E-SHOPPING DATABASE MANAGEMENT SYSTEM

A PROJECT REPORT
SUBMITTED IN COMPLETE FULFILLMENT OF THE
REQUIREMENTS FOR THIRD SEMESTER PROJECT
OF
BACHELOR OF
TECHNOLOGY IN
[INFORMATION TECHNOLOGY]
IN
SUBJECT OF
DATABASE MANAGEMENT SYSTEM

Submitted by:
MEHAK GARG 2K20/IT/87

RAAVI SINGH 2K20/IT/111

Under the
supervision of
MS. GEETANJALI BHOLA

DELHI TECHNOLOGICAL UNIVERSITY
(FORMERLY Delhi College of

Engineering) Bawana Road,

Delhi-110042

CERTIFICATE

I hereby certify that the project titled ”**E-SHOPPING DATABASE MANAGEMENT SYSTEM**“ which is submitted by Mehak Garg(2K20/IT/87), Raavi Singh (2K20/IT/111) [Information technology], Delhi Technological University, Delhi in complete fulfillment of the requirement for the award of the degree of the Bachelor of Information Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University.

Place: Delhi

Date: 19.04.22

ACKNOWLEDGEMENT

In performing our major project, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this assignment gives us much pleasure. We would like to show our gratitude to Ms. Geetanjali Bhola, for this project. Giving us a good guideline for reporting throughout numerous consultations. We would also like to extend our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment.

Many people, our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us inspiration to improve our assignment.

We thank all the people for their help directly and indirectly to complete our assignment. In addition, we would like to thank the Department of Information Technology, Delhi Technological University for giving us the opportunity to work on this topic.

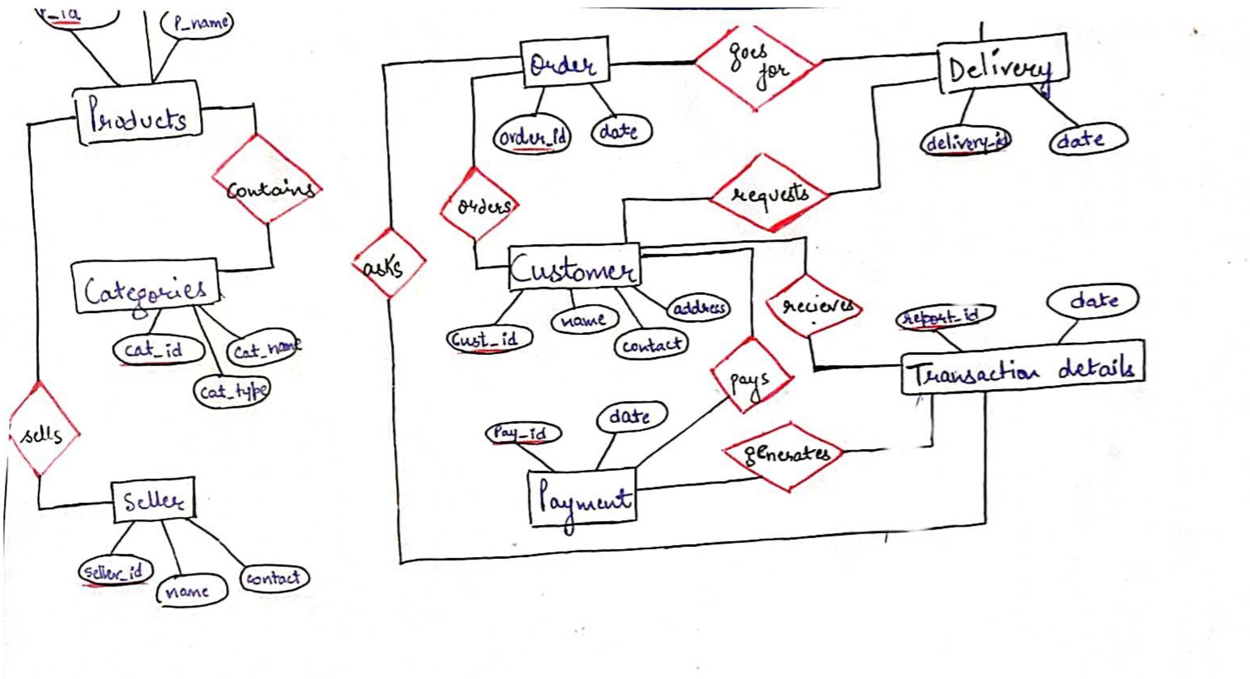
INTRODUCTION

As the name suggests, our project is basically to make a schema for the online shopping application. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through the internet. Thus the customer will get the service of online shopping and home delivery. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won’t be losing any more customers to the trending online shops such as Flipkart or eBay.

