DATABASE MANAGEMENT SYSTEM - CSA0593

ASSIGNMENT 2

K.GAYATHRI

192311448

QUESTION:

"Design a database to manage events, tickets, customers, and payments.

- Model tables for events, tickets, customers, and payments.

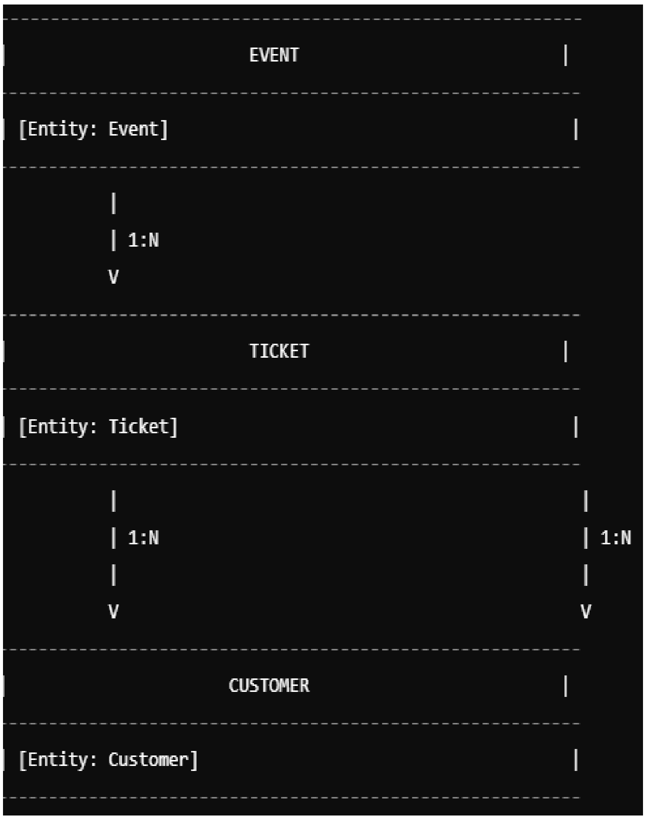
- Write stored procedures for purchasing and canceling tickets.

- Implement triggers to update ticket availability and payment status.

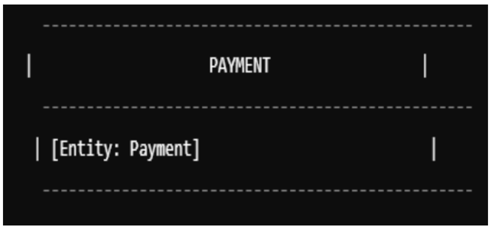
- Write SQL queries to analyze ticket sales and customer demographics.

ANSWER:

CONCEPTUAL E.R.DIAGRAM:

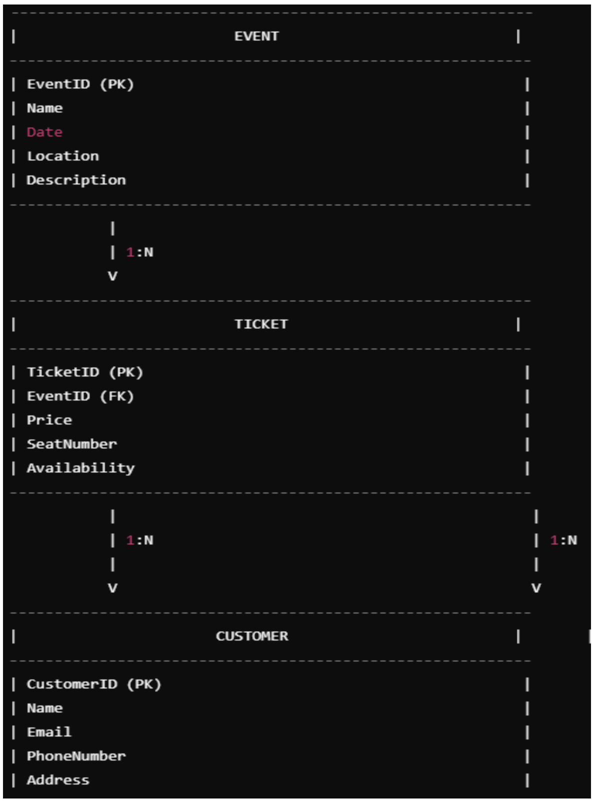


```
------------------------------------------------------------
|                        EVENT                        |
------------------------------------------------------------
| [Entity: Event]                                      |
------------------------------------------------------------
          |
          | 1:N
          V
------------------------------------------------------------
|                        TICKET                       |
------------------------------------------------------------
| [Entity: Ticket]                                     |
------------------------------------------------------------
          |                                              |
          | 1:N                                          | 1:N
          |                                              |
          V                                              V
------------------------------------------------------------
|                      CUSTOMER                       |
------------------------------------------------------------
| [Entity: Customer]                                   |
------------------------------------------------------------
```

