# Assignment: 3 [100 points]

Please submit your assignment to NYU Brightspace with a single PDF document attachment. Please mention Student ID, Name, Course, Section Number, and date of submission on the first page of your submission. This is an individual assignment and you should create your own work. All table names in your submission should have your prefix as your initial, e.g. AP_EMP, where AP is the initial of the student.

Problem 1: 50 points

Consider following relational model and write SQLs to answer given business questions.
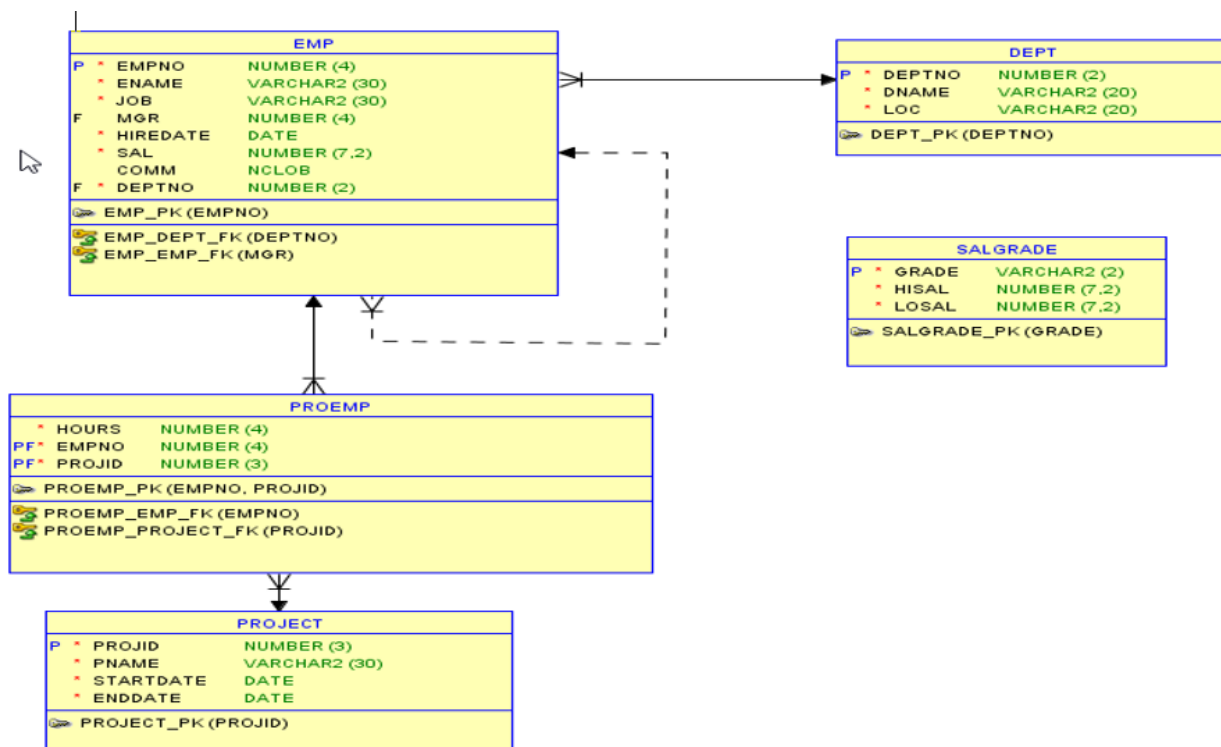
I.   List total salary of each department, only for those departments which total salary is higher than the total salary of employees working in SALES department.

II. List employee name and their respective managers name along with their hire date, for those employees who are hired before their respective manager. Use proper column alias to represent result data.

III. Which employee is 2nd highest earner in department located at NEW YORK and how much salary he/she is having lesser as compared to highest earning employee at NEW YORK? Your answer should display employee name, salary, department name, and difference in salary as compared to highest earning employee

IV. List employee names who are among top two earners in each job function

V.   List employee number, name, department name, project name and number of hours worked on each project for only those employees whose salary is higher than their respective department's average salary. Arrange results in ascending order of employee number and descending order of hours of project.

Submission: For each of the question above, write a SQL query the produce the desired result. The meaning of name in business question is First Name and Last name both. Use the practice data that you have that match to relational model above. Underneath each question, submit screenshot of SQL query and its result. Please note that you should have your tables with your prefix, e.g. AP_EMP etc.

Problem 2: 20 points

FLIGHT_ID	NODE_ID	STATUS	SCHEDULE
A123	SEA	DEP	10/01/2019 07:00:00
A123	MIA	ARR	10/01/2019 11:00:00
A123	MIA	DEP	10/02/2019 08:00:00
A123	LAX	ARR	10/01/2019 12:00:00
A234	SEA	DEP	10/01/2019 11:00:00
A234	MIA	ARR	10/01/2019 14:00:00

This is a FLIGHT table data for an Airline operator. NODE_ID represents Airport Code, and STATUS represents DEP (Departure) or ARR (Arrival)

FLIGHT_ID	FLIGHT_TYPE	CAPACITY
A123	767	10000
A234	737	7000

This is the SIZE table data that represent total CAPACITY of each FLIGHT_ID and FLIGHT_TYPE

Output 1 -

dates	Flight_count
10/01/2019	2
10/02/2019	1

Q1: Write a SQL query to find total number of unique flights that operated on each day. Your output result should appear as shown as picture Output 1.