PROBLEM STATEMENT

- A shop wants to implement an E-Shopping Management System for its platform of Selling and Buying Products. First, they wish to store details of Customers like Customer_id,Name, contact and Address.
- The system stores details of Products in form of their categories like cat_id, cat_name, Along with this, Product details (Product Name, P_id) are also stored so that all Product details can be searched.
- Shopping Order (order_id and Date_of_order) maintains the details of order being placed by a customer.
- Delivery table stores information like (Delivery_id and Date_of_delivery) which helps to track the orders.
- Along with this a payment Table stores (Pay_id and Date_of Payment) informing about the payment details and finally when the order is placed and paid for, Transaction Details are generated having report_id and other details from different tables as a foreign Key.

ER DIAGRAM


P_id
P_name
Products
contains
Categories
cat_id
Cat_name
cat_type
sells
Seller
seller_id
name
contact
Order
goes for
Delivery
order_id
date
delivery_id
date
orders
requests
asks
Customer
Cust_id
name
contact
address
recieves
report_id
date
Transaction details
pays
Pay_id
date
generates
Payment

Concepts used

- Sequences
- Triggers
- Joins
- Normalizations
- DCL
- DDL
- Types of constraints
- Views
- Group by/Order by
- DML
- 5 aggregate functions

Tables

Table Name: Customer

Field	Type
Cust_id(PRIMARY KEY)	Integer NOT NULL
Name	Varchar(50)
Contact	char(11)
Address	Varchar(50)

Table Name: Categories

Field	**Type**
Cat_id(PRIMARY KEY)	Integer NOT NULL
Cat_name	Varchar(50)

Cat_type	Varchar(11)

Table Name: Shopping_Order

Field	**Type**
Order_id (PRIMARY KEY)	Integer NOT NULL
Customer_id (FOREIGN KEY)	Integer NOT NULL
Date_of_order	DATE

Table Name: Delivery

Field	**Type**
delivery_id (PRIMARY KEY)	Integer NOT NULL
Cust_id (FOREIGN KEY)	Integer NOT NULL
Date_of_delivery	DATE

Table Name: Products

Field	**Type**
P_id (PRIMARY KEY)	Integer NOT NULL
Category_id (FOREIGN KEY)	Integer NOT NULL
P_name	Varchar(50)

Table Name: Seller

Field	**Type**
Seller_id (PRIMARY KEY)	Integer NOT NULL
product_id (FOREIGN KEY)	Integer NOT NULL
s_name	Varchar(50)

Table Name: Payment

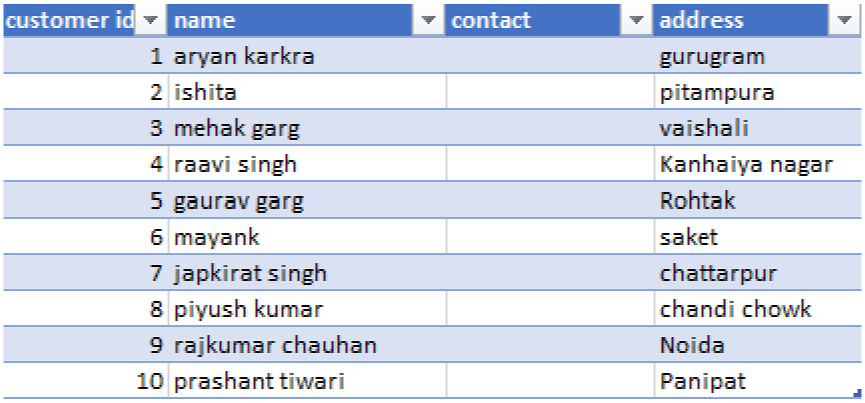
Field	**Type**
Pay_id (PRIMARY KEY)	Integer NOT NULL
Customer_id (FOREIGN KEY)	Integer NOT NULL
Date_of_payment	DATE

