Sri Lanka Institute of Information Technology

BSc Honors in Information Technology Specializing in Cyber Security

Database Management Systems for Security-IE2042

June 2023

Group Assignment

Database Design, Implementation and Security

Members of Group

DISSANAYAKE.D.M.T.V	IT22245342
NAWARATHNA.N.P.D.T	IT22117496
PEIRIS.D.D.N.S	IT22062192
DE SILVA.K.R.K.D	IT22151056

Contents

1.Assumptions which we made

Member Relationship: A member has the option to buy and sell. For this, separate tables for buyers and sellers are used, and both tables have a foreign key connection to the member table. Members have the option of buying or selling.

Unique Identifiers: To maintain data integrity, each entity has a special identifier, such as a Member Number, Item Number, Category ID, Bid ID, and Transaction ID.

Data Validation: To make sure, for instance, that bid prices are within a given range, start dates and finish dates make sense, and references to existing entities are acceptable, the database system should incorporate data validation and constraints.

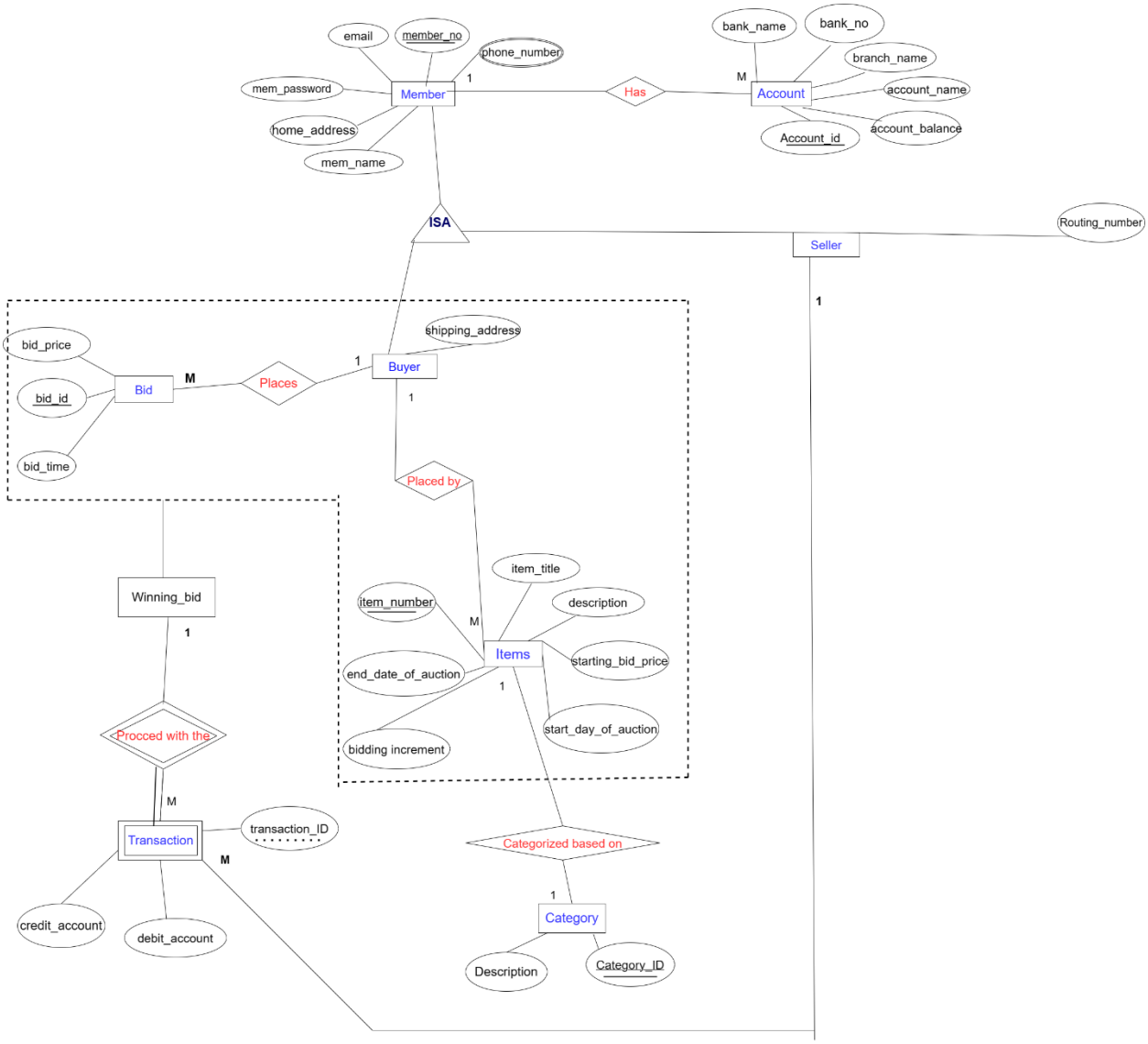
Security: Access to certain areas of the database should be controlled depending on user roles and permissions, and passwords should be securely saved using the necessary encryption mechanisms.

