

```

/*queries*/
/*1*/
    select room_id, room_status from room;
/*2*/
    select *, count(*) from reservation
    group by room_id order by(count(*)) desc limit 10;
/*3*/
    SELECT *
    FROM reservation_details
    WHERE check_in >= date_sub(current_timestamp(), INTERVAL 14 day);
/*4*/
    SELECT employee_id, employee_first_name, employee_last_name, count(*) as
num_of_cleans
    FROM cleaned_by
    INNER JOIN employees
    USING (employee_id)
    group by employee_id
    order by(count(*)) desc limit 1;

/*5*/
    select r.reservation_id, c.client_id, client_first_name, client_last_name
    from reservation_details rd
    inner join reservation r
    on rd.reservation_id = r.reservation_id
    inner join client_c
    on c.client_id = r.client_id
    where check_out >= current_date();
/*6*/
    select *, count(*) from reservation
    inner join client_
    using(client_id)
    group by client_id having count(*) > 1;
/*7*/
    select YEAR(check_in) AS year, MONTH(check_in) AS month, sum(calc_price) as revenue
    from reservation_details
    group by month(check_in)
    order by year desc;

/*Procedures*/
/*1*/
    DELIMITER $$
CREATE PROCEDURE update_startorfinish_cleaning(IN room_id_ INT ,IN employee_id_ INT
,IN action_type VARCHAR(20))
BEGIN
    IF(action_type='action_start')
    THEN

```

```

        insert into cleaned_by (room_id,employee_id,clean_time)
        values (room_id_, employee_id_,current_timestamp());
    update room
        set room.room_status="Waiting for cleaning", changed_status_time =
current_timestamp()
        where room.room_id=room_id_;
    end if;
    IF(action_type='action_end')
        THEN
            update cleaned_by
            set clean_finish = current_timestamp()
            where cleaned_by.room_id = room_id_ and
employee_id=employee_id_ and clean_finish is null;
            update room
            set room.room_status="Available", changed_status_time = current_timestamp()
            where room.room_id=room_id_;
            end if;

END
$$
DELIMITER ;

```

```

DROP PROCEDURE update_startorfinish_cleaning;
select * from cleaned_by;
call update_startorfinish_cleaning(4,2,'action_end');

/*2*/
/*function for the procedure*/
DELIMITER $$
CREATE FUNCTION
    temp_status(res_id_ INT) RETURNS VARCHAR(40)
BEGIN
    DECLARE roomstatus VARCHAR(40) DEFAULT 'The room doesnt exist';
    SELECT
        room_status
    INTO roomstatus
    FROM
room
    INNER JOIN
reservation
    using (room_id)
    WHERE
        reservation_id=res_id_;
    RETURN roomstatus;
END $$

```

```

DELIMITER ;

-- the procedure
DELIMITER $$

CREATE PROCEDURE change_status(IN res_id INT)
    -- In case the room is Available
    BEGIN
    DECLARE new_res_id INT ;
    DECLARE new_room_status VARCHAR (30);
    SET new_room_status= temp_status(res_id_); -- using the utility function temp_status
    for getting the room_status from reservation_id
    CASE
        WHEN new_room_status='Available'
        THEN
        UPDATE room
        INNER JOIN reservation AS r
        ON room.room_id=r.room_id
        SET
changed_status_time=current_timestamp(),room_status='Occupied'
        WHERE reservation_id=res_id_;
        -- In case the room is Occupied
        WHEN new_room_status='Occupied'
        THEN
        UPDATE room
        INNER JOIN reservation AS r
        ON room.room_id=r.room_id
        SET
changed_status_time=current_timestamp(),room_status='Waiting for cleaning'
        WHERE reservation_id=res_id_;
        -- In case the room is Waiting for cleaning
        WHEN new_room_status='Waiting for cleaning'
        THEN
        UPDATE room
        INNER JOIN reservation AS r
        ON room.room_id=r.room_id
        SET
changed_status_time=current_timestamp(),room_status='Available'
        WHERE reservation_id=res_id_;
    END CASE;
    END
    $$
DELIMITER ;

-- Procedure call: dont forget to call the procedure function in line 65 - temp_status
select* from reservation;

```

```
select *from room;
call change_status(20);
DROP PROCEDURE change_status;
```

```
/*function*/
DELIMITER $$
CREATE FUNCTION
    room_status(room_id_ INT) RETURNS varchar(40)
BEGIN
DECLARE roomstatus varchar(40) default 'The room doesnt exist';
SELECT
    room_status
INTO roomstatus FROM
    room
WHERE
    room_id=room_id_;
RETURN roomstatus;
END$$
DELIMITER ;

drop function room_status;
set global log_bin_trust_function_creators = 1;
select room_status(10);
select * from room;
```

-- Calculated price query - calculating the calc\_price of all elements

```
update reservation_details rd
inner join reservation as r
on rd.reservation_id=r.reservation_id
inner join room
on r.room_id=room.room_id
inner join room_price rp
on room.room_type=rp.room_type
set calc_price = DATEDIFF( rd.check_out,rd.check_in)*price_per_day ;
```