CSE 344 Midterm Exam Nov. 3, 2014 Sample Solution

This exam deals with a database that stores information about ballet dancers, shows, and companies.

Company(name, city, country)
Dancer(did, name, birthyear, country)
Show(sid, title, choreographer, composer, year)
Role(did, sid, role, company)

The underlined attributes are keys for each relation. The tables contain the following information:

- *Company* stores information about dance companies. The attributes *name*, *city*, and *country* are all strings; we assume for this exam that all companies have unique names.
  Examples: ('Bolshoi', 'Moscow', 'Russia'), ('PNB', 'Seattle', 'USA')

- *Dancer* stores information about individual dancers. *did* is a unique integer id for each dancer. *name* is a string with the dancer's name, *birthyear* is an integer, and the dancer's native *country* is a string.
  Examples: (101, 'Pavlova', 1881, 'Russia'), (108, 'Korbes', 1981, 'Brazil')

- *Show* stores information about ballet shows (dances). Each show has a unique integer id *sid*, string attributes for the show *title*, *choreographer*, and *composer*, and an integer *year* in which the show was created.
  Examples: (205, 'Swan Lake', 'Petipa', 'Tchaikovsky', 1895), (204, 'Apollo', 'Balanchine', 'Stravinsky', 1928);

- *Role* stores information about which dancers have been in which shows, the name of the role (part) they danced, and the company where they danced that part in that particular show. The dancer and show id's are integers, the *role* and *company* names are strings. A dancer may have danced multiple roles in the same show at the same company, or danced the same role in the same show for different companies, and so forth.
  Examples: (108, 205, 'Black Swan', 'PNB'), (107, 204, 'Apollo', 'NYCB').

  Several attributes in *Role* are foreign keys: *did* references *did* in *Dancer*, *sid* references *sid* in *Show*, and *company* references *name* in *Company*.

For this exam, assume that all data values are not null.

The next page contains some sample data for each of these tables, and this data referenced in one of the later questions. The data may be useful in understanding how the information is stored in the tables.

Answer the questions about this database on the following pages. You may remove this page and the next from the test for reference if that is convenient.

Example data. This data is used in a later question, and may also be useful for understanding the data stored in the tables.

```
select * from Company;
name          city              country
------------  ----------------  ---------------
Imperial      St. Petersburg    Russia
Bolshoi       Moscow            Russia
Ballet Russe  Paris             France
NYCB          New York          USA
PNB           Seattle           USA

select * from Dancer;
did         name              birthyear         country
----------  ----------------  ----------------  ---------------
101         Pavlova           1881              Russia
102         Legnani           1863              Italy
103         Gerdt             1884              Russia
104         Ulanova           1910              Russia
105         Duncan            1877              USA
106         Dumas Ang         1994              USA
107         Boal              1965              USA
108         Korbes            1981              Brazil

select * from Show;
sid         title             choreographer     composer          year
----------  ----------------  ----------------  ----------------  ----
201         The Swan          Fokine            Saint-Seans       1905
202         Cinderella        Ivanov            Filinhoff         1893
203         Cinderella        Zakharov          Prokofiev         1940
204         Apollo            Balanchine        Stravinsky        1928
205         Swan Lake         Petipa            Tchaikovsky       1895
206         Nutcracker        Balanchine        Tchaikovsky       1954
207         Nutcracker        Stowell           Tchaikovsky       1983

select * from Role;
did         sid         role              company
----------  ----------  ----------------  ---------------
108         204         Terpsichore       NYCB
106         207         Warrior Mouse     PNB
107         204         Apollo            NYCB
101         201         Swan              Ballet Russe
102         202         Cinderella        Imperial
103         202         Prince            Imperial
108         205         White Swan        PNB
108         205         Black Swan        PNB
104         203         Cinderella        Bolshoi
```

CSE 344 Midterm Exam Nov. 3, 2014 Sample Solution

Reference Information

This information may be useful during the exam. Feel free to use it or not as you wish. You can remove this page from the exam if that is convenient.

