```sql
-- Kevin Hance
-- DBMS (CPSC 321)
-- HW7 (hw7.sql)

DROP TABLE IF EXISTS border;
DROP TABLE IF EXISTS city;
DROP TABLE IF EXISTS province;
DROP TABLE IF EXISTS country;

CREATE TABLE country(
    country_code VARCHAR(10),
    country_name VARCHAR(50),
    gdp INT UNSIGNED,
    inflation FLOAT,
    PRIMARY KEY (country_code)
);

CREATE TABLE province(
    province_name VARCHAR(50),
    country_code VARCHAR(10),
    area FLOAT UNSIGNED,
    PRIMARY KEY (province_name, country_code),
    FOREIGN KEY (country_code) REFERENCES country (country_code)
);

CREATE TABLE city(
    city_name VARCHAR(50),
    province_name VARCHAR(50),
    country_code VARCHAR(10),
    population INT UNSIGNED,
    PRIMARY KEY (city_name, province_name, country_code),
    FOREIGN KEY (province_name, country_code) REFERENCES province (province_name,
    country_code)
);

CREATE TABLE border(
    country_code_1 VARCHAR(10),
    country_code_2 VARCHAR(10),
    border_length FLOAT UNSIGNED,
    PRIMARY KEY (country_code_1, country_code_2),
    FOREIGN KEY (country_code_1) REFERENCES country (country_code),
    FOREIGN KEY (country_code_2) REFERENCES country (country_code)
);

INSERT INTO country VALUES ('OS', 'Oswaldo', 78000, 6.7);
INSERT INTO country VALUES ('RL', 'Renlandia', 54000, 7.7);
INSERT INTO country VALUES ('GN', 'Geneva', 65000, 2.1);

INSERT INTO province VALUES ('Oslodo', 'OS', 8712);
INSERT INTO province VALUES ('St. Janice', 'OS', 76650);

INSERT INTO province VALUES ('Huport', 'RL', 91030);
INSERT INTO province VALUES ('Flaubury', 'RL', 5690);

INSERT INTO province VALUES ('Antalens', 'GN', 72003);
INSERT INTO province VALUES ('Huport', 'GN', 54041);

INSERT INTO city VALUES ('Tesa', 'Oslodo', 'OS', 964);
INSERT INTO city VALUES ('Sluurgan', 'Oslodo', 'OS', 5919);
INSERT INTO city VALUES ('Slaren', 'Oslodo', 'OS', 590190);
INSERT INTO city VALUES ('Belogonia', 'St. Janice', 'OS', 97635);
INSERT INTO city VALUES ('Britano', 'St. Janice', 'OS', 33434);

INSERT INTO city VALUES ('Piolas', 'Huport', 'RL', 46626);
INSERT INTO city VALUES ('Stombus', 'Huport', 'RL', 49384);
INSERT INTO city VALUES ('Outling', 'Flaubury', 'RL', 127);
INSERT INTO city VALUES ('Whita', 'Flaubury', 'RL', 52743);
INSERT INTO city VALUES ('Gonba', 'Flaubury', 'RL', 53146);

```

```sql
INSERT INTO city VALUES ('Blumore', 'Antalens', 'GN', 5400000);
INSERT INTO city VALUES ('Juefbert', 'Antalens', 'GN', 120);
INSERT INTO city VALUES ('Tesa', 'Huport', 'GN', 4104300);
INSERT INTO city VALUES ('Stombus', 'Huport', 'GN', 1329900);

INSERT INTO border VALUES ('OS', 'GN', 4800);
INSERT INTO border VALUES ('GN', 'RL', 4500);
INSERT INTO border VALUES ('RL', 'OS', 1200);


-- reading question 3

DROP TABLE IF EXISTS border;
DROP TABLE IF EXISTS city;
DROP TABLE IF EXISTS province;
DROP TABLE IF EXISTS country;

CREATE TABLE country(
    country_code VARCHAR(10),
    country_name VARCHAR(50),
    gdp INT UNSIGNED,
    inflation FLOAT,
    PRIMARY KEY (country_code)
);

CREATE TABLE province(
    province_name VARCHAR(50),
    country_code VARCHAR(10),
    area FLOAT UNSIGNED,
    PRIMARY KEY (province_name, country_code),
    FOREIGN KEY (country_code) REFERENCES country (country_code)
);

CREATE TABLE city(
