CSE 154

LECTURE 14: MULTI-TABLE SQL QUERIES (JOINS)

Example world database

code	name	continent	independence_year	population	gnp	head_of_state	...
AFG	Afghanistan	Asia	1919	22720000	5976.0	Mohammad Omar	...
NLD	Netherlands	Europe	1581	15864000	371362.0	Beatrix	...
...	...	...	...	...	...	...	...

countries **(Other columns: region, surface_area, life_expectancy, gnp_old, local_name, government_form, ca pital, code2)**

id	name	country_code	district	population
3793	New York	USA	New York	8008278
1	Los Angeles	USA	California	3694820
...	...	...	...	...

cities

country_code	language	official	percentage
AFG	Pashto	T	52.4
NLD	Dutch	T	95.6
...	...	...	...

languages

- to test queries on this database, use username `traveler`, password `packmybags`

Example imdb database

id	first_name	last_name	gender
433259	William	Shatner	M
797926	Britney	Spears	F
831289	Sigourney	Weaver	F
...			

actors

id	name	year	rank
112290	Fight Club	1999	8.5
209658	Meet the Parents	2000	7
210511	Memento	2000	8.7
...			

movies

actor_id	movie_id	role
433259	313398	Capt. James T. Kirk
433259	407323	Sgt. T.J. Hooker
797926	342189	Herself
...		

roles

movie_id	genre
209658	Comedy
313398	Action
313398	Sci-Fi
...	

movies_genres

id	first_name	last_name
24758	David	Fincher
66965	Jay	Roach
72723	William	Shatner
...		

directors

director_id	movie_id
24758	112290
66965	209658
72723	313398
...	

movies_directors

- also available, `imdb_small` with fewer records (for testing queries)
- to test queries on this database, use the username/password that we will email to you soon

Basic statements

```
SELECT column(s) FROM table WHERE condition(s);            SQL
SELECT name, population FROM cities WHERE country_code = "FSM";
```

name	population
Weno	22000
Palikir	8600

- the WHERE portion of a SELECT statement can use the following operators:
  - =, >, >=, <, <=
  - <> : not equal
  - BETWEEN *min* AND *max*
  - LIKE *pattern*
  - IN (*value*, *value*, ..., *value*)

Sorting by a column: ORDER BY

```
ORDER BY column(s)                                    SQL
```

```
SELECT code, name, population FROM countries
WHERE name LIKE 'United%' ORDER BY population;        SQL
```

code	name	population
UMI	United States Minor Outlying Islands	0
ARE	United Arab Emirates	2441000
GBR	United Kingdom	59623400
USA	United States	278357000

- can write ASC or DESC to sort in ascending (default) or descending order:

```
SELECT * FROM countries
ORDER BY population
DESC;                          SQL
```

- can specify multiple orderings in decreasing order of significance:

```
SELECT * FROM countries ORDER BY population DESC, gnp; SQL
```

Limiting rows: LIMIT

```
LIMIT number                                        SQL
```

```
SELECT name FROM cities WHERE name LIKE 'K%' LIMIT 5;   SQL
```

name
Kabul
Khulna
Kingston upon Hull
Koudougou
Kafr al-Dawwar

- can be used to get the top-N of a given category (`ORDER BY` and `LIMIT`)
- also useful as a sanity check to make sure your query doesn't return 10^7 rows

Related tables and keys

id	name	email
123	Bart	bart@fox.com
456	Milhouse	milhouse@fox.com
888	Lisa	lisa@fox.com
404	Ralph	ralph@fox.com

students

id	name
1234	Krabappel
5678	Hoover
9012	Obourn

teachers

id	name	teacher_id
10001	Computer Science 142	1234
10002	Computer Science 143	5678
10003	Computer Science 154	9012
10004	Informatics 100	1234

courses

student_id	course_id	grade
123	10001	B-
123	10002	C
456	10001	B+
888	10002	A+
888	10003	A+
404	10004	D+

grades

- **primary key**: a column guaranteed to be unique for each record (e.g. Lisa Simpson's ID 888)
- **foreign key**: a column in table A storing a primary key value from table B
  - (e.g. records in `grades` with `student_id` of 888 are Lisa's grades)
- **normalizing**: splitting tables to improve structure / redundancy (linked by unique IDs)

Querying multi-table databases

When we have larger datasets spread across multiple tables, we need queries that can answer high-level questions such as:

- What courses has Bart taken and gotten a B- or better?
- What courses have been taken by both Bart and Lisa?
- Who are all the teachers Bart has had?
- How many total students has Ms. Krabappel taught, and what are their names?

To do this, we'll have to **join** data from several tables in our SQL queries.

