Assignment 3

Dirk Van Gucht

In this assignment, you will practice working with SQL. Your solutions, containing PostgreSQL statements for solving the problems, should be submitted to IUCanvas in a file Assignment3.sql. It is advised that you also include comments in this file to elaborate on your solutions.

1. Let $A(x)$ be the relation schema for a set of positive integers.

   Write a SQL statement that produces a table which, for each $x \in A$, lists the tuple $(x, \sqrt{x}, x^2, 2^x, x!, \ln x)$.

   For example, if $A = \{1, 2, 3, 4, 5\}$ then your SQL statement should produce the following table:

```
   x |  square_root_x   | x_squared | two_to_the_power_x | x_factorial |    logarithm_x
---+------------------+-----------+--------------------+-------------+-------------------
   1 |                1 |         1 |                  2 |           1 |                 0
   2 |  1.4142135623731 |         4 |                  4 |           2 | 0.693147180559945
   3 | 1.73205080756888 |         9 |                  8 |           6 |  1.09861228866811
   4 |                2 |        16 |                 16 |          24 |  1.38629436111989
   5 | 2.23606797749979 |        25 |                 32 |         120 |   1.6094379124341
( 5 rows)
```

2. Let $A(x)$ and $B(x)$ be two unary relation schemas that represent sets A and B.

   Determine the truth-value of $A - B$ is empty, $(A - B) \cup (B - A)$ is not empty, and $A \cap B$ is empty.

   For example, if $A = \{1, 2, 3\}$ and $B = \{1, 3, 4, 5\}$ then because

$$\begin{aligned} A - B &= \{2\} \\ (A - B) \cup (B - A) &= \{2, 4, 5\} \\ A \cap B &= \{1, 3\} \end{aligned}$$

   your SQL statement should produce the output:

```
 empty_a_minus_b | not_empty_symmetric_difference | empty_a_intersection_b
-----------------+--------------------------------+------------------------
 f               | t                              | f
(1 row)
```

Your solution should work for arbitrary A and B.

3. Let `Pair`(x, y) be a relation of pairs (x, y). (The domain of x and y is INTEGER.) Write a SQL query that produces a relation with attributes (x_1, y_1, x_2, y_2) such that (1) (x_1, y_1) and (x_2, y_2) are different pairs in the relation `Pair`, and (2) $x_1 + y_1 = x_2 + y_2$.

4. SQL uses 3-valued logic where it concerns the treatment of `NULL` values. (Read your textbook or search the web for relevant information.) Consider 3 unary relation schemas `p(value)`, `q(value)`, and `r(value)` where the type of the attribute `value` is `boolean`. Populate each of these 3 unary relations with the values `true`, `false`, and `NULL`.

   Write a SQL statement that generates the 3-valued truth table for the Propositional Logic statement

   $$\neg(\neg p \vee q) \vee r.$$

   Your statement should return the following answer:

```
p     |q     | r     | not(not(p) or q) or r
------+------+-------+----------
 t    | t    | t     | t
 t    | t    | f     | f
 t    | t    |       |
 t    | f    | t     | t
 t    | f    | f     | t
 t    | f    |       | t
 t    |      | t     | t
 t    |      | f     |
 t    |      |       |
 f    | t    | t     | t
 f    | t    | f     | f
 f    | t    |       |
 f    | f    | t     | t
 f    | f    | f     | f
 f    | f    |       |
 f    |      | t     | t
 f    |      | f     | f
 f    |      |       |
      | t    | t     | t
      | t    | f     | f
      | t    |       |
      | f    | t     | t
      | f    | f     |
      | f    |       |
      |      | t     | t
      |      | f     |
      |      |       |
(27 rows)
```

   The blank characters in this table represent the `NULL` (unknown) value.

5. Let $A(x)$, $B(x)$ and $C(x)$ be three unary relation schemas that represent sets A, B and C of integers.

   Give answers to the following problems. You should provide two different SQL queries. One answer wherein you can use the set operations INTERSECT and/or EXCEPT, and a second answer wherein you can not use these operators.[1] You can use user-defined functions but you can not use aggregate functions.

   (a) Determine the truth-value of $A \cap B \neq \emptyset$. For example, if $A = \{1, 2\}$ and $B = \{1, 4, 5\}$ then the result of your SQL statements should be

   ```
    answer
   --------
    t
   (1 row)
   ```

   If, however, $A = \{1, 2\}$ and $B = \{3, 4\}$ then the result of your statement should be

   ```
    answer
   --------
    f
   (1 row)
   ```

   (b) Determine the truth-value of $A \subseteq B$.

   (c) Determine the truth-value of $A \cap B = B$.

   (d) Determine the truth-value of $A \neq B$.

   (e) Determine the truth-value of $|A \cap B| \leq 2$.

   (f) Determine the truth-value of $(A \cup B) \subseteq C$.

   (g) Determine the truth-value of $|(A - B) \cup (B - C)| = 1$.

6. Repeat Problem 5 by using the COUNT aggregate function. For each sub-problem, you only need to provide one answer and you are allowed to use the set operations UNION, INTERSECT, and EXCEPT.

[1] You are permitted to use the UNION operator.

