A swimming pool with a building in the background

Description automatically generated with low confidence

**Concepted and presented by Mohamed A Tounkara**

1. Abstract

The Hotel Management database system aims to simplify the check-in and check-out when customers arrive. The process used will digitalize the whole check-in system of the Hotel to improve customer service and increase process efficiency. Doing so will help the Hotel make more money because employees will process check-in and check-out faster. Finally digitalizing the database system will make access to the database for audit and research purposes easier.

1. Mission statement

The purpose of the database system is to maintain the data that is used and generate to support the Hotel booking system for some property owners, facilitate the research of reservations and improve the quality of the service provided to the customers during the check-in and out process.

1. Mission objectives

To maintain (enter, update, and delete) data on hotel.

To maintain (enter, update, and delete) data on customers.

To maintain (enter, update, and delete) data on reservations.

To maintain (enter, update, and delete) data on categoryroom.

To maintain (enter, update, and delete) data on hotelcategory.

To perform searches on hotels.

To perform searches on customers

To perform searches on reservations

To perform searches on categoryroom

To perform searches on hotelcategory

To track the status of a reservation

To track the status of a room

To track the status of a hotel

To report on hotel

To report on customers

To report on reservation

To report on categoryroom

To report on hotelcategory

User Case Name: Enter a new reservation

Actor: Manager

Steps:

1. Actor clicks on “New reservation” button;
2. A new reservationID is shown
3. Prompt to enter reservationDate Date, arrDate Date, depDate Date, CheckIn, CheckOut, price
4. All information is displayed; ask for confirmation
5. User clicks on “Confirm” button

User Case Name: Enter a new client

Actor: Manager

Steps:

1. Actor clicks on “New client” button;
2. A new clientID is shown
3. Prompt to enter firstName, lastName, clientName
4. All information is displayed; ask for confirmation
5. User clicks on “Confirm” button

User Case Name: Enter a new room

Actor: Manager

Steps:

1. Actor clicks on “New room” button;
2. A new roomID is shown
3. Prompt to enter roomName
4. All information is displayed; ask for confirmation
5. User clicks on “Confirm” button

User Case Name: Enter a new hotel

Actor: Manager

Steps:

1. Actor clicks on “New hotel” button.
2. A new hotelID is shown
3. Prompt to enter hotelName, hotelAdress, hotelCat
4. All information is displayed; ask for confirmation
5. User clicks on “Confirm” button

User Case Name: Enter a new hotelcategory….

Actor: Manager

Steps:

1. Actor clicks on “New hotelcategory” button.
2. A new HotelCat is shown
3. Prompt to enter HotelCat, Stars
4. All information is displayed; ask for confirmation
5. User clicks on “Confirm” button

Graphical user interface, text, application

Description automatically generated

hotel table consists of information related to a hotel. HotelID is the primary key of the table. It has Not Null constraint and Unique constraint.

hotelcategory\_Hotelcat is a foreign key that is related to hotelcategory table. hotel table has a one-to-many relation with the room table and one to one relationship with hotelcategory.

Graphical user interface, text, application

Description automatically generated

client table contains the information related to a client. ClientId is the primary key of the table. It has Not Null constraint and Unique constraint.

Graphical user interface, text, application

Description automatically generated

reservation table consists of information related to the reservations. ReservationID is the primary key of the table. It has Not Null constraint and Unique constraint.

Reservation has a one-to-one relationship with room, period and client.

Graphical user interface, text, application, chat or text message

Description automatically generated

room table consists of information related to a room. roomID is the primary key of the table. It has Not Null constraint and Unique constraint. roomCat is a foreign key that is related to categoryroom table. room table has a one-to-one relation with the hotel and categoryroom, and one-to-many relationship with reservation.

Graphical user interface, text, application, chat or text message

Description automatically generated

HotelCat is the primary key of hotelcategory table which create a one-to-many relationship with hotel.

Graphical user interface, text, application

Description automatically generated

categoryroom has RoomCat as primary key. It has a one-to-many relationship with room.

Application, table

Description automatically generated

period has PeriodID as primary key. It has a one-to-many relationship with reservation.

I completed the first normal form by eliminating repeating data. Coming to the second normal form, I tried to make the relationships depend on the primary key.

On the third normal form, I made sure that all the dependencies are only on the primary key of the tables.

