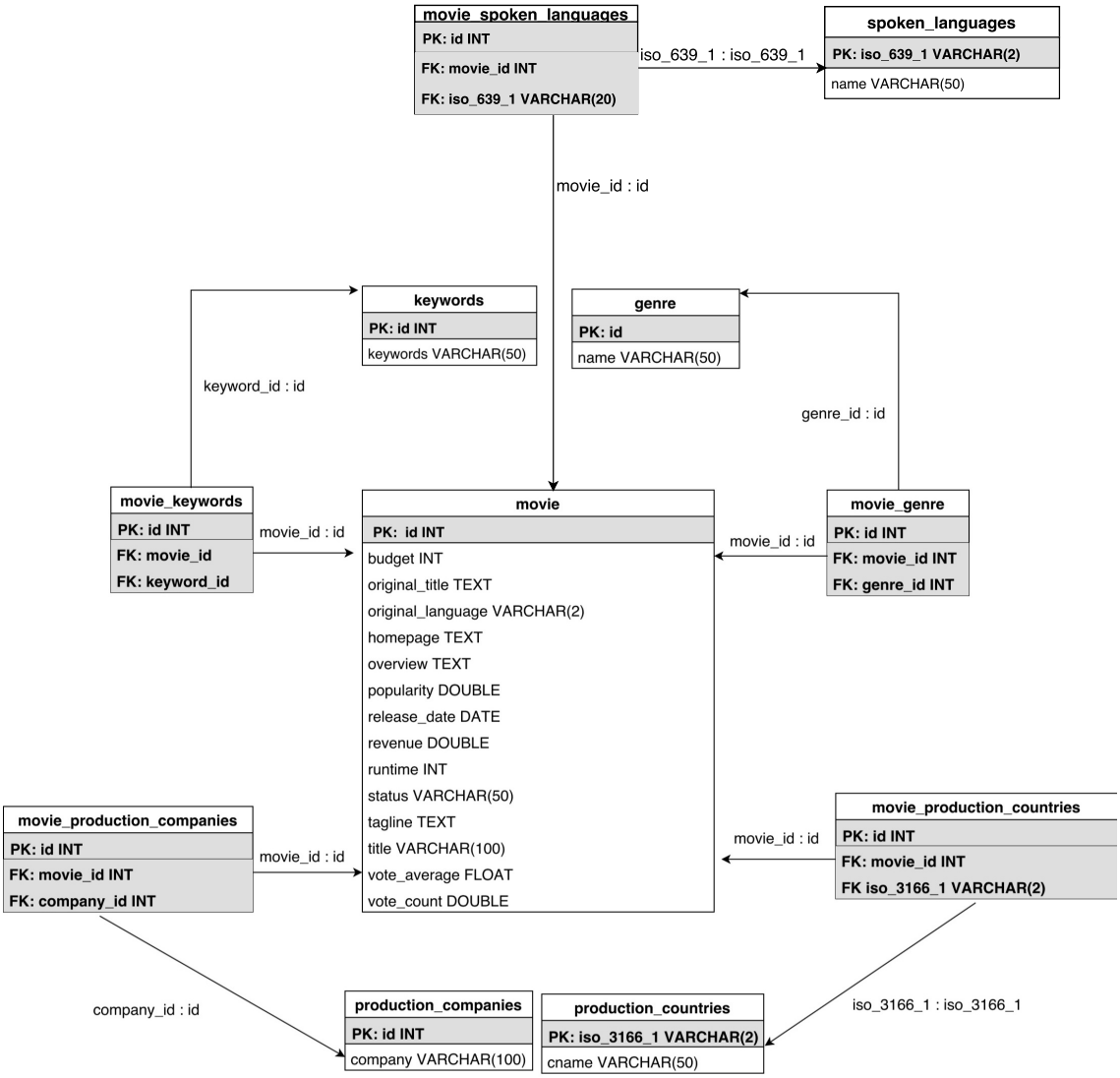
1 Data



Above are all the schemas and their relationships with other schemas. PK stands for primary key and FK stands for foreign key. All the entries in the movie csv that where not atomic where broken down into two separate tables. Since atomic columns like **genre** and **production_companies** had a many-to-many relationship with the movie table, the first table was the relationship table and the second table contained specific information like genre id and genre name. All tables are in BCNF because (1) all attributes are atomic, (2) all non-key attributes depend only on the candidate key, (3) and in all $X \rightarrow Y$ FD's, X is a candidate key. For example, all attributes in the **movie** table only depend on the candidate key id and all attributes in **movie_genre** table only depend on the candidate key id. This satisfies BCNF condition because the attribute on the right side is a candidate key.

