

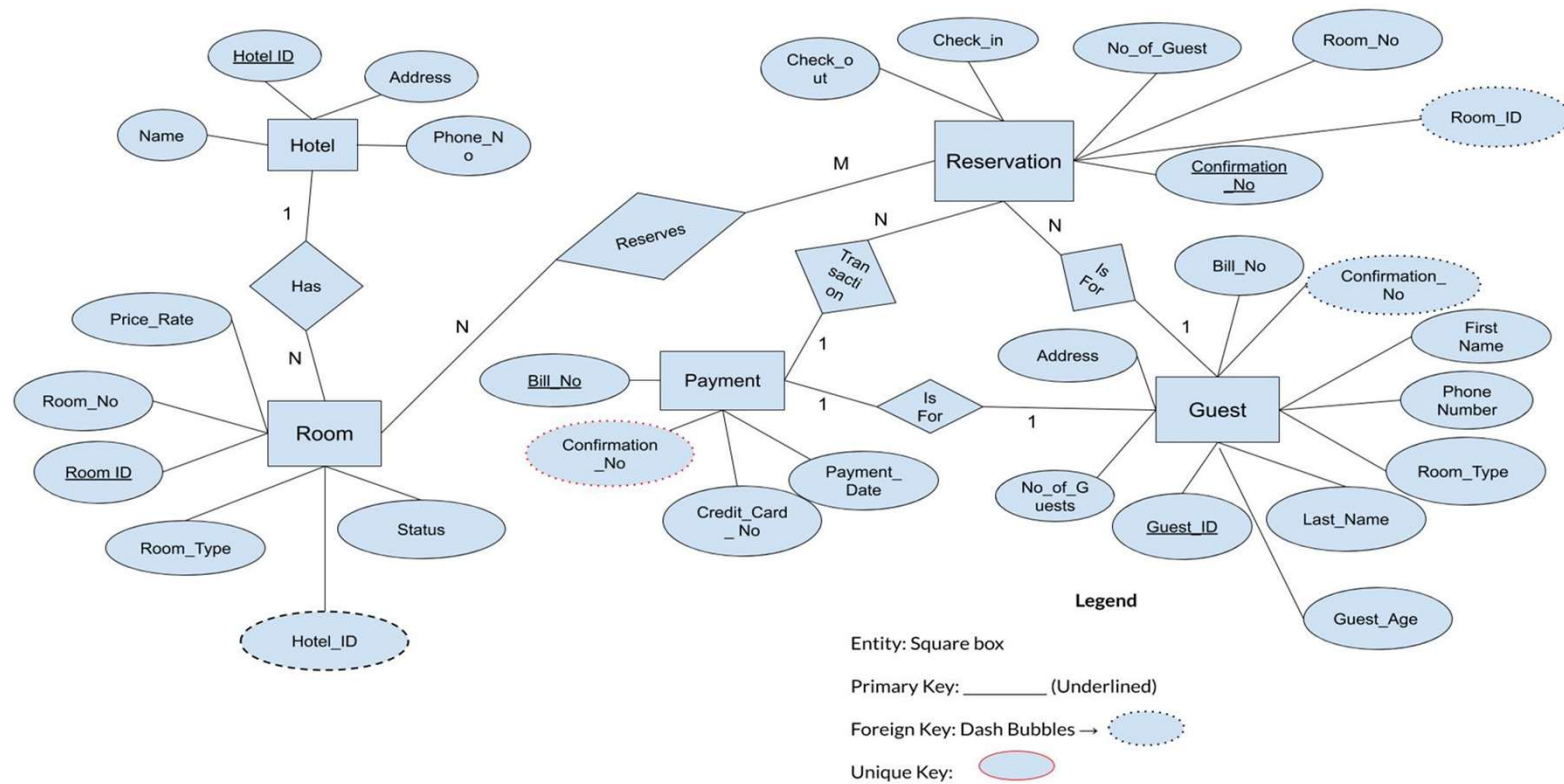
Hotel reservation and booking (customer focus)

