UTM
UNIVERSITI TEKNOLOGI MALAYSIA
Department of Computer Science
Faculty of Computing

GROUP PROJECT

Phase 3

Database Logical Design

Programme	:	Bachelor of Computer Science (*Computer Network and Security*)
Subject Code	:	SECD2523
Subject Name	:	DATABASE
Session-Sem	:	2023/2024-1

Prepared by	:	1) Aidan Andrew Daniel (A22EC0036) 2) Muhammad Amirul Azfar bin Azhar (A22EC0080) 3) Danial Nur Irfan bin Azeze (A22EC0046) 4) Muhammad Haziq bin Saamsol (A22EC0214) 5) Muhammad Nabil Iman bin Mohd Fazli (A22EC0087)
Section	:	03
Lecturer	:	Dr. Izyan Izzati binti Kamsani

Table of Content

1.0 Introduction

In the dynamic realm of information technology, crafting an effective and streamlined database is essential, especially when dealing with intricate systems like the Event Management System on the Nexscholar website. As we move forward with Phase 3 of the Nexscholar Event Management System project, we are more concerned with the database's logical design. In order to turn our conceptual blueprint into a solid and useful database structure, we must complete this phase, we are required to create a database to visualize our logical design.

During this stage, we focus on improving and honing the conceptual design. To make sure the business rules are in line with the changing project dynamics, we start by going over and revising them. To capture the complex interactions between items, we further enhance the Conceptual ERD, a visual representation of our data model. The cornerstone of our logical design is the business rule, which serves as the driving basis for constructing our database. We adjust and update these guidelines as we go along to take into account any new or modified project requirements.

Expanding upon the Phase 2 conceptual design, we further explore the connections and components that serve as the framework for our ticketing system. In addition to summarising our system's current condition, the Conceptual ERD foresees future scalability, guaranteeing the longevity of our architecture. Phase 3 moves forward with the logical design of our database, having established a strong basis in the conceptual domain. To improve data integrity and decrease redundancy, this entails constructing a logical ERD, updating the data dictionary, and normalising the data. The conceptual representation is polished into a format that is ordered and structured by the Logical ERD. It acts as a link between the conceptual and practical designs, giving the database's architecture a more thorough plan.

The Data Dictionary, a valuable resource, is enhanced to provide in-depth understanding of every aspect of our logical structure. It turns into an essential point of reference for stakeholders and developers, guaranteeing a mutual comprehension of the database structure. A crucial first step in improving database efficiency is normalisation. In order to reduce dependencies and redundancies and create a more resilient and efficient database, we use normalisation techniques. The development of relational database schemas is the result of our

logical design process. The basis for building the actual database is provided by these schemas, which represent the normalised structure. The commands from the Data Definition Language (DDL) and Data Manipulation Language (DML) that realise our logical design are contained in the SQL Statements. In order to create and maintain the database, these statements are necessary.

Join us in this exciting phase as we transform abstract concepts into a tangible blueprint for the Nexscholar Event Management System. From requirements to conceptual design, this journey is crucial in realising a powerful and efficient ticket booking system that aligns seamlessly with the unique dynamics of our platform. Together, let's shape the future of event management at Nexscholar.

2.0 Overview of project

Phase 3 places a strong emphasis on developing the database conceptual design, building on the foundation created in the previous phases. This entails upgrading our conceptual Entity-Relationship Diagram (ERD), streamlining business processes, and setting up the foundation for the Logical ERD. The objective is to design a database structure that not only satisfies present needs but can easily grow and adapt to meet new ones in the future.

This phase involves upgrading the ERD, improving the project's business rules, and moving forward with the database's logical design. We review and revise business rules to make sure they are in line with changing project requirements. More refinement is applied to the Conceptual ERD, which now captures the current state of the system and anticipates future scalability. A crucial link between conception and execution is the logical design stage. In order to maximise effectiveness, maintain data integrity, and minimise redundancy, it entails constructing a logical ERD, updating the data dictionary, and using normalisation procedures.

Phase 3 prepares the groundwork for the actual deployment of our database by culminating in the production of relational database schemas and SQL statements (DDL & DML). This is a big step towards ensuring that our technology infrastructure not only satisfies but surpasses the constantly changing requirements of our platform, which will shape the future of event management at Nexscholar. Together, we continue Nexscholar's revolutionary journey into event management with the next chapter.

3.0 Database conceptual design

3.1 Updated business rule

1. Broad Features for Event Management:
    - Customisation: Event templates and details should be highly customisable by users, giving them complete control over:
        - Kind of event (webinar, workshop, symposium, etc.)
        - Type of audience: students, staff, teachers, etc.
        - Format of the event (in-person, virtual, hybrid)
        - Structure of sessions (one-time, multi-day, or recurrent)
        - Options for registration (free, premium, and tier-based pricing)
        - Management of speakers and trainers
        - Uploading and managing content (recordings, handouts, and presentations)
    - Capabilities for scheduling:
        - Adaptable event and session scheduling
        - Time zone coordination
        - Scheduling recurring events
    - Integration of a calendar Extra features:
        - Event waitlist features
        - Management of event capacity
        - Talk and live Q&A during events
        - Surveys and feedback gathering following the event
2. The Event Registration User Experience:

    - Specific details about the event: When consumers click on an event, they ought to see:
        - complete schedule and description of the event
        - Speaker/trainer biographies and credentials
        - Details about the venue (for in-person events)
        - Details about access (for online events)
        - List of materials (records, handouts)

- Easy registration procedure:
    - Options for registration based on the sort of event: free, premium, and tier pricing
    - a convenient and safe payment gateway
    - email confirmation that includes links to the event and details
    - Calendar integration for events that are registered.

3. Integrations of Systems:

    - Integrate permission and authentication with the current user management system.
    - Connect to a payment gateway to provide safe and easy transactions.
    - Connect calendar systems to schedule events and send out reminders.
    - Connect with messaging platforms to receive updates and information about events.

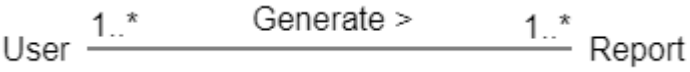
4. Analytics and Reporting:

    - Give statistics on registrations, attendance, interaction, and feedback to the event organisers.
    - Utilise data to monitor user preferences and behaviour in order to plan future events.

5. Accessibility and User Interface:

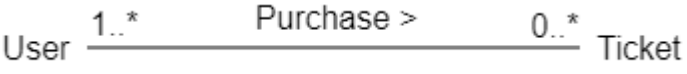
    - Provide an easy-to-use and intuitive interface for guests and event organisers alike.
    - Make sure people with disabilities can access the system.

The overall goal of these suggested business principles is to develop an all-inclusive and intuitive event management system that overcomes the existing drawbacks and gives attendees and organisers more authority. The expanded features, easy registration, system integrations, and data insights that this improved system offers can greatly enhance the user experience and increase the number of events that are utilised inside NexScholar.

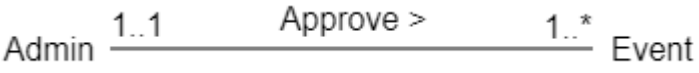


- Each User (event organizer only) can generate one or many reports.
- Each report can be generated from one or many users.

