TA Sections

- Quiz sections: Th 8:30-9:30, 9:30-10:30
    - AA,AB: THO 202
    - AC: MEB 235

- TAs:
    - Jessica Leung
        - Office hours: Fridays, 10:30 - 11:30am @ CSE 006
        - joyleung@cs.washington.edu
    - Daniel Swisher
        - Office hours: Wednesdays @ TBD
        - nguard65@cs.washington.edu
    - Derek Cheng
        - Office hours: Mondays, 11:30 - 12:20pm @ CSE 216
        - imcheng@cs.washington.edu

Today: SQL Server Overview

- Login & change password
- SQL Server’s Object Explorer
- Queries on the IMDB database
- Indexes, Query Plans

Login & Change Password

- SQL Server Authentication
- Windows Authentication
  - You don’t want to put UW/CSE password into Java code, so use SQL Server Authentication

Security

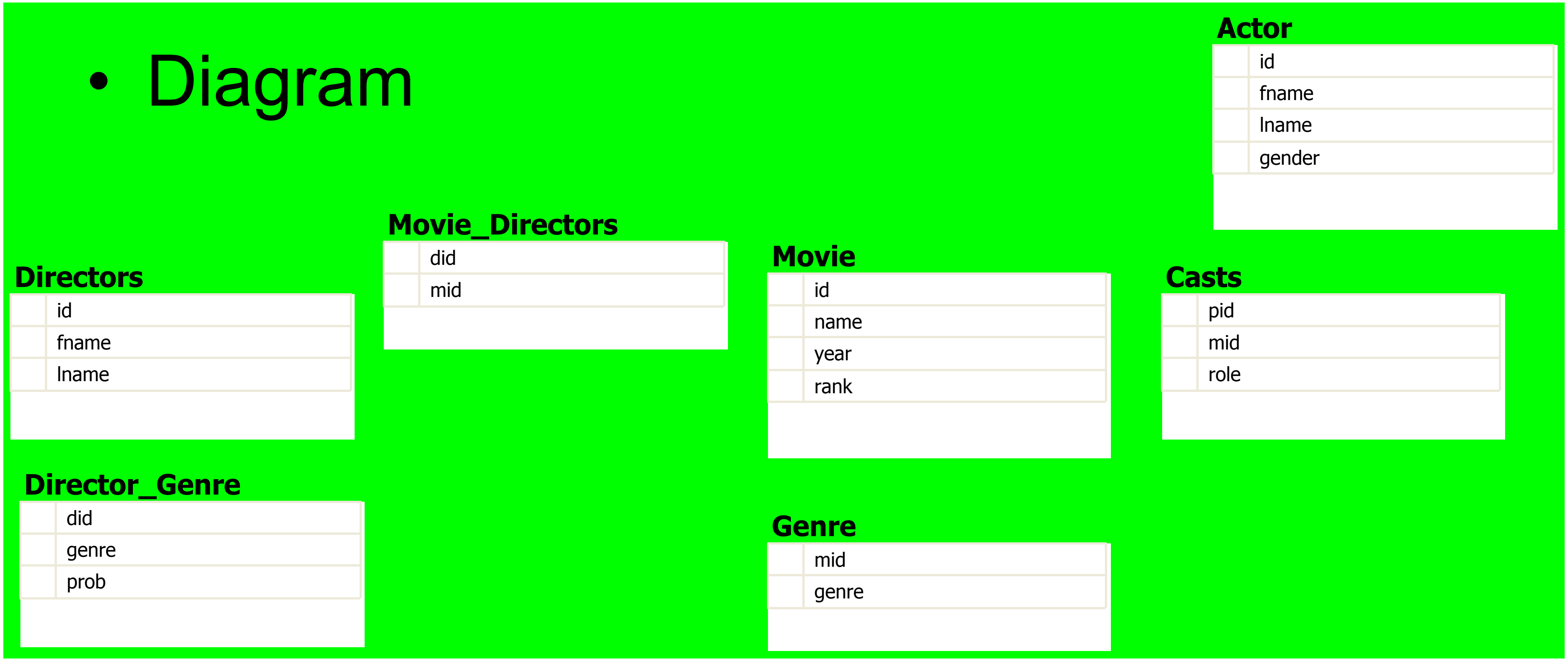
- Logins (Authentication)
  - SQL-Server-Level Security
  - Server-Roles
- User permissions
  - Database-Level Security
  - Database-Roles (db_datareader)

System Databases

- master
  - user accounts, info about other databases, server's processes
- model
  - template for creating new user databases
- msdb
  - scheduling maintenance tasks (alerts, backups)
- tempdb
  - temporary storage (dropped on shutdown)

IMDB Database

- Diagram



Actor

- id
- fname
- lname
- gender

Movie_Directors

- did
- mid

Directors

- id
- fname
- lname

Movie

- id
- name
- year
- rank

Casts

- pid
- mid
- role

Director_Genre

- did
- genre
- prob

Genre

- mid
- genre

Queries

- Star Wars movies
- Star Wars movies between 2003 and 2008
- Number of actors by gender
- Number of movies by genre
- ... for movies released during last 5 years
- ... for movies before 1950

Queries

- Top 20 most popular first (last) names among actors
- Actors who played in most movies
- ... in recent years?
- ... in sci-fi movies?

Queries

- What does this query return?
  - SELECT COUNT(*) FROM
    (SELECT did, mid FROM Movie_Directors) md
- Same as this?
  - SELECT COUNT(*) FROM Movie_Directors
- And this one?
  - SELECT COUNT(*) FROM
    (SELECT DISTINCT did, mid FROM
    Movie_Directors) md
- What’s going on here?

