CSCB20 - Databases and Web Applications

Assignment 1 Part A Bonus: More SQL Queries

Names: Sibo Dong, Dann Sioson
Student Numbers: 1003400269, 1001346377
Instructor: Professor Attarwala
Date: February 9, 2018

1. This question involves the following relations:

```
Classes(class, type, country, numGuns, bore, displacement)
Ships(name, class, launched)
Battles(name, date)
Outcomes(ship, battle, result
```

Find the following using SQL queries, and resulting sample data from figures 1 to 4 on the assignment:

(a) “Give the class names and countries of the classes that carried guns of at least 16 inch bore”

```
SELECT DISTINCT class, country
FROM   classes
WHERE  bore >= 16;
```

Resulting sample data:

class	country
Iowa	USA
North Carolina	USA
Yamato	Japan

(b) “Find the ships launched prior to 1921”

```
SELECT DISTINCT name
FROM   ships
WHERE  launched < 1921;
```

Resulting sample data:

name
Haruna
Hiei
Kirishima
Kongo
Ramillies
Renown
Repulse
Resolution
Revenge
Royal Oak
Royal Sovereign
Tennessee

(c) “Find the ships sunk in the battle of the Denmark Strait”

```
SELECT DISTINCT ship
FROM   outcomes
WHERE  battle = ‘Denmark Strait’
       AND result = ‘sunk’
```

Resulting sample data:

ship
Bismarck
Hood

(d) "The "Treaty of Washington" in 1921 prohibited capital ships heavier than 35,000 tons. List the ships that violated the Treaty of Washington"

```
DROP VIEW IF EXISTS heavy_class;
create view heavy_class AS
SELECT *
FROM   classes
WHERE  displacement > 35000;

SELECT DISTINCT name
FROM   heavy_class
       INNER JOIN
       ships
ON     heavy_class.class = ships.class;
```

Resulting sample data:

name
Iowa
Missouri
Musashi
New Jersey
North Carolina
Washington
Wisconsin
Yamato

(e) "List the name, displacement and number of guns of the ships engaged in the battle of Guadalcanal"

```
DROP VIEW IF EXISTS guada_fighters;
CREATE VIEW guada_fighters AS
SELECT ship, battle, result
FROM   outcomes
WHERE  battle = ‘Guadalcanal’;

DROP VIEW IF EXISTS guada_ships;
CREATE VIEW guada_ships AS
SELECT name, class
FROM   guada_fighters
       INNER JOIN
       ships
ON     ships.name = guada_fighters.ship;

SELECT name, displacement, numGuns
```

```
FROM   classes
       INNER JOIN
       guada_ships
ON     classes.class = guada_ships.class;
```

Resulting sample data:

ship	displacement	numGuns
Kirishima	32000	8

(f) “List all the capital ships mentioned in the database. (Remember that all these ships may not appear in the Ship relation.)”

```
SELECT DISTINCT ship_name
FROM ((SELECT ship AS ship_name
       FROM   outcomes)
     UNION
     (SELECT name AS ship_name
      FROM   ships)) as all_ships;
```

Resulting sample data (on the next page):

ship_name
Arizona
Bismarck
California
Duke of York
Fuso
Hood
Iowa
King George V
Kirishima
Prince of Wales
Rodney
Scharnhorst
South Dakota
Tennessee
Washington
West Virginia
Yamashiro
Haruna
Hiei
Kongo
Missouri
Musashi
New Jersey
North Carolina
Ramillies
Renown
Repulse
Resolution
Revenge
Royal Oak
Royal Sovereign
Wisconsin
Yamato

(g) “Find the classes that had only one ship as a member of that class”

```
DROP VIEW IF EXISTS mt_one;
CREATE VIEW mt_one AS
SELECT DISTINCT ships1.class
FROM   ships AS ships1,
       ships AS ships2
WHERE ships1.name != ships2.name
      AND ships1.class = ships2.class;

SELECT DISTINCT ships.class
FROM   ships
       LEFT JOIN
       mt_one
ON     ships.class = mt_one.class
WHERE  mt_one.class IS NULL;
```

Resulting sample data (all classes of the sample data had more than one ship of that class):

class

(h) “Find those countries that had both battleships and battle cruisers”

```
SELECT DISTINCT relevant.country
FROM   classes AS relevant,
       classes AS ship_checker
WHERE  relevant.country = ship_checker.country
       AND relevant.type = ‘bb’
       AND ship_checker.type = ‘bc’;
```

Resulting sample data:

country
Japan
Gt. Britain

(i) “Find those ships that “lived to fight another day”; they were damaged in one battle, but later fought in another”

```
DROP VIEW IF EXISTS survival;
CREATE VIEW survival AS
SELECT *
FROM   outcomes
       INNER JOIN
       battles
ON     outcomes.battle = battles.name;

SELECT survival.ship
FROM   survival, survival AS stronger
WHERE survival.ship = stronger.ship
  AND survival.result = ’damaged’
      AND survival.date < stronger.date;
```

With the sample data given, it appears that there are only two different ships that were damaged, but both of these ships did not fight in other (future) battles, so we have an empty table.

ship