



AGUASCALIENTES
GOBIERNO DEL ESTADO
Contigo al 100



Universidad Politécnica de Aguascalientes

Ingeniería en Sistemas Computacionales

Administración de Base de Datos

ISC06A

Mariana Andrade Garcia UP200651

Diego Sanchez Olvera UP210010

Cristopher Kaleb Ruiz Ortiz UP200674

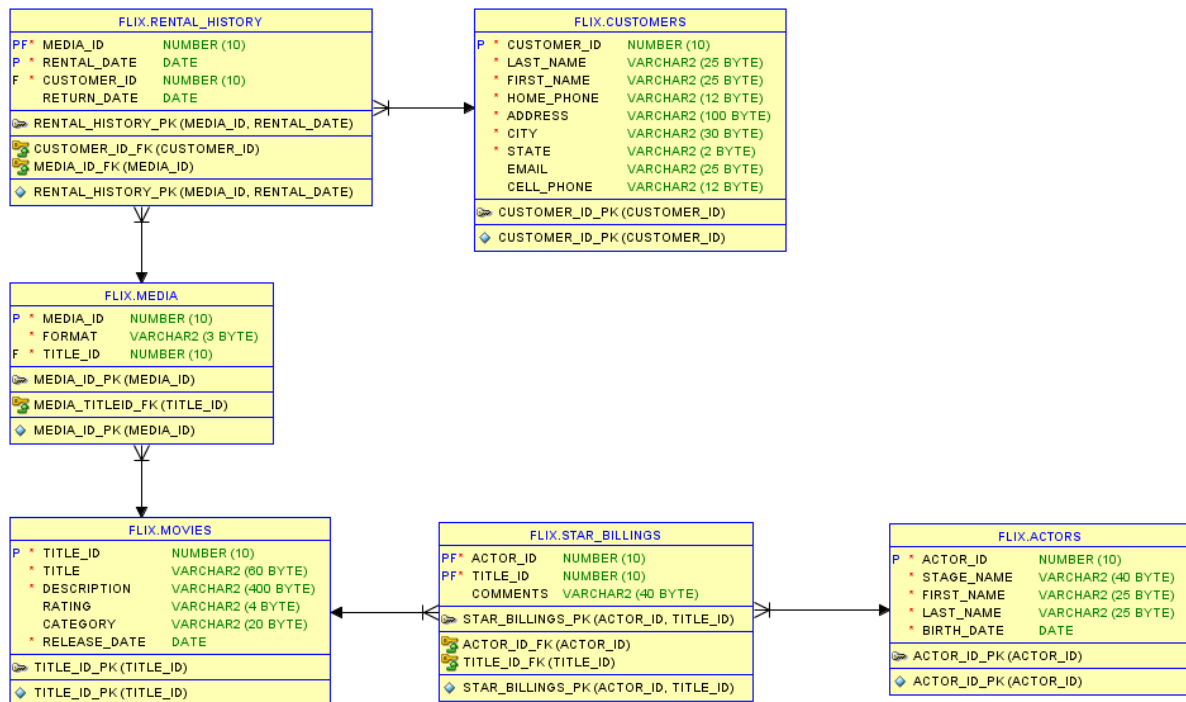
Period 2022-2

November 06 2022

Diagram:	2
Create User:	2
Inserte data:	2
Index, synonym and view:	3
Queries:	3
Create User:	5
Create tables:	5
Insert Data:	6
Index, synonym and view:	6
Conclusion:	6

Oracle

Diagram:



Create User:

For this step, which is the main one, we must make sure that the user that is going to be created does not exist, in this case we will use on behalf of Flix, once we are sure that it does not exist, we proceed to create it in order to give it the permissions it requires to be able to manage the database without affecting its operation. This will help us in the following steps such as creating the tables, queries and inserting the necessary data to fill the database.

Inserte data:

The INSERT statement adds one or more new rows of data to a database table.

We can use this method to easily fill our data tables with a simple command, it is easier to add data to tables like actors and movies because in these tables there are no limitations or extra information needed so it is only inserting the data required and done.

In the other hand we have the different tables such as rental_history that is a table that depends on the information that other tables have in this case customers for example that is a table that can easily add data by using the insert data command

Sequences

A sequence is used to generate unique sequential integer values and assign them to numeric fields.

