Name: 
UFID: 
Email Address: 

Exam 2 Part 1 Solutions

Pledge (Must be signed according to UF Honor Code)

On my honor, I have neither given nor received unauthorized aid in doing this assignment.

_______________________________________________
Signature

For scoring use only:

<table>
<thead>
<tr>
<th>Question</th>
<th>Maximum</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>
Question 1 (SQL)  [20 points]

The database schema below provides support for a library management system. Retrieve the required information below by SQL queries. [5 points each]

1. How many copies of the book titled “The Lost Tribe” are owned by the library branch whose name is “Sharpstown”?

   ```sql
   SELECT BC.numCopies
   FROM Book B, BookCopy BC, LibraryBranch LB
   WHERE B.bookId = BC.bookId
   AND BC.branchId = LB.branchId
   AND title = 'The Lost Tribe'
   AND branchName = 'Sharpstown';
   ```

2. For each book that is loaned out from the “Sharpstown” branch and whose DueDate is today, retrieve the book title, the borrower’s name, and the borrower’s address.

   ```sql
   SELECT B.Title, R.Name, R.Address
   FROM Book B, Borrower R, BookLoan BL, LibraryBranch LB
   WHERE LB.branchName = 'Sharpstown'
   AND LB.branchId = BL.branchId
   AND BL.dueDate = 'today'
   AND BL.cardNo = R.cardNo
   AND BL.bookId = B.bookId;
   ```
3. For each library branch, retrieve the branch name and the total number of books loaned out from that branch.

```sql
SELECT L.branchName, COUNT(*)
FROM LibraryBranch L, BookLoan BL
WHERE BL.branchId = L.branchId
GROUP BY L.branchName;
```

4. Retrieve the names, address, and number of books checked out for all borrowers who have checked out more than five books.

```sql
SELECT B.Name, B.Address, COUNT(*)
FROM Borrower B, BookLoad L
WHERE B.cardNo = L.cardNo
GROUP BY B.cardNo, B.name, B.address
HAVING COUNT(*) > 5;
```
Question 2 (Relational Algebra and SQL) [30 points]

Given the following schema:

Person (SSN, name, address)
Car (license, year, model)
Accident (license, accidentDate, driverSSN, damageAmount)
Owns (SSN, license)

Primary keys are underlined. Note that the driver involved in a car accident may not always be the owner of the car. We assume that a car cannot get involved in more than one accident at a certain date.

1. For each query A to F below, write one sentence that describes what it does. Which of these queries returns the SSN of every person who owns one or more cars, none of which has ever been involved in a car accident? Choose all that apply. [6 points]

A. SELECT DISTINCT O.SSN
   FROM Owns O
   WHERE O.license NOT IN (SELECT A.license
                           FROM Accident A);

B. SELECT DISTINCT O.SSN
   FROM Owns O
   WHERE NOT EXISTS (SELECT *
                      FROM Accident A
                      WHERE A.license = O.license);

C. SELECT DISTINCT O.SSN
   FROM Owns O, Accident A
   WHERE O.license <> A.license;

D. SELECT DISTINCT O.SSN
   FROM Owns O, Accident A
   WHERE O.license = A.license
   GROUP BY O.SSN
   HAVING COUNT (DISTINCT accidentDate) = 0;

E. πSSN(Owns) – πSSN(Owns ∘ Accident)

F. πSSN, license(Owns ∘ Accident) ÷ πlicense(Owns)

G. None of the above

Solution: E.

A and B return the SSN of every person that has at least one car that is not involved in an
accident. C returns all car owners if they have a car not involving in every existent accident. D never returns anything--the inner join eliminates all owners and licenses not in an accident. A left outer join should be used. F finds the SSN of all people who have had all of their cars involved in an accident.

For questions 2-5, write the relational algebra query and the SQL query. [6 points each]

2. Find the SSN of every person who owns a TOYOTA and a DODGE.

\[ \pi_{\text{SSN}} (\text{Ow} \bowtie \sigma_{\text{model} = \text{TOYOTA}} (\text{Car})) \cap \pi_{\text{SSN}} (\text{Ow} \bowtie \sigma_{\text{model} = \text{DODGE}} (\text{Car})) \]

\[
\text{SELECT O1.SSN FROM Owns O1 JOIN Owns O2 ON O1.SSN = O2.SSN JOIN Car C1 ON O1.license = C1.license JOIN Car C2 ON O2.license = C2.license WHERE C1.model = 'TOYOTA' AND C2.model = 'DODGE';}
\]

\[
\text{OR}
\text{(SELECT O.SSN FROM Owns O, Car C WHERE O.license = C.license AND C.model='TOYOTA') INTERSECT (SELECT O.SSN FROM Owns O, Car C WHERE O.license = C.license AND C.model='DODGE');}
\]

3. Find the license number of all cars that have been involved in more than one accident.

\[ \pi_{\text{A1.license}} (\rho_{\text{A1}(\text{Accident}) \bowtie \text{A1.license} = \text{A2.license} \land \text{A1.accidentDate <> A2.accidentDate} } \rho_{\text{A2}(\text{Accident})}) \]

\[
\text{SELECT A1.license FROM Accident A1, Accident A2 WHERE A1.license = A2.license AND A1.accidentDate <> A2.accidentDate;}
\]

\[
\text{OR}
\]

\[
\text{SELECT license FROM Accident GROUP BY license HAVING COUNT(accidentDate) > 1;}
\]
4. Who is the driver who participated in the most costly accident? Return the name of the driver and the amount of damage.

\[
\pi_{\text{name, damageAmount}}(\text{Person} \bowtie \text{Person.SSN} = \text{driverSSN} (\pi_{\text{driverSSN, damageAmount}}(\text{Accident}) - \pi_{\text{A1,driverSSN, A1.damageAmount}}(\rho_{\text{A1,Accident}} \bowtie \text{A1.damageAmount < A2.damageAmount} \rho_{\text{A2,Accident}})))
\]

SELECT name, damageAmount FROM Person JOIN Accident ON Person.SSN = Accident.driverSSN WHERE damageAmount >= ALL (SELECT damageAmount FROM Accident);

5. Find the model of cars, for which all cars of that model have been involved in an accident.

\[
\pi_{\text{license, model}}(\text{Car}) ÷ \pi_{\text{license}}(\text{Accident})
\]

SELECT model FROM Car C WHERE NOT EXISTS (SELECT license FROM Car C2 WHERE C2.model = C.model MINUS (SELECT license FROM Accident));