**INFO 6210**

**Data Management and Database Design**

**Suyog Sathe**

**NUID: 001863458**

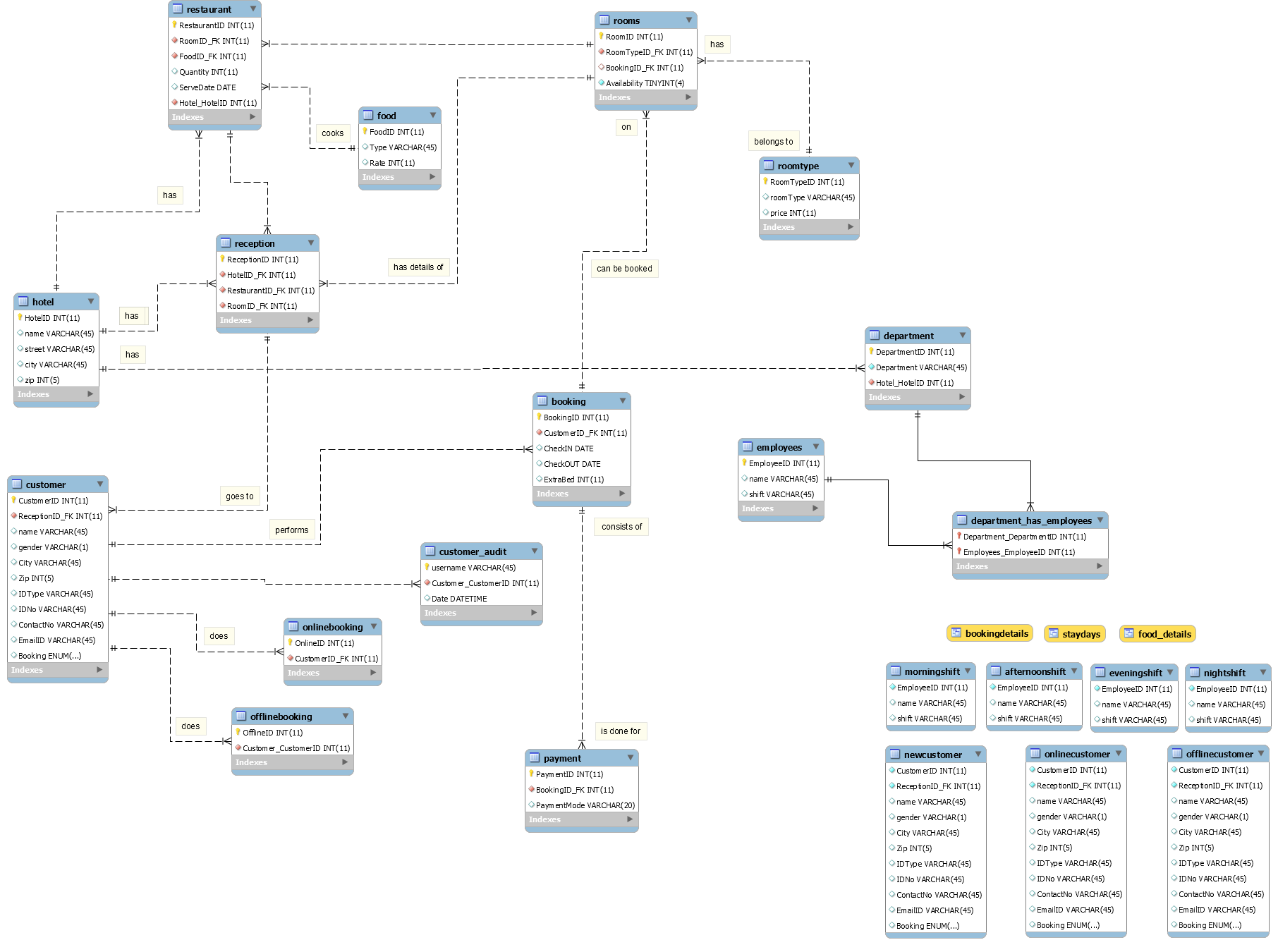
**HOTEL MANAGEMENT SYSTEM**

**FINAL PROJECT**

**PROBLEM STATEMENT:**

* In this project, I am implementing Hotel Management System. There are various hotels around the world, and they serve the customers with the rooms and food they got.
* So, the customer visits one hotel, enquire about the booking process and details at the reception. Then a room is allotted to him, depending on his room type and a booking ID is generated for his unique booking. A bill would be later generated when the customer will checkout.
* There are various departments such as Cooking staff, Housekeeping Staff, Security Staff which are assigned their particular work-load.
* There is a restaurant in the hotel where the customers will go to have their breakfast, lunch or dinner. Depending upon their choice and order, a bill would be generated about the same.
* A total bill would be generated at the time of checkout including the restaurant bill, if he has eaten there.

**E-R Diagram**



**Stored Procedures:**

1. **PROCEDURE FOR VIEWING THE RESTAURANT BILL**

DELIMITER //

CREATE PROCEDURE proc\_res\_bill(IN id INT(10))

BEGIN

SELECT RoomID\_FK AS `Room No`,`Type` AS `Food Type Ordered`,sum(Quantity\*Rate) AS Amount

FROM food

INNER JOIN restaurant

ON food.FoodID = restaurant.FoodID\_FK

WHERE RoomID\_FK=id

GROUP BY `Type` WITH ROLLUP;

END //

DELIMITER ;

DROP PROCEDURE proc\_res\_bill;

CALL proc\_res\_bill(102);

**2.) PROCEDURE FOR VIEWING THE HOTEL BILL**

DELIMITER //

CREATE PROCEDURE proc\_hotel\_bill(IN id INT(10))

BEGIN

SELECT CustomerID, `name`, roomType, CheckOUT-CheckIN AS `Days Stayed`,SUM((CheckOUT-CheckIN)\*price) AS `Total`

FROM customer

inner join booking

on customer.CustomerID = booking.CustomerID\_FK

inner join rooms

on booking.BookingID = rooms.BookingID\_FK

inner join roomtype

on rooms.RoomTypeID\_FK = roomtype.RoomTypeID

WHERE CheckIN IS NOT NULL and CheckOUT IS NOT NULL and CustomerID=1

GROUP BY roomType WITH ROLLUP;

END //

DELIMITER ;

DROP PROCEDURE proc\_hotel\_bill;

