# Siddaganga Institute of Technology, Tumakuru

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi, Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified)

## DATABASE MANAGEMENT SYSTEM

On

# HOTEL DATABASE MANAGEMENT SYSTEM

submitted
*in the partial fulfillment of the requirements V semester*
*Bachelor of Engineering*
In
***Computer Science and Engineering***
By


**ADITYA RANJAN 1SI20CS005**
**ANKIT KUMAR 1SI20CS013**
**ANSHIKA TYAGI 1SI20CS015**
**BIPUL KUMAR 1SI20CS024**




**Department of Computer Science & Engineering**
(Program Accredited by NBA)
**Siddaganga Institute of Technology**
B.H Road, Tumakuru-572 103, Karnataka, India.
Web: www.sit.ac.in

# Siddaganga Institute of Technology, Tumakuru

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi,

Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified)

## Department of Computer Science and Engineering

(Program Accredited by NBA)


# CERTIFICATE

This is to certify that the group activity entitled "HOTEL DATABASE MANAGEMENT SYSTEM" is a bona fide work carried out by **ADITYA RANJAN -1SI20CS005, ANKIT KUMAR – 1SI20CS013, ANSHIKA TYAGI – 1SI20CS015, and BIPUL KUMAR -1SI20CS024** of V semester **Bachelor of Engineering in Computer Science and Engineering** of the **SIDDAGANGA INSTITUTE OF TECHNOLOGY** (An Autonomous Institution, affiliated to VTU, Belagavi, Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified) during the academic year 2022-2023.

**Name of Faculty** **Signature**

**DR . K SRINIVASA**

# PROBLEM STATEMENT

Nowadays hotels and restaurants are committed to bringing the best way of management in various forms. In every part of the hotel like the front office, booking, and reservation, inventory, material management, quality management, security, energy management, housekeeping, CRM, and more., there is a need for software that must provide efficient and faster completion of work. Among the various methods of maintaining records of students most used once are maintaining ledgers or excel files which are commonly used in organizations. These methods are time-consuming and inefficient. Hence an alternative is needed. The alternative must need to have a few objectives according to the requirements of the organization.

## Objectives

- Create an end-to-end solution that enables a hotel to manage its operations efficiently and effectively.
- The system should be designed to automate routine tasks, reduce manual errors, and provide real-time visibility into hotel operations.
- It should also provide a comprehensive suite of features, including online booking, room allocation, inventory management, billing and invoicing, housekeeping, and maintenance.
- It should be designed to meet the needs of different stakeholders, including hotel staff, management, guests, and vendors.
- The system should also be scalable, modular, and customizable, allowing hotels to add or remove features as per their changing needs.
- The system should prioritize security and data privacy, with robust access control mechanisms, data encryption, and backup and recovery capabilities.
- Overall, the hotel database management system should enable hotels to optimize their operations, enhance guest experiences, and achieve business growth and profitability.

# REQUIREMENT COLLECTION

Requirements collected for a Hotel Database Management System:

Room Management: A system to manage room inventory, pricing, and availability, including room types, rates, and restrictions.

Reservation Management: A system to manage guest reservations, including check-in and check-out dates, guest information, room preferences, and room assignments.

Guest Management: A system to manage guest information, including contact details, preferences, services used, and special requests.

Housekeeping Management: A system to manage housekeeping schedules, including cleaning schedules, room status updates, and maintenance requests.

Front Desk Management: A system to manage front desk operations, including check-in and check-out, room changes, and guest requests.

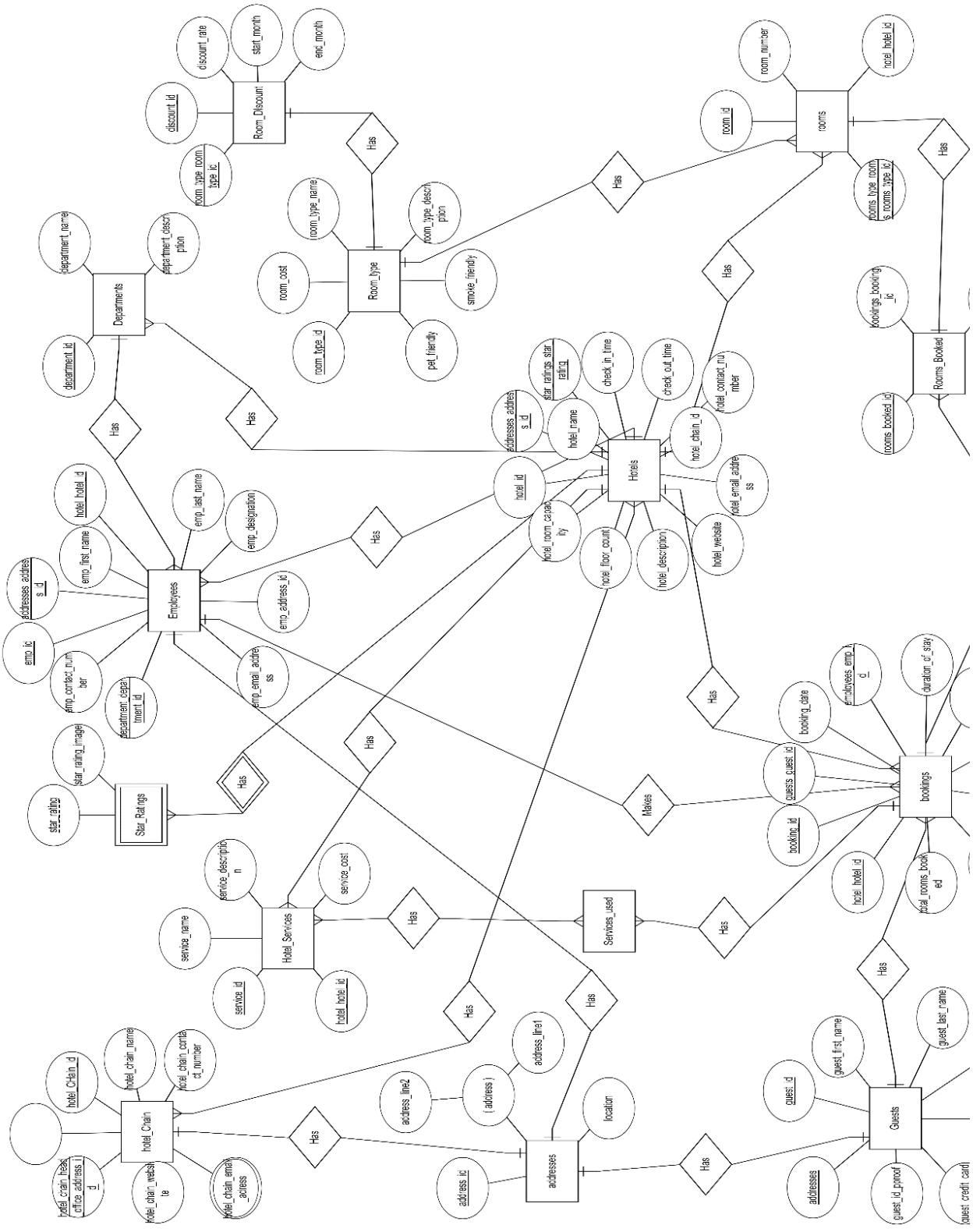
Inventory Management: A system to manage inventory of supplies, such as linens, toiletries, and other items, and track inventory levels, usage, and ordering.

Employee Management: A system to manage employee schedules, including shift assignments, time off requests, and payroll management.

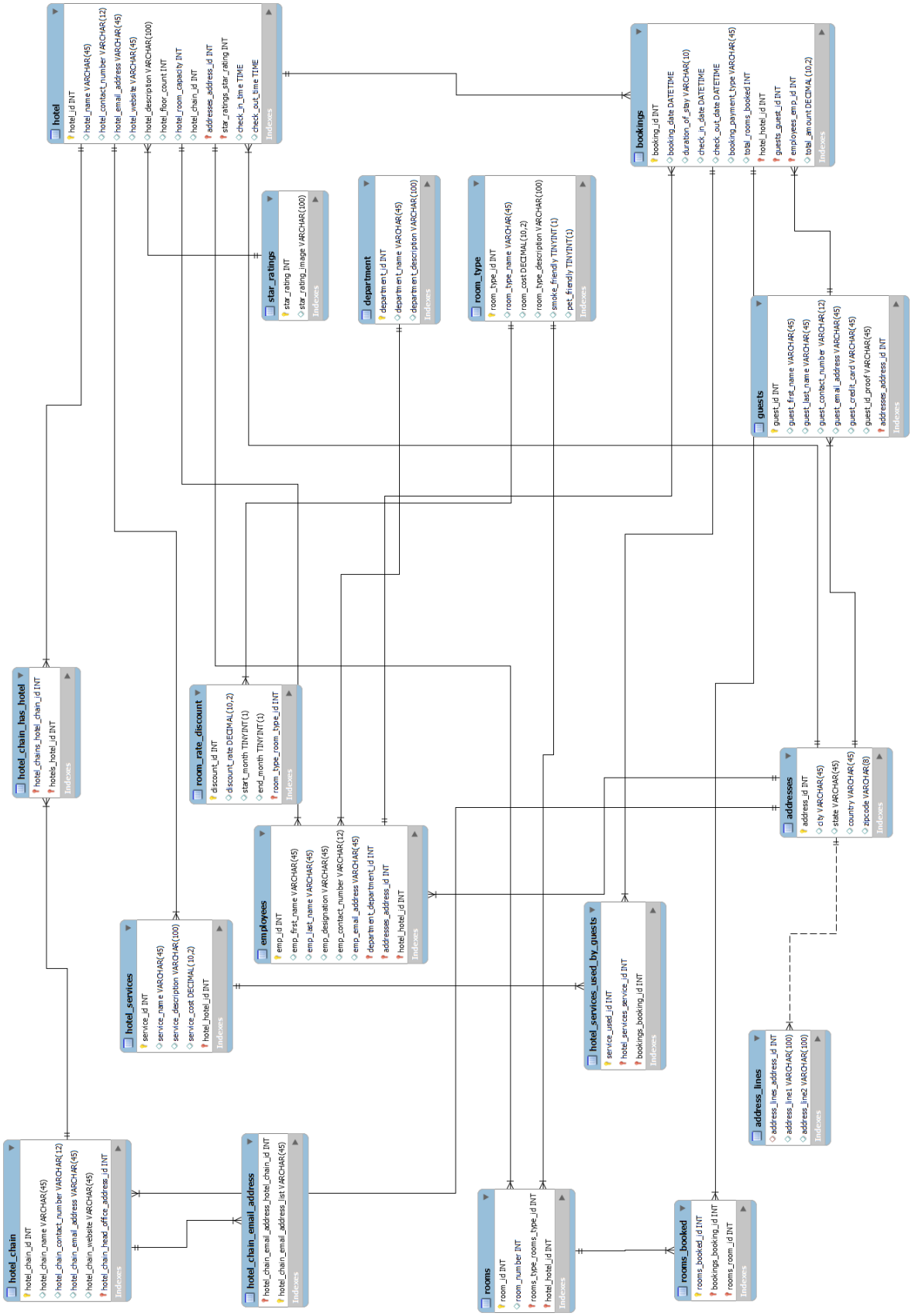
Reporting and Analytics: A system to generate reports and analytics on hotel performance, including ratings and guest satisfaction.