Table Name: Transaction_details

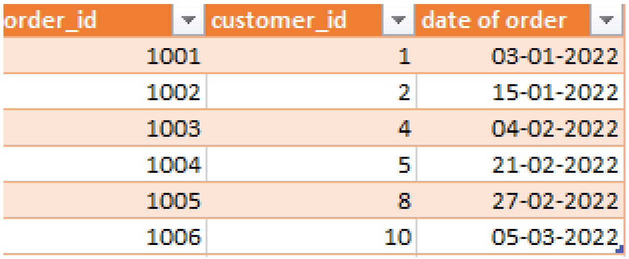
Field	**Type**
report_id (PRIMARY KEY)	Integer NOT NULL
customer_id (FOREIGN KEY)	Integer NOT NULL
order_id(FOREIGN KEY)	Integer NOT NULL

product_id(FOREIGN KEY)	Integer NOT NULL
payment_id(FOREIGN KEY)	Integer NOT NULL

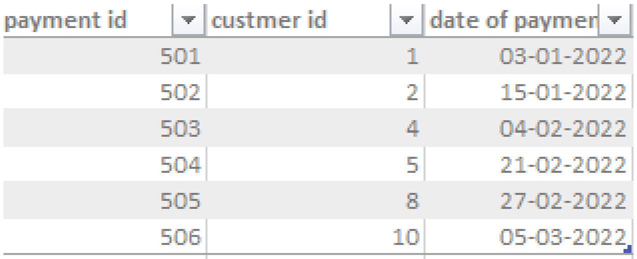
TABLES



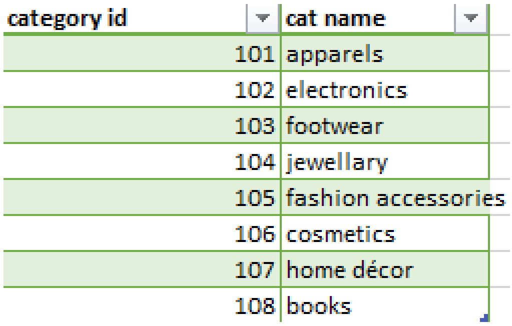
customer id	name	contact	address
1	aryan karkra		gurugram
2	ishita		pitampura
3	mehak garg		vaishali
4	raavi singh		Kanhaiya nagar
5	gaurav garg		Rohtak
6	mayank		saket
7	japkirat singh		chattarpur
8	piyush kumar		chandi chowk
9	rajkumar chauhan		Noida
10	prashant tiwari		Panipat



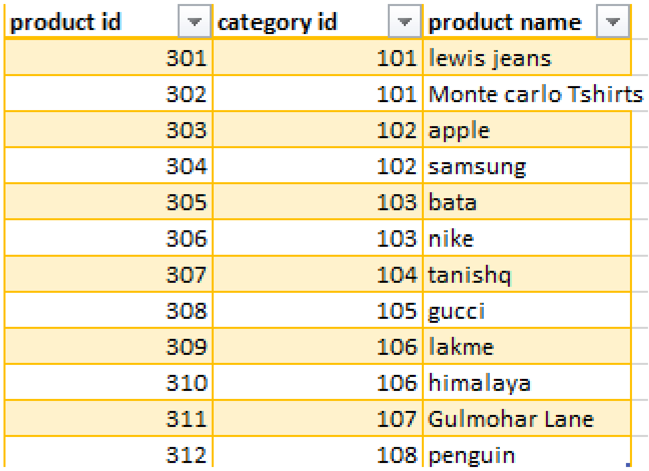
order_id	customer_id	date of order
1001	1	03-01-2022
1002	2	15-01-2022
1003	4	04-02-2022
1004	5	21-02-2022
1005	8	27-02-2022
1006	10	05-03-2022