DBMS Group Project

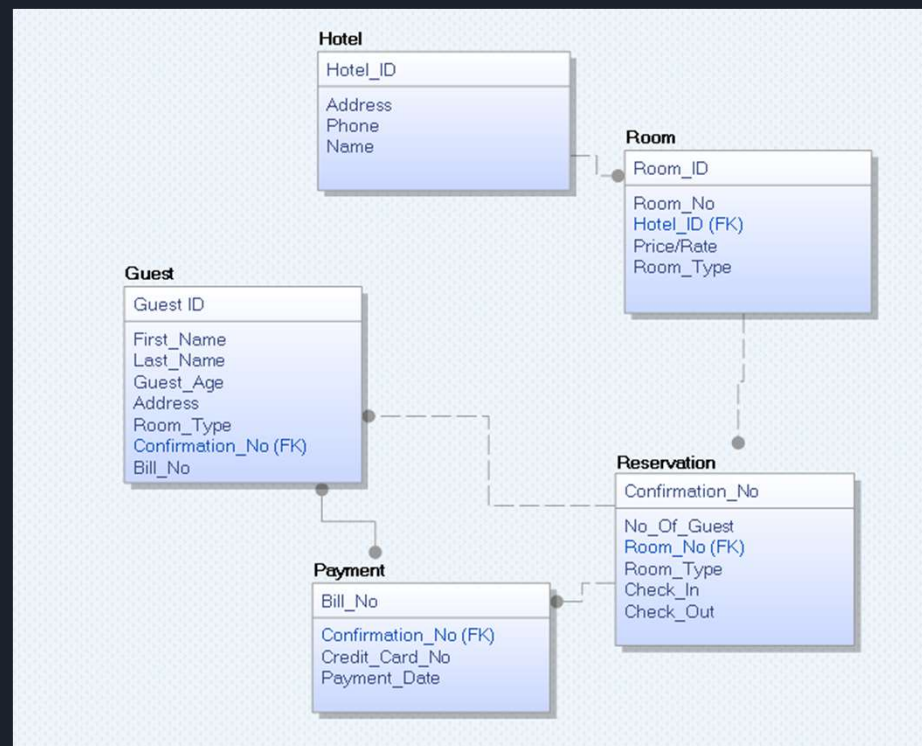
By:
Alvin Munar
Katherine Gallego



Hotel Schema



Relational Model: Implementational Schema



Code

```
CREATE TABLE Hotel(  
    Name VARCHAR(20) NOT NULL,  
    Address VARCHAR(70) NOT NULL,  
    Phone_No VARCHAR(15) NOT NULL,  
    Hotel_ID VARCHAR(3) NOT NULL,  
    PRIMARY KEY(Hotel_ID)  
);  
  
CREATE TABLE Room(  
    Room_No VARCHAR(2) NOT NULL,  
    Room_Type VARCHAR(10) NOT NULL,  
    Hotel_ID VARCHAR(3),  
    Room_ID VARCHAR(3),  
    Price_Rate DECIMAL(10, 2) NOT NULL,  
    STATUS VARCHAR(10),  
    PRIMARY KEY(Room_ID),  
    FOREIGN KEY(Hotel_ID) REFERENCES  
Hotel(Hotel_ID)  
);  
  
CREATE TABLE Reservation(  
    Confirmation_No VARCHAR(10) NOT NULL,  
    No_of_Guest INT,  
    Check_In VARCHAR(15) NOT NULL,  
    Check_Out VARCHAR(15) NOT NULL,  
    Room_No VARCHAR(2) NOT NULL,  
    Room_ID VARCHAR(3) NOT NULL,  
    PRIMARY KEY(Confirmation_No),  
    FOREIGN KEY(Room_ID) REFERENCES
```

```
);  
  
CREATE TABLE Payment(  
    Bill_No VARCHAR(10) NOT NULL,  
    Confirmation_No VARCHAR(10) NOT NULL,  
    Credit_Card_No INT(9) NOT NULL,  
    Payment_Date VARCHAR(20) NOT NULL,  
    PRIMARY KEY(Bill_No),  
    UNIQUE(Confirmation_No),  
    FOREIGN KEY(Confirmation_No)  
REFERENCES Reservation(Confirmation_No)  
);  
  
CREATE TABLE Guest(  
    Guest_ID VARCHAR(10) NOT NULL,  
    First_Name VARCHAR(15) NOT NULL,  
    Last_Name VARCHAR(15) NOT NULL,  
    Guest_Age VARCHAR(2) NOT NULL,  
    Address VARCHAR(100) NOT NULL,  
    Room_Type VARCHAR(10) NOT NULL,  
    Confirmation_No VARCHAR(10) NOT NULL,  
    PRIMARY KEY(Guest_ID),  
    FOREIGN KEY(Confirmation_No)  
REFERENCES Reservation(Confirmation_No)  
);
```

