Practice Review for SQL

Question 1:

Find all ward names that are not unique, and print them in alphabetical order (only once each)

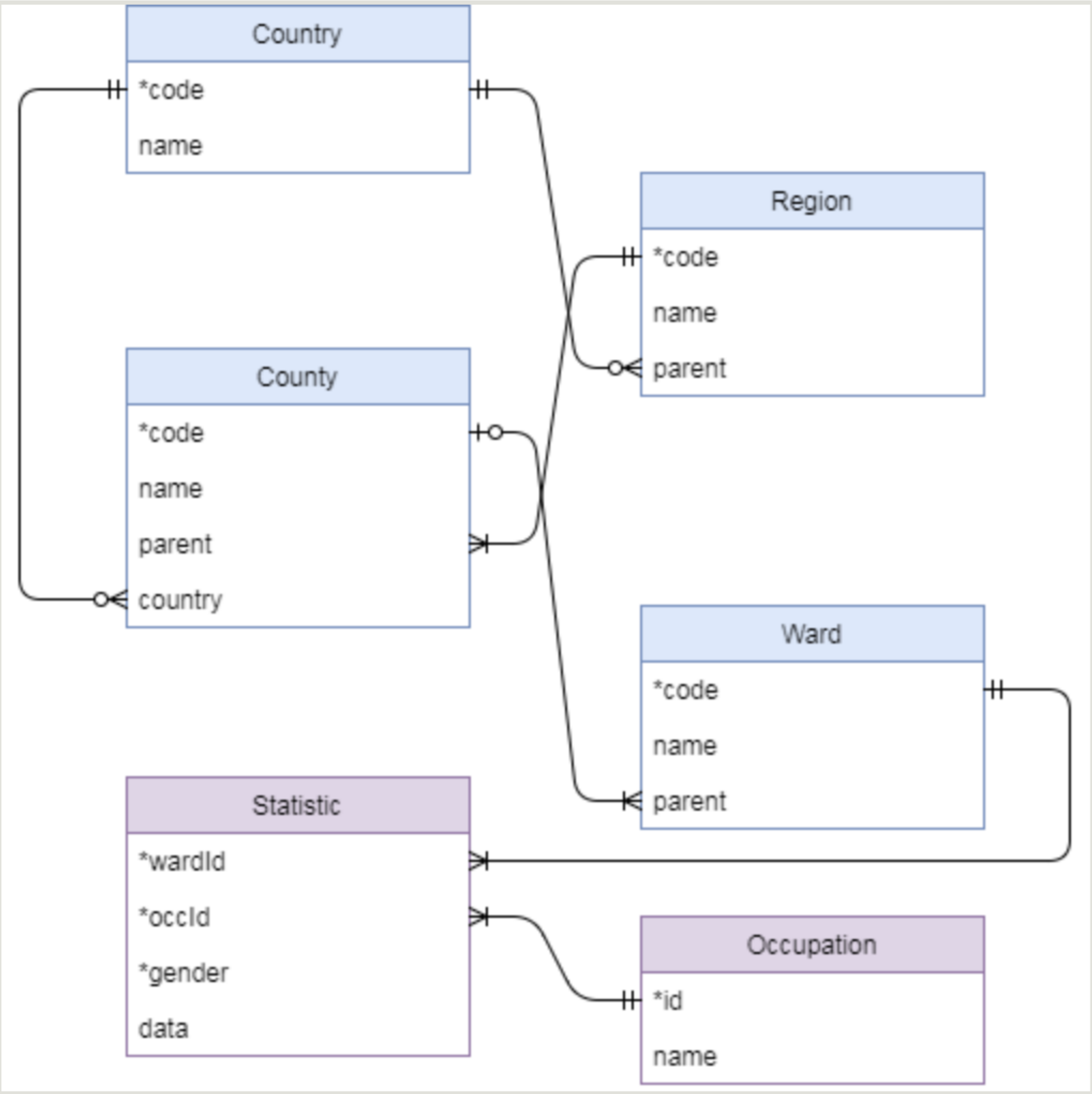
- use a **subquery** to first find the ward names that occur more than once in the database then select from that result

```
SELECT Ward.name
FROM WARD
GROUP BY Ward.name
HAVING COUNT(Ward.name) > 1
ORDER BY Ward.name ASC;
```

