Project 4: Group Work
Group Fight Club

Step 1: Datacleaning

Users and zip codes
63 entries from the **user** table didn't reference existing zip codes in the **zipcode** table. Therefore we removed all users referencing non-existing zip codes. Before deleting we created an index on **zipcode.zip**.

```
CREATE INDEX zip_index ON zipcode (zip);

DELETE FROM user WHERE zip NOT IN (SELECT zip FROM zipcode);
Query OK, 63 rows affected (0.02 sec)
```

Ratings and user ids
10360 entries from the **rating** table referenced user ids which didn't exist in the **user** table. Therefore we removed all ratings referencing non-existing user ids. Before deleting entries we created an index on **user.id**.

```
CREATE INDEX user_id_index ON user (id);

DELETE FROM rating WHERE userId not in (SELECT id FROM user);
Query OK, 10360 rows affected (1.31 sec)
```

User age
In the data set, age 1 means Under 18, age 18 means 18-24, age 25 means 25-34, etc., up to 56 which means 56+. We choose to change each age group to the average age for that age group (eg. 18 to 24 averages to 21) as following:

1 (0-17): 9
18 (18-24): 21
25 (25-34): 30
35 (35-44): 39
45 (45-49): 47
50 (50-55): 53
56 (56-84): 70
(we rounded up where necessary)

```
UPDATE user SET age=9 WHERE age=1;
UPDATE user SET age=21 WHERE age=18;
UPDATE user SET age=30 WHERE age=25;
UPDATE user SET age=39 WHERE age=35;
UPDATE user SET age=47 WHERE age=45;
UPDATE user SET age=53 WHERE age=50;
UPDATE user SET age=70 WHERE age=56;
```

Step 2: Enriching

```
CREATE TABLE incomes (zip char(5), median double, mean double, pop
double);

LOAD DATA LOCAL INFILE 'MedianZIP-3.csv' INTO TABLE incomes;
Query OK, 32635 rows affected, 65535 warnings (0.14 sec)
```

Step 3: OLAP

Pre-aggregated table #1: Popularity by movie by genre:

CREATE TABLE popularity_by_movie_by_genre
SELECT COUNT(rating) as popularity, genre.name as genre_name, movie.title as movie_title
FROM rating
JOIN movieGenre
ON movieGenre.movieId=rating.movieId
JOIN movie
ON movie.id=movieGenre.movieId
JOIN genre
ON movieGenre.genreId=genre.id
GROUP BY genre.name, movie.title;

Query OK, 6190 rows affected (5.45 sec)

```
mysql> SELECT * FROM popularity_by_movie_by_genre LIMIT 10;
+------------+------------+-------------------------------------------+
| popularity | genre_name | movie_title                               |
+------------+------------+-------------------------------------------+
|        744 | Action     | 13th Warrior, The (1999)                  |
|         46 | Action     | 3 Ninjas: High Noon On Mega Mountain (1998) |
|        140 | Action     | 52 Pick-Up (1986)                         |
|        252 | Action     | 7th Voyage of Sinbad, The (1958)          |
|       1700 | Action     | Abyss, The (1989)                         |
|        125 | Action     | Aces: Iron Eagle III (1992)               |
|        184 | Action     | Action Jackson (1988)                     |
|         10 | Action     | Adrenalin: Fear the Rush (1996)           |
|        375 | Action     | Adventures of Robin Hood, The (1938)      |
|       1049 | Action     | African Queen, The (1951)                 |
+------------+------------+-------------------------------------------+
```

```
CREATE INDEX popularity_aggregation_index
ON popularity_by_movie_by_genre (popularity);
```

Project 4: Group Work
Group Fight Club

Usage scenario:

Which movie is the most popular in each genre:

```
SELECT y.popularity, y.genre_name, movie_title
FROM (
      SELECT max(popularity) AS max_pop, genre_name
      FROM popularity_by_movie_by_genre
      GROUP BY genre_name
) AS x INNER JOIN popularity_by_movie_by_genre AS y
ON x.max_pop = y.popularity
AND x.genre_name = y.genre_name;
```

```
+------------+-------------+-----------------------------------------------------+
| popularity | genre_name  | movie_title                                         |
+------------+-------------+-----------------------------------------------------+
|       2964 | Action      | Star Wars: Episode V - The Empire Strikes Back (1980) |
|       2964 | Adventure   | Star Wars: Episode V - The Empire Strikes Back (1980) |
|       2061 | Animation   | Toy Story (1995)                                    |
|       2244 | Children's  | E.T. the Extra-Terrestrial (1982)                   |
|       3391 | Comedy      | American Beauty (1999)                              |
|       2488 | Crime       | Fargo (1996)                                        |
|        791 | Documentary | Roger & Me (1989)                                   |
|       3391 | Drama       | American Beauty (1999)                              |
|       2960 | Fantasy     | Star Wars: Episode IV - A New Hope (1977)           |
|       2267 | Film-Noir   | L.A. Confidential (1997)                            |
|       2159 | Horror      | Ghostbusters (1984)                                 |
|       1704 | Musical     | Wizard of Oz, The (1939)                            |
|       2267 | Mystery     | L.A. Confidential (1997)                            |
|       2855 | Romance     | Star Wars: Episode VI - Return of the Jedi (1983)   |
|       2964 | Sci-Fi      | Star Wars: Episode V - The Empire Strikes Back (1980) |
|       2623 | Thriller    | Terminator 2: Judgment Day (1991)                   |
|       2964 | War         | Star Wars: Episode V - The Empire Strikes Back (1980) |
|       1438 | Western     | Dances with Wolves (1990)                           |
+------------+-------------+-----------------------------------------------------+
18 rows in set (0.00 sec)
```

