Resort Management System

Dinesh Sai Pappuru

Sejal Hirji Gothi

SQL DDL Queries:

```
CREATE TABLE `bill` (
  `customer_id` int NOT NULL,
  `bill` int NOT NULL,
  PRIMARY KEY (`customer_id`),
  CONSTRAINT `bill_ibfk_1` FOREIGN KEY (`customer_id`) REFERENCES `customer` (`customer_id`)
)
CREATE TABLE `booking` (
  `booking_id` int NOT NULL AUTO_INCREMENT,
  `customer_id` int DEFAULT NULL,
  `room_id` int DEFAULT NULL,
  `check_in` datetime DEFAULT NULL,
  `check_out` datetime DEFAULT NULL,
  `meal_plan_id` int DEFAULT NULL,
  PRIMARY KEY (`booking_id`),
  KEY `room_id` (`room_id`),
  KEY `booking_ibfk_2` (`customer_id`),
  CONSTRAINT `booking_ibfk_1` FOREIGN KEY (`room_id`) REFERENCES `rooms` (`room_id`),
  CONSTRAINT `booking_ibfk_2` FOREIGN KEY (`customer_id`) REFERENCES `customer` (`customer_id`)
)

CREATE TABLE `customer` (
  `customer_id` int NOT NULL AUTO_INCREMENT,
  `first_name` char(20) NOT NULL,
  `last_name` char(20) NOT NULL,
  `address` varchar(50) DEFAULT NULL,
  `contact` bigint NOT NULL,
  `date` datetime NOT NULL,
  PRIMARY KEY (`customer_id`),
  UNIQUE KEY `contact_UNIQUE` (`contact`)
)

CREATE TABLE `department` (
  `department_id` int NOT NULL AUTO_INCREMENT,
  `resort_id` int DEFAULT NULL,
  `dep_name` varchar(20) NOT NULL,
  `description` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`department_id`),
  KEY `resort_id` (`resort_id`),
  CONSTRAINT `department_ibfk_1` FOREIGN KEY (`resort_id`) REFERENCES `resort` (`resort_id`)
)

CREATE TABLE `employee` (
  `employee_id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(20) NOT NULL,
  `age` int NOT NULL,
  `salary` int NOT NULL,
  `department_id` int NOT NULL,
  `mobile_no` bigint NOT NULL,
  PRIMARY KEY (`employee_id`),
  KEY `department_id` (`department_id`),
  CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`department_id`) REFERENCES `department` (`department_id`)
)

CREATE TABLE `feedback` (
  `customer_id` int NOT NULL,
  `rating` decimal(2,1) NOT NULL,
  `feedback` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`customer_id`),
  CONSTRAINT `feedback_ibfk_1` FOREIGN KEY (`customer_id`) REFERENCES `customer` (`customer_id`)
)

CREATE TABLE `meal_plan` (
  `meal_plan_id` int NOT NULL AUTO_INCREMENT,
  `resort_id` int NOT NULL,
  `name` varchar(50) NOT NULL,
  `description` varchar(50) DEFAULT NULL,
  `cost` int NOT NULL,
  PRIMARY KEY (`meal_plan_id`),
  KEY `resort_id` (`resort_id`),
  CONSTRAINT `meal_plan_ibfk_1` FOREIGN KEY (`resort_id`) REFERENCES `resort` (`resort_id`)
)

CREATE TABLE `membership` (
  `contact` bigint NOT NULL,
  `discount` int NOT NULL,
  PRIMARY KEY (`contact`),
  CONSTRAINT `membership_ibfk_1` FOREIGN KEY (`contact`) REFERENCES `customer` (`contact`)
)

CREATE TABLE `resort` (
  `resort_id` int NOT NULL AUTO_INCREMENT,
  `resort_name` varchar(20) NOT NULL,
  `location` varchar(50) NOT NULL,
  `contact_info` int DEFAULT NULL,
  `rating` int DEFAULT NULL,
  PRIMARY KEY (`resort_id`)
```

```
)

CREATE TABLE `room_status` (
  `room_id` int NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`room_id`),
  CONSTRAINT `room_status_ibfk_1` FOREIGN KEY (`room_id`) REFERENCES `rooms` (`room_id`)
)

CREATE TABLE `room_type` (
  `room_id` int NOT NULL,
  `room_type` varchar(20) NOT NULL,
  `discount_percent` int NOT NULL,
  `smoke_friendly` tinyint(1) NOT NULL DEFAULT '1',
  `pet_friendly` tinyint(1) NOT NULL DEFAULT '1',
  `cost` int NOT NULL,
  PRIMARY KEY (`room_id`),
  CONSTRAINT `room_type_ibfk_1` FOREIGN KEY (`room_id`) REFERENCES `rooms` (`room_id`)
)

CREATE TABLE `rooms` (
  `room_no` int NOT NULL,
  `room_id` int NOT NULL AUTO_INCREMENT,
  `resort_id` int NOT NULL,
  PRIMARY KEY (`room_id`),
  KEY `resort_id` (`resort_id`),
  CONSTRAINT `rooms_ibfk_1` FOREIGN KEY (`resort_id`) REFERENCES `resort` (`resort_id`)
)

CREATE TABLE `utilities` (
  `utility_id` int NOT NULL AUTO_INCREMENT,
  `resort_id` int NOT NULL,
  `utility_name` varchar(20) NOT NULL,
  `decription` varchar(50) DEFAULT NULL,
  `status` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`utility_id`),
  KEY `resort_id` (`resort_id`),
  CONSTRAINT `utilities_ibfk_1` FOREIGN KEY (`resort_id`) REFERENCES `resort` (`resort_id`)
)
```