    city_name VARCHAR(50),
    province_name VARCHAR(50),
    country_code VARCHAR(10),
    population INT UNSIGNED,
    PRIMARY KEY (city_name, province_name, country_code),
    FOREIGN KEY (province_name, country_code) REFERENCES province (province_name,
    country_code)
);

CREATE TABLE border(
    country_code_1 VARCHAR(10),
    country_code_2 VARCHAR(10),
    border_length FLOAT UNSIGNED,
    PRIMARY KEY (country_code_1, country_code_2),
    FOREIGN KEY (country_code_1) REFERENCES country (country_code),
    FOREIGN KEY (country_code_2) REFERENCES country (country_code)
);

INSERT INTO country VALUES ('OS', 'Oswaldo', 78000, 6.7);
INSERT INTO country VALUES ('RL', 'Renlandia', 54000, 7.7);
INSERT INTO country VALUES ('GN', 'Geneva', 65000, null);

SELECT count(c.inflation)
FROM country c
GROUP BY country_code
HAVING count(DISTINCT c.country_code) > 0.0;


-- TECHNICAL WORK
-- question 1
SELECT c.country_name, c.country_code, c.GDP, c.inflation, sum(cit.population)
FROM country c JOIN city cit ON cit.country_code = c.country_code
GROUP BY c.country_code;

-- question 2
```

```sql
set @population = 5000000;
SELECT p.country_code, p.province_name, p.area, sum(c.population)
FROM province p JOIN city c ON p.province_name = c.province_name AND p.country_code =
c.country_code
GROUP BY p.province_name
HAVING sum(c.population) > @population;

-- question 3
SELECT c.country_code, c.country_name, count(DISTINCT cit.city_name)
FROM country c JOIN city cit ON c.country_code = cit.country_code
GROUP BY c.country_code
ORDER BY count(DISTINCT cit.city_name) DESC;

-- question 4
SELECT c.country_code, c.country_name, sum(p.area)
FROM country c JOIN province p ON c.country_code = p.country_code
GROUP BY c.country_code
ORDER BY sum(p.area) DESC;

-- question 5
set @min_cities = 5;
set @min_provinces = 1;
SELECT country_name
FROM country c JOIN province p JOIN city cit ON c.country_code = p.country_code AND
p.country_code = cit.country_code
GROUP BY c.country_code
HAVING count(DISTINCT cit.city_name) >= @min_cities AND count(DISTINCT p.province_name)
>= @min_provinces;

-- question 6
set @gdp = 60000;
SELECT c.country_code, c.gdp, sum(p.area)
FROM country c JOIN province p ON c.country_code = p.country_code
GROUP BY c.country_code
HAVING c.gdp >= @gdp
ORDER BY
CASE WHEN sum(p.area) <> 0 THEN sum(p.area) END DESC,
CASE WHEN c.gdp <> 0 THEN c.gdp END DESC;

-- question 7
DROP VIEW IF EXISTS sym_borders;
CREATE VIEW sym_borders AS
SELECT *
FROM border;
INSERT INTO sym_borders
SELECT country_code_2, country_code_1, border_length
FROM border;

SELECT * FROM sym_borders;

-- question 8
-- original
(SELECT c1.country_name, c2.country_name as c_gdp_high
FROM country c1 JOIN country c2 JOIN border bord
ON (bord.country_code_1 = c1.country_code AND bord.country_code_2 = c2.country_code)
WHERE c1.gdp < c2.gdp
AND c1.inflation > c2.inflation)
UNION
(SELECT c2.country_name, c1.country_name as c_gdp_high
FROM country c1 JOIN country c2 JOIN border bord
ON (bord.country_code_1 = c1.country_code AND bord.country_code_2 = c2.country_code)
WHERE c2.gdp < c1.gdp
AND c2.inflation > c1.inflation);
-- new query
SELECT c1.country_name, c2.country_name as c_gdp_high
FROM country c1 JOIN country c2 JOIN sym_borders bord