phpMyAdmin



Recent Favorites

- New
- hotel_project
 - New
 - guest
 - hotel
 - payment
 - reservation
 - room
- information_schema
- mysql
- performance_schema
- phpmyadmin

Server: 127.0.0.1 » Database: hotel_project

Structure SQL Search Query Export Import Operations Privileges Routines Events

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> guest	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> hotel	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> payment	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> reservation	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> room	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	32.0 KiB	-
5 tables	Sum	17	InnoDB	utf8mb4_general_ci	144.0 KiB	0 B

↑ ☐ Check all With selected: ▼

Print Data dictionary

Create table

Name: Number of columns:

Insert Hotel Code

```
INSERT INTO hotel  
VALUES(  
    ,(  
        'The Plaza',  
        '768 5th Ave New York, NY, 10019',  
        '1-212-759-3000',  
        'TP1'  
    ),(  
        'The Tipton Hotel',  
        'P Sherman 42 Wallaby Way, Sydney, AU',  
        '1-347-949-9850',  
        'P42'  
    ),(  
        'The Continental',  
        '2 South William Street Court, NEW York, NY, 1  
        '1-212344-0208',  
        'JWT'  
    );
```

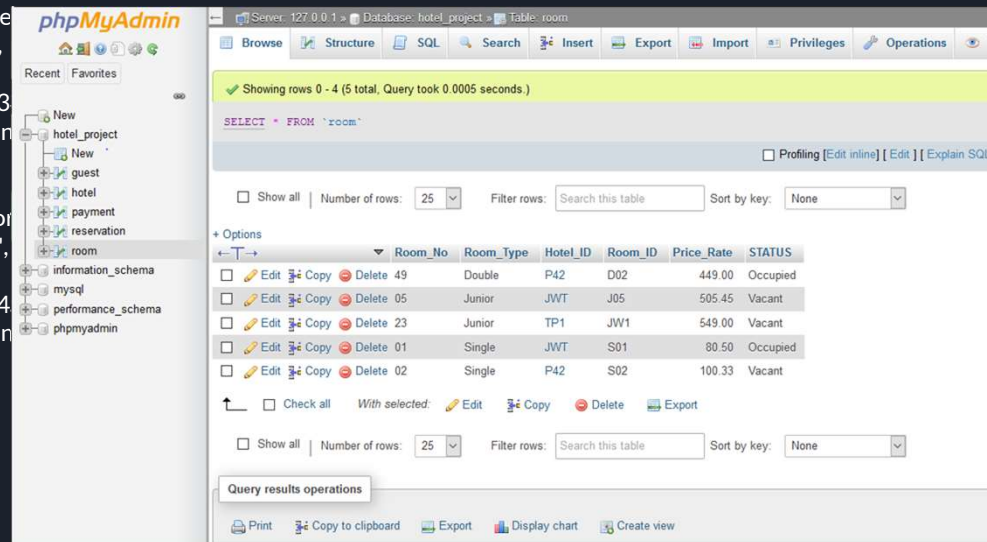
The screenshot shows the phpMyAdmin interface. On the left, the database structure tree is visible, with 'hotel' selected under 'hotel_project'. The main panel displays the 'hotel' table data. A status bar at the top indicates 'Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)'. Below this, a SQL query is shown: 'SELECT * FROM `hotel`'. The table data is as follows:

