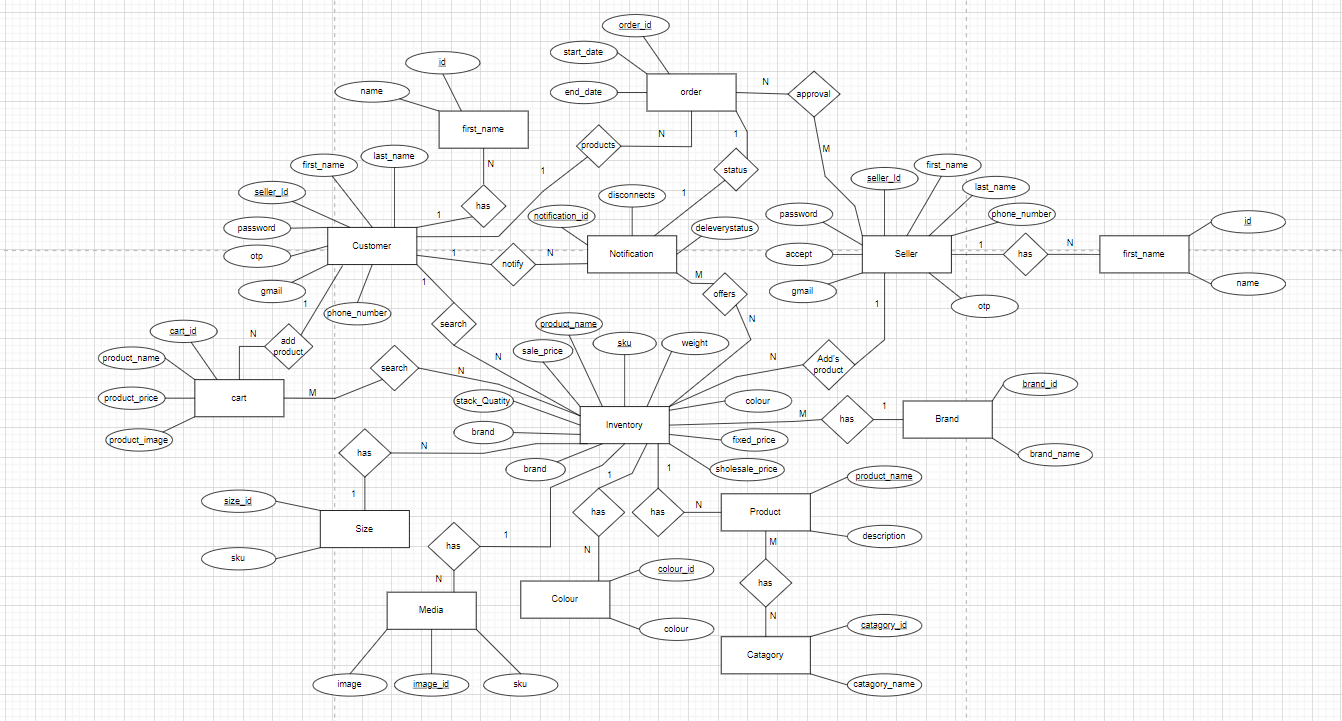
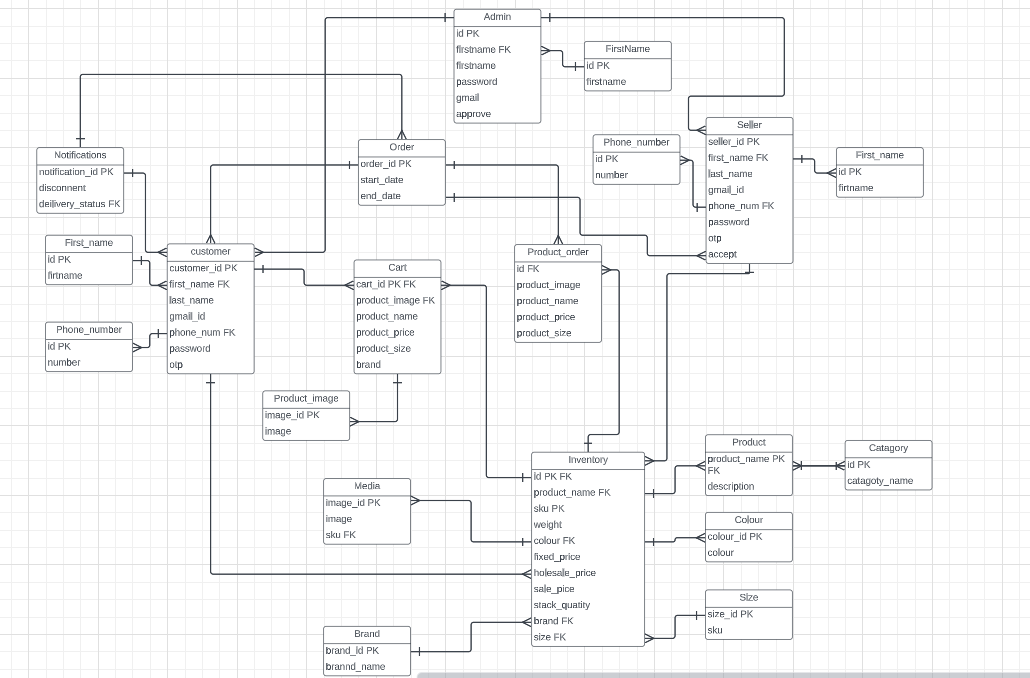
# **Database Design Assignment**

