```
--Load movies table
CREATE EXTERNAL TABLE IF NOT EXISTS movies (
  movieId INT,
  title STRING,
  genres STRING,
  year INT)
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
  LINES TERMINATED BY '\n'
  LOCATION 's3://bdfprojectv02/movies/';

--Load movies ratings
CREATE EXTERNAL TABLE IF NOT EXISTS ratings (
  userId INT,
  movieId INT,
  rating FLOAT,
  timestamp TIMESTAMP)
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
  LINES TERMINATED BY '\n'
  LOCATION 's3://bdfprojectv02/ratings/';

--Load movies tags
CREATE EXTERNAL TABLE IF NOT EXISTS tags (
userId INT,
movieId INT,
tags STRING,
timestamp TIMESTAMP)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
LOCATION 's3://bdfprojectv02/tags/';
```

```
--Load links table

CREATE EXTERNAL TABLE IF NOT EXISTS links (

movieId INT,

imdbId INT,

tmdbId INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

LOCATION 's3://bdfprojectv02/links/';


--Partition: year

CREATE TABLE movies_partitioned

WITH (format='PARQUET',
external_location='s3://bdfprojectv02/movies_partitioned/',
partitioned_by = ARRAY['year'])

AS SELECT movieid, title, genres, year

FROM movies

WHERE year >= 1920;


--Partition: rating

CREATE TABLE ratings_partitioned

WITH (format='PARQUET',
external_location='s3://bdfprojectv02/ratings_partitioned/',
partitioned_by = ARRAY['rating'])

AS SELECT userid, movieid, rating

FROM ratings;


--Partition: rating & Buckets (only for large dataset): movieid

CREATE TABLE ratingsl_partitioned

WITH (format='PARQUET',
external_location='s3://bdfprojectlargev02/ratingsl_partitioned/
', partitioned_by = ARRAY['rating'],

bucketed_by = ARRAY['movieid'], bucket_count = 7)

AS SELECT userid, movieid, rating

FROM ratings_large;
```

```
--Q1: What are the movie genres in the dataset?

CREATE TABLE IF NOT EXISTS genres

WITH (format='PARQUET',
external_location='s3://bdfprojectv02/genres/') AS

SELECT movieid, t.genres_u

FROM (SELECT movieid, "split"(genres, '|') genres FROM movies)

AS t CROSS JOIN UNNEST (genres) AS t(genres_u;


--Partition: genres_u

CREATE TABLE genres_partitions

WITH (format='PARQUET',
external_location='s3://bdfprojectv02/genres_partitions/',
partitioned_by = ARRAY['genres_u'])

AS SELECT movieid, genres_u

FROM genres;


--Q2: What are the genres with the highest average rating?

CREATE TABLE IF NOT EXISTS ratings_by_genre

WITH (format='PARQUET',
external_location='s3://bdfprojectv02/ratings_by_genre/') AS

SELECT genres.genres_u, AVG(ratings.rating) AS avg_rating

FROM ratings

JOIN genres ON ratings.movieid=genres.movieid

GROUP BY genres.genres_u ORDER BY AVG(ratings.rating) DESC;


--Q3: What are the genres with the highest average rating per
year?

CREATE TABLE IF NOT EXISTS avgrating_by_genre_year

WITH (format='PARQUET',
external_location='s3://bdfprojectv02/avgrating_by_genre_year/')
AS

SELECT movies.year, genres.genres_u, AVG(ratings.rating)

AS avg_rating

FROM movies
```

```
JOIN genres ON movies.movieid=genres.movieid

JOIN ratings ON movies.movieid=ratings.movieid

GROUP BY movies.year, genres.genres_u ORDER BY avg_rating DESC;


--Q4: What is the most common genre per year?

CREATE TABLE IF NOT EXISTS top_genre_year

WITH (format='PARQUET',
external_location='s3://bdfprojectv02/top_genre_year/') AS

SELECT movies.year, genres.genres_u, COUNT(genres.movieid) AS
no_movies

FROM movies

JOIN genres ON movies.movieid=genres.movieid

GROUP BY movies.year, genres.genres_u ORDER BY movies.year DESC,
COUNT(genres.movieid) DESC;


--Q5: How many ratings, tags and genres each movie has?

CREATE TABLE IF NOT EXISTS no_ratesandtagsandgenres_movies

WITH (format='PARQUET',
external_location='s3://bdfprojectv02/no_ratesandtagsandgenres_m
ovies/') AS

SELECT rates.movieid, rates.no_rates, COUNT(DISTINCT tags.tags)

AS no_tags, gen.no_genres FROM tags

LEFT JOIN (SELECT ratings.movieid, COUNT (ratings.rating) AS
no_rates FROM ratings GROUP by ratings.movieid) AS rates

ON rates.movieid=tags.movieid

LEFT JOIN (SELECT genres.movieid, COUNT (DISTINCT genres_u) AS
no_genres FROM genres GROUP by genres.movieid) AS gen

ON gen.movieid=tags.movieid

GROUP by rates.movieid, rates.no_rates, gen.no_genres ORDER by
rates.no_rates DESC;


--Q6: What are the most common tags per genre?

CREATE TABLE IF NOT EXISTS top_tags_genre

WITH (format='PARQUET',
external_location='s3://bdfprojectv02/top_tags_genre/') AS

SELECT genres.genres_u, tags.tags, COUNT(tags.tags) AS no_tags
```

```
FROM genres

JOIN tags ON tags.movieid=genres.movieid

GROUP BY genres.genres_u, tags.tags

ORDER BY genres.genres_u, no_tags DESC;


--Q7: What are the most common tags per year?

CREATE TABLE IF NOT EXISTS top_tags_year

WITH (format='PARQUET',
external_location='s3://bdfprojectv02/top_tags_year/') AS

SELECT movies.year, tags.tags, COUNT(tags.tags) AS no_tags

FROM movies

JOIN tags ON movies.movieid=tags.movieid

GROUP BY movies.year, tags.tags ORDER BY movies.year DESC,
no_tags DESC;


--Q8: What are the worst rated movies?

SELECT ratings.movieid, movies.title, COUNT(ratings.rating) FROM
movies

JOIN ratings ON movies.movieid=ratings.movieid

WHERE ratings.rating < 1

GROUP BY ratings.movieid, movies.title ORDER BY
COUNT(ratings.rating) DESC;


--Q9: What are the best rated movies?

SELECT ratings.movieid, movies.title, COUNT(ratings.rating) FROM
movies

JOIN ratings ON movies.movieid=ratings.movieid

WHERE ratings.rating > 4.5

GROUP BY ratings.movieid, movies.title ORDER BY
COUNT(ratings.rating) DESC;


--Q10: What are the top best rated comedy movies (on average)?

SELECT genres.movieid, AVG(ratings.rating) FROM genres

JOIN ratings ON genres.movieid=ratings.movieid

WHERE genres.genres_u = 'Comedy'
```

```
GROUP BY genres.movieid ORDER BY AVG(ratings.rating) DESC;

--Q11: What are the most common movie genres in the last 10
years of the dataset?

SELECT genres.genres_u, COUNT(genres.movieid) AS no_movies

FROM genres

JOIN movies ON movies.movieid=genres.movieid

WHERE movies.year >= 2008

GROUP BY genres.genres_u ORDER BY COUNT(genres.movieid) DESC;
```