CS-GY 6083 - B, Fall 2023
Principles of Database Systems

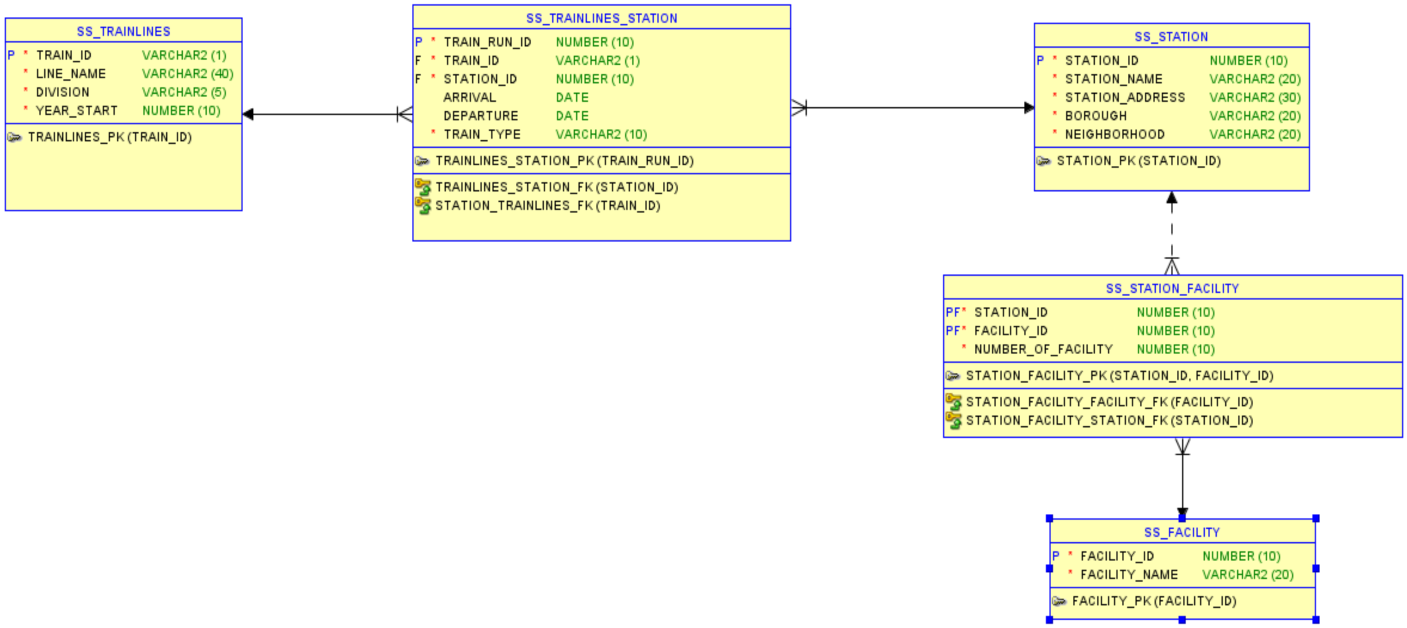
ASSIGNMENT: 3 [100 Points, 5% weight to the Final grades]

Please submit your assignment on NYU Brightspace in a single PDF document attachment. Please mention Student ID, Name, Course, Section Number, and date of submission on first page of your submission. Please use SQL file (DDL and DML code) for writing SQLs in Oracle. You may create equivalent DDL and DML code for MySQL and write SQLs in MySQL. All the tables should be created with your initial as a prefix. DO NOT change data in tables. You can write SQL in either Oracle or MySQL.

Problem A: 50 points [each question has 10 points]

The New York City Subway is a rapid transit system in the New York City boroughs of Manhattan, Brooklyn, Queens, and the Bronx. It is owned by the government of New York City and leased to the New York City Transit Authority, an affiliate agency of the state-run Metropolitan Transportation Authority (MTA). Opened on October 27, 1904, the New York City Subway is one of the world's oldest public transit systems, one of the most used, and the one with the most stations, with 472 stations in operation and 36 train lines.

SAMTA (Subway Administration of Metropolitan Transportation Authority) is the NYU affiliated data analytics startup company. SAMTA has undertaken a database project to normalize subway data systems. For relational schema model in figure, write SQL statements to answer following questions. You don’t need to create a relational model.

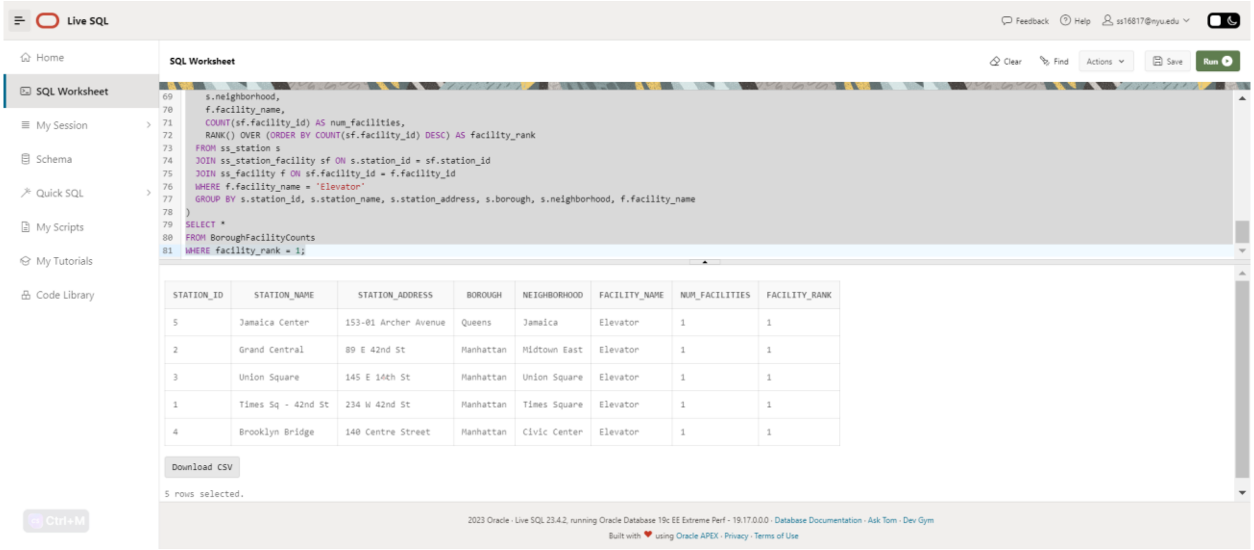

SS_TRAINLINES
P * TRAIN_ID VARCHAR2 (1)
* LINE_NAME VARCHAR2 (40)
* DIVISION VARCHAR2 (5)
* YEAR_START NUMBER (10)
TRAINLINES_PK (TRAIN_ID)
SS_TRAINLINES_STATION
P * TRAIN_RUN_ID NUMBER (10)
F * TRAIN_ID VARCHAR2 (1)
F * STATION_ID NUMBER (10)
ARRIVAL DATE
DEPARTURE DATE
* TRAIN_TYPE VARCHAR2 (10)
TRAINLINES_STATION_PK (TRAIN_RUN_ID)
TRAINLINES_STATION_FK (STATION_ID)
STATION_TRAINLINES_FK (TRAIN_ID)
SS_STATION
P * STATION_ID NUMBER (10)
* STATION_NAME VARCHAR2 (20)
* STATION_ADDRESS VARCHAR2 (30)
* BOROUGH VARCHAR2 (20)
* NEIGHBORHOOD VARCHAR2 (20)
STATION_PK (STATION_ID)
SS_STATION_FACILITY
PF* STATION_ID NUMBER (10)
PF* FACILITY_ID NUMBER (10)
* NUMBER_OF_FACILITY NUMBER (10)
STATION_FACILITY_PK (STATION_ID, FACILITY_ID)
STATION_FACILITY_FACILITY_FK (FACILITY_ID)
STATION_FACILITY_STATION_FK (STATION_ID)
SS_FACILITY
P * FACILITY_ID NUMBER (10)
* FACILITY_NAME VARCHAR2 (20)
FACILITY_PK (FACILITY_ID)

i. **List all the details of the station which has the highest number of facilities in Elevator.**

