



**Universidad Politécnica de Aguascalientes.**

**Database Administration**

# U1EP1.- OracleFlix_Project

**ISC06B**

**Students:**

**Uriel Isaac Vazquez Martínez UP210934**

**Juan Pablo López González UP200053**

**José Miguel Escalera Rubalcava UP200667**

**Derek Gilberto Ramírez López UP200424**

10/23/2022

# INDEX

# OracleFlix_Project

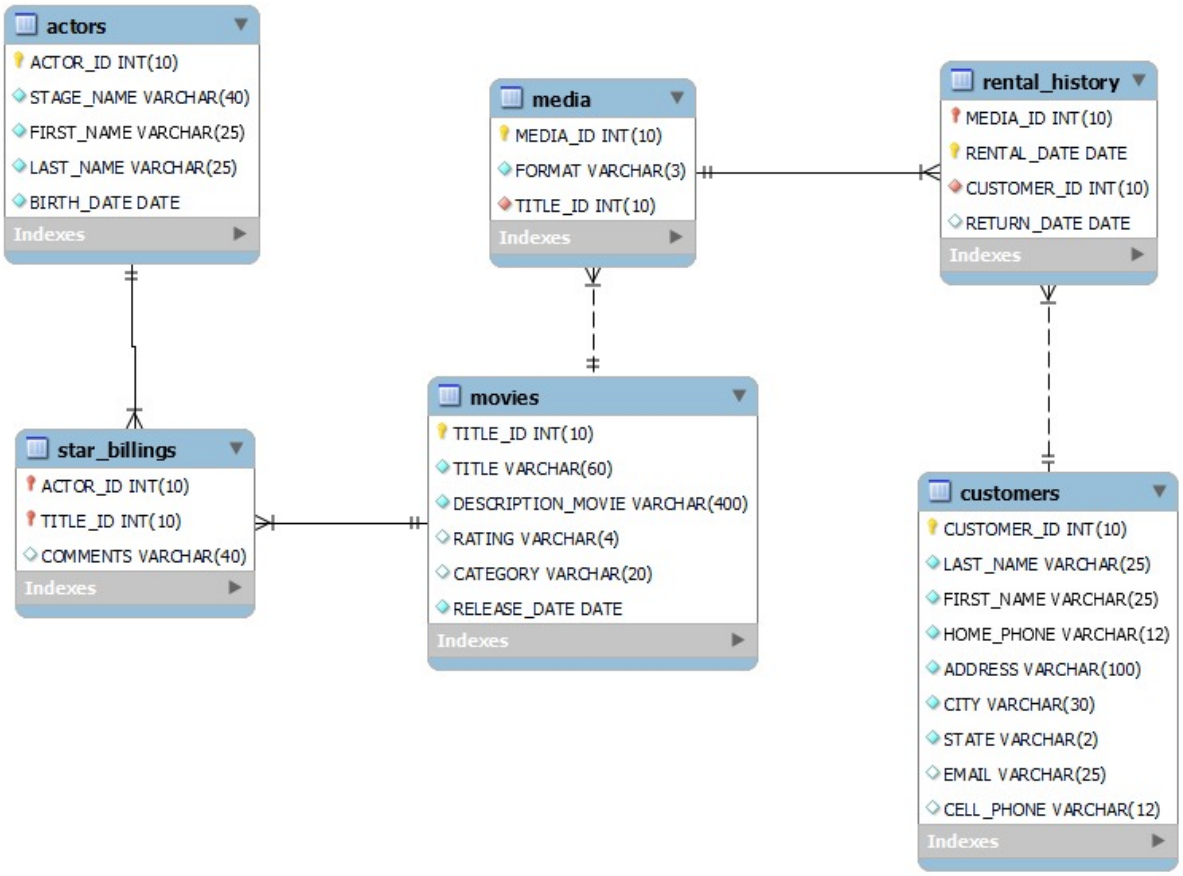
## Introduction

In order to demonstrate the acquired knowledge of database administration, specially of the languages DDL and DML a database called BD_FLIX was created.