CALL proc\_hotel\_bill(1);

**3.) TOTAL BILL AND DETAILS WHILE CHECKING OUT**

DELIMITER //

CREATE PROCEDURE proc\_invoice(IN id INT(10))

BEGIN

SELECT BookingID,`name` AS `Name`,CheckIN, CheckOUT , sum(Quantity\*Rate) AS `Restaurant Bill`, (CheckOUT-CheckIN)\*price AS `Hotel Bill` ,sum(Quantity\*Rate)+((CheckOUT-CheckIN)\*price) AS `Total Bill`

FROM customer

INNER JOIN booking

ON customer.CustomerID = booking.CustomerID\_FK

INNER JOIN rooms

ON booking.BookingID = rooms.BookingID\_FK

INNER JOIN roomtype

ON rooms.RoomTypeID\_FK = roomtype.RoomTypeID

INNER JOIN restaurant

ON rooms.RoomID = restaurant.RoomID\_FK

INNER JOIN food

ON restaurant.FoodID\_FK = food.FoodID

WHERE CheckIN IS NOT NULL and CheckOUT IS NOT NULL AND BookingID=id

GROUP BY name

ORDER BY BookingID;

END //

DELIMITER ;

DROP procedure proc\_invoice;

call proc\_invoice(2);

**4.)** **PROCEDURE TO BACKUP CUSTOMERS AS PER THE BOOKING**

DELIMITER ##

CREATE PROCEDURE backup\_customer\_details()

BEGIN

CREATE TABLE newCustomer AS SELECT \* FROM customer;

CREATE TABLE onlineCustomer AS SELECT \* FROM customer WHERE Booking=”Online”;

CREATE TABLE offlineCustomer AS SELECT \* FROM customer WHERE Booking=”Offline”;

END ##

DELIMITER ;

DROP procedure if exists backup\_customer\_details;

call backup\_customer\_details();

SELECT \* FROM offlinecustomer;

SELECT \* FROM onlinecustomer;

SELECT \* FROM newcustomer;

DROP table newcustomer;

DROP table onlinecustomer;

DROP table offlinecustomer;

**5.)** **PROCEDURE TO COUNT THE NUMBER OF ROOMS AVAILABLE WHERE BOOKING ID IS NULL**

DELIMITER $$

CREATE procedure count\_rooms()

begin

SELECT roomType AS `Room Type`,count(rooms.Availability)AS `Available Rooms` FROM rooms

inner join roomtype

on rooms.RoomTypeID\_FK = roomtype.RoomTypeID

WHERE BookingID\_FK IS NULL

GROUP BY roomType;

end $$

DELIMITER ;

DROP procedure if exists count\_rooms;

call count\_rooms();

6.) **PROCEDURE TO BACKUP EMPLOYEES AS PER THE SHIFT**

DELIMITER ##

CREATE PROCEDURE shift\_details()

BEGIN

CREATE TABLE morningShift AS SELECT \* FROM employees WHERE shift=”Morning”;

CREATE TABLE afternoonShift AS SELECT \* FROM employees WHERE shift=”Afternoon”;

CREATE TABLE eveningShift AS SELECT \* FROM employees WHERE shift=”Evening”;

CREATE TABLE nightShift AS SELECT \* FROM employees WHERE shift=”Night”;

END ##

DELIMITER ;

DROP procedure if exists shift\_details;

call shift\_details();

SELECT \* FROM morningshift;

SELECT \* FROM afternoonshift;

SELECT \* FROM eveningshift;

SELECT \* FROM nightshift;

DROP table morningshift;

DROP table afternoonshift;

DROP table eveningshift;

DROP table nightshift;

**Views:**

**1.)TO SHOW THE DETAILS OF THE BOOKING**

DROP VIEW IF EXISTS BookingDetails;

DROP TABLE IF EXISTS BookingDetails;

CREATE VIEW BookingDetails AS

SELECT BookingID,name AS `Customer Name`,CheckIN AS `Check In Date`, CheckOUT AS `Check out Date`,Booking AS `Booking Type`

FROM booking

INNER JOIN customer

ON booking.CustomerID\_FK = customer.CustomerID

WHERE BookingID IS NOT NULL

ORDER BY BookingID;

SELECT \* FROM BookingDetails;

**2.) TO CHECK WHETHER THE ROOM IS BOOKED OR NOT**

DROP VIEW IF EXISTS room\_status;

DROP TABLE IF EXISTS room\_status;

CREATE VIEW room\_status AS

SELECT RoomID AS `Room Number`, RoomType, CheckOUT AS `Check Out Date`

FROM booking

INNER JOIN rooms

ON booking.BookingID = rooms.BookingID\_FK

INNER JOIN roomtype

ON rooms.RoomTypeID\_FK = roomtype.RoomTypeID

ORDER BY RoomID;

SELECT \* FROM room\_status;

**3.) TO SHOW THE DETAILS OF THE FOOD CUSTOMER ORDERED**

DROP VIEW IF EXISTS food\_details;

DROP TABLE IF EXISTS food\_details;

CREATE VIEW food\_details AS

SELECT RoomID AS `Room Number`,Type AS `Food`, Quantity, Rate, Quantity\*Rate AS Amount, ServeDate AS `Date of Food Served`

FROM rooms

INNER JOIN restaurant

ON rooms.RoomID = restaurant.RoomID\_FK

INNER JOIN food

ON restaurant.FoodID\_FK = food.FoodID

ORDER BY RoomID;

SELECT \* FROM food\_details;

**4.) TO CHECK THE NUMBER OF DAYS THE CUSTOMER STAYED**

DROP view if exists stayDays;

DROP table if exists stayDays;

CREATE VIEW stayDays AS

SELECT RoomID AS `Room Number`, name AS `Customer Name`, roomType AS `Room Type`, BookingID, CheckOUT-CheckIN AS `No of Days Stayed`

FROM customer

INNER JOIN booking

ON customer.CustomerID = booking.CustomerID\_FK

INNER JOIN rooms

ON booking.BookingID = rooms.BookingID\_FK

RIGHT OUTER JOIN roomtype

ON rooms.RoomTypeID\_FK = roomtype.RoomTypeID

WHERE BookingID IS NOT NULL and CheckIN IS NOT NULL and CheckOUT IS NOT NULL -- done

ORDER BY RoomID;