Question 2:

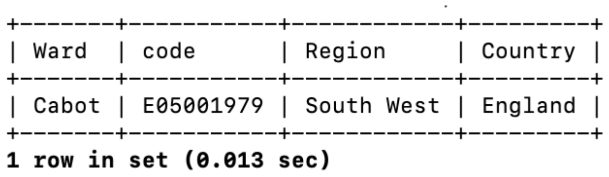
The university of Bristol is situated in the `Cabot` ward (ward names are not always distinct, but this one is). Find the names and codes of the CLU, region and country containing the Cabot ward (CLU

= county level unit = "row in County table")



```
Ward.name AS Ward, Ward.code, Region.name AS Region, Country.name AS Country
FROM Ward
INNER JOIN County ON Ward.parent = County.code
INNER JOIN Region ON County.parent = Region.code
INNER JOIN Country ON Region.parent = COuntry.code
WHERE WArd.name = 'Cabot';
```

Result



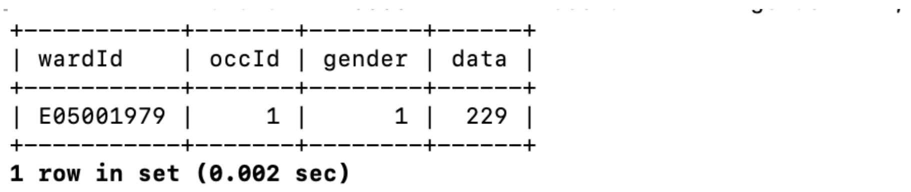
```
+-------+-----------+------------+---------+
| Ward  | code      | Region     | Country |
+-------+-----------+------------+---------+
| Cabot | E05001979 | South West | England |
+-------+-----------+------------+---------+
1 row in set (0.013 sec)
```

Question 3

Find the number of women in occupation class 1 (managers etc.) in the Cabot ward.

```
SELECT SELECT * FROM Statistic
WHERE wardId = 'E05001979' AND occId = 1 AND gender = 1;
```

Result



```
+------------+-------+--------+------+
| wardId     | occId | gender | data |
+------------+-------+--------+------+
| E05001979 |     1 |      1 |  229 |
+------------+-------+--------+------+
1 row in set (0.002 sec)
```

Intermediate SQL query

Question 3

List the names, parties and *percentage* of votes obtained for all candidates in the Southville ward. Order the candidates by percentage of votes obtained descending

- **Subquery for Total Votes**: Inside the SELECT clause, a subquery calculates the total votes in the Southville ward by summing the `votes` from the `Candidate` table, where the candidates are in the Southville ward.

```
SELECT Candidate.name,
Party.name AS party,
(Candidate.votes / SELECT SUM(votes) FROM Candidate JOIN Ward ON Ward.name =
Candidate.ward WHERE Ward.name = 'Southville')) * 100 AS percentage_votes

FROM Candidate
INNER JOIN Ward ON Ward.id = Candidate.ward
INNER JOIN Party ON Party.id = Candidate.party
WHERE ward.name = 'Southville'
ORDER BY percentage_votes DESC;
```

Question 4

How successful (in % of votes cast) was the Conservative party in each ward?