Auction Rules: The method assumes that the highest bidder will win the auction and that the winner must complete the transaction. This fundamental schema does not cover specific auction kinds like reserve price auctions or buy-it-now choices.

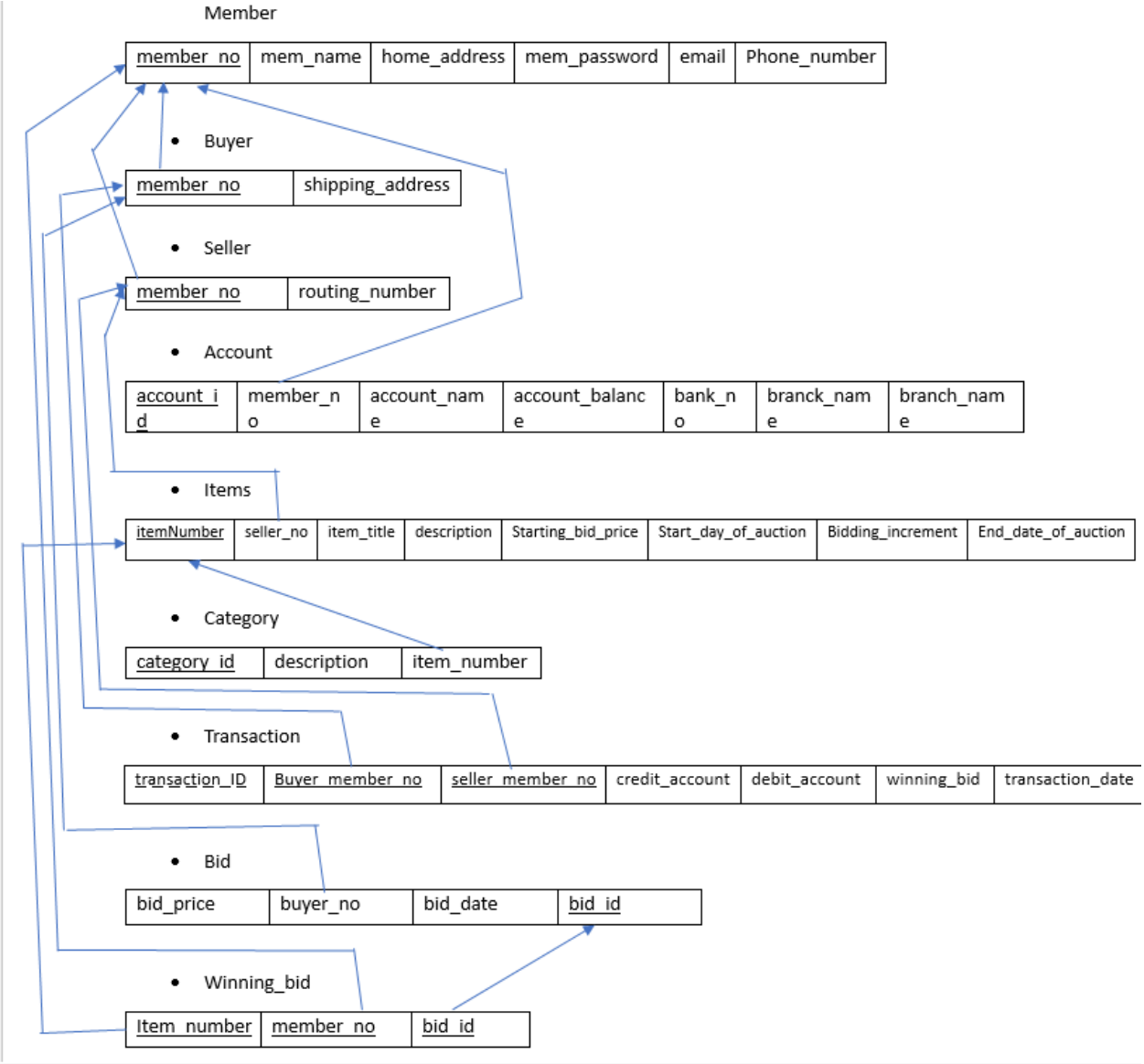
Payment Handling: The method assumes that the highest bidder will win the auction and that the winner must complete the transaction. This fundamental schema does not cover specific auction kinds like reserve price auctions or buy-it-now choices.

Shipping: Buyers can access shipping information, but the actual shipping procedure is not included in this schema.