SQL QUERY -

```
WITH BoroughFacilityCounts AS (
 SELECT
 s.station_id, s.station_name, s.station_address, s.borough, s.neighborhood,  -
f.facility_name,
   COUNT(sf.facility_id) AS num_facilities,
   RANK() OVER (ORDER BY COUNT(sf.facility_id) DESC) AS facility_rank
  FROM ss_station s
  JOIN ss_station_facility sf ON s.station_id = sf.station_id
  JOIN ss_facility f ON sf.facility_id = f.facility_id
  WHERE f.facility_name = 'Elevator'
  GROUP BY s.station_id, s.station_name, s.station_address, s.borough,
s.neighborhood, f.facility_name
)
SELECT *
FROM BoroughFacilityCounts
WHERE facility_rank = 1;
```

RESULT -



SQL Worksheet

```
      s.neighborhood,
      f.facility_name,
      COUNT(sf.facility_id) AS num_facilities,
      RANK() OVER (ORDER BY COUNT(sf.facility_id) DESC) AS facility_rank
    FROM ss_station s
    JOIN ss_station_facility sf ON s.station_id = sf.station_id
    JOIN ss_facility f ON sf.facility_id = f.facility_id
    WHERE f.facility_name = 'Elevator'
    GROUP BY s.station_id, s.station_name, s.station_address, s.borough, s.neighborhood, f.facility_name
  )
  SELECT *
  FROM BoroughFacilityCounts
  WHERE facility_rank = 1;
```

STATION_ID	STATION_NAME	STATION_ADDRESS	BOROUGH	NEIGHBORHOOD	FACILITY_NAME	NUM_FACILITIES	FACILITY_RANK
5	Jamaica Center	153-01 Archer Avenue	Queens	Jamaica	Elevator	1	1
2	Grand Central	89 E 42nd St	Manhattan	Midtown East	Elevator	1	1
3	Union Square	145 E 14th St	Manhattan	Union Square	Elevator	1	1
1	Times Sq - 42nd St	234 W 42nd St	Manhattan	Times Square	Elevator	1	1
4	Brooklyn Bridge	140 Centre Street	Manhattan	Civic Center	Elevator	1	1

5 rows selected.

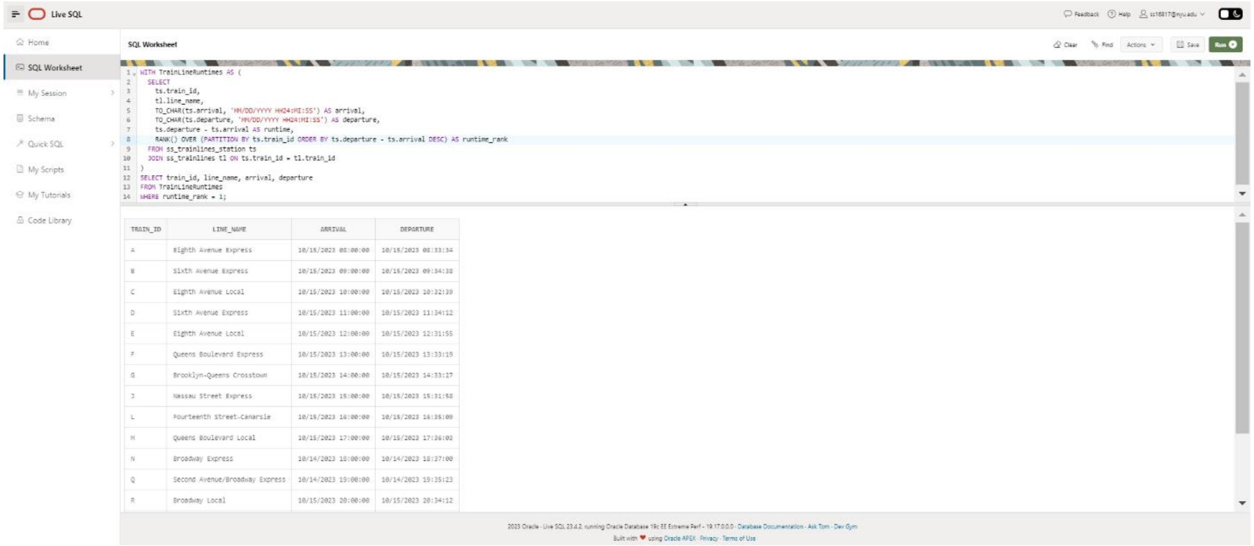
Ctrl+M
2023 Oracle · Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

ii. **List details of each train line and their highest run time. Your result should have TRAIN_ID, LINE_NAME, ARRIVVAL, DEPARTURE,**

SQL QUERY –

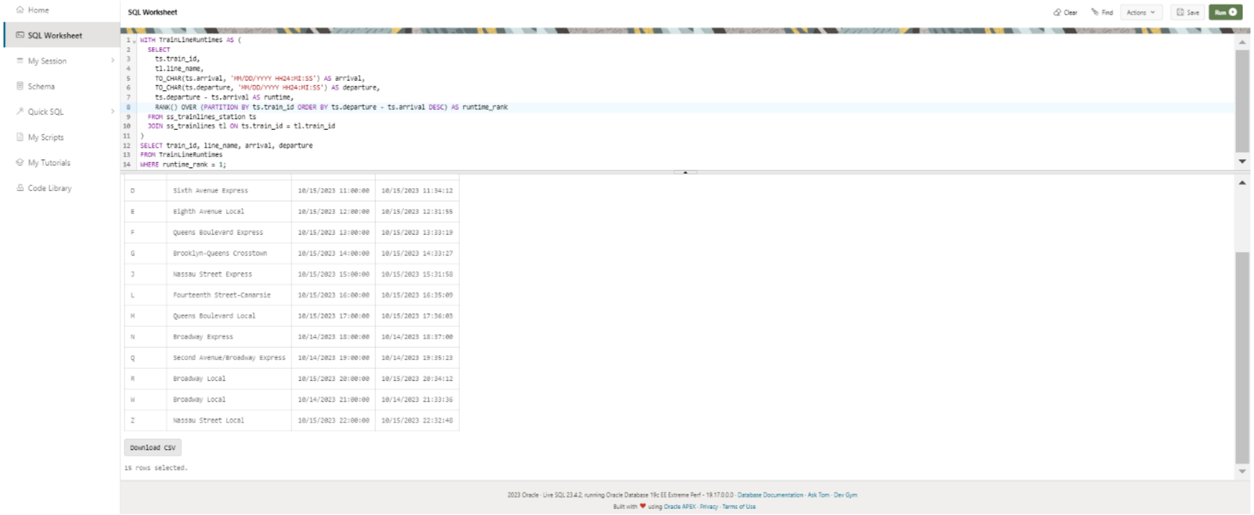
```
WITH TrainLineRuntimes AS (
  SELECT
    ts.train_id,
    tl.line_name,
    TO_CHAR(ts.arrival, 'MM/DD/YYYY HH24:MI:SS') AS arrival,
    TO_CHAR(ts.departure, 'MM/DD/YYYY HH24:MI:SS') AS departure,
    ts.departure - ts.arrival AS runtime,
    RANK() OVER (PARTITION BY ts.train_id ORDER BY ts.departure - ts.arrival DESC) AS runtime_rank
  FROM ss_trainlines_station ts
  JOIN ss_trainlines tl ON ts.train_id = tl.train_id
)
SELECT train_id, line_name, arrival, departure
FROM TrainLineRuntimes
WHERE runtime_rank = 1;
```