Joining with ON clauses

```
SELECT column(s)
FROM table1
JOIN table2 ON condition(s)
...
JOIN tableN ON condition(s);                                    SQL
```

```
SELECT *
FROM students
JOIN grades ON id = student_id;                                 SQL
```

- **join**: combines records from two or more tables if they satisfy certain conditions
- the ON clause specifies which records from each table are matched
- the rows are often linked by their **key** columns (id)

Join example

```sql
SELECT *
FROM students
JOIN grades ON id = student_id;
```

SQL

id	name	email	**student_id**	course_id	grade
123	Bart	bart@fox.com	**123**	10001	B-
123	Bart	bart@fox.com	**123**	10002	C
404	Ralph	ralph@fox.com	**404**	10004	D+
456	Milhouse	milhouse@fox.com	**456**	10001	B+
888	Lisa	lisa@fox.com	**888**	10002	A+
888	Lisa	lisa@fox.com	**888**	10003	A+

table.*column* can be used to disambiguate column names:

```sql
SELECT *
FROM students
JOIN grades ON students.id = grades.student_id;
```

SQL

Filtering columns in a join

```
SELECT name, course_id, grade
FROM students
JOIN grades ON id = student_id;
```

SQL

name	course_id	grade
Bart	10001	B-
Bart	10002	C
Ralph	10004	D+
Milhouse	10001	B+
Lisa	10002	A+
Lisa	10003	A+

Filtered join (JOIN with WHERE)

```sql
SELECT name, course_id, grade
FROM students
JOIN grades ON id = student_id
WHERE name = 'Bart';
```

SQL

name	course_id	grade
Bart	10001	B-
Bart	10002	C

- FROM / JOIN glue the proper tables together, and WHERE filters the results
- what goes in the ON clause, and what goes in WHERE?
  - ON directly links columns of the joined tables
  - WHERE sets additional constraints such as particular values (123, 'Bart')

What's wrong with this?

```
SELECT name, id, course_id, grade
FROM students
JOIN grades ON id = 123
WHERE id = student_id;                                      SQL
```

name	id	course_id	grade
Bart	123	10001	B-
Bart	123	10002	C

- The above query produces the same rows as the previous one, but it is poor style. Why?

- The `JOIN ON` clause is poorly chosen. It doesn't really say what connects a `grades` record to a `students`record.
  - They are related when they are for a student with the same `id`.
  - Filtering out by a specific ID or name should be done in the `WHERE` clause, not `JOIN ON`.

Giving names to tables

```
SELECT s.name, g.*
FROM students s
JOIN grades g ON s.id = g.student_id
WHERE g.grade <= 'C';                                             SQL
```

name	student_id	course_id	grade
Bart	123	10001	B-
Bart	123	10002	C
Milhouse	456	10001	B+
Lisa	888	10002	A+
Lisa	888	10003	A+

- can give names to tables, like a variable name in Java
- to specify all columns from a table, write *table*`.*`
- (`grade` column sorts alphabetically, so grades C or better are ones `<=` it)

Multi-way join

```sql
SELECT c.name
FROM courses c
JOIN grades g ON g.course_id = c.id
JOIN students bart ON g.student_id = bart.id
WHERE bart.name = 'Bart' AND g.grade <= 'B-';
```

SQL

name
Computer Science 142

- More than 2 tables can be joined, as shown above
- What does the above query represent?

- The names of all courses in which Bart has gotten a B- or better.

A suboptimal query

Exercise: What courses have been taken by both Bart and Lisa?

```
SELECT bart.course_id
FROM grades bart
JOIN grades lisa ON lisa.course_id = bart.course_id
WHERE bart.student_id = 123
AND lisa.student_id = 888;
```

SQL

- problem: requires us to know Bart/Lisa's Student IDs, and only spits back course IDs, not names.

- Write a version of this query that gets us the course *names*, and only requires us to know Bart/Lisa's names, not their IDs.

Improved query

What courses have been taken by both Bart and Lisa?

```sql
SELECT DISTINCT c.name
FROM courses c
JOIN grades g1 ON g1.course_id = c.id
JOIN students bart ON g1.student_id = bart.id
JOIN grades g2 ON g2.course_id = c.id
JOIN students lisa ON g2.student_id = lisa.id
WHERE bart.name = 'Bart'
AND lisa.name = 'Lisa';
```

SQL

Practice queries

- What are the names of all teachers Bart has had?

```
SELECT DISTINCT t.name
FROM teachers t
JOIN courses c ON c.teacher_id = t.id
JOIN grades g ON g.course_id = c.id
JOIN students s ON s.id = g.student_id
WHERE s.name = 'Bart';
```

SQL

- How many total students has Ms. Krabappel taught, and what are their names?

```
SELECT DISTINCT s.name
FROM students s
JOIN grades g ON s.id = g.student_id
JOIN courses c ON g.course_id = c.id
JOIN teachers t ON t.id = c.teacher_id
WHERE t.name = 'Krabappel';
```

SQL

Designing a query

- Figure out the proper SQL queries in the following way:
  - Which table(s) contain the critical data? (`FROM`)
  - Which columns do I need in the result set? (`SELECT`)
  - How are tables connected (`JOIN`) and values filtered (`WHERE`)?
- Test on a small data set (`imdb_small`).
- Confirm on the real data set (`imdb`).
- Try out the queries first in the query tester.
- Write the PHP code to run those same queries.
  - Make sure to check for SQL errors at every step!!