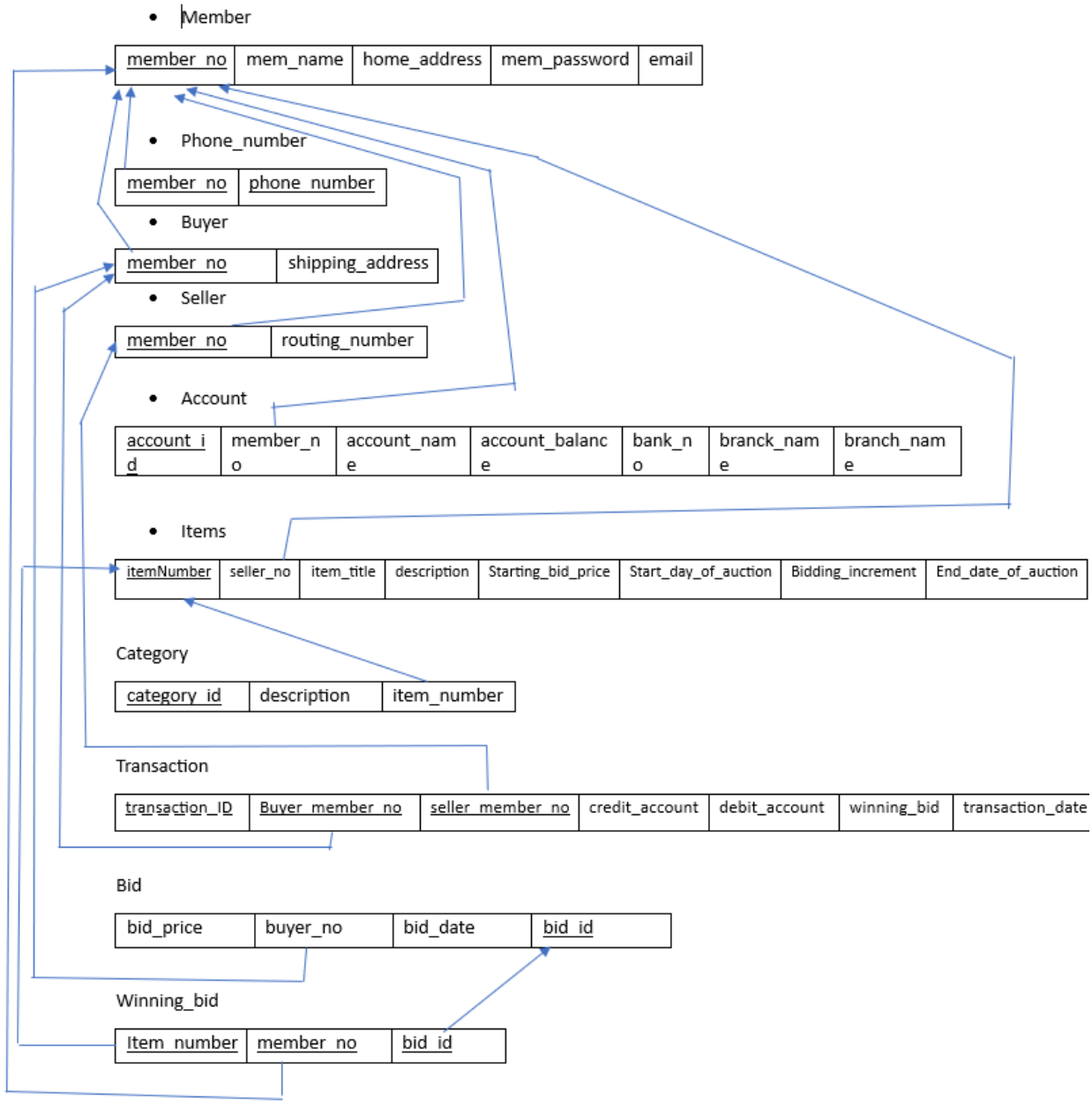
2.Develop the ERD and logical model



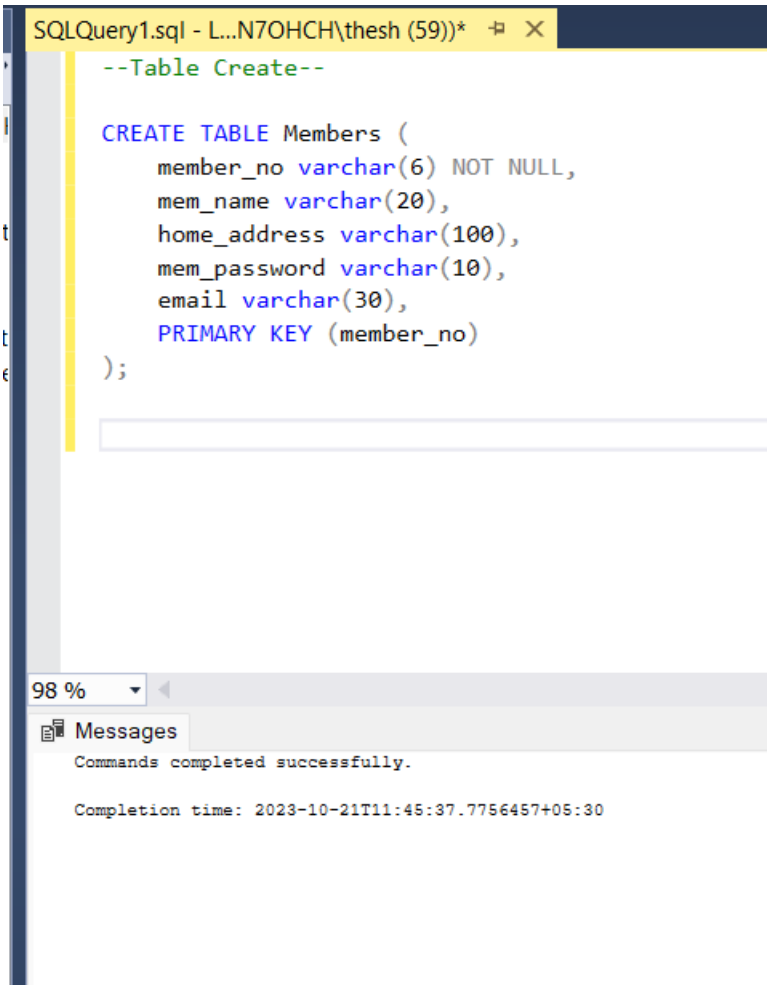
3.Unnormalized Relational Schema



4.Normalized Relational Schema



5.Logical model in MS SQL



```sql
CREATE TABLE Phone_number (
    member_no varchar(6),
    phone_no varchar (10),
    PRIMARY KEY (member_no,phone_no),
    FOREIGN KEY (member_no) REFERENCES Members(member_no)
);
```

98 %

Messages

```
Commands completed successfully.

Completion time: 2023-10-21T11:48:24.8905252+05:30
```

```sql
);

CREATE TABLE Buyer (
    member_no varchar(6),
    shipping_address varchar(100),
    PRIMARY KEY (member_no),
    FOREIGN KEY (member_no) REFERENCES Members(member_no)
);
```

98 %

Messages

```
Commands completed successfully.

Completion time: 2023-10-21T11:50:13.9348742+05:30
```

```sql
CREATE TABLE Seller (
    member_no varchar(6),
    routing_number varchar(20),
    PRIMARY KEY (member_no),
    FOREIGN KEY (member_no) REFERENCES Members(member_no)
);
```

98 %

Messages

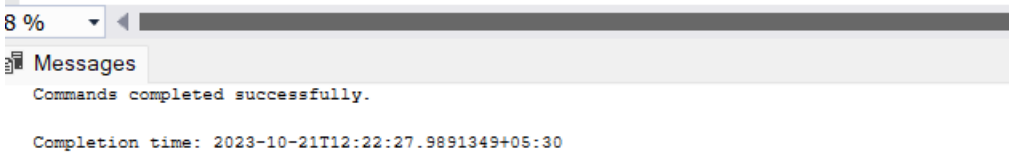
```
Commands completed successfully.

Completion time: 2023-10-21T11:51:35.5202933+05:30
```

```sql
CREATE TABLE Account (
    account_id int NOT NULL,
    member_no varchar(6),
    account_name varchar(40),