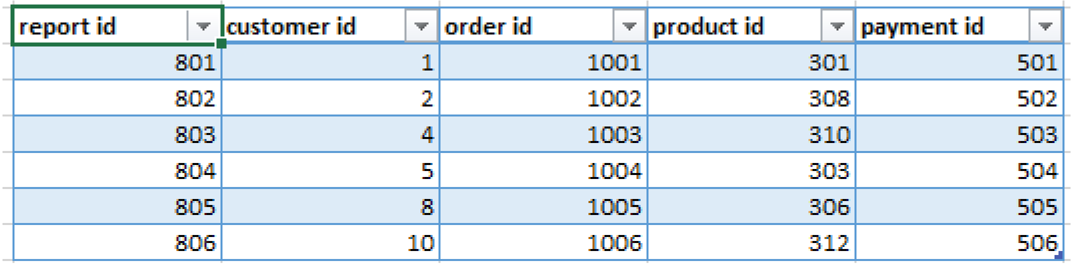
payment id	custmer id	date of paymen
501	1	03-01-2022
502	2	15-01-2022
503	4	04-02-2022
504	5	21-02-2022
505	8	27-02-2022
506	10	05-03-2022



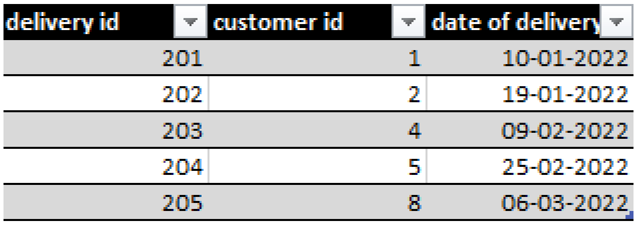
category id	cat name
101	apparels
102	electronics
103	footwear
104	jewellary
105	fashion accessories
106	cosmetics
107	home décor
108	books



product id	category id	product name
301	101	lewis jeans
302	101	Monte carlo Tshirts
303	102	apple
304	102	samsung
305	103	bata
306	103	nike
307	104	tanishq
308	105	gucci
309	106	lakme
310	106	himalaya
311	107	Gulmohar Lane
312	108	penguin



report id	customer id	order id	product id	payment id
801	1	1001	301	501
802	2	1002	308	502
803	4	1003	310	503
804	5	1004	303	504
805	8	1005	306	505
806	10	1006	312	506



delivery id	customer id	date of delivery
201	1	10-01-2022
202	2	19-01-2022
203	4	09-02-2022
204	5	25-02-2022
205	8	06-03-2022

CONVERSION

Strong entities are converted to individual tables with primary attributes as their primary key.

Customer Entity -> CUSTOMER TABLE with Primary Key Customer_id

Categories Entity->CATEGORIES TABLE with Primary Key Cat_id

Entities with 1:N cardinality Ratio are converted into a Table with the Attributes of an entity with cardinality 1 and the Primary key of other as the Foreign Key

Shopping Order TABLE

order_id (Primary Key)

Customer_id (Foreign Key from CUSTOMER TABLE)

Since there are no composite attributes or multi valued attributes we do not need to split further in multiple tables

QUERIES

```
select Cust_id,Name,Contact,Address from Customer,Payment where
date_of_payment>"2022-02-22" and Cust_id=customer_id;
```

```
select P_id,P_name,category_id from products,Categories where
Categories.cat_id=products.category_id and (cat_name="footwear" or
cat_name="jewellary");
```

```
select P_id,P_name,category_id from products,Categories as Ca,Transaction_details as tr
where tr.product_id=products.P_id and Ca.cat_id=products.category_id and
cat_name="footwear";
```

```
select Name,P_id,P_name,Date_of_delivery from
Customer,delivery,products,Transaction_details as tr where Date_of_delivery>"2022-02-09"
and delivery.Cust_id=customer_id and P_id=product_id and Customer.cust_id=customer_id;
```

```
select * from Customer where Cust_id NOT IN (select Customer_id from shopping_order );
```