	NAME	Address	Phone_No	Hotel_ID
<input type="checkbox"/>	Edit Copy Delete	The Continental	2 South William Street Court, NEW York, NY, 10005	1-212344-0208 JWT
<input type="checkbox"/>	Edit Copy Delete	The Tipton Hotel	P Sherman 42 Wallaby Way, Sydney, AU	1-347-949-9850 P42
<input type="checkbox"/>	Edit Copy Delete	The Plaza	768 5th Ave New York, NY, 10019	1-212-759-3000 TP1

Below the table, there are options to 'Check all', 'With selected: Edit Copy Delete Export'. At the bottom, there is a 'Query results operations' section with links for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

Insert Room Code

```
INSERT INTO room (
VALUES(
    '01',
    'Single',
    'P42',
    'S02',
    '100.3',
    'Vacant' ),(
    '49',
    'Double',
    'P42',
    'D02',
    '449',
    'Occupied' );
),(
    '23',
    'Junior',
    'TP1',
    'JW1',
    '549',
    'Vacant'
),
```



The screenshot shows the phpMyAdmin interface for a database named 'hotel_project'. The 'room' table is selected, and its data is displayed in a table format. The table has 5 rows and 6 columns: Room_No, Room_Type, Hotel_ID, Room_ID, Price_Rate, and STATUS. The data is as follows:

Room_No	Room_Type	Hotel_ID	Room_ID	Price_Rate	STATUS
49	Double	P42	D02	449.00	Occupied
05	Junior	JWT	J05	505.45	Vacant
23	Junior	TP1	JW1	549.00	Vacant
01	Single	JWT	S01	80.50	Occupied
02	Single	P42	S02	100.33	Vacant

Insert reservation Code

```
INSERT INTO reservation  
VALUES(
```

```
'987456123',  
'1',  
'Sept 12, 2019',  
'Sept 20, 2019',  
'01',  
'S01'
```

```
),(
```

```
'879136802',  
'2',  
'Aug 08, 2018',  
'Aug 20, 2019',  
'49',  
'D02'
```

```
),(
```

```
'9987456210',  
'3',  
'Jan 3 2017',  
'Jan 9 2017',  
'23',  
'JW1'
```

```
);
```

The screenshot shows the phpMyAdmin interface for a database named 'hotel_project'. The 'reservation' table is selected, and its structure is displayed. The table has the following columns: Confirmation_No, No_of_Guest, Check_In, Check_Out, Room_No, and Room_ID. Three rows of data are shown:

Confirmation_No	No_of_Guest	Check_In	Check_Out	Room_No	Room_ID
879136802	2	Aug 08, 2018	Aug 20, 2019	49	D02
987456123	1	Sept 12, 2019	Sept 20, 2019	01	S01
9987456210	3	Jan 3 2017	Jan 9 2017	23	JW1

The interface also shows the 'Query results operations' section with options like Print, Copy to clipboard, Export, Display chart, and Create view. A 'Bookmark this SQL query' section is also visible at the bottom.

Insert guest Code

INSERT INTO guest
VALUES(

'123456789',
'John',
'Wick',
'42',
'12397 Winston St. New York, NY,10003',
'Single',
'987456123'

),(

'6666666521',
'Dwayne',
'Johnson',
'48',
'8820 Wilshire Blvd Suite 220 Beverly Hills

90211-2618 USA',

'Double',
'879136802'

),(

'119782135',
'Michael',
'Jordan',
'57',
'2700 Point Lane Highland Park, IL',
'Junior',
'9987456210');

The screenshot shows the phpMyAdmin interface for a database named 'hotel_project'. The 'guest' table is selected, and its data is displayed in a table format. The table has columns: Guest_ID, First_Name, Last_Name, Address, Room_Type, and Confirmation_No. There are three rows of data shown. The interface includes a sidebar with a database structure tree, a top menu bar with options like Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and a bottom section for query results operations and bookmarking.

