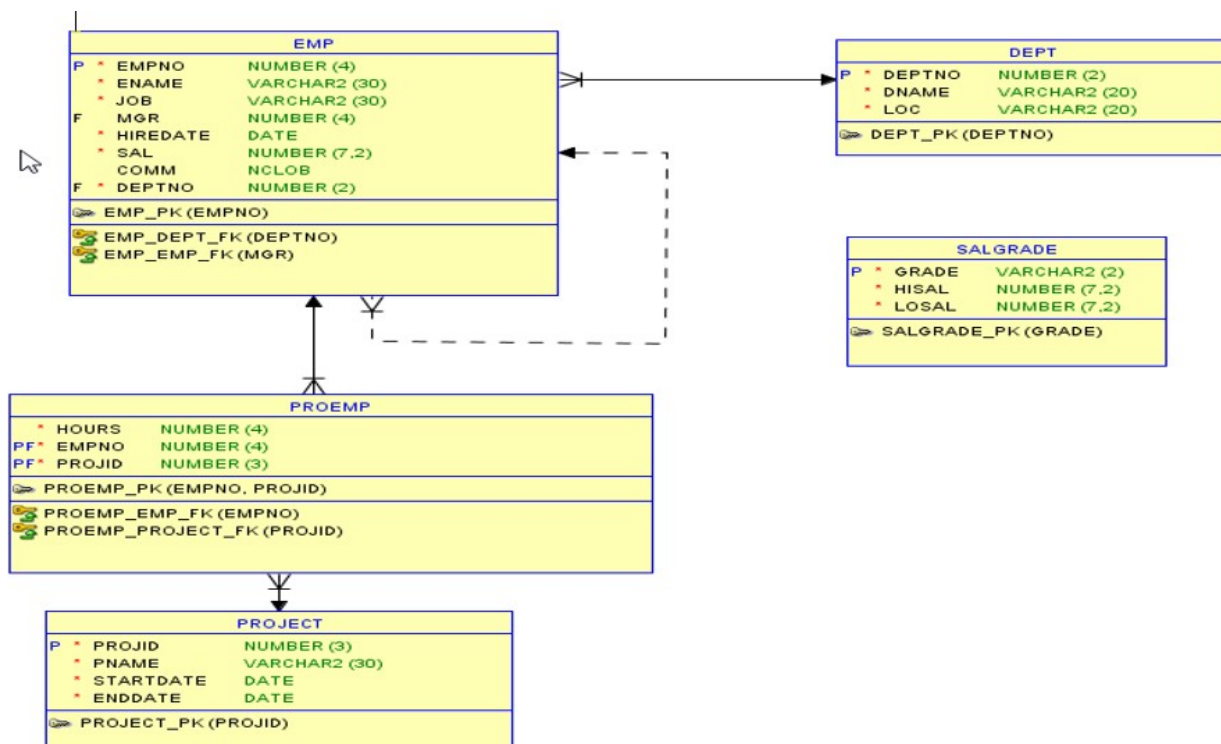


Possible Solution: Assignment: 3 [100 points]

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.

```
SELECT deptno,sum(sal) "Total Salary"
FROM ap_emp
GROUP BY deptno
HAVING sum(sal) > (select sum(sal) from ap_emp where
deptno=(select deptno from ap_dept where
dname='SALES'))
;
```

DEPTNO	Total Salary
20	32625

- 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.

```
SELECT a.empno "Employee Number",a.ename "Employee First Name",
a.lname "Employee Last Name",a.hiredate "Employee Hire Date",
b.empno "Manager employee no", b.lname "Manager Last
Name",b.ename "Manager First Name", b.hiredate "Manager Hire Date"
FROM ap_emp a JOIN ap_emp b ON a.mgr=b.empno
WHERE a.hiredate<b.hiredate;
```

Employee Number	Employee First Name	Employee Last Name	Employee Hire Date	Manager employee no	Manager Last Name	Manager First Name	Manager Hire Date
7369	CHARLES	SMITH	08-OCT-01	7902	FORD	CHRISTIAN	24-SEP-02
7499	GRAYSON	ALLEN	12-DEC-01	7698	GRIFFIN	BLAKE	20-FEB-02
7521	MATTHEW	WARD	14-DEC-01	7698	GRIFFIN	BLAKE	20-FEB-02
7566	NICHOLAS	JONES	22-JAN-02	7839	KING	MARTIN	08-SEP-02
7698	BLAKE	GRIFFIN	20-FEB-02	7839	KING	MARTIN	08-SEP-02
7782	KENT	CLARK	31-MAR-02	7839	KING	MARTIN	08-SEP-02

