1. How many usernames contain the letter A?

```
SELECT COUNT(*)
FROM Customer
WHERE username LIKE '%A%';
```

number
3

1 Row Returned

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

```
SELECT AVG(TIMESTAMPDIFF(second, startTime, endTime)) AS 'average length'
FROM Journey;
```

average length
312.0000

1 Row Returned

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

```
SELECT name AS 'station'
FROM Station
WHERE line=3
ORDER BY sequence ASC;
```

station
Gallery
Library
Shops

3 Rows Returned

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

```
SELECT DATE_FORMAT(startTime, '%W') AS 'day'
FROM Journey
GROUP BY DATE_FORMAT(startTime, '%W')
ORDER BY COUNT(DATE_FORMAT(startTime, '%W')) DESC
LIMIT 1;
```

day
Thursday

1 Row Returned

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

```
SELECT id AS 'station id',name AS 'station name'
FROM Station
WHERE id NOT IN (SELECT startStation FROM Journey)
AND id NOT IN (SELECT endStation FROM Journey);
```

station id	station name
6	House

1 Row Returned

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

```
SELECT a.*, CASE WHEN b.zone!=c.zone THEN '2$'
                 WHEN b.zone=2 AND c.zone=2 AND b.line!=c.line THEN '3$'
                 ELSE '1$'
            END AS 'cost'
FROM Journey AS a INNER JOIN Station AS b INNER JOIN Station AS c ON a.startStation=b.id AND a.endStation=c.id
ORDER BY a.id;
```

id	customer	startStati...	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 Rows Returned

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

```
SELECT Station.id AS 'station id', COUNT(Journey.id) AS 'number'
FROM Journey RIGHT OUTER JOIN Station ON Station.id=Journey.startStation OR Station.id=Journey.endStation
GROUP BY Station.id;
```

station id	number
1	6
2	3
3	3
4	6
5	1
6	0
7	5
8	9
9	1
10	2

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)

```
SELECT *
FROM Journey
WHERE endStation IN
(SELECT id FROM Station
WHERE sequence IN (SELECT MAX(sequence) FROM Station WHERE line=1) AND line=1
OR sequence IN (SELECT MAX(sequence) FROM Station WHERE line=2) AND line=2
OR sequence IN (SELECT MAX(sequence) FROM Station WHERE line=3) AND line=3)
ORDER BY id;
```

id	customer	startStati...	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 Rows Returned

9. For each journey, show how many stations it passed through. (Count the end station but not the start station.)

```
SELECT a.*, CASE WHEN b.line!=c.line
                    THEN b.sequence + c.sequence
                     ELSE ABS(CAST(b.sequence AS SIGNED)- CAST(c.sequence AS SIGNED))
               END AS 'number'
FROM Journey AS a INNER JOIN Station AS b INNER JOIN Station AS c ON a.startStation=b.id AND a.endStation=c.id;
```

id	number
1	1
2	2
3	3
4	1
5	2
6	1
7	3
8	4
9	5
10	6

20 Rows Returned

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

```
SELECT username
FROM Customer
WHERE id IN (SELECT Journey.customer
                FROM Station INNER JOIN Journey ON Station.id=Journey.startStation OR Station.id=Journey.endStation INNER JOIN Line ON Line.id=Station.line
                GROUP BY Journey.customer
                HAVING COUNT(DISTINCT Line.id)=3);
```

username
alice
bob
carol
dan

4 Rows Returned