

Suggested Answers

1. How many usernames contain the letter A? (1)

```
SELECT COUNT(*) AS numNames
FROM customer
WHERE username LIKE '%a%'; /*OR*/ WHERE username REGEXP 'a';
```

OR`

```
SELECT COUNT(*) AS numNames
FROM customer
WHERE INSTR(username, 'a') > 0; /*OR*/WHERE LOCATE('a',username)>0;
```

numNames
3

1 row returned

2. What is the average journey length, in seconds? (1)

```
SELECT ROUND(AVG(TIME_TO_SEC(TIMEDIFF(endTime, startTime))),2)
AS AvgJourneyLength
FROM journey;
```

OR

```
SELECT ROUND(AVG(UNIX_TIMESTAMP(endTime) -
UNIX_TIMESTAMP(startTime)),2)
AS AvgJourneyLength
FROM journey;
```

AvgJourneyLength
312.00

1 row returned

**3. List the stations on the Eastern line (excluding City),
in outbound order. (2)**

```
SELECT station.name, station.line, station.sequence
FROM station
WHERE line in
    (SELECT id FROM line
     WHERE name = 'Eastern')
ORDER BY sequence;
```

OR

```
SELECT station.name, station.line, station.sequence
FROM station JOIN line
ON station.line = line.id
WHERE line.name = 'Eastern'
ORDER BY sequence;
```

name	line	sequence
Gallery	3	1
Library	3	2
Shops	3	3

3 row returned

**4. On which day of the week (Monday, Tuesday etc.) are the most
journeys made? (2)**

```
SELECT DAYNAME(startTime) as day, count(*) as numJourneys
FROM journey
GROUP BY day
ORDER BY count(*) DESC
LIMIT 1;
```

day	numJourneys
Thursday	9

1 row returned

5. List any stations at which no passenger has started or ended a journey. Show the station id and name. (3)

```
SELECT id, name FROM station
WHERE id NOT IN
    (SELECT startStation as station FROM journey
     UNION
     SELECT endStation as station from journey);
```

OR

```
SELECT id, name FROM station
WHERE id NOT IN
    (SELECT startStation FROM journey)
AND id NOT IN
    (SELECT endStation FROM journey);
```

OR

```
SELECT station.id, station.name
FROM station
WHERE NOT EXISTS
    (SELECT * FROM journey
     WHERE station.id = startStation
     OR station.id = endStation);
```

id	name
6	House

1 row returned

6. List the details of each journey, along with its cost. (3)

```

SELECT journey.*,
CASE
    WHEN origin.zone = 2 AND destination.zone = 2
        AND origin.line <> destination.line THEN 3
    WHEN origin.zone <> destination.zone THEN 2
    ELSE 1
END as cost
FROM journey JOIN station origin ON startStation = origin.id
JOIN station destination ON endStation = destination.id
ORDER BY journey.id;

```

id	customer	startStation	startTime	endStation	endTime	cost
1	1	4	2018-02-01 01:01:00	3	2018-02-01 01:03:00	1
2	2	4	2018-02-01 02:02:00	2	2018-02-01 02:06:00	2
3	3	4	2018-02-01 03:03:00	1	2018-02-01 03:08:00	2
4	4	3	2018-02-01 04:04:00	2	2018-02-01 04:05:00	2
5	5	3	2018-02-01 05:05:00	1	2018-02-01 05:09:00	2
6	1	2	2018-02-01 06:06:00	1	2018-02-01 06:08:00	1
7	2	1	2018-02-01 07:07:00	11	2018-02-01 07:14:00	2
8	3	4	2018-02-01 08:08:00	9	2018-02-01 08:14:00	2
9	4	4	2018-02-01 09:09:00	10	2018-02-01 09:17:00	2
10	5	4	2018-02-02 10:10:00	11	2018-02-02 10:22:00	3

20 row returned

7. List the station ids, along with the number of journeys that started or stopped at each station. (4)

```
SELECT stationid, count(*) AS numJourneys
FROM
    (SELECT startStation AS stationid FROM journey
    UNION ALL
    SELECT endStation AS stationid FROM journey) AS stationsUsed
GROUP BY stationid;
```

stationid	numJourneys
1	6
2	3
3	3
4	6
5	1
7	5
8	9
9	1
10	2
11	4

OR

```
SELECT station.id, COUNT(journey.id) AS numJourneys
FROM station LEFT JOIN journey
ON station.id=journey.startStation OR station.id=journey.endStation
GROUP BY station.id;
```