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

```
select result1.empno,result1.efname,result1.elname,result1.loc,result1.sal,
result2.sal-result1.sal "Diffence in Highsest Salary"
FROM
(select * from (
select empno,elname,efname,sal,loc,
      rank() OVER (PARTITION BY loc order by sal desc) rank1
from ap_emp a JOIN ap_dept b ON a.deptno=b.deptno
where loc='NEW YORK')
where rank1=2
) result1
,
(select * from (
select empno,efname,elname,sal,loc,
      rank() OVER (PARTITION BY loc order by sal desc) rank1
from ap_emp a JOIN ap_dept b ON a.deptno=b.deptno
where loc='NEW YORK')
where rank1=1
) result2 ;
```

EMPNO	EFNAME	ELNAME	LOC	SAL	Diffence in Highsest Salary
7782	KENT	CLARK	NEW YORK	7350	7650

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

```
select * from
(SELECT empno, efname, job, sal,
      RANK() OVER (PARTITION BY job ORDER BY sal desc) AS myrank
FROM ap_emp)
where myrank<=2;
```

EMPNO	EFNAME	JOB	SAL	MYRANK
7788	DEVIN	ANALYST	9000	1
7902	CHRISTIAN	ANALYST	9000	1
7934	MIKE	CLERK	3900	1
7876	JOHN	CLERK	3300	2
7566	NICHOLAS	MANAGER	8925	1
7698	BLAKE	MANAGER	8550	2
7839	MARTIN	PRESIDENT	15000	1
7499	GRAYSON	SALESMAN	4800	1
7844	WILLIAM	SALESMAN	4500	2

- IV. 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.

```

WITH QUALEMP
AS
(select empno,efname,ename,a.deptno,sal,dname
from ap_emp a JOIN ap_dept d ON a.deptno=d.deptno and sal >(select
avg(sal) from ap_emp b where a.deptno=b.deptno group by deptno))
select
qualemp.empno,qualemp.ename,qualemp.efname,qualemp.dname,p.pnam
e,pe.hours
from QUALEMP JOIN ap_proemp pe ON QUALEMP.empno=pe.empno JOIN
ap_project p ON pe.projid=p.projid
order by empno, hours desc;

```

EMPNO	ELNAME	EFNAME	DNAME	PNAME	HOURS
7499	ALLEN	GRAYSON	SALES	Upgrade Marketing Database	42
7566	JONES	NICHOLAS	RESEARCH	Design Backup Procedure	75
7566	JONES	NICHOLAS	RESEARCH	Upgrade Marketing Database	30
7698	GRIFFIN	BLAKE	SALES	Design Backup Procedure	60
7698	GRIFFIN	BLAKE	SALES	Develop Employee Leave System	50
7698	GRIFFIN	BLAKE	SALES	Create Sales Data Warehouse	48
7698	GRIFFIN	BLAKE	SALES	Design and Create New DB	48
7788	BOOKER	DEVIN	RESEARCH	Design Backup Procedure	42
7788	BOOKER	DEVIN	RESEARCH	Develop Employee Leave System	38
7788	BOOKER	DEVIN	RESEARCH	Design and Create New DB	28
7902	FORD	CHRISTIAN	RESEARCH	Upgrade Marketing Database	38
7902	FORD	CHRISTIAN	RESEARCH	Design Backup Procedure	36

Download CSV

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.

Create tables:

```
CREATE TABLE FLIGHT (FLIGHT_ID VARCHAR(4), NODE_ID VARCHAR(3), STATUS VARCHAR(3), SCHEDULE
TIMESTAMP );
```

```
CREATE TABLE FLIGHT_SIZE (FLIGHT_ID VARCHAR(4), FLIGHT_TYPE INT, CAPACITY INT );
```