DB_FLIX contains data related to movie rental. Includes 6 tables: movies, customers, actors, starbillings, media and rental_history. Each of them has constraints, some include sequences and indexes.

The following document includes the set of instructions to follow in order to create the OracleFlix database in Oracle SQL developer and in MariaDB, not to mention, the entity relationship diagram of the database.

## Entity Relationship Diagram



## Oracle

Instructions to create BD_FLIX database using Oracle SQL developer:

1) **Create the User:** To create the user BD_FLIX run the commands on the file: “create_user.sql”
2) **Create tables using the attached ERD:** Once connected to BD_FLIX database, run one by one all the CREATE instructions in the file “create_tables.sql”
3) **Add the following integrity constraints:**

   • Create primary key (PK) and foreign key (FK) constraints as needed per ERD

   • Create not null (NN) constraints where necessary as per ERD

   • Create check constraint on rating field in movie table to limit rating values to 'G', 'PG', 'R', 'PG13'

   • Create check constraint on category field in movie table to limit categories to 'DRAMA', 'COMEDY', 'ACTION', 'CHILD', 'SCIFI', 'DOCUMENTARY'

   • Run queries from the data dictionaries for the above constraints.

   Most of the constraints are created with the tables. To add the Foreign Key constraints, run the “ALTER TABLE” commands in the file “create_table.sql”

4) **Create a view called TITLE_UNAVAIL:** To create the view, run the "CREATE OR REPLACE VIEW.." command after all the foregin key constraints in the document "create_table.sql"

5) **Create the following sequences to be used for primary key values:**

   i) Use a sequence to generate PKs for CUSTOMER_ID in CUSTOMERS table
      (1) Begin at 101 and increment by 1

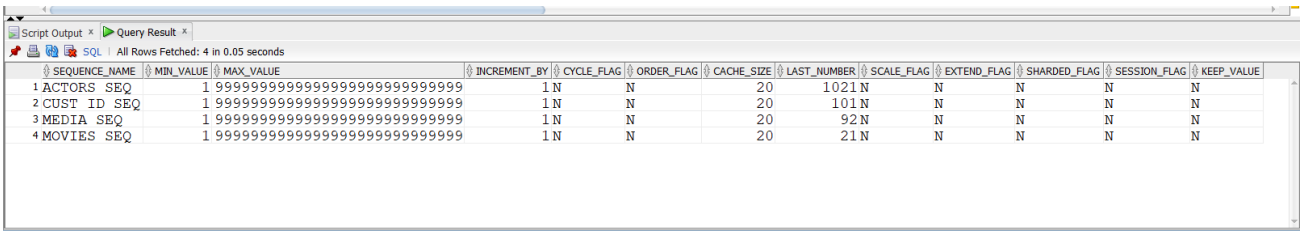
   ii) Use a sequence to generate PKs for TITLE_ID in MOVIES table.
      (1) Begin at 1 and increment by 1

   iii) Use a sequence to generate PKs for MEDIA_ID in MEDIA table
      (1) Begin at 92 and increment by 1

   iv) Use a sequence to generate PKs for ACTOR_ID in ACTOR table
      (1) Begin at 1001 and increment by 1

   v) Run queries from the data dictionaries for the above sequences.

