**Conceptual ERD Model:**

- Correctly identified entities and relationships.

- Consider adding an entity for "Category" to further normalize the data.

**Logical ERD Model:**

- Well-defined attributes for each entity.

- Consider adding a "Status" attribute to the Enrollment entity (e.g., "Enrolled", "Completed", "InProgress").

**Physical ERD Model (Database Schema):**

- Proper use of primary keys, foreign keys, and data types.

- Consider indexing columns used in WHERE, JOIN, and ORDER BY clauses for improved performance.

**Additional Suggestions:**

1. Consider adding a "Rating" attribute to the Course entity to store user ratings.

2. Add a "Timestamp" attribute to the Completion entity to track completion time.

3. Create a separate table for "Questions" to store individual quiz questions.

4. Use ENUM data type for "Category" and "Status" attributes.

5. Use VARCHAR(255) for "Password" to accommodate hashed passwords.

**SQL Enhancements:**

1. Use CONSTRAINTS to enforce data integrity.

2. Create INDEXES on columns used in queries.

3. Consider using VIEWs to simplify complex queries.

Next Steps:

1. Implement the database schema.

2. Create CRUD (Create, Read, Update, Delete) operations for each entity.

3. Develop queries for reporting and analytics.

Would you like me to:

1. Review and refine the database schema?

2. Assist with CRUD operation implementation?

3. Develop queries for reporting and analytics?

4. Discuss database security and backup strategies?

**Conceptual ER Diagram**

**Rooms :**

Room\_ID

Room\_Type

Rate

| - Availability|

**Guests:**

| - Guest\_ID |

| - Name |

| - Email |

| - Phone |

| - Address |

**Reservations:**

| - Reservation\_ID|

| - Room\_ID |

| - Guest\_ID |

| - Arrival\_Date|

| - Departure\_Date|

| - Status |

**Payments:**

| - Payment\_ID |

| - Reservation\_ID|

| - Payment\_Date|

| - Amount |

| - Method |

**Logical ER Diagram:**

CREATE TABLE Rooms (

Room\_ID INT PRIMARY KEY,

Room\_Type VARCHAR(50),

Rate DECIMAL(10, 2),

Availability VARCHAR(20) CHECK(Availability IN ('Available', 'Occupied'))

);

CREATE TABLE Guests (

Guest\_ID INT PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100),

Phone VARCHAR(20),

Address VARCHAR(200)

);

CREATE TABLE Reservations (

Reservation\_ID INT PRIMARY KEY,

Room\_ID INT,

Guest\_ID INT,

Arrival\_Date DATE,

Departure\_Date DATE,

Status VARCHAR(20) CHECK(Status IN ('Pending', 'Confirmed', 'Canceled')),

FOREIGN KEY (Room\_ID) REFERENCES Rooms(Room\_ID),

FOREIGN KEY (Guest\_ID) REFERENCES Guests(Guest\_ID)

);

CREATE TABLE Payments (

Payment\_ID INT PRIMARY KEY,

Reservation\_ID INT,

Payment\_Date DATE,

Amount DECIMAL(10, 2),

Method VARCHAR(20),

FOREIGN KEY (Reservation\_ID) REFERENCES Reservations(Reservation\_ID)

);

**Physical ER Diagram (SQL):**

CREATE TABLE Rooms (

Room\_ID INT PRIMARY KEY,

Room\_Type VARCHAR(50),

Rate DECIMAL(10, 2),

Availability VARCHAR(20) CHECK(Availability IN ('Available', 'Occupied'))

);

CREATE TABLE Guests (

Guest\_ID INT PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100),

Phone VARCHAR(20),

Address VARCHAR(200)

);

CREATE TABLE Reservations (

Reservation\_ID INT PRIMARY KEY,

Room\_ID INT,

Guest\_ID INT,

Arrival\_Date DATE,

Departure\_Date DATE,

Status VARCHAR(20) CHECK(Status IN ('Pending', 'Confirmed', 'Canceled')),

FOREIGN KEY (Room\_ID) REFERENCES Rooms(Room\_ID),

FOREIGN KEY (Guest\_ID) REFERENCES Guests(Guest\_ID)

);

CREATE TABLE Payments (

Payment\_ID INT PRIMARY KEY,

Reservation\_ID INT,

Payment\_Date DATE,

Amount DECIMAL(10, 2),

Method VARCHAR(20),

FOREIGN KEY (Reservation\_ID) REFERENCES Reservations(Reservation\_ID)

);

Stored Procedures

CREATE PROCEDURE sp\_make\_reservation

@Room\_ID INT,

@Guest\_ID INT,

@Arrival\_Date DATE,

@Departure\_Date DATE

AS

BEGIN

INSERT INTO Reservations (Room\_ID, Guest\_ID, Arrival\_Date, Departure\_Date, Status)

VALUES (@Room\_ID, @Guest\_ID, @Arrival\_Date, @Departure\_Date, 'Pending');

END;

CREATE PROCEDURE sp\_update\_reservation

@Reservation\_ID INT,

@Room\_ID INT,

@Guest\_ID INT,

@Arrival\_Date DATE,

@Departure\_Date DATE

AS

BEGIN

UPDATE Reservations

SET Room\_ID = @Room\_ID, Guest\_ID = @Guest\_ID, Arrival\_Date = @Arrival\_Date, Departure\_Date = @Departure\_Date

WHERE Reservation\_ID = @Reservation\_ID;

END;

CREATE PROCEDURE sp\_cancel\_reservation

@Reservation\_ID INT

AS

BEGIN

UPDATE Reservations

SET Status = 'Canceled'

WHERE Reservation\_ID = @Reservation\_ID;END;