Output 2 -

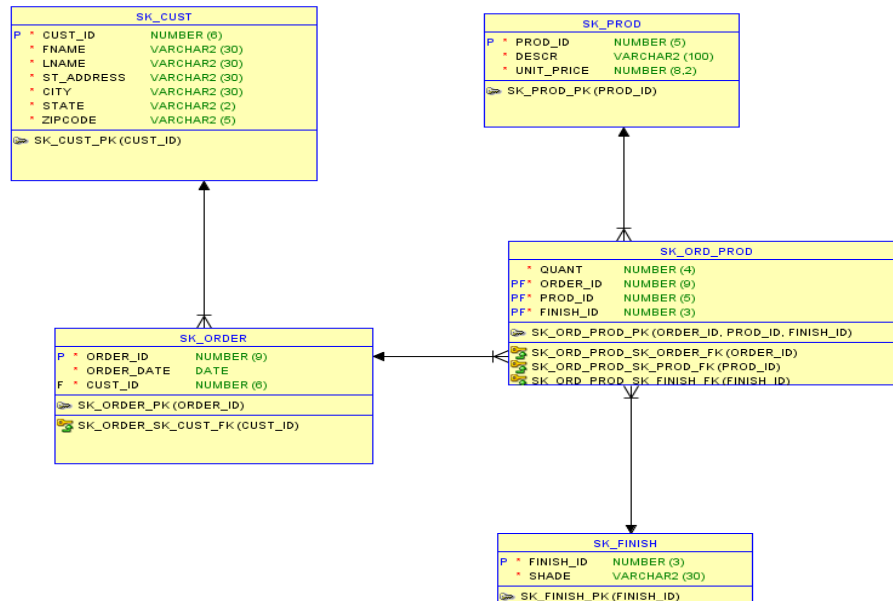| dates	Total_capacity
10/01/2019	17k
10/02/2019	10k

Q2: Write a SQL query to find the total capacity for each day. Your output result should appear as shown in picture Output 2.

Submission:

Create tables as above with your initial as prefix such as AP_FLIGHT, AP_SIZE. Populate exactly the same data as shown in pictures. For Q1 and Q2 write the SQLs that produce the desired results as Output 1 and Output 2. Submit screenshots of your SQL queries and their respective results. You can use Oracle, MySQL or any other relational database.

Problem 3: 30 points

For a given relational model below, please find attached files (in Oracle and MySQL) containing DDL and DMLs. Create tables and insert data by replacing SK to your own initial. You can do this assignment either in Oracle or MySQL. You don't need to draw logical/relational model.

```
                SK_CUST
P * CUST_ID      NUMBER (6)
  * FNAME        VARCHAR2 (30)
  * LNAME        VARCHAR2 (30)
  * ST_ADDRESS   VARCHAR2 (30)
  * CITY         VARCHAR2 (30)
  * STATE        VARCHAR2 (2)
  * ZIPCODE      VARCHAR2 (5)
 SK_CUST_PK (CUST_ID)
```

```
                SK_PROD
P * PROD_ID      NUMBER (5)
  * DESCR        VARCHAR2 (100)
  * UNIT_PRICE   NUMBER (8,2)
 SK_PROD_PK (PROD_ID)
```

```
                SK_ORD_PROD
   * QUANT       NUMBER (4)
PF* ORDER_ID     NUMBER (9)
PF* PROD_ID      NUMBER (5)
PF* FINISH_ID    NUMBER (3)
 SK_ORD_PROD_PK (ORDER_ID, PROD_ID, FINISH_ID)
 SK_ORD_PROD_SK_ORDER_FK (ORDER_ID)
 SK_ORD_PROD_SK_PROD_FK (PROD_ID)
 SK_ORD_PROD_SK_FINISH_FK (FINISH_ID)
```

```
                SK_ORDER
P * ORDER_ID     NUMBER (9)
  * ORDER_DATE   DATE
F * CUST_ID      NUMBER (6)
 SK_ORDER_PK (ORDER_ID)
 SK_ORDER_SK_CUST_FK (CUST_ID)
```

```
                SK_FINISH
P * FINISH_ID    NUMBER (3)
  * SHADE        VARCHAR2 (30)
 SK_FINISH_PK (FINISH_ID)
```

I.   For this relational model of a furniture company, create a read-only database view that represent following dataset. Customer ID, Customer Name (both First and Last name), Order_Id, order date, each product in order with description, quantity, Unit_Price, Total price of each product, and Finish shade. Sort the dataset in order of total order amount. Give appropriate column names in view. Restrict dataset to represent only those orders which has total value over $1000.

II. Find top 3 products and their finish shade in terms of total quantities sold in past 6 months. Your result dataset should have Product_Id, Product Description, shade, and total quantity sold

III. Find products and their finish shade that have not been sold in during the months of Oct, Nov, and Dec in 2020

Submission:
For each question write SQL query and submit both SQL and screenshot of corresponding result underneath each question. Make appropriate use of column alias, and built in functions in your SQL queries.