CSE 154

LECTURE 24: MULTI-TABLE SQL QUERIES (JOINS)
Exceptions for errors

```
$db = new PDO("mysql:dbname=imdb_small", "jessica", "guinness");
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$rows = $db->query("SELECT * FROM movies WHERE year = 2000");
# kaboom!
```

• using `setAttribute`, you can tell PDO to throw (generate) a PDOException when an error occurs

• the exceptions will appear as error messages on the page output

• you can `catch` the exception to gracefully handle the error
### Related tables and keys

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>email</th>
<th>id</th>
<th>name</th>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>Bart</td>
<td><a href="mailto:bart@fox.com">bart@fox.com</a></td>
<td>1234</td>
<td>Krabappel</td>
<td>123</td>
<td>10001</td>
<td>B-</td>
</tr>
<tr>
<td>456</td>
<td>Milhouse</td>
<td><a href="mailto:milhouse@fox.com">milhouse@fox.com</a></td>
<td>5678</td>
<td>Hoover</td>
<td>123</td>
<td>10002</td>
<td>C</td>
</tr>
<tr>
<td>888</td>
<td>Lisa</td>
<td><a href="mailto:lisa@fox.com">lisa@fox.com</a></td>
<td>9012</td>
<td>Obourn</td>
<td>123</td>
<td>10003</td>
<td>B+</td>
</tr>
<tr>
<td>404</td>
<td>Ralph</td>
<td><a href="mailto:ralph@fox.com">ralph@fox.com</a></td>
<td></td>
<td></td>
<td>123</td>
<td>10004</td>
<td>D+</td>
</tr>
</tbody>
</table>

#### students

- **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)
Giving names to tables

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

<table>
<thead>
<tr>
<th>name</th>
<th>student_id</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bart</td>
<td>123</td>
<td>10001</td>
<td>B-</td>
</tr>
<tr>
<td>Bart</td>
<td>123</td>
<td>10002</td>
<td>C</td>
</tr>
<tr>
<td>Milhouse</td>
<td>456</td>
<td>10001</td>
<td>B+</td>
</tr>
<tr>
<td>Lisa</td>
<td>888</td>
<td>10002</td>
<td>A+</td>
</tr>
<tr>
<td>Lisa</td>
<td>888</td>
<td>10003</td>
<td>A+</td>
</tr>
</tbody>
</table>

- 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)
A suboptimal query

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

```sql
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;
```

- 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';
```
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';
```

• 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';
```
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 tool.
- Write the PHP code to run those same queries.
  - Make sure to check for SQL errors at every step!!
### Example imdb database

<table>
<thead>
<tr>
<th>id</th>
<th>first_name</th>
<th>last_name</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>433259</td>
<td>William</td>
<td>Shatner</td>
<td>M</td>
</tr>
<tr>
<td>797926</td>
<td>Britney</td>
<td>Spears</td>
<td>F</td>
</tr>
<tr>
<td>831289</td>
<td>Sigourney</td>
<td>Weaver</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>year</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>112290</td>
<td>Fight Club</td>
<td>1999</td>
<td>8.5</td>
</tr>
<tr>
<td>209658</td>
<td>Meet the Parents</td>
<td>2000</td>
<td>7</td>
</tr>
<tr>
<td>210511</td>
<td>Memento</td>
<td>2000</td>
<td>8.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>actor_id</th>
<th>movie_id</th>
<th>role</th>
</tr>
</thead>
<tbody>
<tr>
<td>433259</td>
<td>313398</td>
<td>Capt. James T. Kirk</td>
</tr>
<tr>
<td>433259</td>
<td>407323</td>
<td>Sgt. T.J. Hooker</td>
</tr>
<tr>
<td>797926</td>
<td>342189</td>
<td>Herself</td>
</tr>
</tbody>
</table>

### movies_genres
- 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?