    account_balance int,
    bank_no int,
    bank_name varchar(30),
    branch_name varchar(30),
    PRIMARY KEY (account_id),
    FOREIGN KEY (member_no) REFERENCES Members(member_no)
);
```

```sql
CREATE TABLE Items (
    item_number int NOT NULL,
    seller_no varchar(6),
    item_title varchar(100),
    description varchar(500),
    starting_bid_price int,
    start_day_of_auction date,
    end_day_of_auction date,
    bidding_increment int,
    PRIMARY KEY (item_number),
    FOREIGN KEY (seller_no) REFERENCES Members(member_no)
);
```

```sql
CREATE TABLE Category (
    category_id varchar(6) NOT NULL,
    description varchar(500),
    PRIMARY KEY (category_id),
    CONSTRAINT CHK_Category CHECK (category_id LIKE 'c%')
);
```



SQLQuery1[1].sql...N7OHCH\thesh (65))

```sql
CREATE TABLE Transactions (
    transaction_id varchar(6) NOT NULL,
    buyer_member_no varchar(6),
    seller_member_no varchar(6),
    credit_account varchar(20),
    debit_account varchar(20),
    transaction_date date,
    winning_bid int,
    PRIMARY KEY (buyer_member_no, seller_member_no, transaction_id),
    FOREIGN KEY (buyer_member_no) REFERENCES Buyer(member_no),
    FOREIGN KEY (seller_member_no) REFERENCES Seller(member_no)
);
```