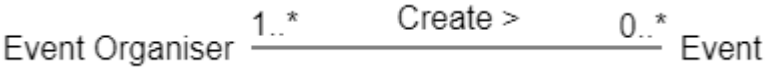
- Each user can join one or many events.
- Each event can be joined by one or many users.



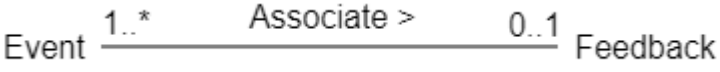
- Each user can purchase zero or many tickets.
- Each ticket can be purchased from one or many users.



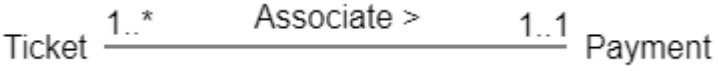
- Each admin can approve one or many events.
- Each event can be approved from one admin.



- Each event organiser can create zero or many events.
- Each event can be created from one or many event organisers.



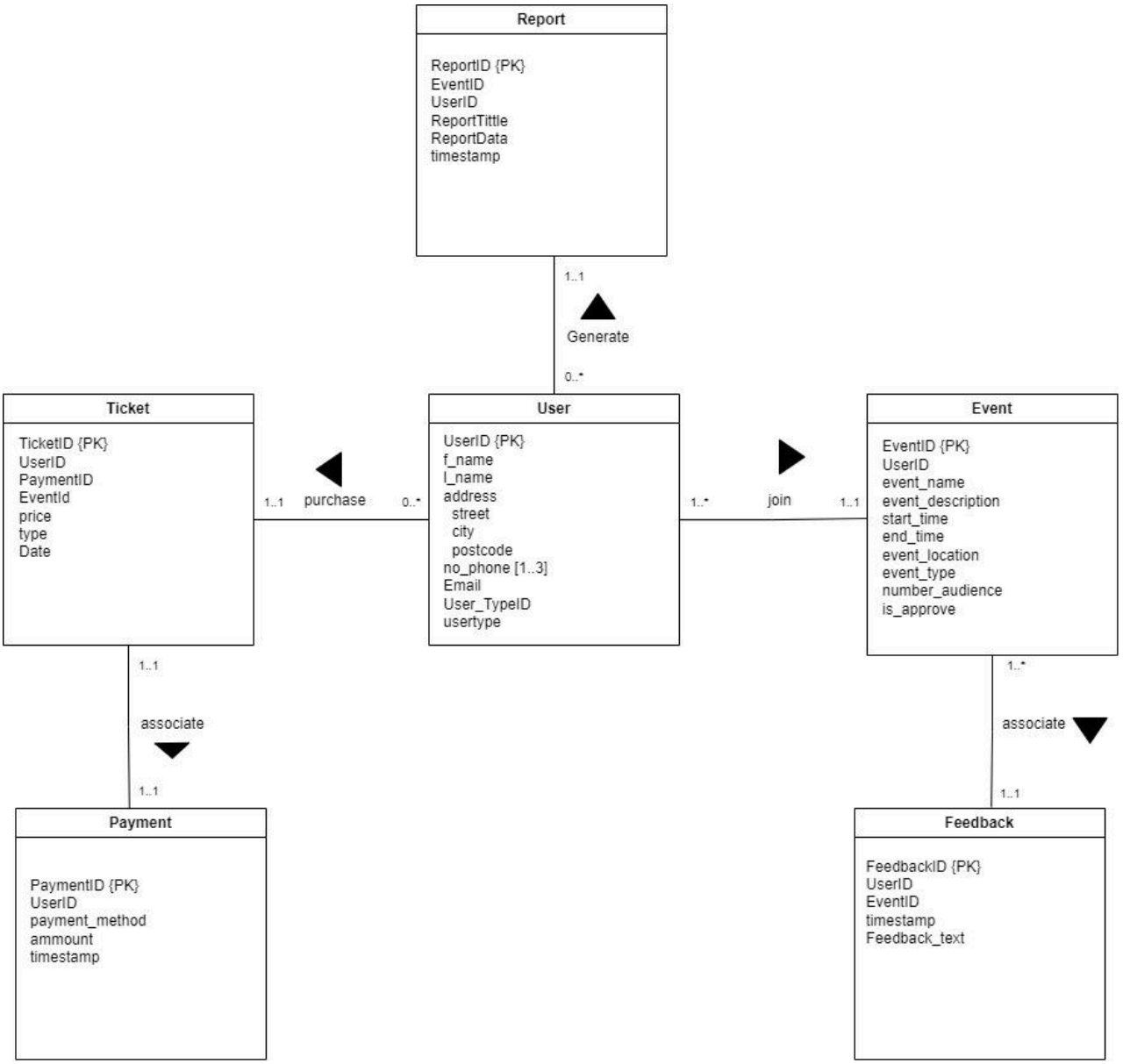
- Each event can associate zero or one feedback.
- Each feedback can be associated with one or many events.



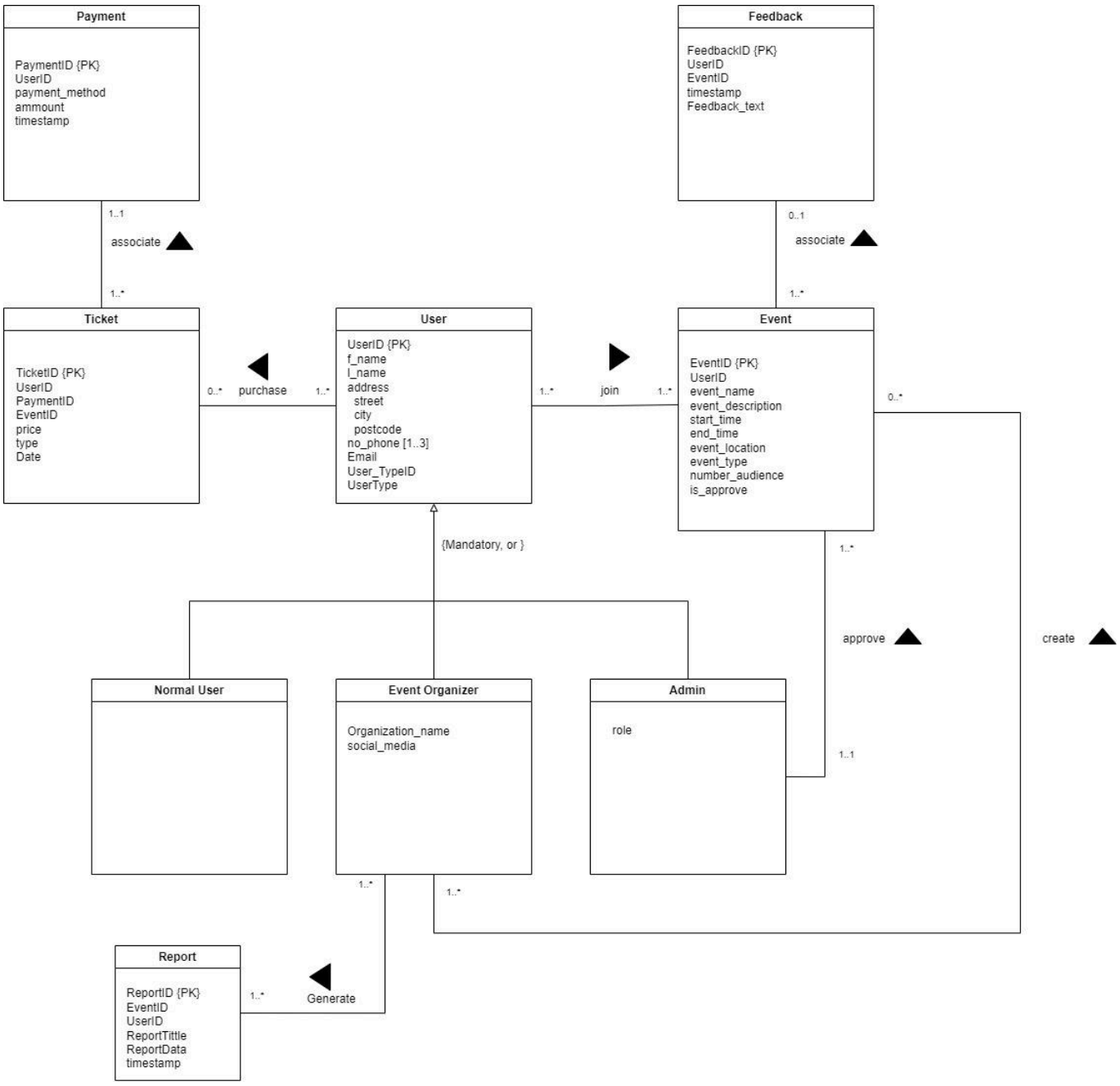
- Each ticket can associate one payment.
- Each payment can be associated with one or many tickets.

