UTM
UNIVERSITI TEKNOLOGI MALAYSIA

School of Computing
Faculty of Engineering

SECD2523 – DATABASE

SEMESTER 1 2023/2024

P3 - Database Logical Design

NexScholar Event System

Group Name: ***PCK***

Members:

1. **Omar Mohammed Rezk Ebid Elashry – A22EC0026**
2. **Hasan Hatem Hasan – A22EC0013**
3. **Ahmed Amir – A22EC0007**
4. **THANES A/L SIVAM – A22EC0284**
5. **Zheng Shuoyu – A22EC4019**

CONTENTS

PROPOSED SYSTEM

1. Introduction

Getting into Database Conceptual Design is like laying the first bricks for a strong data system. It's about making a basic plan that grabs the important stuff like what things we're dealing with (entities), how they relate, and what details we care about (attributes). It's not diving into techy details but teaming up with others to create a simple map of how our data world should look. A good start here sets the tone for the whole database adventure, making sure our system fits just right with what our organization needs.

2. Overview of project

After completing Phase 2, we moved on to Phase 3 to implement database logic design and generate SQL statements (DDL and DML) for our Nexsolar project. In this article, we will transform the conceptual ERD of stage 3 into a logical ERD and generate relational patterns. Standardization will also be carried out to minimize duplicate dates and enhance the logical storage of data. Afterwards, we will update the Nexscholer project with reference to normalized relationships. Finally, the database of the website will be created using SQL statements.

3. Database conceptual design

3.1 Current business rule:

1. Enhancement of User Interface:

- Present Situation: The interface of the NexScholar platform is not user-friendly, which might have an impact on user satisfaction.

- The proposed Change: is to create a user interface that is both visually appealing and simple to use, improving the user experience overall and making it simple for users to interact with the platform's features and navigate.

2. Communication:

- Present Situation: Event coordinators relay information about the event to administrators, which could lead to a confusing and extended process.

- Proposed Change: Establish a direct line of communication via emails or a website chat system between registered users and event organizers in the website itself.

3. Integration of Payments:

- Present Situation: Participation in fee-based events may be limited if there is no payment option available.

- The proposed change: would be to incorporate a safe payment gateway (such as PayPal, credit cards, TNG, or GrabPay) so that people can register for events and make payments through the NexScholar website.

4. Reservation System:

- Present Situation: There isn't a straightforward way for users to book events right now.

- Proposed Change: Create a reservation system that speeds up the registration process by allowing users to sign up right on the website.

5. Fields for Event Information:

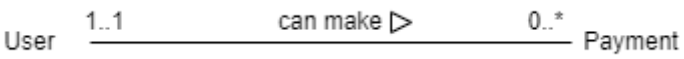
- Present Situation: It could be difficult for event planners to provide complete event details.

- Proposed Change: Provide an easy-to-use form that allows event coordinators to input necessary details including the event's name, description, date, time, location, organizers, website, and budget.

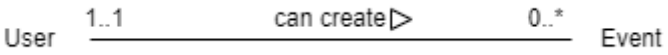
6. Notification from the Administrator:

- Present Situation: New event submissions for evaluation might not reach administrators right away.

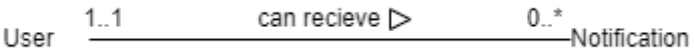
- Proposed Change: Implement an admin notification system that notifies administrators of every event request ensuring quick approval and review processes.



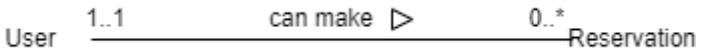
- Each user can make zero or many payments.
  - Each payment can be made by 1 User.



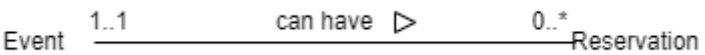
- Each user can create zero or many Events.
  - Each event can be created by 1 User.



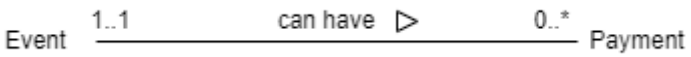
- Each user can receive zero or many Notifications.
  - Each Notification can be received by 1 User.



- Each user can make zero or many Reservations.
  - Each Reservation can be made by 1 User.