Reference for SQL Syntax

Outer Joins

```
-- left outer join with two selections:
select *
from R left outer join S on R.x=55 and R.y=S.z and S.u=99
```

The UNION Operation

```
select R.k from R union select S.k from S
```

The CASE Statement

```
select R.name, (case when R.rating=1 then 'like it'
                when R.rating=0 then 'do not like it'
                when R.rating is null then 'do not know'
                else 'unknown' end)
                as a_rating
from R;
```

The WITH Statement

Note: with is not supported in sqlite, but it is supported SQL Server and in postgres.

```
with T as (select * from R where R.K>10)
select * from T where T.K<20
```

Reference for Relational Algebra

Name	Symbol
Selection	σ
Projection	π
Join	⋈
Group By	γ
Set Difference	–
Duplicate Elimination	δ

CSE 344 Midterm Exam Nov. 3, 2014 Sample Solution

Question 1. (12 points) SQL tables. Write the SQL commands needed to create the *Dancer* and *Role* tables described on page 2. Be sure to include the correct names and types for all attributes, and any key or foreign key constraints. (You do *not* need to give SQL commands to create the other tables – just the ones asked for.)

```
CREATE TABLE Dancer (
  did       int PRIMARY KEY,
  name      varchar(20),
  birthyear int,
  country   varchar(20)
);

CREATE TABLE Role (
  did int references Dancer,
  sid int references Show,
  role varchar(20),
  company varchar(20) references Company,
  PRIMARY KEY(did, sid, role, company)
);
```

Question 2. (40 points) SQL queries. Write SQL queries to retrieve the requested information from the dance database tables described previously. The queries you write must be proper SQL that would be accepted by SQL Server or any other SQL implementation. You should not use incorrect SQL, even if sqlite might produce some sort of answer from the buggy SQL.

(a) (10 points) For every dancer who has performed the role 'Black Swan' in the show 'Swan Lake' for one or more companies, list the name of the dancer and the company name(s), sorted by dancer name. If the dancer has performed that role for more than one company, there should be one line of output for each dancer, company pair. The companies can be listed in any order.

```
SELECT distinct dancer.name, role.company
FROM role, show, dancer
WHERE role.sid = show.sid
  AND role.did = dancer.did
  AND role.role = "Black Swan"
  AND show.title = "Swan Lake"
ORDER BY dancer.name;
```

(b) (10 points) List the dancer ids (did) and names of all dancers who have danced in a show choreographed by 'Fosse' but have not danced in a show choreographed by 'Robbins'. Each did/name pair should only appear once in the output.

```
SELECT distinct dancer.did, dancer.name
FROM role, show, dancer
WHERE role.sid = show.sid
  AND role.did = dancer.did
  AND show.choreographer = "Fosse"
  AND role.did NOT IN (
                SELECT role.did
                FROM role, show
                WHERE role.sid = show.sid
                AND show.choreographer = "Robbins"
  );
```

(continued next page)

Question 2. (cont.) (c) (10 points) List the dancer ids (did) and names of all dancers born on or before 1950 and who have danced in at least three different shows. If a dancer has danced different roles in the same show, it still only counts once in the total number of shows. Each dancer/did pair should only be listed once.

```
SELECT dancer.did, dancer.name
FROM dancer, role
WHERE dancer.birthyear <= 1950
  AND dancer.did = role.did
GROUP BY dancer.did, dancer.name
HAVING count(DISTINCT role.sid) >= 3;
```

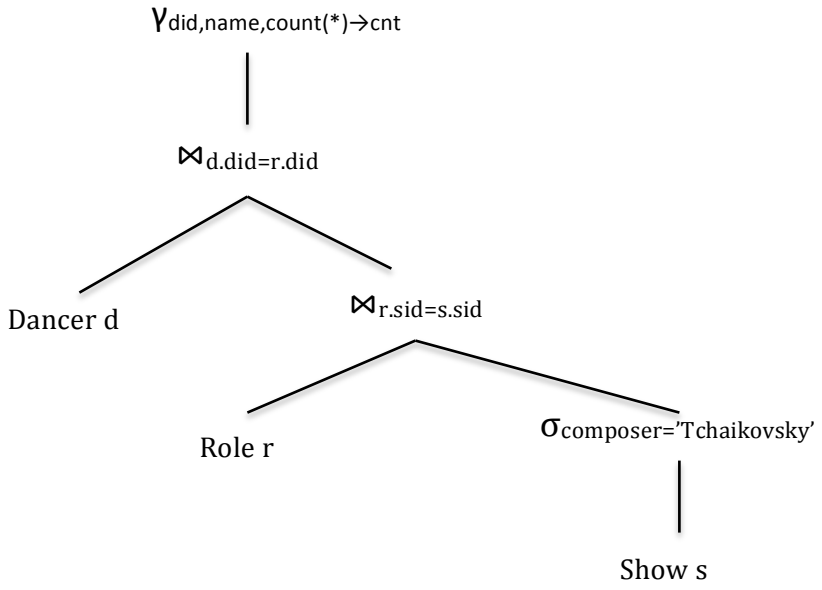
(d) (10 points) For every dancer who has danced for one or more companies in a different country than where they were born, list the name of the dancer and the names of those companies.

```
SELECT distinct dancer.name, company.name
FROM dancer, role, company
WHERE dancer.did = role.did
  AND role.company = company.name
  AND company.country <> dancer.country;
```