RESULT -


Live SQL
Feedback
Help
ss16817@nyu.edu
Home
SQL Worksheet
My Session
Schema
Quick SQL
My Scripts
My Tutorials
Code Library
SQL Worksheet
Clear
Find
Actions
Save
Run
2023 Oracle · Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use

```sql
WITH TrainLineRuntimes AS (
  SELECT
    ts.train_id,
    tl.line_name,
    TO_CHAR(ts.arrival, 'MM/DD/YYYY HH24:MI:SS') AS arrival,
    TO_CHAR(ts.departure, 'MM/DD/YYYY HH24:MI:SS') AS departure,
    ts.departure - ts.arrival AS runtime,
    RANK() OVER (PARTITION BY ts.train_id ORDER BY ts.departure - ts.arrival DESC) AS runtime_rank
  FROM ss_trainlines_station ts
  JOIN ss_trainlines tl ON ts.train_id = tl.train_id
)
SELECT train_id, line_name, arrival, departure
FROM TrainLineRuntimes
WHERE runtime_rank = 1;
```

TRAIN_ID	LINE_NAME	ARRIVAL	DEPARTURE
A	Eighth Avenue Express	10/15/2023 08:00:00	10/15/2023 08:33:34
B	Sixth Avenue Express	10/15/2023 09:00:00	10/15/2023 09:34:38
C	Eighth Avenue Local	10/15/2023 10:00:00	10/15/2023 10:32:39
D	Sixth Avenue Express	10/15/2023 11:00:00	10/15/2023 11:34:12
E	Eighth Avenue Local	10/15/2023 12:00:00	10/15/2023 12:31:55
F	Queens Boulevard Express	10/15/2023 13:00:00	10/15/2023 13:33:19
G	Brooklyn-Queens Crosstown	10/15/2023 14:00:00	10/15/2023 14:33:27
J	Nassau Street Express	10/15/2023 15:00:00	10/15/2023 15:31:58
L	Fourteenth Street-Canarsie	10/15/2023 16:00:00	10/15/2023 16:35:09
M	Queens Boulevard Local	10/15/2023 17:00:00	10/15/2023 17:36:03
N	Broadway Express	10/14/2023 18:00:00	10/14/2023 18:37:00
Q	Second Avenue/Broadway Express	10/14/2023 19:00:00	10/14/2023 19:35:23
R	Broadway Local	10/15/2023 20:00:00	10/15/2023 20:34:12


Home
SQL Worksheet
My Session
Schema
Quick SQL
My Scripts
My Tutorials
Code Library
SQL Worksheet
Clear
Find
Actions
Save
Run
Download CSV
2023 Oracle · Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use

```sql
WITH TrainLineRuntimes AS (
  SELECT
    ts.train_id,
    tl.line_name,
    TO_CHAR(ts.arrival, 'MM/DD/YYYY HH24:MI:SS') AS arrival,
    TO_CHAR(ts.departure, 'MM/DD/YYYY HH24:MI:SS') AS departure,
    ts.departure - ts.arrival AS runtime,
    RANK() OVER (PARTITION BY ts.train_id ORDER BY ts.departure - ts.arrival DESC) AS runtime_rank
  FROM ss_trainlines_station ts
  JOIN ss_trainlines tl ON ts.train_id = tl.train_id
)
SELECT train_id, line_name, arrival, departure
FROM TrainLineRuntimes
WHERE runtime_rank = 1;
```

D	Sixth Avenue Express	10/15/2023 11:00:00	10/15/2023 11:34:12
E	Eighth Avenue Local	10/15/2023 12:00:00	10/15/2023 12:31:55
F	Queens Boulevard Express	10/15/2023 13:00:00	10/15/2023 13:33:19
G	Brooklyn-Queens Crosstown	10/15/2023 14:00:00	10/15/2023 14:33:27
J	Nassau Street Express	10/15/2023 15:00:00	10/15/2023 15:31:58
L	Fourteenth Street-Canarsie	10/15/2023 16:00:00	10/15/2023 16:35:09
M	Queens Boulevard Local	10/15/2023 17:00:00	10/15/2023 17:36:03
N	Broadway Express	10/14/2023 18:00:00	10/14/2023 18:37:00
Q	Second Avenue/Broadway Express	10/14/2023 19:00:00	10/14/2023 19:35:23
R	Broadway Local	10/15/2023 20:00:00	10/15/2023 20:34:12
W	Broadway Local	10/14/2023 21:00:00	10/14/2023 21:33:36
Z	Nassau Street Local	10/15/2023 22:00:00	10/15/2023 22:32:48

```
15 rows selected.
```

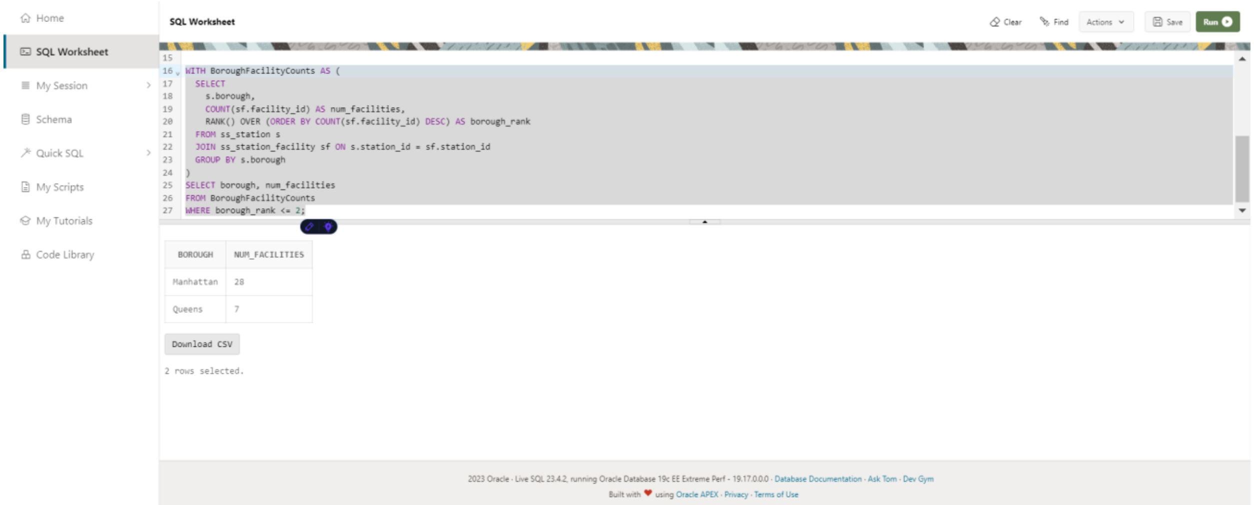
iii. **Find the top two boroughs in terms of numbers of station facilities. Your result should have the name of the borough and number of facilities.**

SQL QUERY -

```sql
WITH BoroughFacilityCounts AS (
  SELECT s.borough,
    COUNT(sf.facility_id) AS num_facilities,
    RANK() OVER (ORDER BY COUNT(sf.facility_id) DESC) AS borough_rank
  FROM ss_station s
  JOIN ss_station_facility sf ON s.station_id = sf.station_id
  GROUP BY s.borough
)
SELECT borough, num_facilities
FROM BoroughFacilityCounts
```

WHERE borough_rank <= 2;

RESULT-



```sql
WITH BoroughFacilityCounts AS (
  SELECT
    s.borough,
    COUNT(sf.facility_id) AS num_facilities,
    RANK() OVER (ORDER BY COUNT(sf.facility_id) DESC) AS borough_rank
  FROM ss_station s
  JOIN ss_station_facility sf ON s.station_id = sf.station_id
  GROUP BY s.borough
)
SELECT borough, num_facilities
FROM BoroughFacilityCounts
WHERE borough_rank <= 2;
```

BOROUGH	NUM_FACILITIES
Manhattan	28
Queens	7

2 rows selected.

