1. [10 pts] For each airport, return its IATA code and number of lounges. You don't need to return an airport if it doesn't have any lounge.

a) [7pts] SQL

```
SELECT IATA_code, COUNT(*)
FROM Airport, Lounge
WHERE Airport.IATA_code = Lounge.airport_IATA_code
GROUP BY Airport.IATA_CODE;
```

b) [3pts] Results

IATA_code	COUNT(*)
JFK	2
LAX	1
SAT	2
SFO	3
SJC	1
SNA	2

For Questions 2 through 4, we use two types of flight duration (as a number in seconds):

- projected_flight_duration = projected_arrival_datetime - projected_departure_datetime
- actual_flight_duration = actual_arrival_datetime - actual_departure_datetime

The "-" operation can be implemented by using the timestampdiff() function.

2. [10pts] Among all the flights, find the flight number and actual duration of the latest flight based on projected departure datetimes.

a) [7 pts] SQL

```
SELECT flight_number, TIMESTAMPDIFF(SECOND, actual_departure_datetime,
actual_arrival_datetime) as duration
FROM Flight
WHERE projected_departure_datetime = (SELECT MAX(projected_departure_datetime) FROM
Flight);
```

b) [3 pts] Results

flight_number	duration
UC2084	8100

3. [10 pts] Among all the flights, return the maximum absolute difference between a flight's projected duration and its actual duration.

a) [7 pts] SQL

```
SELECT MAX( ABS(TIMESTAMPDIFF(SECOND, actual_departure_datetime,
    actual_arrival_datetime) - TIMESTAMPDIFF(SECOND, projected_departure_datetime,
    projected_arrival_datetime)) ) as max_difference
FROM Flight;
```

b) [3 pts] Results

max_difference
3600

4. [10 pts] For each flight number, return its flight number, minimum and maximal absolute difference between its projected duration and actual duration. For example, if there are two instances of a flight number 'UC9049' and their duration differences are 100 and 200 seconds, respectively, then its maximum duration difference for 'UC9049' is 200 seconds.

a) [7 pts] SQL

```
SELECT
  flight_number,  min( ABS((TIMESTAMPDIFF(SECOND,
    actual_departure_datetime,
    actual_arrival_datetime) - TIMESTAMPDIFF(SECOND,
    projected_departure_datetime,
    projected_arrival_datetime)))), max( ABS((TIMESTAMPDIFF(SECOND,
    actual_departure_datetime,
    actual_arrival_datetime) - TIMESTAMPDIFF(SECOND,
    projected_departure_datetime,
    projected_arrival_datetime))))
FROM
  Flight group by flight_number;
```

b) [3 pts] Results

flight_number	min	max
N124	0	600
U987	0	120
UC2084	0	3600
UC6024	300	900
UC725	0	0

5. [10 pts] Return the employee id and number of flights for the pilots with the maximum number of flight instances that they operated.

a) [7 pts] SQL

```
SELECT pid, COUNT(*)
FROM Pilot_Operates_Flight POF
GROUP BY pid
HAVING COUNT(*) = (SELECT MAX(CNT) FROM (
SELECT pid,COUNT(*) as cnt
FROM Pilot_Operates_Flight
GROUP BY pid) A);
```

b) [3 pts] Results

pid	count(*)
990201	23

6. [10 pts] Find the ids and average menu price of lounges whose average menu price is greater than the overall average menu price.

a) [7 pts] SQL

```
SELECT lid, avg(price)
FROM Dish
GROUP BY lid
HAVING avg(price) > (
SELECT avg(price) FROM Dish);
```

b) [3 pts] Results

lid	avg(price)
212	26.096
213	21.3725
315	36.014

7. [10 pts] For each dish order with at least two different dishes, return its order id, name, and quantity of each dish in the order.

a) [7 pts] SQL

```
SELECT o.oid, oc.name, quantity
FROM DishOrder_Contains_Dish OC, DishOrder O, Dish D
WHERE
   d.lid = oc.lid AND d.name = oc.name AND oc.oid = o.oid and o.oid IN
  (SELECT oid FROM
   DishOrder_Contains_Dish
   GROUP BY oid
   HAVING COUNT(*) > 1);
```

b) [3 pts] Results

oid	name	quantity
4	fresh lemonade	2
4	sandwich	2
4	the thai wrap	2
12	salmon	20
12	skewered shrimp	20
12	swordfish	20
5	galbitang	4
5	samgyetang	3

3	salmon	3
3	swordfish	5
8	hummus	10
8	the burger combo	5
8	the karma burger	3

8. [10 pts] Return the customer ids along with their total prices for all their flight reservations. Make sure to include customers without any flight reservation. Sort the final result by customer id.

a) [7 pts] SQL

```
SELECT C.cid, total_price
FROM Customer C LEFT OUTER JOIN ((SELECT cid, sum(purchased_price) as total_price
FROM Customer_Reserves_Flight
GROUP BY cid) CRF) ON C.cid = CRF.cid ORDER BY C.cid;
```

b) [3 pts] Results

cid	total_price
1	2782.1
2	137.22
3	250
4	500
5	400
6	125
7	150
8	2500
9	2800

10	290
11	NULL
12	NULL
13	2821
14	473
15	4730
16	3530
17	9672.88
18	1572.34
19	1572.34
20	420.98

9. [10 pts] For each airport, return its IATA_code and count of orders received by all its lounges. Include an airport on the list only if it has at least one lounge. Sort the results by IATA_code.

a) [7 pts] SQL

```
SELECT airport_IATA_code, sum(cnt) FROM (
(SELECT L.lid, L.airport_IATA_code, cnt FROM
Lounge L LEFT OUTER JOIN ((SELECT lid, count(*) as cnt FROM DishOrder GROUP BY lid) DCD)
ON
L.lid = DCD.lid) B) GROUP BY airport_IATA_code ORDER BY airport_IATA_code;
```

b) [3 pts] Results

airport_IATA_code	sum(cnt)
JFK	4
LAX	1
SAT	5
SFO	NULL
SJC	NULL
SNA	4

10. [10 pts] For each lounge, return its id, IATA_code, number of dishes, lowest dish price, highest dish price, and average dish price. The id and IATA_code of a lounge should be returned even if there are no dishes served at the lounge. Sort the results by the lounge id.

a) [7 pts] SQL

```
SELECT L1.lid, L1.airport_IATA_code as IATA_code, cnt, minv, maxv, avgv
From Lounge L1 LEFT OUTER JOIN
((SELECT lid, count(*) as cnt, avg(price) as avgv, min(price) as minv, max(price) as maxv
FROM Dish
GROUP BY lid) D) ON L1.lid = D.lid
ORDER BY L1.lid;
```

b) [3 pts] Results

lid	IATA_code	count(*)	avg(price)	min(price)	max(price)
112	SNA	4	10.2375	6.5	19
113	SNA	4	16.25	10.5	31.5
212	SAT	5	26.096	10.99	49
213	SAT	4	21.3725	12	35.5
314	JFK	4	17.7425	11.99	29.99
315	JFK	5	36.014	13.99	97.1

409	LAX	4	13.495	11	16.99
501	SFO	1	19.12	19.12	19.12
502	SFO	1	17.12	17.12	17.12
503	SFO	1	13.12	13.12	13.12
601	SJC	NULL	NULL	NULL	NULL