CS386 Homework 3

Russell Miller

February 1, 2011

Part 1

Turn in the SQL query, the first five rows of the query answer, and the number of rows in your query answer.

1. **Using the spy database, write an SQL query to find the names of teams who have failed exactly two missions.**
   SELECT team.name
   FROM mission,team
   WHERE mission.team_id=team.team_id
   AND mission.mission_status='failed'
   GROUP BY team.name
   HAVING COUNT(*) = 2;

name
Roadkill
Vikings
Blaster
Failsafe
ShowBiz

   13 rows.

2. **Using the library database, list the authors that have published the largest number of books.**
   SELECT author.firstname,author.middlename,author.lastname
   FROM author,relauth
   WHERE author.authorid=relauth.authorid
   GROUP BY author.firstname,author.middlename,author.lastname
   HAVING COUNT(*) = (SELECT max(COUNT)
                   FROM (SELECT COUNT(*)
                           FROM author,relauth,book_description
                           WHERE author.authorid=relauth.authorid
                           AND relauth.bookdescid=book_description.bookdescid
                           GROUP BY author.authorid) AS totals);

firstname	middlename	lastname
DEMETRIOS	G	LAINIOTIS
KRITHI		RAMAMRITHA
D.		BAKER
THOMAS	C	MCINTIRE
JOHN		MCCONNELL

   5 rows.

3. **Using the spy databse, are the following two SQL qeueries equivalent? Describe the two queries in English and explain how they are different.**

No. They are different queries. (a) uses a natural join on the Team table and the Mission table. This means equality is specified on all fields with the same name. Since Team and Mission both have a 'name' column, the two would have to be equal in order for the row to show up in the query answer. (a) lists the successful missions where the team name is the same as the mission name. (b) lists the successful missions for each team, along with the team name.

4. **List the agents who were assigned to at least one 'failed' mission. Write this query in two versions, one using EXISTS and another using IN.**

SELECT a.first,a.last
FROM agent a
WHERE EXISTS (SELECT DISTINCT agent.agent_id
FROM agent,teamrel,team,mission
WHERE agent.agent_id=teamrel.agent_id
AND teamrel.team_id=team.team_id
AND team.team_id=mission.team_id
AND mission_status='failed'
AND agent.agent_id=a.agent_id
GROUP BY agent.agent_id
HAVING COUNT(*) ¿ 0);

first	last
Nick	Black
Mathew	Cohen
Jim	Cowan
George	Fairley
George	Jones

221 rows.

SELECT a.first,a.last
FROM agent a
WHERE a.agent_id IN (SELECT DISTINCT agent.agent_id
FROM agent,teamrel,team,mission
WHERE agent.agent_id=teamrel.agent_id
AND teamrel.team_id=team.team_id
AND team.team_id=mission.team_id
AND mission_status='failed'
GROUP BY agent.agent_id
HAVING COUNT(*) > 0);

first	last
Nick	Black
Mathew	Cohen
Jim	Cowan
George	Fairley
George	Jones

221 rows.

5. **List the agents that have skills of 'Firearms' or 'Biologist' using the IN clause.**
SELECT a.first,a.last
FROM agent a
WHERE a.agent_id IN (SELECT agent.agent_id
                     FROM agent,skillrel,skill
                     WHERE agent.agent_id=skillrel.agent_id
                     AND skillrel.skill_id=skill.skill_id
                     AND (skill.skill='Firearms' OR skill.skill='Biologist'));

first	last
Nick	Black
Chris	Leen
Kristin	Moody
Nick	Steere
Pete	Consel

84 rows.

6. **Rewrite the query described in question 5 without using IN.**
SELECT agent.first,agent.last
FROM agent,skillrel,skill
WHERE agent.agent_id=skillrel.agent_id
AND skillrel.skill_id=skill.skill_id
AND (skill.skill='Firearms' or skill.skill='Biologist');

first	last
Nick	Steere
Helen	Hermansky
Pete	Pickering
Robert	Smith
George	Berkling

84 rows.

7. **Write a relational algebra query that is equivalent to the query described in question 5.**

$\pi_{agent.first,agent.last}($
$agent \bowtie_{agent.agent\_id=skillrel.agent\_id}($
$skillrel \bowtie_{skillrel.skill\_id=skill.skill\_id}($
$\sigma_{skill.skill='Firearms' \vee skill.skill='Biologist'} skill)))$

Part 2

Outer joins - Write two responses for each question, one with LEFT OUTER JOIN, and the second with RIGHT OUTER JOIN.

1. **Write a query that counts the number of agents that are not on a team.**
LEFT :
SELECT q.first,q.last
FROM (SELECT a.agent_id,a.first,a.last,t.team_id
      FROM agent a LEFT OUTER JOIN teamrel t
      ON a.agent_id=t.agent_id) AS q
                          WHERE q.team_id IS NULL;

first	last
Bill	Bundt
Bill	Heeman
Andrew	James
Kristin	Delcambre
John	Johnston

368 rows.

RIGHT :
SELECT q.first,q.last
FROM (SELECT a.agent_id,a.first,a.last,t.team_id
      FROM teamrel t RIGHT OUTER JOIN agent a
      ON a.agent_id=t.agent_id) AS q
                WHERE q.team_id IS NULL;

first	last
Bill	Bundt
Bill	Heeman
Andrew	James
Kristin	Delcambre
John	Johnston

368 rows.

2. **Write a query that lists teams that do not have agents.**
LEFT :
SELECT q.name
FROM (SELECT t.team_id,t.name,r.agent_id
      FROM team t LEFT OUTER JOIN teamrel r
      ON t.team_id=r.team_id) AS q
                WHERE q.agent_id IS NULL;

0 rows.

RIGHT :
SELECT q.name
FROM (SELECT t.team_id,t.name,r.agent_id
      FROM teamrel r RIGHT OUTER JOIN team t
      ON t.team_id=r.team_id) AS q
                WHERE q.agent_id IS NULL;

0 rows.

Part 3

Relational Algebra - For each exercise, describe in English what the expression performs and write the equivalent SQL query, the first five rows of the query answer, and the number of rows in your query answer.

1. $\pi_{team\_id}(Mission - (\sigma_{mission_status='failed'} Mission))$