Populate data:

```
INSERT INTO FLIGHT(FLIGHT_ID, NODE_ID, STATUS, SCHEDULE) VALUES ('A123', 'SEA', 'DEP',
TO_DATE('10/01/2019 07:00:00','MM/DD/YYYY
HH24:MI:SS'));
INSERT INTO FLIGHT(FLIGHT_ID, NODE_ID, STATUS, SCHEDULE) VALUES ('A123', 'MIA', 'ARR',
TO_DATE('10/01/2019 07:00:00','MM/DD/YYYY
HH24:MI:SS'));
INSERT INTO FLIGHT(FLIGHT_ID, NODE_ID, STATUS, SCHEDULE) VALUES ('A123', 'MIA', 'DEP',
TO_DATE('10/02/2019 08:00:00','MM/DD/YYYY
HH24:MI:SS'));
INSERT INTO FLIGHT(FLIGHT_ID, NODE_ID, STATUS, SCHEDULE) VALUES ('A123', 'LAX', 'ARR',
TO_DATE('10/01/2019 12:00:00','MM/DD/YYYY
HH24:MI:SS'));
INSERT INTO FLIGHT(FLIGHT_ID, NODE_ID, STATUS, SCHEDULE) VALUES ('A234', 'SEA', 'DEP',
TO_DATE('10/01/2019 11:00:00','MM/DD/YYYY
HH24:MI:SS'));
INSERT INTO FLIGHT(FLIGHT_ID, NODE_ID, STATUS, SCHEDULE) VALUES ('A234', 'MIA', 'ARR',
TO_DATE('10/01/2019 14:00:00','MM/DD/YYYY
HH24:MI:SS'));

INSERT INTO FLIGHT_SIZE(FLIGHT_ID, FLIGHT_TYPE, CAPACITY) values('A123', 767, 10000);
INSERT INTO FLIGHT_SIZE(FLIGHT_ID, FLIGHT_TYPE, CAPACITY) values('A234', 737, 7000);
```

Students' answers may differ if the same result achieved.

Q1:

```
SELECT TO_CHAR(SCHEDULE,'MM/DD/YYYY') AS DATES, COUNT(DISTINCT(FLIGHT_ID)) AS
FLIGHT_COUNT FROM FLIGHT
GROUP BY TO_CHAR(SCHEDULE,'MM/DD/YYYY')
ORDER BY 1;
```

Q2:

```
SELECT DATES,SUM(DISTINCT CAPACITY/1000) || 'K' as TOTAL_CAPACITY
FROM (SELECT TO_CHAR(a.SCHEDULE,'MM/DD/YYYY') AS DATES , b.CAPACITY FROM FLIGHT
a ,FLIGHT_SIZE b WHERE
a.FLIGHT_ID=b.FLIGHT_ID )
GROUP BY DATES
ORDER BY TOTAL_CAPACITY DESC;
```

OR:

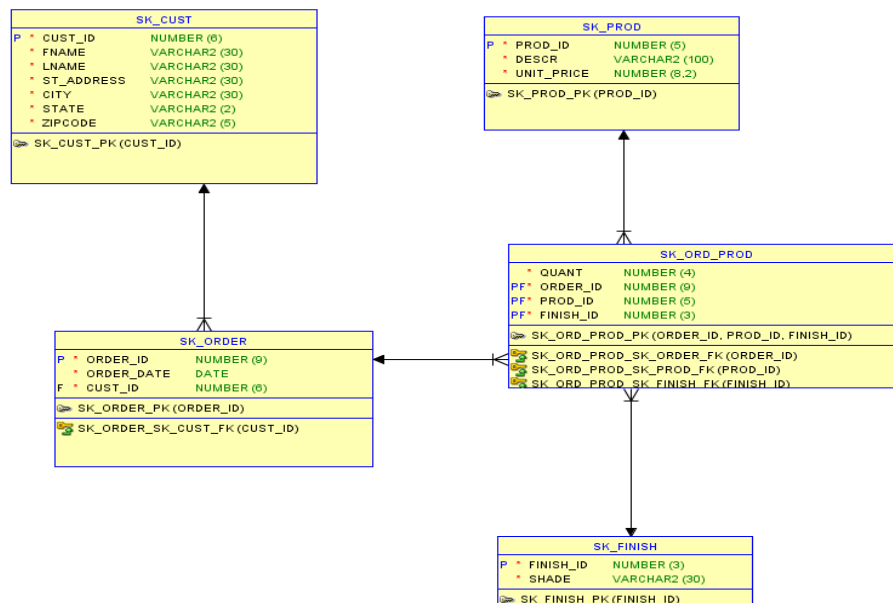
```
SELECT TO_CHAR(TRUNC(f.SCHEDULE),'MM/DD/YYYY') AS DATES, (SUM(DISTINCT s.CAPACITY) / 1e3)
|| 'K' AS TOTAL_CAPACITY
FROM FLIGHT f, FLIGHT_SIZE s WHERE f.FLIGHT_ID = s.FLIGHT_ID
GROUP BY TRUNC(SCHEDULE)
```

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.



