

In class activity:

Given the displayed database, answer the following:

1. What is “wrong” with this “database”? Provide at least one concrete example and describe why this could be a problem. NOTE: This is a small example; think about if there were 1000 records and you wanted to send a card to everyone who’s birthday was in May.
2. What are some ways that you would want to “fix” what’s wrong?

Lab: Let's Try Some SQL

In this lab, you will try out some SQL queries using a postgresql database, imdb which contains movie information. The tables (and fields) we will work with are:

```
movie (id, title, kind_id, year)
movie_info (id, movie_id, info_id, info, note)
info_type(id, info)
kind_type(id, kind)
```

Part 1 – Warm-up

Clauses:

SELECT (chooses the attributes)
FROM (chooses the table)
WHERE (chooses the tuples)
LIMIT (limits the display to the first n entries, necessary with this very large db)

Try these queries exactly as written.

```
SELECT * FROM kind_type;
```

This query will show you everything in the kind_type table. You can see from this that there are many kinds of media in the movie table. Jot down the code for “movie”.

```
SELECT info FROM info_type;
```

This query will show you everything in the info_type table. You can see from this that there is a lot of different kinds of information in the movie_info table.

```
SELECT title, year FROM movie WHERE kind_id = 1 LIMIT 100;
```

This query will show you the title and year for the first 100 movies in the database.

Now, you try it:

Q1 - Write a query that will list the id and titles of the movies from 1955. (HINT: You can have two criteria in a WHERE clause if you use AND or OR to combine them. For example, WHERE z < q AND x = 12) What query did you write?

Q2 - Choose one of the movie ids from query 1 and write a query that will list the information (movie_info) for that movie only. What query did you write?

Q3 - Write a query to show all of the movie information for movie, 2139191. What query did you write?

Taking a look at the information for that movie, can you tell what the info_ids mean?

Part 2 – Joins

In the last example, query Q3, There was a lot of information, but we don't necessarily know what the id codes indicate. It would be useful to show information from both tables, movie_info and info_type in the same query. JOINS let us do that.

Clauses:

INNER JOIN – Both tables from the JOIN must have a row for the data to display.

LEFT OUTER JOIN – The leftmost table must have a row for the corresponding right hand table to display. If there is data in the left table, but not in the right, the data from the left only will display.

Try these queries using the JOINS exactly as written:

```
SELECT movie_id, info_type.info, movie_info.info
FROM movie_info INNER JOIN info_type
ON info_id = info_type.id WHERE movie_id = 2139191
```

Now we can see what the different categories of information are.

Try another:

```
SELECT title, year, info
FROM movie LEFT OUTER JOIN movie_info
ON movie.id = movie_info.movie_id
WHERE info_id = 2 AND year = 1955 AND kind_type = 1
```

How many movies in 1955 don't have color info? The above query will only give us those with color info because we are querying that value. Try this one.

```
SELECT title, year, info
FROM movie LEFT OUTER JOIN movie_info
ON movie.id = movie_info.movie_id
WHERE (info_id = 2 OR info_id IS NULL) AND year = 1955 AND kind_type = 1
```

This gives us all movies from 1955 with color information if it is present.

Text matching:

The LIKE clause let's us match characters in a WHERE clause. A '?' symbol matches precisely one character while '%' matches 0 or more characters.

WHERE name LIKE '%John%' would match John, Johnson, Johns, Van Johns, etc.

Try these queries:

Q4 – Write a query to find all of the movies with the word 'Balloon' in their name. What query did you write?

Q5 – Write a query to find all black and white movies from 1963. What query did you write?

Q6 – Write a query to find the titles of all movies whose language is French from 1982. What query did you write?