



Question 1: 4 points.

Let's say you want to produce a time-of-day summary for the records in the **mail** table, which looks like this:

```
mysql> SELECT * FROM mail;
```

t	srcuser	srchost	dstuser	dsthost	size
2001-05-11 10:15:08	barb	saturn	tricia	mars	58274
2001-05-12 12:48:13	tricia	mars	gene	venus	194925
2001-05-12 15:02:49	phil	mars	phil	saturn	1048
2001-05-13 13:59:18	barb	saturn	tricia	venus	271
2001-05-14 09:31:37	gene	venus	barb	mars	2291
2001-05-14 11:52:17	phil	mars	tricia	saturn	5781

a) Write query to determine how many messages were sent for each hour of the day?

Answer:

```
mysql> SELECT HOUR(t) AS hour, COUNT(HOUR(t)) AS count
-> FROM mail GROUP BY hour;
```

hour	count
7	1
8	1
9	2
10	2
11	1
12	2
13	1
14	1
15	1
17	2
22	1
23	1

b) Write query to produce a summary that includes all hours of the day, even those during which no messages were sent?

Answer:

First create a reference table that lists each hour:

```
mysql> CREATE TABLE ref (h INT);
mysql> INSERT INTO ref (h)
-> VALUES (0), (1), (2), (3), (4), (5), (6), (7), (8), (9), (10),
-> (11), (12), (13), (14), (15), (16), (17), (18), (19), (20),
-> (21), (22), (23);
```

Then join the reference table to the mail table using a LEFT JOIN:

```
mysql> SELECT ref.h AS hour, COUNT(HOUR(mail.t)) AS count
-> FROM ref LEFT JOIN mail ON ref.h = HOUR(mail.t)
-> GROUP BY hour;
```

hour	count
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	1
8	1
9	2
10	2
11	1
12	2
13	1
14	1
15	1
16	0
17	2
18	0
19	0
20	0
21	0
22	1
23	1

Question 2: 2 points

Assume that, you take an automobile trip and write down the total miles traveled at each stopping point, you want to compute the difference between successive points to determine the distance from one stop to the next. Here is such a table that shows the stops for a trip from San Antonio, Texas to Madison, Wisconsin. Each row shows the total miles driven as of each stop:

```
mysql> SELECT seq, city, miles FROM trip_log;
```

seq	city	miles
1	San Antonio, TX	0
2	Dallas, TX	263
3	Benton, AR	566
4	Memphis, TN	745
5	Portageville, MO	878
6	Champaign, IL	1164
7	Madison, WI	1412

Write query to compute the differences between each pair of mileage values?

seq1	seq2	city1	city2	miles1	miles2	dist
1	2	San Antonio, TX	Dallas, TX	0	263	263
2	3	Dallas, TX	Benton, AR	263	566	303
3	4	Benton, AR	Memphis, TN	566	745	179
4	5	Memphis, TN	Portageville, MO	745	878	133
5	6	Portageville, MO	Champaign, IL	878	1164	286
6	7	Champaign, IL	Madison, WI	1164	1412	248

Answer:

```
mysql> SELECT t1.seq AS seq1, t2.seq AS seq2,  
-> t1.city AS city1, t2.city AS city2,  
-> t1.miles AS miles1, t2.miles AS miles2,  
-> t2.miles-t1.miles AS dist  
-> FROM trip_log AS t1, trip_log AS t2  
-> WHERE t1.seq+1 = t2.seq  
-> ORDER BY t1.seq;
```