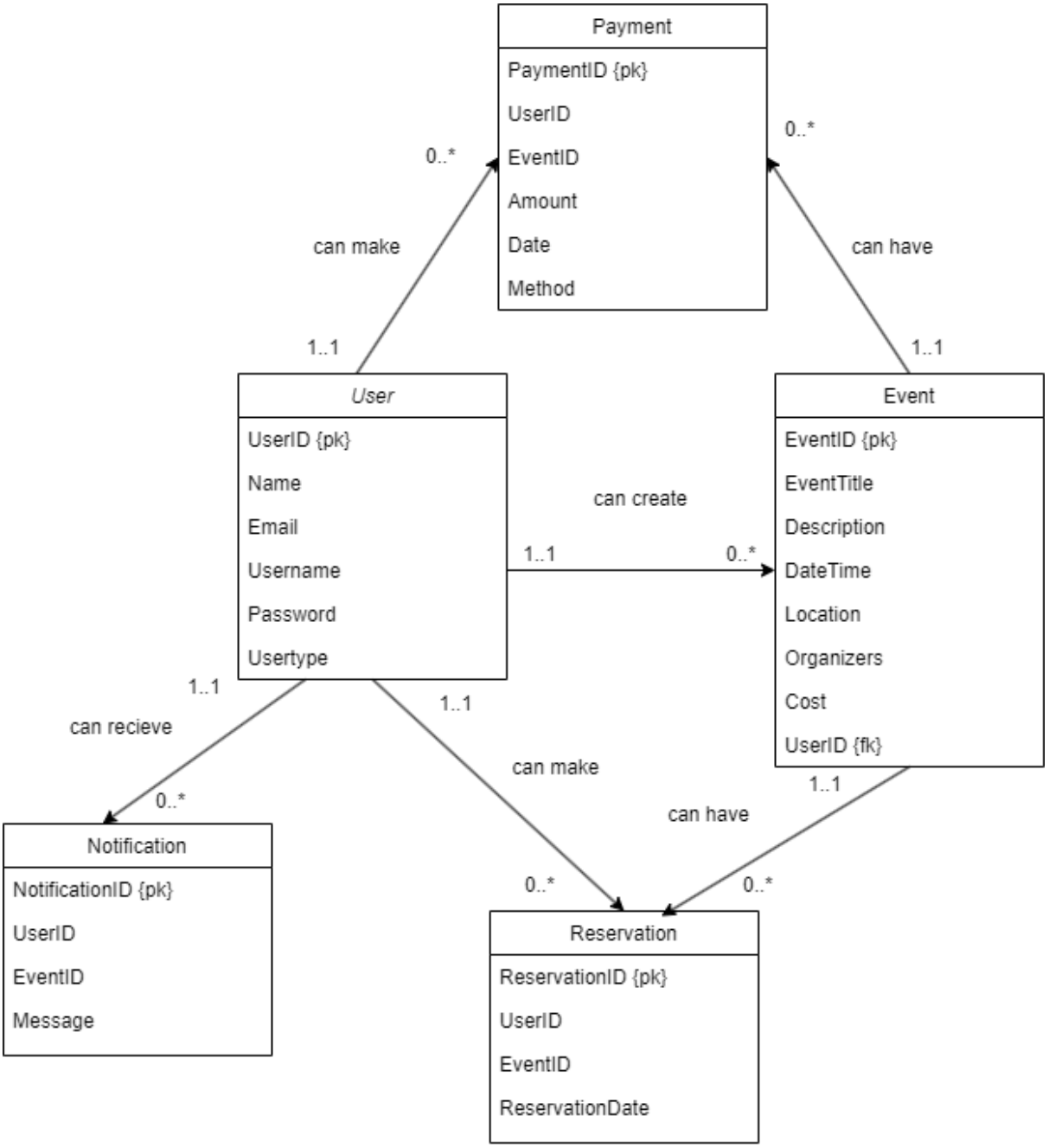


- Each Event can have zero or many Reservations.
  - Each reservation can have 1 Event.

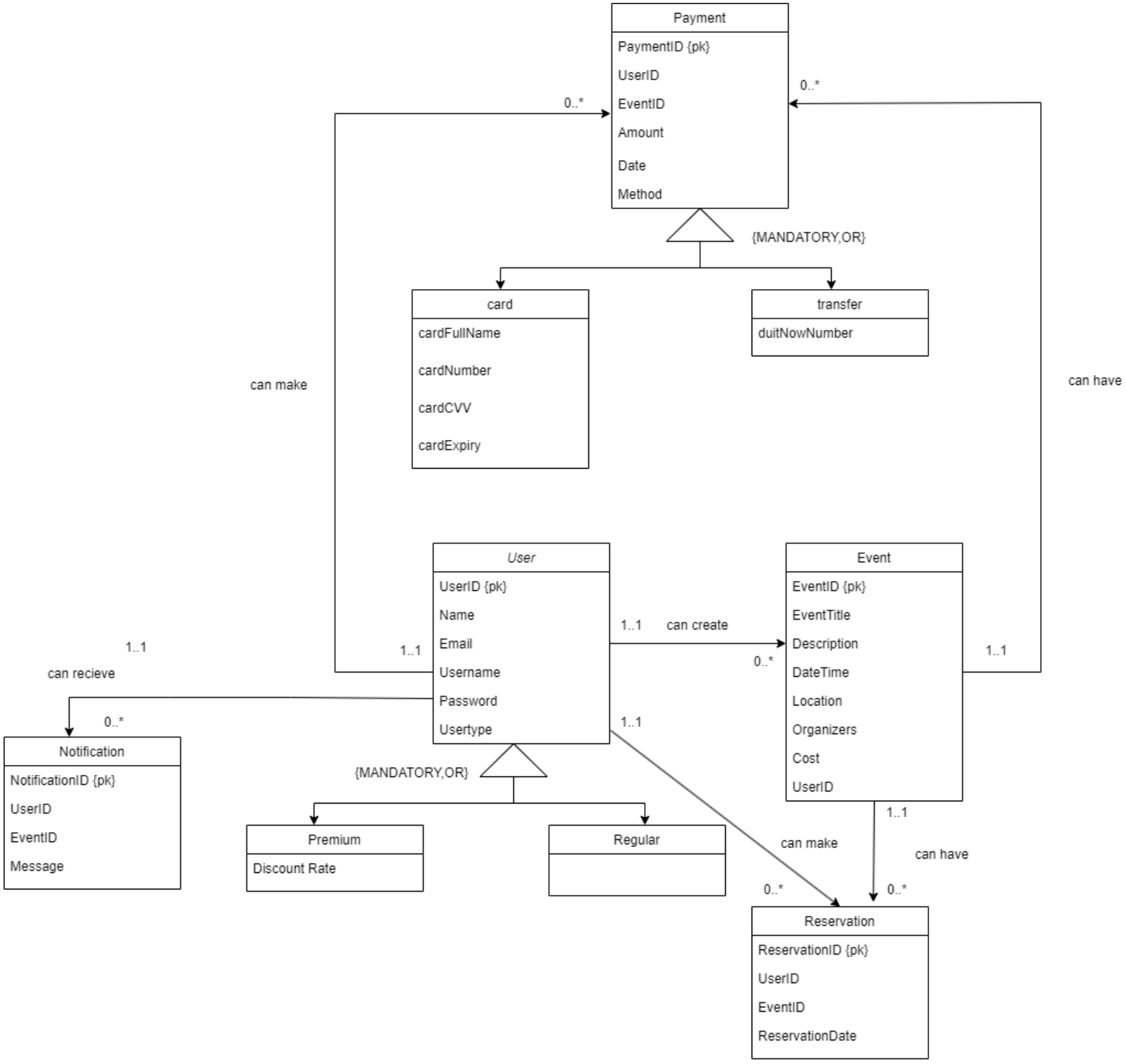


- Each Event can have zero or many Payments.
  - Each Payment can have 1 Event.