What are the genres of the top 5 most popular movies:

```
SELECT genre_name FROM popularity_by_movie_by_genre ORDER BY
popularity DESC LIMIT 5;
```

```
+------------+
| genre_name |
+------------+
| Drama      |
| Comedy     |
| Adventure  |
| Drama      |
| Action     |
+------------+
```

Pre-aggregated table #2: Average rating by occupation by genre:

```
CREATE TABLE average_rating_by_occ_by_genre
SELECT AVG(rating.rating) as avg_rating, genre.name as genre_name,
occupation.description as occupation
FROM rating
JOIN movieGenre
ON movieGenre.movieId=rating.movieId
JOIN genre
ON movieGenre.genreId=genre.id
JOIN user
ON rating.userId=user.id
JOIN occupation
ON user.occupation=occupation.id
GROUP BY genre.name, occupation.description;

Query OK, 378 rows affected (7.02 sec)
```

```
mysql> SELECT * FROM average_rating_by_occ_by_genre LIMIT 10;
+------------+------------+----------------------+
| avg_rating | genre_name | occupation           |
+------------+------------+----------------------+
|     3.3995 | Action     | academic/educator    |
|     3.4574 | Action     | artist               |
|     3.5487 | Action     | clerical/admin       |
|     3.4531 | Action     | college/grad student |
|     3.4930 | Action     | customer service     |
|     3.5639 | Action     | doctor/health care   |
|     3.5686 | Action     | executive/managerial |
|     3.4579 | Action     | farmer               |
|     3.6015 | Action     | homemaker            |
|     3.5129 | Action     | K-12 student         |
+------------+------------+----------------------+
```

```
CREATE INDEX average_rating_aggregation_index
ON average_rating_by_occ_by_genre (avg_rating);
```

Project 4: Group Work
Group Fight Club

Usage scenarios:

Find the 5 occupations that give the highest average rating to western movies.

```
SELECT occupation FROM average_rating_by_occ_by_genre WHERE
genre_name="Western" ORDER BY avg_rating DESC LIMIT 5;
```

```
+----------------------+
| occupation           |
+----------------------+
| farmer               |
| scientist            |
| programmer           |
| retired              |
| tradesman/craftsman |
+----------------------+
```

Find which occupation gives each genre the highest average rating.

```
SELECT y.avg_rating, y.genre_name, occupation
FROM (
      SELECT max(avg_rating) AS max_avg, genre_name
      FROM average_rating_by_occ_by_genre
      GROUP BY genre_name
) AS x INNER JOIN average_rating_by_occ_by_genre AS y
ON x.max_avg = y.avg_rating
AND x.genre_name = y.genre_name;
```

```
+------------+-------------+----------------+
| avg_rating | genre_name  | occupation     |
+------------+-------------+----------------+
|     3.6793 | Action      | retired        |
|     3.6890 | Adventure   | homemaker      |
|     3.8477 | Animation   | scientist      |
|     3.6674 | Children's  | homemaker      |
|     3.6845 | Comedy      | scientist      |
|     3.8325 | Crime       | retired        |
|     4.1414 | Documentary | lawyer         |
|     3.9481 | Drama       | retired        |
|     3.6780 | Fantasy     | retired        |
|     4.2130 | Film-Noir   | K-12 student   |
|     3.3479 | Horror      | programmer     |
|     3.8570 | Musical     | clerical/admin |
|     3.9436 | Mystery     | retired        |
|     3.8243 | Romance     | retired        |
|     3.5599 | Sci-Fi      | programmer     |
|     3.8026 | Thriller    | retired        |
|     4.0844 | War         | retired        |
|     3.8286 | Western     | farmer         |
+------------+-------------+----------------+
```