Practical 4

1. **Arithmetic Operators :**

-- Create a Table and Insert a data into it

```
CREATE TABLE Employee
(
Employee_ID INT AUTO_INCREMENT PRIMARY KEY,
Emp_Name VARCHAR (50),
Emp_City VARCHAR (20),
Emp_Salary INT NOT NULL,
Emp_Bonus INT NOT NULL
) ;
INSERT INTO Employee (Employee_ID, Emp_Name, Emp_City, Emp_Salary,
Emp_Bonus) VALUES (101, Anuj, Ghaziabad, 25000, 2000),
(102, Tushar, Lucknow, 29000, 1000),
(103, Vivek, Kolkata, 35000, 2500),
(104, Shivam, Goa, 22000, 3000);
```

Output

Employee

Employee_ID	Emp_Name	Emp_City	Emp_Salary	Emp_Bonus
101	Anuj	Ghaziabad	25000	2000
102	Tushar	Lucknow	29000	1000
103	Vivek	Kolkata	35000	2500
104	Shivam	Goa	22000	3000

--The following query adds the Emp_Salary and Emp_Bonus of each employee of the Employee table using the addition o perator:

```
SELECT * FROM Employee;

SELECT Emp_Salary + Emp_Bonus AS Emp_Total_Salary FROM Employee;
```

Emp_Total_Salary
27000
30000
37500
25000

2. **Logical Operator :**

```
CREATE TABLE CUSTOMERS(
   ID INT NOT NULL,
   NAME VARCHAR(15) NOT NULL,
   AGE INT NOT NULL,
   ADDRESS CHAR(25),
   SALARY DECIMAL(18, 2),
   PRIMARY KEY(ID)
);
INSERT INTO CUSTOMERS(ID, NAME, AGE, ADDRESS, SALARY) VALUES(1, 'Ramesh',
32, 'Ahmedabad', 2000.00);
INSERT INTO CUSTOMERS(ID, NAME, AGE, ADDRESS, SALARY) VALUES(2, 'khilan', 25,
'Delhi', 1500.00);
INSERT INTO CUSTOMERS(ID, NAME, AGE, ADDRESS, SALARY) VALUES(3, 'Kaushik',
23, 'Kota', 2000.00);
INSERT INTO CUSTOMERS(ID, NAME, AGE, ADDRESS, SALARY) VALUES(4, 'chaitali',
25, 'Mumbai', 6500.00);
INSERT INTO CUSTOMERS(ID, NAME, AGE, ADDRESS, SALARY) VALUES(5, 'Hardhik',
27, 'Bhopal', 8500.00);
INSERT INTO CUSTOMERS(ID, NAME, AGE, ADDRESS, SALARY) VALUES(6, 'komal', 22,
'MP', 4500.00);
INSERT INTO CUSTOMERS(ID, NAME, AGE, ADDRESS, SALARY) VALUES(7, 'Muffy', 24,
'Indore', 10000.00 );
```

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000
2	khilan	25	Delhi	1500
3	Kaushik	23	Kota	2000
4	chaitali	25	Mumbai	6500
5	Hardhik	27	Bhopal	8500
6	komal	22	MP	4500
7	Muffy	24	Indore	10000

```
SELECT * FROM CUSTOMERS;
```

SQL AND Operator:

--In here, we are fetching the ID, Name and Salary of the customers whose salary is greater than 2000 and age is less than 25 years.

```
SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY > 2000 AND age < 25;
```

ID	NAME	SALARY
6	komal	4500
7	Muffy	10000

3. **Comparison Operator:**

Example of SQL NOT EQUAL operator

```
CREATE TABLE Cars
(
Car_Number INT PRIMARY KEY,
Car_Name VARCHAR (50),
Car_Price INT NOT NULL,
Car_AmountINT NOT NULL
) ;
INSERT INTO Cars (Car_Number, Car_Name, Car_Amount, Car_Price)
VALUES (2578, Creta, 3, 1500000),
(9258, Audi, 2, 3000000),
(8233, Venue, 6, 900000),
(6214, Nexon, 7, 1000000);

SELECT * FROM Cars;
```

Car_Number	Car_Name	Car_Price	Car_Amount
2578	Creta	1500000	3
9258	Audi	3000000	2
8233	Venue	900000	6
6214	Nexon	1000000	7

--The following query shows the record of those cars from the Cars table whose Car_Price is not equal to 900000:

```
SELECT * FROM Cars WHERE Car_Price != 900000;
```

Car_Number	Car_Name	Car_Price	Car_Amount
2578	Creta	1500000	3
9258	Audi	3000000	2
6214	Nexon	1000000	7