Guest_ID	First_Name	Last_Name	Address	Room_Type	Confirmation_No
119782135	Michael	Jordan	2700 Point Lane Highland Park, IL	Junior	9987456210
123456789	John	Wick	12397 Winston St. New York, NY,10003	Single	987456123
6666666521	Dwayne	Johnson	8820 Wilshire Blvd Suite 220 Beverly Hills, CA 90211-2618 USA	Double	879136802

Insert payment Code

INSERT INTO payment
VALUES(

'345697123',
'987456123',
'654789321',
'Sept 12, 2019'

), (

'1788965431',
'879136802',
'369852147',
'Aug 8, 2018'

), (

'1127945613',
'9987456210',
'309781456',
'Jan 4, 2017'

)

The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, with 'hotel_project' expanded to show tables: 'guest', 'hotel', 'payment', 'reservation', and 'room'. The 'payment' table is selected. The main panel shows the table's contents, displaying 3 rows of data. The table has columns: 'Bill_No', 'Confirmation_No', 'Credit_Card_No', and 'Payment_Date'. The data rows are:

Bill_No	Confirmation_No	Credit_Card_No	Payment_Date
1127945613	9987456210	309781456	Jan 4, 2017
1788965431	879136802	369852147	Aug 8, 2018
345697123	987456123	654789321	Sept 12, 2019

At the bottom, the 'Query results operations' section includes links for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

Modify Code -Before

UPDATE

room

SET

STATUS

= 'Occupied'

WHERE

room.Room_ID = 'JW1'

The screenshot shows the phpMyAdmin interface for a database named 'hotel_project'. The left sidebar displays the database structure, with the 'room' table selected. The main panel shows the 'room' table structure and a list of rows. The table has columns: Room_No, Room_Type, Hotel_ID, Room_ID, Price_Rate, and STATUS. The rows are as follows:

Room_No	Room_Type	Hotel_ID	Room_ID	Price_Rate	STATUS
49	Double	P42	D02	449.00	Occupied
05	Junior	JWT	J05	505.45	Vacant
23	Junior	TP1	JW1	549.00	Vacant
01	Single	JWT	S01	80.50	Occupied
02	Single	P42	S02	100.33	Vacant

The interface also shows a query editor with the SQL query: `SELECT * FROM 'room'`. The query results are displayed below the editor, showing 5 rows. The 'room' table is currently selected, and the 'room.Room_ID = 'JW1'' condition is highlighted in the WHERE clause.

Modify Code -After

UPDATE

room

SET

STATUS

= 'Occupied'

WHERE

room.Room_ID = 'J'

The screenshot shows the phpMyAdmin interface for a database named 'hotel_project'. The 'room' table is selected, and the query results are displayed. The query is `SELECT * FROM `room``. The results show 5 rows. The row with Room_ID 'JW1' has a STATUS of 'Occupied', which is highlighted in yellow.

	Room_No	Room_Type	Hotel_ID	Room_ID	Price_Rate	STATUS
<input type="checkbox"/>	49	Double	P42	D02	449.00	Occupied
<input type="checkbox"/>	05	Junior	JWT	J05	505.45	Vacant
<input type="checkbox"/>	23	Junior	TP1	JW1	549.00	Occupied
<input type="checkbox"/>	01	Single	JWT	S01	80.50	Occupied
<input type="checkbox"/>	02	Single	P42	S02	100.33	Vacant

Delete Code - Before

DELETE

FROM

guest

WHERE

guest.Guest_ID = '11978213'

