UTM

UNIVERSITI TEKNOLOGI MALAYSIA

Group Project

SECD2523 DATABASE SEMESTER I, SESSION 2023/2024

STAGE : PHASE 3

LECTURER : Dr. IZYAN IZZATI KAMSANI

NAME	MATRIC NUMBER
MUHAMMAD ILMAN BIN MOHD KHAIRI	A22EC0215
MUHAMMAD FAIZ BIN ZAKARIA	A22EC0208
ALIF AIMAN BIN MANSOR	A22EC0137
MUHAMMAD IMAN AMIER BIN ABU BAKAR	A22EC0128
MUHAMAD HAZIQ AMSYAR BIN MOHD HIZWAN	A22EC0079

TABLE OF CONTENT

1.0 Introduction

The team, under the direction of Dr. Seah Choon Sen and with assistance from Drs. Ahmad Najmi Amerhaider Nuar and Muhammad Aliif Ahmad from Universiti Teknologi Malaysia, are preparing to launch a ground-breaking feature on the platform in this last stage of the Nexscholar project. Nexscholar, the CHIPTA 2023 second runner-up, is growing into an even more vibrant and all-encompassing academic networking platform.

This phase's main goal is to incorporate a safe payment gateway and a sophisticated ticketing system into the Nexscholar website. With the introduction of a quick and easy method to gain entry to special events, this improvement seeks to improve the user experience. These events, which are known for their restricted availability and related attendance fees, will now include an expedited booking procedure that necessitates online payments from customers in order to guarantee their spot.

In addition to improving user experience, Nexscholar's integration with this ticketing system gives event organisers insightful data. The procedure for handling exclusive events will be fast and well-organized, providing real-time attendance data and facilitating improved planning and coordination for the organisers.

This cutting-edge function demonstrates Nexscholar's dedication to encouraging cooperation across academic groups. Nexscholar is a powerful force in the digital environment, and the addition of sophisticated ticketing solidifies this as the platform continues to define the future of academic networking. The team sees a future in which the Nexscholar platform smoothly facilitates participation and information sharing throughout the academic community, hence fostering more collaboration among all users.

2.0 Overview of project

Our team is ready to use the abundance of data obtained from administrative needs in this last stage of the Nexscholar project to implement critical system enhancements. The thorough information requirements gleaned from the interviews have been important in comprehending the complexities of the present system and offering a path forward for improvements.

Our team is prepared to translate these insights into a concrete vision for the future system since we have a deep grasp of the goals, processes, and important stakeholders from the previous phase, which has helped us to comprehend the system's demands in more detail. A crucial component of this project is the integration of database improvements. Through the integration of this essential component, we want to strengthen data administration and accessibility, guaranteeing a more resilient and effective Nexscholar platform.

The team will go from idea to visualization as the project nears its conclusion. As data flow diagrams take shape, context diagrams at the parent and child levels will be carefully created. These diagrams will be quite useful in explaining the major business processes, displaying data storage techniques, showing output, and clarifying the entities and information flow.

Crucially, the visualization process will center on the integration of database enhancements, guaranteeing that the TO-BE system not only best serves present requirements but also establishes the groundwork for future scalability and data management. Our team wants to explore the nuances of the present system's flow in order to improve user experience and provide the conditions for a smooth transition to the proposed Nexscholar system.

3.0 Database conceptual design

3.1 Updated business rule

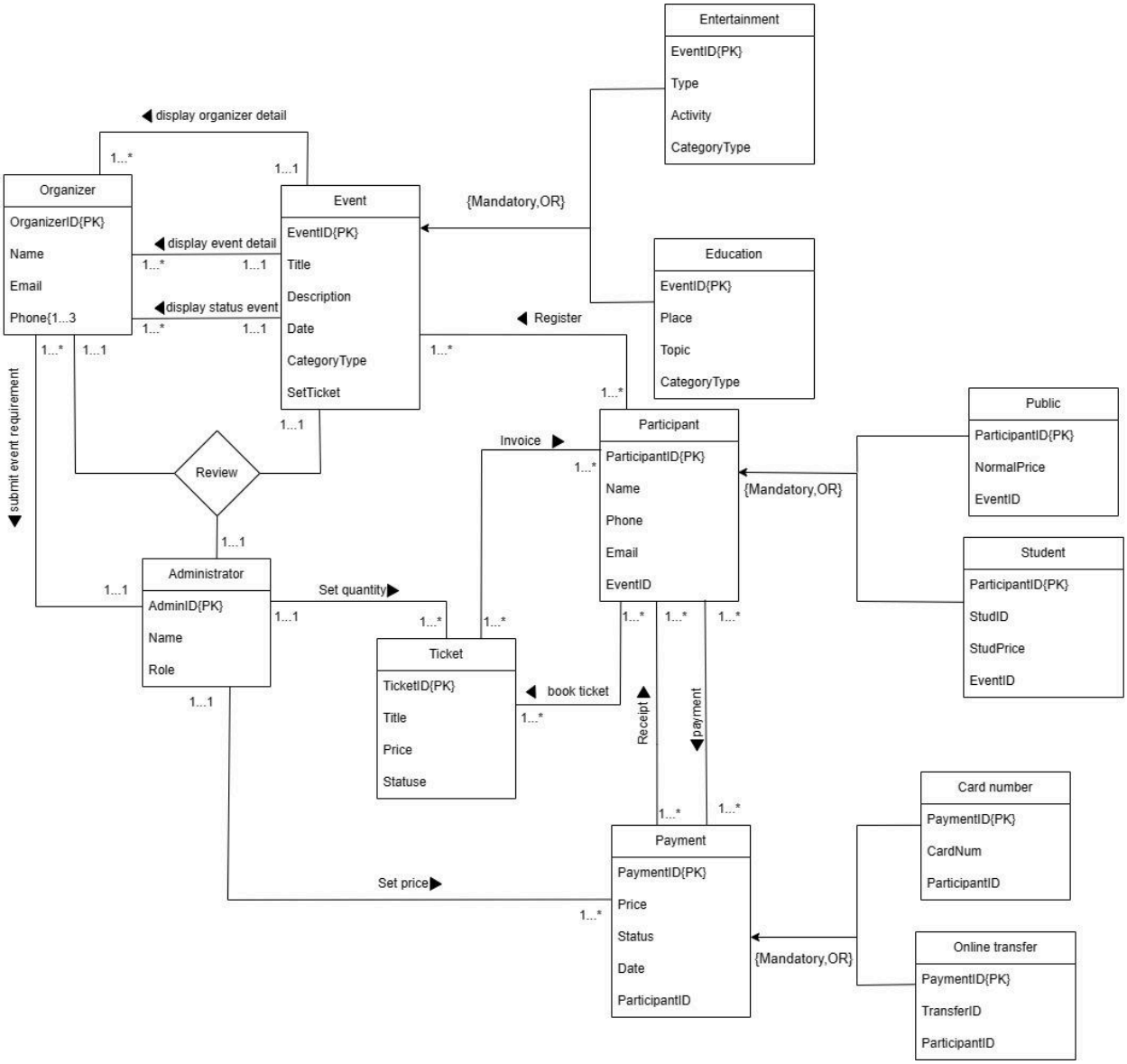
<u>The scenarios and workflow of the updated process for the Administrator:</u>

1. the organizer gets the event information that has to be completed, and they input all the required data.
2. Completed information is kept in the database for event data.
3. The event status is then entered by the admin into the system and is kept in the database of event status information.
4. Whether the event status was successful or not will then be shown in the NexScholar system.
5. the organizer database will contain all of the organizer's information.
6. The administrator will post about event that contains occasion and coordinator in the Nexscholar user interface.