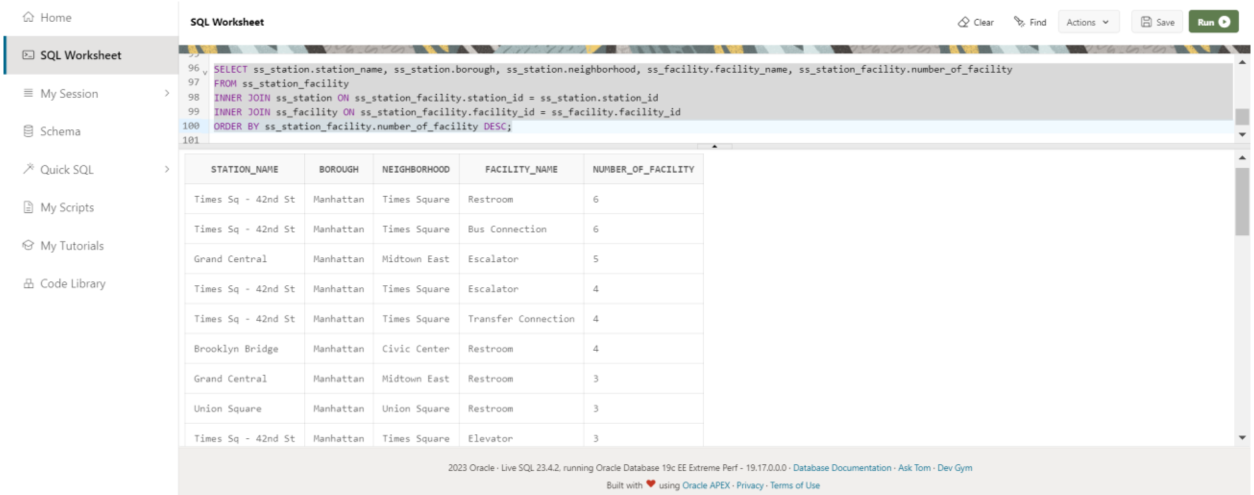
2023 Oracle · Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

iv. **List station name, borough, neighborhood, facility name, and number of facilities. Arrange the result in descending order of number of facilities.**

SQL QUERY -

SELECT ss_station.station_name, ss_station.borough, ss_station.neighborhood, ss_facility.facility_name, ss_station_facility.number_of_facility
FROM ss_station_facility
INNER JOIN ss_station ON ss_station_facility.station_id = ss_station.station_id
INNER JOIN ss_facility ON ss_station_facility.facility_id = ss_facility.facility_id
ORDER BY ss_station_facility.number_of_facility DESC;

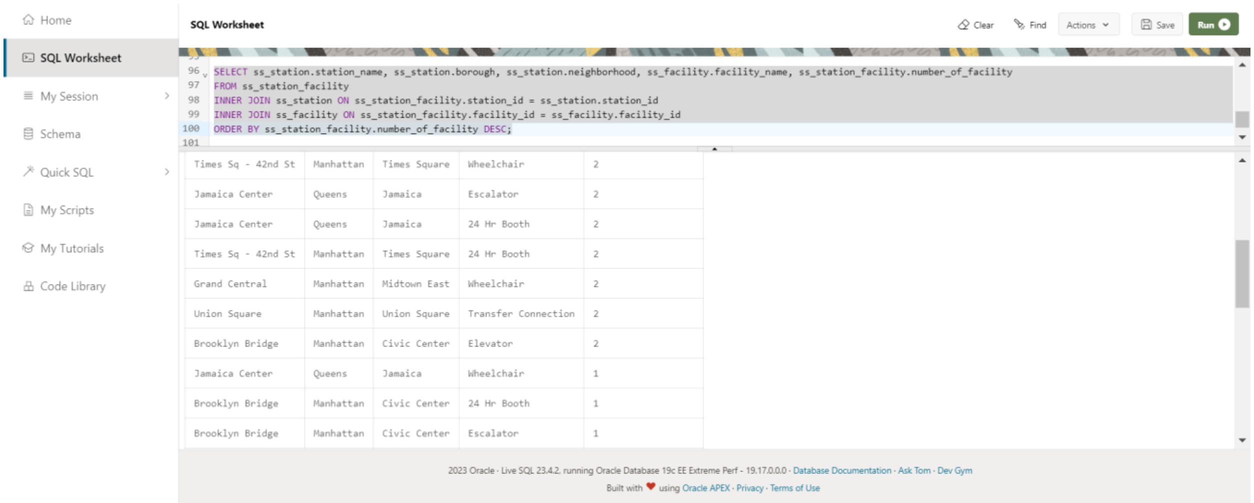
RESULT-



```sql
SELECT ss_station.station_name, ss_station.borough, ss_station.neighborhood, ss_facility.facility_name, ss_station_facility.number_of_facility
FROM ss_station_facility
INNER JOIN ss_station ON ss_station_facility.station_id = ss_station.station_id
INNER JOIN ss_facility ON ss_station_facility.facility_id = ss_facility.facility_id
ORDER BY ss_station_facility.number_of_facility DESC;
```

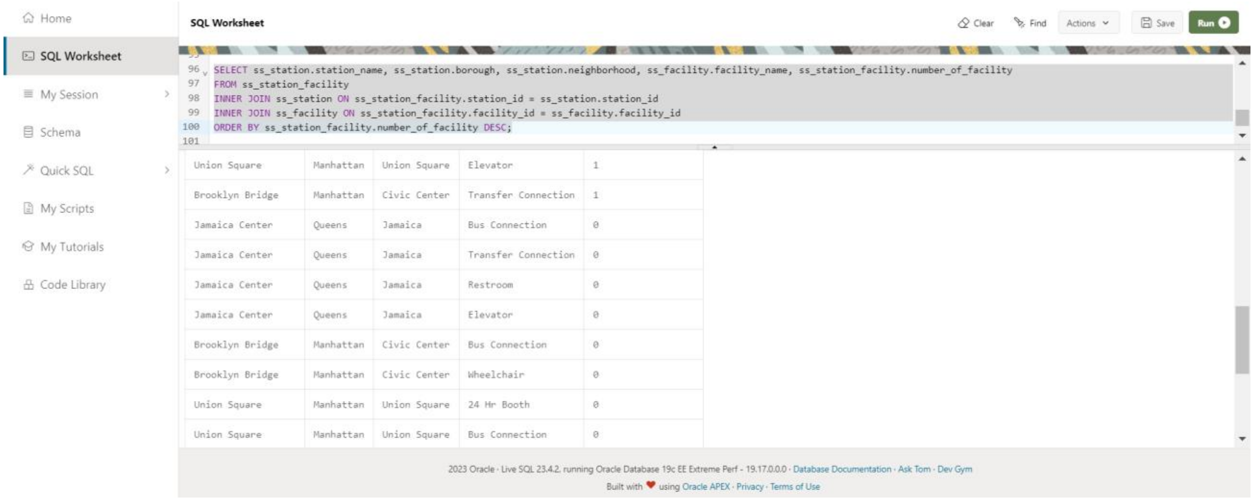
STATION_NAME	BOROUGH	NEIGHBORHOOD	FACILITY_NAME	NUMBER_OF_FACILITY
Times Sq - 42nd St	Manhattan	Times Square	Restroom	6
Times Sq - 42nd St	Manhattan	Times Square	Bus Connection	6
Grand Central	Manhattan	Midtown East	Escalator	5
Times Sq - 42nd St	Manhattan	Times Square	Escalator	4
Times Sq - 42nd St	Manhattan	Times Square	Transfer Connection	4
Brooklyn Bridge	Manhattan	Civic Center	Restroom	4
Grand Central	Manhattan	Midtown East	Restroom	3
Union Square	Manhattan	Union Square	Restroom	3
Times Sq - 42nd St	Manhattan	Times Square	Elevator	3