The screenshot displays the phpMyAdmin web interface. On the left, the database structure tree shows 'hotel_project' containing tables like 'guest', 'hotel', 'payment', 'reservation', 'room', 'information_schema', 'mysql', 'performance_schema', and 'phpmyadmin'. The 'guest' table is selected. The main panel shows the table's structure with columns: Guest_ID, First_Name, Last_Name, Address, Room_Type, and Confirmation_No. Below the structure, the table's data is displayed in a grid. The first row shows a guest with Guest_ID 119782135, First_Name Michael, Last_Name Jordan, Address 2700 Point Lane Highland Park, IL, Room_Type Junior, and Confirmation_No 9987456210. The second row shows a guest with Guest_ID 123456789, First_Name John, Last_Name Wick, Address 12397 Winston St. New York, NY,10003, Room_Type Single, and Confirmation_No 987456123. The third row shows a guest with Guest_ID 6666666521, First_Name Dwayne, Last_Name Johnson, Address 8820 Wilshire Blvd Suite 220 Beverly Hills, CA 902..., Room_Type Double, and Confirmation_No 879136802. The interface includes various navigation and action buttons like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', and 'Tracking'. A status bar at the bottom indicates 'Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)'.

Guest_ID	First_Name	Last_Name	Address	Room_Type	Confirmation_No
119782135	Michael	Jordan	2700 Point Lane Highland Park, IL	Junior	9987456210
123456789	John	Wick	12397 Winston St. New York, NY,10003	Single	987456123
6666666521	Dwayne	Johnson	8820 Wilshire Blvd Suite 220 Beverly Hills, CA 902...	Double	879136802

Delete Code - After

DELETE

FROM

guest

WHERE

guest.Guest_ID = '1197821'

The screenshot shows the phpMyAdmin interface for a database named 'hotel_project'. The 'guest' table is selected, and the 'Structure' tab is active. The table structure is displayed with columns: Guest_ID, First_Name, Last_Name, Address, Room_Type, and Confirmation_No. Below the structure, the 'SQL' tab is active, showing a query: `SELECT * FROM `guest``. The query results are displayed in a table with 2 rows. The first row has Guest_ID 123456789, First_Name John, Last_Name Wick, Address 12397 Winston St. New York, NY,10003, Room_Type Single, and Confirmation_No 987456123. The second row has Guest_ID 6666666521, First_Name Dwayne, Last_Name Johnson, Address 8820 Wilshire Blvd Suite 220 Beverly Hills, CA 902..., Room_Type Double, and Confirmation_No 879136802. The interface includes various navigation and action buttons like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', 'Tracking', and 'Profiling'.

Guest_ID	First_Name	Last_Name	Address	Room_Type	Confirmation_No
123456789	John	Wick	12397 Winston St. New York, NY,10003	Single	987456123
6666666521	Dwayne	Johnson	8820 Wilshire Blvd Suite 220 Beverly Hills, CA 902...	Double	879136802

Sample SQL Retrieval Code

SELECT DISTINCT

Room_Type,

Room_No

FROM

room,

hotel

The screenshot shows the phpMyAdmin interface. On the left is the database structure tree with 'hotel_project' selected. The main panel shows the 'room' table selected. A SQL query is entered in the query box: `SELECT DISTINCT Room_Type, Room_No FROM room, hotel`. The results show 5 rows: Double (49), Junior (05), Junior (23), Single (01), and Single (02). The interface includes navigation tabs (Browse, Structure, SQL, Search, Insert, Export, Import) and a 'Query results operations' section at the bottom with options like Print, Copy to clipboard, Export, Display chart, and Create view.

Server: 127.0.0.1 » Database: hotel_project » Table: room

Browse Structure SQL Search Insert Export Import

Show query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are disabled.

✓ Showing rows 0 - 4 (5 total, Query took 0.0010 seconds.)

