

CSC343 Worksheet 7

June 22, 2020

1. **Exercise 9.3.1:** Write the following embedded SQL queries, based on the database schema

```
1 Product(maker, model, type)
2 PC(model, speed, ram, hd, price)
3 Laptop(model, speed, ram, hd, screen, price)
4 Printer(model, color, type, price)
5
```

of Exercise 2.4.1. You may use any host language with which you are familiar, and details of host-language programming may be replaced by clear comments if you wish.

- a) Ask the user for a price and find the PC whose price is closest to the desired price. Print the maker, model number, and speed of the PC.
- b) Ask the user for minimum values of the speed, RAM, hard-disk size, and screen size that they will accept. Find all the laptops that satisfy these requirements. Print their specifications (all attributes of Laptop) and their manufacturer.
- c) Ask the user for a manufacturer. Print the specifications of all products by that manufacturer. That is, print the model number, product-type, and all the attributes of whichever relation is appropriate for that type.
- d)
- e) Ask the user for a manufacturer, model number, speed, RAM, hard-disk size, and price of a new PC. Check that there is no PC with that model number. Print a warning if so, and otherwise insert the information into tables Product and PC.

2. **Exercise 9.3.2:** Write the following embedded SQL queries, based on the database schema

```
1 Classes(class, type, country, numGuns, bore, displacement)
2 Ships(name, class, launched)
3 Battles(name, date)
4 Outcomes(ship, battle, result)
5
```

of Exercise 2.4.3.

- The firepower of a ship is roughly proportional to the number of guns times the cube of the bore of the guns. Find the class with the largest firepower.
- Ask the user for the name of a battle. Find the countries of the ships involved in the battle. Print the country with the most ships sunk and the country with the most ships damaged.
- Ask the user for the name of a class and the other information required for a tuple of table Classes. Then ask for a list of the names of the ships of that class and their dates launched. However, the user need not give the first name, which will be the name of the class. Insert the information gathered into Classes and Ships.
- Examine the Battles , Outcomes, and Ships relations for ships that were in battle before they were launched. Prompt the user when there is an error found, offering the option to change the date of launch or the date of the battle. Make whichever change is requested.

3. **Exercise 9.4.1:** Using our running movie database:

```

1  Movies(title, year, length, genre, studioName, producerC#)
2  StarsIn(movieTitle, movieYear, starName)
3  MovieStar(name, address, gender, birthdate)
4  MovieExec(name, address, cert#, netWorth)
5  Studio(name, address, presC#)
6

```

write PSM procedures or functions to perform the following tasks:

- Given the name of a movie studio, produce the net worth of its president.
- Given a name and address, return 1 if the person is a movie star but not an executive, 2 if the person is an executive but not a star, 3 if both, and 4 if neither.
- Given a studio name, assign to output parameters the titles of the two longest movies by that studio. Assign NULL to one or both parameters if there is no such movie (e.g., if there is only one movie by a studio, there is no “second-longest”).
- Given a star name, find the earliest (lowest year) movie of more than 120 minutes length in which they appeared. If there is no such movie, return the year 0.
- Given an address, find the name of the unique star with that address if there is exactly one, and return NULL if there is none or more than one.
- Given the name of a star, delete them from MovieStar and delete all their movies from StarsIn and Movies.

4. **Exercise 9.4.2:** Write the following PSM functions or procedures, based on the database schema

```

1  Product(maker, model, type)
2  PC(model, speed, ram, hd, price)
3  Laptop(model, speed, ram, hd, screen, price)
4  Printer(model, color, type, price)
5

```

of Exercise 2.4.1.

- a) Take a price as argument and return the model number of the PC whose price is closest.
- b) Take a maker and model as arguments, and return the price of whatever type of product that model is.
- c) Take model, speed, ram, hard-disk, and price information as arguments, and insert this information into the relation PC. However, if there is already a PC with that model number (tell by assuming that violation of a key constraint on insertion will raise an exception with SQLSTATE equal to '23000'), then keep adding 1 to the model number until you find a model number that is not already a PC model number.
- d) Given a price, produce the number of PC 's, the number of laptops, and the number of printers selling for more than that price.

5. **Exercise 9.4.3:** Write the following PSM functions or procedures, based on the database schema

```

1  Classes(class, type, country, numGuns, bore, displacement)
2  Ships(name, class, launched)
3  Battles(name, date)
4  Outcomes(ship, battle, result)
5

```

of Exercise 2.4.3.

- a) The firepower of a ship is roughly proportional to the number of guns times the cube of the bore. Given a class, find its firepower.
- b) Given the name of a battle, produce the two countries whose ships were involved in the battle. If there are more or fewer than two countries involved, produce NULL for both countries.
- c) Take as arguments a new class name, type, country, number of guns, bore, and displacement. Add this information to C lasses and also add the ship with the class name to Ships.
- d) Given a ship name, determine if the ship was in a battle with a date before the ship was launched. If so, set the date of the battle and the date the ship was launched to 0