Room Service Management: A system to manage room service orders, including order taking, preparation, and delivery.

# ER MODEL


Departments
department_id
department_name
department_description
Has
Room_type
room_type_id
room_type_name
room_type_description
room_cost
smoke_friendly
pet_friendly
Room_Discount
discount_id
discount_rate
start_month
end_month
room_type_room_type_id
rooms
room_id
room_number
hotel_hotel_id
rooms_type_rooms_type_id
Rooms_Booked
rooms_booked_id
bookings_booking_id
Hotels
hotel_id
hotel_name
hotel_chain_id
hotel_contact_number
hotel_email_address
hotel_website
hotel_description
hotel_floor_count
hotel_room_capacity
check_in_time
check_out_time
star_ratings_star_rating
addresses_address_id
Employees
emp_id
emp_first_name
emp_last_name
emp_designation
emp_contact_number
emp_email_address
emp_address_id
department_department_id
addresses_address_id
hotel_hotel_id
Star_Ratings
star_rating
star_rating_image
Hotel_Services
service_id
service_name
service_description
service_cost
hotel_hotel_id
Services_used
Makes
bookings
booking_id
guess_guest_id
booking_date
duration_of_stay
total_rooms_booked
employees_emp_id
hotel_hotel_id
Guests
guest_id
guest_first_name
guest_last_name
guest_id_proof
guest_credit_card
addresses
address_id
address_line1
address_line2
location
hotel_Chain
hotel_Chain_id
hotel_chain_name
hotel_chain_contact_number
hotel_chain_website
hotel_chain_email_address
hotel_chain_head_office_address_id

# RELATIONAL MAPPING



# NORMALIZATION

## First Normal Form

To normalize a hotel database management system up to 1NF, the following steps can be taken:

Identify the entities and relationships in the system, such as guests, rooms, reservations, and payments.

Create an initial table structure for each entity, with columns for the attributes of each entity, such as guest name, room type, booking date, and payment amount.

Analyze the tables for any repeating groups, which occur when two or more values of the same attribute are included in a single row. For example, a guest may have multiple phone numbers or multiple reservations may be associated with a single guest.

For each repeating group, create a separate table with a foreign key to the original table, and move the repeating attribute(s) to the new table. For example, create a separate table for phone numbers with a foreign key to the guest table.

Repeat step 4 for any other repeating groups in the system, creating new tables and foreign keys as necessary.

Ensure that each table has a primary key that uniquely identifies each row and that no two rows have identical primary keys.

By following these steps, a hotel database management system can be normalized up to 1NF, ensuring that data is stored efficiently and accurately, and reducing the risk of data anomalies and inconsistencies.

## Second Normal Form

To normalize a hotel database management system up to 2NF, the following steps can be taken after the system has been normalized up to 1NF:

Identify the primary key of each table, which should uniquely identify each row in the table.

Identify any columns that are functionally dependent on only part of the primary key. These columns should be moved to a separate table with a foreign key to the original table.

Create a new table for each set of columns that are functionally dependent on only part of the primary key, and move these columns to the new table along with the corresponding portion of the primary key.

Add a foreign key to the new table that references the original table's primary key.

Ensure that each table has a primary key that uniquely identifies each row and that no two rows have identical primary keys.

By following these steps, a hotel database management system can be normalized up to 2NF, ensuring that data is stored efficiently and accurately, and reducing the risk of data anomalies and inconsistencies.

## Third Normal Form

To normalize a hotel database management system up to 3NF, the following steps can be taken after the system has been normalized up to 2NF:

Identify any columns that are functionally dependent on other non-key columns in the same table.

Create a new table for each set of dependent columns, with a foreign key to the original table's primary key.

Remove the dependent columns from the original table and add a foreign key to the new table.

Ensure that each table has a primary key that uniquely identifies each row and that no two rows have identical primary keys.

By following these steps, a hotel database management system can be normalized up to 3NF, ensuring that data is stored efficiently and accurately, and reducing the risk of data anomalies and inconsistencies.

## BCNF

To normalize a hotel database management system up to BCNF (Boyce-Codd Normal Form), the following steps can be taken after the system has been normalized up to 3NF:

Identify any functional dependencies where the determinant is not a candidate key.

Create a new table for each set of dependent attributes that violates BCNF, with a foreign key to the original table's primary key.

Remove the dependent attributes from the original table and add a foreign key to the new table.

Ensure that each table has a primary key that uniquely identifies each row and that no two rows have identical primary keys.

Repeat the above steps until all tables in the database satisfy BCNF.

By following these steps, a hotel database management system can be normalized up to BCNF, ensuring that data is stored efficiently and accurately, and reducing the risk of data anomalies and inconsistencies.

# DDL STATEMENTS AND INSERTION STATEMENTS

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,
ALLOW_INVALID_DATES';


-- -----------------------------------------------------

-- hotel_database

-- -----------------------------------------------------


CREATE DATABASE IF NOT EXISTS `hotel_database` ;

USE `hotel_database` ;


-- -----------------------------------------------------

-- Table `hotel_database`.`addresses`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `hotel_database`.`addresses` ;


CREATE TABLE IF NOT EXISTS `hotel_database`.`addresses` (

  `address_id` INT NOT NULL,

  `city` VARCHAR(45) NULL,

  `state` VARCHAR(45) NULL,
```

```
  `country` VARCHAR(45) NULL,
  `zipcode` VARCHAR(8) NULL,
  PRIMARY KEY (`address_id`));


INSERT INTO `addresses`(`address_id`, `city`, `state`, `country`, `zipcode`)
VALUES
        (1,'Kitchener','ON','Canada','N2C 2P6'),
        (2, 'Kitchener','ON','Canada','N2C 2M6'),
        (3, 'London','ON','Canada','N2C 2K3'),
        (4, 'London','ON','Canada','N2A 0E4'),
        (5, 'Guelph','ON','Canada','N2C 2E8'),
        (6, NULL,'AZ','USA','AZ85027'),
        (7, 'Surrey','BC','Canada','V3W 5B4'),
        (8, 'Globe','AZ','USA','85501'),
        (9, 'Mumbai','Maharashtra','India','534076'),
        (10, 'Saskatoon','SK','Bangladesh','S2L 562'),
        (11, 'Kitchener','ON','Canada','Sd3 d35'),
        (12, 'London','ON','Canada','234 987'),
        (13, 'Paris','ON','Canada','467 289'),
        (14, 'Ottawa','BC','Canada','263 987'),
        (15, 'Guelph','BC','Canada','267 387'),
        (16, NULL,'AZ','USA','263 762'),
        (17,'Ottawa','New York','USA','756 145'),
```

```
        (18, 'NULL','San Jose','USA','675 846'),
        (19, 'NUll','Gujarat','India','145 895');

-- -----------------------------------------------------
-- Table `hotel_database`.`address_lines`
-- -----------------------------------------------------

DROP TABLE IF EXISTS `hotel_database`.`address_lines` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`address_lines` (
  `address_lines_address_id` INT,
  `address_line1` VARCHAR(100) NULL,
  `address_line2` VARCHAR(100) NULL,
  CONSTRAINT `fk_address_lines_address_id1`
    FOREIGN KEY (`address_lines_address_id`)
    REFERENCES `hotel_database`.`addresses` (`address_id`));

INSERT INTO `address_lines`(`address_lines_address_id`, `address_line1`, `address_line2`)
VALUES
        (1,49, 'Dave Street'),
        (2,64, 'Victoria Street'),
        (3,79, 'Connaught Street'),
        (4,45, 'Sweden St. Street'),
        (5,60, 'Lincoln Street'),
```

```
(6,20400, 'Phoenix'),
(7,8033, 'King George Boulevard'),
(8,1565, 'E South St'),
(9,32, ' Gandhi Road'),
(10,706, 'Idle rd'),
(11,45, 'Vanier Park'),
(12,41, 'Greenfield'),
(13,89, 'Jacob Rd'),
(14,85, 'Martin Street'),
(15,78, 'Josseph St. Street'),
(16,156, 'James Road'),
(17,7598, 'Atomic Street'),
(18,5476, 'Saint Jake Rd'),
(19,7465, 'Thames Rd');

-- -----------------------------------------------------
-- Table `hotel_database`.`hotel_chain`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`hotel_chain` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`hotel_chain` (
  `hotel_chain_id` INT NOT NULL,
  `hotel_chain_name` VARCHAR(45) NULL,
  `hotel_chain_contact_number` VARCHAR(12) NULL,
```

```
  `hotel_chain_email_address` VARCHAR(45) NULL,
  `hotel_chain_website` VARCHAR(45) NULL,
  `hotel_chain_head_office_address_id` INT NOT NULL,
  PRIMARY KEY (`hotel_chain_id`, `hotel_chain_head_office_address_id`),
  CONSTRAINT `fk_hotel_chains_addresses1`
    FOREIGN KEY (`hotel_chain_head_office_address_id`)
    REFERENCES `hotel_database`.`addresses` (`address_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);


INSERT INTO `hotel_chain`(`hotel_chain_id`, `hotel_chain_name`,
`hotel_chain_contact_number`, `hotel_chain_website`, `hotel_chain_head_office_address_id`)
VALUES
        (1,'Best Western Hotels','456-865-8956','https://www.bestwestern.com/',6),
        (2,'China Town Hotels','110-526-5647','https://www.chinatown.com/',16),
        (3,'Elite Hotels','546-874-6547','https://www.elitendhe.com/',17),
        (4,'Cosmopolitan Hotels','852-741-9765','https://www.cosmopolitan.com/',18),
        (5,'Prestige Hotels','657-784-3647','https://www.prestige.com/',19);


-- -----------------------------------------------------
-- Table `hotel_database`.`hotel_chain_email_address`
-- -----------------------------------------------------


DROP TABLE IF EXISTS `hotel_database`.`hotel_chain_email_address` ;
```

```
CREATE TABLE IF NOT EXISTS `hotel_database`.`hotel_chain_email_address` (
  `hotel_chain_email_address_hotel_chain_id` INT NOT NULL,
  `hotel_chain_email_address_list` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`hotel_chain_email_address_hotel_chain_id`,
`hotel_chain_email_address_list`),
  CONSTRAINT `fk_hotel_chain_id1`
    FOREIGN KEY (`hotel_chain_email_address_hotel_chain_id`)
    REFERENCES `hotel_database`.`hotel_chain` (`hotel_chain_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);