3.2 Conceptual ERD

3.2.1 Conceptual ERM

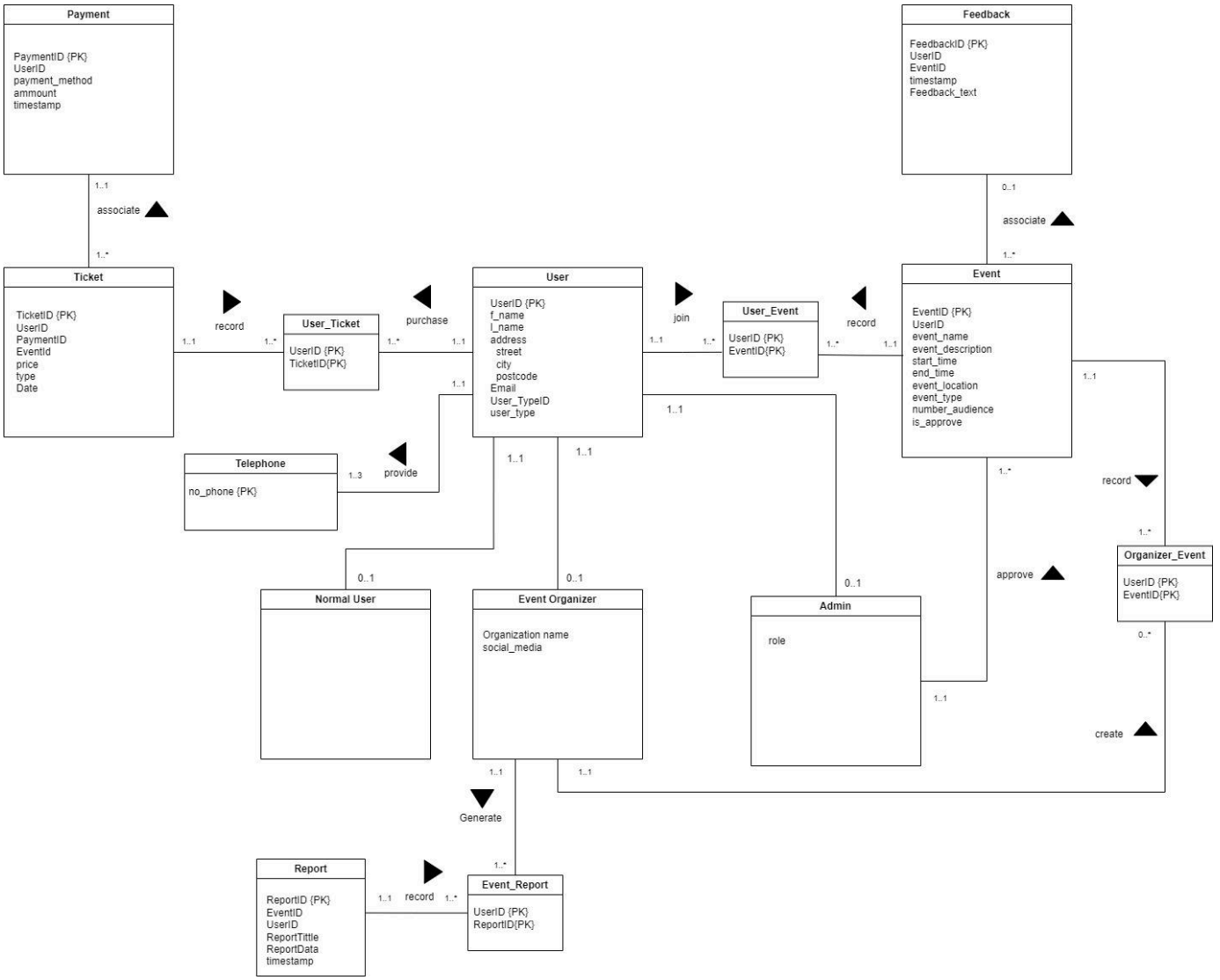


3.2.2 Enhanced ERD (EERD)



4.0 DB logical design

4.1 Logical ERD



4.2 Updated Data Dictionary

Entity Name	Attributes	Description	Data Type & Length	Null	Multi-Valued
User	userID	Unique user identification	5 variable char	No	No
	f_name	User first name	15 variable char	No	No
	l_name	User last name	15 variable char	No	No
	address				
	street	Address street	20 variable char	No	No
	city	Address city	20 variable char	No	No
	postcode	Address postcode	5 variable char	No	No
	no_phone	User phone number	10 variable char	No	Yes
	email	User email	20 variable char	No	No
	user_TypeID	Unique user type identification	10 variable char	No	No
	user_type	User type	10 variable char	No	No
Feedback	FeedBackID	Unique feedback identification	5 variable char	No	No
	userID	Unique user identification	5 variable char	No	No
	eventID	Unique event identification	5 variable char	No	No
	timestamp	Feedback timestamp	8 variable char	No	No
	Feedback_text	Feedback text content	100 variable char	No	No
Ticket	ticketID	Unique ticket identification	5 variable char	No	No
	userID	Unique user identification	5 variable char	No	No
	eventID	Unique event identification		No	No
	price	Ticket price	integer	No	No
	Date	Ticket purchase date	8 variable char	No	No
	type	Ticket type	5 variable char	No	No
Event	eventID	Unique event identification	5 variable char	No	No
	UserID	Unique user identification	5 variable char	No	No
	event_name	Event name	20 variable char	No	No
	event_description	Event description	100 variable char	Yes	No
	start_time	Event start time	8 variable char	No	No
	end_time	Event end time	8 variable char	No	No
	event_location	Event location	20 variable char	No	No
	event_type	Event type	10 variable char	No	No
	number_audience	Number of audience	integer	No	No
	is_approve	Event approval	1 character (Y or N)	No	No
Report	ReportID	Unique report identification	5 variable char	No	No
				No	No
	EventID	Unique event identification	5 variable char	No	No
				No	No
	UserID	Unique user identification	5 variable char	No	No
	ReportTitle	Report title	20 variable char	No	No
	ReportData	Report data	100 variable char	Yes	No
	timestamp	Report timestamp	8 variable char	No	No