3.2 Conceptual ERD:

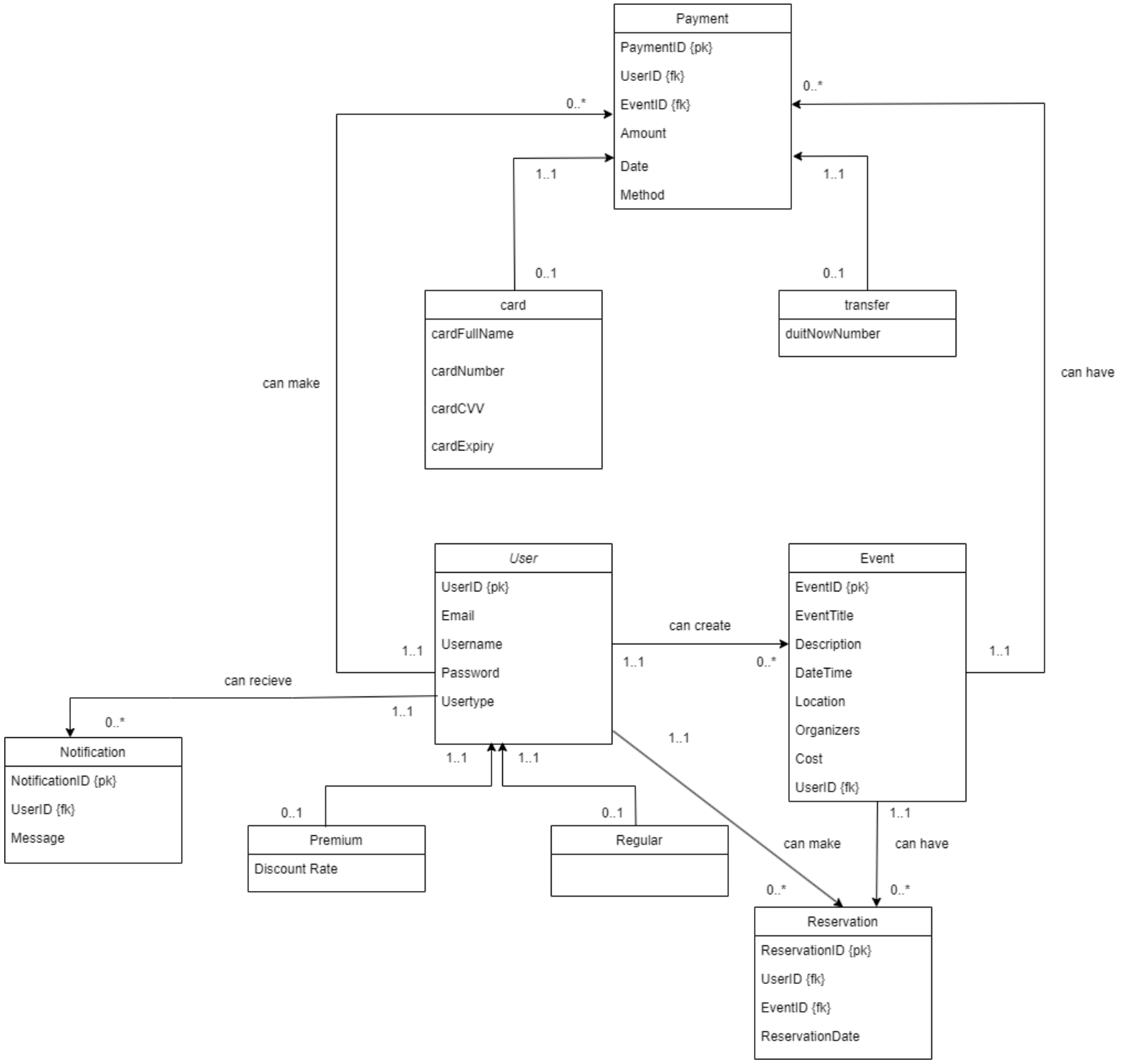


3.3 Enhanced ERD:



4. DB Logical Design

4.1 Logical ERD



4.2 Updated Data Dictionary

4.2.1

Entity : Payment

Attribute	Data Type	Description
PaymentID	Integer	Unique Identifier for each payment (Primary Key)
UserID	Integer	Unique Identifier for each user (Foreign Key)
EventID	Integer	Unique identifier for each event (Foreign Key)
Amount	Float	Amount to be paid
Date	Date	Date of payment/booking
Method	Varchar(20)	Method of Payment

4.2.2

Entity : User

Attribute	Data Type	Description
UserID	Integer	Unique Identifier for each user (Primary Key)
Email	alphanumeric	Email of the user
Username	alphanumeric	username of the user
Password	alphanumeric	Password entered by the user
UserType	Varchar(20)	Identifies the type of user(Manager, Participant , Viewer etc)

Name	varchar(20)	Name of user

4.2.3

Entity : Reservation

Attribute	**Data Type**	**Description**
ReservationID	Integer	Unique Identifier for each reservation (Primary Key)
UserID	Integer	Unique Identifier for each user (Foreign Key)
EventID	Integer	Unique Identifier for each event (Foreign Key)
ReservationDate	Date	Date of Reservation

4.2.4

Entity : Notification

Attribute	Data Type	Description
NotificationID	Integer	Unique Identifier for each notification (Primary Key)
UserID	Integer	Unique Identifier for each user (Foreign Key)
EventID	Integer	Unique Identifier for each event (Foreign Key)
Message	Varchar(50)	Message received by the user

4.2.5

Entity : Event

Attribute	Data Type	Description
EventID	Integer	Unique Identifier for each event (Primary Key)
EventTitle	varchar(20)	Title of the Event
Description	varchar(50)	Description of the Event
DateTime	Date	Date and time of the Reservation
Location	varchar(30)	Location of the Event
Organization	varchar(20)	Name of the Organization holding the Event
Cost	float	Cost of the Event