INSERT INTO `hotel_chain_email_address`(`hotel_chain_email_address_hotel_chain_id`,
`hotel_chain_email_address_list`)
VALUES
        (1,'bw765@gmail.com'),
    (1,'bw123@gmail.com'),
        (2,'chinatown123@gmail.com'),
        (3,'elite.tea213@gmail.com'),
        (4,'cosmo.hotels123@gmail.com'),
    (4,'cosmo.hotels248@gmail.com'),
        (5,'prestige2453@gmail.com');


CREATE INDEX `fk_hotel_chains_addresses1_idx` ON `hotel_database`.`hotel_chain`
(`hotel_chain_head_office_address_id` ASC);
```

```
CREATE TABLE IF NOT EXISTS `hotel_database`.`hotel_chain_email_address` (
  `hotel_chain_email_address_hotel_chain_id` INT NOT NULL,
  `hotel_chain_email_address_list` VARCHAR(45) NULL,
  PRIMARY KEY (`hotel_chain_email_address_hotel_chain_id`,
`hotel_chain_email_address_list`),
  CONSTRAINT `fk_hotel_chain_id1`
    FOREIGN KEY (`hotel_chain_email_address_hotel_chain_id`)
    REFERENCES `hotel_database`.`hotel_chain` (`hotel_chain_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);


-- -----------------------------------------------------
-- Table `hotel_database`.`hotel_chain_email_address`
-- -----------------------------------------------------


-- -----------------------------------------------------
-- Table `hotel_database`.`star_ratings`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`star_ratings` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`star_ratings` (
  `star_rating` INT NOT NULL,
  `star_rating_image` VARCHAR(100) NULL,
  PRIMARY KEY (`star_rating`));
```

```
-- insert into star ratings table
INSERT INTO `star_ratings`(`star_rating`, `star_rating_image`)
VALUES
	(1,"/images/one_star.jpg"),
	(2,"/images/two_star.jpg"),
	(3,"/images/three_star.jpg"),
	(4,"/images/four_star.jpg"),
	(5,"/images/five_star.jpg");



-- -----------------------------------------------------
-- Table `hotel_database`.`hotel`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`hotel` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`hotel` (
  `hotel_id` INT NOT NULL,
  `hotel_name` VARCHAR(45) NULL,
  `hotel_contact_number` VARCHAR(12) NULL,
  `hotel_email_address` VARCHAR(45) NULL,
  `hotel_website` VARCHAR(45) NULL,
  `hotel_description` VARCHAR(100) NULL,
  `hotel_floor_count` INT NULL,
  `hotel_room_capacity` INT NULL,
```

```
  `hotel_chain_id` INT NULL,
  `addresses_address_id` INT NOT NULL,
  `star_ratings_star_rating` INT NOT NULL,
  `check_in_time` TIME NULL,
  `check_out_time` TIME NULL,
  PRIMARY KEY (`hotel_id`, `addresses_address_id`, `star_ratings_star_rating`),
  CONSTRAINT `fk_hotels_addresses1`
    FOREIGN KEY (`addresses_address_id`)
    REFERENCES `hotel_database`.`addresses` (`address_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_hotel_star_ratings1`
    FOREIGN KEY (`star_ratings_star_rating`)
    REFERENCES `hotel_database`.`star_ratings` (`star_rating`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);


INSERT INTO `hotel`(`hotel_id`, `hotel_name`, `hotel_contact_number`, `hotel_email_address`,
`hotel_website`, `hotel_description`, `hotel_floor_count`, `hotel_room_capacity`,
`hotel_chain_id`, `addresses_address_id`, `star_ratings_star_rating`, `check_in_time`,
`check_out_time`)
        VALUES
                (1,'King George Inn & Suites','604-502-
9564','kgi123@gmail.com','https://www.kgi123.com/','A 2-mile drive from Besh Ba Gowah
Archaeological Park.',5,45,1,7,4,'12:00:00','23:00:00'),
```

```
(2,'Copper Hills Inn','547-964-
9564','chinni123@gmail.com','https://www.chin23.com/','A 2-mile drive from Besh Ba Gowah
Archaeological Park.',6,55,1,8,5,'12:00:00','23:00:00'),

(3,'Sawmill Inn','547-964-
3452','sawmill.inn@gmail.com','https://www.chin23.com/','A 3-mile drive from Fairview
Park.',4,50,1,9,5,'12:00:00','23:00:00'),

(4,'Northgate Inn','547-876-
5422','northgate.inn@gmail.com','https://www.chin23.com/','A 4-mile drive from Conestoga
Mall',3,40,1,10,5,'12:00:00','23:00:00');


CREATE INDEX `fk_hotels_addresses1_idx` ON `hotel_database`.`hotel`
(`addresses_address_id` ASC);


CREATE INDEX `fk_hotel_star_ratings1_idx` ON `hotel_database`.`hotel`
(`star_ratings_star_rating` ASC);



-- -----------------------------------------------------

-- Table `hotel_database`.`room_type`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `hotel_database`.`room_type` ;


CREATE TABLE IF NOT EXISTS `hotel_database`.`room_type` (

  `room_type_id` INT NOT NULL,

  `room_type_name` VARCHAR(45) NULL,

  `room_cost` DECIMAL(10,2) NULL,
```

```
  `room_type_description` VARCHAR(100) NULL,
  `smoke_friendly` TINYINT(1) NULL,
  `pet_friendly` TINYINT(1) NULL,
  PRIMARY KEY (`room_type_id`));

INSERT INTO `room_type`(`room_type_id`, `room_type_name`, `room_cost`,
`room_type_description`, `smoke_friendly`, `pet_friendly`)
 VALUES
        (1, 'Standard Room','103',"1 King Bed 323-sq-foot (30-sq-meter) room with city
views",0,1),
        (2, 'Standard Twin Room','123',"Two Twin Bed 323-sq-foot (30-sq-meter) room with city
views",1,1),
        (3, 'Executive Room','130',"1 King Bed 323-sq-foot (30-sq-meter) room with city
views",0,0),
        (4, 'Club Room','159',"2 King Bed 323-sq-foot (30-sq-meter) room with city views",1,1);



-- -----------------------------------------------------
-- Table `hotel_database`.`rooms`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`rooms` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`rooms` (
  `room_id` INT NOT NULL,
  `room_number` INT(4) NULL,
```

```
 `rooms_type_rooms_type_id` INT NOT NULL,
 `hotel_hotel_id` INT NOT NULL,
 PRIMARY KEY (`room_id`, `rooms_type_rooms_type_id`, `hotel_hotel_id`),
 CONSTRAINT `fk_rooms_rooms_type1`
   FOREIGN KEY (`rooms_type_rooms_type_id`)
   REFERENCES `hotel_database`.`room_type` (`room_type_id`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION,
 CONSTRAINT `fk_rooms_hotel1`
   FOREIGN KEY (`hotel_hotel_id`)
   REFERENCES `hotel_database`.`hotel` (`hotel_id`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION);


INSERT INTO `rooms`(`room_id`, `room_number`, `rooms_type_rooms_type_id`,
`hotel_hotel_id`)
 VALUES
        (1,1101,1,1),
        (2,1102,1,1),
        (3,1103,1,1),
        (4,1104,1,1),
        (5,1105,1,1),
        (6,1106,1,1),
        (7,1107,1,1),
```

(8,1108,1,1),

(9,1109,1,1),

(10,1110,1,1),

(11,1111,1,1),

(12,1112,1,1),

(13,1113,1,1),

(14,1114,1,1),

(15,1115,1,1),

(16,1116,1,1),

(17,1117,2,1),

(18,1118,2,1),

(19,1119,2,1),

(20,1120,2,1),

(21,1121,2,1),

(22,1122,2,1),

(23,1123,2,1),

(24,1124,2,1),

(25,1125,2,1),

(26,1126,2,1),

(27,1127,2,1),

(28,1128,2,1),

(29,1129,2,1),

(30,1130,2,1),

(31,1131,2,1),

```
(32,1132,2,1),
(33,1133,2,1),
(34,1134,2,1),
(35,1135,2,1);

CREATE INDEX `fk_rooms_rooms_type1_idx` ON `hotel_database`.`rooms`
(`rooms_type_rooms_type_id` ASC);

CREATE INDEX `fk_rooms_hotel1_idx` ON `hotel_database`.`rooms` (`hotel_hotel_id` ASC);

-- -----------------------------------------------------
-- Table `hotel_database`.`guests`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`guests` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`guests` (
  `guest_id` INT NOT NULL,
  `guest_first_name` VARCHAR(45) NULL,
  `guest_last_name` VARCHAR(45) NULL,
  `guest_contact_number` VARCHAR(12) NULL,
  `guest_email_address` VARCHAR(45) NULL,
  `guest_credit_card` VARCHAR(45) NULL,
  `guest_id_proof` VARCHAR(45) NULL,
```

```
  `addresses_address_id` INT NOT NULL,
  PRIMARY KEY (`guest_id`, `addresses_address_id`),
  CONSTRAINT `fk_guests_addresses1`
    FOREIGN KEY (`addresses_address_id`)
    REFERENCES `hotel_database`.`addresses` (`address_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);


INSERT INTO `guests`(`guest_id`, `guest_first_name`, `guest_last_name`,
`guest_contact_number`, `guest_email_address`, `guest_credit_card`, `guest_id_proof`,
`addresses_address_id`)
VALUES
        (1,'Jane','Doe','132-456-
8564','jane.doe@gmail.com',NULL,'/images/drivingLicense1023',1),
        (2,'Jerry','Thachter','564-896-
4752','jerry.ytsvg@gmail.com',NULL,'/images/passport45612',2),
        (3,'Rihanna','Perry','745-986-
7451','rih.vfdj89@gmail.com',NULL,'/images/drivingLicense4889',3),
        (4,'Mathew','Jose','489-624-
8633','mathew.jose@gmail.com',NULL,'/images/drivingLicense8945',4),
        (5,'Jessica','Smith','487-956-
8963','jess.smith@gmail.com',NULL,'/images/passport7896',5);


CREATE INDEX `fk_guests_addresses1_idx` ON `hotel_database`.`guests`
(`addresses_address_id` ASC);
```

```
-- -----------------------------------------------------
-- Table `hotel_database`.`department`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`department` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`department` (
  `department_id` INT NOT NULL,
  `department_name` VARCHAR(45) NULL,
  `department_description` VARCHAR(100) NULL,
  PRIMARY KEY (`department_id`));

INSERT INTO `department`(`department_id`, `department_name`, `department_description`)
VALUES
        (1,'Kitchen','cooking'),
        (2,'Cleaning','sweep and mop'),
        (3,'Front Staff','handle bookings and query resolution'),
        (4,'Management','handles customer and resolve complaints'),
        (5,'Commute','pick up and drop');


-- -----------------------------------------------------
-- Table `hotel_database`.`employees`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`employees` ;
```