<u>The scenarios and workflow of the updates process for Users:</u>

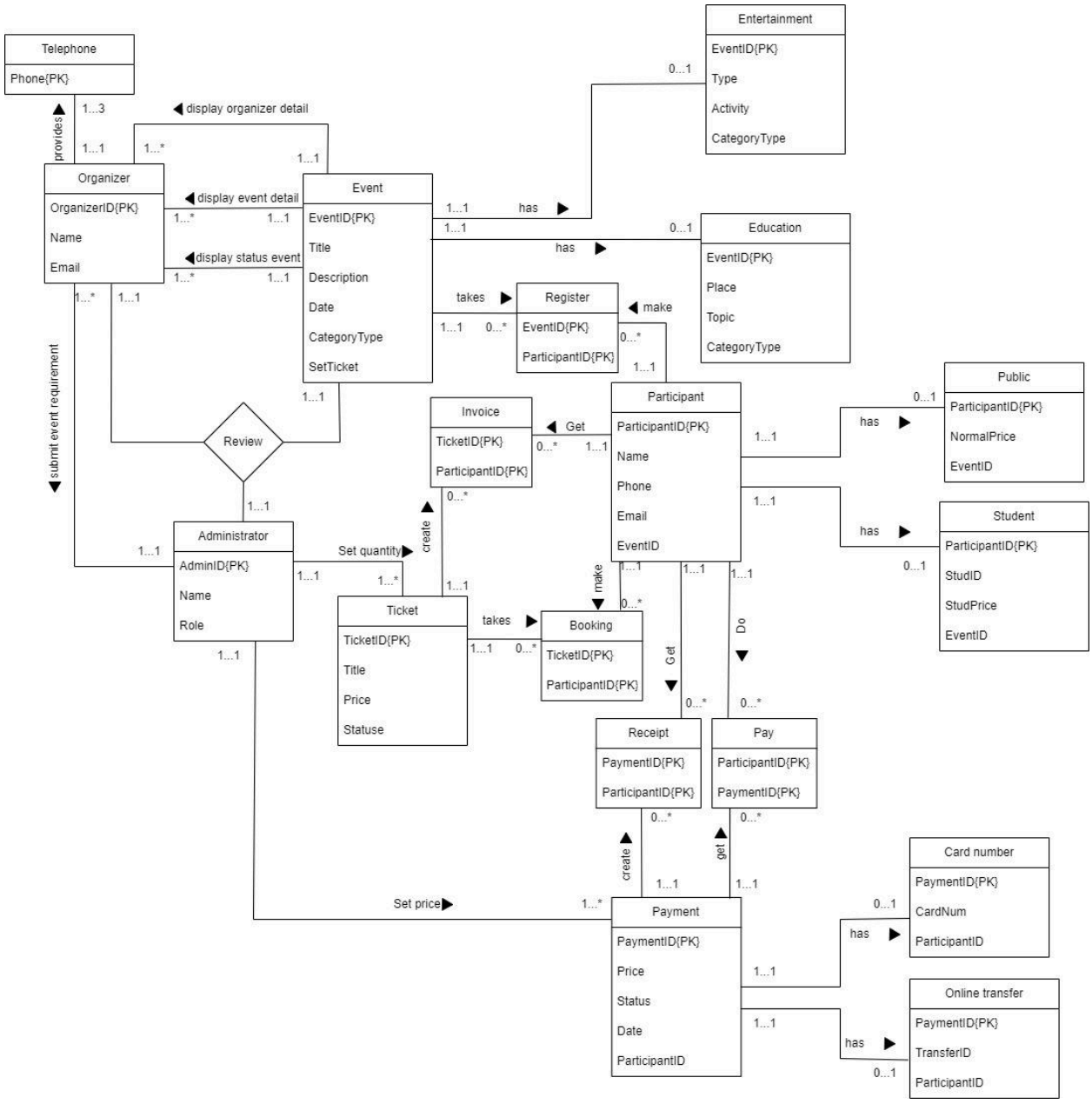
1. The administrator has determined and recorded the quantity of tickets in the ticket database.
2. The administrator has determined the ticket price, which is also recorded in the ticket database.
3. Login to the system as a user.
4. Start on the Activity interface that shows the activity of other users.
5. Click on the Event & Conference.
6. There are two options:
    a. Events Calendar
    b. Events List
7. Events Calendar shows a big calendar with marked dates for all of the events.
8. Events List shows all of the events with event details such as the event's date and poster
9. The user will click to purchase an event ticket.
10. The administrator will receive the information once the order for a ticket has been filled out.
11. The system will notify the user that they must continue with the ticket payment.
12. The user makes the payment for the reserved ticket. Additionally, every payment evidence is added to the ticket database.
13. The user receives an update from the system on the e-ticket and evidence of payment.

3.2 Conceptual ERD



4.0 DB logical design

4.1 Logical ERD



4.2 Updated Data Dictionary

4.2.1 Description of Entity

Entity	Description	Occurrence
Event	Holds event information	Show all the details of the event
Organizer	Holds organizer information	Organizer review the events Organizer submit the event to the system
Administrator	Holds administrator information	Administrator key in the data of the event into the system Administrator track the participant involvement
Participant	Holds participant information	Participant register to join event Participant make the payment Participant book the ticket for the event
Category	Holds the category of the events	Category is click by the user
Ticket	Holds the ticket details	All the price,title and the status of the confirmation of ticket stay here
Payment	Holds the payment details	Administrator set the price for participants Participants do the payment of ticket Sending the receipt to the participants
Education	Holds the category of event for education	User can click for education type of event
Entertainment	Holds the category of event for entertainment	User can click for entertainment type of event
Student	Holds the type of participants which is student	User need to inform whether they are student or not

Public	Holds the type of participants which is public	User need to inform whether they are public or not
Online Transfer	Holds the type of payment	Participants pay the ticket fee by online transfer
Card Number	Holds the type of payment	Participants pay the ticket fee by card number
Telephone	Hold multiple phone number for participant	Participant will input their number if they want to join the event
Register	Holds the registration information	Show the registration information
Invoice	Holds the Invoice information	Show the Invoice information
Booking	Holds the booking information	Show the booking information
Pay	Holds pay information	Show pay information
Receipt	Holds the receipt information	Show the receipt information

4.2.2 Description of Relationship

Entity	Multiplicity	Relationship	Multiplicity	Entity
Organizer	1…*	Submit event	1…1	Administrator
	1…1	review	1…1	Event
	1…1	provides	1…3	Telephone
Event	1…1	Display organizer	1…*	Organizer
	1…1	Display event	1…*	Organizer
	1…1	Display status	1…*	Organizer
	1…1	has	0…1	Education
	1…1	has	0…1	Entertainment
	1…1	takes	0…*	Register
Administrator	1…1	Review	1…1	Event
	1…1	Set quantity	1…*	Ticket
	1…1	Set price	1…*	Payment
Ticket	1…1	create	0…*	Invoice
	1…1	takes	0…*	Booking
Participant	1…*	get	1…*	Invoice
	1…*	make	1…*	Register
	1…1	has	0…1	Public
	1…1	has	0…1	Student
	1…1	get	0…*	Receipt
	1…1	make	0…*	Booking
	1…1	do	0…*	Pay
Payment	1…1	create	0…*	Receipt

	1…1	get	0…*	Pay
	1…1	has	0…1	Card number
	1…1	has	0…1	Online transfer

4.2.3 Description Attributes

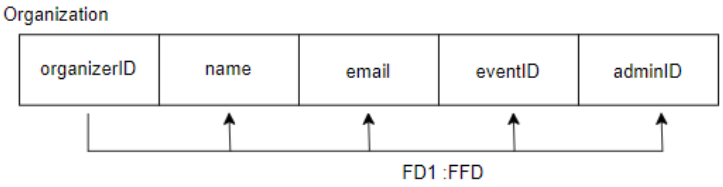
Entity	Attributes	Description	Data Type	Null	Multi-valued
Organizer	OrganizerID(PK)	Uniquely identify an organizer	VARCHAR2(10)	NO	NO
	Name	Name of the organizer	VARCHAR2(30)	NO	NO
	Email	Email of the Organizer	VARCHAR2(50)	NO	NO
Telephone	Phone(PK)	Number phone of the customer	VARCHAR2(13)	NO	NO
Event	EventID(PK)	Uniquely identify an event	VARCHAR2(10)	NO	NO
	Tittle	Title of the event	VARCHAR2(100)	NO	NO
	Description	Description of the event	VARCHAR2(250)	NO	NO
	Date	Date the event will be held	DATE	NO	NO
	CategoryType	identify a category	VARCHAR2(15)	NO	NO
	SetTicket	Set ticket value	NUMBER(4)	NO	NO
Administrator	AdminID (PK)	Uniquely identify an admin	VARCHAR2(10)	NO	NO