SELECT \* FROM stayDays;

**TRIGGERS:**

**1.) WHEN CUSTOMER IS ADDED, TRIGGER SHOULD HIT AND CREATE A VIRTUAL TABLE WITH CUSTOMER NAME AND DATETIME**

DELIMITER %

CREATE TRIGGER tr\_customer\_forInsert

AFTER INSERT

ON customer FOR EACH ROW

BEGIN

DECLARE vUser varchar(20);

-- SELECT user() into vUser;

SELECT `name` FROM customer WHERE CustomerID = New.CustomerID into Vuser;

INSERT INTO customer\_audit(username,Customer\_CustomerID,`Date`) VALUES (vUser,New.CustomerID,current\_timestamp());

END %

DELIMITER ;

DROP trigger tr\_customer\_forInsert;

describe customer;

describe customer\_audit;

**Privileges:**

**1.) SELECT PRIVILEGE**

CREATE user 'suyog'@'localhost' identified by 'suyog';

GRANT SELECT on `HotelManagement`.\* to 'suyog'@'localhost';

SHOW GRANTs for 'suyog'@'localhost';

DROP user 'suyog'@'localhost';

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

**2.) EXECUTE AND SELECT PRIVILEGE**

CREATE user 'mayur'@'localhost' identified by 'mayur';

GRANT SELECT, EXECUTE on `HotelManagement`.\* to 'mayur'@'localhost';

SHOW GRANTs for 'mayur'@'localhost';

DROP user 'mayur'@'localhost';

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

**3.) UPDATE SELECT PRIVILEGE**

CREATE user 'chai'@'localhost' identified by 'chai';

GRANT UPDATE,SELECT on `HotelManagement`.\* to 'chai'@'localhost';

SHOW GRANTs for 'chai'@'localhost';

DROP user 'chai'@'localhost';

**4.) DELETE SELECT PRIVILEGE**

CREATE user 'chinmay'@'localhost' identified by 'chinmay';

GRANT DELETE,SELECT on `HotelManagement`.\* to 'chinmay'@'localhost';

SHOW GRANTs for 'chinmay'@'localhost';

DROP user 'chinmay'@'localhost';

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

**5.) SELECT, UPDATE, DELETE PRIVILEGE**

CREATE user 'rohan'@'localhost' identified by 'rohan';

GRANT UPDATE,DELETE,SELECT on `HotelManagement`.\* to 'rohan'@'localhost';

SHOW GRANTs for 'rohan'@'localhost';

DROP user 'rohan'@'localhost';

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

**6.) SPECIFIC COLUMNS PRIVILEGE**

CREATE user 'ankur'@'localhost' identified by 'ankur';

GRANT SELECT(Availability) on `HotelManagement`.`rooms` to 'ankur'@'localhost';

SHOW GRANTs for 'ankur'@'localhost';

DROP user 'ankur'@'localhost';

**Event:**

1. **BACKING UP CUSTOMERS**

DELIMITER $$

CREATE EVENT customerBackups ON schedule every 5 hour starts '2017-12-13 08:30:00'

do begin

SELECT \* into outfile 'C:\STUDY\DBMS\eg.csv' FROM customer;

end $$

DELIMITER ;

DROP event customerBackups;

**MYSQL Scripts:**

**-- MYSQL WORKBENCH FORWARD ENGINEERING**

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';

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

-- Schema hotelmanagement

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

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

-- Schema hotelmanagement

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

CREATE SCHEMA IF NOT EXISTS `hotelmanagement` DEFAULT CHARACTER SET utf8 ;

SHOW WARNINGS;

USE `hotelmanagement` ;

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

-- Table `hotelmanagement`.`afternoonshift`

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

DROP TABLE IF EXISTS `hotelmanagement`.`afternoonshift` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`afternoonshift` (

`EmployeeID` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`shift` VARCHAR(45) NULL DEFAULT NULL)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`hotel`

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

DROP TABLE IF EXISTS `hotelmanagement`.`hotel` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`hotel` (

`HotelID` INT(11) NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL DEFAULT NULL,

`street` VARCHAR(45) NULL DEFAULT NULL,

`city` VARCHAR(45) NULL DEFAULT NULL,

`zip` INT(5) NULL DEFAULT NULL,

PRIMARY KEY (`HotelID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`food`

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

DROP TABLE IF EXISTS `hotelmanagement`.`food` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`food` (

`FoodID` INT(11) NOT NULL AUTO\_INCREMENT,

`Type` VARCHAR(45) NULL DEFAULT NULL,

`Rate` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`FoodID`))

ENGINE = InnoDB

AUTO\_INCREMENT = 4

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`roomtype`

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

DROP TABLE IF EXISTS `hotelmanagement`.`roomtype` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`roomtype` (

`RoomTypeID` INT(11) NOT NULL AUTO\_INCREMENT,

`roomType` VARCHAR(45) NULL DEFAULT NULL,

`price` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`RoomTypeID`))

ENGINE = InnoDB

AUTO\_INCREMENT = 4

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`rooms`

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

DROP TABLE IF EXISTS `hotelmanagement`.`rooms` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`rooms` (

`RoomID` INT(11) NOT NULL AUTO\_INCREMENT,

`RoomTypeID\_FK` INT(11) NOT NULL,

`BookingID\_FK` INT(11) NULL DEFAULT NULL,

`Availability` TINYINT(4) NOT NULL,

PRIMARY KEY (`RoomID`),

INDEX `fk\_Rooms\_RoomType1\_idx` (`RoomTypeID\_FK` ASC),

INDEX `fk\_Rooms\_Booking1\_idx` (`BookingID\_FK` ASC),

CONSTRAINT `fk\_Rooms\_Booking1`

FOREIGN KEY (`BookingID\_FK`)

REFERENCES `hotelmanagement`.`booking` (`BookingID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Rooms\_RoomType1`

FOREIGN KEY (`RoomTypeID\_FK`)

REFERENCES `hotelmanagement`.`roomtype` (`RoomTypeID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

AUTO\_INCREMENT = 305

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`restaurant`

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

DROP TABLE IF EXISTS `hotelmanagement`.`restaurant` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`restaurant` (

