# Database Design: Hotel Booking

*A guide for a database design for a sample hotel booking system*

Ben Brumm

www.databasestar.com

# Database Design: Hotel Booking

This guide is a companion to my YouTube video on designing a database for a hotel booking system.

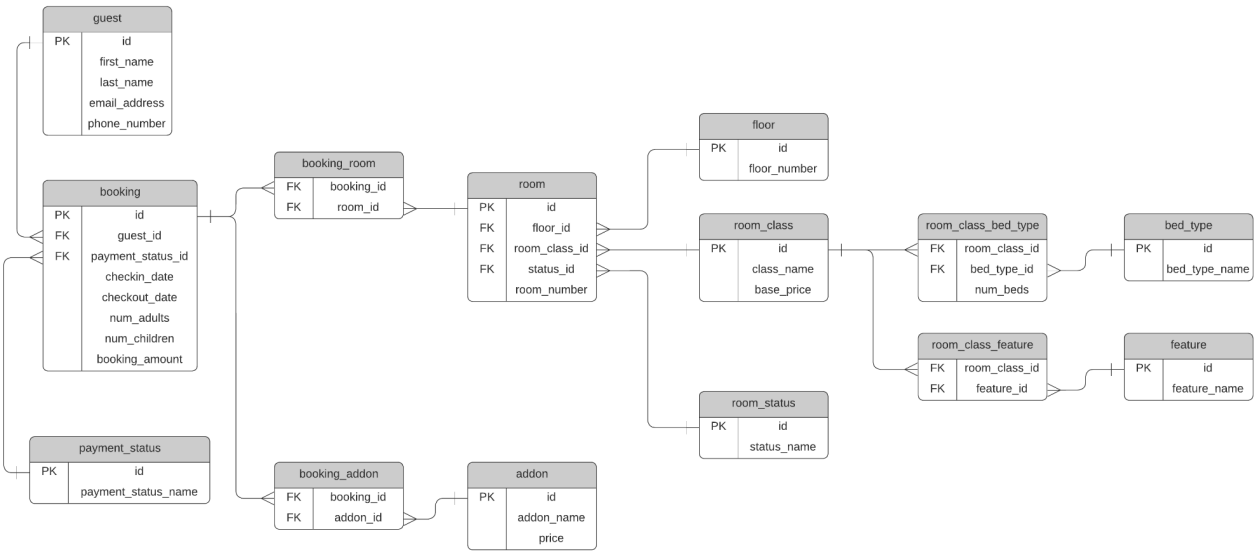
In this guide, you'll see:

- An Entity Relationship Diagram for a hotel booking system, from my YouTube video.
- An explanation of the purpose of each table and field, with sample data
- SQL scripts to create each of these tables with some sample data

Let's get into it.

## Entity Relationship Diagram

Here's the ERD for this database:




A PNG file of this ERD is available here:
https://dbshostedfiles.s3.us-west-2.amazonaws.com/dbs/erd_hotel.png

# Database Definition

This section explains each of these tables and fields.

## guest

A guest is a person who stays at the hotel and the one who makes the booking.

| Column | Description | Sample Data |
|---|---|---|
| id | Primary key. A unique identifier for the row. | 1, 2, 3 |
| first_name | The first name of the guest | John<br>Sandip<br>Stephanie |
| last_name | The last name of the guest | Kumar<br>Smith<br>Jones |
| email_address | The email address for the guest, which can be used to send booking details to and to log in. | john@apple.com |
| phone_number | The phone number for the guest which can be used to contact them. | 0139029321 |

## booking

A record of a guest making a booking to stay in the hotel. This is created when the guest makes the booking, which can be some time before the date the guest stays in the hotel (the check in date).

| Column | Description | Sample Data |
|---|---|---|
| id | Primary key. A unique identifier for the row. | 1, 2, 3 |
| guest_id | The foreign key to the guest table, which indicates the guest that the booking is for. | 1, 5, 6 |
| payment_status_id | The foreign key to the payment_status table, which indicate the status of the | 1, 2 |

| | | |
|---|---|---|
| | booking | |
| checkin_date | The date that the guest is checking in to their room. | 30 Sep 2023 |
| checkout_date | The date that the guest is checking out of their room. | 4 Oct 2023 |
| num_adults | The number of adults included in the booking. Multiple adults may lead to booking multiple rooms. | 2 |
| num_children | The number of children included in the booking. | 2 |
| booking_amount | The amount the guest needs to pay for their booking. | 810.00 |

## payment_status

A lookup table for the different payment statuses for a booking.