Queries

- How can we find the duplicates?
- To get #unique pairs, we sum duplicates and subtract them
- Sum - Counts
- Compare Results

Indexes

- Clustered Index
  - Determines order in which rows of the table are physically stored
  - Can only have 1 clustered index per table
  - Primary Key Index: unique clustered index
- Non-Clustered Index
  - Separate objects that points to specific rows in table

Queries

- Movies that have at least one actor

  -
    ```
    SELECT COUNT(DISTINCT mid) FROM Casts;
    ```

  -
    ```
    SELECT COUNT(*) FROM
    (SELECT mid FROM Casts GROUP BY mid) a;
    ```

  -
    ```
    SELECT SUM(num) FROM
      (SELECT COUNT(DISTINCT m.id) as num
       FROM Movie m, Actor a, Casts c
       WHERE a.id = c.pid AND m.id = c.mid
       AND a.fname <> " GROUP BY m.id) b;
    ```

- Compare Query Execution Plans

Dan’s example

Actor:

id	fName	lName	gender
. . .		Hanks	
. . .			

Cast:

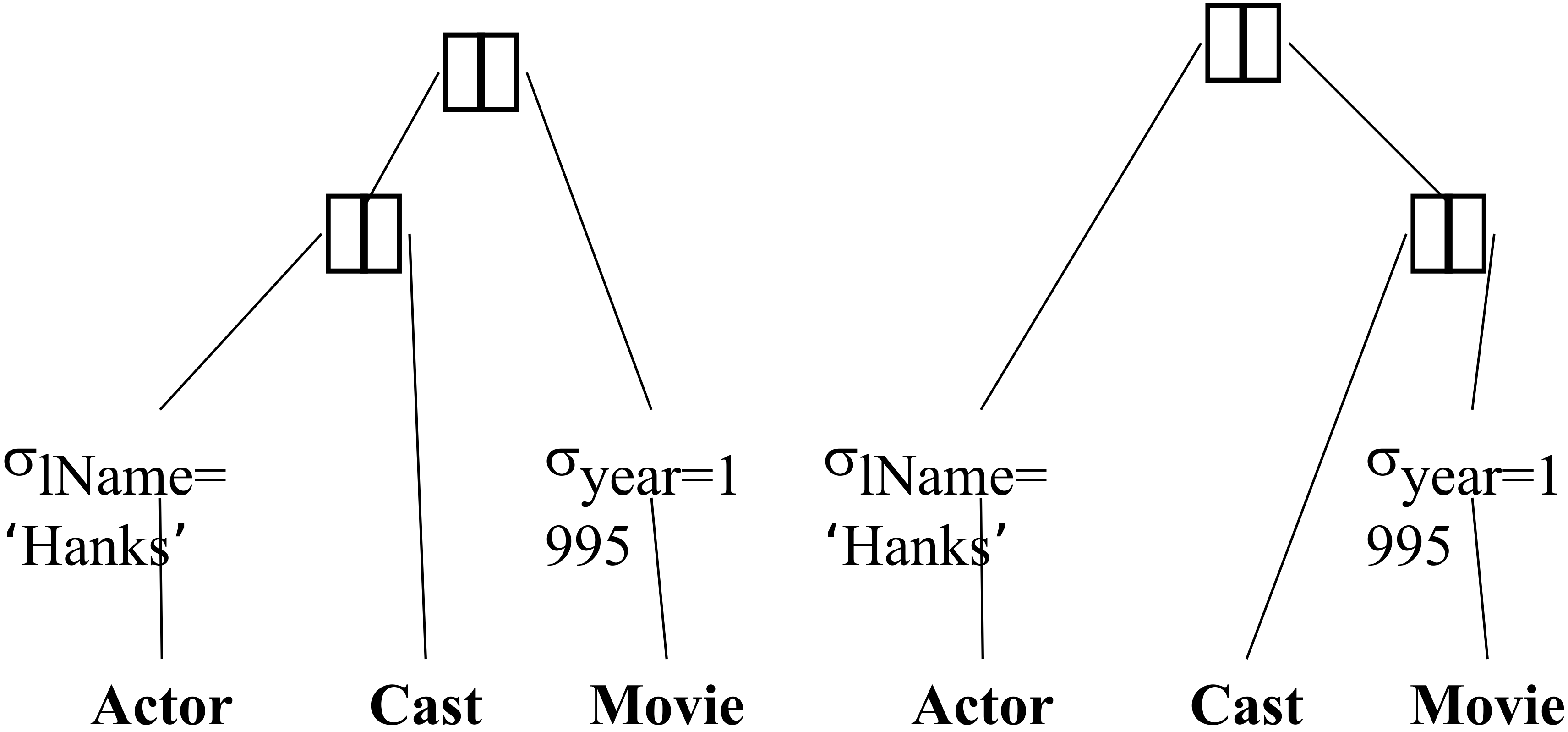
pid	mid
. . .	
. . .	

Movie:

id	Name	year
. . .		1995
. . .		

```
SELECT *
FROM  Actor, Casts, Movie
WHERE lname='Hanks' and Actor.id = Casts.pid
  and Casts.mid=Movie.id and Movie.year=1995 ┌
```

Which one got picked?



$\sigma_{lName=}$ 'Hanks'

$\sigma_{year=1}$ 995

$\sigma_{lName=}$ 'Hanks'

$\sigma_{year=1}$ 995

Actor **Cast** **Movie**

Actor **Cast** **Movie**

That’s It!

Queries

- Number of actors for each movie
- Problem!
- Solution 1
- Solution 2