`RestaurantID` INT(11) NOT NULL AUTO\_INCREMENT,

`RoomID\_FK` INT(11) NOT NULL,

`FoodID\_FK` INT(11) NOT NULL,

`Quantity` INT(11) NULL DEFAULT NULL,

`ServeDate` DATE NULL DEFAULT NULL,

`Hotel\_HotelID` INT(11) NOT NULL,

PRIMARY KEY (`RestaurantID`),

INDEX `fk\_Restaurant\_Rooms1\_idx` (`RoomID\_FK` ASC),

INDEX `fk\_Restaurant\_FoodCategory1\_idx` (`FoodID\_FK` ASC),

INDEX `fk\_Restaurant\_Hotel1\_idx` (`Hotel\_HotelID` ASC),

CONSTRAINT `fk\_Restaurant\_FoodCategory1`

FOREIGN KEY (`FoodID\_FK`)

REFERENCES `hotelmanagement`.`food` (`FoodID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Restaurant\_Hotel1`

FOREIGN KEY (`Hotel\_HotelID`)

REFERENCES `hotelmanagement`.`hotel` (`HotelID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Restaurant\_Rooms1`

FOREIGN KEY (`RoomID\_FK`)

REFERENCES `hotelmanagement`.`rooms` (`RoomID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

AUTO\_INCREMENT = 11

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`reception`

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

DROP TABLE IF EXISTS `hotelmanagement`.`reception` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`reception` (

`ReceptionID` INT(11) NOT NULL AUTO\_INCREMENT,

`HotelID\_FK` INT(11) NOT NULL,

`RestaurantID\_FK` INT(11) NOT NULL,

`RoomID\_FK` INT(11) NOT NULL,

PRIMARY KEY (`ReceptionID`),

INDEX `fk\_Reception\_Hotel1\_idx` (`HotelID\_FK` ASC),

INDEX `fk\_Reception\_Restaurant1\_idx` (`RestaurantID\_FK` ASC),

INDEX `fk\_Reception\_Rooms1\_idx` (`RoomID\_FK` ASC),

CONSTRAINT `fk\_Reception\_Hotel1`

FOREIGN KEY (`HotelID\_FK`)

REFERENCES `hotelmanagement`.`hotel` (`HotelID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Reception\_Restaurant1`

FOREIGN KEY (`RestaurantID\_FK`)

REFERENCES `hotelmanagement`.`restaurant` (`RestaurantID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Reception\_Rooms1`

FOREIGN KEY (`RoomID\_FK`)

REFERENCES `hotelmanagement`.`rooms` (`RoomID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`customer`

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

DROP TABLE IF EXISTS `hotelmanagement`.`customer` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`customer` (

`CustomerID` INT(11) NOT NULL AUTO\_INCREMENT,

`ReceptionID\_FK` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`gender` VARCHAR(1) NULL DEFAULT NULL,

`City` VARCHAR(45) NULL DEFAULT NULL,

`Zip` INT(5) NULL DEFAULT NULL,

`IDType` VARCHAR(45) NULL DEFAULT NULL,

`IDNo` VARCHAR(45) NULL DEFAULT NULL,

`ContactNo` VARCHAR(45) NULL DEFAULT NULL,

`EmailID` VARCHAR(45) NULL DEFAULT NULL,

`Booking` ENUM('Online', 'Offline') NULL DEFAULT NULL,

PRIMARY KEY (`CustomerID`),

INDEX `fk\_Customer\_Reception1\_idx` (`ReceptionID\_FK` ASC),

CONSTRAINT `fk\_Customer\_Reception1`

FOREIGN KEY (`ReceptionID\_FK`)

REFERENCES `hotelmanagement`.`reception` (`ReceptionID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

AUTO\_INCREMENT = 11

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`booking`

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

DROP TABLE IF EXISTS `hotelmanagement`.`booking` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`booking` (

`BookingID` INT(11) NOT NULL AUTO\_INCREMENT,

`CustomerID\_FK` INT(11) NOT NULL,

`CheckIN` DATE NULL DEFAULT NULL,

`CheckOUT` DATE NULL DEFAULT NULL,

`ExtraBed` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`BookingID`),

INDEX `fk\_Booking\_Customer1\_idx` (`CustomerID\_FK` ASC),

CONSTRAINT `fk\_Booking\_Customer1`

FOREIGN KEY (`CustomerID\_FK`)

REFERENCES `hotelmanagement`.`customer` (`CustomerID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

AUTO\_INCREMENT = 11

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`customer\_audit`

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

DROP TABLE IF EXISTS `hotelmanagement`.`customer\_audit` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`customer\_audit` (

`username` VARCHAR(45) NOT NULL,

`Customer\_CustomerID` INT(11) NOT NULL,

`Date` DATETIME NULL DEFAULT NULL,

PRIMARY KEY (`username`),

INDEX `fk\_customer\_Audit\_Customer1\_idx` (`Customer\_CustomerID` ASC),

CONSTRAINT `fk\_customer\_Audit\_Customer1`

FOREIGN KEY (`Customer\_CustomerID`)

REFERENCES `hotelmanagement`.`customer` (`CustomerID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`department`

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

DROP TABLE IF EXISTS `hotelmanagement`.`department` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`department` (

`DepartmentID` INT(11) NOT NULL,

`Department` VARCHAR(45) NOT NULL,

`Hotel\_HotelID` INT(11) NOT NULL,

PRIMARY KEY (`DepartmentID`),

INDEX `fk\_Department\_Hotel1\_idx` (`Hotel\_HotelID` ASC),

CONSTRAINT `fk\_Department\_Hotel1`

FOREIGN KEY (`Hotel\_HotelID`)

REFERENCES `hotelmanagement`.`hotel` (`HotelID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`employees`

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

DROP TABLE IF EXISTS `hotelmanagement`.`employees` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`employees` (

`EmployeeID` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`shift` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`EmployeeID`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`department\_hAS\_employees`

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

DROP TABLE IF EXISTS `hotelmanagement`.`department\_hAS\_employees` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`department\_hAS\_employees` (

`Department\_DepartmentID` INT(11) NOT NULL,

`Employees\_EmployeeID` INT(11) NOT NULL,

PRIMARY KEY (`Department\_DepartmentID`, `Employees\_EmployeeID`),

INDEX `fk\_Department\_hAS\_Employees\_Employees1\_idx` (`Employees\_EmployeeID` ASC),

INDEX `fk\_Department\_hAS\_Employees\_Department1\_idx` (`Department\_DepartmentID` ASC),