Payment	PaymentID UserID payment_method amount timestamp	Payment unique identification User unique identification Payment method Payment amount Payment timestamp	5 variable char 5 variable char 10 variable char integer 8 variable char	No No No Yes No	No No No No No
User_Ticket	UserID TicketID	User unique identification Ticket unique identification	5 variable char 5 variable char	No No	No No
Telephone	no_phone	Phone number	10 variable char	No	No
User_Event	UserID EventID	User unique identification Event unique identification	5 variable char 5 variable char	No No	No No
Organizer_ Event	UserID EventID	User unique identification Event unique identification	5 variable char 5 variable char	No No	No No
Event Organizer	Organization name Social media	Name of organization Social media tags	20 variable char 10 variable char	No Yes	No Yes
Admin	Role	Admin role	10 variable char	No	No
Event_ Report	UserID ReportID	User unique identification Report unique identification	5 variable char 5 variable char	No No	No No

4.3 Normalization

Entity User

1 NF , 2 NF :

User (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID, usertype)

3NF :

User (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)

UserType (User_TypeID, userType)

Telephone (UserID, no_phone)

Entity Normal User

1 NF , 2 NF, 3NF :

Normal User (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)

Entity Admin

1 NF , 2 NF, 3NF :

Admin (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID, role)

Entity Event Organizer

1 NF , 2 NF, 3NF :

Event Organizer (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID, organization_name, social_media)

Entity Event

1 NF , 2 NF :

Event (EventID, UserID, Event_Name, Event_Description, start _time, end_time, Event_Location, Event_Type, Number_Audience, Is_Approve)

3 NF :

Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve)

EventTime (EventID, Start_time, End_Time)

Entity Feedback

1 NF , 2 NF, 3NF :

Feedback (FeedbackID, UserID, EventID, Feedback_text, Timestamp)

Entity Report

1 NF , 2 NF, 3NF :

Report (ReportID, EventID, UserID, ReportTittle, ReportData, Timestamp)

Entity Payment

1 NF :

Payment (PaymentID, UserID,payment_method, amount, timestamp)

2 NF, 3NF :

Payment (PaymentID, UserID, MethodID, amount, timestamp)

PaymentMethod (MethodID, payment_method)

Entity Ticket

1 NF , 2 NF :

Ticket (TicketID, UserID,PaymentID, EventID, price, type, Date)

3 NF :

Ticket (TicketID, UserID,PaymentID, EventID, Ticket_TypeID, price, Date)

TicketType (Ticket_TypeID, ticket_type)

Normalize Table

User Table

UserID	F_Name	L_Name	Address	City	State	Postcode	Email	UserTypeID

UserType Table

User_TypeID	userType

Normal User Table

UserID	F_Name	L_Name	Address	City	State	Postcode	Email	UserTypeID

Admin Table

UserID	F_Name	L_Name	Address	City	State	Postcode	Email	UserTypeID	role

Event Organizer Table

UserID	F_Name	L_Name	Address	City	State	Postcode	Email	User TypeID	organization_name	social_media

Event Table

EventID	UserID	Event_Name	Event_Description	Event_Location	Event_Type	Number_Audience	Is_Approve

EventTime Table

EventID	Start_time	End_time

Feedback Table

FeedbackID	UserID	EventID	Feedback_text	Timestamp

Report Table

ReportID	EventID	UserID	ReportTittle	ReportData	Timestamp

Payment Table

PaymentID	UserID	MethodID	amount	Timestamp

Payment Table

PaymentID	payment_method

Ticket Table

TicketID	UserID	PaymentID	EventID	Ticket_TypeID	price	Date

TicketType Table

Ticket_TypeID	ticket_type

Telephone Table

UserID	no_phone

5.0 Relational DB Schemas (after normalization)

- **User** (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
- **Telephone** (UserID, no_phone)
- **Normal User** (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID,)
- **Admin** (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID, role)
- **Event Organizer** (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID, Organization_Name, Social_Media)
- **UserType** (User_TypeID, userType)
- **Event** (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve)
- **EventTime** (EventID, Start_time, End_Time)
- **Feedback** (FeedbackID, UserID, EventID, Feedback_text, Timestamp)
- **Report** (ReportID, EventID, UserID, ReportTittle, ReportData, Timestamp)
- **Ticket** (TicketID, UserID,PaymentID, EventID, Ticket_TypeID, price, Date)
- **TicketType** (Ticket_TypeID, ticket_type)
- **Payment** (PaymentID, UserID, MethodID, amount, timestamp)
- **PaymentMethod** (MethodID, payment_method)