	Name	Name of the admin	VARCHAR2(30)	NO	NO
	Role	Role of the admin	VARCHAR2(10)	NO	NO
Participant	ParticipantID(PK)	Uniquely identify a participant	VARCHAR2(10)	NO	NO
	Name	Name of the participant	VARCHAR2(30)	NO	NO
	Phone	Number phone of participant	VARCHAR2(13)	NO	NO
	Email	Email of participant	VARCHAR2(50)	NO	NO
	EventID(FK)	Uniquely identify an event	VARCHAR2(10)	NO	NO
Student	ParticipantID(PK)	Uniquely identify a participant	VARCHAR2(10)	NO	NO
	StudPrice	Price for the student	FLOAT(8)	NO	NO
	StudID	Uniquely identify a student	VARCHAR2(10)	NO	NO
	EventID(FK)	Uniquely identify an event	VARCHAR2(10)	NO	NO
Public	ParticipantID(PK)	Uniquely identify a participant	VARCHAR2(10)	NO	NO
	NormalPrice	Price for the public	FLOAT(8)	NO	NO
	PublicID(PK)	Uniquely identify the public	VARCHAR2(10)	NO	NO
	EventID(FK)	Uniquely identify an event	VARCHAR2(10)	NO	NO
Payment	PaymentID(PK)	Uniquely identify a payment	VARCHAR2(10)	NO	NO
	Price	Amount of the payment	FLOAT(8)	NO	NO
	Status	Status of the payment	VARCHAR2(100)	NO	NO
	Date	Date of the payment	DATE	NO	NO
	ParticipantID(FK)	Uniquely identify the participant	VARCHAR2(10)	NO	NO
Education	EventID(PK)	Uniquely identify an event	VARCHAR2(10)	NO	NO
	Place	The place of the event	VARCHAR2(15)	NO	NO
	Topic	Topic to be discuss	VARCHAR2(30)	NO	NO
	Category Type	Specified education	VARCHAR2(15)	NO	NO
Entertainment	EventID(PK)	Uniquely identify an event	VARCHAR2(10)	NO	NO

	Type	The type of entertainment	VARCHAR2(15)	NO	NO
	Activity	Activity that be held in the event	VARCHAR2(30)	NO	NO
	Category Type	Specified education	VARCHAR2(15)	NO	NO
Online Transfer	PaymentID(PK)	Uniquely identify a payment	VARCHAR2(10)	NO	NO
	TransferID	Uniquely identify the transfer process	VARCHAR2(10)	NO	NO
	ParticipantID(FK)	Uniquely identify the participant	VARCHAR2(10)	NO	NO
Card Number	PaymentID(PK)	Uniquely identify a payment	VARCHAR2(10)	NO	NO
	CardNum	Uniquely identify the card	VARCHAR2(10)	NO	NO
	ParticipantID(FK)	Uniquely identify the participant	VARCHAR2(10)	NO	NO
Ticket	TicketID(PK)	Uniquely identify the ticket	VARCHAR2(10)	NO	NO
	Title	Title of the event	VARCHAR2(30)	NO	NO
	Price	The price of the ticket	FLOAT(8)	NO	NO
	Status	Status of the ticket	VARCHAR2(50)	NO	NO
Register	EventID(PK)	Uniquely identify an event	VARCHAR2(10)	NO	NO
	ParticipantID(PK)	Uniquely identify a participant	VARCHAR2(10)	NO	NO
Invoice	TicketID(PK)	Uniquely identify the ticket	VARCHAR2(10)	NO	NO
	ParticipantID(PK)	Uniquely identify a participant	VARCHAR2(10)	NO	NO
Booking	TicketID(PK)	Uniquely identify the ticket	VARCHAR2(10)	NO	NO
	ParticipantID(PK)	Uniquely identify a participant	VARCHAR2(10)	NO	NO
Receipt	PaymentID(PK)	Uniquely identify a payment	VARCHAR2(10)	NO	NO
	ParticipantID(PK)	Uniquely identify a participant	VARCHAR2(10)	NO	NO
Pay	PaymentID(PK)	Uniquely identify a payment	VARCHAR2(10)	NO	NO
	ParticipantID(PK)	Uniquely identify a participant	VARCHAR2(10)	NO	NO

4.3 Normalization

4.3.1 **Normalization of Organizer relationship**



FD1 : <u>organizerID</u> →name,email,eventID,adminID

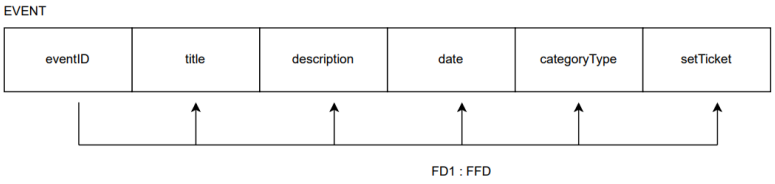
1NF&2NF&3NF : Organizer(<u>organizerID</u>,name,email,eventID,adminID)

PK : organizerID

FK: eventID reference Event(eventID)

FK: adminID reference Administrator(adminID)

4.3.2 **Normalization of Event relationship**

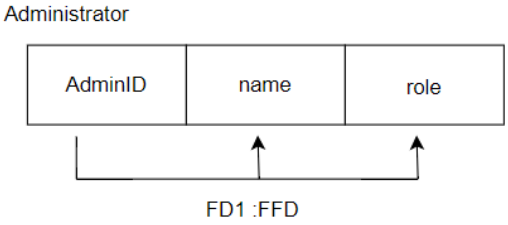


FD1 : <u>eventID</u> →title,description,date,categoryType,setTicket

1NF&2NF&3NF : Event(<u>eventID</u>,title,description,date,categoryType,setTicket)

PK : eventID

4.3.3 **Normalization of Administrator relationship**

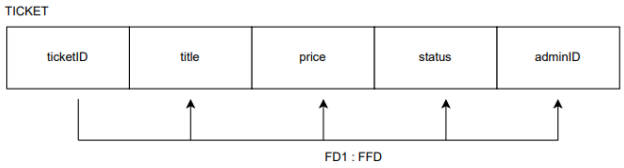


FD1 : adminID →name,role

1NF&2NF&3NF : Administrator(adminID ,name,role)

PK : adminID

4.3.4 **Normalization of Ticket relationship**



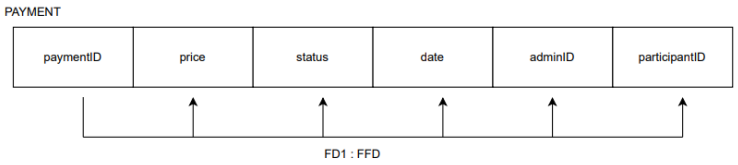
FD1 : ticketID →title,price,status,setTotal,adminID

1NF&2NF&3NF :Ticket(ticketID,title,price,status,adminID)

PK : ticketID

FK: adminID reference Administrator(adminID)

4.3.5 **Normalization of Payment relationship**



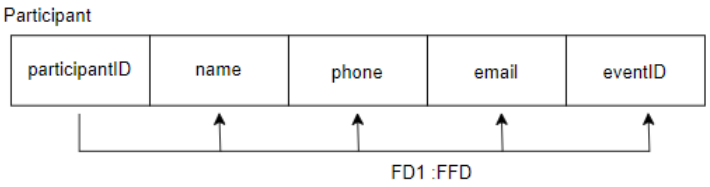
FD1 : paymentID →price,status,date,participantID,adminID

1NF&2NF&3NF :Payment(paymentID,price,status,date,participantID,adminID)

PK : paymentID

FK: adminID reference Administrator(adminID)

4.3.6 **Normalization of Participant relationship**



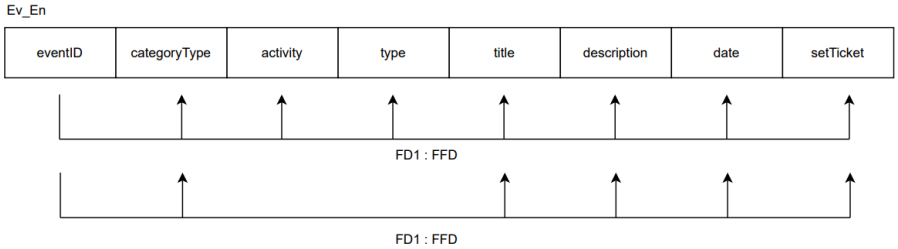
FD1 : participantID →name,phone,email,eventID

1NF&2NF&3NF : Participant(participantID,name,phone,email,eventID)

PK : particpantID

FK: eventID reference Event(eventID)