98 %

Messages

```
Commands completed successfully.

Completion time: 2023-10-21T12:24:26.3991325+05:30
```

```sql
CREATE TABLE Bid1 (
    buyer_no varchar(6),
    bid_id varchar(6) NOT NULL,
    bid_price int,
    bid_date date,
    PRIMARY KEY (bid_id),
    FOREIGN KEY (buyer_no) REFERENCES Buyer(member_no)
);

CREATE TABLE Winning_bid (
```

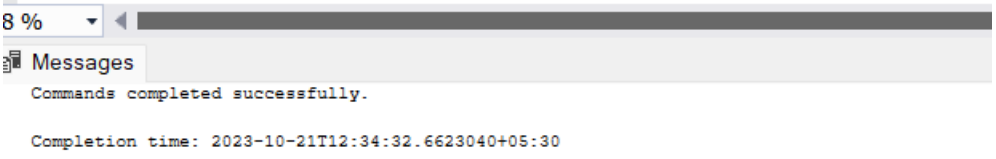
98 %

Messages

```
Commands completed successfully.

Completion time: 2023-10-21T12:25:25.8683568+05:30
```

```
CREATE TABLE Winning_bid (
    bid_id varchar(6),
    item_number int,
    member_no varchar(6),
    PRIMARY KEY (bid_id,item_number,member_no),
    FOREIGN KEY (bid_id) REFERENCES Bid1(bid_id),
    FOREIGN KEY (item_number) REFERENCES Items(item_number),
    FOREIGN KEY (member_no) REFERENCES Members(member_no)
);drop table Winning_bid
```



6.Enter Sample data

SQLQuery1[1].sql...N7OHCH\thesh (65))

```
        --Insert Data--

--Member Table--
INSERT INTO Members VALUES ('M001', 'Aravinda', 'Colombo1', 'secret', 'aravinda@gmail.com');
INSERT INTO Members VALUES ('M002', 'Saman', 'Gampaha', 'p@ssword', 'saman.1@gmail.com');
INSERT INTO Members VALUES ('M003', 'Piumi', 'Galle', 'system1', 'piumi.23@gmail.com');
INSERT INTO Members VALUES ('M004', 'kaveesha', 'Kandy', 'cat902', 'kavee.kandy@gmail.com');
INSERT INTO Members VALUES ('M005', 'Kavindu', 'Nugegoda', 'secure45', 'kavindu222@gmail.com');


--Phone_number--
```

98 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-10-21T12:46:43.0536369+05:30
```

```
--Phone_number--
INSERT INTO Phone_number VALUES ('M001', '0332123456');
INSERT INTO Phone_number VALUES ('M002', '0112156456');
INSERT INTO Phone_number VALUES ('M003', '0112009423');
INSERT INTO Phone_number VALUES ('M004', '0112703312');
INSERT INTO Phone_number VALUES ('M005', '0111029507');
```

98 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-10-21T12:47:40.4207575+05:30
```

```sql
--Buyer--
INSERT INTO Buyer VALUES ('M001', 'Kaluthara');
INSERT INTO Buyer VALUES ('M002', 'Gampaha');
```

98 %

Messages

```
(1 row affected)

(1 row affected)

Completion time: 2023-10-21T12:50:13.5662362+05:30
```

```sql
--Seller--
INSERT INTO Seller VALUES ('M003', '7632');
INSERT INTO Seller VALUES ('M004', '4302');
INSERT INTO Seller VALUES ('M005', '7681');
```

8 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-10-21T12:50:56.8240631+05:30
```

```sql
--Seller--
INSERT INTO Seller VALUES ('M003', '7632');
INSERT INTO Seller VALUES ('M004', '4302');
INSERT INTO Seller VALUES ('M005', '7681');
```

8 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-10-21T12:50:56.8240631+05:30
```

```sql
--Category--
INSERT INTO Category VALUES ('C001', 'Antique');
INSERT INTO Category VALUES ('C002', 'Instrument');
INSERT INTO Category VALUES ('C003', 'Art');
```

98 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-10-21T12:52:50.8273555+05:30
```

```sql
--Transactions--
INSERT INTO Transactions VALUES ('T001', 'M001', 'M003', 'CR001', 'D002', '2023-10-16', 5000);
INSERT INTO Transactions VALUES ('T002', 'M001', 'M004', 'CR002', 'D003', '2023-10-14', 25000);
INSERT INTO Transactions VALUES ('T003', 'M002', 'M005', 'CR003', 'D004', '2023-10-19', 34000);
```

98 %

Messages

```
(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2023-10-21T12:53:47.2308265+05:30
```