Question 3. (16 points) Relational algebra, queries, and indexes. Consider the following SQL query:

```
SELECT d.did, d.name, count(*)
FROM   dancer d, role r, show s
WHERE  d.did=r.did AND r.sid=s.sid AND s.composer='Tchaikovsky'
GROUP BY d.did, d.name;
```

(a) (6 points) Give a relational algebra tree that corresponds to this query.



A different relational algebra tree that produced the correct results would also receive full credit.

(b) (6 points) If we execute this query using the data on page 3, what output is produced?

did	name	count(*)
106	Dumas Ang	1
108	Korbes	2

Question 3. (cont.) (c) (4 points) Here is a list of possible indexes that might be useful in processing the query given in part (a). Pick **up to *three*** indexes that collectively would be most useful in speeding up processing of that query. Assume that there are no existing indexes and that the data in all tables is not clustered. Circle your answers. Hint: There might be more than one possible correct (i.e., "best") answer to this question.

Company(name)

Company(city)

Dancer(did)

Dancer(name)

Dancer(country)

Dancer(name, did)

Dancer(did, name)

Show(sid)

Show(title)

Show(choreographer)

Show(composer)

Show(title, composer)

Show(composer, name)

Role(did)

Role(sid)

Role(role)

Role(company)

Role(sid, did)

Role(company, sid)

The best choices would be one index from each of the three following sets:
{Dancer(did) or Dancer(did, name) or Role(did)}
{Show(sid) or Role(sid) or Role(sid, did)}
{Show(composer) or Show(composer, name)}

CSE 344 Midterm Exam Nov. 3, 2014 Sample Solution

Question 4. (16 points) Relational calculus and datalog. Suppose we want the following information: Give the names of all dancers that have danced with exactly one company.

(a) (8 points) Write this query using relational calculus.

Q(n) = ∃ b, ctry, d. Dancer(d, n, b, ctry) ∧ ∃ s1, r1, c1. Role(d, s1, r1, c1) ∧

∀ c2. (∃ s2, r2. Role (d, s2, r2, c2) ⇒ (c1 = c2))

There are other possible solutions to this and similar questions. Any solution that produces the correct result should receive full credit.

(b) (8 points) Write this query in datalog with negation. (You can use your answer from part (a) to help with this part of the question, but you are not required to do so.)

Companies (d, c) :- Role(d, -, -, c)

MultipleCompanies (d) :- Companies(d, c1), Companies(d, c2), c1 != c2

OneCompany (n) :- Dancer(d, n, -,-), Companies(d, -), not MultipleCompanies(d)

Question 5. (16 points) Relational calculus and algebra. Suppose we want the following information: List the names of all Companies whose dancers are from only one single country. (Note: this may not be true of any of the Companies in the sample data.) If it matters, you can assume that all Companies in the database have employed at least one dancer.

(a) (8 points) Write this query using relational calculus.

$$Q(n) = \exists\, d, dn, y, ctry, s1, r1 \,.\, Role(d, s1, r1, n) \wedge Dancer(d, dn, y, ctry) \wedge$$
$$\forall\, d2\ (\exists\, r2, s2 \,.\, Role(d2, s2, r2, n) \Rightarrow \exists\, dn2, y2 \,.\, Dancer(d2, dn2, y2, ctry))$$

(b) (8 points) Draw a relational algebra tree for this query. (Hint: your answer to part (a) may be helpful, but you are not required to use it.)