CONSTRAINT `fk\_Department\_hAS\_Employees\_Department1`

FOREIGN KEY (`Department\_DepartmentID`)

REFERENCES `hotelmanagement`.`department` (`DepartmentID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Department\_hAS\_Employees\_Employees1`

FOREIGN KEY (`Employees\_EmployeeID`)

REFERENCES `hotelmanagement`.`employees` (`EmployeeID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`eveningshift`

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

DROP TABLE IF EXISTS `hotelmanagement`.`eveningshift` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`eveningshift` (

`EmployeeID` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`shift` VARCHAR(45) NULL DEFAULT NULL)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`morningshift`

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

DROP TABLE IF EXISTS `hotelmanagement`.`morningshift` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`morningshift` (

`EmployeeID` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`shift` VARCHAR(45) NULL DEFAULT NULL)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`newcustomer`

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

DROP TABLE IF EXISTS `hotelmanagement`.`newcustomer` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`newcustomer` (

`CustomerID` INT(11) NOT NULL DEFAULT '0',

`ReceptionID\_FK` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`gender` VARCHAR(1) NULL DEFAULT NULL,

`City` VARCHAR(45) NULL DEFAULT NULL,

`Zip` INT(5) NULL DEFAULT NULL,

`IDType` VARCHAR(45) NULL DEFAULT NULL,

`IDNo` VARCHAR(45) NULL DEFAULT NULL,

`ContactNo` VARCHAR(45) NULL DEFAULT NULL,

`EmailID` VARCHAR(45) NULL DEFAULT NULL,

`Booking` ENUM('Online', 'Offline') NULL DEFAULT NULL)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`nightshift`

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

DROP TABLE IF EXISTS `hotelmanagement`.`nightshift` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`nightshift` (

`EmployeeID` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`shift` VARCHAR(45) NULL DEFAULT NULL)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`offlinebooking`

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

DROP TABLE IF EXISTS `hotelmanagement`.`offlinebooking` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`offlinebooking` (

`OfflineID` INT(11) NOT NULL AUTO\_INCREMENT,

`Customer\_CustomerID` INT(11) NOT NULL,

PRIMARY KEY (`OfflineID`),

INDEX `fk\_OfflineBooking\_Customer1\_idx` (`Customer\_CustomerID` ASC),

CONSTRAINT `fk\_OfflineBooking\_Customer1`

FOREIGN KEY (`Customer\_CustomerID`)

REFERENCES `hotelmanagement`.`customer` (`CustomerID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`offlinecustomer`

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

DROP TABLE IF EXISTS `hotelmanagement`.`offlinecustomer` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`offlinecustomer` (

`CustomerID` INT(11) NOT NULL DEFAULT '0',

`ReceptionID\_FK` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`gender` VARCHAR(1) NULL DEFAULT NULL,

`City` VARCHAR(45) NULL DEFAULT NULL,

`Zip` INT(5) NULL DEFAULT NULL,

`IDType` VARCHAR(45) NULL DEFAULT NULL,

`IDNo` VARCHAR(45) NULL DEFAULT NULL,

`ContactNo` VARCHAR(45) NULL DEFAULT NULL,

`EmailID` VARCHAR(45) NULL DEFAULT NULL,

`Booking` ENUM('Online', 'Offline') NULL DEFAULT NULL)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`onlinebooking`

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

DROP TABLE IF EXISTS `hotelmanagement`.`onlinebooking` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`onlinebooking` (

`OnlineID` INT(11) NOT NULL AUTO\_INCREMENT,

`CustomerID\_FK` INT(11) NOT NULL,

PRIMARY KEY (`OnlineID`),

INDEX `fk\_OnlineBooking\_Customer1\_idx` (`CustomerID\_FK` ASC),

CONSTRAINT `fk\_OnlineBooking\_Customer1`

FOREIGN KEY (`CustomerID\_FK`)

REFERENCES `hotelmanagement`.`customer` (`CustomerID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`onlinecustomer`

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

DROP TABLE IF EXISTS `hotelmanagement`.`onlinecustomer` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`onlinecustomer` (

`CustomerID` INT(11) NOT NULL DEFAULT '0',

`ReceptionID\_FK` INT(11) NOT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`gender` VARCHAR(1) NULL DEFAULT NULL,

`City` VARCHAR(45) NULL DEFAULT NULL,

`Zip` INT(5) NULL DEFAULT NULL,

`IDType` VARCHAR(45) NULL DEFAULT NULL,

`IDNo` VARCHAR(45) NULL DEFAULT NULL,

`ContactNo` VARCHAR(45) NULL DEFAULT NULL,

`EmailID` VARCHAR(45) NULL DEFAULT NULL,

`Booking` ENUM('Online', 'Offline') NULL DEFAULT NULL)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

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

-- Table `hotelmanagement`.`payment`

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

DROP TABLE IF EXISTS `hotelmanagement`.`payment` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`payment` (

`PaymentID` INT(11) NOT NULL AUTO\_INCREMENT,

`BookingID\_FK` INT(11) NOT NULL,

`PaymentMode` VARCHAR(20) NULL DEFAULT NULL,

PRIMARY KEY (`PaymentID`),

INDEX `fk\_Payment\_Booking1\_idx` (`BookingID\_FK` ASC),

CONSTRAINT `fk\_Payment\_Booking1`

FOREIGN KEY (`BookingID\_FK`)

REFERENCES `hotelmanagement`.`booking` (`BookingID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

SHOW WARNINGS;

USE `hotelmanagement` ;

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

-- Placeholder table for view `hotelmanagement`.`bookingdetails`

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

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`bookingdetails` (`BookingID` INT, `Customer Name` INT, `Check In Date` INT, `Check out Date` INT, `Booking Type` INT);

SHOW WARNINGS;

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

-- Placeholder table for view `hotelmanagement`.`food\_details`

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

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`food\_details` (`Room Number` INT, `Food` INT, `Quantity` INT, `Rate` INT, `Amount` INT, `Date of Food Served` INT);

SHOW WARNINGS;

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

-- Placeholder table for view `hotelmanagement`.`staydays`

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

CREATE TABLE IF NOT EXISTS `hotelmanagement`.`staydays` (`Room Number` INT, `Customer Name` INT, `Room Type` INT, `BookingID` INT, `No of Days Stayed` INT);