6.0 SQL Statements (DDL & DML)

```
CREATE TABLE UserType (
    UserTypeID NUMBER PRIMARY KEY,
    userType VARCHAR2(50) NOT NULL
);

CREATE TABLE Users (
    UserID NUMBER PRIMARY KEY,
    F_Name VARCHAR2(50),
    L_Name VARCHAR2(50),
    Address VARCHAR2(255),
    City VARCHAR2(50),
    State VARCHAR2(50),
    Postcode VARCHAR2(20),
    Email VARCHAR2(100),
    UserTypeID NUMBER,
    CONSTRAINT fk_UserType FOREIGN KEY (UserTypeID) REFERENCES
UserType(UserTypeID)
);

CREATE TABLE Telephone (
    UserID NUMBER,
    no_phone VARCHAR2(15),
    CONSTRAINT pk_Telephone PRIMARY KEY (UserID, no_phone),
    CONSTRAINT fk_Telephone_User FOREIGN KEY (UserID) REFERENCES
Users(UserID)
);

CREATE TABLE NormalUser (
    UserID NUMBER PRIMARY KEY,
    CONSTRAINT fk_NormalUser_User FOREIGN KEY (UserID) REFERENCES
Users(UserID)
);
```

```
CREATE TABLE Admin (
    UserID NUMBER PRIMARY KEY,
    Role VARCHAR2(50),
    CONSTRAINT fk_Admin_User FOREIGN KEY (UserID) REFERENCES Users(UserID)
);

CREATE TABLE EventOrganizer (
    UserID NUMBER PRIMARY KEY,
    OrganizationName VARCHAR2(100),
    Social_Media VARCHAR2(50),
    CONSTRAINT fk_EventOrganizer_User FOREIGN KEY (UserID) REFERENCES
Users(UserID)
);

CREATE TABLE Event (
    EventID NUMBER PRIMARY KEY,
    UserID NUMBER,
    Event_Description VARCHAR2(255),
    Event_Location VARCHAR2(100),
    Event_Type VARCHAR2(50),
    Number_Audience NUMBER,
    Is_Approve VARCHAR2(1),
    Event_Name VARCHAR2(100),
    CONSTRAINT fk_Event_User FOREIGN KEY (UserID) REFERENCES Users(UserID)
);

CREATE TABLE EventTime (
    EventID NUMBER PRIMARY KEY,
    Start_Time TIMESTAMP,
    End_Time TIMESTAMP,
    CONSTRAINT fk_EventTime_Event FOREIGN KEY (EventID) REFERENCES
Event(EventID)
);
```

```
CREATE TABLE Feedback (
   FeedbackID NUMBER PRIMARY KEY,
   UserID NUMBER,
   EventID NUMBER,
   Feedback_Text VARCHAR2(500),
   Timestamp TIMESTAMP,
   CONSTRAINT fk_Feedback_User FOREIGN KEY (UserID) REFERENCES
Users(UserID),
   CONSTRAINT fk_Feedback_Event FOREIGN KEY (EventID) REFERENCES
Event(EventID)
);

CREATE TABLE PaymentMethod (
   MethodID NUMBER PRIMARY KEY,
   Payment_Method VARCHAR2(50)
);

CREATE TABLE Payment (
   PaymentID NUMBER PRIMARY KEY,
   UserID NUMBER,
   MethodID NUMBER,
   Amount NUMBER,
   Timestamp TIMESTAMP,
   CONSTRAINT fk_Payment_User FOREIGN KEY (UserID) REFERENCES
Users(UserID),
   CONSTRAINT fk_Payment_Method FOREIGN KEY (MethodID) REFERENCES
PaymentMethod(MethodID)
);

CREATE TABLE TicketType (
   Ticket_TypeID NUMBER PRIMARY KEY,
   Ticket_Type VARCHAR2(50)
);
```

```
CREATE TABLE Ticket (
    TicketID NUMBER PRIMARY KEY,
    UserID NUMBER,
    PaymentID NUMBER,
    EventID NUMBER,
    Ticket_TypeID NUMBER,
    Price NUMBER(10, 2),
    Date_of_Purchase DATE,
    CONSTRAINT fk_Ticket_User FOREIGN KEY (UserID) REFERENCES Users(UserID),
    CONSTRAINT fk_Ticket_Payment FOREIGN KEY (PaymentID) REFERENCES
Payment(PaymentID),
    CONSTRAINT fk_Ticket_Event FOREIGN KEY (EventID) REFERENCES
Event(EventID),
    CONSTRAINT fk_Ticket_TicketType FOREIGN KEY (Ticket_TypeID) REFERENCES
TicketType(Ticket_TypeID)
);

CREATE TABLE Report (
    ReportID NUMBER PRIMARY KEY,
    EventID NUMBER,
    UserID NUMBER,
    ReportTitle VARCHAR2(100),
    ReportData CLOB,
    Timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT fk_Report_User FOREIGN KEY (UserID) REFERENCES
Users(UserID),
    CONSTRAINT fk_Report_Event FOREIGN KEY (EventID) REFERENCES
Event(EventID)
);
```

Insert Data

-- Inserting data into UserType table

INSERT INTO UserType (UserTypeID, userType) VALUES (1, 'Normal User');
INSERT INTO UserType (UserTypeID, userType) VALUES (2, 'Admin');
INSERT INTO UserType (UserTypeID, userType) VALUES (3, 'Event Organizer');

-- Inserting data into Users table
INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (101, 'John', 'Doe', '123 Main Street', 'Kuala Lumpur', 'Wilayah Persekutuan', '50200', 'john.doe@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (102, 'Jane', 'Smith', '456 Oak Avenue', 'Penang', 'Penang', '10300', 'jane.smith@email.com', 2);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (103, 'Ahmad', 'Rahman', '789 Pine Road', 'Johor Bahru', 'Johor', '80000', 'ahmad.rahman@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (104, 'Siti', 'Hassan', '321 Cedar Lane', 'Kota Kinabalu', 'Sabah', '88000', 'siti.hassan@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (105, 'David', 'Lim', '555 Maple Street', 'Ipoh', 'Perak', '30000', 'david.lim@email.com', 2);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)

```
VALUES (106, 'Aisha', 'Ng', '987 Birch Avenue', 'Shah Alam', 'Selangor', '40100',
'aisha.ng@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email,
UserTypeID)
VALUES (107, 'Michael', 'Wong', '222 Elm Road', 'George Town', 'Penang', '10200',
'michael.wong@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email,
UserTypeID)
VALUES (108, 'Emily', 'Tan', '777 Pine Lane', 'Kuching', 'Sarawak', '93000',
'emily.tan@email.com', 3);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email,
UserTypeID)
VALUES (109, 'Hassan', 'Ibrahim', '444 Cedar Avenue', 'Kuala Terengganu', 'Terengganu',
'20000', 'hassan.ibrahim@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email,
UserTypeID)
VALUES (110, 'Nina', 'Chong', '888 Oak Road', 'Petaling Jaya', 'Selangor', '46200',
'nina.chong@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email,
UserTypeID)
VALUES (111, 'Ravi', 'Kumar', '111 Pine Lane', 'Alor Setar', 'Kedah', '5000',
'ravi.kumar@email.com', 3);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email,
UserTypeID)
VALUES (112, 'Anita', 'Raj', '666 Maple Avenue', 'Miri', 'Sarawak', '98000',
'anita.raj@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email,
UserTypeID)
```

VALUES (113, 'Chen', 'Yong', '333 Birch Road', 'Seremban', 'Negeri Sembilan', '70200', 'chen.yong@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (114, 'Saravanan', 'Naidu', '999 Elm Avenue', 'Langkawi', 'Kedah', '7000', 'saravanan.naidu@email.com', 3);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (115, 'Farah', 'Ali', '777 Cedar Lane', 'Kuala Lumpur', 'Wilayah Persekutuan', '50400', 'farah.ali@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (116, 'Tengku', 'Aziz', '555 Pine Road', 'Johor Bahru', 'Johor', '80200', 'tengku.aziz@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (117, 'Lily', 'Chan', '222 Maple Avenue', 'Kota Bharu', 'Kelantan', '15000', 'lily.chan@email.com', 3);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (118, 'Arjun', 'Rajput', '888 Birch Road', 'Shah Alam', 'Selangor', '40200', 'arjun.rajput@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)
VALUES (119, 'Nor', 'Azizah', '444 Elm Lane', 'Kuantan', 'Pahang', '25000', 'nor.azizah@email.com', 1);

INSERT INTO Users (UserID, F_Name, L_Name, Address, City, State, Postcode, Email, UserTypeID)

VALUES (120, 'Wei', 'Liu', '666 Oak Road', 'George Town', 'Penang', '10100',
'wei.liu@email.com', 3);

