COMP9120 Relational Database Systems

Tutorial Week 5: Complex SQL and NULL Values

Exercise 1. Three-Valued Logic in SQL

Consider a RDBMS table R with two attributes a and b, both of which are integer valued and may be NULL in some tuples. For each of the following conditions (as may appear in a WHERE clause), describe exactly the set of (a, b) tuples that satisfy the condition, including the case where a and/or b is NULL.

a) a = 10

b) a = 10 OR b = 20

c) a = 10 AND b = 20

d) a < 10 AND NOT b = 20

You can verify your answers by actually issuing SQL queries on the database table created by the SQL file available in Canvas.

(Use pgAdmin to connect to PostgreSQL server and run the SQL file to create the table R and populate it with tuples.)

Exercise 2. Grouping and Nested SQL Queries

Download the University schema from Canvas.

Use pgAdmin to connect to PostgreSQL server and run the SQL file to create all the tables and populate them with data. You might need to refresh your client after running the script to see the created tables (eg: right click on Tables node and click on Refresh). Then try writing queries to answer the following questions based on this university schema:

a) Which lecturers (by id and name) have taught both 'INFO2120' and 'INFO3404'? Write a SQL query to answer this question using a SET operator.

b) Which lecturers (by id and name) have taught both 'INFO2120' and 'INFO3404'? Answer this using a sub-query without SET operators. Make sure your result doesn't include duplicates.

c) Write a SQL query to give the **student IDs** of all students who have enrolled in only one lecture using GROUP BY, and order the result by student ID. A lecture is a unit_of_study in a semester of a year.

d) Write a SQL query to give the **names** of all students who have enrolled in only one lecture using a sub-query. A lecture is a unit_of_study in a semester of a year.

e) Write a SQL query to give the **student IDs** and **names** of all students who have enrolled in only one lecture **without** using a sub-query, and order the result by student ID. A lecture is a unit_of_study in a semester of a year.

f) Write a SQL query to give the **names** of all students who have enrolled in only one lecture **without** using a sub-query. A lecture is a unit_of_study in a semester of a year.

g) [Advanced, Optional] Write a SQL query to give the **student IDs** of all students who have enrolled in only one **unit_of_study**, and order the result by student ID. Note that, a student can enrol in the same unit_of_study multiple times, which is still counted as one unit_of_study.

h) [Advanced, Optional] Write a SQL query to give the **student IDs** and **names** of all students who have enrolled in only one **unit_of_study**, and order the result by student ID. Note that, a student can enrol in the same unit_of_study multiple times, which is still counted as one unit_of_study.