# Database Systems, CSCI 4380-01 Exam #2 Answers
# Thursday November 4, 2010 at 2 pm

1.a.

```
SELECT
   distinct a.id
   , r.songid
FROM
   artists a
   , recordings r
   , albumsReleasedBy arb
   , songComposedBy c
   , songWrittenBy w
WHERE
   a.id = arb.artistid
   and arb.albumid = r.albumid
   and ((r.songId = c.songId and c.artistid = a.artistid)
       or
        (r.songid = w.songid and w.artistid = a.artistid)
       )
```

1.b

```
SELECT
   DISTINCT
   s.id
FROM
   songs s
   left outer join
   songsComposedBy sc on s.id = sc.songId
   left outer join
   songsWrittenBy sw on s.id = sw.songId and sc.artistId = sw.artistID
WHERE
   sw.artistId is null
```

or

```
SELECT
   s.id
FROM
   songs s
WHERE
   not exists (
```

```
     SELECT 1
     FROM
        songsComposedBy sc
        , songsWritten sw
     WHERE
        sc.songId = s.id
        and sc.songId = sw.songId
        and sc.artistId = sw.artistId
```

1.c

```
select
  a.id
  , r.characteristicid
  , count(*)
from
   albums a
   , recordings r
   , recordingHasCharacteristics rhc
where
   now() - a.releaseDate <= interval '1 year'
   and r.albumid = a.id
   and r.recordingid = r.id
group by
   a.id
   , r.characteristicid
```

1.d

```
select
   s.id
   , count(distinct r.songId) as numRecordings
   , count(distinct r.artistid) as numArtists
from
   song s
   left outer join recordings r
   on s.id = r.songid
   left outer join albumsReleasedBy arb
   on r.albumid = arb.albumid
group by
   s.id
```

2.

Artists 1,2,3 Songs 1,2,3

SongW/CBy: (1,1) (1,2), (2,1), (2,2), (2,3)

(1,1)(1,2)(1,3)(2,1)(2,2)(2,3)(3,1)(3,2)(3,3) - (1,1) (1,2), (2,1), (2,2), (2,3) = (1,3)(3,1)(3,2)(3,3)
Return 1, 3

a. Artists who have not composed or written all the songs in the database.

```
select
   distinct
   a.id
from
   ((select s.id as SongId, a.id as artistId from artists a, songs s)
   except
   ((select * from songComposedBy) union (select * from songWrittenBy)) a
```

or

```
select
   distinct a.id
from
   artists a, songs s
where
   not exists (select 1
      from  songComposedBy sc, songWrittenBy sw
      where (a.id = sc.artistId and s.id = sc.songId)
            or (a.id = sw.artistId and s.id = sw.songId))
```

3. a. delete only (3,c) from abc, the remaining tables are the same

| abc | |
|---|---|
| id | a1 |
| 1 | 'a' |
| 2 | 'b' |

| def | |
|---|---|
| id | abc_id |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |

| ghi | |
|---|---|
| id | def_id |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |

b. delete (1,a) from abc, which triggers delete of (1,1), (2,1) from def, which triggers update of (1,1), (2,1), (3,2) to (1,null), (2,null), (3,null). However as null is not allowed for def_id, the whole transaction fails. As a result, no changes are made to the database.

| abc | |
|---|---|
| id | a1 |
| 1 | 'a' |
| 2 | 'b' |
| 3 | 'c' |

| def | |
|---|---|
| id | abc_id |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |

| ghi | |
|---|---|
| id | def_id |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |

c.(2,b) is changed to (4,b), which means (3,2) is changed to (3,null).

| abc | |
|---|---|
| id | a1 |
| 1 | 'a' |
| 4 | 'b' |
| 3 | 'c' |

| def | |
|---|---|
| id | abc_id |
| 1 | 1 |
| 2 | 1 |
| 3 | null |

| ghi | |
|---|---|
| id | def_id |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |

d. (3,2) is changed to (4,2), which means (4,3) is changed to (4,4).

| abc | |
|---|---|
| id | a1 |
| 1 | 'a' |
| 2 | 'b' |
| 3 | 'c' |

| def | |
|---|---|
| id | abc_id |
| 1 | 1 |
| 2 | 1 |
| 4 | 2 |

| ghi | |
|---|---|
| id | def_id |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 4 |

4. See figure. Note that people may hold offices without winning a specific election.

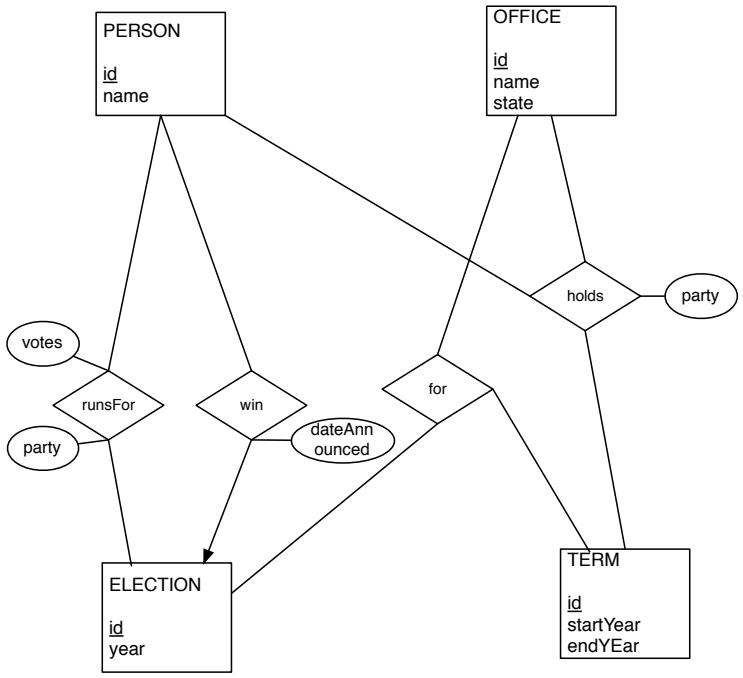


Figure 1: E-R Diagram for Question 4

5.a. read committed, repeatable read and serializable. A dirty read is an uncommitted value being read by a transaction. If the value is later invalidated, then the transaction reading it might need to be aborted. If the transaction acting on dirty read has committed, then it may have caused an error as it acted in on now incorrect data.

5.b. Finds all albums which have missing tracks (do not start at 1 or have a missing value in between like 1,2,4).

# Appendix

Suppose you are given the below data model for an application providing music services. This is the same data model that we have created in class and used in Exam #1 (except I did not include the data relating to the users).

Note that, albumId refers to Albums(id), songId refers to Songs(id), artistId refers Artists(id), recordingId refers to Recordings(id) and characteristingsId refers to Characteristings(id).

**DATA MODEL.**

Artists(id, name, bio, birthYear)

Albums(id, name, releaseDate)

Songs(id, name, lyrics)

Recordings(id, trackNum, length, songId, albumId)

Characteristics(id, name)

RecordingHasCharacteristics(recordingId, characteristicId)

AlbumsReleasedBy(albumId, artistId)

SongComposedBy(songId, artistId)

SongWrittenBy(songId, artistId)