-- Inserting data into Telephone table

INSERT INTO Telephone (UserID, no_phone) VALUES (101, '0123456789');
INSERT INTO Telephone (UserID, no_phone) VALUES (102, '0112345678');
INSERT INTO Telephone (UserID, no_phone) VALUES (103, '0198765432');
INSERT INTO Telephone (UserID, no_phone) VALUES (104, '0176543210');
INSERT INTO Telephone (UserID, no_phone) VALUES (105, '0167890123');
INSERT INTO Telephone (UserID, no_phone) VALUES (106, '0134567890');
INSERT INTO Telephone (UserID, no_phone) VALUES (107, '0109876543');
INSERT INTO Telephone (UserID, no_phone) VALUES (108, '0145678901');
INSERT INTO Telephone (UserID, no_phone) VALUES (109, '0156789012');
INSERT INTO Telephone (UserID, no_phone) VALUES (110, '0123456782');
INSERT INTO Telephone (UserID, no_phone) VALUES (111, '0112345671');
INSERT INTO Telephone (UserID, no_phone) VALUES (112, '0198765434');
INSERT INTO Telephone (UserID, no_phone) VALUES (113, '0176543230');
INSERT INTO Telephone (UserID, no_phone) VALUES (114, '0167890113');
INSERT INTO Telephone (UserID, no_phone) VALUES (115, '0134567290');
INSERT INTO Telephone (UserID, no_phone) VALUES (116, '0109816543');
INSERT INTO Telephone (UserID, no_phone) VALUES (117, '0125678901');
INSERT INTO Telephone (UserID, no_phone) VALUES (118, '0176789012');
INSERT INTO Telephone (UserID, no_phone) VALUES (119, '0123456783');
INSERT INTO Telephone (UserID, no_phone) VALUES (120, '0112345675');

```
-- Inserting data into NormalUser table
INSERT INTO NormalUser (UserID)
VALUES (101);

INSERT INTO NormalUser (UserID)
VALUES (103);

INSERT INTO NormalUser (UserID)
VALUES (104);

INSERT INTO NormalUser (UserID)
VALUES (106);

INSERT INTO NormalUser (UserID)
VALUES (107);

INSERT INTO NormalUser (UserID)
VALUES (109);

INSERT INTO NormalUser (UserID)
VALUES (110);

INSERT INTO NormalUser (UserID)
VALUES (112);

INSERT INTO NormalUser (UserID)
VALUES (113);

INSERT INTO NormalUser (UserID)
VALUES (115);

INSERT INTO NormalUser (UserID)
VALUES (116);

INSERT INTO NormalUser (UserID)
```

VALUES (118);

INSERT INTO NormalUser (UserID)
VALUES (119);

-- Inserting data into Admin table
INSERT INTO Admin (UserID, Role)
VALUES (102, 'Event Admin');

INSERT INTO Admin (UserID, Role)
VALUES (105, 'Technical Admin');

-- Inserting data into EventOrganizer table with popular OrganizationNames
INSERT INTO EventOrganizer (UserID, OrganizationName, Social_Media) VALUES (108, 'EventPro', 'Facebook');
INSERT INTO EventOrganizer (UserID, OrganizationName, Social_Media) VALUES (111, 'GlobalEvents', 'Instagram');
INSERT INTO EventOrganizer (UserID, OrganizationName, Social_Media) VALUES (114, 'EpicEntertainment', 'Twitter');
INSERT INTO EventOrganizer (UserID, OrganizationName, Social_Media) VALUES (117, 'PrimePromotions', 'LinkedIn');
INSERT INTO EventOrganizer (UserID, OrganizationName, Social_Media) VALUES (120, 'SpectacularShows', 'YouTube');

-- Inserting data into Event table

INSERT ALL

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (201, 101, 'Health and Wellness Seminar', 'Educational seminar on maintaining a healthy lifestyle.', 'Penang Medical Center', 'Health', 200, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (202, 113, 'Cultural Festival', 'A celebration of diverse cultures through music, dance, and food.', 'Merdeka Square, Kuala Lumpur', 'Cultural', 1000, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (203, 103, 'Startup Pitch Day', 'Entrepreneurs pitch their innovative startup ideas to potential investors.', 'Cyberjaya Innovation Hub', 'Business', 300, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (204, 104, 'Art Exhibition', 'Showcasing local and international artists masterpieces.', 'National Art Gallery, Kuala Lumpur', 'Art', 150, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (205, 107, 'Music Festival', 'Live performances from top local and international artists.', 'Sunway Lagoon, Selangor', 'Music', 800, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (206, 106, 'Green Living Expo', 'Promoting sustainable living practices and eco-friendly products.', 'Mid Valley Exhibition Center', 'Environment', 400, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (207, 107, 'Sports Carnival', 'Various sports competitions and activities for all ages.', 'Stadium Negara, Kuala Lumpur', 'Sports', 600, 'n')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (208, 108, 'Food Truck Festival', 'A culinary delight featuring a variety of food trucks and vendors.', 'Dataran Merdeka, Kuala Lumpur', 'Food', 250, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (209, 109, 'Science Fair', 'Interactive exhibits showcasing scientific advancements and experiments.', 'Petrosains, The Discovery Centre', 'Science', 180, 'n')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (210, 110, 'Fashion Show', 'Showcasing the latest trends in fashion and design.', 'KL Fashion Pavilion', 'Fashion', 350, 'n')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (211, 111, 'Film Festival', 'Screening of independent and international films.', 'Golden Screen Cinemas, Pavilion', 'Film', 200, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (212, 112, 'Education Expo', 'Explore educational opportunities and career paths.', 'Sunway Pyramid Convention Center', 'Education', 300, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (213, 113, 'Book Fair', 'A paradise for book lovers with a wide selection of books and literary events.', 'Times Square Mall, Kuala Lumpur', 'Literature', 120, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (214, 114, 'Technology Conference', 'Bringing together experts to discuss the latest trends in technology.', 'Berjaya Times Square Hotel', 'Technology', 400, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (215, 115, 'Dance Showcase', 'A mesmerizing showcase of various dance styles.', 'The Star Performing Arts Center', 'Performing Arts', 250, 'n')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (216, 116, 'Gaming Expo', 'Explore the world of gaming with exhibitions and tournaments.', 'Mid Valley Megamall', 'Gaming', 180, 'n')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (217, 117, 'Charity Run', 'A fundraising run for various charitable causes.', 'Lake Gardens, Kuala Lumpur', 'Charity', 500, 'n')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (218, 118, 'Travel Fair', 'Discover exciting travel destinations and deals.', 'KL Convention Center', 'Travel', 300, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (219, 119, 'Automotive Expo', 'Showcasing the latest trends in the automotive industry.', 'Malaysia International Trade and Exhibition Centre', 'Automotive', 400, 'y')

INTO Event (EventID, UserID, Event_Name, Event_Description, Event_Location, Event_Type, Number_Audience, Is_Approve) VALUES (220, 120, 'Tech Expo 2024', 'Annual technology exhibition showcasing the latest innovations.', 'KL Convention Center', 'Technology', 500, 'y')

SELECT 1 FROM dual;