4.3.7 **Normalization of Ev_En relationship**



FD1 : eventID →categoryType,activity,type,title,description,date,setTicket

FD2 : eventID→title,description,date,setTicket

1NF : Ev_En(eventID,categoryType,activity,type,title,description,date,setTicket)

PK : eventID

2NF : a) Event(eventID,title,description,date,categoryType,setTicket)

PK : eventID

b) Ev_En(eventID,activity,type)

PK : eventID

FK : eventID reference Event(eventID)

3NF : a) Event(<u>eventID</u>,title,description,date,categoryType,setTicket)

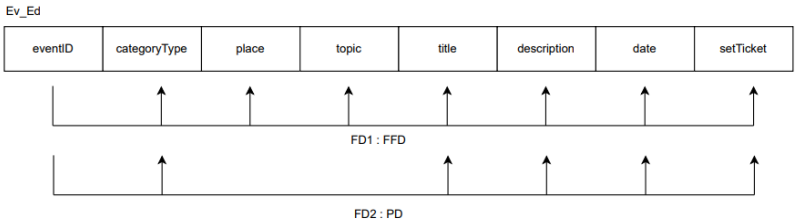
PK : eventID

b) Ev_En(<u>eventID</u>,activity,type)

PK : eventID

FK : eventID reference Event(eventID)

4.3.8 **Normalization of Ev_Ed relationship**



FD1 : <u>eventID</u> →categoryType,place,topic,title,description,date,setTicket

FD2 : <u>eventID</u>→title,description,date,setTicket

1NF : Ev_Ed(<u>eventID</u>,categoryType,place,topic,title,description,date,setTicket)

PK : eventID

2NF : a) Event(<u>eventID</u>,title,description,date,categoryType,setTicket)

PK : eventID

b) Ev_Ed(<u>eventID</u>,place,topic)

PK : eventID

FK : eventID reference Event(eventID)

3NF : a) Event(<u>eventID</u>,title,description,date,categoryType,setTicket)

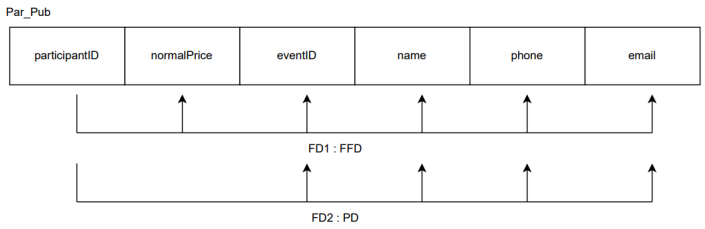
PK : eventID

b) Ev_Ed(eventID,place,topic)

PK : eventID

FK : eventID reference Event(eventID)

4.3.9 **Normalization of Par_Pub relationship**



FD1 : participantID →normalPrice,eventID,name,phone,email

FD2 : eventID→event,IDname,phone,email

1NF : Par_Pub(participantID,normalPrice,eventID,name,phone,email)

PK : participantID

FK : eventID reference Event(eventID)

2NF : a) Participant(participantID,eventID,name,phone,email)

PK : participantID

FK : eventID reference Event(eventID)

b) Par_Pub(participantID,normalPrice)

PK : participantID

FK : participantID reference Participant(participantID)

3NF :a) Participant(participantID,eventID,name,phone,email)

PK : participantID

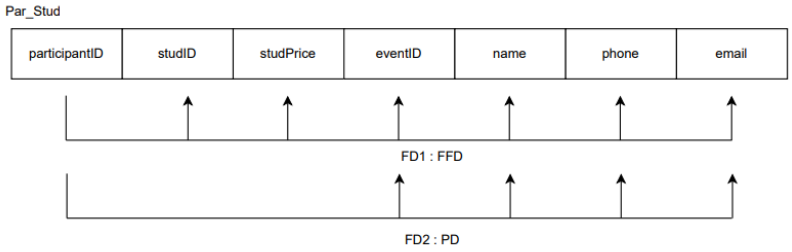
FK : eventID reference Event(eventID)

b) Par_Pub(participantID,normalPrice)

PK : participantID

FK : participantID reference Participant(participantID)

4.3.10 **Normalization of Par_Stud relationship**



FD1 : participantID →studID,studPrice,eventID,name,phone,email

FD2 : participantID→name,phone,email,eventID

1NF : Par_Stud(participantID,studID,studPrice,eventID,name,phone,email)

PK : participantID

FK : eventID reference Event(eventID)

2NF : a) Participant(participantID,name,phone,email,eventID)

PK : participantID

FK : eventID reference Event(eventID)

b) Par_Stud(participantID,studID,studPrice)

PK : participantID

FK : participantID reference Participant(participantID)

3NF : a) Participant(participantID,name,phone,email,eventID)

PK : participantID

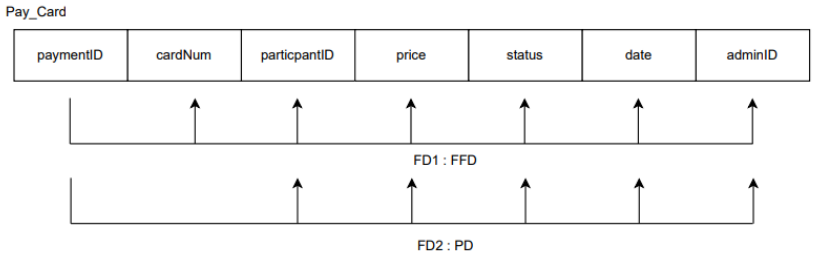
FK : eventID reference Event(eventID)

b) Par_Stud(participantID,studID,studPrice)

PK : participantID

FK : participantID reference Participant(participantID)

4.3.11 **Normalization of Pay_Card relationship**



FD1 : paymentID →cardNum,participantID,price,status,date,adminID

FD2 : paymentID→price,status,date,participantID,adminID

1NF : Pay_Card(paymentID,cardNum,participantID,price,status,date,adminID)

PK : paymentID

FK : participantID reference Participant(participantID)

FK : adminID reference Administrator(adminID)

2NF : a) Payment(paymentID,price,status,date,participantID,adminID)

PK : paymentID

FK : participantID reference Participant(participantID)

FK : adminID reference Administrator(adminID)

b) Pay_Card(paymentID,cardNum)

PK : paymentID

FK : paymentID reference Payment(paymentID)

3NF : a) Payment(paymentID,price,status,date,participantID,adminID)

PK : paymentID

FK : participantID reference Participant(participantID)

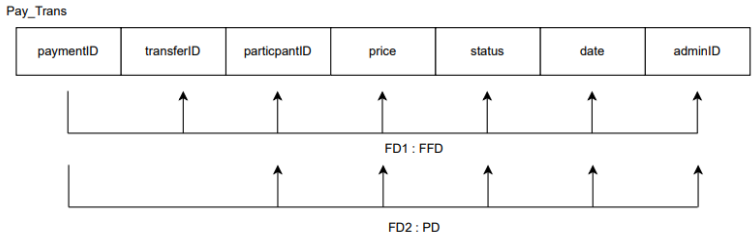
FK : adminID reference Administrator(adminID)

b) Pay_Card(paymentID,cardNum)

PK : paymentID

FK : paymentID reference Payment(paymentID)

4.3.12 **Normalization of Pay_Trans relationship**



FD1 : paymentID →transferID,participantID,price,status,date,adminID

FD2 : paymentID→participantID,price,status,date,adminID

1NF : Pay_Trans(paymentID,transferID,participantID,price,status,date,adminID)

PK : paymentID

FK : participantID reference Participant(participantID)

FK : adminID reference Administrator(adminID)

2NF : a) Payment(paymentID,participantID,price,status,date,adminID)

PK : paymentID

FK : participantID reference Participant(participantID)

FK : adminID reference Administrator(adminID)

b) Pay_trans(paymentID,transferID)

PK : paymentID

FK : paymentID reference Payment(paymentID)

3NF : a) Payment(paymentID,participantID,price,status,date,adminID)

PK : paymentID

FK : participantID reference Participant(participantID)

FK : adminID reference Administrator(adminID)

b) Pay_trans(paymentID,transferID)

PK : paymentID

FK : paymentID reference Payment(paymentID)

