IISQLSRV and SQL (re-) introduction

CSE 444 section

September 30, 2010

Today

- IISQLSRV and Management Studio
- SQL practice

About section and the TA

- Section in EE1 025 on Thursdays
  - AA: 8:30-9:20, AB: 9:30-10:20
  - Feel free to come to either

- Me: Michael Ratanapintha
  - michaelr@cs.washington.edu
  - Office hours: Thurs. 10:30-noon in CSE 006

The dreaded icebreaker…

Connecting to IISQLSRV

IISQLSRV connection settings

From yesterday's lecture:

- Server: iisqlsrv.cs.washington.edu
- Use *SQL Server Authentication*
- Username: your UW NetID
- Password: see lecture 1
  - Write this down NOW, we won't say it again
  - You'll have to change it on first login

IMDB database

Actor (*id*, fname, lname, gender)

Movie (*id*, name, year, rank*)

Directors (*id*, fname, lname)

Casts (pid, mid, role)

Movie_Directors (did, mid)

Genre (genre, mid)

* *currently unused, always null*

A simple query

Tell me all you know about every movie called “Go Tell It On The Mountain”.

A simple query

Tell me about every movie called "Go Tell It On The Mountain":

```
SELECT *
FROM Movie
WHERE name = 'Go Tell It On The
  Mountain';
```

A simple query

Tell me about every movie called "Go Tell It On The Mountain":

```
SELECT *
FROM Movie
WHERE name = 'Go Tell It On The Mountain';
```

← Every column…

← … of every row in Movie…

← … whose "name" field is this

A simple query

Now tell me only the year each such movie was made:

```
SELECT YEAR ← only the Year column…
FROM Movie
WHERE name = 'Go Tell It On The
  Mountain';
```

More examples

- Names of all Star Wars movies
- All Star Wars movies made in 2000 or later
- Names and production years of all Star Wars movies from earliest to latest

"Star Wars movie" = movie with "Star Wars" in the name

Something a little harder…

Who directed The Empire Strikes Back?

Answer: joins!

Who directed The Empire Strikes Back?

Movie (*id*, name, year, rank)

Directors (*id*, fname, lname)

Movie_Directors (did, mid)

Need to *join* (combine) the data from these tables!

Director of Empire Strikes Back

Director of Empire Strikes Back

```
SELECT d.id, d.fname, d.lname
FROM Movie m, Movie_Directors md,
  Directors d
WHERE m.id = md.mid AND
  md.did = d.id AND
  m.name = 'Star Wars: Episode V -
  The Empire Strikes Back';
```

Director of Empire Strikes Back

```
SELECT d.id, d.fname, d.lname
FROM Movie m, Movie_Directors md,
  Directors d
WHERE m.id = md.mid AND
  md.did = d.id AND
  m.name = 'Star Wars: Episode V -
  The Empire Strikes Back';
```

Join conditions

How do joins work formally?

Recall from discrete math (311 or 321) the *Cartesian product* of sets **X** and **Y**:

- All ordered pairs (x, y) such that x in **X**, y in **Y**

How do joins work formally?, cont.

Logically, joins work as follows:

1. Take Cartesian product of the sets of all rows in tables being joined
2. Use the join conditions to filter out only those tuples that match

In practice: much faster, uses less memory

Aggregates

Sometimes we just want summary or extreme-case data

- All Star Wars movies → number of Star Wars movies
- Dates of all movies → date of earliest movie

Aggregates

SQL has *aggregation operators* to help with this

- count, sum, avg, min, max

Aggregates

Sometimes we just want summary or extreme-case data

- All Star Wars movies → number of Star Wars movies
- Dates of all movies → date of earliest movie

Aggregates

Sometimes we just want summary or extreme-case data

- SELECT * FROM Movie WHERE name LIKE… → SELECT COUNT(*) FROM Movie…
- Dates of all movies → date of earliest movie

Aggregates

Sometimes we just want summary or extreme-case data

- SELECT * FROM Movie WHERE name LIKE… → SELECT COUNT(*) FROM Movie…
- SELECT year FROM Movie → SELECT MIN(year) FROM Movie

Aggregates and grouping

Aggregates are not so useful by themselves…

But combined with *grouping* (lecture 3), they become very powerful!

Aggregates and grouping

List actors' first names and their frequencies, from most to least popular:

```
SELECT fname, COUNT(*) AS freq
FROM Actor
GROUP BY fname ← grouping by first name
ORDER BY freq DESC;
```

Project 1

More fun with the IMDB database!

Some queries need more advanced SQL

Posted now, due October 15

This weekend: log in to IISQLSRV!

If you can't, email me: michaelr@cs