

);

CREATE TABLE Department (

dept\_no INT PRIMARY KEY,

dept\_name VARCHAR(100),

location\_id INT

);

CREATE TABLE Employee\_Manager (

emp\_no INT PRIMARY KEY,

emp\_hiredate DATE, mng\_no INT,

mng\_name VARCHAR(100)

);

1. Create a report that displays the employee number, name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary

**Query:** SELECT emp\_no, emp\_name, emp\_sal FROM Employee

WHERE emp\_sal > (SELECT AVG(emp\_sal) FROM Employee)

ORDER BY emp\_sal ASC;

1. The HR department needs a report that displays the name, department number, and job ID of all employees whose department location ID is 1700.

**Query:** SELECT e.emp\_name, d.dept\_no, e.job\_id FROM Employee e

JOIN department d ON e.dept\_no = d.dept\_no

WHERE d.location\_id = 1700;

1. Create a report for HR that displays the department number, name, and job ID for every employee in the Executive department

**Query:** SELECT d.dept\_no, d.dept\_name, e.job\_id

FROM Employee e JOIN department d ON e.dept\_no = d.dept\_no

WHERE d.dept\_name = 'Executive';

1. Create a report that displays a list of all employees whose salary is more than the salary of any employee from department 60..

**Query:** SELECT emp\_no, emp\_name, emp\_sal FROM Employee

WHERE emp\_sal > ANY (SELECT emp\_sal FROM Employee

WHERE dept\_no = 60

);

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| customer\_id | first\_name | last\_name | age | country |
| 1 | John | Doe | 31 | USA |
| 2 | Robert | Luna | 22 | USA |
| 3 | David | Robinson | 22 | UK |
| 4 | John | Reinhardt | 25 | UK |
| 5 | Betty | Doe | 28 | UAE |

Table: Customers

|  |  |  |
| --- | --- | --- |
| order\_id | amount | customer\_id |
| 1 | 200 | 4 |
| 2 | 500 | 1 |
| 3 | 300 | 3 |
| 4 | 800 | 1 |
| 5 | 150 | 2 |

Table: Orders

CREATE TABLE Customer ( Customer\_id INT PRIMARY KEY, First\_name

VARCHAR(50), Last\_name VARCHAR(50), Age INT, Country VARCHAR(50)

);

INSERT INTO Customer (Customer\_id, First\_name, Last\_name, Age, Country)

VALUES(1, 'John', 'Doe', 31,'USA');

INSERT INTO Customer (Customer\_id, First\_name, Last\_name, Age, Country)

VALUES(2, 'Robert', 'Luna', 22, 'USA');

INSERT INTO Customer (Customer\_id, First\_name, Last\_name, Age, Country)

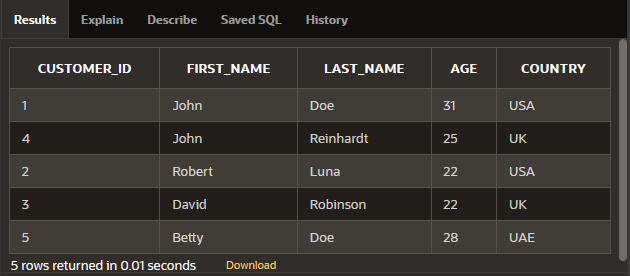
VALUES(3, 'David', 'Robinson', 22, 'UK');

INSERT INTO Customer (Customer\_id, First\_name, Last\_name, Age, Country)

VALUES(4, 'John', 'Reinhardt', 25, 'UK');

INSERT INTO Customer (Customer\_id, First\_name, Last\_name, Age, Country)

VALUES(5, 'Betty', 'Doe', 28, 'UAE');



CREATE TABLE Orders (

Order\_id INT PRIMARY KEY,

Amount INT,

Customer\_id INT,

FOREIGN KEY (Customer\_id) REFERENCES Customer(Customer\_id)

);