- 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.


```

create or replace view sk_custorder_v
AS
SELECT cu.cust_id          "Customer Id",
       cu.fname            "First Name",
       cu.lname            "Last Name",
       o.order_id          "Order Id",
       o.order_date        "Order Date",
       pd.prod_id          "Product Id",
       pd.descr            "Product Description",
       fn.finish_id        "Finish Id",
       fn.shade            "Finish",
       pd.unit_price        "Unit Price",
       orpr.quant          "Order Quantity",
       orpr.quant * pd.unit_price "Total Price"
FROM sk_cust cu,
     sk_order o,
     sk_ord_prod orpr,
     sk_prod pd,
     sk_finish fn,
     ( SELECT op.order_id orid, SUM (op.quant * pr.unit_price) toa
       FROM sk_ord_prod op JOIN sk_prod pr ON op.prod_id = pr.prod_id
       GROUP BY op.order_id
       HAVING SUM (op.quant * pr.unit_price) > 1000) ord
WHERE  cu.cust_id = o.cust_id
      AND o.order_id = orpr.order_id
      AND orpr.prod_id = pd.prod_id
      AND orpr.finish_id = fn.finish_id
      AND o.order_id = ord.orid
ORDER BY 12
WITH READ ONLY;

select * from sk_custorder_v;

```

Customer Id	First Name	Last Name	Order Id	Order Date	Product Id	Product Description	Finish Id	Finish	Unit Price	Order Quantity	Total Price
45678	John	Miller	1022	10-JUN-18	9	Washstand	444	Gunstock	230	1	230
10987	Karen	Iglesias	1006	24-OCT-15	5	Writer Desk	111	Driftwood	325	2	650
12345	James	Potter	1007	23-NOV-20	4	Entertainment Center	444	Gunstock	650	1	650
10987	Karen	Iglesias	1006	24-OCT-15	4	Entertainment Center	202	Cherry	650	1	650
98211	Joseph	Lopez	1014	05-DEC-18	1	Cradle Bed	111	Driftwood	700	1	700
10987	Karen	Iglesias	1006	24-OCT-15	7	Dinning Table	101	Natural Ash	800	1	800
98211	Joseph	Lopez	1014	05-DEC-18	10	Bookcase	333	Red Chestnut	1000	1	1000
12345	James	Potter	1007	23-NOV-20	11	4-Dr Dresser	404	Oak	500	2	1000
98987	Richard	Martin	1015	20-FEB-17	2	Cabinets	101	Natural Ash	525	2	1050
45678	John	Miller	1022	10-JUN-18	4	Entertainment Center	333	Red Chestnut	650	2	1300
67890	Mary	Brown	1020	09-JUL-20	3	Couch	555	Provincial	670	2	1340
10987	Karen	Iglesias	1009	12-APR-21	13	Armoire	707	Fruitwood	1500	1	1500
99876	Linda	Wilson	1024	05-MAY-18	14	Windsor Chair	777	Natural	890	2	1780
99876	Linda	Wilson	1012	18-SEP-18	8	Wardrobe	808	Classic Gray	900	2	1800
90876	Robert	Jones	1010	24-FEB-20	10	Bookcase	202	Cherry	1000	2	2000
45678	John	Miller	1022	10-JUN-18	10	Bookcase	202	Cherry	1000	3	3000
22334	William	Martinez	1023	03-JUL-17	13	Armoire	505	Simply White	1500	2	3000

Download CSV
17 rows selected.