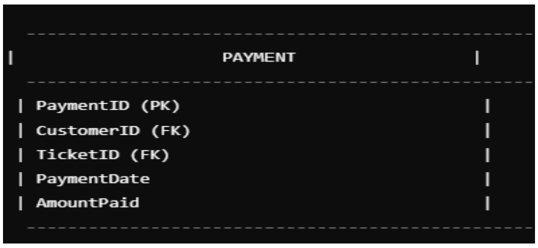


```
   ---------------------------------------------
|                 PAYMENT                 |
   ---------------------------------------------
 | [Entity: Payment]                        |
   ---------------------------------------------
```

LOGICAL E.R.DIAGRAM:

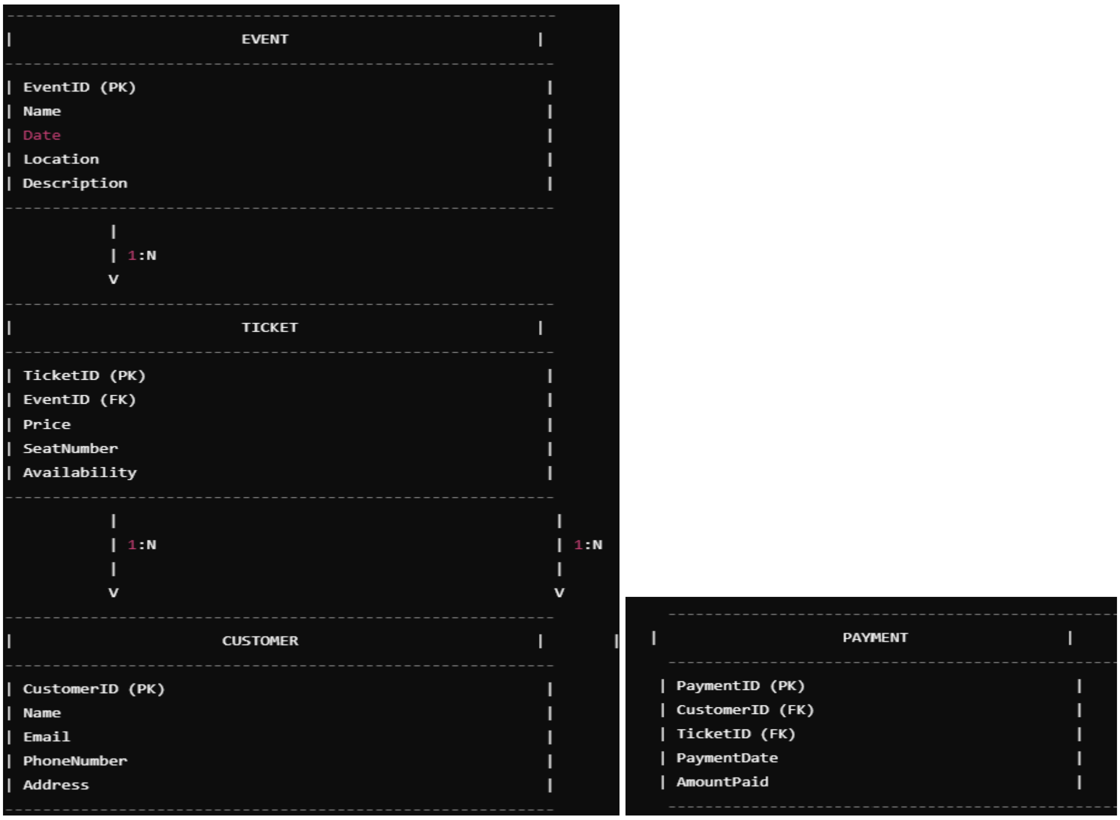


```
-----------------------------------------------------------
|                        EVENT                            |
-----------------------------------------------------------
| EventID (PK)                                             |
| Name                                                     |
| Date                                                     |
| Location                                                 |
| Description                                              |
-----------------------------------------------------------
           |
           | 1:N
           v
-----------------------------------------------------------
|                        TICKET                           |
-----------------------------------------------------------
| TicketID (PK)                                            |
| EventID (FK)                                             |
| Price                                                    |
| SeatNumber                                               |
| Availability                                             |
-----------------------------------------------------------
           |                                                |
           | 1:N                                            | 1:N
           |                                                |
           v                                                v
-----------------------------------------------------------
|                       CUSTOMER                          |      |
-----------------------------------------------------------
| CustomerID (PK)                                          |
| Name                                                     |
| Email                                                    |
| PhoneNumber                                              |
| Address                                                  |
```



```
  -----------------------------------------------------------
|                        PAYMENT                          |
  -----------------------------------------------------------
 | PaymentID (PK)                                          |
 | CustomerID (FK)                                         |
 | TicketID (FK)                                           |
 | PaymentDate                                             |
 | AmountPaid                                              |
  -----------------------------------------------------------
```

