Greg Finkelberg – Module 4 WinRDBI Assignment

Queries

The output schema for the query is given in parentheses following the query specification.

1) List the CD title, recording label and number of CDs sold for CDs with a top 5 rating (1<= topCDs.rating <= 5) in 2003 where the recording label is located in Los Angeles. **(cdCode, cdTitle, labelName, numberSold)**

```
% Query 1 - List the CD title, recording label and number of CDs sold for CDs with a top 5 rating (1<= topCDs.rating <= 5) in 2003 where the recording label is located in Los Angeles. (cdCode, cdTitle, labelName, numberSold)
% Define a relation 'labelLocation' to represent recording labels located in Los Angeles.
labelLocation :=
    project labelID, labelName
        (select location = 'Los Angeles' (recordingLabel));

% Define a relation 'cdRatings' to represent CD ratings for the year 2003 with a rating less than or equal to 5.
cdRatings :=
    project cdCode, year, rating
        (select year = 2003 and rating <= 5 (topCDs));

% Define a relation 'q1' to represent the query result which includes CD code, title, label name, and number sold.
q1 :=
    project cdCode, cdTitle, labelName, numberSold
        (cd njoin (labelLocation njoin cdRatings));
```

Relation Name	# Tuples	cdCode/char	cdTitle/char	labelName/char	numberSold/numeric
labelLocation	2	'C4'	'Friday Night'	'A&M Records'	800,000
cdRatings	4	'C7'	'Broken'	'A&M Records'	600,000
q1	4	'C8'	'Golden Road'	'Disney Records'	800,000
cd2003	8	'C10'	'Be Here'	'A&M Records'	790,000

2) Which songs appeared on the best-selling (max number of CDs sold) CDs published in 2003? **(songCode, songTitle)**

```
% Query 2 - Which songs appeared on the best-selling (max number of CDs sold) CDs published in 2003? (songCode, songTitle)
% -Define a relation 'cd2003' to represent CDs sold in the year 2003, including their code and number sold.
cd2003 :=
    project cdCode, numberSold
        (select year = 2003 (cd));

% Create a copy of relation 'cd2003' to compare against for finding not highest selling CDs.
cd2003Copy (cdCodeCopy, numberSoldCopy) := project cdCode, numberSold (cd2003);

% Identify CDs that are not the highest selling ones in 2003.
notHighestSellingCDs := project cdCode, numberSold (select numberSold < numberSoldCopy (cd2003 product cd2003Copy));

% Identify the highest selling CDs in 2003 by subtracting notHighestSellingCDs from cd2003.
highestSellingCDs :=
    project cdCode, numberSold
        (cd2003 difference notHighestSellingCDs);

% Define a relation 'q2' to represent the query result which includes song code and title for songs on the highest selling CDs.
q2 :=
    project songCode, songTitle
        (song njoin(highestSellingCDs njoin composedOf));
```

Relation Name	# Tuples	songCode/char	songTitle/char
labelLocation	2	'S1'	'Sweet Dreams'
cdRatings	4	'S2'	'Goodnight'
q1	4	'S7'	'Breathless'
cd2003	8	'S8'	'My Oh My'
cd2003Copy	8	'S9'	'Ooh La La'
notHighestSellingCDs	6	'S11'	'Hot As Hell'
highestSellingCDs	2	'S12'	'Goldilocks'
q2	10	'S17'	'Days Go By'
artistSongwriters	20	'S18'	'Another Day in Paradise'
topRatedSongs	13	'S19'	'Smile'
topSongsBySongwriters	14		

3) List names of artists that are song writers and have never had a song receive a top 5 rating. **(artistID, firstName, lastName)**

```
% Query 3 - List names of artists that are song writers and have never had a song receive a top 5 rating. (artistID, firstName, lastName)
% Define a relation 'artistSongwriters' to represent artists who are also songwriters, including their IDs, first names, last names, and the codes of songs they've written.
artistSongwriters :=
    project artistID, firstName, lastName, songCode
        (artist njoin writtenBy);

% Define a relation 'topRatedSongs' to represent songs with ratings equal to or less than 5.
topRatedSongs :=
    (select rating <= 5 (topSongs));

% Identify top-rated songs written by songwriters, including their IDs, first names, last names, and song codes.
topSongsBySongwriters :=
    project artistID, firstName, lastName, songCode
        (artistSongwriters njoin topRatedSongs);

% Identify artists who are songwriters but have not written top-rated songs.
q3 :=
    project artistID, firstName, lastName
        (artistSongwriters difference topSongsBySongwriters);
```

Relation Name	# Tuples
labelLocation	2
cdRatings	4
q1	4
cd2003	8
cd2003Copy	8
notHighestSellingCDs	6
highestSellingCDs	2
q2	10
artistSongwriters	20
topRatedSongs	13
topSongsBySongwriters	14
q3	6

artistID/char	firstName/char	lastName/char
'A3'	'John'	'Stark'
'A6'	'Francis'	'McDermott'
'A7'	'Steve'	'Nash'
'A8'	'Lisa'	'Raymond'
'A10'	'Henry'	'Brown'
'A12'	'John'	'Hopkins'

4) List the artists that have been a member of more than one group. (artistID, firstName, lastName)

```
% Query 4 - List the artists that have been a member of more than one group. (artistID, firstName, lastName)
% Define a relation 'artistGroupMemberships' to represent artist memberships in musical groups, including their IDs and group codes.
artistGroupMemberships := project artistID, groupCode(member);

% Create a copy of relation 'artistGroupMemberships' to compare against for finding different group memberships.
artistGroupMembershipsCopy (artistIDCopy, groupCodeCopy) := project artistID, groupCode(member);

% Identify artists who are members of different groups by comparing against the copy.
q4 := project artistID (select artistID = artistIDCopy and groupCode <> groupCodeCopy
                        (artistGroupMemberships product artistGroupMembershipsCopy));
```

Relation Name	# Tuples
labelLocation	2
cdRatings	4
q1	4
cd2003	8
cd2003Copy	8
notHighestSellingCDs	6
highestSellingCDs	2
q2	10
artistSongwriters	20
topRatedSongs	13
topSongsBySongwriters	14
q3	6
artistGroupMemberships	21
artistGroupMembershipsCopy	21
q4	6

artistID/char
'A1'
'A2'
'A3'
'A4'
'A9'
'A10'