5.0 Relational DB Schemas (after normalization)

- Organizer(organizerID,name,email,eventID,adminID)
- Event(eventID,title,description,date,categoryType,setTicket)
- Administrator(adminID ,name,role)
- Ticket(ticketID,title,price,status,adminID)
- Payment(paymentID,price,status,date,participantID,adminID)
- Participant(participantID,name,phone,email,eventID)
- Ev_En(eventID,activity,type)
- Ev_Ed(eventID,place,topic)
- Par_Pub(participantID,normalPrice)
- Par_Stud(participantID,studID,studPrice)
- Pay_Card(paymentID,cardNum)
- Pay_trans(paymentID,transferID)
- Booking(ticketID,participantID)
- Receipt(paymentID,participantID)
- Pay(paymentID,participantID)
- Register(eventID,participantID)
- Telephone(phone,organizerID)

6.0 SQL Statements (DDL & DML)

6.1 Creating table

Table Organizer

```
CREATE TABLE Organizer (
        OrganizerID NUMBER (5),
        Name VARCHAR2 (30),
        Email VARCHAR2 (50),
        EventID NUMBER (5),
        adminID NUMBER (5),
        CONSTRAINT organizer_pk PRIMARY KEY (OrganizerID),
        CONSTRAINT email_uk UNIQUE (Email),
        CONSTRAINT event_fk FOREIGN KEY (EventID)
                REFERENCES Event (EventID),
        CONSTRAINT admin_fk FOREIGN KEY (adminID)
                        REFERENCES Administrator (adminID)
        );
```

Table Event

```
CREATE TABLE Event (
EventID NUMBER (5),
    title VARCHAR2 (100),
 description VARCHAR2 (100),
 dateEvent DATE,
    categoryType VARCHAR2 (15),
       setTicket VARCHAR2 (100),
    CONSTRAINT event_pk PRIMARY KEY (EventID)
);
```

Table Administrator

```
CREATE TABLE Administrator (
    adminID NUMBER (5)
    name VARCHAR2 (30),
    role VARCHAR2 (20),
    CONSTRAINT admin_pk PRIMARY KEY (adminID)
  );
```

Table Ticket

```
CREATE TABLE Ticket (
        ticketID NUMBER (5),
        title VARCHAR2 (100),
        price NUMBER (4,2),
        status VARCHAR2(10),
        adminID NUMBER (5),
        CONSTRAINT ticket_pk PRIMARY KEY (ticketID),
        CONSTRAINT adminID_fk FOREIGN KEY (adminID)
            REFERENCES Administrator (adminID)
);
```

Table Payment

```
CREATE TABLE Payment (
        paymentID NUMBER (5),
        price NUMBER (4,2),
        status VARCHAR2 (10),
        dateEvent DATE,
        adminID NUMBER (5),
        participantID NUMBER (5),
        CONSTRAINT payment_pk PRIMARY KEY (paymentID),
        CONSTRAINT admin_id_fk FOREIGN KEY (adminID)
            REFERENCES Administrator (adminID),
        CONSTRAINT par_id_fk8 FOREIGN KEY (participantID)
            REFERENCES Participant (participantID);
```

);

Table EventEntertainment (Ev_En)

```
CREATE TABLE Ev_En (
        eventID NUMBER (5),
        activity VARCHAR2 (100),
        type VARCHAR2 (100),
        CONSTRAINT ev_en_pk PRIMARY KEY (eventID),
        CONSTRAINT ev_en_fk FOREIGN KEY (eventID)
                REFERENCES Event (eventID)
                );
```

Table EventEducation (Ev_Ed)

```
CREATE TABLE Ev_Ed (
        eventID NUMBER (5),
        place VARCHAR2 (100),
        topic VARCHAR2 (100),
        CONSTRAINT event_id_pk2 PRIMARY KEY (eventID),
        CONSTRAINT ev_en_fk2 FOREIGN KEY (eventID)
                REFERENCES Event (eventID)
);
```

Table ParticipantPublic(Par_Pub)

```
CREATE TABLE Par_Pub (
        participantID NUMBER (20),
        normalPrice NUMBER (4,2),
        CONSTRAINT par_pk PRIMARY KEY (participantID),
        CONSTRAINT par_fk FOREIGN KEY (participantID)
                REFERENCES Participant (participantID)
);
```

Table ParticipantStud(Par_Stud)

```
CREATE TABLE Par_Stud (
        participantID NUMBER (20),
        studID NUMBER (20),
        studPrice NUMBER (4,2),
        CONSTRAINT par_pk2 PRIMARY KEY (participantID),
        CONSTRAINT par_fk2 FOREIGN KEY (participantID)
                REFERENCES Participant (participantID)
```

```
);
```

Table PaymentCard (Pay_Card)

```
CREATE TABLE Pay_card (
        paymentID NUMBER (5),
        cardNum NUMBER (5),
        CONSTRAINT pay_pk2 PRIMARY KEY (paymentID),
        CONSTRAINT pay_fk FOREIGN KEY (paymentID)
                REFERENCES Payment (paymentID)
);
```

Table PaymentTransfer (Pay_Trans)

```
CREATE TABLE Pay_Trans(
        paymentID NUMBER (20),
        transferID NUMBER (20),
        CONSTRAINT pay_pk3 PRIMARY KEY (paymentID),
        CONSTRAINT pay_fk2 FOREIGN KEY (paymentID)
                REFERENCES Payment (paymentID)
);
```

Table Participant

```
CREATE TABLE Participant (
        participantID NUMBER (5),
        name VARCHAR2 (100),
        phone VARCHAR2 (100),
        email VARCHAR2 (100),
        eventID NUMBER (5),
        CONSTRAINT par_id_pk4 PRIMARY KEY (participantID),
        CONSTRAINT event_pk2 FOREIGN KEY (eventID)
                REFERENCES Event (eventID)
        CONSTRAINT email_uk4 UNIQUE (email)
);
```

Table Invoice

```
CREATE TABLE Invoice (
        ticketID NUMBER (5),
        participantID NUMBER (5),
        CONSTRAINT ticket_pk2 PRIMARY KEY (ticketID,participantID),
        CONSTRAINT ticket_fk FOREIGN KEY (ticketID)
                REFERENCES Ticket (ticketID),
        CONSTRAINT par_id_fk3 FOREIGN KEY (participantID)
                REFERENCES Participant (participantID)
```

```
);
```

Table Booking

```
CREATE TABLE Booking (
        ticketID NUMBER (5),
        participantID NUMBER (5),
        CONSTRAINT book_pk PRIMARY KEY (ticketID, participantID),
        CONSTRAINT ticket_fk2 FOREIGN KEY (ticketID)
                REFERENCES Ticket (ticketID),
        CONSTRAINT par_id_fk4 FOREIGN KEY (participantID)
                REFERENCES Participant (participantID)
);
```

Table Receipt

```
CREATE TABLE Receipt (
        paymentID NUMBER (5),
        participantID NUMBER (5),
        CONSTRAINT rec_pk PRIMARY KEY (paymentID, participantID),
        CONSTRAINT pay_id_fk3 FOREIGN KEY (paymentID)
        REFERENCES Payment (paymentID),
        CONSTRAINT par_id_fk5 FOREIGN KEY (participantID)
        REFERENCES Participant (participantID)
);
```

Table Pay

```
CREATE TABLE Pay (
        paymentID NUMBER (5),
        participantID NUMBER (5),
        CONSTRAINT pay_pk PRIMARY KEY (paymentID, participantID),
        CONSTRAINT pay_id_fk4 FOREIGN KEY (paymentID)
        REFERENCES Payment (paymentID),
        CONSTRAINT par_id_fk6 FOREIGN KEY (participantID)
        REFERENCES Participant (participantID)
);
```

Table Register

```
CREATE TABLE Register (
        eventID NUMBER (5),
        participantID NUMBER (5),
        CONSTRAINT reg_pk PRIMARY KEY (eventID, participantID),
        CONSTRAINT event_id_fk3 FOREIGN KEY (eventID)
                REFERENCES Event (eventID),
        CONSTRAINT par_id_fk7 FOREIGN KEY (participantID)
```

```
REFERENCES Participant (participantID));
```

Table Telephone

```
CREATE TABLE Telephone (
        phone VARCHAR2 (100),
        organizerID NUMBER (5),
        CONSTRAINT phone_pk PRIMARY KEY (phone),
        CONSTRAINT org_id_fk FOREIGN KEY (organizerID)
        REFERENCES Organizer (organizerID)
);
```