PHYSICAL E.R.DIAGRAM:



```
---------------------------------------------------------
|                     EVENT                           |
---------------------------------------------------------
| EventID (PK)                                         |
| Name                                                 |
| Date                                                 |
| Location                                             |
| Description                                          |
---------------------------------------------------------
           |
           | 1:N
           v
---------------------------------------------------------
|                     TICKET                          |
---------------------------------------------------------
| TicketID (PK)                                        |
| EventID (FK)                                         |
| Price                                                |
| SeatNumber                                           |
| Availability                                         |
---------------------------------------------------------
           |                                              |
           | 1:N                                          | 1:N
           |                                              |
           v                                              v
---------------------------------------------------------
|                   CUSTOMER                          |      |
---------------------------------------------------------
| CustomerID (PK)                                      |
| Name                                                 |
| Email                                                |
| PhoneNumber                                          |
| Address                                              |
---------------------------------------------------------

    ---------------------------------------------------------
    |                     PAYMENT                         |
    ---------------------------------------------------------
     | PaymentID (PK)                                      |
     | CustomerID (FK)                                     |
     | TicketID (FK)                                       |
     | PaymentDate                                         |
     | AmountPaid                                          |
    ---------------------------------------------------------
```

MYSQL STATEMENTS:

Database Design

```
CREATE DATABASE event_management;

USE event_management;
```

```
CREATE TABLE events (
  event_id INT PRIMARY KEY,
  event_name VARCHAR(255),
  event_date DATE,
  event_time TIME,
  venue VARCHAR(255),
  capacity INT
);

CREATE TABLE tickets (
  ticket_id INT PRIMARY KEY,
  event_id INT,
  ticket_type VARCHAR(20),
  price DECIMAL(10, 2),
  availability INT,
  FOREIGN KEY (event_id) REFERENCES events(event_id)
);

CREATE TABLE customers (
  customer_id INT PRIMARY KEY,
  name VARCHAR(255),
  email VARCHAR(255),
  phone VARCHAR(20),
  address VARCHAR(255)
);
```

```
CREATE TABLE payments (
  payment_id INT PRIMARY KEY,
  customer_id INT,
  event_id INT,
  ticket_id INT,
  payment_date DATE,
  payment_method VARCHAR(20),
  amount DECIMAL(10, 2),
  status VARCHAR(20),
  FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
  FOREIGN KEY (event_id) REFERENCES events(event_id),
  FOREIGN KEY (ticket_id) REFERENCES tickets(ticket_id)
);

CREATE TABLE orders (
  order_id INT PRIMARY KEY,
  customer_id INT,
  event_id INT,
  ticket_id INT,
  order_date DATE,
  quantity INT,
  total_amount DECIMAL(10, 2),
  FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
  FOREIGN KEY (event_id) REFERENCES events(event_id),
  FOREIGN KEY (ticket_id) REFERENCES tickets(ticket_id)
);
```

Stored Procedures

```
DELIMITER //

CREATE PROCEDURE purchase_ticket(
  IN customer_id INT,
  IN event_id INT,
  IN ticket_id INT,
  IN quantity INT
)
BEGIN
  DECLARE available_tickets INT;
  SELECT availability INTO available_tickets
  FROM tickets
  WHERE ticket_id = ticket_id;

  IF available_tickets >= quantity THEN
    INSERT INTO orders (customer_id, event_id, ticket_id, order_date, quantity,
total_amount)
    VALUES (customer_id, event_id, ticket_id, CURDATE(), quantity, quantity *
(SELECT price FROM tickets WHERE ticket_id = ticket_id));

    UPDATE tickets
    SET availability = availability - quantity
    WHERE ticket_id = ticket_id;
  ELSE
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Insufficient tickets available';
  END IF;
END //

CREATE PROCEDURE cancel_ticket(
  IN order_id INT
)
BEGIN
  DECLARE ticket_id INT;
  DECLARE quantity INT;
  DECLARE event_id INT;

  SELECT ticket_id, quantity, event_id INTO ticket_id, quantity, event_id
  FROM orders
  WHERE order_id = order_id;

  UPDATE tickets
  SET availability = availability + quantity
  WHERE ticket_id = ticket_id;

  DELETE FROM orders
  WHERE order_id = order_id;
END //
```

Triggers

```
DELIMITER //

CREATE TRIGGER update_payment_status
AFTER UPDATE ON payments
FOR EACH ROW
BEGIN
  IF NEW.status = 'Paid' THEN
    UPDATE orders
    SET status = 'Confirmed'
    WHERE order_id = (SELECT order_id FROM payments WHERE payment_id =
NEW.payment_id);
  END IF;
END //

CREATE TRIGGER update_ticket_availability
AFTER INSERT ON orders
FOR EACH ROW
BEGIN
  UPDATE tickets
  SET availability = availability - NEW.quantity
  WHERE ticket_id = NEW.ticket_id;
END //
```

SQL Queries

```
-- Analyze ticket sales
SELECT
  events.event_name,
  SUM(orders.quantity) AS total_tickets_sold,
  SUM(orders.total_amount) AS total_revenue
FROM
  events
  JOIN orders ON events.event_id = orders.event_id
GROUP BY
  events.event_name;

-- Customer demographics
SELECT
  customers.name,
  customers.email,
  customers.phone,
  customers.address,
  COUNT(orders.order_id) AS number_of_orders
FROM
  customers
  JOIN orders ON customers.customer_id = orders.customer_id
```

```
GROUP BY
  customers.name;
```

Conclusion:

Designing a database to manage events, tickets, customers, and payments requires careful consideration of various factors.

Key benefits of this system include:

1. Efficient ticket purchasing and management.
2. Automated updates to ticket availability and payment status.
3. Centralized storage of customer information and order history.
4. Data-driven insights into ticket sales and customer demographics.

By implementing this database management system, event organizers can improve operational efficiency, enhance customer experiences, and increase revenue.