2023 Oracle · Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use


Home
SQL Worksheet
My Session
Schema
Quick SQL
My Scripts
My Tutorials
Code Library
SQL Worksheet
Clear
Find
Actions
Save
Run
2023 Oracle · Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

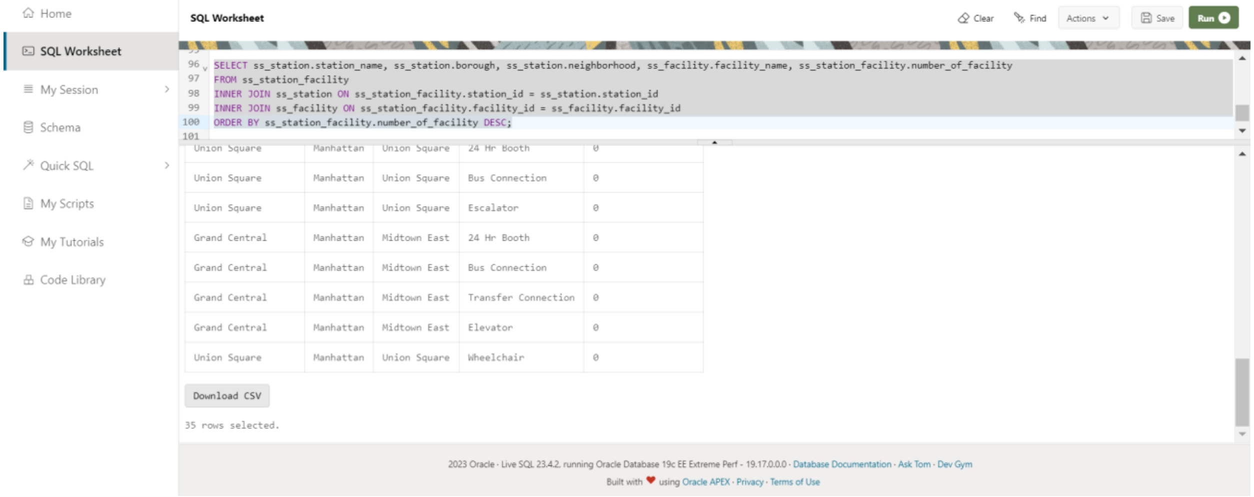
```sql
SELECT ss_station.station_name, ss_station.borough, ss_station.neighborhood, ss_facility.facility_name, ss_station_facility.number_of_facility
FROM ss_station_facility
INNER JOIN ss_station ON ss_station_facility.station_id = ss_station.station_id
INNER JOIN ss_facility ON ss_station_facility.facility_id = ss_facility.facility_id
ORDER BY ss_station_facility.number_of_facility DESC;
```

Times Sq - 42nd St	Manhattan	Times Square	Wheelchair	2
Jamaica Center	Queens	Jamaica	Escalator	2
Jamaica Center	Queens	Jamaica	24 Hr Booth	2
Times Sq - 42nd St	Manhattan	Times Square	24 Hr Booth	2
Grand Central	Manhattan	Midtown East	Wheelchair	2
Union Square	Manhattan	Union Square	Transfer Connection	2
Brooklyn Bridge	Manhattan	Civic Center	Elevator	2
Jamaica Center	Queens	Jamaica	Wheelchair	1
Brooklyn Bridge	Manhattan	Civic Center	24 Hr Booth	1
Brooklyn Bridge	Manhattan	Civic Center	Escalator	1


Home
SQL Worksheet
My Session
Schema
Quick SQL
My Scripts
My Tutorials
Code Library
SQL Worksheet
Clear
Find
Actions
Save
Run
2023 Oracle · Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

```sql
SELECT ss_station.station_name, ss_station.borough, ss_station.neighborhood, ss_facility.facility_name, ss_station_facility.number_of_facility
FROM ss_station_facility
INNER JOIN ss_station ON ss_station_facility.station_id = ss_station.station_id
INNER JOIN ss_facility ON ss_station_facility.facility_id = ss_facility.facility_id
ORDER BY ss_station_facility.number_of_facility DESC;
```

Union Square	Manhattan	Union Square	Elevator	1
Brooklyn Bridge	Manhattan	Civic Center	Transfer Connection	1
Jamaica Center	Queens	Jamaica	Bus Connection	0
Jamaica Center	Queens	Jamaica	Transfer Connection	0
Jamaica Center	Queens	Jamaica	Restroom	0
Jamaica Center	Queens	Jamaica	Elevator	0
Brooklyn Bridge	Manhattan	Civic Center	Bus Connection	0
Brooklyn Bridge	Manhattan	Civic Center	Wheelchair	0
Union Square	Manhattan	Union Square	24 Hr Booth	0
Union Square	Manhattan	Union Square	Bus Connection	0


Home
SQL Worksheet
My Session
Schema
Quick SQL
My Scripts
My Tutorials
Code Library
SQL Worksheet
Clear
Find
Actions
Save
Run
Download CSV
2023 Oracle · Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

```sql
SELECT ss_station.station_name, ss_station.borough, ss_station.neighborhood, ss_facility.facility_name, ss_station_facility.number_of_facility
FROM ss_station_facility
INNER JOIN ss_station ON ss_station_facility.station_id = ss_station.station_id
INNER JOIN ss_facility ON ss_station_facility.facility_id = ss_facility.facility_id
ORDER BY ss_station_facility.number_of_facility DESC;
```

Union Square	Manhattan	Union Square	24 Hr Booth	0
Union Square	Manhattan	Union Square	Bus Connection	0
Union Square	Manhattan	Union Square	Escalator	0
Grand Central	Manhattan	Midtown East	24 Hr Booth	0
Grand Central	Manhattan	Midtown East	Bus Connection	0
Grand Central	Manhattan	Midtown East	Transfer Connection	0
Grand Central	Manhattan	Midtown East	Elevator	0
Union Square	Manhattan	Union Square	Wheelchair	0

```
35 rows selected.
```

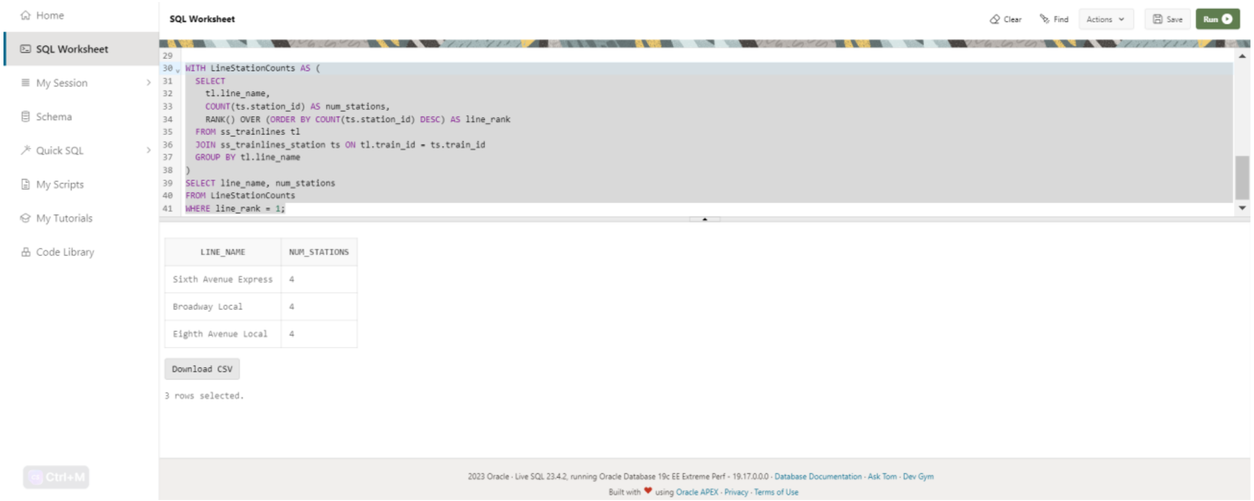
v. **List the train line that has maximum number of stations.**