To create the sequences, run the commands starting with "CREATE SEQUENCE…" in the file "create_tables.sql"


Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.05 seconds

| | SEQUENCE_NAME | MIN_VALUE | MAX_VALUE | INCREMENT_BY | CYCLE_FLAG | ORDER_FLAG | CACHE_SIZE | LAST_NUMBER | SCALE_FLAG | EXTEND_FLAG | SHARDED_FLAG | SESSION_FLAG | KEEP_VALUE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ACTORS_SEQ | 1 | 9999999999999999999999999999 | 1 | N | N | 20 | 1021 | N | N | N | N | N |
| 2 | CUST_ID_SEQ | 1 | 9999999999999999999999999999 | 1 | N | N | 20 | 101 | N | N | N | N | N |
| 3 | MEDIA_SEQ | 1 | 9999999999999999999999999999 | 1 | N | N | 20 | 92 | N | N | N | N | N |
| 4 | MOVIES_SEQ | 1 | 9999999999999999999999999999 | 1 | N | N | 20 | 21 | N | N | N | N | N |

6) **Add the data to the tables:** To add data to every table, run the commands in the file “insert_into.sql” in order

STAR_BILLINGS

| | ACTOR_ID | TITLE_ID | COMMENTS |
|---|---|---|---|
| 1 | 1001 | 2 | Romantic Lead |
| 2 | 1002 | 1 | Unexpected end |
| 3 | 1003 | 3 | Nice soundtrack |
| 4 | 1004 | 4 | Good characters |
| 5 | 1005 | 5 | Very exciting story |
| 6 | 1006 | 6 | Very interesting |

MOVIES

| | TITLE_ID | TITLE | DESCRIPTION |
|---|---|---|---|
| 1 | 1 | Remember the Titans | The true story of a newly appointed African-American coach and his high school team on their first season as a |
| 2 | 2 | Bullet Train | Five assassins aboard a swiftly-moving bullet train to find out that their missions have something in common. |
| 3 | 3 | Sing | In a city of humanoid animals, a hustling theater impresarios attempt to save his theater with a singing compe |
| 4 | 4 | How to Train Your Dragon | A hapless young Viking who aspires to hunt dragons becomes the unlikely friend of a young dragon himself, and |
| 5 | 5 | Top Gun: Maverick | After thirty years, Maverick is still pushing the envelope as a top naval aviator, but must confront ghosts of |
| 6 | 6 | The Greatest Showman | Celebrates the birth of show business and tells of a visionary who rose from nothing to create a spectacle tha |

ACTORS

| | ACTOR_ID | STAGE_NAME | FIRST_NAME | LAST_NAME | BIRTH_DATE |
|---|---|---|---|---|---|
| 1 | 1001 | Brad Pitt | William | Pitt | 18-DEC-63 |
| 2 | 1002 | Rihanna | Rihanna | Fenty | 21-FEB-86 |
| 3 | 1003 | Charlie Sheen | Carlos | Estevez | 03-SEP-65 |
| 4 | 1004 | Marilyn Monroe | Marilyn | Miller | 01-JUN-26 |
| 5 | 1005 | Natalie Portman | Neta-Lee | Hershlag | 09-JUN-81 |
| 6 | 1006 | Keanu Reeves | Keanu Charles | Reeves | 02-SEP-64 |

CUSTOMERS

| | CUSTOMER_ID | LAST_NAME | FIRST_NAME | HOME_PHONE | ADDRESS | CITY | STATE | EMAIL | CELL_PHONE |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 101 | Todoroki | Shoto | 267-367-3763 | Av. Paseo San Gerado #134 | Aguascalientes | AG | shotot@gmail.com | 126-273-2749 |
| 2 | 102 | Readus | Norman | 267-467-4659 | Av. Paseo San Lucas #454 | Chihuahua | CH | normanr@gmail.com | 126-477-3644 |
| 3 | 103 | Picapiedra | Pedro | 267-456-3763 | C. Las Lomas #456 | Puebla | PB | pedrop@gmail.com | 126-273-2749 |
| 4 | 104 | Grimes | Rick | 267-957-4658 | c. Walker Street #346 | Sonora | SN | rickg@gmail.com | 126-264-4759 |
| 5 | 105 | Ramirez | Yoel | 267-347-248 | Av. Francisco I Madero #347 | Zacatecas | ZC | yoelr@gmail.com | 126-857-3467 |
| 6 | 106 | Forger | Anya | 267-367-2398 | C. Francisco Villa #467 | Guadalajara | GD | anyaf@gmail.com | 126-947-1123 |