- 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

```
WITH prodrank
AS
( SELECT op.prod_id          prid,
      op.finish_id          fid,
      SUM (op.quant)        qt,
      RANK () OVER (ORDER BY SUM (op.quant) DESC)  qrank
  FROM sk_ord_prod op
  JOIN sk_order ord ON op.order_id = ord.order_id
  JOIN sk_prod pr
    ON   op.prod_id = pr.prod_id
      AND ord.order_date > ADD_MONTHS (SYSDATE, -6)
  GROUP BY op.prod_id, op.finish_id)
SELECT prd.prod_id  "Product Id",
      prd.descr    "Description",
      fin.shade     "Finish",
      prk.qt       "Total Quantity Sold"
  FROM sk_ord_prod orp,
      sk_prod   prd,
      sk_finish fin,
      prodrank  prk
 WHERE   orp.prod_id = prd.prod_id
        AND orp.finish_id = fin.finish_id
        AND orp.prod_id = prk.prid
        AND orp.finish_id = prk.fid
        AND prk.qrank < 4
 ORDER BY prk.qrank ASC;
```

Product Id	Description	Finish	Total Quantity Sold
5	Writer Desk	Natural	1

Download CSV

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

```
SELECT p.prod_id      "Product_Id",
       p.descr        "Description",
       p.unit_price    "Unit_Price",
       f.finish_id     "Finish_Id",
       f.shade         "Shade",
       op.quant        "Quantity_Sold",
       o.order_id      "Order_Id",
       o.order_date    "Sold_Date"
FROM SK_ORD_PROD op
FULL OUTER JOIN SK_PROD p ON p.prod_id = op.prod_id
FULL OUTER JOIN SK_FINISH f ON f.finish_id = op.finish_id
LEFT OUTER JOIN SK_ORDER o ON o.order_id = op.order_id
MINUS
SELECT p.prod_id      "Product_Id",
       p.descr        "Description",
       p.unit_price    "Unit_Price",
       f.finish_id     "Finish_Id",
       f.shade         "Shade",
       op.quant        "Quantity_Sold",
       o.order_id      "Order_Id",
       o.order_date    "Sold_Date"
FROM SK_ORD_PROD op,
     SK_PROD      p,
     SK_FINISH    f,
     SK_ORDER      o
WHERE  p.prod_id = op.prod_id
AND f.finish_id = op.finish_id
AND o.order_id = op.order_id
AND EXTRACT (MONTH FROM o.order_date) in ('10','11','12')
AND EXTRACT (YEAR FROM o.order_date)='2020';
```

Product_Id	Description	Unit_Price	Finish_Id	Shade	Quantity_Sold	Order_Id	Sold_Date
1	Cradle Bed	700	111	Driftwood	1	1014	05-DEC-18
1	Cradle Bed	700	666	Puritan Prine	1	1019	10-JAN-21
2	Cabinets	525	101	Natural Ash	2	1015	20-FEB-17
3	Couch	670	555	Provincial	2	1020	09-JUL-20
4	Entertainment Center	650	202	Cherry	1	1006	24-OCT-15
4	Entertainment Center	650	333	Red Chestnut	2	1022	10-JUN-18
4	Entertainment Center	650	909	Green Velvet	1	1017	01-MAY-20
5	Writer Desk	325	111	Driftwood	2	1006	24-OCT-15
5	Writer Desk	325	777	Natural	1	1018	03-SEP-21
6	Settee	500	707	Fruitwood	1	1013	11-NOV-17
7	Dinning Table	800	101	Natural Ash	1	1006	24-OCT-15
8	Wardrobe	900	808	Classic Gray	2	1012	18-SEP-18
8	Wardrobe	900	909	Green Velvet	1	1030	11-MAR-21
9	Washstand	230	444	Gunstock	1	1022	10-JUN-18
9	Washstand	230	505	Simply White	3	1016	06-MAR-19
9	Washstand	230	909	Green Velvet	1	1011	10-MAY-19
10	Bookcase	1000	202	Cherry	2	1010	24-FEB-20
10	Bookcase	1000	202	Cherry	3	1022	10-JUN-18
10	Bookcase	1000	333	Red Chestnut	1	1014	05-DEC-18
12	Rolltop Desk	2050	-	-	-	-	-
13	Armoire	1500	505	Simply White	2	1023	03-JUL-17
13	Armoire	1500	707	Fruitwood	1	1009	12-APR-21
14	Windsor Chair	890	777	Natural	2	1024	05-MAY-18
15	Nightstand	300	666	Puritan Prine	1	1008	01-JAN-20
15	Nightstand	300	707	Fruitwood	2	1025	06-FEB-19
-	-	-	222	Golden Pecan	-	-	-
-	-	-	303	Natural Maple	-	-	-

The result may have only product name and finish columns only. Also, the answer is still correct if used with CARTESIAN JOIN (CROSS JOIN) instead of OUTER joins, in that situation the result set may be quite large

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