4.2.6

Entity: Card

Attribute	Data Type	Description
cardFullName	varchar(30)	Full Name found in card
cardNo	Integer	Number found in card
cardCVV	Integer	Number found behind card
cardExpiry	date	Date of expiry of card

4.2.7

Entity: Premium

Attribute	Data Type	Description
DiscountRate	Integer	The discount rate for a user

4.2.8

Entity: Transfer

Attribute	Data Type	Description
duitNowNumber	Integer	The duitNow number for transfer

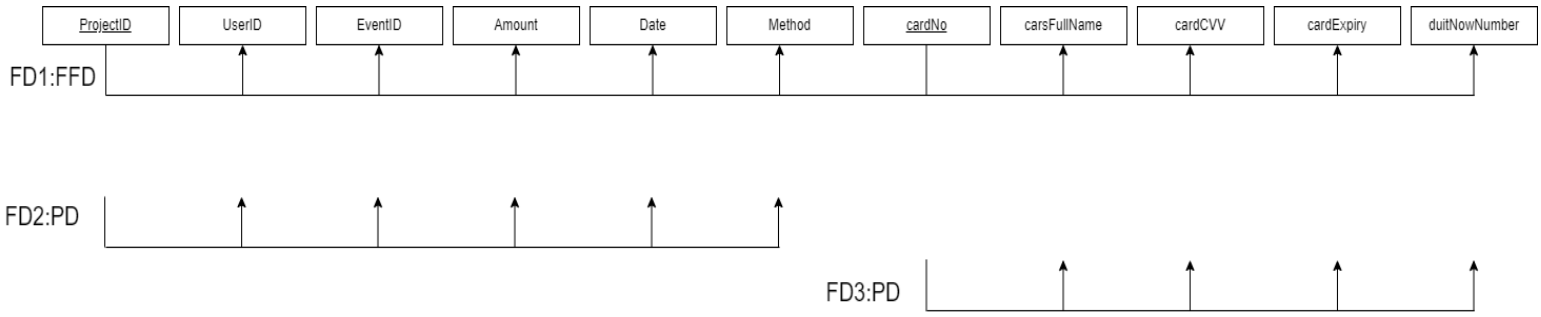
4.3 Normalization

1. **Payment:**

Example:

PaymentID	UserID	EventID	Amount	Date	Method
0023	3210	0009	17.85	19-MAY-23	VISA
0102	3211	0010	90.10	02-JAN-24	TnG

cardFullName	cardNo	cardCVV	cardExpiry	duitNowNumber
Omar Mohammed Rezk	1214-2342-4235-2142	456	01-JAN-29	897128978123
Hasan Hatem Hasan	2131-2312-8043-1349	123	02-DEC-30	904728475920



1NF:

1. payment(PaymentID, UserID, EventID, Amount, Date, Method, cardFullName, cardNo, cardCVV, cardExpiry, duitNowNumber)
    a. Primary Keys: (PaymentID, cardNo)

2NF:

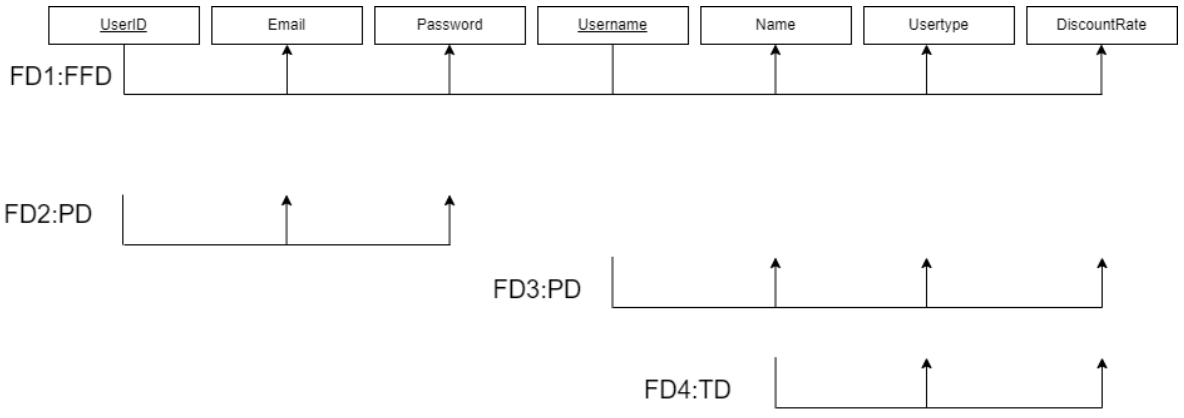
1. infoP(PaymentID, UserID, EventID, Amount, Date, Method)
    a. Primary Keys: (PaymentID)

2. infoC(cardFullName, cardNo, cardCVV, cardExpiry, duitNowNumber)
    a. Primary Keys: (cardNo)
3. payment(paymentID, cardNo)
    a. Primary Keys: (PaymentID, cardNo)
    b. Foreign Keys: (PaymentID) REFERENCES infoP(PaymentID)
    c. Foreign Keys: (cardNo) REFERENCES infoC(cardNo)

2. User:

Example:

UserID	Name	Email	Username	Password	Usertype	DiscountRate
3210	Omar	rezk@graduate.utm.my	3mR	password11	Organizer	30%
3211	Hasan	0102	hsn1	910pswd	Regular	7%



1NF:

1. user(UserID, Name, Email, Username, Password, Usertype, DiscountRate)
    a. Primary Keys: (UserID, Username)

2NF:

1. infoUser(UserID, Email, Password)
    a. Primary Keys: (UserID)
2. fullUserInfo(Username, Name, Usertype, DiscountRate)
    a. Primary Keys: (Username)