MEDIA

| | MEDIA_ID | FORMAT | TITLE_ID |
|---|---|---|---|
| 1 | 92 | DVD | 1 |
| 2 | 93 | BLR | 1 |
| 3 | 94 | VHS | 2 |
| 4 | 95 | DVD | 3 |
| 5 | 96 | VHS | 3 |
| 6 | 97 | DVD | 4 |
| 7 | 98 | VHS | 5 |
| 8 | 99 | BLR | 5 |
| 9 | 100 | VHS | 6 |
| 10 | 101 | DVD | 6 |

RENTAL_HISTORY

| | MEDIA_ID | RENTAL_DATE | CUSTOMER_ID | RETURN_DATE |
|---|---|---|---|---|
| 1 | 92 | 15-OCT-22 | 101 | 30-OCT-22 |
| 2 | 93 | 03-OCT-22 | 102 | 25-OCT-22 |
| 3 | 94 | 29-SEP-22 | 103 | 22-OCT-22 |
| 4 | 95 | 22-OCT-22 | 104 | 07-NOV-22 |

7) **Create an index on the last_name column of the Customers table:** Run the “CREATE INDEX…” Command in the file: “create_table.sql”

8) **Create a synonym called TU for the TITLE_UNAVAIL view:** Run the command “CREATE SYNONYM…” in the document “create_table.sql”

MariaDB

Instructions to create BD_FLIX database using MariaDB:

1) **Create the User:** To create the user and the database BD_FLIX run the commands on the file: “MySQL_Flix_USER.sql”

2) **Create tables using the attached ERD:** Once connected to BD_FLIX database, run one by one all the CREATE instructions in the file “MySQL_Flix_Create.sql”

3) **Add the following integrity constraints:**

• Create primary key (PK) and foreign key (FK) constraints as needed per ERD

• Create not null (NN) constraints where necessary as per ERD

• Create check constraint on rating field in movie table to limit rating values to 'G', 'PG', 'R', 'PG13'

• Create check constraint on category field in movie table to limit categories to 'DRAMA', 'COMEDY', 'ACTION', 'CHILD', 'SCIFI', 'DOCUMENTARY'

• Run queries from the data dictionaries for the above constraints.

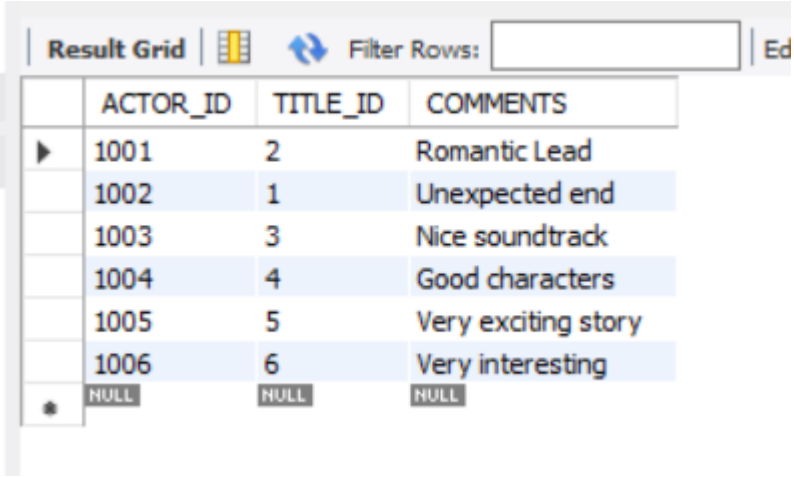
Most of the constraints are created with the tables. To add the Foreign Key constraints, run the “ALTER TABLE” commands in the file “MySQL_Flix_Create.sql”

4) **Create a view called TITLE_UNAVAIL:** To create the view, run the “CREATE OR REPLACE VIEW..” command after all the foregin key constraints in the document “MySQL_Flix_Create.sql”

5) **Create the following sequences to be used for primary key values:** There are no sequences in MySQL, MariaDb, the equivalent is the use of “Auto_increment” while creating the tables.

