DBMS PROJECT Final Evaluation

Original Scope of the project-
This is a project to make a buying and selling platform(like OLX) in which we have to design a database for the application. This application will contain the ER diagram, Database, Relational Schema and set of queries to retrieve the data from the database. Every project member contributed equally to each of these aspects. This project will help us better understand the topics of Database Management Systems and lay the foundation for it to become a fully functioning website where people can buy or sell items.
With this application for buying and selling, platforms can efficiently store and retrieve their data. We store the details of users, products, the area products are in, the price which the user asked for and the price which he actually got etc, can be done with the help of this application. The users can chat with other users to negotiate and share locations to deliver the items to each other. The application preserves the user's exact location, and the user can decide the location from where he wants to sell his product. This is also a great area to sell 2nd hand items as you get value for something that is not of use to you now.

Updated Scope of the project-
Our application is a database management system for our online buying and selling platform OLLX. Users will use it to locate the products around near their location. User's can chat with other users to bargain and sell their products. It is a hub for people looking to buy and sell their products. We can compare which users have bought items and for what amount, as well the ratio for people of the value of products that they have sold to the value they bought. People can see the users who are selling all kinds of products and in which location. We can compare the number of people of all age groups in our application. We can also see the number of text messages sent sorted and grouped based on their length. In future, our database can be modified to accommodate for delivery of items from one location to another so that people can buy items from places far away.

Stakeholders-
1. Users: A user of the application can be a buyer, a seller or both. A seller will use the database to register his product to sell it. A buyer will access the database to get relevant products that he might be interested in buying.
2. Companies: The companies who manufacture these products will be affected if their products' resale value is high or low. Depending on how much it gets sold

on our application. The companies will also access the user data to advertise
their products.

3. Transportation Companies: Many users might want to buy something, not near
them. So they can use some courier services to get their product.
4. Payment Gateways Companies: Users might use online modes of payment to
pay for the product they purchased.
5. Offline store owners: The stores such as pawn shops that buy and sell 2nd hand
goods will be affected as people might get much better prices for their products
online.

Entities-
1. Chats- The chats the users do to negotiate and buy items.
2. Users- People who use the app to buy or sell.
3. Products- The items that people buy or sell on the platform.
4. Bought items- The items that are sold through the application.
5. Types- Some generic items such as cars, electronics, education etc.
6. Areas- The locality where the items are.
7. Description- A description for each product describing the necessary features.

Relationships between the entities-

Entity1	Entity2	Relationship
Product	Description	Each product has a description through which the people can see the details of the attributes of the item.
User	Chats	Two users together can chat with each other to negotiate, get more details and discuss the transportation of an item.
User	BoughtItems	Once a user buys an item, the database keeps a record of what the user has bought and for how much.
User	Product	The user puts his item for sale as a product on the application so other users can see and buy it.

Product	BoughtItems	Once a user buys a product, the product id is stored along with the price details and the userId.
Product	Type	Each product falls in a category of items that helps sort the items.
Product	Area	Each product has an area where it is located, so the users can see if and how they want to get their item.

List of attributes of entities-
1. Chats- <u>chat_id</u>, sender ID, receiver ID, chat_message, date, time.
2. Users- <u>user_id</u>, phone, dob, age, name, email, password
3. Products- <u>product_id</u>, name, price, description, area
4. Bought items- <u>product_Id</u>, buyer ID, date, price of purchase
5. Types- <u>TypeID,</u> total items, name
6. Areas- <u>AreaID,</u> state, city, locality, Pincode, country, state
7. Description- <u>product_id</u>, short description, long description

Weak Entities-
Description and bought items. These items do not have a primary key of their own.
They reference their primary key as a foreign key from productID in products tables.

Entity Relationship participation and their types-

Entity1	Entity2	Description
Product (total)	Description (Partial)	(has) Product has total participation as each product has a description. Still, Description has partial participation as each description may not be of a product in the database.
User (Partial)	Chats (total)	(chats) Chats have total participation as for each chat; there are two users. Here the user has partial participation as any two users may not chat.
User (Partial)	BoughtItems (total)	(buys) Here bought items have total participation as every item that is sold as a user that buys it. But every user does not buy an item, hence the user has partial participation.