SQL QUERY -

WITH LineStationCounts AS (

```
  SELECT
    tl.line_name,
    COUNT(ts.station_id) AS num_stations,
    RANK() OVER (ORDER BY COUNT(ts.station_id) DESC) AS line_rank
  FROM ss_trainlines tl
  JOIN ss_trainlines_station ts ON tl.train_id = ts.train_id
  GROUP BY tl.line_name
)
SELECT line_name, num_stations
FROM LineStationCounts
WHERE line_rank = 1;
```

RESULT-



```
WITH LineStationCounts AS (
  SELECT
    tl.line_name,
    COUNT(ts.station_id) AS num_stations,
    RANK() OVER (ORDER BY COUNT(ts.station_id) DESC) AS line_rank
  FROM ss_trainlines tl
  JOIN ss_trainlines_station ts ON tl.train_id = ts.train_id
  GROUP BY tl.line_name
)
SELECT line_name, num_stations
FROM LineStationCounts
WHERE line_rank = 1;
```

LINE_NAME	NUM_STATIONS
Sixth Avenue Express	4
Broadway Local	4
Eighth Avenue Local	4

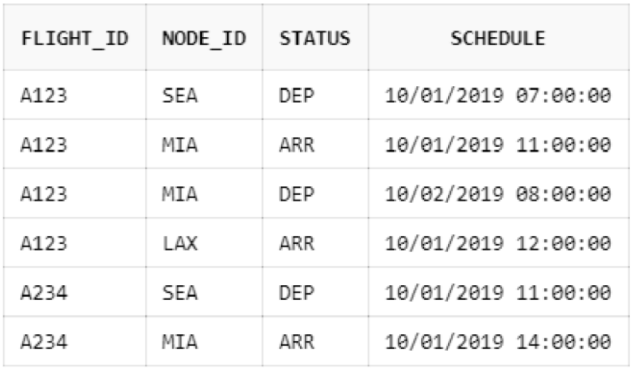
Download CSV

3 rows selected.

2023 Oracle · Live SQL 23.4.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use

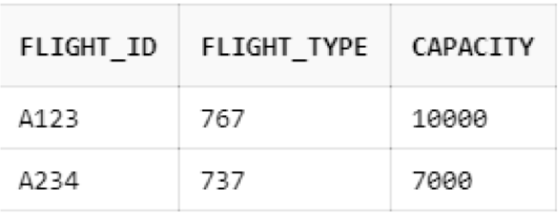
Submit: For each of questions submit a) SQL b) SQL result. All SQL and corresponding results must be visible clearly on screenshots.

Problem 2: 20 points [each question has 10 points]



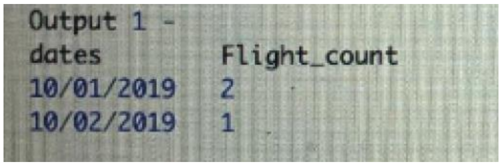
FLIGHT_ID	NODE_ID	STATUS	SCHEDULE
A123	SEA	DEP	10/01/2019 07:00:00
A123	MIA	ARR	10/01/2019 11:00:00
A123	MIA	DEP	10/02/2019 08:00:00
A123	LAX	ARR	10/01/2019 12:00:00
A234	SEA	DEP	10/01/2019 11:00:00
A234	MIA	ARR	10/01/2019 14:00:00

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



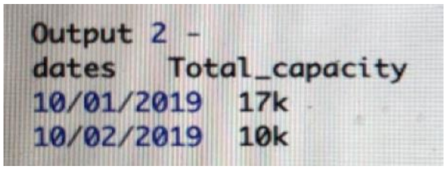
FLIGHT_ID	FLIGHT_TYPE	CAPACITY
A123	767	10000
A234	737	7000

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.

part1

SELECT TRUNC(schedule) AS "dates", COUNT(DISTINCT flight_id) AS "flight_count"

FROM flight GROUP BY TRUNC(schedule) ORDER BY TRUNC(schedule);

part 2

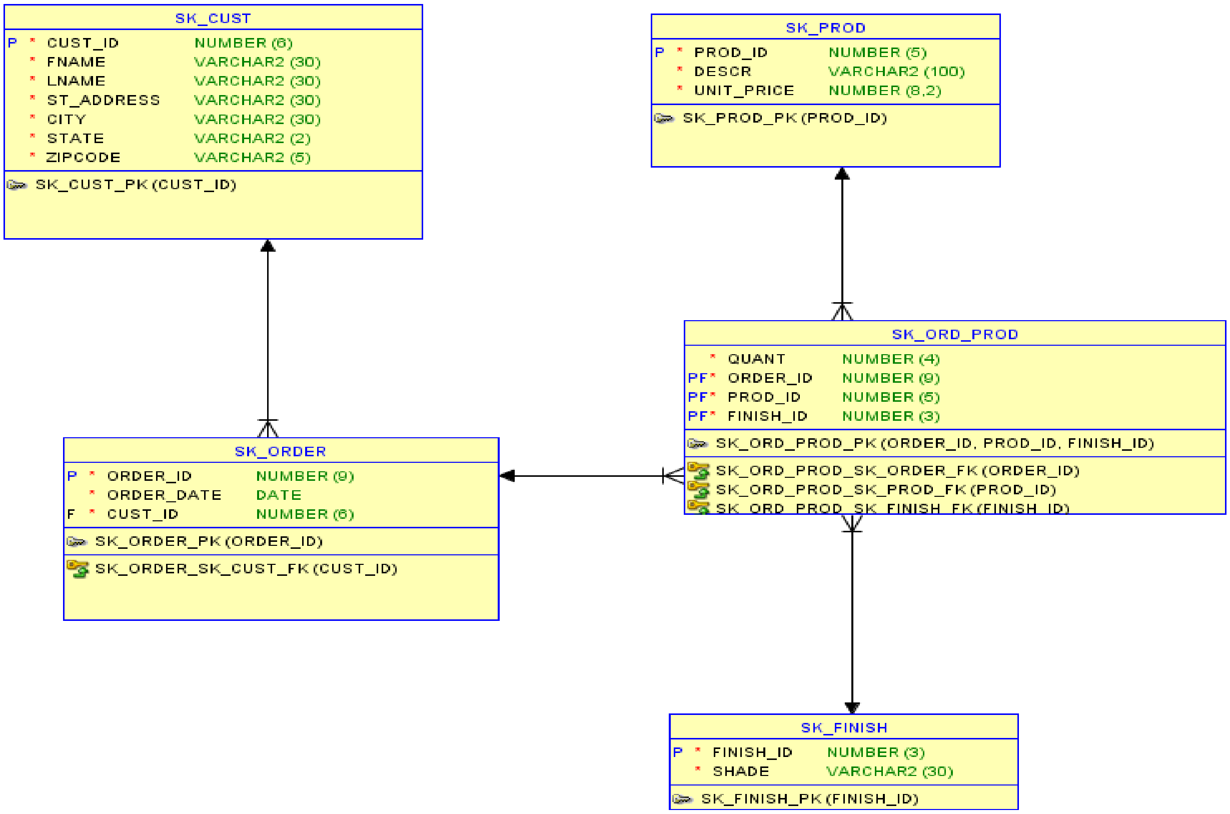
```
SELECT schedule AS "dates", SUM(capacity)/1000||'K' AS "total_capacity" FROM
 (SELECT DISTINCT(f.flight_id), TRUNC(schedule) AS schedule, capacity
FROM flight f
JOIN flight_size fs ON fs.flight_id = f.flight_id
) GROUP BY schedule ORDER BY schedule;
```

Submission:

Create tables as above with your initial as prefix such as AP_FLIGHT, AP_SIZE. Populate 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 or MySQL relational database. All SQL and corresponding results must be visible clearly on screenshots.

