Questions on View (25 marks +25 marks)

Q1. Create a view for the ARC administrator called **"Top_Machines_Used"** which contains the following data for the interval of 6 months from Jan to June

Equipment Name | Total Number Of Days Used | Number Of Unique Users Using Equipment| Rank

Also, rank refers to the (non-unique) ranking based on the number of users using the machine.
Limit your results to top 15 machines used.

```
CREATE VIEW Top_Machines_Used AS
SELECT eq.equipment_type as`Equipment Name`,count(distinct timestamp) as`Total Number of Days Used`,count(distinct card_id) as `Number Of Unique Users Using Equipment` , RANK() OVER(ORDER BY count(distinct card_id) desc ) as `Rank`
FROM usage_reading ur
join equipment eq on ur.equipment_id = eq.equipment_id
where ur.timestamp between '2023-01-01' AND '2023-06-30'
group by equipment_type
order by count(distinct card_id) desc
limit 15
```

Q2. Create a view "Machines_Used_By_Day_Of_Week" that shows
Equipment Name | Day of Week | Type of Member | Count

Type of Member is student_type if member is a student, member_type if user is non_student or 'Family' otherwise.
Day Of Week is Monday/Tuesday/Wednesday etc etc
Count is the count of each instance of a member type using an equipment in a particular day
The view should roll up across both days of week and type of member

```
CREATE VIEW Machines_Used_By_Day_Of_Week AS
SELECT eq.equipment_type as`Equipment Name`,DAYNAME(ur.timestamp) as`Day of Week`,member_type as `Type of Member` , Count(*) as Count
FROM usage_reading ur
join equipment eq on ur.equipment_id = eq.equipment_id
left join (select card_id,student_type as member_type from student union all select card_id, member_type from non_student union all select card_id, 'Family' as member_type from family)s on s.card_id =ur.card_id
group by equipment_type,member_type,DAYNAME(ur.timestamp) with rollup;
```

`Equipment Name`,Day of Week`,`Type of Member`,`Count`

Question on Trigger (5 marks)

Q3. Create a row level trigger that no update can reduce an employee salary.

```
DELIMITER |
CREATE TRIGGER NoLowerSalary BEFORE UPDATE ON employee
FOR EACH ROW
BEGIN
IF (OLD.salary_hour > NEW.salary_hour) THEN
        SET NEW.salary_hour= OLD.salary_hour;
END IF;
end;|
DELIMITER ;
```

Question on Constraints (5 marks)

Q4. Create a tuple level check constraint that checks that all employees make atleast 12 dollars per hour

```
ALTER TABLE employee
ADD CONSTRAINT chk_salary_range
CHECK (salary_hour >= 12);
```

Recursive Queries (25 marks)

Q5. Find the maximum length of supervisor employees for any employee of ARC?
(Eg if an employee reports to someone who in turn reports to someone without a boss, their length is 2)

```
WITH RECURSIVE employee_hierarchy AS (
   -- Base case
   SELECT card_id, supervisor_card_id, 1 as depth
```

```
    FROM employee
    WHERE supervisor_card_id IS NULL

    UNION ALL

    -- Recursive case
    SELECT e.card_id, e.supervisor_card_id,eh.depth +1
    FROM employee e
    INNER JOIN employee_hierarchy eh ON e.supervisor_card_id = eh.card_id
)
SELECT max(depth) FROM employee_hierarchy as max_depth;
```

Rank Query (15 marks)

Q6: Find the 2nd youngest employee who earns the most salary in ARC

```
select A.name from
(select name, rank() over (partition by salary_hour order by dob desc ) as p_rank, dob,
salary_hour
from employee, person
where employee.card_id = person.card_id
order by salary_hour )A
where salary_hour in (select max(salary_hour) from employee) and A.p_rank = 2
```