-- Inserting data into EventTime table

```
INSERT  ALL
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (201,
TO_TIMESTAMP('2024-02-15 09:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-02-15 17:00:00', 'YYYY-MM-DD HH24:MI:SS'))
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (202,
TO_TIMESTAMP('2024-03-10 10:30:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-03-10 15:00:00', 'YYYY-MM-DD HH24:MI:SS'))
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (203,
TO_TIMESTAMP('2024-04-05 18:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-04-05 22:00:00', 'YYYY-MM-DD HH24:MI:SS'))
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (204,
TO_TIMESTAMP('2024-05-20 13:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-05-20 18:00:00', 'YYYY-MM-DD HH24:MI:SS'))
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (205,
TO_TIMESTAMP('2024-06-15 11:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-06-15 20:00:00', 'YYYY-MM-DD HH24:MI:SS'))
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (206,
TO_TIMESTAMP('2024-07-08 17:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-07-09 02:00:00', 'YYYY-MM-DD HH24:MI:SS'))
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (207,
TO_TIMESTAMP('2024-08-03 09:30:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-08-03 16:30:00', 'YYYY-MM-DD HH24:MI:SS'))
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (208,
TO_TIMESTAMP('2024-09-18 08:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-09-18 17:00:00', 'YYYY-MM-DD HH24:MI:SS'))
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (209,
TO_TIMESTAMP('2024-10-12 12:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-10-12 21:00:00', 'YYYY-MM-DD HH24:MI:SS'))
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (210,
TO_TIMESTAMP('2024-11-05 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-11-05 16:00:00', 'YYYY-MM-DD HH24:MI:SS'))
   INTO EventTime (EventID, Start_Time, End_Time) VALUES (211,
TO_TIMESTAMP('2024-12-20 19:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2024-12-20 23:00:00', 'YYYY-MM-DD HH24:MI:SS'))
```

```
    INTO EventTime (EventID, Start_Time, End_Time) VALUES (212,
TO_TIMESTAMP('2025-01-15 14:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2025-01-15 22:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO EventTime (EventID, Start_Time, End_Time) VALUES (213,
TO_TIMESTAMP('2025-02-28 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2025-02-28 16:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO EventTime (EventID, Start_Time, End_Time) VALUES (214,
TO_TIMESTAMP('2025-03-22 11:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2025-03-22 19:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO EventTime (EventID, Start_Time, End_Time) VALUES (215,
TO_TIMESTAMP('2025-04-10 09:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2025-04-10 18:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO EventTime (EventID, Start_Time, End_Time) VALUES (216,
TO_TIMESTAMP('2025-05-05 17:30:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2025-05-05 22:30:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO EventTime (EventID, Start_Time, End_Time) VALUES (217,
TO_TIMESTAMP('2025-06-20 12:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2025-06-20 20:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO EventTime (EventID, Start_Time, End_Time) VALUES (218,
TO_TIMESTAMP('2025-07-15 07:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2025-07-15 11:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO EventTime (EventID, Start_Time, End_Time) VALUES (219,
TO_TIMESTAMP('2025-08-08 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2025-08-08 17:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO EventTime (EventID, Start_Time, End_Time) VALUES (220,
TO_TIMESTAMP('2025-09-25 09:30:00', 'YYYY-MM-DD HH24:MI:SS'),
TO_TIMESTAMP('2025-09-25 16:30:00', 'YYYY-MM-DD HH24:MI:SS'))
SELECT 1 FROM dual;
```

-- Inserting data into Feedback table

```
INSERT ALL
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (1, 101, 201, 'Great seminar!', TO_TIMESTAMP('2024-02-15 12:30:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (2, 113, 202, 'Amazing cultural experience!', TO_TIMESTAMP('2024-03-10 15:45:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (3, 103, 203, 'Innovative startup ideas!', TO_TIMESTAMP('2024-04-05 20:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (4, 104, 204, 'Beautiful art pieces!', TO_TIMESTAMP('2024-05-20 16:30:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (5, 107, 205, 'Fantastic music performances!', TO_TIMESTAMP('2024-06-15 22:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (6, 106, 206, 'Loved the eco-friendly products!', TO_TIMESTAMP('2024-07-08 02:30:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (7, 107, 207, 'Great sports carnival!', TO_TIMESTAMP('2024-08-03 11:15:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (8, 108, 208, 'Delicious food truck options!', TO_TIMESTAMP('2024-09-18 18:45:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (9, 109, 209, 'Impressive science exhibits!', TO_TIMESTAMP('2024-10-12 21:30:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (10, 110, 210, 'Fashion show was fantastic!', TO_TIMESTAMP('2024-11-05 16:15:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES (11, 111, 211, 'Enjoyed the film festival!', TO_TIMESTAMP('2024-12-20 23:45:00', 'YYYY-MM-DD HH24:MI:SS'))
```

```
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES
(12, 112, 212, 'Informative education expo!', TO_TIMESTAMP('2025-01-15 22:30:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES
(13, 113, 213, 'Book fair was a paradise!', TO_TIMESTAMP('2025-02-28 16:45:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES
(14, 114, 214, 'Tech conference was insightful!', TO_TIMESTAMP('2025-03-22 19:30:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES
(15, 115, 215, 'Mesmerizing dance showcase!', TO_TIMESTAMP('2025-04-10 18:15:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES
(16, 116, 216, 'Gaming expo was awesome!', TO_TIMESTAMP('2025-05-05 22:45:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES
(17, 117, 217, 'Charity run for a good cause!', TO_TIMESTAMP('2025-06-20 20:30:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES
(18, 118, 218, 'Exciting travel fair!', TO_TIMESTAMP('2025-07-15 11:30:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES
(19, 119, 219, 'Latest trends in automotive!', TO_TIMESTAMP('2025-08-08 17:15:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Feedback (FeedbackID, UserID, EventID, Feedback_Text, Timestamp) VALUES
(20, 120, 220, 'Impressive tech innovations!', TO_TIMESTAMP('2025-09-25 16:45:00',
'YYYY-MM-DD HH24:MI:SS'))
SELECT 1 FROM dual;
```

-- Inserting data into PaymentMethod table

```
INSERT ALL
    INTO PaymentMethod (MethodID, Payment_Method) VALUES (1, 'Credit Card')
    INTO PaymentMethod (MethodID, Payment_Method) VALUES (2, 'Debit Card')
    INTO PaymentMethod (MethodID, Payment_Method) VALUES (3, 'PayPal')
    INTO PaymentMethod (MethodID, Payment_Method) VALUES (4, 'FPX')
SELECT 1 FROM dual;
```

-- Inserting data into Payment table

```
INSERT ALL
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (1, 101,
1, 50.00, TO_TIMESTAMP('2024-02-10 13:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (2, 113,
2, 75.00, TO_TIMESTAMP('2024-03-01 14:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (3, 103,
3, 100.00, TO_TIMESTAMP('2024-04-01 19:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (4, 104,
4, 60.00, TO_TIMESTAMP('2024-04-20 15:30:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (5, 107,
4, 120.00, TO_TIMESTAMP('2024-05-15 21:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (6, 106,
2, 40.00, TO_TIMESTAMP('2024-06-08 03:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (7, 107,
1, 90.00, TO_TIMESTAMP('2024-06-03 12:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (8, 108,
1, 80.00, TO_TIMESTAMP('2024-06-18 19:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (9, 109,
2, 55.00, TO_TIMESTAMP('2024-09-12 22:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (10, 110,
3, 70.00, TO_TIMESTAMP('2024-02-05 17:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (11, 111,
4, 65.00, TO_TIMESTAMP('2024-02-20 22:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (12, 112,
4, 110.00, TO_TIMESTAMP('2025-01-01 23:00:00', 'YYYY-MM-DD HH24:MI:SS'))
```

```
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (13, 113,
4, 45.00, TO_TIMESTAMP('2025-01-28 17:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (14, 114,
4, 95.00, TO_TIMESTAMP('2025-02-22 20:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (15, 115,
1, 85.00, TO_TIMESTAMP('2025-02-10 19:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (16, 116,
2, 50.00, TO_TIMESTAMP('2025-03-05 23:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (17, 117,
3, 30.00, TO_TIMESTAMP('2025-05-20 21:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (18, 118,
4, 75.00, TO_TIMESTAMP('2025-06-15 12:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (19, 119,
4, 40.00, TO_TIMESTAMP('2025-05-08 18:00:00', 'YYYY-MM-DD HH24:MI:SS'))
    INTO Payment (PaymentID, UserID, MethodID, Amount, Timestamp) VALUES (20, 120,
4, 60.00, TO_TIMESTAMP('2025-03-25 17:00:00', 'YYYY-MM-DD HH24:MI:SS'))
SELECT 1 FROM dual;
```