Problem 3: 30 points

For a given relational model below, please find attached file containing DDL and DMLs. You may create equivalent code for MySQL. Create tables and insert data by replacing SK with your own initial. You can do this assignment either in Oracle or MySQL. <u>You don't need to draw logical/relational models.</u>

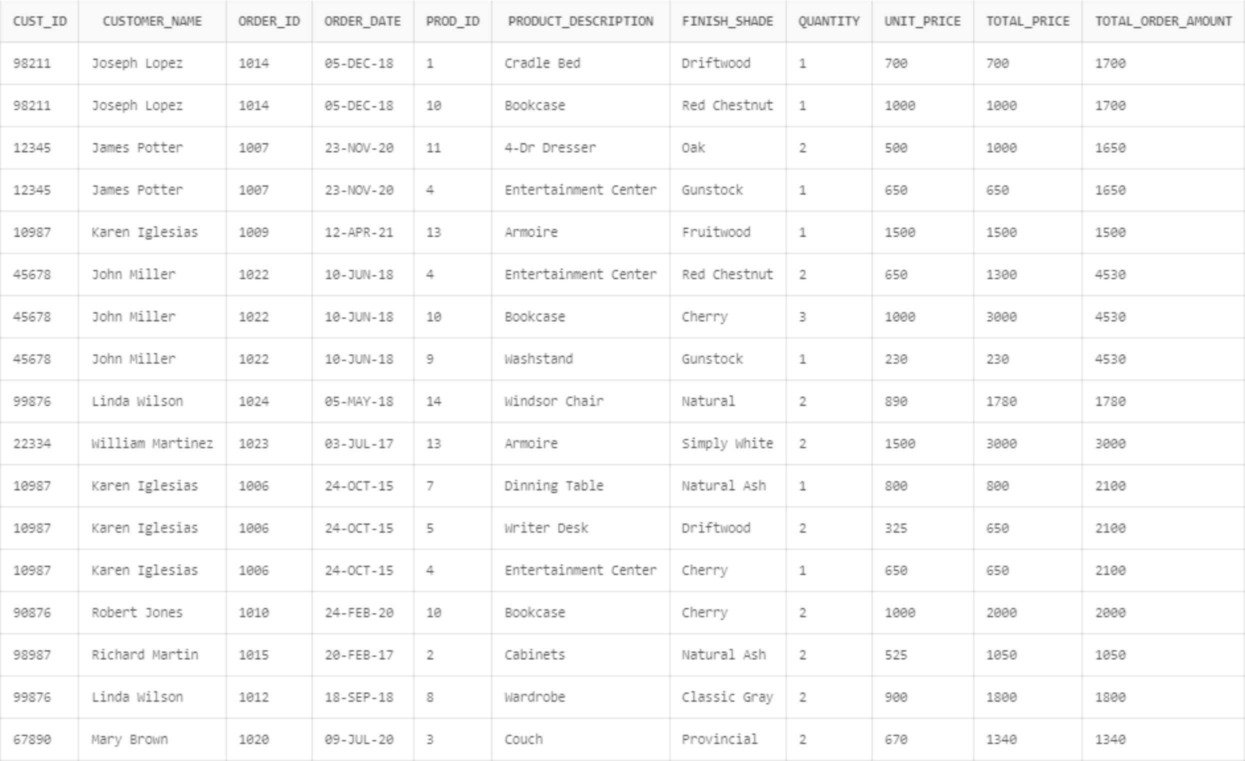


I. **For this relational model of a furniture company, create a read-only database view that represents the following dataset. Customer ID, Customer Name (both First and Last name), Order_Id, order date, each product in order with description, quantity, Unit_Price, Total price of each product, and Finish shade. Sort the dataset in order of total order amount. Give appropriate column names in view. Restrict dataset to represent only those orders which have total value over $1000. Once view is created submit View code and then retrieve result of view using SQL query.**

ANS:

```
CREATE VIEW orde_view AS
SELECT
    o.cust_id,
    c.fname || ' ' || c.lname AS customer_name,
    o.order_id,
    o.order_date,
    p.prod_id,
    p.descr AS product_description,
    f.shade AS finish_shade,
    op.quant AS quantity,
    p.unit_price,
    (op.quant * p.unit_price) AS total_price,
    order_totals.total_order_amount
FROM
    sk_order o
    JOIN sk_cust c ON o.cust_id = c.cust_id
    JOIN sk_ord_prod op ON o.order_id = op.order_id
    JOIN sk_prod p ON op.prod_id = p.prod_id
    JOIN sk_finish f ON op.finish_id = f.finish_id
    JOIN (
        SELECT
            o.order_id,
            SUM(op.quant * p.unit_price) AS total_order_amount
        FROM
            sk_order o
            JOIN sk_ord_prod op ON o.order_id = op.order_id
            JOIN sk_prod p ON op.prod_id = p.prod_id
        GROUP BY
            o.order_id
        HAVING
            SUM(op.quant * p.unit_price) > 1000
    ) order_totals ON o.order_id = order_totals.order_id;


SELECT *
FROM ORDE_VIEW;
```