1) Physical Entity Relationship diagram of database.





2) Explain about searching performance. How will you handle replication in SQL for searching & Reporting?

* Searching performance in the context of SQL refers to how exactly and correctly the data is getting from the database, while customer is searching for a product from the web page how correctly the data is coming from such a large amount data in the database.
* Replication in SQL is copying data from one database to another, it is used in various purposes such as backup and load balancing.
* Indexing: Indexing is a crucial technique to optimize search performance. Create indexes on columns that are frequently used in search conditions. For text-based searches, consider using Full-Text Indexing capabilities provided by some DBMSs. Full-Text Indexing enables more efficient searching within large blocks of text.

3) Explain what major factors are taken into consideration for performance

* Load Balancing: In systems with multiple servers or nodes, load balancing ensures that the workload is distributed evenly, preventing individual components from being overwhelmed and improving overall performance.
* Database Indexing: Indexing plays a crucial for database, where it allows use to retrieve the data faster for the database. Well-designed and properly utilized indexes can significantly improve query performance.

4) Mention about Indexing, Normalization and Denormalization.

­­ Indexing: -

* SQL Index is a quick lookup table which is used to retrieve the data from the database very fast.
* It also retrieves a vast amount of data from the tables frequently.
* Index helps to speed up select queries but slows down insert and update queries.

Syntax: -

CREATE INDEX Index\_name ON Table\_name (column\_name)

The Index\_name is the name of the index that we want to specify.

The Table\_name is the name of the table that on which the index is to be created.

The Column\_name is used to specify on which column we need to specify the index.

Normalization: -

* The normalization is the process of organising the data in the database and it is a process of eliminate data redundancy and enhance data integrity in the table.
* Normalization is the process of divides larger table into the smaller tables and link them using relationships.

**Indexing**: Indexes are data structures that speed up data retrieval by allowing the database to quickly locate rows based on the values of certain columns. By creating indexes on columns frequently used in search conditions, the database can minimize the number of rows it needs to scan and thus improve search performance.

Steps to normalization: -

First Normalization Form(1Nf): -

* A relation is in first normal form if every attribute in that relation is singled valued attribute.

Second Normalization Form(2Nf): -

* It needs to be in 1NF.
* All non-key attributes are fully functional dependent on the primary key. It does not have any non-prime attribute that is functionally dependent on any proper subset of any candidate key.

Three Normalization Form(3NF): -

* It needs to be in 2NF.
* No transition dependency exists.

Boyce Codd Normal Form (BCNF): -

* It needs to be in 3NF.
* Every Right-Hand Side (RHS) attribute of the functional dependencies should depend on the super key of that table.

Denormalization: -

* The denormalization it is the opposite to the normalization, where we can add redundant data to one or more tables one attribute can store multiply data.
* This helps us avoid costly joins in a relational database.

5) How will you handle scaling, if required at any point of time.

* Handling the scaling effectively is crucial to ensure that a system can accommodate increased workload and growing user demands. There two ways to achieve the scaling horizontal and vertical scaling. I have used the horizontal scaling.

Horizontal Scaling: -

* Horizontal scaling it involves add multiply machines or nodes to distribute the workload, In this each machine work independently with their assigned work.
* The advantage of using horizontal scaling it can handle the large number of requests.
* By using the horizontal scaling improved the fault tolerance.
* As the workload grows, we can add the additional machines and increases the performance.

6) Mention all the assumptions you are taking for solutions.

* The data types for attributes, such as uniqueness, nullability and referential integrity (foreign key relationships), are enforced at the database level to maintain consistency and integrity.
* The ER diagram assumes that data integrity constraints, such as referential integrity (foreign key relationships), are enforced at the database level to maintain consistency and integrity.
* To reduce data redundancy, I have used normalization and split the larger table into small tables and the columns which are stores multiply values divided them into the separate table by using foreign key I am getting the product data.
* While ordering a product I have used two different tables in one table the product details will be saved and another table the order status will be stored like start date and end date because while ordering the data it may fail for some reasons which are exceptions for that the product should not get disturbed that's I was used two table.