```sql
--Bid1--
INSERT INTO Bid1 VALUES ('M001','B001', 3000, '2023-10-20');
INSERT INTO Bid1 VALUES ('M002','B002', 5500, '2023-10-18');

--Winning bid--
```

98 %

Messages

```
(1 row affected)

(1 row affected)

Completion time: 2023-10-21T12:55:32.7847871+05:30
```

```sql
--Winning bid--
INSERT INTO Winning_bid VALUES ('B001', 10, 'M001');
INSERT INTO Winning_bid VALUES ('B002', 11, 'M002');
```

7.SQL Code

```
--Table Create--

CREATE TABLE Members (
	member_no varchar(6) NOT NULL,
	mem_name varchar(20),
	home_address varchar(100),
	mem_password varchar(10),
	email varchar(30),
	PRIMARY KEY (member_no)
);

CREATE TABLE Phone_number (
	member_no varchar(6),
	phone_no varchar (10),
	PRIMARY KEY (member_no,phone_no),
	FOREIGN KEY (member_no) REFERENCES Members(member_no)
);

CREATE TABLE Buyer (
	member_no varchar(6),
	shipping_address varchar(100),
	PRIMARY KEY (member_no),
	FOREIGN KEY (member_no) REFERENCES Members(member_no)
);

CREATE TABLE Seller (
	member_no varchar(6),
	routing_number varchar(20),
	PRIMARY KEY (member_no),
	FOREIGN KEY (member_no) REFERENCES Members(member_no)
);

CREATE TABLE Account (
	account_id int NOT NULL,
	member_no varchar(6),
	account_name varchar(40),
	account_balance int,
	bank_no int,
	bank_name varchar(30),
	branch_name varchar(30),
	PRIMARY KEY (account_id),
	FOREIGN KEY (member_no) REFERENCES Members(member_no)
);

CREATE TABLE Items (
	item_number int NOT NULL,
	seller_no varchar(6),
	item_title varchar(100),
	description varchar(500),
	starting_bid_price int,
	start_day_of_auction date,
	end_day_of_auction date,
	bidding_increment int,
	PRIMARY KEY (item_number),
	FOREIGN KEY (seller_no) REFERENCES Members(member_no)
```

```
);

CREATE TABLE Category (
	category_id varchar(6) NOT NULL,
	description varchar(500),
	PRIMARY KEY (category_id),
	CONSTRAINT CHK_Category CHECK (category_id LIKE 'c%')
);

CREATE TABLE Transactions (
	transaction_id varchar(6) NOT NULL,
	buyer_member_no varchar(6),
	seller_member_no varchar(6),
	credit_account varchar(20),
	debit_account varchar(20),
	transaction_date date,
	winning_bid int,
	PRIMARY KEY (buyer_member_no, seller_member_no, transaction_id),
	FOREIGN KEY (buyer_member_no) REFERENCES Buyer(member_no),
	FOREIGN KEY (seller_member_no) REFERENCES Seller(member_no)
);

CREATE TABLE Bid1 (
	buyer_no varchar(6),
	bid_id varchar(6) NOT NULL,
	bid_price int,
	bid_date date,
	PRIMARY KEY (bid_id),
	FOREIGN KEY (buyer_no) REFERENCES Buyer(member_no)
);

CREATE TABLE Winning_bid (
	bid_id varchar(6),
	item_number int,
	member_no varchar(6),
	PRIMARY KEY (bid_id,item_number,member_no),
	FOREIGN KEY (bid_id) REFERENCES Bid1(bid_id),
	FOREIGN KEY (item_number) REFERENCES Items(item_number),
	FOREIGN KEY (member_no) REFERENCES Members(member_no)
);drop table Winning_bid
```

8.Data insert

```
 --Insert Data--

--Member Table--
INSERT INTO Members VALUES ('M001', 'Aravinda', 'Colombo1', 'secret',
'aravinda@gmail.com');
INSERT INTO Members VALUES ('M002', 'Saman', 'Gampaha', 'p@ssword', 'saman.1@gmail.com');
INSERT INTO Members VALUES ('M003', 'Piumi', 'Galle', 'system1', 'piumi.23@gmail.com');
INSERT INTO Members VALUES ('M004', 'kaveesha', 'Kandy', 'cat902',
'kavee.kandy@gmail.com');
INSERT INTO Members VALUES ('M005', 'Kavindu', 'Nugegoda', 'secure45',
'kavindu222@gmail.com');


--Phone number--
INSERT INTO Phone_number VALUES ('M001', '0332123456');
INSERT INTO Phone_number VALUES ('M002', '0112156456');
INSERT INTO Phone_number VALUES ('M003', '0112009423');
INSERT INTO Phone_number VALUES ('M004', '0112703312');
INSERT INTO Phone_number VALUES ('M005', '0111029507');