3. User(<u>UserID</u>, <u>Username</u>)
    a. Primary Keys: (UserID, Username)
    b. Foreign Key: UserID reference infoUser(UserID)
    c. Foreign Key: Username reference fullUserInfo(Username)

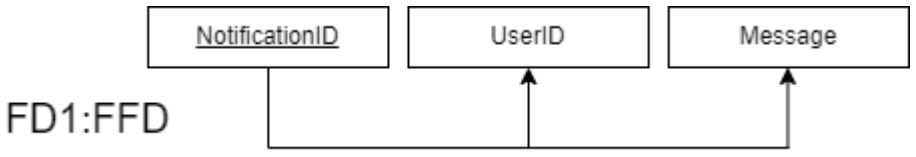
3NF:

1. infoName(<u>Name</u>, Usertype, DiscountRate)
    a. Primary Keys: (Name)
2. infoUser(<u>UserID</u>, Email, Password)
    b. Primary Keys: (UserID)
3. fullUserInfo(<u>Username</u>, Name)
    a. Primary Keys: (Username)
    b. Foreign Key: Name reference infoName(name)
4. User(<u>UserID</u>, <u>Username</u>)
    a. Primary Keys: (UserID, Username)
    b. Foreign Key: Username reference fullUserInfo(Username)
    c. Foreign Key: UserID reference infoUser(UserID)

3. Notification:

Example:

NotificationID	UserID	Message
8218	3210	“Payment Received”
8291	3211	“Reserved”



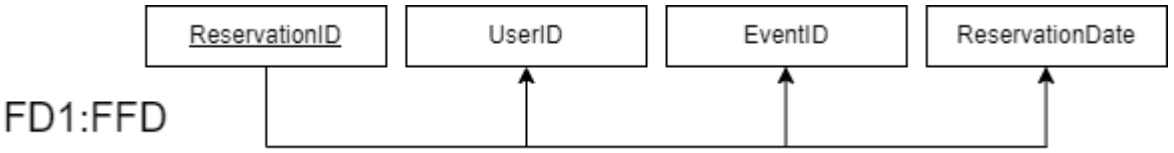
1NF:

1. Notification(<u>NotificationID</u>, UserID, Message)
    a. Primary Keys: (NotificationID)

4. Reservation:

Example:

ReservationID	UserID	EventID	ReservationDate
2111	3210	0009	06-JUN-24
2112	3211	0010	31-MAY-24



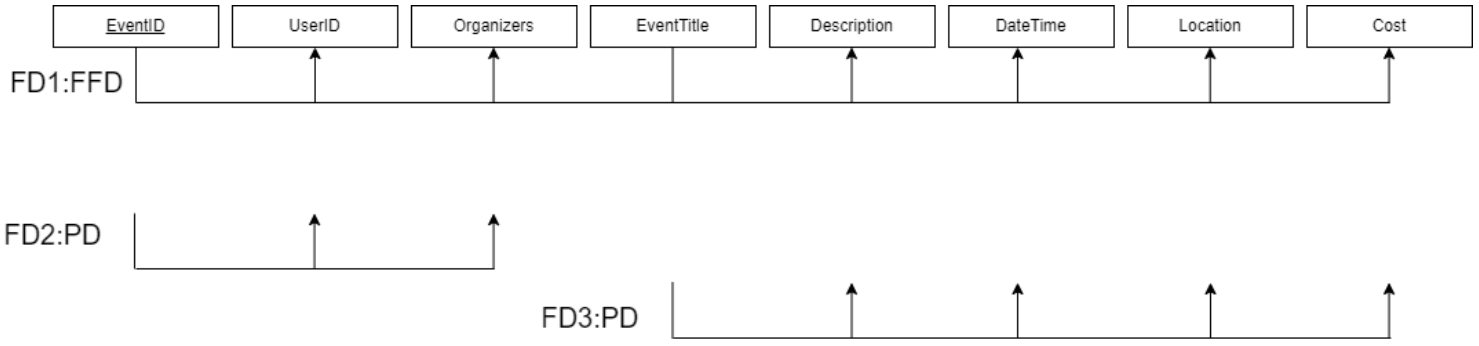
1NF:

1. **Reservation(ReservationID, UserID, EventID, ReservationDate)**
    a. **Primary Keys: (ReservationID)**

5. Event:

Example:

EventID	UserID	EventTitle	Description	DateTime	Location	Organizers	Cost
0009	3210	Road to SQL	Sign Up for a journey full of SQL.	06-MAY-24	UTM, N28, MPK 7	Omar	25.50



1NF:

1. **Event(<u>EventID</u>, UserID, <u>EventTitle</u>, Description, Location, Organizers, Cost)**
    a. **Primary Keys: (EventID, EventTitle)**

2NF:

1. **infoEvent(<u>EventID</u>, UserID, Organizers)**
    c. **Primary Keys: (EventID)**
2. **fullEventInfo(<u>EventTitle</u>, Description, DateTime, Location, Cost)**
    a. **Primary Keys: (EventTitle)**
3. **Event(<u>EventID</u>, <u>EventTitle</u>)**
    a. **Primary Keys: (EventID, EventTitle)**
    b. **Foreign Key: EventID reference for infoEvent(EventID)**
    c. **Foreign Key: EventTitle reference for fullEventInfo(EventTitle)**

5. Relational DB Schemas (after normalization)

These are the set of relation schemas in relational database schema for NexScholar database.

infoP(PaymentID, UserID, EventID, Amount, Date, Method)
infoC(cardFullName, cardNo, cardCVV, cardExpiry, duitNowNumber)
payment(paymentID, cardNo)
infoName(Name, Usertype, DiscountRate)
infoUser(UserID, Email, Password)
fullUserInfo(Username, Name)
User(UserID, Username)
Notification(NotificationID, UserID, Message)
Reservation(ReservationID, UserID, EventID, ReservationDate)
infoEvent(EventID, UserID, Organizers)
fullEventInfo(EventTitle, Description, DateTime, Location, Cost)
Event(EventID, EventTitle)