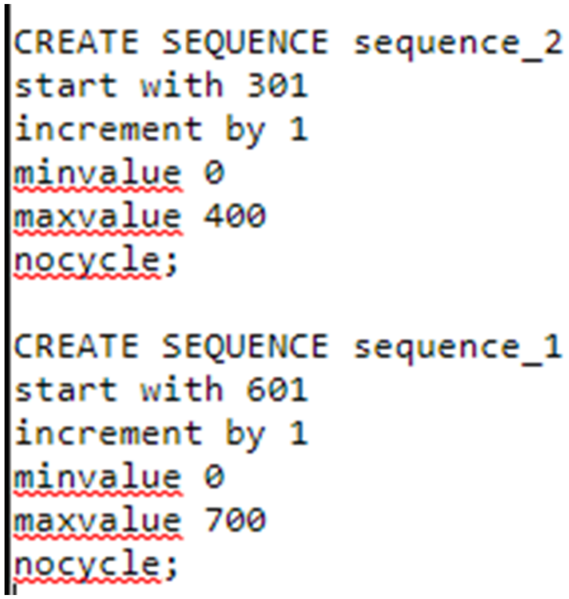
```
select category_id,count(*) from products group by category_id;
```

```
select P_id,P_name,shopping_order.Order_id,Name from
products,shopping_order,Customer,Transaction_details where Name="prashant tiwari" and
Transaction_details.order_id=shopping_order.Order_id and P_id=product_id and
Customer.Cust_id=shopping_order.customer_id;
```

SEQUENCES

•A sequence is a user-defined schema bound object that generates a sequence of numeric values.

•Sequence is a set of integers 1, 2, 3, … that are generated and supported by some database systems to produce unique values on demand.



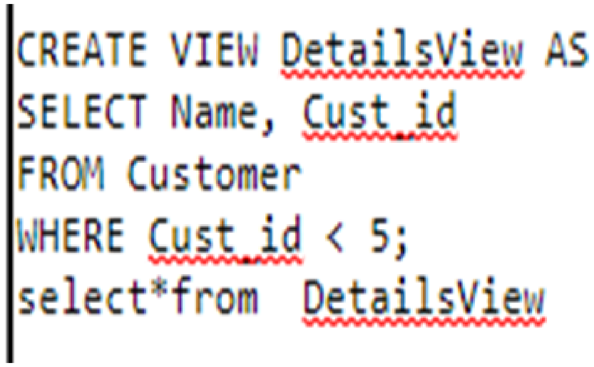
```
CREATE SEQUENCE sequence_2
start with 301
increment by 1
minvalue 0
maxvalue 400
nocycle;

CREATE SEQUENCE sequence_1
start with 601
increment by 1
minvalue 0
maxvalue 700
nocycle;
```

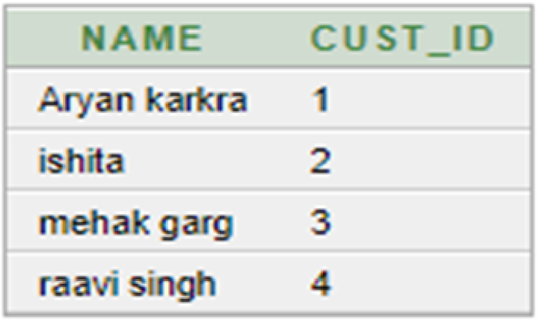
We have made use of sequences in the table sales_person to assign unique values to its primary key

VIEWS

Views in SQL is a kind of virtual table. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on certain conditions.



```
CREATE VIEW DetailsView AS
SELECT Name, Cust_id
FROM Customer
WHERE Cust_id < 5;
select*from  DetailsView
```



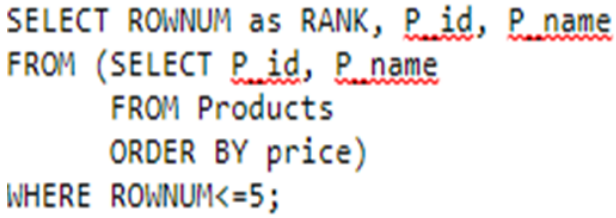
NAME	CUST_ID
Aryan karkra	1
ishita	2
mehak garg	3
raavi singh	4

Select top/SELECT INTO

•Top-N Analysis in SQL deals with How to limit the number of rows returned from ordered sets of data in SQL.

Top-N queries ask for the *n* smallest or largest values of a column. Both smallest and largest values sets are considered Top-N queries.

•Top-N analysis are useful in cases where the need is to display only the *n bottom-most* or the *n top-most* records from a table based on a condition. This result set can be used for further analysis.