```
CREATE TABLE IF NOT EXISTS `hotel_database`.`employees` (
  `emp_id` INT NOT NULL,
  `emp_first_name` VARCHAR(45) NULL,
  `emp_last_name` VARCHAR(45) NULL,
  `emp_designation` VARCHAR(45) NULL,
  `emp_contact_number` VARCHAR(12) NULL,
  `emp_email_address` VARCHAR(45) NULL,
  `department_department_id` INT NOT NULL,
  `addresses_address_id` INT NOT NULL,
  `hotel_hotel_id` INT NOT NULL,
  PRIMARY KEY (`emp_id`, `department_department_id`, `addresses_address_id`,
`hotel_hotel_id`),
  CONSTRAINT `fk_employees_services1`
    FOREIGN KEY (`department_department_id`)
    REFERENCES `hotel_database`.`department` (`department_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_employees_addresses1`
    FOREIGN KEY (`addresses_address_id`)
    REFERENCES `hotel_database`.`addresses` (`address_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_employees_hotel1`
    FOREIGN KEY (`hotel_hotel_id`)
```

```
  REFERENCES `hotel_database`.`hotel` (`hotel_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);


INSERT INTO `employees`(`emp_id`, `emp_first_name`, `emp_last_name`, `emp_designation`,
`emp_contact_number`, `emp_email_address`, `department_department_id`,
`addresses_address_id`, `hotel_hotel_id`)
 VALUES
 (1,'Jen','Fen','Waiter','123-789-7896','jen.rds@gmail.com',1,11,1),
 (2,'Tom','Pitt','Manager','565-789-7896','tom.pit@gmail.com',3,12,1),
 (3,'David','Lawrence','Cashier','852-789-7896','david.lawr@gmail.com',2,13,1),
 (4,'Joseph','Aniston','Cook','765-789-7896','joseph.anis@gmail.com',2,14,1),
 (5,'Jeny','Patel','Manager','531-789-7896','jeny.patel@gmail.com',3,15,1);


CREATE INDEX `fk_employees_services1_idx` ON `hotel_database`.`employees`
(`department_department_id` ASC);


CREATE INDEX `fk_employees_addresses1_idx` ON `hotel_database`.`employees`
(`addresses_address_id` ASC);


CREATE INDEX `fk_employees_hotel1_idx` ON `hotel_database`.`employees`
(`hotel_hotel_id` ASC);
```

```
-- -----------------------------------------------------
-- Table `hotel_database`.`bookings`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`bookings` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`bookings` (
  `booking_id` INT NOT NULL,
  `booking_date` DATETIME NULL,
  `duration_of_stay` VARCHAR(10) NULL,
  `check_in_date` DATETIME NULL,
  `check_out_date` DATETIME NULL,
  `booking_payment_type` VARCHAR(45) NULL,
  `total_rooms_booked` INT NULL,
  `hotel_hotel_id` INT NOT NULL,
  `guests_guest_id` INT NOT NULL,
  `employees_emp_id` INT NOT NULL,
  `total_amount` DECIMAL(10,2) NULL,
  PRIMARY KEY (`booking_id`, `hotel_hotel_id`, `guests_guest_id`, `employees_emp_id`),
  CONSTRAINT `fk_bookings_hotel1`
    FOREIGN KEY (`hotel_hotel_id`)
    REFERENCES `hotel_database`.`hotel` (`hotel_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_bookings_guests1`
```

```
  FOREIGN KEY (`guests_guest_id`)
  REFERENCES `hotel_database`.`guests` (`guest_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
 CONSTRAINT `fk_bookings_employees1`
  FOREIGN KEY (`employees_emp_id` )
  REFERENCES `hotel_database`.`employees` (`emp_id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);
```

```
INSERT INTO `bookings` (`booking_id`, `booking_date`, `duration_of_stay`, `check_in_date`, `check_out_date`, `booking_payment_type`, `total_rooms_booked`, `hotel_hotel_id`, `guests_guest_id`, `employees_emp_id`, `total_amount`)
 VALUES
        ('1', '2018-08-08 00:00:00', '5', '2018-08-10 12:00:00', '2018-08-15 23:00:00', 'cash', '1', '1', '1', '3', '590'),
        ('2', '2018-06-08 00:00:00', '20', '2018-06-08 12:00:00', '2018-06-28 23:00:00', 'card', '1', '1', '2', '1', '2300'),
        ('3', '2018-06-08 00:00:00', '10', '2018-06-08 12:00:00', '2018-06-18 23:00:00', 'card', '1', '1', '1', '3', '1100'),
        ('4', '2018-06-08 00:00:00', '2', '2018-06-08 12:00:00', '2018-06-10 23:00:00', 'card', '1', '1', '4', '1', '290'),
        ('5', '2018-06-08 00:00:00', '3', '2018-06-08 12:00:00', '2018-06-11 23:00:00', 'card', '1', '1', '2', '3', '350'),
```

('6', '2018-06-08 00:00:00', '5', '2018-06-08 12:00:00', '2018-06-13 23:00:00', 'card', '1', '1', '3', '3', '570'),

('7', '2018-08-13 00:00:00', '2', '2018-06-13 12:00:00', '2018-06-15 23:00:00', 'cash', '2', '1', '5', '4', '280'),

('8', '2018-08-10 00:00:00', '3', '2018-08-11 12:00:00', '2018-08-13 23:00:00', 'card', '1', '1', '3', '3', '350'),

('9', '2018-08-10 00:00:00', '5', '2018-08-12 12:00:00', '2018-08-16 23:00:00', 'card', '1', '1', '4', '3', '570'),

('10', '2018-08-14 00:00:00', '2', '2018-08-15 12:00:00', '2018-08-17 23:00:00', 'cash', '2', '1', '5', '4', '280'),

('11', '2018-08-14 00:00:00', '5', '2018-08-16 12:00:00', '2018-08-21 23:00:00', 'cash', '1', '1', '1', '3', '590'),

('12', '2018-08-14 00:00:00', '20', '2018-08-17 12:00:00', '2018-09-07 23:00:00', 'card', '1', '1', '2', '1', '2300'),

('13', '2018-08-14 00:00:00', '10', '2018-08-15 12:00:00', '2018-08-25 23:00:00', 'card', '1', '1', '1', '3', '1100'),

('14', '2018-08-14 00:00:00', '2', '2018-08-16 12:00:00', '2018-08-18 23:00:00', 'card', '2', '1', '4', '1', '290'),

('15', '2018-08-14 00:00:00', '3', '2018-08-17 12:00:00', '2018-08-20 23:00:00', 'card', '3', '1', '2', '3', '350');

CREATE INDEX `fk_bookings_hotel1_idx` ON `hotel_database`.`bookings` (`hotel_hotel_id` ASC);

CREATE INDEX `fk_bookings_guests1_idx` ON `hotel_database`.`bookings` (`guests_guest_id` ASC);

```
CREATE INDEX `fk_bookings_employees1_idx` ON `hotel_database`.`bookings`
(`employees_emp_id` ASC);



-- -----------------------------------------------------

-- Table `hotel_database`.`hotel_chain_has_hotel`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `hotel_database`.`hotel_chain_has_hotel` ;


CREATE TABLE IF NOT EXISTS `hotel_database`.`hotel_chain_has_hotel` (

  `hotel_chains_hotel_chain_id` INT NOT NULL,


  `hotels_hotel_id` INT NOT NULL,

  PRIMARY KEY (`hotel_chains_hotel_chain_id`,  `hotels_hotel_id`),

  CONSTRAINT `fk_hotel_chains_has_hotels_hotel_chains1`

    FOREIGN KEY (`hotel_chains_hotel_chain_id`)

    REFERENCES `hotel_database`.`hotel_chain` (`hotel_chain_id` )

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  CONSTRAINT `fk_hotel_chains_has_hotels_hotels1`

    FOREIGN KEY (`hotels_hotel_id`)

    REFERENCES `hotel_database`.`hotel` (`hotel_id`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION);
```

```
INSERT INTO `hotel_chain_has_hotel`(`hotel_chains_hotel_chain_id`, `hotels_hotel_id`)
 VALUES
 (1,1),
 (1,2),
 (1,3),
 (1,4),
 (2,3),
 (2,4);

CREATE INDEX `fk_hotel_chains_has_hotels_hotels1_idx` ON
`hotel_database`.`hotel_chain_has_hotel` (`hotels_hotel_id` ASC);

CREATE INDEX `fk_hotel_chains_has_hotels_hotel_chains1_idx` ON
`hotel_database`.`hotel_chain_has_hotel` (`hotel_chains_hotel_chain_id` ASC);


-- -----------------------------------------------------
-- Table `hotel_database`.`room_rate_discount`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`room_rate_discount` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`room_rate_discount` (
  `discount_id` INT NOT NULL,
  `discount_rate` DECIMAL(10,2) NULL,
```

```
 `start_month` TINYINT(1) NULL,
 `end_month` TINYINT(1) NULL,
 `room_type_room_type_id` INT NOT NULL,
 PRIMARY KEY (`discount_id`, `room_type_room_type_id`),
 CONSTRAINT `fk_room_rate_discount_room_type1`
   FOREIGN KEY (`room_type_room_type_id`)
   REFERENCES `hotel_database`.`room_type` (`room_type_id`)
   ON DELETE NO ACTION
   ON UPDATE NO ACTION);


INSERT INTO `room_rate_discount`(`discount_id`, `discount_rate`, `start_month`,
`end_month`, `room_type_room_type_id`)
 VALUES
        (1,50,1,3,1),
        (2,15,6,8,1),
        (3,15,9,12,1),
        (4,0,4,6,1),
        (1,50,1,3,2),
        (2,80,6,8,2),
        (3,15,9,12,2),
        (4,0,4,6,2),
        (1,50,1,3,3),
        (2,80,6,8,3),
        (3,15,9,12,3),
```

```
        (4,0,4,6,3);


CREATE INDEX `fk_room_rate_discount_room_type1_idx` ON
`hotel_database`.`room_rate_discount` (`room_type_room_type_id` ASC);



-- -----------------------------------------------------
-- Table `hotel_database`.`rooms_booked`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`rooms_booked` ;


CREATE TABLE IF NOT EXISTS `hotel_database`.`rooms_booked` (
  `rooms_booked_id` INT NOT NULL,
  `bookings_booking_id` INT NOT NULL,
  `rooms_room_id` INT NOT NULL,
  PRIMARY KEY (`rooms_booked_id`, `bookings_booking_id`, `rooms_room_id`),
  CONSTRAINT `fk_rooms_booked_bookings1`
    FOREIGN KEY (`bookings_booking_id`)
    REFERENCES `hotel_database`.`bookings` (`booking_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_rooms_booked_rooms1`
    FOREIGN KEY (`rooms_room_id`)
    REFERENCES `hotel_database`.`rooms` (`room_id`)
```

```
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);