| Column | Description | Sample Data |
|---|---|---|
| id | Primary key. A unique identifier for the row. | 1, 2, 3 |
| payment_status_name | A name for the payment status which can be understood by guests and other users. | Paid<br>Pending<br>Not Paid |

## addon

A lookup table for the different things that a guest can pay for in their booking.

| Column | Description | Sample Data |
|---|---|---|
| id | Primary key. A unique identifier for the row. | 1, 2, 3 |
| addon_name | The name of the addon that is understood by the guest | Valet Parking<br>Late Checkout |

| | | |
|---|---|---|
| | | Minibar - Small Alcohol |
| price | The price the guest needs to pay for the addon | 20<br>50<br>8 |

## booking_addon

A record of all of the addons that the customer adds to their booking.

| Column | Description | Sample Data |
|---|---|---|
| booking_id | A foreign key to the booking table, to indicate which booking this record relates to. | 1, 2, 3 |
| addon_id | A foreign key to the addon table, to indicate which addon this record relates to. | 1, 2, 3 |

## booking_room

A record of rooms for a booking, because a room can be booked many times, and a booking can include multiple rooms.

| Column | Description | Sample Data |
|---|---|---|
| booking_id | A foreign key to the booking table, to indicate which booking this record relates to. | 1, 2, 3 |
| room_id | A foreign key to the room table, to indicate which room this record relates to. | 1, 2, 3 |

## room

A representation of a hotel room that guests can stay in and that guests can book.

| Column | Description | Sample Data |
|---|---|---|

| | | |
|---|---|---|
| id | Primary key. A unique identifier for the row. | 1, 2, 3 |
| floor_id | A foreign key to the floor table, to indicate which floor this room is on. | 1, 5, 6 |
| room_class_id | A foreign key to the room_class table, to indicate the class of this room | 2, 4, 5 |
| status_id | A foreign key to the status table, to indicate the status of the room. | 1, 3, 4 |
| room_number | The number of the room that may be shown near the door and written on the swipe card | 502, 1001 |

## floor

A lookup table for the floor that a room exists on.

| Column | Description | Sample Data |
|---|---|---|
| id | Primary key. A unique identifier for the row. | 1, 2, 3 |
| floor_number | The number of the floor that is known by the staff and guests. | 1, 4, 5A, 5B |

## room_status

A lookup table to contain the different statuses of a room's lifecycle.

| Column | Description | Sample Data |
|---|---|---|
| id | Primary key. A unique identifier for the row. | 1, 2, 3 |
| status_name | The name of the step in the lifecycle | Occupied, Ready to Clean, Available |

## room_class

A lookup table that defines the different classes of a room.

| Column | Description | Sample Data |
|---|---|---|
| id | Primary key. A unique identifier for the row. | 1, 2, 3 |
| class_name | The name of the class, and is familiar to staff and guests. | Deluxe, Standard, Premium, Presidential |
| base_price | The price for a room of this class | 100, 350, 1200 |

## bed_type

A lookup table for the different types of beds that are available for rooms.

| Column | Description | Sample Data |
|---|---|---|
| id | Primary key. A unique identifier for the row. | 1, 2, 3 |
| bed_type_name | The type of bed that can exist in a room. | Single, Double, Queen, King |

## room_class_bed_type

A record of the number and type of beds that exist in a particular room, because a room can have different types of beds.

| Column | Description | Sample Data |
|---|---|---|
| room_class_id | A foreign key to the room_class table, to indicate which room class the record belongs to. | 1, 3, 6 |
| bed_type_id | A foreign key to the bed_type table, to indicate which type of bed the record belongs to | 1, 2, 4 |
| num_beds | The number of beds of this type in this room | 1, 2 |

## feature

A lookup table for all of the features available in a room.

| Column | Description | Sample Data |
|---|---|---|
| id | Primary key. A unique identifier for the row. | 1, 2, 3 |
| feature_name | The name of the feature that the guests know it as | Coffee Machine<br>Air Conditioning<br>Free Wifi |

## room_class_feature

A record of the features available in a particular room.

| Column | Description | Sample Data |
|---|---|---|
| room_class_id | A foreign key for the room_class table, to indicate the class of room for this record. | 1, 2, 4 |
| feature_id | A foreign key for the feature table, to indicate the feature for this record. | 2, 3, 7 |

# SQL Scripts

Here is the SQL code to create the tables for this database.