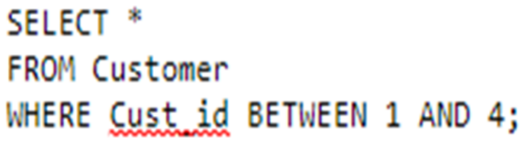


```
SELECT ROWNUM as RANK, P_id, P_name
FROM (SELECT P_id, P_name
      FROM Products
      ORDER BY price)
WHERE ROWNUM<=5;
```

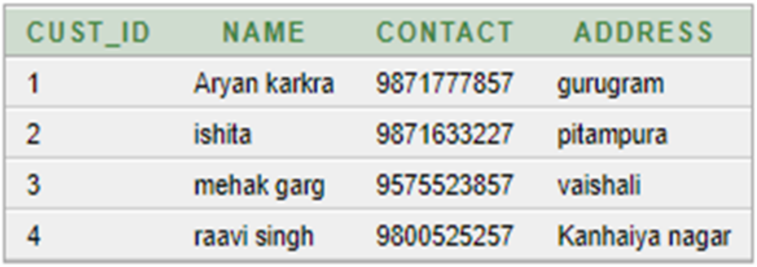
BETWEEN

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

The BETWEEN operator is inclusive: begin and end values are included.



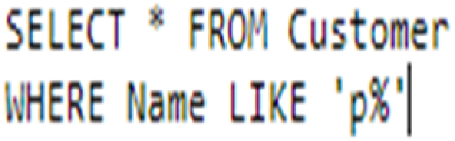
```
SELECT *
FROM Customer
WHERE Cust_id BETWEEN 1 AND 4;
```



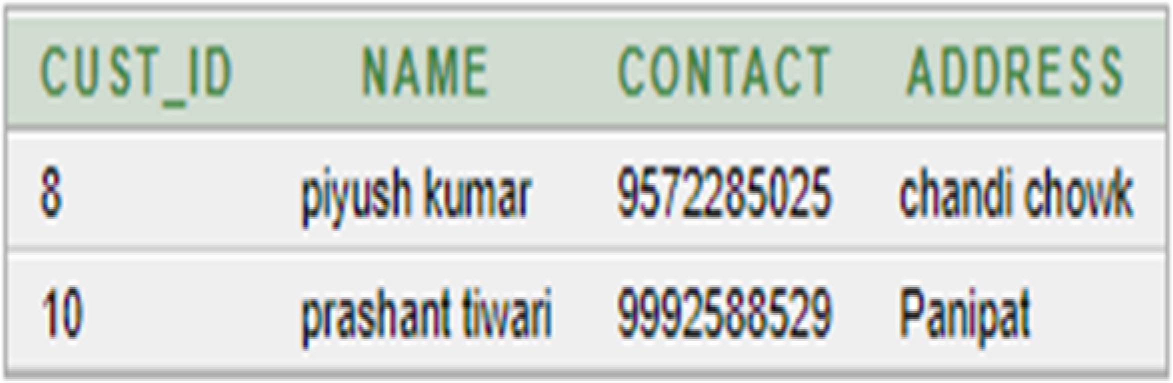
CUST_ID	NAME	CONTACT	ADDRESS
1	Aryan karkra	9871777857	gurugram
2	ishita	9871633227	pitampura
3	mehak garg	9575523857	vaishali
4	raavi singh	9800525257	Kanhaiya nagar

LIKE

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.



```
SELECT * FROM Customer
WHERE Name LIKE 'p%'
```



CUST_ID	NAME	CONTACT	ADDRESS
8	piyush kumar	9572285025	chandi chowk
10	prashant tiwari	9992588529	Panipat

DML

Data Manipulation Language (DML) commands in SQL deal with the manipulation of data records stored within the database tables. It does not deal with changes to database objects and their structure. The commonly known DML commands are INSERT, UPDATE and DELETE..

SQL JOINs

- `INNER JOIN`: Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN`: Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN`: Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN`: Returns all records when there is a match in either left or right table

GROUP BY and ORDER BY Clause in SQL

GROUP BY:- The GROUP BY clause is used to arrange identical data into groups. The GROUP BY clause is used with the SELECT statement. The GROUP BY clause is used with aggregate functions like COUNT, MAX, MIN, SUM, and AVG.

ORDER BY:-

The ORDER BY clause is used to sort the result-set in ascending or descending order.

The ORDER BY clause sorts the records in ascending order by default.

THANK YOU