6.2 Inserting value in tables

Insert data for Administrator

```
INSERT INTO Administrator (adminID, name, role)
        VALUES(3001,  'Scarlett Johansson',  'Project Manager');
        INSERT INTO Administrator (adminID, name, role)
        VALUES(3002,  'Chris Evans',  'Lead Software Developer');
        INSERT INTO Administrator (adminID, name, role)
        VALUES(3003,  'Priyanka Chopra',  'UX/UI Designer');
        INSERT INTO Administrator (adminID, name, role)
        VALUES(3004,  'Tom Hanks',  'Quality Assurance Tester');
        INSERT INTO Administrator (adminID, name, role)
        VALUES(3005,  'Zendaya',  'Scrum Master');
        INSERT INTO Administrator (adminID, name, role)
        VALUES(3006,  'Dwayne Johnson',  'DevOps Engineer');
        INSERT INTO Administrator (adminID, name, role)
        VALUES(3007,  'Emma Watson',  'Business Analyst');
        INSERT INTO Administrator (adminID, name, role)
```

```
VALUES(3008,  'Ryan Reynolds',  'Systems Architect');

INSERT INTO Administrator (adminID, name, role)

VALUES(3009,  'Gal Gadot',  'Database Administrator');

INSERT INTO Administrator (adminID, name, role)

VALUES(3010,  'Idris Elba',  'Mobile App Developer');

INSERT INTO Administrator (adminID, name, role)

VALUES(3011,  'Margot Robbie',  'Frontend Developer');

INSERT INTO Administrator (adminID, name, role)

VALUES(3012,  'Keanu Reeves',  'Backend Developer');

INSERT INTO Administrator (adminID, name, role)

VALUES(3013,  'Jennifer Lawrence',  'Technical Writer');

INSERT INTO Administrator (adminID, name, role)

VALUES(3014,  'Robert Downey Jr.',  'Software Engineer Intern');

INSERT INTO Administrator (adminID, name, role)

VALUES(3015,  'Charlize Theron',  'Release Manager');

INSERT INTO Administrator (adminID, name, role)

VALUES(3016,  'Tom Holland',  'AI/Machine Learning Specialist');

INSERT INTO Administrator (adminID, name, role)

VALUES(3017,  'Anne Hathaway',  'Cybersecurity Analyst');

INSERT INTO Administrator (adminID, name, role)

VALUES(3018,  'Will Smith',  'DevSecOps Engineer');

INSERT INTO Administrator (adminID, name, role)

VALUES(3019,  'Zoe Saldana',  'Cloud Solutions Architect');
```

```
INSERT INTO Administrator (adminID, name, role)
VALUES(3020,  'Daniel Radcliffe',  'Full Stack Developer');
```

Insert data for Organizer

```
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1001,  'John Smith',  'john.smith@email.com',  2001,  3001);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1002,  'Emily Johnson',  'emily.johnson@email.com',  2002,  3002);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1003,  'Michael Brown',  'michael.brown@email.com',  2003,  3003);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1004,  'Sarah Davis',  'sarah.davis@email.com',  2004,  3004);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1005,  'Kevin Lee',  'kevin.lee@email.com',  2005,  3005);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1006,  'Jessica White',  'jessica.white@email.com',  2006,  3006);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1007,  'Brian Taylor',  'brian.taylor@email.com',  2007,  3007);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1008,  'Olivia Moore',  'olivia.moore@email.com',  2008,  3008);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1009,  'James Miller',  'james.miller@email.com',  2009,  3009);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
```

```
VALUES(1010, 'Ashley Wilson', 'ashley.wilson@email.com', 2010, 3010);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1011, 'David Hall', 'david.hall@email.com', 2011, 3011);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1012, 'Emma Carter', 'emma.carter@email.com', 2012, 3012);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1013, 'Christopher Davis', 'christopher.davis@email.com', 2013, 3013)
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1014, 'Jennifer Reed', 'jennifer.reed@email.com', 2014, 3014);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1015, 'Robert Garcia', 'robert.garcia@email.com', 2015, 3015);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1016, 'Natalie Johnson', 'natalie.johnson@email.com', 2016, 3016);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1017, 'Daniel Brown', 'daniel.brown@email.com', 2017, 3017);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1018, 'Megan Taylor', 'megan.taylor@email.com', 2018, 3018);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1019, 'Nicholas Smith', 'nicholas.smith@email.com', 2019, 3019);
INSERT INTO Organizer (OrganizerID, Name, Email, EventID, adminID)
VALUES(1020, 'Lauren White', 'lauren.white@email.com', 2020, 3020);
```

Insert data for Ticket

```
INSERT INTO Ticket (TicketID, title, price, status, adminID)
VALUES(4001, 'Tech Expo 2024', 15.25, 'Draft', 3001);
INSERT INTO Ticket (TicketID, title, price, status, adminID)
VALUES(4002, 'Jazz Night Concert', 42.5, 'Draft', 3002);
INSERT INTO Ticket (TicketID, title, price, status, adminID)
VALUES(4003, 'Wellness Workshop', 76.8, 'Draft', 3003);
INSERT INTO Ticket (TicketID, title, price, status, adminID)
VALUES(4004, 'Art Exhibition', 33.75, 'Draft', 3004);
INSERT INTO Ticket (TicketID, title, price, status, adminID)
VALUES(4005, 'Science Fair', 64.2, 'Draft', 3005);
INSERT INTO Ticket (TicketID, title, price, status, adminID)
VALUES(4006, 'Charity Gala', 21.4, 'Pending', 3006);
INSERT INTO Ticket (TicketID, title, price, status, adminID)
VALUES(4007, 'Movie Night Under the Stars', 95.6, 'Pending', 3007);
INSERT INTO Ticket (TicketID, title, price, status, adminID)
VALUES(4008, 'Business Networking Mixer', 12.7, 'Pending', 3008);
INSERT INTO Ticket (TicketID, title, price, status, adminID)
VALUES(4009, 'Sustainable Living Seminar', 54.9, 'Pending', 3009);
INSERT INTO Ticket (TicketID, title, price, status, adminID)
VALUES(4010, 'Cooking Class: International Cuisine', 88.1, 'Pending', 3010);
INSERT INTO Ticket (TicketID, title, price, status, adminID)
```

```
VALUES(4011, 'Book Club Meeting', 18.3, 'Approved', 3011);

INSERT INTO Ticket (TicketID, title, price, status, adminID)

VALUES(4012, 'Fitness Challenge', 72.4, 'Approved', 3012);

INSERT INTO Ticket (TicketID, title, price, status, adminID)

VALUES(4013, 'Photography Workshop', 46.6, 'Approved', 3013,);

INSERT INTO Ticket (TicketID, title, price, status, adminID)

VALUES(4014, 'Career Development Seminar', 29.9, 'Approved', 3014);

INSERT INTO Ticket (TicketID, title, price, status, adminID)

VALUES(4015, 'Comedy Night', 57.3, 'Approved', 3015);

INSERT INTO Ticket (TicketID, title, price, status, adminID)

VALUES(4016, 'Fashion Show', 82.15, 'Rejected', 3016);

INSERT INTO Ticket (TicketID, title, price, status, adminID)

VALUES(4017, 'Gardening Workshop', 38.05, 'Rejected', 3017);

INSERT INTO Ticket (TicketID, title, price, status, adminID)

VALUES(4018, 'Cultural Diversity Celebration', 66.75, 'Rejected', 3018);

INSERT INTO Ticket (TicketID, title, price, status, adminID)

VALUES(4019, 'Virtual Reality Experience', 10, 'Rejected', 3019);

INSERT INTO Ticket (TicketID, title, price, status, adminID)

VALUES(4020, 'TEDx Talk: Ideas Worth Spreading', 91.45, 'Rejected', 3020);
```

Insert data for Payment

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID, participantID)