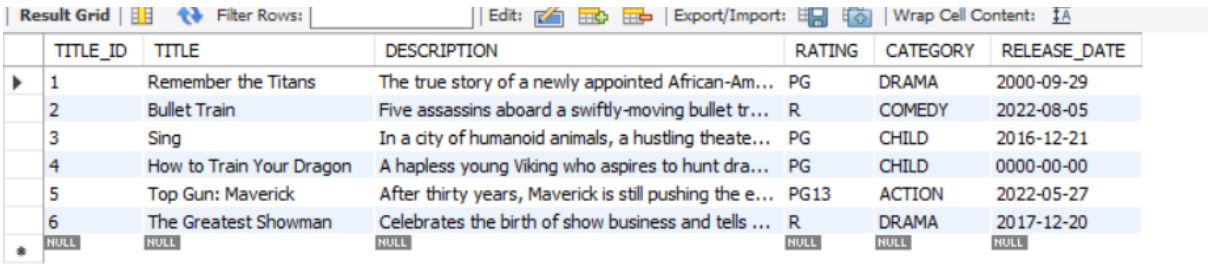
6) **Add the data to the tables:** To add data to every table, run the commands in the file “MySQL_Flix_Data.sql” in order

STAR_BILLINGS



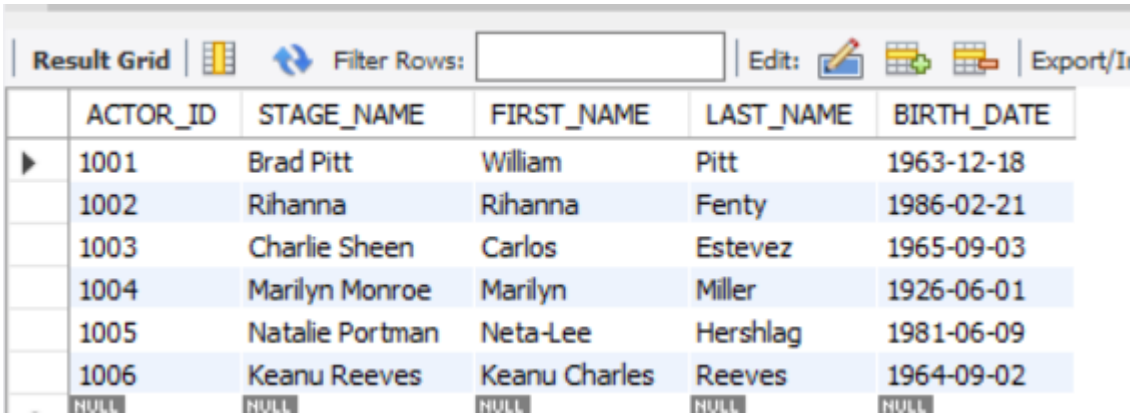

| ACTOR_ID | TITLE_ID | COMMENTS |
|---|---|---|
| 1001 | 2 | Romantic Lead |
| 1002 | 1 | Unexpected end |
| 1003 | 3 | Nice soundtrack |
| 1004 | 4 | Good characters |
| 1005 | 5 | Very exciting story |
| 1006 | 6 | Very interesting |
| NULL | NULL | NULL |