The script is written for MySQL, but it can easily be modified to work on your preferred database vendor by changing the data types and removing the IF EXISTS (if your database doesn't support it).

```
CREATE DATABASE hotel_booking;

USE hotel_booking;

DROP TABLE IF EXISTS booking_room;
DROP TABLE IF EXISTS room;
DROP TABLE IF EXISTS floor;
DROP TABLE IF EXISTS room_status;
DROP TABLE IF EXISTS room_class_feature;
DROP TABLE IF EXISTS feature;
DROP TABLE IF EXISTS room_class;
DROP TABLE IF EXISTS bed_type;
DROP TABLE IF EXISTS booking_addon;
DROP TABLE IF EXISTS addon;
DROP TABLE IF EXISTS booking;
DROP TABLE IF EXISTS payment_status;
DROP TABLE IF EXISTS guest;

CREATE TABLE guest (
  id INT AUTO_INCREMENT,
  first_name VARCHAR(200),
  last_name VARCHAR(200),
  email_address VARCHAR(350),
  phone_number VARCHAR(20),
  CONSTRAINT pk_guest PRIMARY KEY (id)
);

CREATE TABLE payment_status (
  id INT AUTO_INCREMENT,
  payment_status_name VARCHAR(50),
  CONSTRAINT pk_paystatus PRIMARY KEY (id)
);


CREATE TABLE booking (
  id INT AUTO_INCREMENT,
  guest_id INT,
  payment_status_id INT,
  checkin_date DATE,
  checkout_date DATE,
```

```
  num_adults INT,
  num_children INT,
  booking_amount INT,
  CONSTRAINT pk_booking PRIMARY KEY (id),
  CONSTRAINT fk_booking_guest FOREIGN KEY (guest_id) REFERENCES guest
(id),
  CONSTRAINT fk_booking_paystatus FOREIGN KEY (payment_status_id)
REFERENCES payment_status (id)
);

CREATE TABLE addon (
  id INT AUTO_INCREMENT,
  addon_name VARCHAR(100),
  price INT,
  CONSTRAINT pk_addon PRIMARY KEY (id)
);

CREATE TABLE booking_addon (
  booking_id INT,
  addon_id INT,
  CONSTRAINT fk_bkaddon_booking FOREIGN KEY (booking_id) REFERENCES
booking (id),
  CONSTRAINT fk_bkaddon_addon FOREIGN KEY (addon_id) REFERENCES addon
(id)
);

CREATE TABLE bed_type (
  id INT AUTO_INCREMENT,
  bed_type_name VARCHAR(50),
  CONSTRAINT pk_bedtype PRIMARY KEY (id)
);

CREATE TABLE room_class (
  id INT AUTO_INCREMENT,
  class_name VARCHAR(100),
  base_price INT,
  CONSTRAINT pk_addon PRIMARY KEY (id)
);

CREATE TABLE feature (
  id INT AUTO_INCREMENT,
  feature_name VARCHAR(100),
  CONSTRAINT pk_addon PRIMARY KEY (id)
);

CREATE TABLE room_class_feature (
  room_class_id INT,
  feature_id INT,
```

```
  CONSTRAINT fk_rmclsft_roomclass FOREIGN KEY (room_class_id) REFERENCES
room_class (id),
  CONSTRAINT fk_rmclsft_feature FOREIGN KEY (feature_id) REFERENCES
feature (id)
);

CREATE TABLE room_status (
  id INT AUTO_INCREMENT,
  status_name VARCHAR(100),
  CONSTRAINT pk_addon PRIMARY KEY (id)
);

CREATE TABLE floor (
  id INT AUTO_INCREMENT,
  floor_number VARCHAR(5),
  CONSTRAINT pk_addon PRIMARY KEY (id)
);

CREATE TABLE room (
  id INT AUTO_INCREMENT,
  floor_id INT,
  room_class_id INT,
  status_id INT,
  room_number VARCHAR(10),
  CONSTRAINT pk_addon PRIMARY KEY (id),
  CONSTRAINT fk_room_floor FOREIGN KEY (floor_id) REFERENCES floor (id),
  CONSTRAINT fk_room_roomclass FOREIGN KEY (room_class_id) REFERENCES
room_class (id),
  CONSTRAINT fk_room_status FOREIGN KEY (status_id) REFERENCES
room_status (id)
);

CREATE TABLE booking_room (
  booking_id INT,
  room_id INT,
  CONSTRAINT fk_bkroom_booking FOREIGN KEY (booking_id) REFERENCES
booking (id),
  CONSTRAINT fk_bkroom_room FOREIGN KEY (room_id) REFERENCES room (id)
);
```

# Conclusion

I hope you found this guide useful. If you have any questions or issues with it, let me know at ben@databasestar.com.

Thanks,

Ben Brumm

www.DatabaseStar.com