- conservative votes / total votes
- `GROUP BY` ward
- `CASE WHEN...`

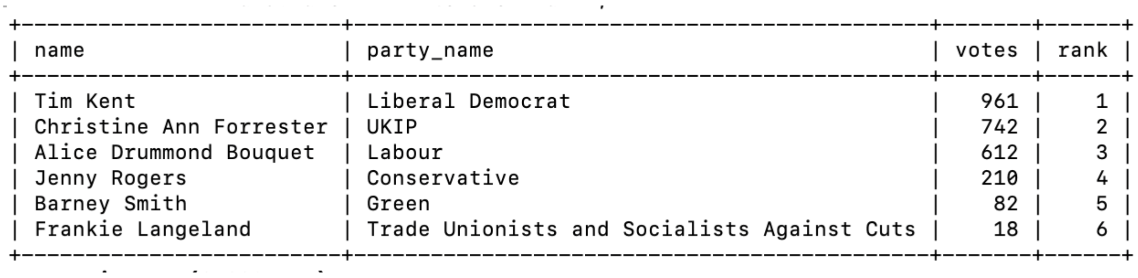
```
SELECT Ward.name AS ward,
SUM(CASE WHEN Party.name = 'Conservative' THEN Candidate.votes ELSE 0 END) AS
conservative_votes,
SUM(Candidate.votes),
SUM(CASE WHEN Party.name = 'Conservative' THEN Candidate.votes ELSE 0
END)/SUM(Candidate.votes) AS percentage_success

FROM Candidate
INNER JOIN Ward ON Ward.id = Candidate.ward
INNER JOIN Party ON Party.id = Candidate.party
GROUP BY Ward.id, Ward.name
```

Question 5

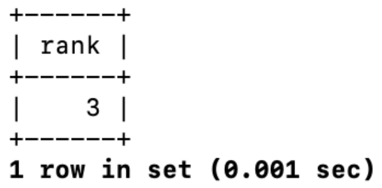
Which rank did Labour end up in the 'Whitchurch Park' ward? Your statement should produce a table with a single row and column containing the answer as a number. You can assume no ties.

- inner query, rank all the candidates' votes in the 'Whitchurch park' ward
  create a table like this



```
+-------------------------+--------------------------------------------+-------+------+
| name                    | party_name                                 | votes | rank |
+-------------------------+--------------------------------------------+-------+------+
| Tim Kent                | Liberal Democrat                           |   961 |    1 |
| Christine Ann Forrester | UKIP                                       |   742 |    2 |
| Alice Drummond Bouquet  | Labour                                     |   612 |    3 |
| Jenny Rogers            | Conservative                               |   210 |    4 |
| Barney Smith            | Green                                      |    82 |    5 |
| Frankie Langeland       | Trade Unionists and Socialists Against Cuts |    18 |    6 |
+-------------------------+--------------------------------------------+-------+------+
```

- outer query, limit to Labour party



```
+------+
| rank |
+------+
|    3 |
+------+
1 row in set (0.001 sec)
```

```
SELECT rank
FROM
(SELECT Candidate.votes,
Candidate.name,
Party.name AS party,
RANK() OVER(ORDER BY Candidate.votes DESC) AS rank
FROM Candidate
INNER JOIN Ward ON Ward.id = Candidate.ward
INNER JOIN Party ON Party.id = Candidate.party
WHERE ward.name = 'Whitchurch Park') AS ranked_candidates

WHERE party = 'Labour';
```

Question 7

Find all wards where the Green party beat Labour and create a table with two columns **ward**, **difference** where the difference column is the number of Green votes minus the number of Labour votes. Your table should be ordered by difference, with the highest one first.

```
SELECT Ward.name AS ward
(SUM(CASE WHEN Party.name = 'Green' THEN Candidate.votes ELSE 0 END)-SUM(CASE
WHEN Party.name = 'Labour' THEN Candidate.votes ELSE 0 END)) AS difference

FROM Candidate
INNER JOIN Ward ON Ward.id = Candidate.ward
INNER JOIN Party ON Party.id = Candidate.party

GROUP BY Ward.id, Ward.name
```

```
HAVING difference > 0
ORDER BY difference
```