COMP9120 Relational Database Systems

Tutorial Week 5 Solution: 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

Answer: All tuples with a being 10 combined with any value for b, including NULL. Examples: (10, 0), (10, 1), ..., (10,-1), ..., (10, NULL)

b) a = 10 OR b = 20

Answer: All tuples where either a is 10, while b takes any value (incl. NULL);

Examples: (10, 0), (10, 1), ..., (10, -1), ..., (10, NULL)

Or b is 20, while a can take any value including NULL.

Examples: (0, 20), (1, 20), ..., (-1, 20), ..., (NULL, 20)

c) a = 10 AND b = 20

Answer: All tuples where a is 10 and b is 20: (10, 20)

d) a < 10 AND NOT b = 20

Answer: Similar to the previous answer: All tuples where a < 10 and not NULL, and b ≠ 20 and also not NULL.

Exercise 2. Grouping and Nested SQL Queries

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.

Answer :

```
SELECT id, name
FROM AcademicStaff JOIN UoSOffering ON id=instructorId
WHERE uosCode = 'INFO2120'
INTERSECT
SELECT id, name
FROM AcademicStaff JOIN UoSOffering ON id=instructorId
WHERE uosCode = 'INFO3404';
```

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.

Answer :

```
SELECT DISTINCT id, name
FROM AcademicStaff JOIN UoSOffering ON id=instructorId
WHERE uosCode = 'INFO2120'
AND id IN ( SELECT instructorId
 FROM UoSOffering
 WHERE uosCode = 'INFO3404' );
```

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.

Answer:

```
SELECT studId
FROM Transcript
GROUP BY studId
HAVING count(*) = 1
ORDER BY studId;
```

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.

Answer :

```
SELECT name FROM Student
Where studId IN (
```

```
SELECT studId
FROM Transcript
GROUP BY studId
HAVING count(*) = 1 );
```

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.

Answer :

```
SELECT studId, name
FROM Student NATURAL JOIN Transcript
GROUP BY studId, name
HAVING count(*) = 1
ORDER BY studId;
```

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.

Answer :

```
SELECT name
FROM Student NATURAL JOIN Transcript
GROUP BY studId, name
HAVING count(*) = 1;
```

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.

Answer :

```
SELECT studId
FROM Transcript
GROUP BY studId
HAVING count(DISTINCT uoSCode) = 1
ORDER BY studId;
```

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.

Answer :

```
SELECT studId, name
```

```
FROM Student NATURAL JOIN (
 SELECT DISTINCT studId, uoSCode
 FROM Transcript ) AS T
GROUP BY studId, name
HAVING count(*) = 1
ORDER BY studId;
```