`SELECT DISTINCT Room_Type, Room_No FROM room, hotel`

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

Room_Type	Room_No
Double	49
Junior	05
Junior	23
Single	01
Single	02

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view



Write-up/Summary

- For our DBMS project, we were assigned Hotel reservation and booking with concentration to customer focus.
- We present our project using an Entity Relationship(ER) model, which denotes the hotel schema concept and design for this database. With the Relational Model, we were able to implement our schema and create a code for our Hotel Database.
- Our mini world consists of five entities: Hotel, Room, Payment, Reservation and Guest.
- Each entities has a Primary Key.
 - Hotel = Hotel_ID
 - Room = Room_ID
 - Reservation = Confrimation_ID
 - Guest = Guest_ID
 - Payment = Bill_No
- Similarly, there were entities that had the same foreign key (FK) and unique key (U)
 - Room (FK) = Hotel_ID from Hotel
 - Reservation (FK) = Room_ID from Room
 - Payment (FK) = Confirmation_No from Reservation
 - Payment (U) = Confirmation_No
 - Guest (FK) = Confirmation_No from Reservation



Write-up/Summarize

- Each entity has its own attributes and attributes from other entities.
- For example, the Hotel entity has the attributes:
 - Hotel_ID
 - Address
 - Phone
 - Name
- Room entity has:
 - Room_No
 - Hotel_ID
 - Price/Rate
 - Room_ID
 - Room_Type



Write-up/Summarize

- Reservation:
 - Confirmation_No
 - No_of_Guest
 - Room_No
 - Room_Type
 - Check_In
 - Check_Out
 - Room_ID
- Payment:
 - Bill_No
 - Confirmation_No
 - Credit_Card_No
 - Payment_Date



Write-up/Summarize

- Guest:
 - Guest_ID
 - First_Name
 - Last_Name
 - Address
 - Bill_No
 - Room_Type
 - Confirmation_No
- In the ER diagram, each Primary key attribute is underlined inside the oval as illustrated and primary keys that appear in a different entity becomes a foreign key linking entities together.
- Primary and foreign key is unique because it cannot be repeated.
- Similarly, attributes that are primary key from a different entity which appears in another entity is known as a foreign key.



Write-up/Summarize

- We realized this mini world has a weak entity, known as Room.
- The existence of room is entirely dependent on the existence of the hotel.
- In this mini world, we chose attributes from entities that cannot have null values.
- Null values in a table can be left blank and by using common sense we decided which attributes that cannot have null values.
- For example, in the Hotel entity we chose the attribute Name to be NOT NULL because the name of the hotel is important in a sense and leaving it blank wouldn't make sense.
- Similarly, in this mini world, there are cardinality ratios. These ratios specifies in the maximum number of relationship instances that an entity can participate in.
- For example, a 1:N relationship is represented with the Hotel entity having the Room entity, a particular Hotel (1) can have many rooms(N) .
- On the other hand, the relationship ratio for Reservation and Rooms is (M:N), because you can have as many reservations and as rooms to accomodate guests.



Write-up/Summarize

- Insertion Code was performed to insert data into the attributes of the entities.

```
INSERT INTO hotel
VALUES(
    'The Plaza',
    '768 5th Ave New York, NY, 10019',
    '1-212-759-3000',
    'TP1'
)
```



Write-up/Summarize

- In order to modify the code, we chose to modify the status of the room from VACANT to OCCUPIED.

The following code was performed to:

```
UPDATE
room
SET
STATUS
    = 'Occupied'
WHERE
    room.Room_ID = 'JW1';
```



Write-up/Summarize

- In order to delete data from an attribute, we chose to delete the information for guest : Michael Jordan.

The following code was performed to: Deletion Code

```
DELETE  
FROM  
      guest  
WHERE  
      guest.Guest_ID = '119782135'
```



Write-up/Summarize

- To sample our SQL code, we used the following retrieval codes for the database to test:

```
SELECT DISTINCT
    Room_Type,
    Room_No
FROM
    room, hotel
```

- After each retrieval code entered in SQL, we were able to successfully retrieve information from our database.

Room_Type	Room_No
Double	49
Junior	05
Junior	23
Single	01
Single	02

Stay Safe!!!

