



Global Academy of Technology

*Growing Ahead Of Time....*

Autonomous Institute, Affiliated To VTU

# Department of Artificial Intelligence & Machine Learning

## AML23403

### Database Management System Laboratory

**VASUGI I**

ASSISTANT PROFESSOR

DEPARTMENT OF AI & ML

# Exercise-

Consider the following schema for Order Database:

**SALESMAN(Salesman\_id, Name, City, Commission)**

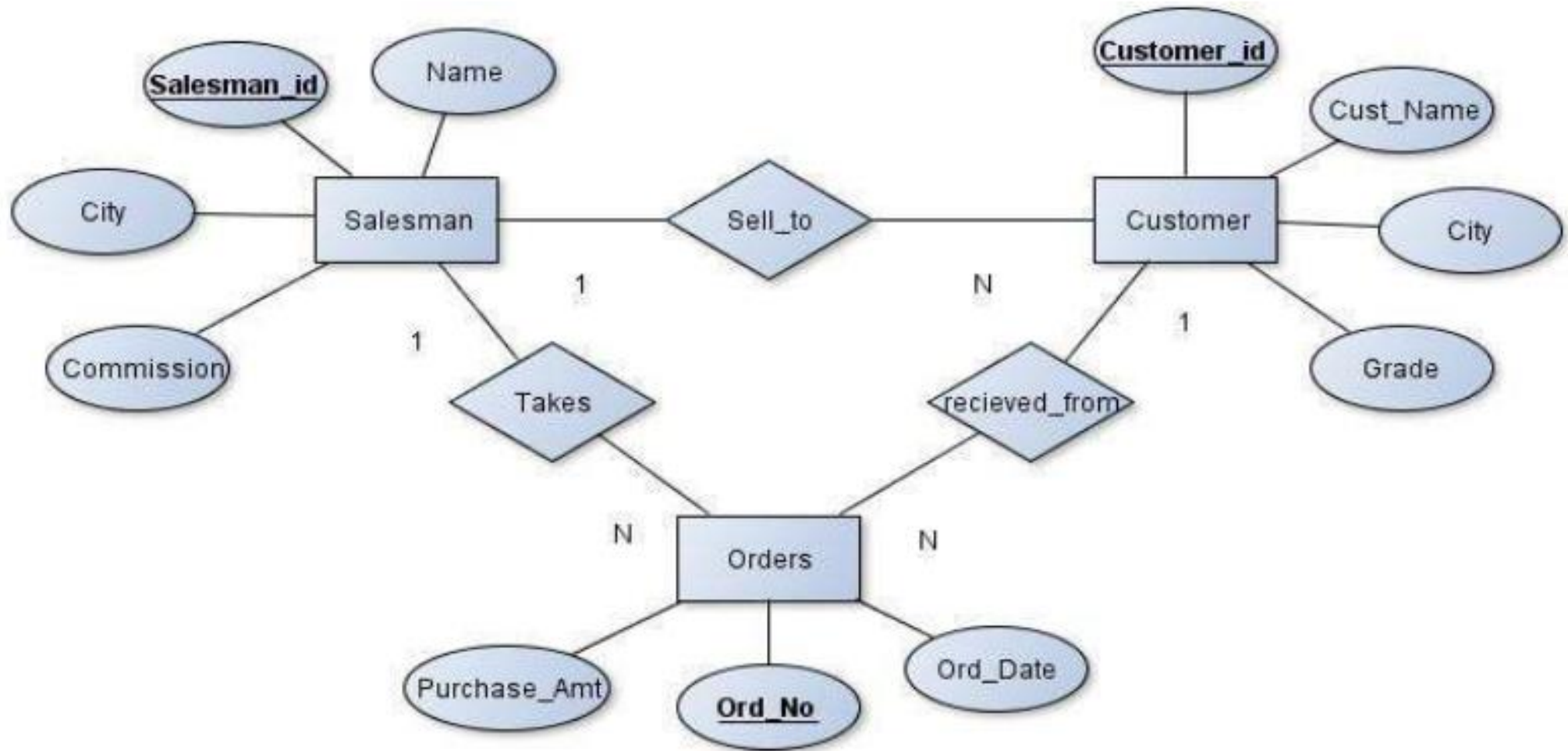
**CUSTOMER(Customer\_id, Cust\_Name, City, Grade, Salesman\_id)**

**ORDERS(Ord\_No, Purchase\_Amt, Ord\_Date, Customer\_id,  
Salesman\_id)**

# Write SQL queries to

1. Count the customers with grades above Bangalore's average.
2. Find the name and numbers of all salesman who had more than one customer.
3. List all the salesman and indicate those who have and do not have customers in their cities (Use UNION operation.)
4. Create a view that finds the salesman who has the customer with the highest order of a day.
5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