INSERT INTO `rooms_booked` (`rooms_booked_id`, `bookings_booking_id`,
`rooms_room_id`)
VALUES
('1', '1', '1'),
('2', '2', '2'),
('3', '2', '3'),
('4', '2', '4'),
('5', '2', '5'),
('6', '2', '6'),
('7', '7', '7'),
('8', '7', '8'),
('9', '6', '9'),
('10','8','10'),
('11','9','11'),
('12','10','12'),
('13','10','13'),
('14', '11', '14'),
('15', '12', '15'),
('16', '13', '16'),
('17', '14', '17'),
('18', '14', '18'),
```

```
('19', '15', '19'),
('20', '15', '20'),
('21', '15', '21');

CREATE INDEX `fk_rooms_booked_bookings1_idx` ON `hotel_database`.`rooms_booked`
(`bookings_booking_id` ASC);

CREATE INDEX `fk_rooms_booked_rooms1_idx` ON `hotel_database`.`rooms_booked`
(`rooms_room_id` ASC);


-- -----------------------------------------------------
-- Table `hotel_database`.`hotel_services`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`hotel_services` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`hotel_services` (
  `service_id` INT NOT NULL,
  `service_name` VARCHAR(45) NULL,
  `service_description` VARCHAR(100) NULL,
  `service_cost` DECIMAL(10,2) NULL,
  `hotel_hotel_id` INT NOT NULL,
  PRIMARY KEY (`service_id`, `hotel_hotel_id`),
  CONSTRAINT `fk_hotel_services_hotel1`
```

```
  FOREIGN KEY (`hotel_hotel_id`)
  REFERENCES `hotel_database`.`hotel` (`hotel_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION);

INSERT INTO `hotel_services`(`service_id`, `service_name`, `service_description`,
`service_cost`, `hotel_hotel_id`)
 VALUES
        (1,'24-hour Room Service','There will be 24-hour Room Service to take care of customers
needs',20,1),
        (2,'Currency Exchange','Foreign Currency Exchange facility available',80,1),
        (3,'Laundry','Laundry/Dry Cleaning available same day',10,1),
        (4,'Entertainment Room','Book and watch movies',50,2),
        (5,'Swimming Pool','Pool access to all the guests',100,2),
        (6,'Gym','24 Hour Gym',140,2);

CREATE INDEX `fk_hotel_services_hotel1_idx` ON `hotel_database`.`hotel_services`
(`hotel_hotel_id` ASC);

-- -----------------------------------------------------
-- Table `hotel_database`.`hotel_services_used_by_guests`
-- -----------------------------------------------------
DROP TABLE IF EXISTS `hotel_database`.`hotel_services_used_by_guests` ;

CREATE TABLE IF NOT EXISTS `hotel_database`.`hotel_services_used_by_guests` (
```

```
  `service_used_id` INT NOT NULL,
  `hotel_services_service_id` INT NOT NULL,
  `bookings_booking_id` INT NOT NULL,
  PRIMARY KEY (`service_used_id`, `hotel_services_service_id`, `bookings_booking_id`),
  CONSTRAINT `fk_hotel_services_has_bookings_hotel_services1`
    FOREIGN KEY (`hotel_services_service_id`)
    REFERENCES `hotel_database`.`hotel_services` (`service_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_hotel_services_has_bookings_bookings1`
    FOREIGN KEY (`bookings_booking_id`)
    REFERENCES `hotel_database`.`bookings` (`booking_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);


INSERT INTO `hotel_services_used_by_guests` (`service_used_id`, `hotel_services_service_id`,
`bookings_booking_id`)
VALUES ('1', '1', '2'),
 ('2', '2', '2'),
 ('3', '3', '2');


CREATE INDEX `fk_hotel_services_has_bookings_bookings1_idx` ON
`hotel_database`.`hotel_services_used_by_guests` (`bookings_booking_id` ASC);
```

```
CREATE INDEX `fk_hotel_services_has_bookings_hotel_services1_idx` ON
`hotel_database`.`hotel_services_used_by_guests` (`hotel_services_service_id` ASC);


SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

# QUERIES, STORED PROCEDURES, AND TRIGGERS

-- How many distinct guest have made bookings for a particular month?

```
SELECT guest_first_name, guest_last_name,guest_contact_number
FROM guests
WHERE guest_id IN
    ( SELECT distinct guests_guest_id          -- get distinct guests
      FROM bookings
      WHERE MONTH(check_in_date) = 8);      -- bookings for the month of August
```

-- How many available rooms are in a particular hotel for a given date?

```
SELECT h.hotel_room_capacity AS 'Total Rooms', SUM(total_rooms_booked) AS 'Total
Rooms Booked' , h.hotel_room_capacity - SUM(b.total_rooms_booked)    AS 'Available Rooms'
    -- get available rooms
FROM `bookings` b JOIN hotel h
ON b.hotel_hotel_id = h.hotel_id
WHERE booking_date LIKE '2018-08-14%' AND hotel_hotel_id = 1;        -- for given
date and for hotel(King George Inn & Suites) with id 1
```

-- How many hotels are in a hotel chain?

```
SELECT count(*) AS 'Total Hotels'          -- count of total hotels
FROM hotel_chain_has_hotel
WHERE hotel_chains_hotel_chain_id = 1;  -- for hotel chain 'best western hotels'
```

```
-- How many books has a customer made in one year?

	SELECT count(*) AS 'Total Bookings'		-- count of total bookings

	FROM bookings

	WHERE YEAR(booking_date) = 2018 AND guests_guest_id = 1;		-- bookings in Year 2018 by guest Jane with id 1


-- How many rooms are booked in a particular hotel on a given date?

	SELECT SUM(total_rooms_booked) AS 'Total Rooms Booked'		-- sum of total rooms

	FROM `bookings`

	WHERE booking_date LIKE '2018-06-08%' AND hotel_hotel_id = 1;	-- for date 6th August,2018; and for hotel(King George Inn & Suites) with id 1


-- List all the unique countries hotels are located in.

	SELECT DISTINCT country				-- unique countries

	FROM addresses

	WHERE address_id IN				-- compare to get addresses of hotels

		( SELECT  addresses_address_id		-- address id of hotels

		  FROM hotel);


-- How many rooms are available in a given hotel?
```

```
SELECT  h.hotel_room_capacity - SUM(b.total_rooms_booked)    AS 'Available Rooms'
-- get available rooms
FROM `bookings` b JOIN hotel h
ON b.hotel_hotel_id = h.hotel_id
WHERE booking_date LIKE '2018-06-08%' AND hotel_hotel_id = 1;        -- for given
date and for hotel(King George Inn & Suites) with id 1


-- List all the hotels that have a URL available.
SELECT *
FROM `hotel`
WHERE hotel_website IS NOT NULL;        -- get the hotels whose URL is not null


-- List the rate for a room at a given time during the year.
SELECT ROUND((r_type.room_cost - ((r_dis.discount_rate * r_type.room_cost)/100)),
2) AS 'Room Rate' -- get room price on the basis od discount
FROM room_rate_discount r_dis JOIN room_type r_type                -- join rate discount
table with room type
ON r_dis.room_type_room_type_id = r_type.room_type_id
  WHERE r_type.room_type_id
IN ( Select rooms_type_rooms_type_id from rooms where room_id = 1)    -- get room
type id for room with id 1
  AND MONTH(NOW()) BETWEEN r_dis.start_month AND r_dis.end_month;


-- create view named hotel_employees to get details of all the employees
CREATE OR REPLACE VIEW hotel_employees AS
```

```
	SELECT emp_first_name AS 'First Name', emp_last_name AS 'Last Name',
emp_email_address AS 'Email Address', emp_contact_number AS 'Contact Number',
department_name AS 'Department'

	FROM employees

	JOIN department

	ON department.department_id = employees.department_department_id;



-- create view named hotel_guests to get details of the guests

CREATE OR REPLACE VIEW hotel_guests AS

SELECT guest_first_name AS 'First Name', guest_last_name AS 'Last Name',

guest_email_address AS 'Email Address', guest_contact_number AS 'Contact

Number',country,state,zipcode

FROM guests

JOIN addresses ON addresses.address_id = guests.addresses_address_id

WHERE guests.guest_id IN

		(SELECT distinct guests_guest_id		-- get distinct guests

		FROM bookings

		WHERE hotel_hotel_id = 1);			-- for hotel (King George Inn
& Suites) with id 1



USE hotel_database;



SET sql_notes = 0;	-- Temporarily disable the "Table already exists" warning



-- create table for bookings audit
```

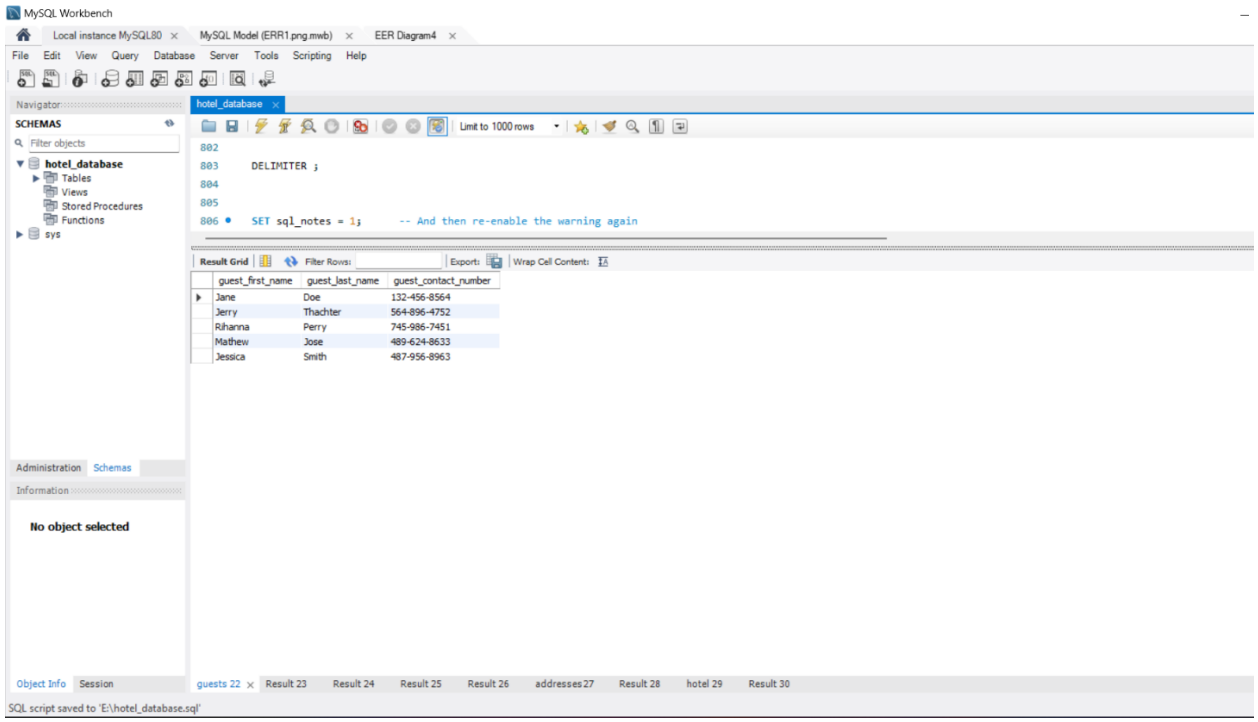
```
CREATE TABLE IF NOT EXISTS hotel_database.Bookings_Audit(

 audit_id int NOT NULL PRIMARY KEY AUTO_INCREMENT,
 `booking_id` INT NOT NULL,
 `booking_date` DATETIME NULL,
 `duration_of_stay` VARCHAR(10) NULL,
 `check_in_date` DATETIME NULL,
 `check_out_date` DATETIME NULL,
 `booking_payment_type` VARCHAR(45) NULL,
 `total_rooms_booked` INT NULL,
 `hotel_hotel_id` INT NOT NULL,
 `guests_guest_id` INT NOT NULL,
 `employees_emp_id` INT NOT NULL,
 `total_amount` DECIMAL(10,2) NULL,
 action_type varchar(50) NOT NULL,
 date_updated datetime NOT NULL
);

DROP TRIGGER IF EXISTS bookings_after_delete;

DELIMITER //

CREATE TRIGGER bookings_after_delete
AFTER DELETE ON bookings
```