Question 3: 2points

Suppose you have two tables that list prospective and actual customers, a third that lists vendors from whom you purchase supplies, and you want to create a single mailing list by merging names and addresses from all three tables. UNION provides a way to do this. Assume the three tables have the following contents:

```
mysql> SELECT * FROM prospect;
```

```
+-----+-----+-----+
| fname | lname | addr |
+-----+-----+-----+
| Peter | Jones | 482 Rush St., Apt. 402 |
| Bernice | Smith | 916 Maple Dr. |
+-----+-----+-----+
```

```
mysql> SELECT * FROM customer;
```

```
+-----+-----+-----+
| last_name | first_name | address |
+-----+-----+-----+
| Peterson | Grace | 16055 Seminole Ave. |
| Smith | Bernice | 916 Maple Dr. |
| Brown | Walter | 8602 1st St. |
+-----+-----+-----+
```

```
mysql> SELECT * FROM vendor;
```

```
+-----+-----+
| company | street |
+-----+-----+
| ReddyParts, Inc. | 38 Industrial Blvd. |
| Parts-to-go, Ltd. | 213B Commerce Park. |
+-----+-----+
```

Answer:

```
mysql> SELECT fname, lname, addr FROM prospect
-> UNION
-> SELECT first_name, last_name, address FROM customer
-> UNION
-> SELECT company, '', street FROM vendor;
```

```
+-----+-----+-----+
| fname | lname | addr |
+-----+-----+-----+
| Peter | Jones | 482 Rush St., Apt. 402 |
| Bernice | Smith | 916 Maple Dr. |
| Grace | Peterson | 16055 Seminole Ave. |
| Walter | Brown | 8602 1st St. |
| ReddyParts, Inc. | | 38 Industrial Blvd. |
| Parts-to-go, Ltd. | | 213B Commerce Park. |
+-----+-----+-----+
```

Question 4: 4 points

Assume the following table:

```
mysql> CREATE TABLE log (  
-> name CHAR(30) NOT NULL,  
-> location CHAR(30) NOT NULL,  
-> counter INT UNSIGNED NOT NULL,  
-> PRIMARY KEY (name, location));
```

Normally, if you attempt to insert a row into a log table that would result in a duplicate-key error for a unique-valued index. Write two methods to deal with duplicate values in a unique-valued index when adding new records to a table.

Answer:

First Solution:

```
mysql> INSERT INTO log (name, location, counter)  
-> VALUES ('Tantor', 'Waterhole', 1)  
-> ON DUPLICATE KEY UPDATE counter=counter+1;
```

```
mysql> SELECT * FROM log;
```

name	location	counter
Tantor	Waterhole	1

```
mysql> INSERT INTO log (name, location, counter)  
-> VALUES ('Tantor', 'Waterhole', 1)  
-> ON DUPLICATE KEY UPDATE counter=counter+1;
```

```
mysql> SELECT * FROM log;
```

name	location	counter
Tantor	Waterhole	2

Second Solution:

```
mysql> REPLACE INTO people (name, location, counter)  
-> VALUES ('Tantor', 'Waterhole', 1);
```

Third Solution:

```
mysql> INSERT IGNORE INTO log (name, location, counter)  
-> VALUES ('Tantor', 'Waterhole', 1);
```

Question 5: 3 points

Assume the following tables:

Country(Code, Name, Continent)

CountryLanguage(CountryCode , Language)

Write three queries to find countries on the European continent where Spanish is spoken.

Answer:

```
mysql> SELECT Name
-> FROM Country
-> WHERE Continent = 'Europe'
-> AND Code = ANY (SELECT CountryCode
-> FROM CountryLanguage
-> WHERE Language = 'Spanish')
-> ORDER BY Name;
```

```
mysql> SELECT Name
-> FROM Country
-> WHERE Continent = 'Europe'
-> AND Code in (SELECT CountryCode
-> FROM CountryLanguage
-> WHERE Language = 'Spanish')
-> ORDER BY Name;
```

```
SELECT Code c, Name
-> FROM Country
-> WHERE Continent = 'Europe'
-> AND EXISTS (SELECT *
-> FROM CountryLanguage
-> WHERE CountryCode = c
-> AND Language = 'Spanish');
```

```
SELECT Code, Name
-> FROM Country, CountryLanguage
-> WHERE Continent = 'Europe' AND CountryCode = code
-> AND Language = 'Spanish';
```

