CISC 7610 Lecture 2
Review of relational databases

Topics:

Relational database management systems
Example data modeling problem
Entity-relationship diagrams
Structured query language

A relational database management system (RDBMS)

- Uses relational data structures
- Has a declarative data manipulation language at least as powerful as the relational algebra

Uses relational data structures

- Relation: table with rows and columns
- Attribute: column
- Tuple: row
- Key: combination of attributes that uniquely identifies each row
- Integrity rules: Constraints imposed upon the database

Has a declarative data manipulation language

- Declarative: says what, not how to manipulate data
- Relational algebra
  - Selection: extract a subset of tuples
  - Projection: extract a subset of attributes
  - Cartesian product: extract all combinations of pairs of tuples from two relations
  - Union: combine two sets of tuples
  - Set difference: remove one set of tuples from another

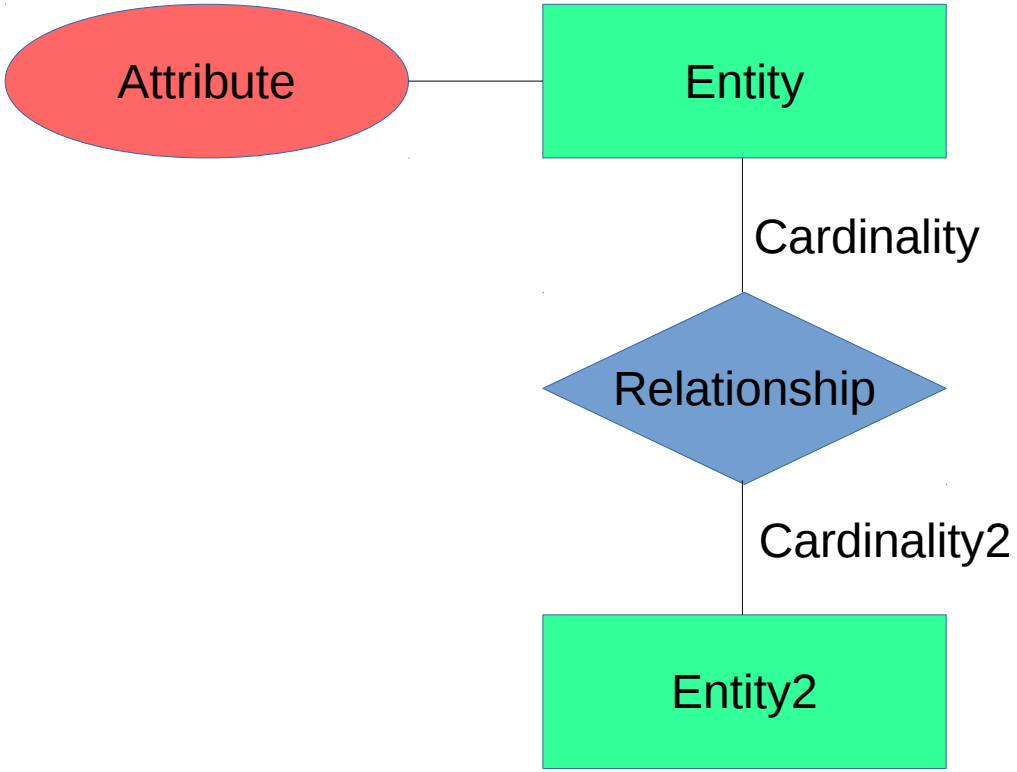
Structured query language (SQL)

- Data definition language
  - Define relational schemata (pl of schema)
  - Create/alter/delete tables and the attributes
- Data manipulation language
  - Insert/delete/modify tuples in relations
  - Query one or more tables
- Can implement relational algebra, but also takes some liberties with it

Example data: Music collection

- Artists: Name
- Albums: Name, Release date
- Tracks: Name, Duration, Number
- Each album has one artist
- Tracks can appear on multiple albums (compilations)

Entity-relationship diagrams



Do: Draw ER diagram for ex data

- Artists: Name
- Albums: Name, Release date
- Tracks: Name, Duration, Number
- Each album has one artist
- Tracks can appear on multiple albums (compilations)

Translating ER diagrams to schema

- Entities become tables
- Attributes become their attributes
- Many-to-many relationships become join tables
  - Can have additional attributes
- Other relationships become foreign keys
  - One-to-one, many-to-one, one-to-many
  - Attributes added to table

Do: Translate ER diagram to schema for example data

SQL CREATE statement

```
CREATE TABLE table_name
(
  column_name1 data_type(size),
  column_name2 data_type(size),
  column_name3 data_type(size),
  ....
);
```

Do: Create tables for example data

SQL INSERT statement

```
INSERT INTO table_name
   (column1,column2,column3,...)
   VALUES
   (value1,value2,value3,...);
```

Do: Populate tables with ex data

Artists	
Id	Name
1	David Bowie
2	Queen

Albums			
Id	Name	Release	ArtistId
1	Space oddity	1969	1
2	… Ziggy startdust ...	1972	1
3	Best of Bowie	2002	1
4	Hot space	1982	2

Track		
Id	Name	Duration
1	Space oddity	5:15
2	Suffragette city	3:25
3	Under pressure	4:02

AlbumsHaveTracks		
AlbumId	TrackId	Number
1	1	1
2	2	10
3	1	1
3	2	8
4	3	11

Reminder: Main question of course

How can we process and store multimedia data so that we can find what we are looking for in the future?

Queries: find what we are looking for

- Search through the data
- Search through complex relationships
- Aggregate over the data for reporting
- And do all of this efficiently...

SQL SELECT, single table

```
SELECT attribute1, attribute2
FROM relation
WHERE attribute1 = 'condition'
ORDER BY attribute2;
```

Do: Write a select query to answer

What is the duration of “Suffragette City”?

SQL SELECT, multiple tables

```
SELECT r1.attribute1, r2.attribute1
FROM relation1 AS r1,
  Relation2 AS r2
WHERE attribute1 = 'condition'
  AND r1.attribute1 = r2.attribute2
ORDER BY r1.attribute1;
```

Do: Write a select query to answer

Find the AlbumIds of all of David Bowie's albums

Do: Write a select query to answer

Find the TrackIds of all of David Bowie's tracks

Do: Write a select query to answer

- Find all songs containing David Bowie's vocals
- Find all songs at 120 beats per minute
- Find all songs sampled by other artists
  - These all require further modeling or analysis of the audio...

How do we make databases that are

- Effective (correct, durable, coherent, ...)
  - Transactions
- Efficient
  - Concurrency
  - Memory hierarchy
  - Indexing
  - Query optimization