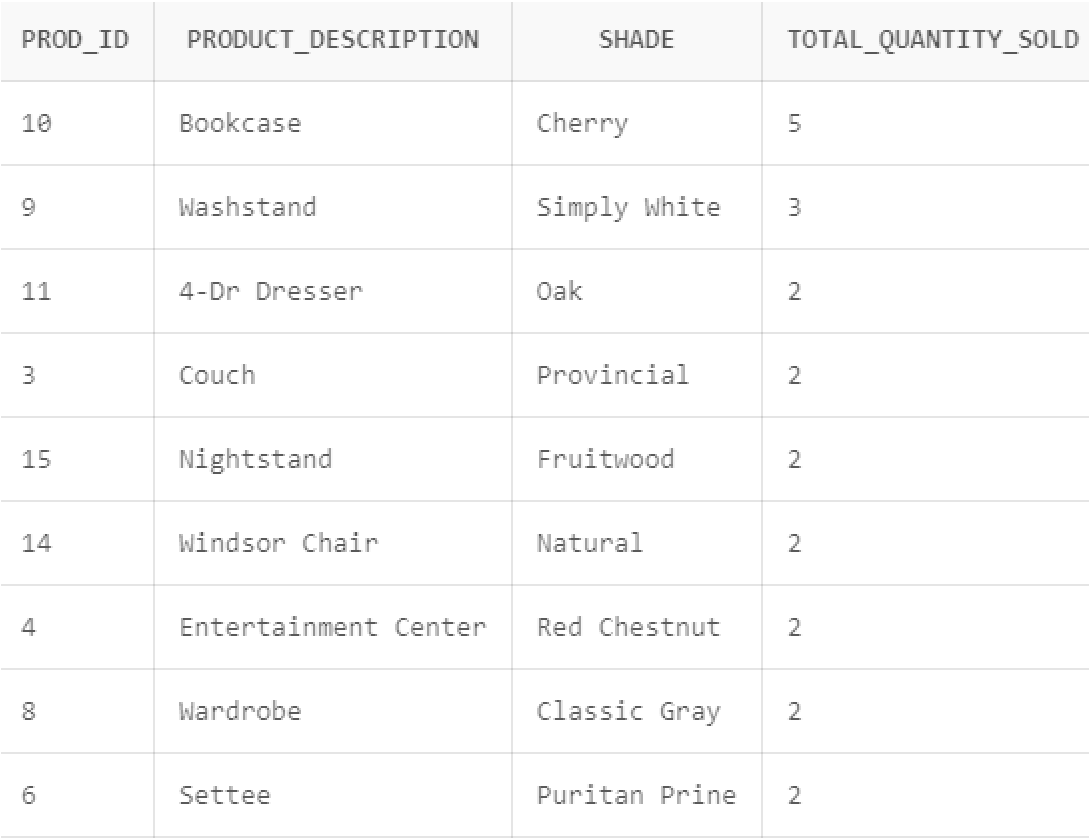


CUST_ID	CUSTOMER_NAME	ORDER_ID	ORDER_DATE	PROD_ID	PRODUCT_DESCRIPTION	FINISH_SHADE	QUANTITY	UNIT_PRICE	TOTAL_PRICE	TOTAL_ORDER_AMOUNT
98211	Joseph Lopez	1014	05-DEC-18	1	Cradle Bed	Driftwood	1	700	700	1700
98211	Joseph Lopez	1014	05-DEC-18	10	Bookcase	Red Chestnut	1	1000	1000	1700
12345	James Potter	1007	23-NOV-20	11	4-Dr Dresser	Oak	2	500	1000	1650
12345	James Potter	1007	23-NOV-20	4	Entertainment Center	Gunstock	1	650	650	1650
10987	Karen Iglesias	1009	12-APR-21	13	Armoire	Fruitwood	1	1500	1500	1500
45678	John Miller	1022	10-JUN-18	4	Entertainment Center	Red Chestnut	2	650	1300	4530
45678	John Miller	1022	10-JUN-18	10	Bookcase	Cherry	3	1000	3000	4530
45678	John Miller	1022	10-JUN-18	9	Washstand	Gunstock	1	230	230	4530
99876	Linda Wilson	1024	05-MAY-18	14	Windsor Chair	Natural	2	890	1780	1780
22334	William Martinez	1023	03-JUL-17	13	Armoire	Simply White	2	1500	3000	3000
10987	Karen Iglesias	1006	24-OCT-15	7	Dinning Table	Natural Ash	1	800	800	2100
10987	Karen Iglesias	1006	24-OCT-15	5	Writer Desk	Driftwood	2	325	650	2100
10987	Karen Iglesias	1006	24-OCT-15	4	Entertainment Center	Cherry	1	650	650	2100
90876	Robert Jones	1010	24-FEB-20	10	Bookcase	Cherry	2	1000	2000	2000
98987	Richard Martin	1015	20-FEB-17	2	Cabinets	Natural Ash	2	525	1050	1050
99876	Linda Wilson	1012	18-SEP-18	8	Wardrobe	Classic Gray	2	900	1800	1800
67890	Mary Brown	1020	09-JUL-20	3	Couch	Provincial	2	670	1340	1340

II. Find top 3 products and their finish shade in terms of total quantities sold between 01/01/2018 and 12/31/2021. Your result dataset should have Product_Id, Product Description, shade, and total quantity sold.

ANS:

```
WITH ProductQuantityRank AS (
    SELECT
        p.prod_id,
        p.descr AS product_description,
        f.shade,
        SUM(op.quant) AS total_quantity_sold,
        RANK() OVER (ORDER BY SUM(op.quant) DESC) AS quantity_rank
    FROM
        sk_prod p
        JOIN sk_ord_prod op ON p.prod_id = op.prod_id
        JOIN sk_finish f ON op.finish_id = f.finish_id
        JOIN sk_order o ON op.order_id = o.order_id
    WHERE
        o.order_date BETWEEN TO_DATE('01-JAN-2018', 'DD-MON-YYYY') AND TO_DATE('31-DEC-2021', 'DD-MON-YYYY')
    GROUP BY
        p.prod_id, p.descr, f.shade
)
SELECT
    prod_id,
    product_description,
    shade,
    total_quantity_sold
FROM
    ProductQuantityRank
WHERE
    quantity_rank <= 3;
```



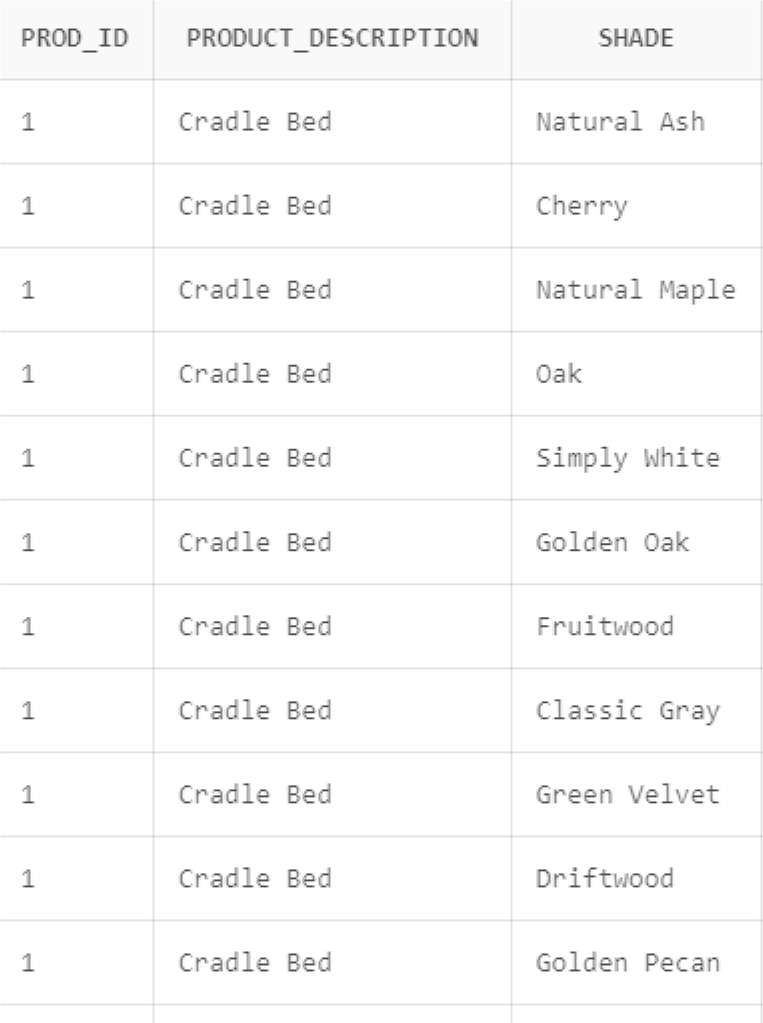
PROD_ID	PRODUCT_DESCRIPTION	SHADE	TOTAL_QUANTITY_SOLD
10	Bookcase	Cherry	5
9	Washstand	Simply White	3
11	4-Dr Dresser	Oak	2
3	Couch	Provincial	2
15	Nightstand	Fruitwood	2
14	Windsor Chair	Natural	2
4	Entertainment Center	Red Chestnut	2
8	Wardrobe	Classic Gray	2
6	Settee	Puritan Prine	2

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

ANS:

```
SELECT
   p.prod_id,
   p.descr AS product_description,
   f.shade
FROM
   sk_prod p
   CROSS JOIN sk_finish f
WHERE
   NOT EXISTS (
      SELECT 1
      FROM sk_ord_prod op
      JOIN sk_order o ON op.order_id = o.order_id
      WHERE
         p.prod_id = op.prod_id
```

```
                    AND f.finish_id = op.finish_id
                    AND EXTRACT(MONTH FROM o.order_date) IN (10, 11, 12)
                    AND EXTRACT(YEAR FROM o.order_date) = 2020
            );
```



PROD_ID	PRODUCT_DESCRIPTION	SHADE
1	Cradle Bed	Natural Ash
1	Cradle Bed	Cherry
1	Cradle Bed	Natural Maple
1	Cradle Bed	Oak
1	Cradle Bed	Simply White
1	Cradle Bed	Golden Oak
1	Cradle Bed	Fruitwood
1	Cradle Bed	Classic Gray
1	Cradle Bed	Green Velvet
1	Cradle Bed	Driftwood
1	Cradle Bed	Golden Pecan

And many more

Submission:

For each question write an SQL query and submit both SQL and a clearly visible screenshot of corresponding result underneath each question. Make appropriate use of column alias and built in functions in your SQL queries.