-- Inserting data into TicketType table

```
INSERT ALL
    INTO TicketType (Ticket_TypeID, Ticket_Type) VALUES (1, 'Early Bird')
    INTO TicketType (Ticket_TypeID, Ticket_Type) VALUES (2, 'Normal')
    INTO TicketType (Ticket_TypeID, Ticket_Type) VALUES (3, 'VIP')
SELECT 1 FROM dual;
```

-- Inserting data into Ticket table

```
INSERT ALL
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (1, 101, 1, 201, 1, 50.00, TO_DATE('2024-02-10 13:00:00',
'YYYY-MM-DD HH24:MI:SS'))
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (2, 113, 2, 202, 2, 75.00, TO_DATE('2024-03-01 14:00:00',
'YYYY-MM-DD HH24:MI:SS'))
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (3, 103, 3, 203, 3, 100.00, TO_DATE('2024-04-01 19:00:00',
'YYYY-MM-DD HH24:MI:SS'))
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (4, 104, 4, 204, 1, 60.00, TO_DATE('2024-04-20 15:30:00',
'YYYY-MM-DD HH24:MI:SS'))
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (5, 107, 5, 205, 2, 120.00, TO_DATE('2024-05-15 21:00:00',
'YYYY-MM-DD HH24:MI:SS'))
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (6, 106, 6, 206, 3, 40.00, TO_DATE('2024-06-08 03:00:00',
'YYYY-MM-DD HH24:MI:SS'))
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (7, 107, 7, 207, 1, 90.00, TO_DATE('2024-06-03 12:00:00',
'YYYY-MM-DD HH24:MI:SS'))
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (8, 108, 8, 208, 2, 80.00, TO_DATE('2024-06-18 19:00:00',
'YYYY-MM-DD HH24:MI:SS'))
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (9, 109, 9, 209, 3, 55.00, TO_DATE('2024-09-12 22:00:00',
'YYYY-MM-DD HH24:MI:SS'))
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (10, 110, 10, 210, 1, 70.00, TO_DATE('2024-05-02 17:00:00',
'YYYY-MM-DD HH24:MI:SS'))
   INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (11, 111, 11, 211, 2, 65.00, TO_DATE('2024-02-20 22:00:00',
'YYYY-MM-DD HH24:MI:SS'))
```

```
    INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (12, 112, 12, 212, 3, 110.00, TO_DATE('2025-01-01 23:00:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (13, 113, 13, 213, 1, 45.00, TO_DATE('2025-01-28 17:00:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (14, 114, 14, 214, 2, 95.00, TO_DATE('2025-02-22 20:00:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (15, 115, 15, 215, 3, 85.00, TO_DATE('2025-02-10 19:00:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (16, 116, 16, 216, 1, 50.00, TO_DATE('2025-03-05 23:00:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (17, 117, 17, 217, 2, 30.00, TO_DATE('2025-05-20 21:00:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (18, 118, 18, 218, 3, 75.00, TO_DATE('2025-06-15 12:00:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (19, 119, 19, 219, 1, 40.00, TO_DATE('2025-05-08 18:00:00',
'YYYY-MM-DD HH24:MI:SS'))
    INTO Ticket (TicketID, UserID, PaymentID, EventID, Ticket_TypeID, Price,
Date_of_Purchase) VALUES (20, 120, 20, 220, 2, 60.00, TO_DATE('2025-03-25 17:00:00',
'YYYY-MM-DD HH24:MI:SS'))
SELECT 1 FROM dual;
```

-- Inserting data into Report table

```
INSERT ALL
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (1, 201, 101, 'Health and Wellness Seminar Report', TO_CLOB('Report content for Health and Wellness Seminar.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (2, 202, 113, 'Cultural Festival Report', TO_CLOB('Report content for Cultural Festival.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (3, 203, 103, 'Startup Pitch Day Report', TO_CLOB('Report content for Startup Pitch Day.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (4, 204, 104, 'Art Exhibition Report', TO_CLOB('Report content for Art Exhibition.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (5, 205, 107, 'Music Festival Report', TO_CLOB('Report content for Music Festival.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (6, 206, 106, 'Green Living Expo Report', TO_CLOB('Report content for Green Living Expo.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (7, 208, 108, 'Food Truck Festival Report', TO_CLOB('Report content for Food Truck Festival.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (8, 211, 111, 'Film Festival Report', TO_CLOB('Report content for Film Festival.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (9, 212, 112, 'Education Expo Report', TO_CLOB('Report content for Education Expo.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (10, 213, 113, 'Book Fair Report', TO_CLOB('Report content for Book Fair.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (11, 214, 114, 'Technology Conference Report', TO_CLOB('Report content for Technology Conference.'), CURRENT_TIMESTAMP)
```

```
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (12, 218, 118, 'Travel Fair Report', TO_CLOB('Report content for Travel Fair.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (13, 219, 119, 'Automotive Expo Report', TO_CLOB('Report content for Automotive Expo.'), CURRENT_TIMESTAMP)
    INTO Report (ReportID, EventID, UserID, ReportTitle, ReportData, Timestamp) VALUES (14, 220, 120, 'Tech Expo 2024 Report', TO_CLOB('Report content for Tech Expo 2024.'), CURRENT_TIMESTAMP)
SELECT 1 FROM dual;
```

7.0 Summary

To sum up, the Nexscholar Event Management System project has reached a major milestone with the completion of Phase 3. We are prepared to create a database structure that not only satisfies present requirements but also is flexible and scalable to suit the changing needs of our platform thanks to a well-honed business rule set, an updated Conceptual ERD, and a rigorous logical design process. By laying the foundation for the subsequent phases, this phase moves us one step closer to implementing a strong and effective ticketing system and event management at Nexscholar.

Creating a thorough Logical ERD, updating the Data Dictionary, and using normalisation techniques are all part of the logical design phase. This painstaking procedure seeks to reduce redundancy, improve data integrity, and maximise database efficiency. The result is a coherent design that acts as a link between the physical execution and the conceptual model.

Phase 3 lays the groundwork for the actual database implementation by culminating in the production of relational database schemas and SQL statements (DDL & DML). These elements give developers the useful framework they need to create and maintain the database, transforming our abstract system design into a functional system.

We further develop our visual representation of our data model, the Entity-Relationship Diagram (ERD). By anticipating future scalability and capturing the system's current state, this technique lays the groundwork for a reliable and flexible architecture.