Example imdb database

id	first_name	last_name	gender
433259	William	Shatner	M
797926	Britney	Spears	F
831289	Sigourney	Weaver	F
...			

actors

id	name	year	rank
112290	Fight Club	1999	8.5
209658	Meet the Parents	2000	7
210511	Memento	2000	8.7
...			

movies

actor_id	movie_id	role
433259	313398	Capt. James T. Kirk
433259	407323	Sgt. T.J. Hooker
797926	342189	Herself
...		

roles

movie_id	genre
209658	Comedy
313398	Action
313398	Sci-Fi
...	

movies_genres

id	first_name	last_name
24758	David	Fincher
66965	Jay	Roach
72723	William	Shatner
...		

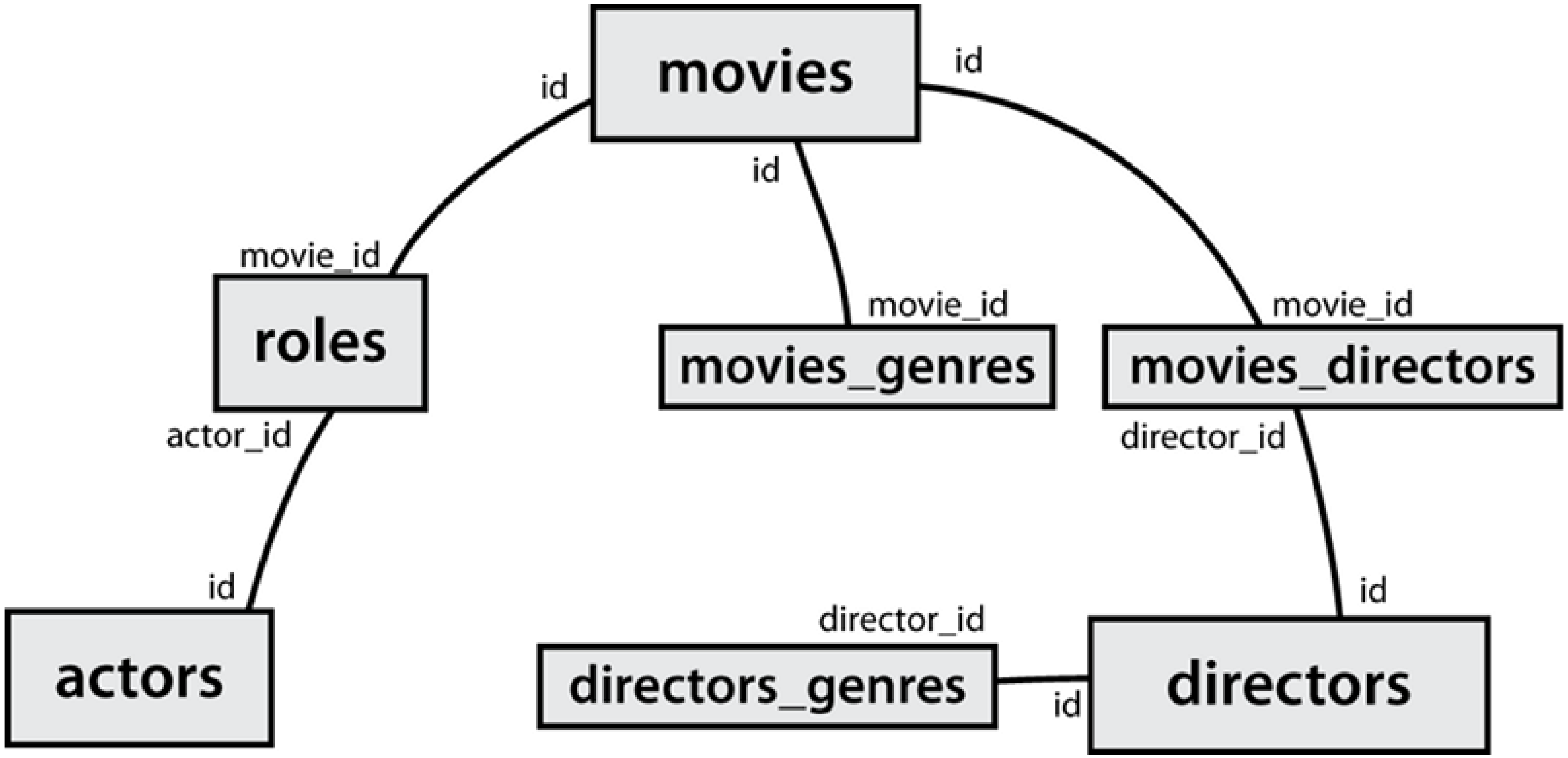
directors

director_id	movie_id
24758	112290
66965	209658
72723	313398
...	

movies_directors

- also available, `imdb_small` with fewer records (for testing queries)

IMDb table relationships / ids



IMDb practice queries

- What are the names of all movies released in 1995?
- How many people played a part in the movie "Lost in Translation"?
- What are the *names* of all the people who played a part in the movie "Lost in Translation"?
- Who directed the movie "Fight Club"?
- How many movies has Clint Eastwood directed?
- What are the *names* of all movies Clint Eastwood has directed?
- What are the names of all directors who have directed at least one horror film?
- What are the names of every actor who has appeared in a movie directed by Christopher Nolan?