--Buyer--
INSERT INTO Buyer VALUES ('M001', 'Kaluthara');
INSERT INTO Buyer VALUES ('M002', 'Gampaha');

--Seller--
INSERT INTO Seller VALUES ('M003', '7632');
INSERT INTO Seller VALUES ('M004', '4302');
INSERT INTO Seller VALUES ('M005', '7681');

--Account--
INSERT INTO Account VALUES (1, 'M001', 'Savings', 10000, 123, 'Sampath Bank', 'Colombo');
INSERT INTO Account VALUES (2, 'M002', 'Saving', 25000, 234, 'BOC Bank', 'Gampaha');
INSERT INTO Account VALUES (3, 'M003', 'checking', 12300, 876, 'HNB Bank', 'Galle');
INSERT INTO Account VALUES (4, 'M004', 'Savings', 19200, 655, 'BOC Bank', 'Kandy');
INSERT INTO Account VALUES (5, 'M005', 'Checking', 21100, 780, 'HNB Bank', 'Nugegoda');
--Items--
INSERT INTO Items VALUES (10, 'M003', 'Antique Watch', 'Rare vintage watch', 3500, '2023-
10-15', '2023-10-20', 25);
INSERT INTO Items VALUES (11, 'M004', 'Vintage Guitar', 'Collectible vintage electric
guitar with case.', 15000, '2023-10-16', '2023-10-25', 100);
INSERT INTO Items VALUES (12, 'M005', 'Oil Painting', 'Original oil painting by a
renowned artist', 22000, '2023-10-20', '2023-10-30', 500);

--Category--
INSERT INTO Category VALUES ('C001', 'Antique');
INSERT INTO Category VALUES ('C002', 'Instrument');
INSERT INTO Category VALUES ('C003', 'Art');

--Transactions--
INSERT INTO Transactions VALUES ('T001', 'M001', 'M003', 'CR001', 'D002', '2023-10-16',
5000);
INSERT INTO Transactions VALUES ('T002', 'M001', 'M004', 'CR002', 'D003', '2023-10-14',
25000);
```

```
INSERT INTO Transactions VALUES ('T003', 'M002', 'M005', 'CR003', 'D004', '2023-10-19',
34000);


--Bid1--
INSERT INTO Bid1 VALUES ('M001','B001', 3000, '2023-10-20');
INSERT INTO Bid1 VALUES ('M002','B002', 5500, '2023-10-18');

--Winning bid--
INSERT INTO Winning_bid VALUES ('B001', 10, 'M001');
INSERT INTO Winning_bid VALUES ('B002', 11, 'M002');
```

9.Two triggers that can be applied on the database

01)

```
CREATE TRIGGER check_transaction_date
BEFORE INSERT ON Transactions
FOR EACH ROW
BEGIN
    IF transaction_date > CURDATE() THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Transaction date cannot be be in the future';
    END IF;
END;
```

If the user inputs a future date into the “transaction_date” field, this trigger shows an error message using the SQLSTATE code ‘45000’.

02)

```
CREATE TRIGGER prevent_member_deletion
BEFORE DELETE ON Member
FOR EACH ROW
BEGIN
    DECLARE transaction_count INT;
    SELECT COUNT(*) INTO transaction_count FROM Transaction T JOIN Member M WHERE
T.buyer_member_no=M.member_no OR T.seller_member_no=M.member_no;
    IF transaction_count > 0 THEN
        SIGNAL SQLSTATE '10000'
        SET Message_text = ' cannot delete a member with associated transactions ';
    END IF;
END;
```

This trigger is to display an error message in an attempt to delete a member if the member is associated with a transaction.

10.Views

Two possible users of the database;

01)Buyer

```
CREATE VIEW Buyers_list AS
    SELECT
        M.member_no AS BuyerID,
        M.mem_name AS BuyerName,
        M.email AS BuyerEmail
    FROM
        Member M
        INNER JOIN Buyer AS B ON B.member_no = M.member_no
```

02)Seller

```
CREATE VIEW Sellers_list AS
    SELECT
        M.member_no AS SellerID,
        M.mem_name AS SellerName,
        M.email AS SellerEmail
    FROM
        Member M
        INNER JOIN Seller AS S ON S.member_no = M.member_no
```

11.Indexes

Among the tables “Account” and the “Member” are the mostly used tables.

01)Index for Account

```
CREATE INDEX Account_ID ON Account (account_id);
```

02)Index for Member

```
CREATE INDEX member_no ON Member (member_no);
```

12.Stored Procedures

01)

```
CREATE PROCEDURE RetrieveSampathBankMembers()
BEGIN
    SELECT mem_name, home_address
    FROM Members M JOIN Account A ON M.member_no=A.member_no
    WHERE A.bank_name LIKE "%sampath bank%";
END
```

02)