MOVIES



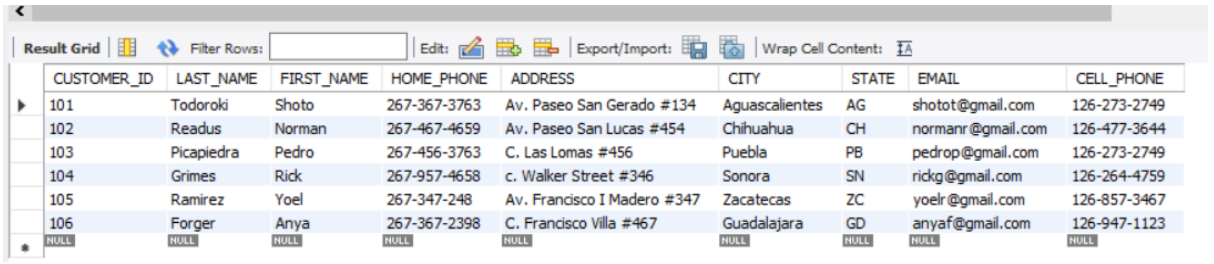
| TITLE_ID | TITLE | DESCRIPTION | RATING | CATEGORY | RELEASE_DATE |
|---|---|---|---|---|---|
| 1 | Remember the Titans | The true story of a newly appointed African-Am... | PG | DRAMA | 2000-09-29 |
| 2 | Bullet Train | Five assassins aboard a swiftly-moving bullet tr... | R | COMEDY | 2022-08-05 |
| 3 | Sing | In a city of humanoid animals, a hustling theate... | PG | CHILD | 2016-12-21 |
| 4 | How to Train Your Dragon | A hapless young Viking who aspires to hunt dra... | PG | CHILD | 0000-00-00 |
| 5 | Top Gun: Maverick | After thirty years, Maverick is still pushing the e... | PG13 | ACTION | 2022-05-27 |
| 6 | The Greatest Showman | Celebrates the birth of show business and tells ... | R | DRAMA | 2017-12-20 |

ACTORS



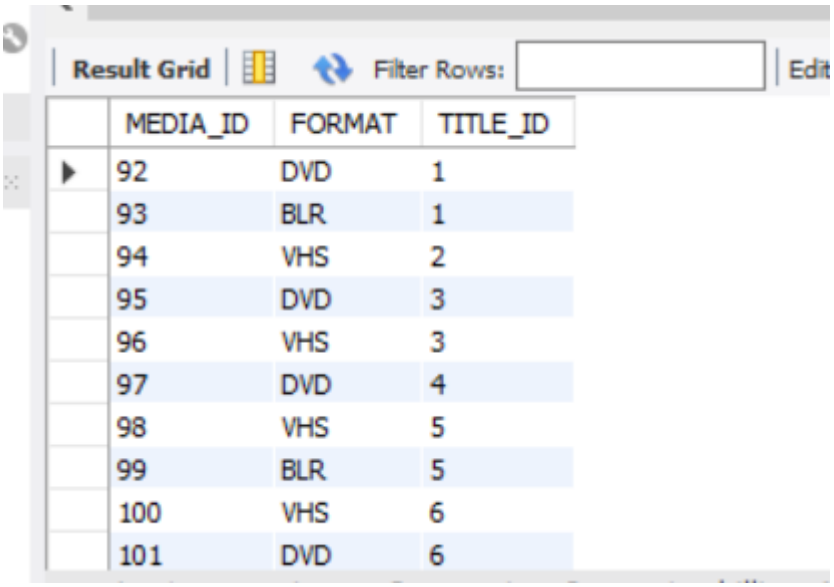
| ACTOR_ID | STAGE_NAME | FIRST_NAME | LAST_NAME | BIRTH_DATE |
|---|---|---|---|---|
| 1001 | Brad Pitt | William | Pitt | 1963-12-18 |
| 1002 | Rihanna | Rihanna | Fenty | 1986-02-21 |
| 1003 | Charlie Sheen | Carlos | Estevez | 1965-09-03 |
| 1004 | Marilyn Monroe | Marilyn | Miller | 1926-06-01 |
| 1005 | Natalie Portman | Neta-Lee | Hershlag | 1981-06-09 |
| 1006 | Keanu Reeves | Keanu Charles | Reeves | 1964-09-02 |