6. SQL Statements (DDL & DML)

```
CREATE TABLE infoP (
    PaymentID INTEGER PRIMARY KEY,
    UserID INTEGER,
    EventID INTEGER,
    Amount FLOAT,
    Dte DATE,
    Methods VARCHAR2(20)
);

CREATE TABLE infoC (
    cardFullName VARCHAR2(30),
    cardNo INTEGER PRIMARY KEY,
    cardCVV INTEGER,
    cardExpiry DATE,
    duitNowNumber INTEGER
);

CREATE TABLE payment (
    PaymentID INTEGER,
    cardNo INTEGER,
    PRIMARY KEY (PaymentID, cardNo),
             FOREIGN   KEY   (PaymentID)   REFERENCES
infoP(PaymentID),
    FOREIGN KEY (cardNo) REFERENCES infoC(cardNo)
);

CREATE TABLE infoName (
    Name VARCHAR2(20) PRIMARY KEY,
    Usertype VARCHAR2(20),
    DiscountRate INTEGER
);
```

```
CREATE TABLE infoUser (
    UserID INTEGER PRIMARY KEY,
    Email VARCHAR2(50),
    Password VARCHAR2(20)
);

CREATE TABLE fullUserInfo (
    Username VARCHAR2(20) PRIMARY KEY,
    Name VARCHAR(25),
    FOREIGN KEY (Name) REFERENCES infoName(Name)
);

CREATE TABLE Users (
    UserID INTEGER,
    Username VARCHAR(20),
    PRIMARY KEY (UserID, Username),
              FOREIGN   KEY   (Username)   REFERENCES
fullUserInfo(Username),
    FOREIGN KEY (UserID) REFERENCES infoUser(UserID)
);

CREATE TABLE Notification (
    NotificationID INTEGER PRIMARY KEY,
    UserID INTEGER,
    Message VARCHAR2(255)
);

CREATE TABLE Reservation (
    ReservationID INTEGER PRIMARY KEY,
    UserID INTEGER,
    EventID INTEGER,
```

```
    ReservationDate DATE
);

CREATE TABLE infoEvent (
    EventID INTEGER PRIMARY KEY,
    UserID INTEGER,
    Organizers VARCHAR2(255)
);

CREATE TABLE fullEventInfo (
    EventTitle VARCHAR2(30) PRIMARY KEY,
    Description VARCHAR2(70),
    DateTime Date,
    Location VARCHAR2(50),
    Cost FLOAT
);

CREATE TABLE Event (
    EventID INTEGER,
    EventTitle VARCHAR2(30),
    PRIMARY KEY (EventID, EventTitle),
              FOREIGN   KEY   (EventID)   REFERENCES
infoEvent(EventID),
             FOREIGN   KEY   (EventTitle)   REFERENCES
fullEventInfo(EventTitle)
);
```

Record 1:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Omar Mohammed', 'Organizer', 30);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0001, 'omar@nexscholar.com', 'password123');

INSERT INTO fullUserInfo (Username, Name) VALUES ('3mr', 'Omar Mohammed');

INSERT INTO Users (UserID, Username) VALUES (0001, '3mr');

INSERT INTO infoEvent (EventID, UserID, Organizers) VALUES (0001, 0001, 'Omar Mohammed');

INSERT INTO fullEventInfo (EventTitle, Description, Location, Cost) VALUES ('Road to SQL', 'Get to know about SQL more!', 'UTM, N28A', 30.50);

INSERT INTO Event (EventID, EventTitle) VALUES (0001, 'Road to SQL');
```

Record 2:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Hasan Hatem', 'Organizer', 25);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0002, 'hasan@nexscholar.com', 'password456');

INSERT INTO fullUserInfo (Username, Name) VALUES ('hasn1', 'Hasan Hatem');

INSERT INTO Users (UserID, Username) VALUES (0002, 'hasn1');

INSERT INTO infoEvent (EventID, UserID, Organizers) VALUES (0002, 0002, 'Hasan Hatem');

INSERT INTO fullEventInfo (EventTitle, Description, DateTime, Location, Cost) VALUES ('Art Exhibition', 'Explore beautiful artworks',
    TO_TIMESTAMP ('15-Nov-24 18:30:00.000000', 'DD-Mon-RR HH24:MI:SS.FF'), 'Art Gallery, A23', 15.00);

INSERT INTO Event (EventID, EventTitle) VALUES (0002, 'Art Exhibition');
```

Record 3:

```
 INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Ahmed Amir', 'Organizer', 25);

 INSERT INTO infoUser (UserID, Email, Password) VALUES (0003, 'ahmed@nexscholar.com', 'password789');

 INSERT INTO fullUserInfo (Username, Name) VALUES ('ahmd1', 'Ahmed Amir');

 INSERT INTO Users (UserID, Username) VALUES (0003, 'ahmd1');

 INSERT INTO infoEvent (EventID, UserID, Organizers) VALUES (0003, 0003, 'Ahmed Amir');

 INSERT INTO fullEventInfo (EventTitle, Description, DateTime, Location, Cost) VALUES ('Tech Symposium', 'Discover the latest in technology',
     TO_TIMESTAMP ('20-Mar-03 09:00:00.000000', 'DD-Mon-RR HH24:MI:SS.FF'), 'Convention Center, C12', 50.00);

 INSERT INTO Event (EventID, EventTitle) VALUES (0003, 'Tech Symposiumn');
```

Record 4:

```
 INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Thanes', 'Organizer', 30);

 INSERT INTO infoUser (UserID, Email, Password) VALUES (0004, 'thanes@nexscholar.com', 'password789');

 INSERT INTO fullUserInfo (Username, Name) VALUES ('thanes21', 'Thanes');

 INSERT INTO Users (UserID, Username) VALUES (0004, 'thanes21');

 INSERT INTO infoEvent (EventID, UserID, Organizers) VALUES (0004, 0004, 'Thanes');

 INSERT INTO fullEventInfo (EventTitle, Description, DateTime, Location, Cost)
     VALUES ('Fitness Workshop', 'Stay fit and healthy', TO_TIMESTAMP ('05-Jun-24 17:00:00.000000', 'DD-Mon-RR HH24:MI:SS.FF'), 'Fitness Studio, F5', 20.00);

 INSERT INTO Event (EventID, EventTitle) VALUES (0004, 'Fitness Workshop');
```

