

Program 6 : Order Database

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 salesmen who had more than one customer.
3. List all salesmen and indicate those who have and don't 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.

SELECT * FROM SALESMAN;

SALESMAN_ID	NAME	CITY	COMMISSION
1000	JOHN	BANGALORE	25 %
2000	RAVI	BANGALORE	20 %
3000	KUMAR	MYSORE	15 %
4000	SMITH	DELHI	30 %
5000	HARSHA	HYDRABAD	15 %

SELECT * FROM CUSTOMER1;

CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID
10	PREETHI	BANGALORE	100	1000
11	VIVEK	MANGALORE	300	1000
12	BHASKAR	CHENNAI	400	2000
13	CHETHAN	BANGALORE	200	2000
14	MANATHA	BANGALORE	400	3000

SELECT * FROM ORDERS;

ORD_NO	PURCHASE_AMT	ORD_DATE	CUSTOMER_ID	SALESMAN_ID
50	5000	04-MAY-17	10	1000
51	450	20-JAN-17	10	2000
52	1000	24-FEB-17	13	2000
53	3500	13-APR-17	14	3000
54	550	09-MAR-17	12	2000

```
CREATE DATABASE ORDER_DATABASE;
```

```
USE ORDER_DATABASE;
```

```
CREATE TABLE SALESMAN
```

```
(
```

```
    Salesman_id int primary key,
```

```
    Name varchar(40),
```

```
    City varchar(20),
```

```
    Commission decimal
```

```
);
```

```
CREATE TABLE CUSTOMER
```

```
(
```

```
    Customer_id int primary key,
```

```
    Cust_Name varchar(40),
```

```
    City varchar(20),
```

```
    Grade int,
```

```
    Salesman_id int,
```

```
    Foreign Key(Salesman_id) references SALESMAN(Salesman_id)
```

```
);
```

```
CREATE TABLE ORDERS
```

```
(
```

```
    Ord_No int primary key,
```

```
    Purchase_Amt int,
```

```
    Ord_Date Date,
```

```
    Customer_id int,
```

```
    Salesman_id int,
```

```
    Foreign Key(Salesman_id) references SALESMAN(Salesman_id),
```

Foreign Key(Customer_id) references CUSTOMER(Customer_id)

);

insert into SALESMAN

values (1000,"John","Bangalore",25),

(2000,"Ravi","Bangalore",20),

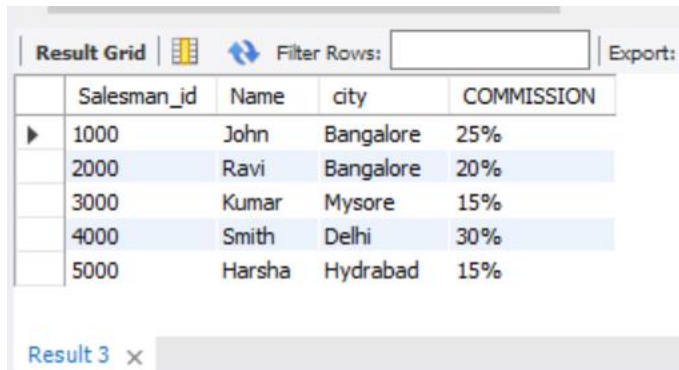
(3000,"Kumar","Mysore",15),

(4000,"Smith","Delhi",30),

(5000,"Harsha","Hydrabad",15);

SELECT Salesman_id,Name,city,concat(Commission,'%') AS COMMISSION FROM SALESMAN;

select * from SALESMAN;



The screenshot shows a database query result grid for the SALESMAN table. The grid has a header row with columns: Salesman_id, Name, city, and COMMISSION. There are five data rows. The first row is highlighted with a blue background. The grid is titled 'Result 3' and has a close button (x) next to it.

Salesman_id	Name	city	COMMISSION
1000	John	Bangalore	25%
2000	Ravi	Bangalore	20%
3000	Kumar	Mysore	15%
4000	Smith	Delhi	30%
5000	Harsha	Hydrabad	15%

insert into CUSTOMER

values (10,"Preethi","Bangalore",100,1000),

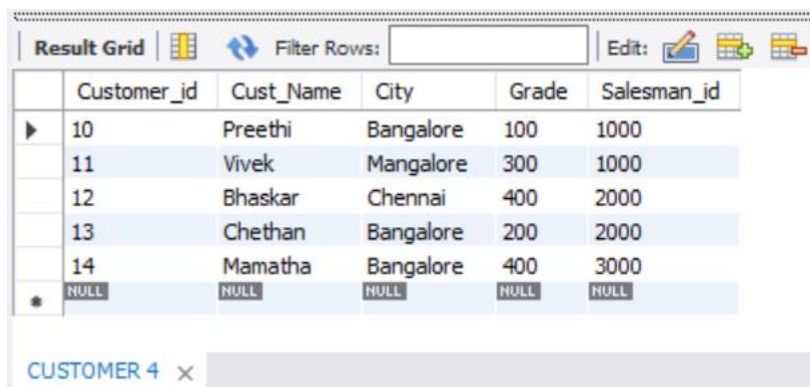
(11,"Vivek","Mangalore",300,1000),

(12,"Bhaskar","Chennai",400,2000),

(13,"Chethan","Bangalore",200,2000),

(14,"Mamatha","Bangalore",400,3000);






select * from CUSTOMER;



The screenshot shows a database query result grid for the CUSTOMER table. The grid has a header row with columns: Customer_id, Cust_Name, City, Grade, and Salesman_id. There are five data rows. The first row is highlighted with a blue background. The grid is titled 'CUSTOMER 4' and has a close button (x) next to it.