OR

```
SELECT id,
    (SELECT COUNT( * ) FROM journey
    WHERE journey.startStation = station.id
    OR journey.endStation = station.id) AS numberOfJourney
FROM station;
```

stationId	numJourneys
1	6
2	3
3	3
4	6
5	1
6	0
7	5
8	9
9	1
10	2
11	4

10 or 11 rows returned

8. List the journeys that ended at the last station on the line.
(the station with the highest sequence number for a line) (4)

```
SELECT * FROM journey
WHERE endStation in
    (SELECT id FROM station JOIN
        (SELECT line, MAX(sequence) as last
        FROM station
        WHERE line != 0
        GROUP BY line) AS lastSequence
    ON station.line = lastSequence.line
    AND station.sequence = lastSequence.last
)
ORDER BY journey.id;
```

id	customer	startStation	startTime	endStation	endTime
7	2	1	2018-02-01 07:07:00	11	2018-02-01 07:14:00
10	5	4	2018-02-02 10:10:00	11	2018-02-02 10:22:00
14	4	7	2018-02-02 14:14:00	8	2018-02-02 14:16:00
15	1	7	2018-02-02 15:15:00	8	2018-02-02 15:16:00
19	2	8	2018-02-03 19:19:00	11	2018-02-03 19:33:00
20	1	1	2018-02-03 20:20:00	11	2018-02-03 20:26:00

6 row returned

9. For each journey, show how many stations it passed through.

(Count the end station but not the start station.)

(5)

```
SELECT journey.*,  
CASE  
    WHEN origin.line = destination.line  
        THEN ABS(CAST(origin.sequence as SIGNED) -  
                CAST(destination.sequence as SIGNED))  
    WHEN origin.line != destination.line  
        THEN origin.sequence + destination.sequence  
END as numStations  
FROM journey JOIN station origin ON startStation = origin.id  
JOIN station destination ON endStation = destination.id  
ORDER BY journey.id;
```

id	customer	startStation	startTime	endStation	endTime	numStations
1	1	4	2018-02-01 01:01:00	3	2018-02-01 01:03:00	1
2	2	4	2018-02-01 02:02:00	2	2018-02-01 02:06:00	2
3	3	4	2018-02-01 03:03:00	1	2018-02-01 03:08:00	3
4	4	3	2018-02-01 04:04:00	2	2018-02-01 04:05:00	1
5	5	3	2018-02-01 05:05:00	1	2018-02-01 05:09:00	2
6	1	2	2018-02-01 06:06:00	1	2018-02-01 06:08:00	1
7	2	1	2018-02-01 07:07:00	11	2018-02-01 07:14:00	3
8	3	4	2018-02-01 08:08:00	9	2018-02-01 08:14:00	4
9	4	4	2018-02-01 09:09:00	10	2018-02-01 09:17:00	5
10	5	4	2018-02-02 10:10:00	11	2018-02-02 10:22:00	6

20 row returned

10. List the usernames of customers who have travelled on all lines. (5)

```
SELECT username FROM CUSTOMER
WHERE NOT EXISTS
    (SELECT * FROM LINE
    WHERE NOT EXISTS
        (SELECT * FROM
            (SELECT customer, line FROM journey JOIN station
            ON journey.startStation = station.id
            WHERE line != 0
            UNION
            SELECT customer, line FROM journey JOIN station
            ON journey.endStation = station.id
            WHERE line != 0) AS customerLine
        WHERE customerLine.customer = customer.id
        AND customerLine.line = line.id)
    );
```

OR

```
SELECT username FROM customer
WHERE id in (
    SELECT customer FROM
        (SELECT customer, line FROM journey JOIN station
        ON journey.startStation = station.id
        WHERE line != 0
        UNION
        SELECT customer, line FROM journey JOIN station
        ON journey.endStation = station.id
        WHERE line != 0) AS customerLine
    GROUP BY customer
    HAVING count(line) = (SELECT COUNT(*) FROM LINE)
);
```


OR

```
SELECT Username FROM customer
WHERE id IN
    (SELECT Customer
     FROM journey INNER JOIN station
     ON journey.startStation = station.id
     OR journey.endStation = station.id
     WHERE station.line > 0
     GROUP BY Customer
     HAVING COUNT(DISTINCT station.line) =
        (SELECT COUNT(DISTINCT line)
         FROM station
         WHERE line > 0))
```

id	username
1	alice
2	bob
3	carol
4	dan

4 row returned