**SQL QUERIES**

1. Client

* Insertion

INSERT INTO `hotels`.`client` (`ClientId`, `FirstName`, `LastName`) VALUES ('5', 'AKON', 'BADIAM');

* Update

UPDATE `hotels`.`client` SET `ClientEmail` = 'alioubadiam@gmail.com' WHERE (`ClientId` = '5');

* Delete

DELETE FROM `hotels`.`client` WHERE (`ClientId` = '5');

* Aggregate

SELECT COUNT(ClientId) AS TotalClient

FROM hotels.client;

1. Hotel

* Insertion

INSERT INTO `hotels`.`hotel` (`HotelID`, `HotelName`, `HotelAdressl`, `HotelCat`) VALUES ('HLSA', 'HILTON LOUISIANNA', '1200 Louisiana st', 'D');

* Update

UPDATE `hotels`.`hotel` SET `HotelAdressl` = '1402 Post Oak Boulevard' WHERE (`HotelID` = 'HPO');

* Delete

DELETE FROM `hotels`.`hotel` WHERE (`HotelID` = 'HWSE');

* Aggregate

SELECT HotelCat, COUNT(HotelCat) as total

FROM hotels.hotel

GROUP BY HotelCat;

* Join

SELECT Stars

FROM hotels.HotelCategory, hotels.hotel

WHERE hotel.HotelCat = hotelcategory.HotelCat

AND HotelID = 'HDTN';

1. Reservation

* Insertion

INSERT INTO `hotels`.`reservation` (`ReservationID`, `ClientID`, `RoomID`, `PeriodID`, `ReeservationDate`, `ArrDate`, `DepDate`, `Price`) VALUES ('4', '1', 'HDTN1', 'SPG', '2023-02-05', '2023-01-03', '2023-01-06', '179');

* Update

UPDATE `hotels`.`reservation` SET `Price` = '180' WHERE (`ReservationID` = '4');

* Delete

DELETE FROM `hotels`.`reservation` WHERE (`ReservationID` = '4');

* Aggregate

SELECT MIN(Price)

FROM hotels.reservation;

* Join

SELECT LastName

FROM hotels.reservation, hotels.client

where reservation.ClientID= client.ClientID

and ClientEMAIL= 'stewartj23@yahoo.com';

1. Room

* Insertion

INSERT INTO `hotels`.`room` (`RoomID`, `RoomName`, `HotelID`, `RoomCat`) VALUES ('HPO4', 'HILTON POST OAK 4', 'HPO', '4');

* Update

UPDATE `hotels`.`room` SET `HotelID` = 'HWSE' WHERE (`RoomID` = 'HWSE1');

* Delete

DELETE FROM `hotels`.`room` WHERE (`RoomID` = 'HWSE2');

* Aggregate

SELECT HotelID, COUNT(RoomID) AS TOTAL

FROM hotels.room

GROUP BY HotelID;

* Join

SELECT RoomID

FROM hotels.hotel, hotels.room

WHERE hotel.HotelID = room.HotelID AND HotelName= 'HILTON DOWNTOWN';

1. Hotelcategory

* Insertion

INSERT INTO `hotels`.`hotelcategory` (`HotelCat`, `Stars`) VALUES ('E', '1 STAR');

* Update

UPDATE `hotels`.`hotelcategory` SET `Stars` = '2 STARS' WHERE (`HotelCat` = 'D');

* Delete

DELETE FROM `hotels`.`hotelcategory` WHERE (`HotelCat` = 'D');

* Aggregate

SELECT COUNT(HotelCat)

FROM hotels.hotelcategory;

1. Categoryroom

* Insert

INSERT INTO `hotels`.`categoryroom` (`RoomCat`, `RoomType`) VALUES ('4', 'PENTHOUSE ROOM');

* Update

UPDATE `hotels`.`categoryroom` SET `RoomType` = 'JUNIOR ROOM' WHERE (`RoomCat` = '4');

* Delete

DELETE FROM `hotels`.`categoryroom` WHERE (`RoomCat` = '4');

* Aggregate

SELECT COUNT(RoomCat)

FROM hotels.categoryroom;

1. Period

* Insert

INSERT INTO `hotels`.`period` (`PeriodID`, `PeriodName`) VALUES ('S', 'SPRING');

* Update

UPDATE `hotels`.`period` SET `PeriodName` = 'SPRING' WHERE (`PeriodID` = 'SPG');

* Delete

DELETE FROM `hotels`.`period` WHERE (`PeriodID` = 'S');

* Aggregate

SELECT COUNT(PeriodID)

FROM hotels.period;

Conclusion

I constructed a Hotel Management database system that aimed to simplify and digitalize the check-in and check-out process when customers arrive while improving process efficiency. I then created an ER Diagram that showed all the tables involved in our database, their types of relationships (one-to-one or one-to-many between tables) and identified primary and foreign keys. I also performed a series of queries (update, delete, insert, aggregate, and join) to test the functionality of my database.

Finally, digitalizing the database system will make access to the database for audits, manipulations, and research purposes easier.

Reference

1-<https://medium.com/@datamatecrescent/what-is-a-hotel-management-system-c5d8b9249d1>

2- https://www.altexsoft.com/blog/hotel-data-management-best-practices/

3-https://itsourcecode.com/uml/hotel-management-system-er-diagram-best-database-design-2021/