Views:

```
CREATE VIEW employee_info AS
    SELECT
        name, age, mobile_no
    FROM
        Employee
```

```
CREATE VIEW available_utilities AS
    SELECT
        utility_name
    FROM
        utilities
    WHERE
        status = 1;
```

Triggers:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `update_booking_on_customer` AFTER INSERT ON
`customer` FOR EACH ROW begin
                declare temp int;
        set @temp = (select MAX(customer_id) from customer);
                insert into booking(customer_id) values (@temp);
end
```

Functions:

```
Delimiter $$

create function bill_amount_after_discount(cont bigint)
returns int
deterministic
begin
        declare discounted_bill int;
    select (b.bill - ((m.discount/100)* b.bill)) into discounted_bill
    from membership m, customer c, bill b
    where m.contact = c.contact and
                      c.customer_id = b.customer_id and
          m.contact = cont;
        return discounted_bill;
end $$
delimiter ;
```

```
delimiter $$
create function room_availibity(room_number int)
returns varchar(20)
deterministic
begin
        declare status int;
    declare output varchar(20);
    select rs.status into status
    from rooms r, room_status rs
    where r.room_id = rs.room_id and
                      r. room_no = room_number;

        if status = 1 then
                    set output = 'Available';
    else
```

```
                set output = 'Not Available';
        end if;
    return output;
end $$
delimiter ;
```

```
delimiter $$
create function peak_time_of_the_year()
returns varchar(20)
deterministic
begin
        declare month varchar(20);
        select monthname(check_in) into month
        from booking
        group by Month(check_in)
        order by Month(check_in) desc
        limit 1;
        return month;
end $$
delimiter ;
```

Stored Procedure:

```
delimiter $$;
create procedure get_membership_discount(in contact bigint)
begin
        select first_name, last_name, discount
                from customer c, membership m
        where c.contact = m.contact
end$$
delimiter ;
delimiter $$
create procedure get_most_used_meal_plan_details()
begin
    select *
  from booking b, meal_plan m
  where b.meal_plan_id = m.meal_plan_id
  group by b.meal_plan_id
  order by b.meal_plan_id desc
  limit 1;
end$$
delimiter ;
```

```
delimiter $$
create procedure get_department_employees( in dept_name varchar(20))
begin
    select d.dep_name, count(e.employee_id) as no_of_employees
  from department d, employee e
  where d.department_id = e.department_id and d.dep_name = dept_name
  group by e.department_id;

end$$
delimiter ;
```