7. Let $W(A, B)$ be a relation schema. The domain of A is INTEGER and the domain of B is VARCHAR(5).

   Write a SQL query with returns the A-values of tuples in W if A is a primary key of W. Otherwise, i.e., if A is not a primary key, then your query should return the A-values of tuples in W for which the primary key property is violated. (In this query you should consider creating views for intermediate results.)

   For example, consider the following relation instance for W:

W

A	B
1	John
2	Ellen
3	Ann

   Then your query should return the following answer since, in this case, A satisfies the primary property for W.

```
a
---
1
2
3
(3 rows)
```

   However, if we have the following relation instance for W

W

A	B
1	John
2	Ellen
2	Linda
3	Ann
4	Ann
4	Nick
4	Vince
4	Lisa

   then your query should return the following answer because the primary key property of A for W is violated for the A-values 2 and 4.

```
a
---
2
4
(2 rows)
```

8. Use the same files student.txt, majors.txt, book.txt, and buys.txt from Assignment 2.

   Consider the following relation schemas about students and books.

   Student($Sid, Sname$)
   Major($Sid, Major$)
   Book($BookNo, Title, Price$)
   Buys($Sid, BookNo$)

   The relation Major stores students and their majors. A student can have multiple majors but we also allow that a student has no major.

   Assume the following domains for the attributes:

Attribute	Domain
Sid	INTEGER
Sname	VARCHAR(15)
Major	VARCHAR(15)
BookNo	INTEGER
Title	VARCHAR(30)
Price	INTEGER

   (a) i. Write a function

```
booksBoughtbyStudent(sid int)
    returns table(bookno int, title VARCHAR(30), price int)
```

   that takes a student sid as input and returns the book information of books bought by that student.

   ii. Test this function on the student with sid 1001 and on the student with sid 1015.

   iii. Using this function, write the following queries:

   A. Find the sids and names of students who bought exactly one book that cost less than $50.

   B. Find the pairs of different student sids (s1,s2) such that student s1 and student s2 bought the same books.

   (b) i. Write a function

```
studentsWhoBoughtBook(bookno int)
    returns table(sid int, sname VARCHAR(15))
```

   that takes a bookno as input and returns the student information of students who bought that book.

ii. Test your function on the book with bookno 2001 and that with bookno 2010.

iii. Using this function and the `booksBoughtbyStudent` function from problem 8(a)i write the query "Find the booknos of books bought by a least two CS students who each bought at least one book that cost more that $30."

(c) Write the following queries in SQL by using aggregate functions and user-defined functions. You can not use the EXISTS and NOT EXISTS predicates.

i. Find the sid and major of each student who bought at least 4 books that cost more than $30.

ii. Find the pairs (s_1, s_2) of different students who spent the same amount of money on the books they bought.

iii. Find the sid and name of each student who spent more money on the books he or she bought than the average cost that was spent on books by students who major in 'CS'.

iv. Find the booknos and titles of the third most expensive books.

v. Find the bookno and title of each book that is only bought by students who major in 'CS'.

vi. Find the sid and name of each student who did not only buy books that were bought by at least two 'CS' students.

vii. Find each (s, b) pair where s is the sid of a student and where b is the bookno of a book bought by that student whose price is strictly below the average price of the books bought by that student.

viii. Find each pair (s_1, s_2) where s_1 and s_2 are the sids of different students who have a common major and who bought the same number of books.

ix. Find the triples (s_1, s_2, n) where s_1 and s_2 are the sids of two students who share a major and where n is the number of books that was bought by student s_1 but not by student s_2.

x. Find the bookno of each book that was bought buy all-but-one student who majors in 'CS'.

9. Consider a database with the following relations:

Student(sid int, sname text)	sid is primary key
Course(cno int, cname text, total int, max int)	cno is primary key total is the number of students enrolled in course cno max is the maximum permitted enrollment for course cno
Prerequisite(cno int, prereq int)	cno is foreign key referencing Course prereq is foreign key referencing Course course cno has as a prerequisite course prereq
HasTaken(sid int,cno int)	sid foreign key referencing Student cno foreign key referencing Course student sid has taken course cno
Enroll(sid int,cno int)	sid foreign key referencing Student cno foreign key referencing Course student sid is enrolled in course cno
Waitlist(sid int,cno int, position int)	sid foreign key referencing Student cno foreign key referencing Course position is the place of student sid on the waitlist to enroll in course cno

Develop appropriate triggers to permit (1) inserts and deletes in the Enroll relation and (2) deletes in the Waitlist governed by the following constraints:

- A student can only enroll in a course if he or she has taken all the prerequisites for that course. If the enrollment succeeds, the total enrollment for that course needs to be incremented by 1.
- A student can only enroll in a course if his or her enrollment does not exceed the maximum enrollment for that course. However, the student must then be placed at the next available position on the waitlist for that course.
- A student can drop a course. When this happens and if there are students on the waitlist for that course, then the student who is at the first position gets enrolled and removed from the waitlist. If there are no students on the waitlist, then the total enrollment for that course needs to decrease by 1.
- A student may remove himself or herself from the waitlist for a course. When this happens, the positions of the other students who are waitlisted for that course need to be adjusted.