Record 5:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Zheng', 'Organizer', 20);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0005, 'zheng@nexscholar.com', 'password789');

INSERT INTO fullUserInfo (Username, Name) VALUES ('zheng12', 'Zheng');

INSERT INTO Users (UserID, Username) VALUES (0005, 'zheng12');

INSERT INTO infoEvent (EventID, UserID, Organizers) VALUES (0005, 0005, 'Zheng');

INSERT INTO fullEventInfo (EventTitle, Description, DateTime, Location, Cost)
    VALUES ('Music Festival', 'Enjoy live music performances', TO_TIMESTAMP ('12-Aug-24 19:30:00.000000', 'DD-Mon-RR HH24:MI:SS.FF'), 'Outdoor Arena, 08', 40.00);

INSERT INTO Event (EventID, EventTitle) VALUES (0005, 'Music Festival');
```

Record 6:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Adam', 'Admin', 40);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0006, 'adam@nexscholar.com', 'password239');

INSERT INTO fullUserInfo (Username, Name) VALUES ('ad4m110', 'Adam');

INSERT INTO Users (UserID, Username) VALUES (0006, 'ad4m110');
```

Record 7:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Eve', 'Admin', 15);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0007, 'eve@nexscholar.com', 'password123');

INSERT INTO fullUserInfo (Username, Name) VALUES ('eve_user', 'Eve');

INSERT INTO Users (UserID, Username) VALUES (0007, 'eve_user');
```

Record 8:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Bob', 'Admin', 10);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0008, 'bob@nexscholar.com', 'securepass');

INSERT INTO fullUserInfo (Username, Name) VALUES ('bob_123', 'Bob');

INSERT INTO Users (UserID, Username) VALUES (0008, 'bob_123');
```

Record 9:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Charlie', 'Admin', 25);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0009, 'charlie@nexscholar.com', 'strongpassword');

INSERT INTO fullUserInfo (Username, Name) VALUES ('charlie11', 'Charlie');

INSERT INTO Users (UserID, Username) VALUES (0009, 'charlie11');
```

Record 10:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('David', 'Admin', 5);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0010, 'david@nexscholar.com', 'pass110');

INSERT INTO fullUserInfo (Username, Name) VALUES ('david10', 'David');

INSERT INTO Users (UserID, Username) VALUES (0010, 'david10');
```

Record 11:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Cole', 'Regular', 0);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0011, 'david@example.com', '9akpass');

INSERT INTO fullUserInfo (Username, Name) VALUES ('cole99', 'Cole');

INSERT INTO Users (UserID, Username) VALUES (0011, 'cole99');

INSERT INTO Reservation (ReservationID, UserID, EventID, ReservationDate) VALUES (0001, 0003, 0001, TO_DATE('2024-09-10', 'YYYY-MM-DD'))

INSERT INTO Notification (NotificationID, UserID, Message) VALUES (0001, 0011, 'Reserve Completed');

INSERT INTO infoP (PaymentID, UserID, EventID, Amount, Dte, Methods) VALUES (0001, 0011, 0001, 30.5, TO_DATE('2024-01-15', 'YYYY-MM-DD'), 'Credit Card');

INSERT INTO infoC (cardFullName, cardNo, cardCVV, cardExpiry, duitNowNumber) VALUES ('Cole', 1234567890123456, 123, TO_DATE('2027-11-25', 'YYYY-MM-DD'), 9876543210);

INSERT INTO payment (PaymentID, cardNo) VALUES (0001, 1234567890123456);

INSERT INTO Notification (NotificationID, UserID, Message) VALUES (0002, 0011, 'Payment Completed');
```

Record 12:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Liam', 'Regular', 0);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0012, 'Liam@example.com', '9akpass');

INSERT INTO fullUserInfo (Username, Name) VALUES ('Liamole99', 'Cole');

INSERT INTO Users (UserID, Username) VALUES (0012, 'Liamole99');

INSERT INTO Reservation (ReservationID, UserID, EventID, ReservationDate) VALUES (0002, 0012, 0001, TO_DATE('2024-09-10', 'YYYY-MM-DD'));

INSERT INTO Notification (NotificationID, UserID, Message) VALUES (0003, 0012, 'Reserve Completed');

INSERT INTO infoP (PaymentID, UserID, EventID, Amount, Dte, Methods) VALUES (0001, 0012, 0001, 30.5, TO_DATE('2024-01-15', 'YYYY-MM-DD'), 'Credit Card');

INSERT INTO infoC (cardFullName, cardNo, cardCVV, cardExpiry, duitNowNumber) VALUES ('Cole', 1234567890123456, 123, TO_DATE('2029-05-22', 'YYYY-MM-DD'), 2643534534);

INSERT INTO payment (PaymentID, cardNo) VALUES (0001, 2947294065818075);

INSERT INTO Notification (NotificationID, UserID, Message) VALUES (0004, 0012, 'Payment Completed');
```

Record 13:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Liam', 'Regular', 0);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0012, 'Liam@example.com', '9akpass');

INSERT INTO fullUserInfo (Username, Name) VALUES ('Liamole99', 'Liam');

INSERT INTO Users (UserID, Username) VALUES (0012, 'Liamole99');

INSERT INTO Reservation (ReservationID, UserID, EventID, ReservationDate) VALUES (0002, 0012, 0001, TO_DATE('2024-09-10', 'YYYY-MM-DD'));

INSERT INTO Notification (NotificationID, UserID, Message) VALUES (0003, 0012, 'Reserve Completed');