VALUES(5001, 15.25, 'Paid', TO_DATE('12/10/2023','MM-DD-YYYY'), 3001, 6001);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5002, 42.5, 'Paid', TO_DATE('08/25/2023','MM-DD-YYYY'), 3002, 6002);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5003, 76.8, 'Paid', TO_DATE('03/15/2023','MM-DD-YYYY'), 3003, 6003);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5004, 33.75, 'Paid', TO_DATE('06/02/2023','MM-DD-YYYY'), 3004, 6004);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5005, 64.2, 'Paid', TO_DATE('11/05/2023','MM-DD-YYYY'), 3005, 6005);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5006, 21.4, 'Paid', TO_DATE('09/18/2023','MM-DD-YYYY'), 3006, 6006);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5007, 95.6, 'Paid', TO_DATE('07/14/2023','MM-DD-YYYY'), 3007, 6007);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5008, 12.7, 'Paid', TO_DATE('04/30/2023','MM-DD-YYYY'), 3008, 6008);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5009, 54.9, 'Paid', TO_DATE('02/12/2023','MM-DD-YYYY'), 3009, 6009);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5010, 88.1, 'Paid', TO_DATE('10/20/2023','MM-DD-YYYY'), 3010, 6010);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5011, 18.3, 'Paid', TO_DATE('05/08/2023','MM-DD-YYYY'), 3011, 6011);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5012, 72.4, 'Not Paid', TO_DATE('07/01/2023','MM-DD-YYYY'), 3012, 6012);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5013, 46.6, 'Not Paid', TO_DATE('09/22/2023','MM-DD-YYYY'), 3013, 6013);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5014, 29.9, 'Not Paid', TO_DATE('11/13/2023','MM-DD-YYYY'), 3014, 6014);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5015, 57.3, 'Not Paid', TO_DATE('04/15/2023','MM-DD-YYYY'), 3015, 6015);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5016, 82.15, 'Not Paid', TO_DATE('08/08/2023','MM-DD-YYYY'), 3016, 6016);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5017, 38.05, 'Not Paid', TO_DATE('06/28/2023','MM-DD-YYYY'), 3017, 6017);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5018, 66.75, 'Not Paid', TO_DATE('03/05/2023','MM-DD-YYYY'), 3018, 6018);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5019, 10, 'Not Paid', TO_DATE('10/10/2023','MM-DD-YYYY'), 3019, 6019);

INSERT INTO Payment (paymentID, price, status, dateEvent, adminID)

VALUES(5020, 91.45, 'Not Paid', TO_DATE('09/05/2023','MM-DD-YYYY'), 3020, 6020a);

Insert data for Participant

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6001, 'Amina Ahmed', '019-123-456789', 'amina.ahmed@email.com', 2001);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6002, 'Yusuf Khan', '019-456-123456', 'yusuf.khan@email.com', 2001);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6003, 'Fatima Ali', '019-789-012345', 'fatima.ali@email.com', 2001);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6004, 'Mohammad Hassan', '019-876-543210', 'mohammad.hassan@email.com', 2004);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6005, 'Aisha Malik', '019-223-344556', 'aisha.malik@email.com', 2004);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6006, 'Ahmed Rahman', '019-499-887766', 'ahmed.rahman@email.com', 2004);

```
INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6007, 'Zainab Abbas', '019-777-889990', 'zainab.abbas@email.com', 2004);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6008, 'Bilal Ahmed', '019-556-667788', 'bilal.ahmed@email.com', 2004);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6009, 'Maryam Shah', '019-888-777665', 'maryam.shah@email.com', 2004);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6010, 'Ilyas Khan', '019-994-443322', 'ilyas.khan@email.com', 2010);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6011, 'Sumaya Ali', '019-888-777665', 'sumaya.ali@email.com', 2011);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6012, 'Faridah Rahman', '019-233-445566', 'faridah.rahman@email.com', 2012);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6013, 'Karim Khan', '019-556-667788', 'karim.khan@email.com', 2013);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6014, 'Sana Malik', '019-499-887766', 'sana.malik@email.com', 2014);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6015, 'Zayd Ahmed', '019-789-012345', 'zayd.ahmed@email.com', 2015);

INSERT INTO Participant (participantID, name, phone, email, eventID)
```

VALUES(6016, 'Aaliyah Hassan', '019-456-123456', 'aaliyah.hassan@email.com', 2016);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6017, 'Jamal Abbas', '019-123-456789', 'jamal.abbas@email.com', 2017);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6018, 'Sara Rahman', '019-888-777665', 'sara.rahman@email.com', 2018);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6019, 'Imran Shah', '019-556-667788', 'imran.shah@email.com', 2019);

INSERT INTO Participant (participantID, name, phone, email, eventID)

VALUES(6020, 'Khadijah Ali', '019-223-344556', 'khadijah.ali@email.com', 2020);

Insert data for Ev_En

INSERT INTO Ev_En (eventID, activity, type)

VALUES (2002, 'Jazz Performance', 'Concert/Music');

INSERT INTO Ev_En (eventID, activity, type)

VALUES (2004, 'Art Display', 'Exhibition');

INSERT INTO Ev_En (eventID, activity, type)

VALUES (2006, 'Philanthropic Evening', 'Gala/Entertainment');

INSERT INTO Ev_En (eventID, activity, type)

VALUES (2007, 'Outdoor Cinema', 'Movie Night');

INSERT INTO Ev_En (eventID, activity, type)

VALUES (2016, 'Runway Display', 'Fashion Show');

```
INSERT INTO Ev_En (eventID, activity, type)
VALUES (2018,  'Multicultural Activities',  'Celebration');
INSERT INTO Ev_En (eventID, activity, type)
VALUES (2019,  'VR Adventures',  'Technology Experience');
```

Insert data for Ev_Ed

```
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2001,  'Convention Center',  'Technology Showcase');
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2003,  'Community Center',  'Holistic Well-being');
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2005,  'School Auditorium',  'Hands-On Discovery');
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2008,  'Hotel Conference Room',  'Professional Networking');
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2009,  'Environmental Center',  'Eco-Friendly Practices');
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2010,  'Culinary School',  'Global Culinary Journey');
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2011,  'Local Library',  'Literary Discussions');
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2012,  'Outdoor Park',  'Physical Fitness');
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2013,  'Art Studio',  'Visual Storytelling');
```

```
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2014,  'Corporate Office',  'Professional Growth');
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2017,  'Botanical Garden',  'Green Thumb Tips');
INSERT INTO Ev_Ed (eventID, place, topic)
VALUES(2020,  'Performing Arts Center',  'Inspirational Ideas');
```

Insert data for Par_Pub

```
INSERT INTO Par_Pub (participantID, normalPrice)
VALUES (6001,  15.25);
INSERT INTO Par_Pub (participantID, normalPrice)
VALUES (6002,  42.5);
INSERT INTO Par_Pub (participantID, normalPrice)
VALUES (6003,  76.8);
INSERT INTO Par_Pub (participantID, normalPrice)
VALUES (6004,  33.75);
INSERT INTO Par_Pub (participantID, normalPrice)
VALUES (6005,  64.2);
INSERT INTO Par_Pub (participantID, normalPrice)
VALUES (6006,  21.4);
INSERT INTO Par_Pub (participantID, normalPrice)
VALUES (6007,  95.6);
INSERT INTO Par_Pub (participantID, normalPrice)
```

```
VALUES (6008,  12.7);
INSERT INTO Par_Pub (participantID, normalPrice)
VALUES (6009,  54.9);
INSERT INTO Par_Pub (participantID, normalPrice)
VALUES (6010,  88.1);
```

Insert data for Par_Stud

```
INSERT INTO Par_Stud (participantID, studID, studPrice)
VALUES (6011,  7001,  9.15);
INSERT INTO Par_Stud (participantID, studID, studPrice)
VALUES (6012,  7002,  36.2);
INSERT INTO Par_Stud (participantID, studID, studPrice)
VALUES (6013,  7003,  23.3);
INSERT INTO Par_Stud (participantID, studID, studPrice)
VALUES (6014,  7004,  14.95);
INSERT INTO Par_Stud (participantID, studID, studPrice)
VALUES (6015,  7005,  28.65);
INSERT INTO Par_Stud (participantID, studID, studPrice)
VALUES (6016,  7006,  41.10);
INSERT INTO Par_Stud (participantID, studID, studPrice)
VALUES (6017,  7007,  19.05);
INSERT INTO Par_Stud (participantID, studID, studPrice)
VALUES (6018,  7008,  33.40);
```

```
INSERT INTO Par_Stud (participantID, studID, studPrice)
VALUES (6019,  7009,  5.00);
INSERT INTO Par_Stud (participantID, studID, studPrice)
VALUES (6020,  7010,  45.75);
```