Customer_id	Cust_Name	City	Grade	Salesman_id
10	Preethi	Bangalore	100	1000
11	Vivek	Mangalore	300	1000
12	Bhaskar	Chennai	400	2000
13	Chethan	Bangalore	200	2000
14	Mamatha	Bangalore	400	3000

insert into ORDERS

Result Grid			 Filter Rows: <input type="text"/>	Edit:   	
	Ord_No	Purchase_Amt	Ord_Date	Customer_id	Salesman_id
▶	50	5000	2017-05-04	10	1000
	51	450	2017-01-20	10	2000
	52	1000	2017-02-24	13	2000
	53	3500	2017-04-13	14	3000
	54	550	2017-03-09	12	2000
✱	NULL	NULL	NULL	NULL	NULL

- ```
select count(Cust_Name)
from CUSTOMER
where grade> (Select avg(grade)
 from CUSTOMER
 where city ='Bangalore');
```

The screenshot shows the 'Result Grid' tab in SQL Server Enterprise Manager. The query editor contains the text `count(Cust_Name)`. Below the query editor, the result grid displays a single row with the value `3`. The 'Filter Rows' button is visible in the top right corner of the result grid area.

- [illegible]

| Result Grid |             |      | Filter Rows: |
|-------------|-------------|------|--------------|
|             | Salesman_id | name |              |
| ▶           | 1000        | John |              |
|             | 2000        | Ravi |              |

Result 4 x

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

```
select s.Name,c.Cust_name
from SALESMAN s,CUSTOMER c
where s.Salesman_id=c.Salesman_id and c.City=s.City
union
select s.Name,'No Match' from SALESMAN s,CUSTOMER c
where s.Salesman_id = c.Salesman_id and c.City!=s.City;
```

| Result Grid |       |           | Filter Rows: |
|-------------|-------|-----------|--------------|
|             | Name  | Cust_name |              |
| ▶           | John  | Preethi   |              |
|             | Ravi  | Chethan   |              |
|             | John  | No Match  |              |
|             | Ravi  | No Match  |              |
|             | Kumar | No Match  |              |

Result 6 x

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


```
create view Salesman_view as
select o.Ord_Date ,Salesman_id,sum(o.Purchase_Amt)
from ORDERS o
group by Ord_Date
having sum(Purchase_Amt)=(select max(sum(Purchase_Amt))from CUSTOMER
where Ord_Date =o.Ord_Date and Salesman_id =o.Salesman_id);
```


| Result Grid |            |             |                     | Filter Rows: | Export: |
|-------------|------------|-------------|---------------------|--------------|---------|
|             | Ord_Date   | Salesman_id | sum(o.Purchase_Amt) |              |         |
| ▶           | 2017-05-04 | 1000        | 5000                |              |         |
|             | 2017-01-20 | 2000        | 450                 |              |         |
|             | 2017-02-24 | 2000        | 1000                |              |         |
|             | 2017-04-13 | 3000        | 3500                |              |         |
|             | 2017-03-09 | 2000        | 550                 |              |         |

Salesman\_view 8 x

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;
select * from Salesman;
select * from Orders;
```

| Result Grid                                                                               |             |               |           |            |
|-------------------------------------------------------------------------------------------|-------------|---------------|-----------|------------|
| Filter Rows: <input type="text"/>                                                         |             |               |           |            |
| Edit:  |             |               |           |            |
|                                                                                           | salesman_id | salesman_name | city      | commission |
| ▶                                                                                         | 2000        | Ravi          | Bangalore | 20         |
|                                                                                           | 3000        | Kumar         | Mysore    | 15         |
|                                                                                           | 4000        | Smith         | Delhi     | 30         |
|                                                                                           | 5000        | Harsha        | Hyderabad | 15         |
| *                                                                                         | NULL        | NULL          | NULL      | NULL       |

| Result Grid                                                                               |          |              |            |             |             |
|-------------------------------------------------------------------------------------------|----------|--------------|------------|-------------|-------------|
| Filter Rows: <input type="text"/>                                                         |          |              |            |             |             |
| Edit:  |          |              |            |             |             |
|                                                                                           | order_id | purchase_amt | order_date | customer_id | salesman_id |
| ▶                                                                                         | 50       | 5000         | 2017-05-04 | 10          | NULL        |
|                                                                                           | 51       | 450          | 2017-01-20 | 10          | 2000        |
|                                                                                           | 52       | 1000         | 2017-02-24 | 13          | 2000        |
|                                                                                           | 53       | 3500         | 2017-04-13 | 14          | 3000        |
|                                                                                           | 54       | 550          | 2017-03-09 | 12          | 2000        |
| *                                                                                         | NULL     | NULL         | NULL       | NULL        | NULL        |