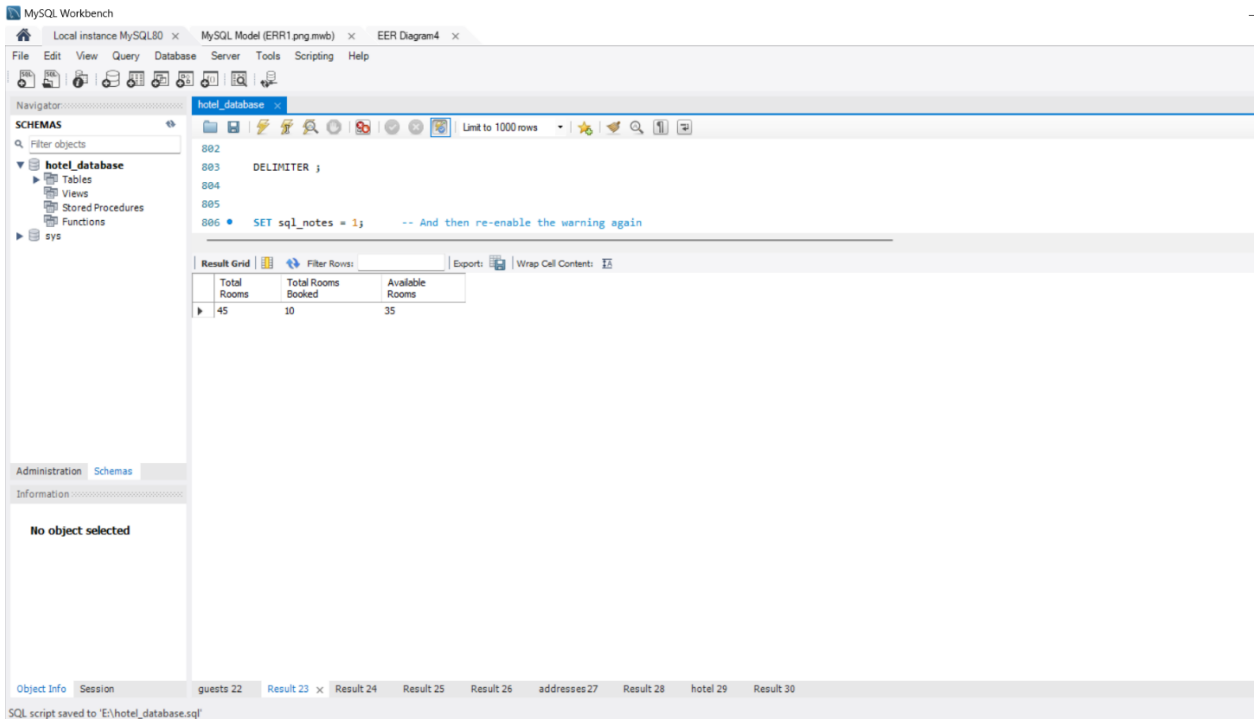
```
FOR EACH ROW

BEGIN
INSERT INTO Bookings_Audit VALUES
(NULL, OLD.booking_id, OLD.booking_date, OLD.duration_of_stay, OLD.`check_in_date`,
OLD.`check_out_date`, OLD.`booking_payment_type`, OLD.`total_rooms_booked`,
OLD.`hotel_hotel_id`, OLD.`guests_guest_id`, OLD.`employees_emp_id`,
OLD.`total_amount`,"DELETED", NOW());
END//

DELIMITER ;

SET sql_notes = 1;      -- And then re-enable the warning again

USE hotel_database;

SET sql_notes = 0;      -- Temporarily disable the "Table already exists" warning

-- create table for bookings audit
CREATE TABLE IF NOT EXISTS hotel_database.Bookings_Audit(
 audit_id int NOT NULL PRIMARY KEY AUTO_INCREMENT,
 `booking_id` INT NOT NULL,
 `booking_date` DATETIME NULL,
 `duration_of_stay` VARCHAR(10) NULL,
 `check_in_date` DATETIME NULL,
```

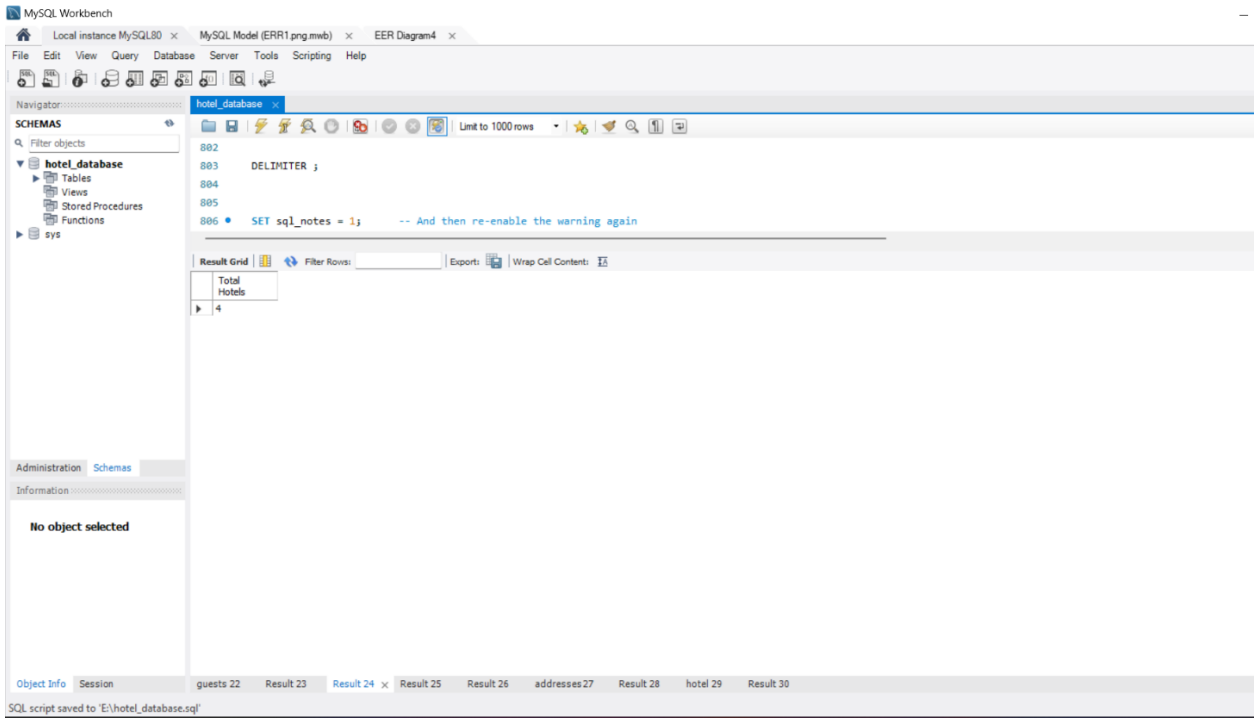
```
 `check_out_date` DATETIME NULL,
 `booking_payment_type` VARCHAR(45) NULL,
 `total_rooms_booked` INT NULL,
 `hotel_hotel_id` INT NOT NULL,
 `guests_guest_id` INT NOT NULL,
 `employees_emp_id` INT NOT NULL,
 `total_amount` DECIMAL(10,2) NULL,
 action_type varchar(50) NOT NULL,
 date_updated datetime NOT NULL
);



DROP TRIGGER IF EXISTS bookings_after_insert;


DELIMITER //


CREATE TRIGGER bookings_after_insert
AFTER INSERT ON bookings
FOR EACH ROW


BEGIN
INSERT INTO Bookings_Audit VALUES
(NEW.booking_id, NEW.booking_date, NEW.duration_of_stay, NEW.`check_in_date`,
NEW.`check_out_date`, NEW.`booking_payment_type`, NEW.`total_rooms_booked`,
```

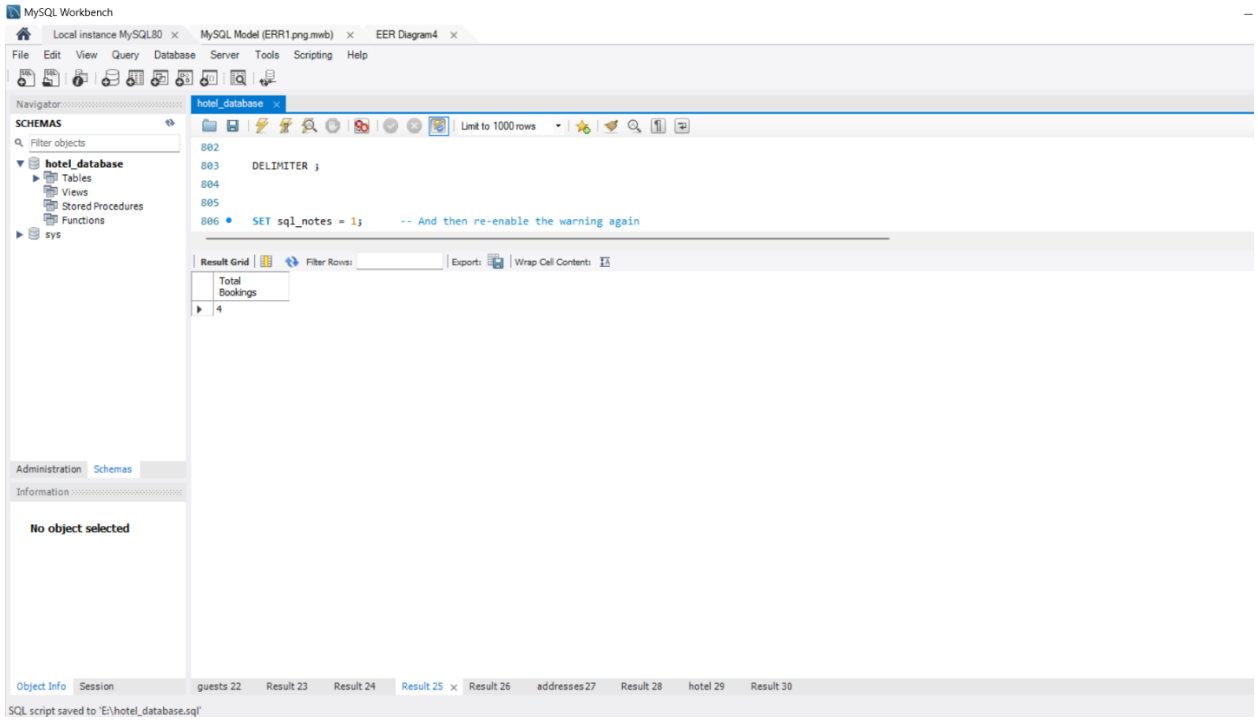
```
NEW.`hotel_hotel_id`, NEW.`guests_guest_id`, NEW.`employees_emp_id`,
NEW.`total_amount`,"INSERTED", NOW());

END//


DELIMITER ;

SET sql_notes = 1;      -- And then re-enable the warning again
```