SHOW WARNINGS;

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

-- procedure backup\_customer\_details

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

USE `hotelmanagement`;

DROP procedure IF EXISTS `hotelmanagement`.`backup\_customer\_details`;

SHOW WARNINGS;

DELIMITER $$

USE `hotelmanagement`$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `backup\_customer\_details`()

BEGIN

CREATE TABLE newCustomer AS SELECT \* FROM customer;

CREATE TABLE onlineCustomer AS SELECT \* FROM customer WHERE Booking="Online";

CREATE TABLE offlineCustomer AS SELECT \* FROM customer WHERE Booking="Offline";

END$$

DELIMITER ;

SHOW WARNINGS;

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

-- procedure count\_rooms

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

USE `hotelmanagement`;

DROP procedure IF EXISTS `hotelmanagement`.`count\_rooms`;

SHOW WARNINGS;

DELIMITER $$

USE `hotelmanagement`$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `count\_rooms`()

begin

SELECT count(rooms.Availability) AS `Available Rooms` FROM rooms WHERE BookingID\_FK IS NULL;

end$$

DELIMITER ;

SHOW WARNINGS;

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

-- procedure shift\_details

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

USE `hotelmanagement`;

DROP procedure IF EXISTS `hotelmanagement`.`shift\_details`;

SHOW WARNINGS;

DELIMITER $$

USE `hotelmanagement`$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `shift\_details`()

BEGIN

CREATE TABLE morningShift AS SELECT \* FROM employees WHERE shift="Morning";

CREATE TABLE afternoonShift AS SELECT \* FROM employees WHERE shift="Afternoon";

CREATE TABLE eveningShift AS SELECT \* FROM employees WHERE shift="Evening";

CREATE TABLE nightShift AS SELECT \* FROM employees WHERE shift="Night";

END$$

DELIMITER ;

SHOW WARNINGS;

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

-- View `hotelmanagement`.`bookingdetails`

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

DROP VIEW IF EXISTS `hotelmanagement`.`bookingdetails` ;

SHOW WARNINGS;

DROP TABLE IF EXISTS `hotelmanagement`.`bookingdetails`;

SHOW WARNINGS;

USE `hotelmanagement`;

CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `hotelmanagement`.`bookingdetails` AS SELECT `hotelmanagement`.`booking`.`BookingID` AS `BookingID`,`hotelmanagement`.`customer`.`name` AS `Customer Name`,`hotelmanagement`.`booking`.`CheckIN` AS `Check In Date`,`hotelmanagement`.`booking`.`CheckOUT` AS `Check out Date`,`hotelmanagement`.`customer`.`Booking` AS `Booking Type` FROM (`hotelmanagement`.`booking` join `hotelmanagement`.`customer` on((`hotelmanagement`.`booking`.`CustomerID\_FK` = `hotelmanagement`.`customer`.`CustomerID`))) WHERE (`hotelmanagement`.`booking`.`BookingID` IS NOT NULL) ORDER BY `hotelmanagement`.`booking`.`BookingID`;

SHOW WARNINGS;

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

-- View `hotelmanagement`.`food\_details`

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

DROP VIEW IF EXISTS `hotelmanagement`.`food\_details` ;

SHOW WARNINGS;

DROP TABLE IF EXISTS `hotelmanagement`.`food\_details`;

SHOW WARNINGS;

USE `hotelmanagement`;

CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `hotelmanagement`.`food\_details` AS SELECT `hotelmanagement`.`rooms`.`RoomID` AS `Room Number`,`hotelmanagement`.`food`.`Type` AS `Food`,`hotelmanagement`.`restaurant`.`Quantity` AS `Quantity`,`hotelmanagement`.`food`.`Rate` AS `Rate`,(`hotelmanagement`.`restaurant`.`Quantity` \* `hotelmanagement`.`food`.`Rate`) AS `Amount`,`hotelmanagement`.`restaurant`.`ServeDate` AS `Date of Food Served` FROM ((`hotelmanagement`.`rooms` join `hotelmanagement`.`restaurant` on((`hotelmanagement`.`rooms`.`RoomID` = `hotelmanagement`.`restaurant`.`RoomID\_FK`))) join `hotelmanagement`.`food` on((`hotelmanagement`.`restaurant`.`FoodID\_FK` = `hotelmanagement`.`food`.`FoodID`))) ORDER BY `hotelmanagement`.`rooms`.`RoomID`;

SHOW WARNINGS;

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

-- View `hotelmanagement`.`staydays`

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

DROP VIEW IF EXISTS `hotelmanagement`.`staydays` ;

SHOW WARNINGS;

DROP TABLE IF EXISTS `hotelmanagement`.`staydays`;

SHOW WARNINGS;

USE `hotelmanagement`;

CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `hotelmanagement`.`staydays` AS SELECT `hotelmanagement`.`rooms`.`RoomID` AS `Room Number`,`hotelmanagement`.`customer`.`name` AS `Customer Name`,`hotelmanagement`.`roomtype`.`roomType` AS `Room Type`,`hotelmanagement`.`booking`.`BookingID` AS `BookingID`,(`hotelmanagement`.`booking`.`CheckOUT` - `hotelmanagement`.`booking`.`CheckIN`) AS `No of Days Stayed` FROM (`hotelmanagement`.`roomtype` left join ((`hotelmanagement`.`customer` join `hotelmanagement`.`booking` on((`hotelmanagement`.`customer`.`CustomerID` = `hotelmanagement`.`booking`.`CustomerID\_FK`))) join `hotelmanagement`.`rooms` on((`hotelmanagement`.`booking`.`BookingID` = `hotelmanagement`.`rooms`.`BookingID\_FK`))) on((`hotelmanagement`.`rooms`.`RoomTypeID\_FK` = `hotelmanagement`.`roomtype`.`RoomTypeID`))) WHERE ((`hotelmanagement`.`booking`.`BookingID` IS NOT NULL) and (`hotelmanagement`.`booking`.`CheckIN` IS NOT NULL) and (`hotelmanagement`.`booking`.`CheckOUT` IS NOT NULL)) ORDER BY `hotelmanagement`.`rooms`.`RoomID`;

SHOW WARNINGS;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**INSERTED VALUES:**

**1.)** **CUSTOMER TABLE**

SELECT \* FROM hotelmanagement.customer;