INSERT INTO infoP (PaymentID, UserID, EventID, Amount, Dte, Methods) VALUES (0002, 0012, 0001, 30.5, TO_DATE('2024-01-15', 'YYYY-MM-DD'), 'Credit Card');

INSERT INTO infoC (cardFullName, cardNo, cardCVV, cardExpiry, duitNowNumber) VALUES ('Liam', 1234567890123456, 123, TO_DATE('2029-05-22', 'YYYY-MM-DD'), 2643534534);

INSERT INTO payment (PaymentID, cardNo) VALUES (0002, 1234567890123456);
```

Record 14:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Mohammed', 'Regular', 0);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0013, 'moh@example.com', 'sfas9');

INSERT INTO fullUserInfo (Username, Name) VALUES ('moh99', 'Mohammed');

INSERT INTO Users (UserID, Username) VALUES (0013, 'moh99');

INSERT INTO Reservation (ReservationID, UserID, EventID, ReservationDate) VALUES (0003, 0013, 0001, TO_DATE('2024-09-10', 'YYYY-MM-DD'));

INSERT INTO Notification (NotificationID, UserID, Message) VALUES (0005, 0013, 'Reserve Completed');

INSERT INTO infoP (PaymentID, UserID, EventID, Amount, Dte, Methods) VALUES (0007, 0012, 0001, 30.5, TO_DATE('2024-01-15', 'YYYY-MM-DD'), 'Credit Card');

INSERT INTO infoC (cardFullName, cardNo, cardCVV, cardExpiry, duitNowNumber) VALUES ('Mohammed', 1234567890123457, 123, TO_DATE('2029-05-22', 'YYYY-MM-DD'), 2643534534);

INSERT INTO payment (PaymentID, cardNo) VALUES (0007, 1234567890123457);
```

Record 15:

```
INSERT INTO infoName (Name, Usertype, DiscountRate) VALUES ('Aseel', 'Regular', 0);

INSERT INTO infoUser (UserID, Email, Password) VALUES (0014, 'aseel@example.com', 'asle9');

INSERT INTO fullUserInfo (Username, Name) VALUES ('aseel99', 'Aseel');

INSERT INTO Users (UserID, Username) VALUES (0014, 'aseel99');

INSERT INTO Reservation (ReservationID, UserID, EventID, ReservationDate) VALUES (0004, 0014, 0001, TO_DATE('2024-09-10', 'YYYY-MM-DD'));

INSERT INTO Notification (NotificationID, UserID, Message) VALUES (0009, 0014, 'Reserve Completed');

INSERT INTO infoP (PaymentID, UserID, EventID, Amount, Dte, Methods) VALUES (0009, 0012, 0001, 30.5, TO_DATE('2024-01-15', 'YYYY-MM-DD'), 'Credit Card');

INSERT INTO infoC (cardFullName, cardNo, cardCVV, cardExpiry, duitNowNumber) VALUES ('Aseel', 1234567890123458, 123, TO_DATE('2029-05-22', 'YYYY-MM-DD'), 234234436);

INSERT INTO payment (PaymentID, cardNo) VALUES (0009, 1234567890123458);
```

DML:

1. **SELECT * FROM NOTIFICATION**

```
1  SELECT * FROM NOTIFICATION
```

NOTIFICATIONID	USERID	MESSAGE
5	13	Reserve Completed
3	12	Reserve Completed
4	12	Payment Completed
9	14	Reserve Completed

2. **DELETE FROM Notification WHERE NotificationID = 5;**

```
1  DELETE FROM Notification WHERE NotificationID = 1;
```

NOTIFICATIONID	USERID	MESSAGE
3	12	Reserve Completed
4	12	Payment Completed
9	14	Reserve Completed

3. **UPDATE infoUser SET Email = 'newemail@example.com' WHERE UserID = 2;**

```
1  UPDATE infoUser SET Email = 'newemail@example.com' WHERE UserID = 2;
```

1 row(s) updated.

USERID	EMAIL	PASSWORD
2	newemail@example.com	password456

4. SELECT P.PaymentID, U.Email, P.Amount, P.Dte, P.Methods, E.EventID, E.Organizers
FROM infoP P
JOIN infoUser U ON P.UserID = U.UserID
JOIN infoEvent E ON P.EventID = E.EventID;

```
SELECT P.PaymentID, U.Email, P.Amount, P.Dte, P.Methods, E.EventID, E.Organizers
FROM infoP P
JOIN infoUser U ON P.UserID = U.UserID
JOIN infoEvent E ON P.EventID = E.EventID;
```

PAYMENTID	EMAIL	AMOUNT	DTE	METHODS	EVENTID	ORGANIZERS
1	david@example.com	30.5	15-JAN-24	Credit Card	1	Omar Mohammed
7	Liam@example.com	30.5	15-JAN-24	Credit Card	1	Omar Mohammed
2	Liam@example.com	30.5	15-JAN-24	Credit Card	1	Omar Mohammed
3	Liam@example.com	30.5	15-JAN-24	Credit Card	1	Omar Mohammed
9	Liam@example.com	30.5	15-JAN-24	Credit Card	1	Omar Mohammed

5. INSERT INTO Notification (NotificationID, UserID, Message) VALUES (2, 1, 'New Event Alert: Event1 on 2024-01-15');

NOTIFICATIONID	USERID	MESSAGE
3	12	Reserve Completed
4	12	Payment Completed
2	1	New Event Alert: Event1 on 2024-01-15
9	14	Reserve Completed

7. Summary

The NexScholar Event System aims to enhance user satisfaction by improving the platform's user interface, incorporating a safe payment gateway, creating a reservation system, and providing an easy-to-use form for event coordinators to input necessary details. The project involves transforming the conceptual ERD into a logical ERD, generating relational patterns, standardizing data to minimize duplication, and updating the database with normalized relationships. Additionally, the system will be updated with SQL statements to create the website's database. These changes are intended to make the platform more visually appealing, user-friendly, and efficient for both users and event coordinators.