CODE

```
-- Question 1


import psycopg2


def loadData():
    conn = None

    try:
        conn = psycopg2.connect(
            host='localhost',
            database='imdb6202',
            user='postgres',
            password='Abhishek@123',
            port=5432)

        cur = conn.cursor()

        loadMovies = """
        COPY (SELECT json_strip_nulls(row_to_json(res))
              FROM
                    (SELECT id as _id, type, title, originalTitle, startYear, endYear, runtime,
          avgRating, numVotes, genres, actors, director as directors, producer as producers, writer
as writers
                    FROM Title

                    LEFT JOIN
                    (
                    SELECT title as genreTitle, array_agg(Genre.genre) as genres
         FROM Title_Genre JOIN Genre
           ON Title_Genre.genre = Genre.id
         GROUP BY genreTitle
                    ORDER BY genreTitle
                    ) as g
                    ON Title.id = g.genreTitle

                    LEFT JOIN
```

```
                (
                SELECT title as directorTitle, array_agg(director) as director
        FROM Title_Director GROUP BY directorTitle) as d
                ON Title.id = d.directorTitle

                LEFT JOIN
                (
                SELECT title as producerTitle, array_agg(producer) as producer
        FROM Title_Producer GROUP BY producerTitle) as p
                ON Title.id = p.producerTitle

                LEFT JOIN
                (
                SELECT title as writerTitle, array_agg(writer) as writer
        FROM Title_Writer GROUP BY writerTitle) as w
                ON Title.id = w.writerTitle

                LEFT JOIN
                (
                SELECT atcTitle, json_agg(actors) as actors FROM
        (
                SELECT ta.atcTitle, json_build_object('actor', ta.actorid, 'roles', ta.roles) as
actors FROM
        (
                SELECT atc.title as atcTitle, atc.actor as actorid, array_agg(character.character)
as roles
        FROM character JOIN actor_title_character atc
          ON character.id = atc.character
        GROUP BY (atc.title, atc.actor)
                ORDER BY (atc.title, atc.actor)) as ta) AS result
        GROUP BY atcTitle) as t ON t.atcTitle = Title.id
                ) res)
        TO '/Users/abhishekshah/Downloads/Movies.json' with (FORMAT TEXT, HEADER
false);
        """
        cur.execute(loadMovies)
        print("Rows affected for Movies: ", cur.rowcount)

        loadMembers = """
        COPY (
             SELECT json_strip_nulls(row_to_json(res))
             FROM(
                  SELECT id as _id, name, birthyear, deathyear
                  FROM Member
                   )res)
```

```
        TO '/Users/abhishekshah/Downloads/Test2.json' with (FORMAT TEXT, HEADER false);
        """
        cur.execute(loadMembers)
        print("Rows affected for Members: ", cur.rowcount)
        cur.close()

    except(Exception, psycopg2.DatabaseError) as error:
        print(error)

    finally:
        if conn is not None:
            conn.close()
            print("Connection closed")


if __name__ == "__main__":
    loadData()
```

```
mongoimport --db imdb620 --collection Movies --file
/Users/abhishekshah/Downloads/Movies.json

mongoimport --db imdb620 --collection Members --file
/Users/abhishekshah/Downloads/Members.json
```

```
-- Question 2

--2.1
db.Movies.aggregate([
  {
    $match: {"startyear": {$ne:2014}}},
  {
    $lookup:
  {
    from: "Members",
    localField: "actors.actor",
    foreignField: "_id",
```

```
  as: "res"}},
 {
  $unwind: "$res"},
 {
  $match:{"res.name": /^Phi/, "res.deathyear": null}},
 {
  $project:{"_id":0,"res.name":1}}
  ]);

--2.2
db.Movies.aggregate([{
 $match:{"startyear": {$eq:2017}, "genres": {$eq: "Talk-Show"}}},
 {
 $lookup:
 {
 from: "Members",
 localField:"producers",
 foreignField:"_id",
 as:"new"}},
 {
  $unwind: "$new"},
 {
  $match:{"new.name":{"$regex":/Gill/}}},
 {
  $group: { _id: "producers", totalcount: {"$sum":1},
  producer:{$push:{name:"$new.name"}}}
 }
 ]);

--2.3
```

```
db.Movies.aggregate([
  {
    $lookup:
  {
    from: "Members",
    localField: "writers",
    foreignField: "_id",
    as: "res"}},
  {
    $match: {"res.name": /Bhardwaj/, "res.deathyear": null}},
  {
    $group: {"_id": null, "average": {"$avg": "$runtime"}}}
  ]);
```

--2.4

```
db.Movies.aggregate([
{
  $match: {"runtime": {$gt: 120}}},
  {
  $lookup:
  {
  from: "Members",
  localField: "producers",
  foreignField: "_id",
  as: "res"}},
  {
  $unwind: "$res"},
  {
  $match:{"res.deathyear": null}},
  {
  $sort:{"runtime": -1}},
  {
  $project:{"_id": 1,"res.name": 1}}
   ]);
```

--2.5

```
db.Movies.aggregate([
  {
  $match:{"genres": {"$eq": "Sci-Fi"}}},
  {
  $unwind: "$directors"},
  {
  $lookup:
    {
    from: "Members",
    localField: "directors",
    foreignField: "_id",
    as: "result"}
    },
          {
          $match:{"result.name": {"$eq": "James Cameron"}}},
          {
          $unwind: "$actors"},
    {
    $lookup:
    {
    from: "Members",
    localField: "actors.actor",
    foreignField: "_id",
    as: "result1"}
    },

    {
    $match:{"result1.name": {"$eq": "Sigourney Weaver"}}},
    {
```

```
    $project:{_id: 1, "title": 1},
  }
]);
```

Question 4

```
db.Members.createIndex(
  { name: 1}
)

db.Members.createIndex(
  { name: 1, deathYear: -1}
)

db.Movies.createIndex(
  { runtime: 1}
  )

db.Movies.createIndex(
{genre: -1}
)

db.Movies.createIndex(
  { startyear: -1}
)
```