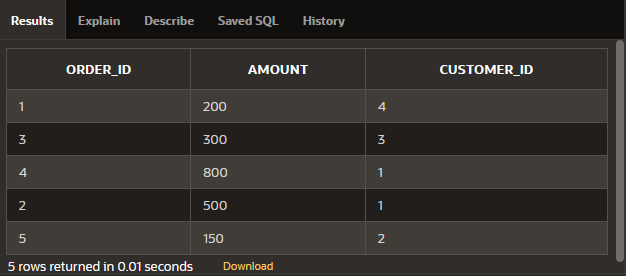
INSERT INTO Orders (Order\_id, Amount, Customer\_id) VALUES(1, 200, 4);

INSERT INTO Orders (Order\_id, Amount, Customer\_id) VALUES(2, 500, 1);

INSERT INTO Orders (Order\_id, Amount, Customer\_id) VALUES(3, 300, 3);

INSERT INTO Orders (Order\_id, Amount, Customer\_id) VALUES(4, 800, 1);

INSERT INTO Orders (Order\_id, Amount, Customer\_id) VALUES(5, 150, 2);



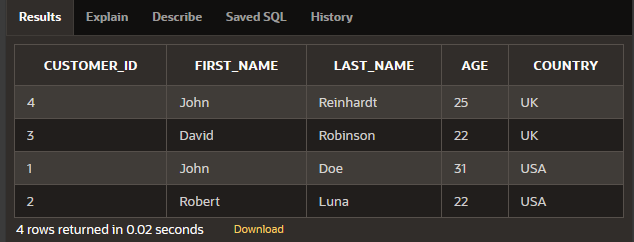
1. Find the details of customers who have placed an order.

**Query:** SELECT c.\* FROM Customer c

WHERE EXISTS (SELECT 1 FROM Orders o

WHERE o.Customer\_id = c.Customer\_id

);



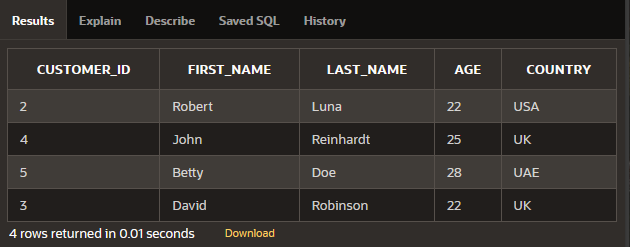
1. Find all customers who do not have order more than $500.

**Query:** SELECT c.\* FROM Customer c

WHERE NOT EXISTS (SELECT 1 FROM Orders o

WHERE o.Customer\_id = c.Customer\_id

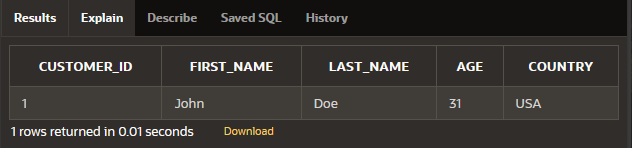
AND o.Amount > 500 );



1. Find the customers who have placed highest order.

**Query:** SELECT c.\* FROM Customer c JOIN Orders o ON c.Customer\_id = o.Customer\_id

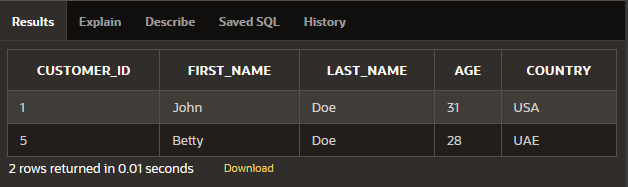
WHERE o.Amount = (SELECT MAX(Amount) FROM Orders);



1. Find all customers whose age is greater than the average age of all customers.

**Query:** SELECT \* FROM Customer

WHERE Age > (SELECT AVG(Age) FROM Customer)



1. Find the name of customers whose age is equal to age of John.

**Query:** SELECT First\_name, Last\_name FROM Customer

WHERE Age = (SELECT MAX(Age) FROM Customer

WHERE First\_name = 'John');

