

Due Date: 3rd December 11:59 PM

Note: We modified the schema of the employee table slightly. Please update the new definition and sample data of the employee table in your local test database.

```
DROP TABLE IF EXISTS `employee`;
```

```
CREATE TABLE `employee` (  
  `card_id` int(11) NOT NULL,  
  `supervisor_card_id` int(11) ,  
  `schedule` varchar(255) NOT NULL,  
  `employee_type` varchar(255) NOT NULL,  
  `salary_hour` decimal(10,2) NOT NULL,  
  PRIMARY KEY (`card_id`),  
  CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`card_id`) REFERENCES `person` (`card_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (1, NULL,  
'6am-2pm', 'student', '20.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (2, 1, '1pm-10pm',  
'student', '19.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (3, 1, '9am-7pm',  
'student', '19.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (4, 1, '9am-7pm',  
'student', '20.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (5, 1, '1pm-10pm',  
'student', '23.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (6, 2, '1pm-10pm',  
'student', '24.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (7, 2, '8am-4pm',  
'student', '22.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (8, 2, '1pm-10pm',  
'student', '23.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (9, 2, '9am-7pm',  
'full time', '20.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (10, 2, '1pm-10pm',  
'student', '25.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (11, 3, '1pm-10pm',  
'full time', '21.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (12, 3, '1pm-10pm',  
'student', '19.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (13, 3, '11am-9pm',  
'full time', '20.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (14, 3, '6am-2pm',  
'student', '22.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (15, 3, '8am-4pm',  
'full time', '20.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (16, 4, '6am-2pm',  
'full time', '21.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (17, 4, '8am-4pm',  
'full time', '20.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (18, 4, '9am-7pm',  
'student', '20.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (19, 4, '6am-2pm',  
'full time', '24.00');  
INSERT INTO `employee` (`card_id`, `supervisor_card_id`, `schedule`, `employee_type`, `salary_hour`) VALUES (20, 4, '8am-4pm',  
'student', '19.00');
```

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.

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

Question on Trigger (5 marks)

Q3. Create a row level trigger named “NoLowerSalary” that no update can reduce an employee salary.

Question on Constraints (5 marks)

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

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)

Rank Query (15 marks)

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