INSERT INTO `hotelmanagement`.`customer` (`CustomerID`, `ReceptionID\_FK`, `name`, `gender`, `City`, `Zip`, `IDType`, `IDNo`, `ContactNo`, `EmailID`, `Booking`) VALUES ('1', '1', 'SUYOG', 'M', 'BOSTON', '02120', 'PASSPORT', 'Z4127946', '8573995537', 'suyogsathe7@gmail.com', 'Online');

INSERT INTO `hotelmanagement`.`customer` (`CustomerID`, `ReceptionID\_FK`, `name`, `gender`, `City`, `Zip`, `IDType`, `IDNo`, `ContactNo`, `EmailID`, `Booking`) VALUES ('2', '1', 'ROHAN', 'M', 'BOSTON', '02120', 'PASSPORT', 'Z4127945', '8573995536', 'rohan@gmail.com', 'Offline');

INSERT INTO `hotelmanagement`.`customer` (`CustomerID`, `ReceptionID\_FK`, `name`, `gender`, `City`, `Zip`, `IDType`, `IDNo`, `ContactNo`, `EmailID`, `Booking`) VALUES ('3', '1', 'CHINAMY', 'M', 'BOSTON', '02120', 'PASSPORT', 'Z4127944', '8573995535', 'chinmay@gmail.com', 'Online');

INSERT INTO `hotelmanagement`.`customer` (`CustomerID`, `ReceptionID\_FK`, `name`, `gender`, `City`, `Zip`, `IDType`, `IDNo`, `ContactNo`, `EmailID`, `Booking`) VALUES ('4', '1', 'CHAITANYA', 'M', 'BOSTON', '02120', 'PASSPORT', 'Z4127943', '8573995534', 'chaitanya@gmail.com', 'Online');

INSERT INTO `hotelmanagement`.`customer` (`CustomerID`, `ReceptionID\_FK`, `name`, `gender`, `City`, `Zip`, `IDType`, `IDNo`, `ContactNo`, `EmailID`, `Booking`) VALUES ('5', '1', 'MAYUR', 'M', 'BOSTON', '02120', 'PASSPORT', 'Z4127942', '8573995533', 'mayur@gmail.com', 'Offline');

INSERT INTO `hotelmanagement`.`customer` (`CustomerID`, `ReceptionID\_FK`, `name`, `gender`, `City`, `Zip`, `IDType`, `IDNo`, `ContactNo`, `EmailID`, `Booking`) VALUES ('6', '1', 'SAMARTH', 'M', 'BOSTON', '02120', 'PASSPORT', 'Z4127941', '8573995532', 'samarth@gmail.com', 'Online');

INSERT INTO `hotelmanagement`.`customer` (`CustomerID`, `ReceptionID\_FK`, `name`, `gender`, `City`, `Zip`, `IDType`, `IDNo`, `ContactNo`, `EmailID`, `Booking`) VALUES ('7', '1', 'PRASHANT', 'M', 'BOSTON', '02120', 'PASSPORT', 'Z4127940', '8573995531', 'prAShant@gmail.com', 'Online');

INSERT INTO `hotelmanagement`.`customer` (`CustomerID`, `ReceptionID\_FK`, `name`, `gender`, `City`, `Zip`, `IDType`, `IDNo`, `ContactNo`, `EmailID`, `Booking`) VALUES ('8', '1', 'PRIYESH', 'M', 'BOSTON', '02120', 'PASSPORT', 'Z4127939', '8573995530', 'priyesh@gmail.com', 'Offline');

INSERT INTO `hotelmanagement`.`customer` (`CustomerID`, `ReceptionID\_FK`, `name`, `gender`, `City`, `Zip`, `IDType`, `IDNo`, `ContactNo`, `EmailID`, `Booking`) VALUES ('9', '1', 'VEDANT', 'M', 'BOSTON', '02120', 'PASSPORT', 'Z4127938', '8573995529', 'vedant@gmail.com', 'Online');

INSERT INTO `hotelmanagement`.`customer` (`CustomerID`, `ReceptionID\_FK`, `name`, `gender`, `City`, `Zip`, `IDType`, `IDNo`, `ContactNo`, `EmailID`, `Booking`) VALUES ('10', '1', 'HARSH', 'M', 'BOSTON', '02120', 'PASSPORT', 'Z4127937', '8573995528', 'harsh@gmail.com', 'Online');

truncate table hotelmanagement.customer;

DELETE FROM `hotelmanagement`.`customer` WHERE CustomerID = 6;

**2.)** **BOOKING TABLE**

SELECT \* FROM hotelmanagement.booking;

truncate table hotelmanagement.booking;

INSERT INTO hotelmanagement.booking VALUES('1', '1', '2017-12-06', '2017-12-09', NULL);

INSERT INTO hotelmanagement.booking VALUES('2', '2', '2017-10-15', '2017-10-22', NULL);

INSERT INTO hotelmanagement.booking VALUES('3', '3', '2017-10-22', '2017-10-23', NULL);

INSERT INTO hotelmanagement.booking VALUES('4', '4', '2017-11-17', NULL, NULL);

INSERT INTO hotelmanagement.booking VALUES('5', '5', '2017-12-11', '2017-12-13', NULL);

INSERT INTO hotelmanagement.booking VALUES('6', '6', '2017-12-06', '2017-12-09', NULL);

INSERT INTO hotelmanagement.booking VALUES('7', '7', '2017-12-01', NULL, NULL);

INSERT INTO hotelmanagement.booking VALUES('8', '8', '2017-12-27', '2017-12-28', NULL);

INSERT INTO hotelmanagement.booking VALUES('9', '9', '2017-08-17', NULL, NULL);

INSERT INTO hotelmanagement.booking VALUES('10', '10', '2017-10-22', NULL, NUL

**3.)** **ROOMS**

SELECT \* FROM rooms;

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('101', '1', '2',false);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('102', '2', '5',false);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('103', '3', '4',false);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('201', '3', '3',false);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('202', '2', '1',false);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('203', '3', '4',false);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('301', '1', '3',false);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('302', '2', '2',false);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('303', '3', '1',false);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('104', '1', null,true);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('204', '2', null,true);

INSERT INTO `hotelmanagement`.`rooms` (`RoomID`, `RoomTypeID\_FK`, `BookingID\_FK`,`availability`) VALUES ('304', '3', null,true);

truncate table rooms;