User (Partial)	Product (Total)	(sells)Here product has total participation as every product has a user that sells it. Still, the user has partial participation as not every user sells a product.
Product (Partial)	BoughtItems (Total)	(sold)Here product has partial participation as not every product is sold. Here bought items have total
		participation as for every sold item, there was a product.
Product (Total)	Type (partial)	(has)Here product has total participation as every product has a type, but the type has partial participation as every type may not have a product.
Product (Total)	Area (partial)	(has)Here product has total participation as every product has an area, but the area has partial participation as every area may not have a product.

Ternary Relationship-

The ternary relationship between the chat table and two users(sender and receiver). So two users together make an entry in the chat table. It is a relationship between three entities sender, receiver and chat table.

Views- Created Views using sql queries

Views are as follow:

('area_products', 'VIEW')

('category_sale', 'VIEW')

('pdt', 'VIEW')

('user_details1', 'VIEW')

Indexes:

Indexes for USerDetails:

('user_details', 0, 'PRIMARY', 1, 'user_id', 'A', 100, None, None, '', 'BTREE', '', '', 'YES', None)

('user_details', 1, 'usr_tbl', 1, 'age', 'A', 37, None, None, '', 'BTREE', '', '', 'YES', None)

('user_details', 1, 'usr_tbl', 2, 'customer_email', 'A', 100, None, None, '', 'BTREE', '', '', 'YES', None)

Indexes for sold product:

('sold_product', 0, 'PRIMARY', 1, 'bought_id', 'A', 20, None, None, '', 'BTREE', '', '', 'YES', None)

('sold_product', 1, 'product_id', 1, 'product_id', 'A', 20, None, None, '', 'BTREE', '', '', 'YES', None)

('sold_product', 1, 'sld_pdt', 1, 'buyer_id', 'A', 19, None, None, '', 'BTREE', '', '', 'YES', None)

('sold_product', 1, 'sld_pdt', 2, 'product_id', 'A', 20, None, None, '', 'BTREE', '', '', 'YES', None)

Indexes for product table:

('product_table', 0, 'PRIMARY', 1, 'product_id', 'A', 7703, None, None, '', 'BTREE', '', '', 'YES', None)

('product_table', 1, 'type_id', 1, 'type_id', 'A', 5, None, None, 'YES', 'BTREE', '', '', 'YES', None)

('product_table', 1, 'seller_id', 1, 'seller_id', 'A', 100, None, None, 'YES', 'BTREE', '', '', 'YES', None)

('product_table', 1, 'Prdt_Table', 1, 'city_code', 'A', 3632, None, None, 'YES', 'BTREE', '', '', 'YES', None)

Indexes for chat table:

('chat_table', 0, 'PRIMARY', 1, 'chat_id', 'A', 10, None, None, '', 'BTREE', '', '', 'YES', None)

('chat_table', 1, 'receiver_id', 1, 'receiver_id', 'A', 6, None, None, '', 'BTREE', '', '', 'YES', None)

('chat_table', 1, 'cht_tbl', 1, 'sender_id', 'A', 6, None, None, '', 'BTREE', '', '', 'YES', None)

('chat_table', 1, 'cht_tbl', 2, 'receiver_id', 'A', 8, None, None, '', 'BTREE', '', '', 'YES', None)

Indexes for area table:

('area_table', 0, 'PRIMARY', 1, 'area_id', 'A', 7274, None, None, '', 'BTREE', '', '', 'YES', None)

('area_table', 1, 'ar_tbl', 1, 'pincode', 'A', 4389, None, None, 'YES', 'BTREE', '', '', 'YES', None)

Indexes for type table:

('type_table', 0, 'PRIMARY', 1, 'type_id', 'A', 5, None, None, '', 'BTREE', '', '', 'YES', None)

('type_table', 1, 'typ_table', 1, 'total_items', 'A', 5, None, None, '', 'BTREE', '', '', 'YES', None)

Indexes for description table:

('description_table', 1, 'product_id', 1, 'product_id', 'A', 7900, None, None, '', 'BTREE', '', '', 'YES', None)

('description_table', 1, 'shrt_desp', 1, 'short_description', 'A', 4402, None, None, '', 'BTREE', '', '', 'YES', None)

Given Grants to different type users.