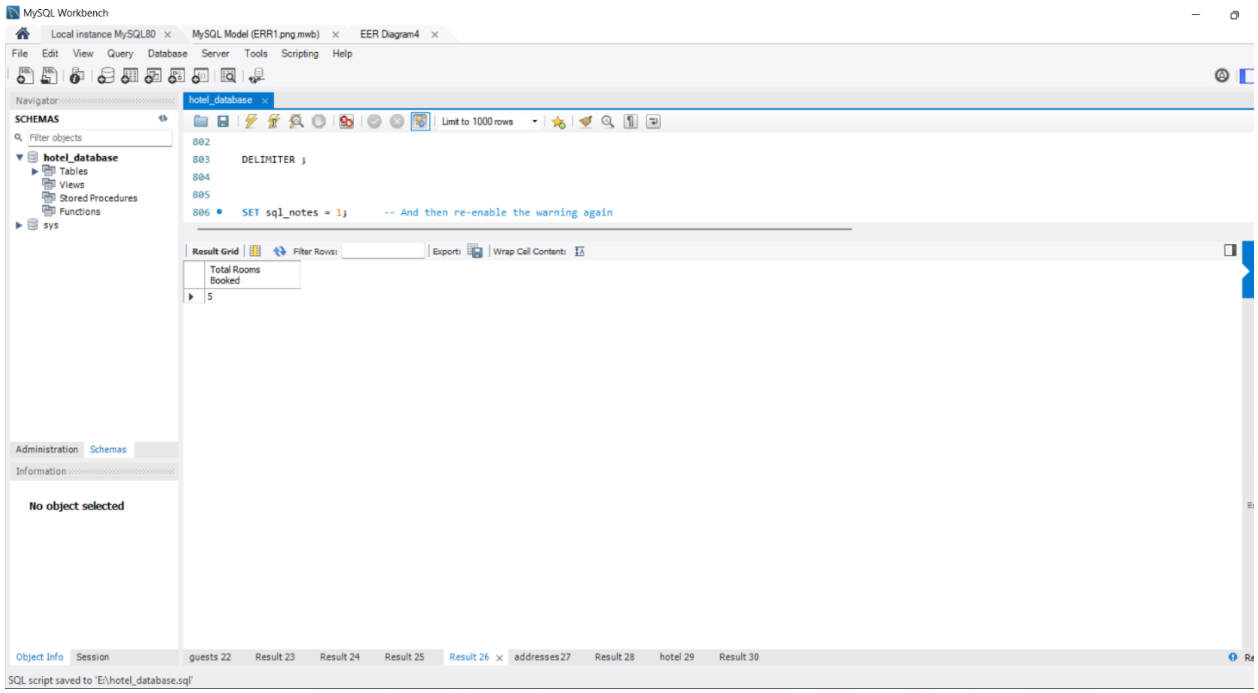
# SNAPSHOTS

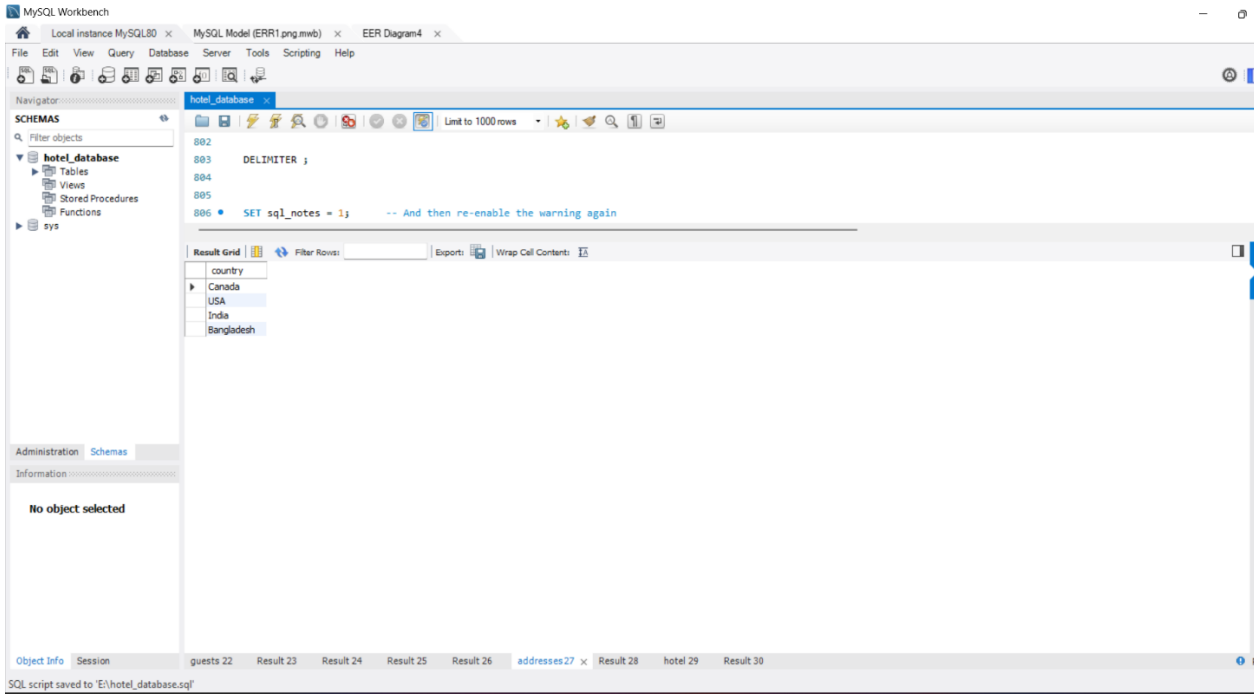

MySQL Workbench
Local instance MySQL80
MySQL Model (ERR1.png.mwb)
EER Diagram4
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
hotel_database
Tables
Views
Stored Procedures
Functions
sys
Administration Schemas
Information
No object selected
Object Info Session
hotel_database
Limit to 1000 rows
802
803 DELIMITER ;
804
805
806 SET sql_notes = 1; -- And then re-enable the warning again
Result Grid Filter Rows: Export: Wrap Cell Content:
guest_first_name guest_last_name guest_contact_number
Jane Doe 132-456-8564
Jerry Thachter 564-896-4752
Rihanna Perry 745-986-7451
Mathew Jose 489-624-8633
Jessica Smith 487-956-8963
guests 22 Result 23 Result 24 Result 25 Result 26 addresses 27 Result 28 hotel 29 Result 30
SQL script saved to 'E:\hotel_database.sql'

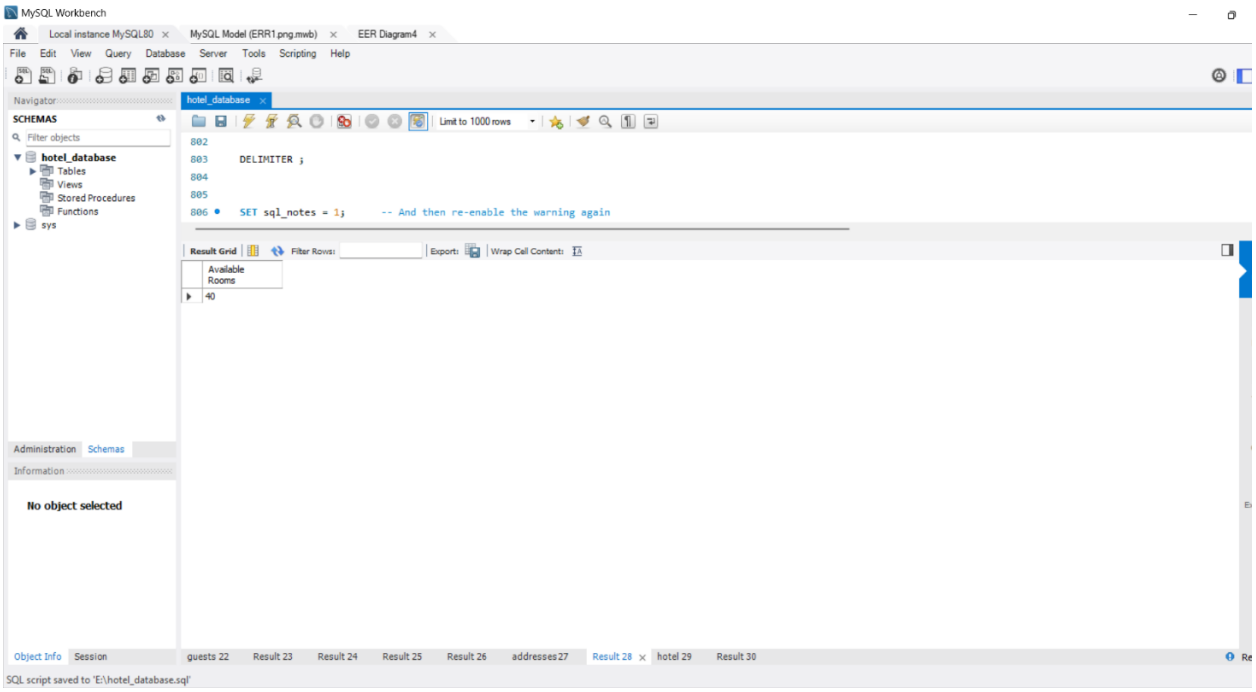

MySQL Workbench
Local instance MySQL80
MySQL Model (ERR1.png.mwb)
EER Diagram4
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
hotel_database
Tables
Views
Stored Procedures
Functions
sys
Administration Schemas
Information
No object selected
Object Info Session
hotel_database
Limit to 1000 rows
802
803 DELIMITER ;
804
805
806 SET sql_notes = 1; -- And then re-enable the warning again
Result Grid Filter Rows: Export: Wrap Cell Content:
Total Rooms
Total Rooms Booked
Available Rooms
45 10 35
guests 22 Result 23 Result 24 Result 25 Result 26 addresses 27 Result 28 hotel 29 Result 30
SQL script saved to 'E:\hotel_database.sql'

