1. List names of all the languages in the database (sorted alphabetically)

   ```
   SELECT name AS language FROM language
   ORDER BY name;

   +----------+
   | language |
   +----------+
   | English  |
   | French   |
   | German   |
   | Italian  |
   | Japanese |
   | Mandarin |
   +----------+
   6 rows in set (0.00 sec)
   ```

2. List full names of actors with ”GER” in their last name, ordered by their first name

   ```
   SELECT CONCAT(first_name,’ ’,last_name) AS fullname FROM actor
   WHERE last_name LIKE ’%GER%’
   ORDER BY first_name;

   +-------------------+
   | fullname          |
   +-------------------+
   | CAMERON ZELLWEGER |
   | JULIA ZELLWEGER   |
   | MICHAEL BOLGER    |
   | MINNIE ZELLWEGER  |
   | VAL BOLGER        |
   | VIVIEN BASINGER   |
   +-------------------+
   6 rows in set (0.06 sec)
   ```

3. Find all the addresses where postal code starts with ”57”, and return addresses sorted

   ```
   SELECT address,postal_code FROM address
   WHERE postal_code LIKE ’57%’
   ORDER BY address;

   +------------------------------------+-------------+
   | address                            | postal_code |
   ```

```
+----------------------------------+-------------+
| 1029 Dzerzinsk Manor             | 57519       |
| 1061 Ede Avenue                  | 57810       |
| 1287 Xiangfan Boulevard          | 57844       |
| 1309 Weifang Street              | 57338       |
| 1515 Korla Way                   | 57197       |
| 1697 Kowloon and New Kowloon Loop | 57807       |
| 231 Kaliningrad Place            | 57833       |
| 390 Wroclaw Way                  | 5753        |
| 431 Szkesfehrvr Avenue           | 57828       |
| 687 Alessandria Parkway          | 57587       |
| 869 Shikarpur Way                | 57380       |
+----------------------------------+-------------+
11 rows in set (0.00 sec)
```

4. How many films involve a ”DWARF” in their titles?

```
SELECT COUNT(*) AS DwarfCount FROM film
WHERE title LIKE ’%DWARF%’;
```

```
+------------+
| DwarfCount |
+------------+
|          2 |
+------------+
1 row in set (0.00 sec)
```

5. Find full names of actors who played in a film involving WAR in title and longer than 2.5 hours, along with the title, run length and release year of the movie, sorted by the actors last names.

```
SELECT CONCAT(first_name,’ ’,last_name) AS fullname,title,(length/60) AS RunLength,
FROM actor
JOIN film_actor
ON actor.actor_id = film_actor.actor_id
JOIN film
ON film_actor.film_id = film.film_id
WHERE title LIKE ’%WAR%’ AND (length/60) >= 2.6
ORDER BY last_name;
```

```
+-------------------+--------------------+-----------+--------------+
| fullname          | title              | RunLength | release_year |
+-------------------+--------------------+-----------+--------------+
| ANGELINA ASTAIRE  | WARDROBE PHANTOM   |    2.9667 |         2006 |
| RIVER DEAN        | WARDROBE PHANTOM   |    2.9667 |         2006 |
| GREGORY GOODING   | WARDROBE PHANTOM   |    2.9667 |         2006 |
| ADAM GRANT        | GLADIATOR WESTWARD |    2.8833 |         2006 |
| KIRK JOVOVICH     | GLADIATOR WESTWARD |    2.8833 |         2006 |
| KIRK JOVOVICH     | WARDROBE PHANTOM   |    2.9667 |         2006 |
| JULIA MCQUEEN     | GLADIATOR WESTWARD |    2.8833 |         2006 |
| WARREN NOLTE      | WARDROBE PHANTOM   |    2.9667 |         2006 |
| JAYNE SILVERSTONE | GLADIATOR WESTWARD |    2.8833 |         2006 |
| IAN TANDY         | GLADIATOR WESTWARD |    2.8833 |         2006 |
| THORA TEMPLE      | WARDROBE PHANTOM   |    2.9667 |         2006 |
| WALTER TORN       | WARDROBE PHANTOM   |    2.9667 |         2006 |
| NICK WAHLBERG     | WARDROBE PHANTOM   |    2.9667 |         2006 |
| CAMERON ZELLWEGER | GLADIATOR WESTWARD |    2.8833 |         2006 |
+-------------------+--------------------+-----------+--------------+
14 rows in set (0.00 sec)
```

6. Find all the film categories in which there are between 55 and 65 films. Return the names of these categories and the number of films per category, sorted by the number of films descending.

```
SELECT name AS Category,COUNT(*) AS CategoryFilms
FROM category
JOIN film_category
ON category.category_id = film_category.category_id
JOIN film
ON film_category.film_id = film.film_id
GROUP BY Category
HAVING COUNT(*) BETWEEN 55 AND 65
ORDER BY CategoryFilms DESC;
```

```
+----------+---------------+
| Category | CategoryFilms |
+----------+---------------+
| Action   |            64 |
| New      |            63 |
| Drama    |            62 |
| Sci-Fi   |            61 |
| Games    |            61 |
| Children |            60 |
| Comedy   |            58 |
| Travel   |            57 |
| Classics |            57 |
| Horror   |            56 |
+----------+---------------+
10 rows in set (0.11 sec)
```

7. In how many film categories is the average difference between the film replacement cost and the rental rate larger than 17?

```
SELECT COUNT(*) AS CategoryFilmsCount
FROM
(
SELECT AVG(replacement_cost-rental_rate) AS AverageDiff
FROM film_category
JOIN film
ON film_category.film_id = film.film_id
GROUP BY category_id
HAVING AverageDiff >= 18
)AS temp_table;
```

```
+--------------------+
| CategoryFilmsCount |
+--------------------+
|                  3 |
+--------------------+
1 row in set (0.00 sec)
```

8. Find the address district(s) name(s) such that the minimal postal code in the district(s) is maximal over all the districts. Make sure your query ignores empty postal codes and district names.

```
SELECT district,MinimalCode AS PostalCode FROM
(
SELECT district,MIN(postal_code) AS MinimalCode FROM address
WHERE district IS NOT NULL AND postal_code IS NOT NULL
GROUP BY district
)AS temp_table
ORDER BY PostalCode DESC LIMIT 0,1;
```

```
+----------+------------+
| district | PostalCode |
+----------+------------+
| Crdoba   | 99865      |
+----------+------------+
1 row in set (0.00 sec)
```

9. Find the names (first and last) of all the actors and customers whose first name is the same as the first name of the actor with ID 101 (exclude the actor with ID 101).

```
SELECT CONCAT(first_name,’ ’,last_name) AS fullname FROM customer
WHERE first_name =
(
SELECT first_name FROM actor
WHERE actor_id = 101
)
UNION
SELECT CONCAT(first_name,’ ’,last_name) AS fullname FROM actor
WHERE actor_id != 101 AND first_name =
(
SELECT first_name FROM actor
WHERE actor_id = 101
);
```

```
+--------------+
| fullname     |
+--------------+
| SUSAN WILSON |
| SUSAN DAVIS  |
+--------------+
2 rows in set (0.00 sec)
```