**4.) ROOM\_TYPE**

SELECT \* FROM hotelmanagement.roomtype;

INSERT INTO `hotelmanagement`.`roomtype` (`RoomTypeID`, `roomType`, `rate`) VALUES ('1', 'Deluxe', '400');

INSERT INTO `hotelmanagement`.`roomtype` (`RoomTypeID`, `roomType`, `rate`) VALUES ('2', 'Premium', '500');

INSERT INTO `hotelmanagement`.`roomtype` (`RoomTypeID`, `roomType`, `rate`) VALUES ('3', 'Supreme', '600');

truncate table roomtype;

**5.) FOOD**

SELECT \* FROM hotelmanagement.food;

INSERT INTO `hotelmanagement`.`food` (`FoodID`, `Type`, `Rate`) VALUES ('1', 'BreakfASt', '20');

INSERT INTO `hotelmanagement`.`food` (`FoodID`, `Type`, `Rate`) VALUES ('2', 'Lunch', '40');

INSERT INTO `hotelmanagement`.`food` (`FoodID`, `Type`, `Rate`) VALUES ('3', 'Dinner', '60');

truncate table food;

**6.)** **RESTAURANT**

SELECT \* FROM hotelmanagement.restaurant;

INSERT INTO `hotelmanagement`.`restaurant` (`RestaurantID`, `RoomID\_FK`, `FoodID\_FK`, `Quantity`, `ServeDate`, `Hotel\_HotelID`) VALUES ('1', '101', '1', '3', '2017-12-06', '1');

INSERT INTO `hotelmanagement`.`restaurant` (`RestaurantID`, `RoomID\_FK`, `FoodID\_FK`, `Quantity`, `ServeDate`, `Hotel\_HotelID`) VALUES ('2', '201', '3', '1', '2017-11-21', '1');

INSERT INTO `hotelmanagement`.`restaurant` (`RestaurantID`, `RoomID\_FK`, `FoodID\_FK`, `Quantity`, `ServeDate`, `Hotel\_HotelID`) VALUES ('3', '102', '2', '5', '2017-10-12', '1');

INSERT INTO `hotelmanagement`.`restaurant` (`RestaurantID`, `RoomID\_FK`, `FoodID\_FK`, `Quantity`, `ServeDate`, `Hotel\_HotelID`) VALUES ('4', '202', '2', '12', '2017-12-07', '1');

INSERT INTO `hotelmanagement`.`restaurant` (`RestaurantID`, `RoomID\_FK`, `FoodID\_FK`, `Quantity`, `ServeDate`, `Hotel\_HotelID`) VALUES ('5', '102', '1', '3', '2017-12-08', '1');

INSERT INTO `hotelmanagement`.`restaurant` (`RestaurantID`, `RoomID\_FK`, `FoodID\_FK`, `Quantity`, `ServeDate`, `Hotel\_HotelID`) VALUES ('6', '301', '3', '7', '2017-12-09', '1');

INSERT INTO `hotelmanagement`.`restaurant` (`RestaurantID`, `RoomID\_FK`, `FoodID\_FK`, `Quantity`, `ServeDate`, `Hotel\_HotelID`) VALUES ('7', '303', '3', '2', '2017-12-10', '1');

INSERT INTO `hotelmanagement`.`restaurant` (`RestaurantID`, `RoomID\_FK`, `FoodID\_FK`, `Quantity`, `ServeDate`, `Hotel\_HotelID`) VALUES ('8', '303', '2', '1', '2017-12-11', '1');

INSERT INTO `hotelmanagement`.`restaurant` (`RestaurantID`, `RoomID\_FK`, `FoodID\_FK`, `Quantity`, `ServeDate`, `Hotel\_HotelID`) VALUES ('9', '300', '2', '3', '2017-12-12', '1');

INSERT INTO `hotelmanagement`.`restaurant` (`RestaurantID`, `RoomID\_FK`, `FoodID\_FK`, `Quantity`, `ServeDate`, `Hotel\_HotelID`) VALUES ('10', '200', '1', '1', '2017-12-13', '1');

truncate table restaurant;

**7.) HOTEL**

INTO `hotelmanagement`.`hotel` (`HotelID`, `name`, `street`, `city`, `zip`) VALUES ('1', 'Suswagat', 'Dattapada Road', 'Mumbai', 40006);

**8.) DEPARTMENT**

SELECT \* FROM hotelmanagement.department;

INSERT INTO department VALUES (1,"Cooking",1);

INSERT INTO department VALUES (2,"Security",1);

INSERT INTO department VALUES (3,"Housekeeping",1);

**9.) EMPLOYEES**

SELECT \* FROM hotelmanagement.employees;

INSERT INTO employees VALUES (1,"Abhay","Morning");

INSERT INTO employees VALUES (2,"Suresh","Afternoon");

INSERT INTO employees VALUES (3,"Ramesh","Morning");

INSERT INTO employees VALUES (4,"AvinASh","Evening");

INSERT INTO employees VALUES (5,"Rohit","Night");

INSERT INTO employees VALUES (6,"Ravi","Morning");

INSERT INTO employees VALUES (7,"Dan","Night");

INSERT INTO employees VALUES (8,"Cyrus","Afternoon");

INSERT INTO employees VALUES (9,"John","Afternoon");

INSERT INTO employees VALUES (10,"Ann","Morning");

**10.) DEPARTMENT HAS EMPLOYEES**

SELECT \* FROM hotelmanagement.department\_hAS\_employees;

INSERT INTO hotelmanagement.department\_hAS\_employees VALUES (1,6);

INSERT INTO hotelmanagement.department\_hAS\_employees VALUES (1,9);

INSERT INTO hotelmanagement.department\_hAS\_employees VALUES (1,7);

INSERT INTO hotelmanagement.department\_hAS\_employees VALUES (2,1);

INSERT INTO hotelmanagement.department\_hAS\_employees VALUES (2,5);

INSERT INTO hotelmanagement.department\_hAS\_employees VALUES (2,2);

INSERT INTO hotelmanagement.department\_hAS\_employees VALUES (3,3);

INSERT INTO hotelmanagement.department\_hAS\_employees VALUES (3,4);

INSERT INTO hotelmanagement.department\_hAS\_employees VALUES (3,8);

INSERT INTO hotelmanagement.department\_hAS\_employees VALUES (3,10);