COMP9120 Relational Database Systems **Semester 2, 2016**

Tutorial Week 6 Solution: Complex SQL and NULL Values

Exercise 1. Three-Valued Logic in SQL

Let a and b be integer-valued attributes that 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 = 10 (but not NULL) 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 = 10 and not NULL, while b takes any value (incl. NULL);

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

Or b = 20 and not NULL, 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 = 10 and b = 20 and none of them is NULL: (10, 20)

d) a < 10 AND NOT b = 20

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

Exercise 2. Grouping and Nested SQL Queries

Consider the University relational database schema from Week 5. You can find the University Schema sql file under Schemas on the left-hand side navigation bar on eLearning. If you haven't done so already, create this schema by running all the sql statements in the downloaded file on Oracle. Then try writing queries to answer the following questions based on this university schema:

a) Which lecturers (by ID and name) have already 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 instructorId) have already taught both 'INFO2120' and 'INFO3404'. Answer this using a sub-query. Make sure your result doesn't include duplicates.

Answer :

```
SELECT DISTINCT instructorId
FROM UoSOffering
WHERE uosCode = 'INFO2120'
AND instructorId IN ( SELECT instructorId
FROM UoSOffering
WHERE uosCode = 'INFO3404' );
```

c) Write a SQL query to give the SIDs of all students who have enrolled in only one lecture using a set operator.

Answer :

```
SELECT studId FROM Lecture NATURAL JOIN Transcript
MINUS
SELECT studId
FROM Lecture NATURAL JOIN Transcript
GROUP BY (studId)
HAVING count(*) <> 1;
(You might also use the "EXCEPT" keyword instead of "MINUS" but
Oracle only allows MINUS)
```

d) Write a SQL query to give the SIDs of all students who have enrolled in only one lecture using a sub-query.

Answer :

```
SELECT studId FROM Lecture NATURAL JOIN Transcript
WHERE studId NOT IN(
 SELECT studId
 FROM Lecture NATURAL JOIN Transcript
 GROUP BY (studId)
 HAVING count(*) <> 1
);
```

e) Write a SQL query to give the SIDs of all students who have enrolled in only one lecture using GROUP BY.

Answer:

```
And we can obviously forego the need of MINUS or NOT IN, and ONLY
use a Group BY clause as follows:
SELECT studId
FROM Lecture NATURAL JOIN Transcript
GROUP BY (studId)
HAVING count(*) = 1;
```