## Entity-Relationship Diagram



# Schema Diagram

# Table Creation

1. CREATE TABLE SALESMAN(SALESMAN\_ID INTEGER (4),NAME VARCHAR (20), CITY VARCHAR (20), COMMISSION VARCHAR (20), PRIMARY KEY (SALESMAN\_ID));
2. CREATE TABLE CUSTOMER(CUSTOMER\_ID INTEGER, CUST\_NAME VARCHAR (20), CITY VARCHAR(20), GRADE INTEGER, PRIMARY KEY (CUSTOMER\_ID), SALESMAN\_ID INTEGER, FOREIGN KEY (SALESMAN\_ID) REFERENCES SALESMAN (SALESMAN\_ID) ON DELETE SET NULL);
3. CREATE TABLE ORDERS (ORD\_NO INTEGER, PURCHASE\_AMT BIGINT, ORD\_DATE DATE, PRIMARY KEY (ORD\_NO), CUSTOMER\_ID INTEGER, SALESMAN\_ID INTEGER, FOREIGN KEY(CUSTOMER\_ID)REFERENCES CUSTOMER(CUSTOMER\_ID) ON DELETE CASCADE, FOREIGN KEY(SALESMAN\_ID) REFERENCES SALESMAN (SALESMAN\_ID) ON DELETE CASCADE);

# Table Description

- DESC SALESMAN;
- DESC CUSTOMER;
- DESC ORDERS;

# Insertion of Values to Tables

## 1. *INSERT INTO SALESMAN TABLE:*

- INSERT INTO SALESMAN VALUES (1000, "AKASH","BANGALORE","25 %");
- INSERT INTO SALESMAN VALUES (2000, "RAVI","BANGALORE","20 %");
- INSERT INTO SALESMAN VALUES (3000, "KUMAR","MYSORE","15 %");
- INSERT INTO SALESMAN VALUES (4000, "SMITH","DELHI","30 %");
- INSERT INTO SALESMAN VALUES (5000, "HARSHA","HYDRABAD","15%");
- SELECT \* FROM SALESMAN;



# Insertion of Values to Tables

## 2. *INSERT INTO CUSTOMER TABLE:*

- INSERT INTO CUSTOMER VALUES (10, "PREETHI","BANGALORE", 100, 1000);
- INSERT INTO CUSTOMER VALUES (11, "VIVEK","MANGALORE", 300, 1000);
- INSERT INTO CUSTOMER VALUES (12, "BHASKAR","CHENNAI", 400, 2000);
- INSERT INTO CUSTOMER VALUES (13, "CHETHAN","BANGALORE", 200, 2000);
- INSERT INTO CUSTOMER VALUES (14, "MAMATHA","BANGALORE", 400, 3000);
- SELECT \*FROM CUSTOMER;

# Insertion of Values to Tables

## *3. INSERT INTO ORDERS TABLE:*

- INSERT INTO ORDERS VALUES (50, 5000, "2021-05-22", 10, 1000);
- INSERT INTO ORDERS VALUES (51, 450, "2021-05-22", 10, 2000);
- INSERT INTO ORDERS VALUES (52, 1000, "2023-02-05", 13, 2000);
- INSERT INTO ORDERS VALUES (53, 3500, "2021-04-13", 14, 3000);
- INSERT INTO ORDERS VALUES (54, 550, "2021-03-09", 12,2000);
  
- SELECT \*FROM ORDERS;

# Query-1

Count the customers with grades above Bangalore's average.

```
SELECT GRADE, COUNT(DISTINCT CUSTOMER_ID)
FROM CUSTOMER
GROUP BY GRADE
HAVING GRADE > (SELECT AVG(GRADE)
                FROM CUSTOMER
                WHERE CITY='BANGALORE');
```

## Query-2

Find the name and numbers of all salesmen who had more than one customer

```
SELECT SALESMAN_ID, NAME  
FROM SALESMAN A  
WHERE 1 < (SELECT COUNT(*)  
           FROM CUSTOMER  
           WHERE SALESMAN_ID=A.SALESMAN_ID);
```

## Query-3

List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation)

```
SELECT SALESMAN.SALESMAN_ID, NAME, CUST_NAME, COMMISSION  
FROM SALESMAN, CUSTOMER  
WHERE SALESMAN.CITY = CUSTOMER.CITY
```

UNION

```
SELECT SALESMAN_ID, NAME, 'NO MATCH', COMMISSION  
FROM SALESMAN  
WHERE NOT CITY = ANY(SELECT CITY  
                      FROM CUSTOMER)  
ORDER BY 2 DESC;
```

## Query-4

Create a view that finds the salesman who has the customer with the highest order of a day.

```
CREATE VIEW ELITSALESMAN AS
SELECT B.ORD_DATE, A.SALESMAN_ID, A.NAME
FROM SALESMAN A, ORDERS B
WHERE A.SALESMAN_ID = B.SALESMAN_ID
AND
B.PURCHASE_AMT=(SELECT MAX(PURCHASE_AMT)
                  FROM ORDERS C
                  WHERE C.ORD_DATE = B.ORD_DATE);

SELECT *FROM ELITSALESMAN;
```

## Query-5

Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted

```
DELETE FROM SALESMAN  
WHERE SALESMAN_ID=1000;
```

*(CHECK DATABASE)*

```
SELECT *FROM SALESMAN;
```