Refer to figure above for table attributes, primary key, and foreign keys.

1. **movie :** Contains all the columns that where atomic in the csv file. This table is in BCNF because in the **movie** FD $X \rightarrow A, B, C, D, E, F, G, H, I, J, K, L, M, N$ the left side X is a candidate key **id**. This satisfies the BCNF condition.

2. **movie_genre :** Relationship table between many-to-many relationship of **genre** and **movie**. Contains a unique **id**, **movie_id**, and **genre_id** where **movie_id** and **genre_id** reference **id** in **id** in **movie** and **id** in **id** in **genre** respectively. This table is in BCNF because in the FD $X \rightarrow Y, Z$ X is the candidate key. This satisfies the BCNF condition.

3. **genre :** Contains all the genre names and their id that where found in the genre column from all rows. **movie_genre** references this table. Also in BCNF because the FD $X \rightarrow Z$ satisfies BCNF condition because X is a candidate key.

4. **spoken_languages :** Contains all the spoken languages id's and names that were encountered in the csv. Also in BCNF, see **genre** for explanation.

5. **movie_spoken_languages :** Relationship table for the many-to-many relationship between **spoken_languages** and **movie**. In BCNF. See **movie_genre** for explanation.

6. **keywords :** Contains all keyword names and id's encountered in csv. In BCNF. See **genre** for explanation.

7. **movie_keywords :** All keywords and their repective id's that where encountered in the csv where stored here. In BCNF. See **movie_genre** for explanation.

8. **production_countries :** Contains the countries id and names that where encountered in the csv. In BCNF, see **genre :** for explanantion.

9. **movie_production_countries :** Relationship table for the many-to-many relationship between **movie** and **production_countries**. In BCNF, see **movie_genre** for explanation.

10. **production_companies :** Contains all production companies and their respective id's that where encountered in the **production_companies** row of csv file. Similar to **genre** also in BCNF.

11. **movie_production_companies :** Relationship table for the many-to-many relationship between **movie** and **production_companies**. Like all other relationship tables it is also in BCNF.

2 Queries

1. ```
   SELECT avg(budget) FROM movie;
   ```

29045039.8753

2. ```
   SELECT title, company
   FROM
       (
           SELECT title, company, iso_3166_1
           FROM movie
           INNER JOIN movie_production_countries
           ON movie.id=movie_production_countries.movie_id
           INNER JOIN movie_production_companies
           ON movie_production_companies.movie_id=movie_production_countries.
               movie_id
           INNER JOIN production_companies company
           ON movie_production_companies.company_id = company.id
       ) AS full
   WHERE full.iso_3166_1='US';
   ```

Four Rooms	Miramax Films
Four Rooms	A Band Apart
Star Wars	Lucasfilm
Star Wars	Twentieth Century Fox Film Corporation
Finding Nemo	Pixar Animation Studios

3. ```
   SELECT title, revenue
   FROM movie
   ORDER BY revenue DESC LIMIT 5;
   ```

Avatar	2787965087
Titanic	1845034188
The Avengers	1519557910
Jurassic World	1513528810
Furious 7	1506249360
   ```

4.
```
SELECT mys.title, genre.name
FROM
  (
      SELECT title, g2.name, g.movie_id
      FROM movie
        INNER JOIN movie_genre g ON movie.id = g.movie_id
        INNER JOIN genre g2 ON g.genre_id = g2.id
      WHERE g2.name="Science Fiction"
  ) AS sci
  INNER JOIN
    (
      SELECT title, g2.name, g.movie_id
      FROM movie
        INNER JOIN movie_genre g ON movie.id = g.movie_id
        INNER JOIN genre g2 ON g.genre_id = g2.id
      WHERE g2.name = "Mystery"
    ) AS mys
  ON sci.movie_id=mys.movie_id
  INNER JOIN movie_genre ON movie_genre.movie_id=mys.movie_id
  INNER JOIN genre ON genre.id=movie_genre.genre_id;
```

Tomorrowland	Adventure
Tomorrowland	Family
Tomorrowland	Mystery
Tomorrowland	Science Fiction
Inception	Action

5.
```
SELECT title, popularity
FROM movie
WHERE popularity > (SELECT avg(popularity) FROM movie);
```

Four Rooms	22.87623
Star Wars	126.393695
Finding Nemo	85.688789
Forrest Gump	138.133331
American Beauty	80.878605