Insert data for Pay_Card

```
INSERT INTO Pay_Card (paymentID, cardNum)
VALUES (5001,  131);
INSERT INTO Pay_Card (paymentID, cardNum)
VALUES (5002,  178);
INSERT INTO Pay_Card (paymentID, cardNum)
VALUES (5003,  115);
INSERT INTO Pay_Card (paymentID, cardNum)
VALUES (5004,  192);
INSERT INTO Pay_Card (paymentID, cardNum)
VALUES (5005,  104);
INSERT INTO Pay_Card (paymentID, cardNum)
VALUES (5006,  159);
INSERT INTO Pay_Card (paymentID, cardNum)
VALUES (5007,  120);
INSERT INTO Pay_Card (paymentID, cardNum)
VALUES (5008,  197);
INSERT INTO Pay_Card (paymentID, cardNum)
```

```
VALUES (5009,    144);

INSERT INTO Pay_Card (paymentID, cardNum)

VALUES (5010,    110);

INSERT INTO Pay_Card (paymentID, cardNum)

VALUES (5011,    183);
```

Insert data for Pay_Trans

```
INSERT INTO Pay_Trans (paymentID, transferID)

VALUES (5012,    8001);

INSERT INTO Pay_Trans (paymentID, transferID)

VALUES (5013,    8002);

INSERT INTO Pay_Trans (paymentID, transferID)

VALUES (5014,    8003);

INSERT INTO Pay_Trans (paymentID, transferID)

VALUES (5015,    8004);

INSERT INTO Pay_Trans (paymentID, transferID)

VALUES (5016,    8005);

INSERT INTO Pay_Trans (paymentID, transferID)

VALUES (5017,    8006);

INSERT INTO Pay_Trans (paymentID, transferID)

VALUES (5018,    8007);

INSERT INTO Pay_Trans (paymentID, transferID)

VALUES (5019,    8008);
```

```
INSERT INTO Pay_Trans (paymentID, transferID)
VALUES (5020,    8009);
```

Insert data for Telephone

```
INSERT INTO Telephone (phone, organizerID)
VALUES ('014-567-8901',    1001);
INSERT INTO Telephone (phone, organizerID)
VALUES ('018-543-2109',    1002);
INSERT INTO Telephone (phone, organizerID)
VALUES ('016-789-0123',    1003);
INSERT INTO Telephone (phone, organizerID)
VALUES ('013-210-9876',    1004);
INSERT INTO Telephone (phone, organizerID)
VALUES ('010-987-6543',    1005);
INSERT INTO Telephone (phone, organizerID)
VALUES ('015-432-1098',    1006);
INSERT INTO Telephone (phone, organizerID)
VALUES ('017-654-3210',    1007);
INSERT INTO Telephone (phone, organizerID)
VALUES ('012-345-6789',    1008);
INSERT INTO Telephone (phone, organizerID)
VALUES ('011-223-4455',    1009);
INSERT INTO Telephone (phone, organizerID)
```

```
VALUES ('019-876-5432',    1010);

INSERT INTO Telephone (phone, organizerID)

VALUES ('018-534-2100',    1011);

INSERT INTO Telephone (phone, organizerID)

VALUES ('013-454-3210',    1012);

INSERT INTO Telephone (phone, organizerID)

VALUES ('012-445-6789',    1013);

INSERT INTO Telephone (phone, organizerID)

VALUES ('014-567-8001',    1014);

INSERT INTO Telephone (phone, organizerID)

VALUES ('017-789-0123',    1015);

INSERT INTO Telephone (phone, organizerID)

VALUES ('015-442-1098',    1016);

INSERT INTO Telephone (phone, organizerID)

VALUES ('013-211-9876',    1017);

INSERT INTO Telephone (phone, organizerID)

VALUES ('012-987-6543',    1018);

INSERT INTO Telephone (phone, organizerID)

VALUES ('011-223-4655',    1019);

INSERT INTO Telephone (phone, organizerID)

VALUES ('018-876-5432',    1020);
```

Insert data for Register

```
INSERT INTO REGISTER (eventID, participantID)
        VALUES (2001,    6001);
        INSERT INTO REGISTER (eventID, participantID)
        VALUES (2001,    6002);
        INSERT INTO REGISTER (eventID, participantID)
        VALUES (2001,    6003);
        INSERT INTO REGISTER (eventID, participantID)
        VALUES (2004,    6004);
        INSERT INTO REGISTER (eventID, participantID)
        VALUES (2004,    6005);
        INSERT INTO REGISTER (eventID, participantID)
        VALUES (2004,    6006);
        INSERT INTO REGISTER (eventID, participantID)
        VALUES (2004,    6007);
        INSERT INTO REGISTER (eventID, participantID)
        VALUES (2004,    6008);
        INSERT INTO REGISTER (eventID, participantID)
        VALUES (2004,    6009);
        INSERT INTO REGISTER (eventID, participantID)
        VALUES (2010,    6010);
        INSERT INTO REGISTER (eventID, participantID)
        VALUES (2011,    6011);
```

```
INSERT INTO REGISTER (eventID, participantID)
VALUES (2012,    6012);
INSERT INTO REGISTER (eventID, participantID)
VALUES (2013,    6013);
INSERT INTO REGISTER (eventID, participantID)
VALUES (2014,    6014);
INSERT INTO REGISTER (eventID, participantID)
VALUES (2015,    6015);
INSERT INTO REGISTER (eventID, participantID)
VALUES (2016,    6016);
INSERT INTO REGISTER (eventID, participantID)
VALUES (2017,    6017);
INSERT INTO REGISTER (eventID, participantID)
VALUES (2018,    6018);
INSERT INTO REGISTER (eventID, participantID)
VALUES (2019,    6019);
INSERT INTO REGISTER (eventID, participantID)
VALUES (2020,    6020);
```

Insert data for Pay

```
INSERT INTO Pay (paymentID, participantID)
VALUES (5001,    6001);
INSERT INTO Pay (paymentID, participantID)
VALUES (5002,    6002);
```

```
INSERT INTO Pay (paymentID, participantID)
VALUES (5003,    6003);
INSERT INTO Pay (paymentID, participantID)
VALUES (5004,    6004);
INSERT INTO Pay (paymentID, participantID)
VALUES (5005,    6005);
INSERT INTO Pay (paymentID, participantID)
VALUES (5006,    6006);
```

Insert data for Receipt

```
INSERT INTO Receipt (paymentID, participantID)
VALUES (5001,    6001);
INSERT INTO Receipt (paymentID, participantID)
VALUES (5002,    6002);
INSERT INTO Receipt (paymentID, participantID)
VALUES (5003,    6003);
INSERT INTO Receipt (paymentID, participantID)
VALUES (5004,    6004);
INSERT INTO Receipt (paymentID, participantID)
VALUES (5005,    6005);
INSERT INTO Receipt (paymentID, participantID)
VALUES (5006,    6006);
```

Insert data for Booking

```
INSERT INTO Booking (ticketID, participantID)
VALUES (4001,    6001);
INSERT INTO Booking (ticketID, participantID)
VALUES (4002,    6002);
INSERT INTO Booking (ticketID, participantID)
VALUES (4003,    6003);
INSERT INTO Booking (ticketID, participantID)
VALUES (4004,    6004);
INSERT INTO Booking (ticketID, participantID)
VALUES (4005,    6005);
INSERT INTO Booking (ticketID, participantID)
VALUES (4006,    6006);
```

Insert data for Invoice

```
INSERT INTO Invoice (ticketID, participantID)
VALUES (4001,    6001);
INSERT INTO Invoice (ticketID, participantID)
VALUES (4002,    6002);
INSERT INTO Invoice (ticketID, participantID)
VALUES (4003,    6003);
INSERT INTO Invoice (ticketID, participantID)
VALUES (4004,    6004);
```

```
INSERT INTO Invoice (ticketID, participantID)

VALUES (4005,    6005);

INSERT INTO Invoice (ticketID, participantID)

VALUES (4006,    6006);
```

7.0 Summary

Our team has successfully finished the database conceptual design in a critical stage of the NexScholar project. This includes revised business rules, a refined conceptual Entity-Relationship Diagram (ERD) and an improved Data Dictionary. Normalisation is included into the logical architecture to facilitate efficient data organization that leading to relational database schemas that are optimized. Interestingly, the TO-BE system fills a need in the market by introducing a booking and payment method. Customers can now book tickets with ease and the system makes sure that the payment process goes well and that tickets are issued with unique IDs at the end. For effective tracking, every data entity including administrators, organizers, and events has a unique identification. Finally, SQL statements (DDL & DML) round out this work and make it easier to put these improvements into practice at the same time laying the groundwork for a reliable and intuitive NexScholar platform.