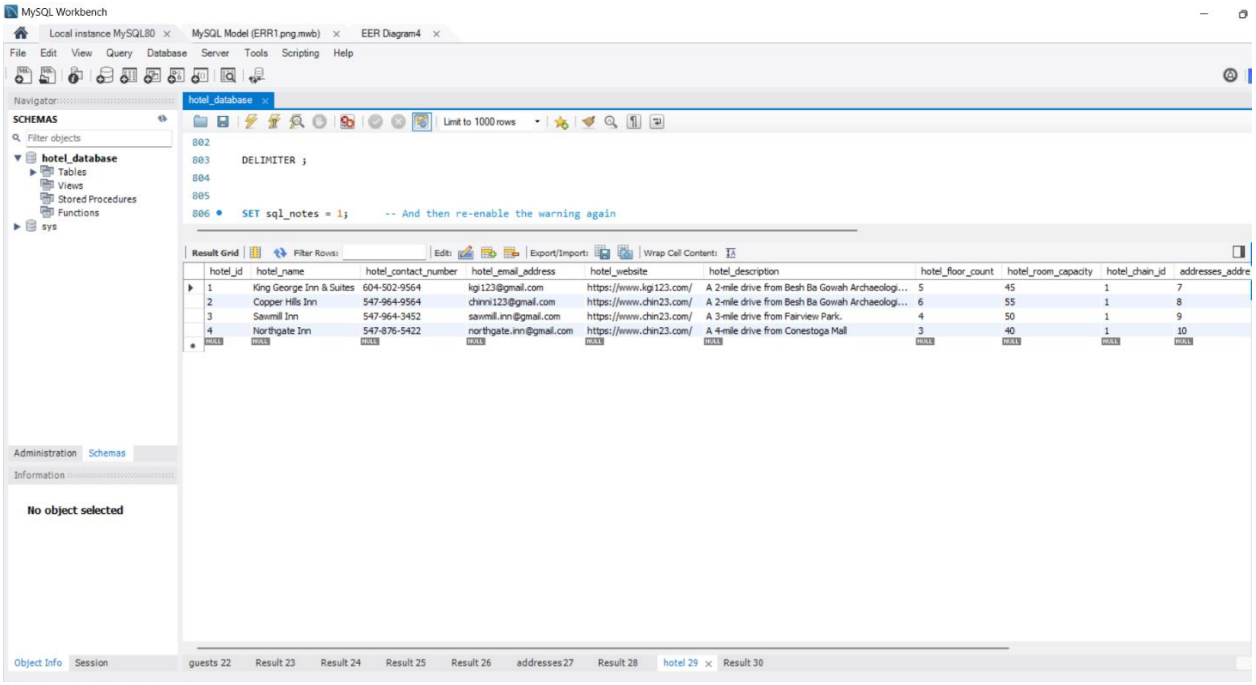

MySQL Workbench
Local instance MySQL80
MySQL Model (ERR1.png.mwb)
EER Diagram4
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
hotel_database
Tables
Views
Stored Procedures
Functions
sys
Administration Schemas
Information
No object selected
Object Info Session
hotel_database
Limit to 1000 rows
802
803 DELIMITER ;
804
805
806 SET sql_notes = 1; -- And then re-enable the warning again
Result Grid Filter Rows: Export: Wrap Cell Content:
Total Hotels
4
guests 22 Result 23 Result 24 Result 25 Result 26 addresses 27 Result 28 hotel 29 Result 30
SQL script saved to 'E:\hotel_database.sql'

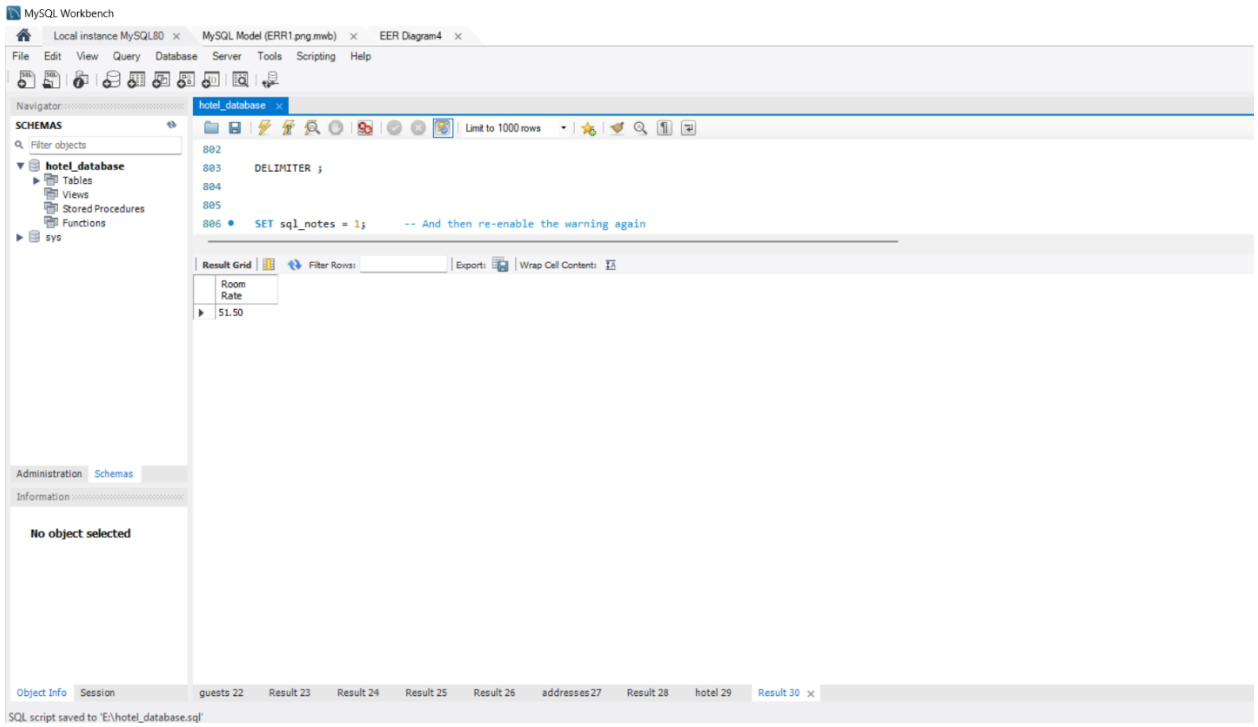

MySQL Workbench
Local instance MySQL80
MySQL Model (ERR1.png.mwb)
EER Diagram4
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
hotel_database
Tables
Views
Stored Procedures
Functions
sys
Administration Schemas
Information
No object selected
Object Info Session
hotel_database
Limit to 1000 rows
802
803 DELIMITER ;
804
805
806 SET sql_notes = 1; -- And then re-enable the warning again
Result Grid Filter Rows: Export: Wrap Cell Content:
Total Bookings
4
guests 22 Result 23 Result 24 Result 25 Result 26 addresses 27 Result 28 hotel 29 Result 30
SQL script saved to 'E:\hotel_database.sql'


MySQL Workbench
Local instance MySQL80
MySQL Model (ERR1.png.mwb)
EER Diagram4
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
hotel_database
Tables
Views
Stored Procedures
Functions
sys
hotel_database
Limit to 1000 rows
802
803 DELIMITER ;
804
805
806 SET sql_notes = 1; -- And then re-enable the warning again
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Total Rooms Booked
5
Administration Schemas
Information
No object selected
Object Info Session
guests 22 Result 23 Result 24 Result 25 Result 26 addresses27 Result 28 hotel 29 Result 30
SQL script saved to 'E:\hotel_database.sql'


MySQL Workbench
Local instance MySQL80
MySQL Model (ERR1.png.mwb)
EER Diagram4
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
hotel_database
Tables
Views
Stored Procedures
Functions
sys
hotel_database
Limit to 1000 rows
802
803 DELIMITER ;
804
805
806 SET sql_notes = 1; -- And then re-enable the warning again
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
country
Canada
USA
India
Bangladesh
Administration Schemas
Information
No object selected
Object Info Session
guests 22 Result 23 Result 24 Result 25 Result 26 addresses27 Result 28 hotel 29 Result 30
SQL script saved to 'E:\hotel_database.sql'


MySQL Workbench
Local instance MySQL80
MySQL Model (ERR1.png.mwb)
EER Diagram4
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
hotel_database
Tables
Views
Stored Procedures
Functions
sys
hotel_database
Limit to 1000 rows
802
803 DELIMITER ;
804
805
806 SET sql_notes = 1; -- And then re-enable the warning again
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Available Rooms
40
Administration Schemas
Information
No object selected
Object Info Session
guests 22 Result 23 Result 24 Result 25 Result 26 addresses 27 Result 28 hotel 29 Result 30
SQL script saved to 'E:\hotel_database.sql'


MySQL Workbench
Local instance MySQL80
MySQL Model (ERR1.png.mwb)
EER Diagram4
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
hotel_database
Tables
Views
Stored Procedures
Functions
sys
hotel_database
Limit to 1000 rows
802
803 DELIMITER ;
804
805
806 SET sql_notes = 1; -- And then re-enable the warning again
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
hotel_id hotel_name hotel_contact_number hotel_email_address hotel_website hotel_description hotel_floor_count hotel_room_capacity hotel_chain_id addresses_addre
1 King George Inn & Suites 604-502-9564 kgi123@gmail.com https://www.kgi123.com/ A 2-mile drive from Besh Ba Gowah Archaeolog... 5 45 1 7
2 Copper Hills Inn 547-964-9564 chinni123@gmail.com https://www.chin23.com/ A 2-mile drive from Besh Ba Gowah Archaeologi... 6 55 1 8
3 Sawmill Inn 547-964-3452 sawmill.inn@gmail.com https://www.chin23.com/ A 3-mile drive from Fairview Park. 4 50 1 9
4 Northgate Inn 547-876-5422 northgate.inn@gmail.com https://www.chin23.com/ A 4-mile drive from Conestoga Mall 3 40 1 10
Administration Schemas
Information
No object selected
Object Info Session
guests 22 Result 23 Result 24 Result 25 Result 26 addresses 27 Result 28 hotel 29 Result 30


MySQL Workbench
Local instance MySQL80
MySQL Model (ERR1.png.mwb)
EER Diagram4
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
hotel_database
Tables
Views
Stored Procedures
Functions
sys
Administration Schemas
Information
No object selected
Object Info Session
hotel_database
Limit to 1000 rows
802
803 DELIMITER ;
804
805
806 SET sql_notes = 1; -- And then re-enable the warning again
Result Grid Filter Rows: Export: Wrap Cell Content:
Room Rate
51.50
guests 22 Result 23 Result 24 Result 25 Result 26 addresses 27 Result 28 hotel 29 Result 30
SQL script saved to 'E:\hotel_database.sql'