ON (bord.country_code_1 = c1.country_code AND bord.country_code_2 = c2.country_code)
WHERE (c1.gdp < c2.gdp
AND c1.inflation > c2.inflation)
```

```
203   OR (c2.gdp < c1.gdp
204   AND c2.inflation > c1.inflation);
205
206   -- question 9
207   SELECT c1.country_name, avg(c2.gdp), avg(c2.inflation)
208   FROM country c1 JOIN country c2 JOIN sym_borders b
209   ON b.country_code_1 = c1.country_code
210   AND b.country_code_2 = c2.country_code
211   GROUP BY b.country_code_1
212   ORDER BY
213   CASE WHEN avg(c2.gdp) <> 0 THEN avg(c2.gdp) END ASC,
214   CASE WHEN avg(c2.inflation) <> 0 THEN avg(c2.inflation) END ASC;
215
216   -- question 10
217   -- part 1:  Show all cities which are in a country with a designated relationship between
218   --          the country's inflation and gdp, and those of a bordering country. Should
      return
219   --          the name of the city, name of the province the city is in, and the
      population of the
220   --          city, ordered by population of the city going from highest to lowest.
221   -- first argument: country 1 has a higher gdp, higher inflation
222   SELECT c.city_name, c.province_name, c.population
223   FROM country c1 JOIN country c2 JOIN sym_borders b JOIN city c
224   ON b.country_code_1 = c1.country_code
225   AND b.country_code_2 = c2.country_code
226   AND c.country_code = c1.country_code
227   WHERE c1.gdp > c2.gdp
228   AND c1.inflation > c2.inflation
229   ORDER BY c.population DESC;
230   -- second argument: country 1 has a higher gdp, lower inflation
231   SELECT c.city_name, c.province_name, c.population
232   FROM country c1 JOIN country c2 JOIN sym_borders b JOIN city c
233   ON b.country_code_1 = c1.country_code
234   AND b.country_code_2 = c2.country_code
235   AND c.country_code = c1.country_code
236   WHERE c1.gdp < c2.gdp
237   AND c1.inflation > c2.inflation
238   ORDER BY c.population DESC;
239
240   -- part 2:  Show all provinces with a designated relationshop between the area of that
      province
241   --          and that of a province in a bordering country. Should return the name of
      each province,
242   --          the area of each province, and the names of the countries that border each
      other.
243   -- first argument: province 1 has a higher area
244   SELECT p1.province_name, p2.province_name, p1.area, p2.area, c1.country_code,
      c2.country_code
245   FROM country c1 JOIN country c2 JOIN sym_borders b
246   JOIN province p1 JOIN province p2
247   ON b.country_code_1 = c1.country_code
248   AND b.country_code_2 = c2.country_code
249   AND c1.country_code = p1.country_code
250   AND c2.country_code = p2.country_code
251   WHERE p1.area > p2.area
252   ORDER BY p1.area DESC;
253   -- second argument: province 1 has a lower area
254   SELECT p1.province_name, p2.province_name, p1.area, p2.area, c1.country_code,
      c2.country_code
255   FROM country c1 JOIN country c2 JOIN sym_borders b
256   JOIN province p1 JOIN province p2
257   ON b.country_code_1 = c1.country_code
258   AND b.country_code_2 = c2.country_code
259   AND c1.country_code = p1.country_code
260   AND c2.country_code = p2.country_code
261   WHERE p1.area < p2.area
262   ORDER BY p1.area DESC;
263
```