1. **Write a query that displays all that is known about Canadian cities.**

SELECT id,  
 name AS 'city\_name',  
 countrycode,  
 district,  
 population  
FROM city  
WHERE countrycode = 'CAN'  
ORDER BY name;

1. **Create a view vw\_Canadian\_cities based on the above query.**

CREATE VIEW vw\_Canadian\_cities AS  
 SELECT id,  
 name AS 'city\_name',  
 countrycode,  
 district,  
 population  
 FROM city  
 WHERE CountryCode = 'CAN'  
 ORDER BY city\_name;

1. **In the top left panel of MySQL Workbench, expand the View menu item of the world database. In the menu bar, click on Query then Refresh to refresh the view and verify that the view vw\_Canadian\_cities is now listed.**

Graphical user interface, text, application, email

Description automatically generated

1. **Using the view from step 2, write a query that displays all that is known about Ottawa.**

SELECT city\_name,  
 countrycode,  
 district,  
 population  
FROM vw\_canadian\_cities  
WHERE city\_name = 'ottawa';

1. **Write an update query on the view from step 2 to update the population of Ottawa to 883391.**

UPDATE vw\_Canadian\_cities  
SET population = '883391'  
WHERE city\_name = 'ottawa';

1. **Rerun the query from Step 4. Is the Population updated?**

Yes

Graphical user interface, text, application, email

Description automatically generated

1. **Create a view called vw\_L5 that displays country codes, city names, country names and independence year of every country whose IndepYear field is not null. Rename the country name field "CountryName". (CREATE VIEW, INNER JOIN..ON, tables name and country). In the top left panel of MySQL Workbench, expand the View menu item of the world database. In the menu bar, click on Query then Refresh to refresh the view and verify that the view vw\_L5 is now listed.**

CREATE VIEW vw\_l5 AS  
 SELECT country.code,  
 city.name AS city\_name,  
 country.name AS countryName,  
 country.indepyear  
 FROM city  
 INNER JOIN country  
 ON city.CountryCode = country.code  
 WHERE IndepYear IS NOT NULL;

Graphical user interface, text

Description automatically generated

1. **Using the view vw\_L5, write a query that lists all distinct CountryNames.**

SELECT DISTINCT countryName  
FROM vw\_l5;

1. **Create a view vw\_L5\_1 based on the above query.**

CREATE VIEW vw\_l5\_1 AS  
 SELECT DISTINCT countryName  
 FROM vw\_l5;

1. **Using view vw\_L5, write a query that reports the number of countries that became independent per year. Rename the number of countries as nCount (GROUP BY)**

SELECT  
 indepyear,  
 COUNT(DISTINCT (countryName)) AS nCount  
FROM  
 vw\_l5  
GROUP BY indepyear;

1. **Create a view vw\_L5\_2 based on the above query.**

CREATE VIEW vw\_l5\_2 AS  
 SELECT indepyear,  
 COUNT(DISTINCT (countryName)) AS 'nCount'  
 FROM vw\_l5  
 GROUP BY indepyear;

1. **Write an update query that updates the view vw\_L5\_2 and sets nCount to 21 for Indepyear 1066. Is the query succesful? Why or why not?**

UPDATE vw\_l5\_2  
SET nCount = '21'  
WHERE indepyear = '1066';

Unsuccessful, the target table is not updatable. We can only manipulate records, not results.

1. **Joining vw\_L5 and CountryLanguage, write a query that lists the countryName along with the languages spoken in each country and their respective percentages. Sort the list by CountryName then by language. Make sure each record occurs only once.**

SELECT vw\_l5.countryName,  
 countrylanguage.language,  
 countrylanguage.percentage  
FROM vw\_l5  
INNER JOIN countrylanguage  
 ON vw\_l5.code = countrylanguage.CountryCode  
ORDER BY vw\_l5.countryName,  
 countrylanguage.Language;

1. **Drop the view vw\_L5;**

DROP VIEW vw\_L5;

1. **Drop the view vw\_L5\_1;**

DROP VIEW vw\_l5\_1;

1. **Drop the view vw\_L5\_2;**

DROP VIEW vw\_L5\_2;

1. **Drop the view vw\_Canadian\_cities;**

DROP VIEW vw\_canadian\_cities;