```
CREATE PROCEDURE SearchForBiddersPlacedLaptops()
BEGIN
    SELECT M.member_no, M.email,B.bid_price ASCE
    FROM Member M JOIN Bid B ON M.member_no=B.buyer_no
    WHERE B.itemNumber=(SELECT itemNumber
                        FROM Items
                        WHERE description LIKE"%laptop%";)
    ORDER BY B.bid_price ASC;
END
```

03)

```
CREATE PROCEDURE SellersWithBidPriceGreaterThan30000()
BEGIN
    SELECT member_no,mem_name
    FROM Member M JOIN Items AS I ON M.member_no=I.seller_no
    GROUP BY M.mem_name
    HAVING SUM(I.starting_bid_price) > 30000;
END
```

04)

```
CREATE PROCEDURE IncrementSaman()
BEGIN
  UPDATE bid
  SET bid_price = bid_price * 1.15
  WHERE buyer_no = (SELECT member_no
                    FROM Member
                    WHERE mem_name LIKE "%saman%";)
END
```

Part 2

13.SQL injection

In cybersecurity field SQL injection is the one type of cyber-attack. An online security flaw known as SQL injection (SQLi) enables an attacker to tamper with database queries that an application makes. An attacker may be able to examine data this way that they would not typically be able to. Other users' data as well as any other data the application has access to may fall under this category. In many instances, an attacker can update or remove this data, permanently altering the application's behavior or content.

SQL injection is not only quite common, but it is also extremely dangerous since it can lead to unwanted access to private information, financial information, intellectual property, and trade secrets. It is rated as the top danger on the OWASP top 10 list of web application security threats. Numerous data breaches were caused by SQL injection attacks.

How SQL injection impact to the data base:

A successful SQL injection attack can cause to data breaches, data loss, or unauthorized attack to system access. Over the years, numerous high-profile data breaches have employed SQL injection techniques. Regulatory fines and reputational harm resulted from them. In some circumstances, an attacker can gain access to a persistent backdoor,

which can result in a long-term breach that can go undetected for a long time.

How to mitigate SQL injection:

Use Parameterized Queries: These are pre-built queries that the developer fills in with user data when the query is run. It is hard for a hacker to modify that function because it informs the database in advance of what the query is intended to achieve.

Consistent Security Audits: To find and fix any system vulnerabilities, conduct routine security audits and penetration tests.

The principle of least privilege: States that database accounts should only have the privileges required for system operation. This lessens the potential harm that a successful injection assault could do.

Validate inputs: Verify and clean up user input to make sure it follows the desired formats and is free of any harmful stuff.

14.Authentication and Authorization Issues

Authentication: confirming a user's identity before allowing them access to a particular system, network, or account.

Authorization: A request for access to a particular set of authentication data is authorized or rejected.

This may lead to unauthorized users getting access to personal information, bids, and financial data in an online auction system.

How Authentication and Authorization Issues impact to the data base:

Injection attacks, if successful, may result in data breaches, data loss, or unapproved access to the system. Additionally, they might lead to data manipulation during auctions, which would cost both buyers and sellers money.

How to mitigate Authentication and Authorization Issues:

1.Multi factor Authentication:

The authentication procedure can be made much more secure by implementing MFA. Before giving access, this entails asking users for two or more kinds of identity. This often entails fusing information about the user's knowledge (such as a password), possessions (such as a security token or smartphone), and potential identity (such as biometric data).

2.Secure password policies:

Strong password policies must be followed. Users must be required to create passwords that are sufficiently complex and are updated on a regular basis. The risk of unauthorized access can be reduced by implementing password managers and establishing rules for password development.

3.Encryption:

To protect sensitive data during transmission and storage, use encryption techniques. Even if data is intercepted during transport or storage, using robust encryption techniques helps prevent unauthorized access. For sensitive information, such as user credentials, financial information, and personal information, this is especially important.

4.Session management

Use effective session management techniques to guarantee that user sessions are safe and well-managed. The use of secure session tokens, setting session timeouts, and routinely terminating idle sessions are all

examples of this. It assists in preventing session hijacking or fixation attacks, which allow unwanted access to sensitive data.

5.Security training and awareness:

Inform users and employees about optimum security procedures, such as the value of secure passwords, spotting phishing scams, and observing security rules. By doing so, social engineering attacks can be avoided, and the organization's general security awareness can be raised. [1] [2]

13.Refferencess

[1] j. c. martienz, "autho by okta," auth0, 28 4 2023. [Online]. Available: https://auth0.com/blog/five-common-authentication-and-authorization-mistakes-to-avoid-in-your-saas-application/. [Accessed 11 10 2023].

[2] c. kime, "esecurity planet," esecurityplanet, 16 5 2023. [Online]. Available: https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks/. [Accessed 10 10 2023].