CUSTOMERS



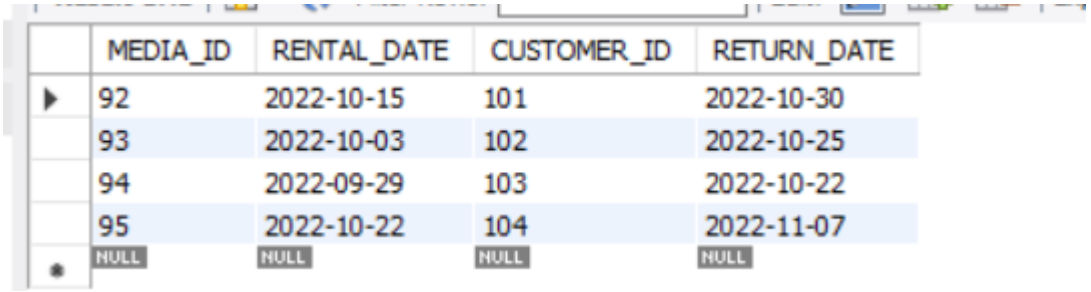
| CUSTOMER_ID | LAST_NAME | FIRST_NAME | HOME_PHONE | ADDRESS | CITY | STATE | EMAIL | CELL_PHONE |
|---|---|---|---|---|---|---|---|---|
| 101 | Todoroki | Shoto | 267-367-3763 | Av. Paseo San Gerado #134 | Aguascalientes | AG | shotot@gmail.com | 126-273-2749 |
| 102 | Readus | Norman | 267-467-4659 | Av. Paseo San Lucas #454 | Chihuahua | CH | normanr@gmail.com | 126-477-3644 |
| 103 | Picapiedra | Pedro | 267-456-3763 | C. Las Lomas #456 | Puebla | PB | pedrop@gmail.com | 126-273-2749 |
| 104 | Grimes | Rick | 267-957-4658 | c. Walker Street #346 | Sonora | SN | rickg@gmail.com | 126-264-4759 |
| 105 | Ramirez | Yoel | 267-347-248 | Av. Francisco I Madero #347 | Zacatecas | ZC | yoelr@gmail.com | 126-857-3467 |
| 106 | Forger | Anya | 267-367-2398 | C. Francisco Villa #467 | Guadalajara | GD | anyaf@gmail.com | 126-947-1123 |

MEDIA




| MEDIA_ID | FORMAT | TITLE_ID |
|---|---|---|
| 92 | DVD | 1 |
| 93 | BLR | 1 |
| 94 | VHS | 2 |
| 95 | DVD | 3 |
| 96 | VHS | 3 |
| 97 | DVD | 4 |
| 98 | VHS | 5 |
| 99 | BLR | 5 |
| 100 | VHS | 6 |
| 101 | DVD | 6 |

RENTAL_HISTORY



| MEDIA_ID | RENTAL_DATE | CUSTOMER_ID | RETURN_DATE |
|---|---|---|---|
| 92 | 2022-10-15 | 101 | 2022-10-30 |
| 93 | 2022-10-03 | 102 | 2022-10-25 |
| 94 | 2022-09-29 | 103 | 2022-10-22 |
| 95 | 2022-10-22 | 104 | 2022-11-07 |
| NULL | NULL | NULL | NULL |

7) **Create an index on the last_name column of the Customers table**: Run the “CREATE INDEX…” Command in the file: “MySQL_Flix_Create.sql”

8) **Create a synonym called TU for the TITLE_UNAVAIL view:** There is no equivalent for synonym in MariaDB according to https://jira.mariadb.org/browse/MDEV-16482

## Conclusion

Creating a database is more than just creating tables, you also have to consider user permissions, the constraints in each table, foreing keys to create the relations between tables, sequences, views, synonyms, etc.

Also you need to know the syntax of each software, you can use Oracle or MySQL to create your database, but the structure is the same.

It's important to implement techniques like sequences in your database because you can use it if your database grows. For example the sequence can help you to auto increment your primary key in oracle.