These are the sequences that were used:

- Use a sequence to generate PKs for CUSTOMER_ID in CUSTOMERS table
 - Begin at 101 and increment by 1
- Use a sequence to generate PKs for TITLE_ID in MOVIES table
 - Begin at 1 and increment by 1
- Use a sequence to generate PKs for MEDIA_ID in MEDIA table
 - Begin at 92 and increment by 1
- Use a sequence to generate PKs for ACTOR_ID in ACTOR table
 - Begin at 1001 and increment by 1

Index, synonym and view:

Indexes are structures associated with tables, a table that stores the indexed fields and are created to speed up queries.

In this case an index was created in customers.

A synonym is a local or public representation of an object belonging to a schema.

In this case it is for "TITLE_UNAVAIL.

A view is an alternative to displaying data from multiple tables; it is like a virtual table that stores a query. A query that returns the join of 3 of the tables.

Queries:

- 1) Shows the name of the drama category movies and their actors

```
SELECT a.actor_id, a.stage_name, m.title, m.category FROM movies m
INNER JOIN star_billings s ON m.title_id = s.title_id
INNER JOIN actors a ON s.actor_id = a.actor_id
WHERE category = 'ACTION';
```

	⚡ ACTOR_ID	⚡ STAGE_NAME	⚡ TITLE	⚡ CATEGORY
1	1002	Will Smith	Titanic	DRAMA
2	1001	Brad Pitt	Remember the Titans	DRAMA
3	1004	Scarlet Johanson	The Sound of Music	DRAMA

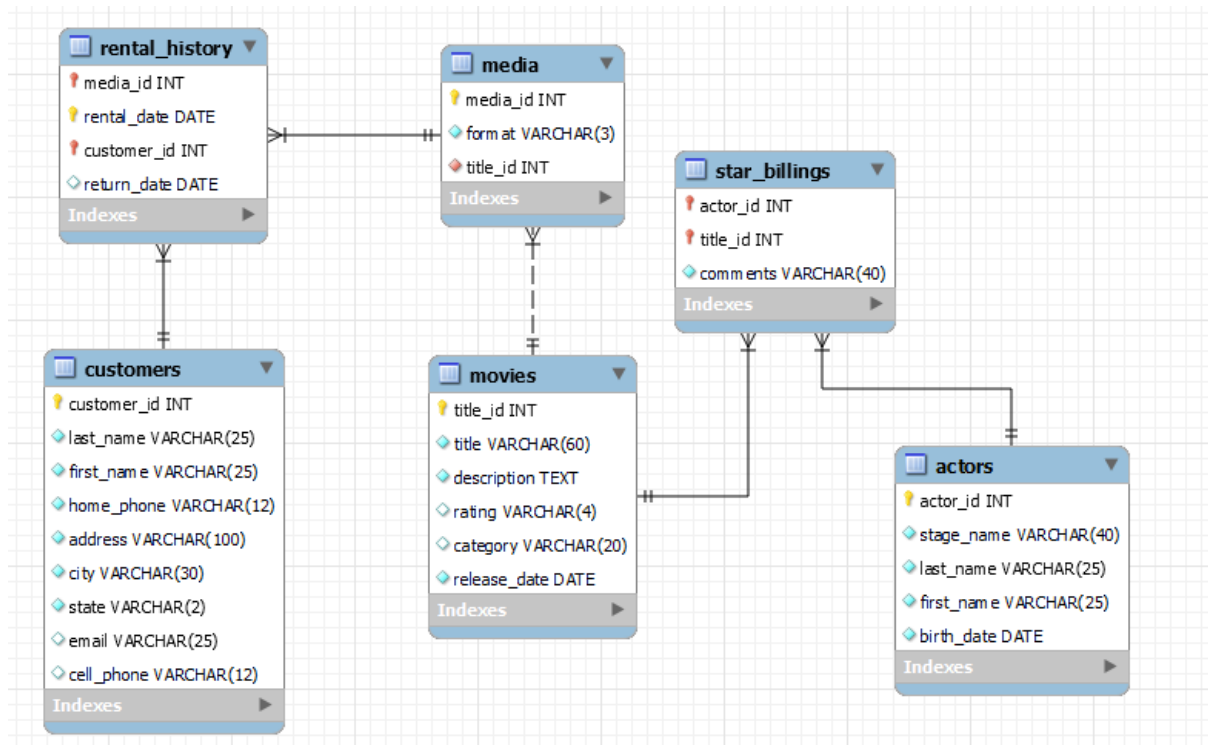
- 2) Shows the last_name of the clients who have made more than three rentals

```
SELECT last_name,customer_id  
FROM customers  
WHERE customer_id IN (SELECT (customer_id)  
                     FROM rental_history  
                     GROUP BY customer_id  
                     HAVING COUNT (customer_id) >= 3);
```

3) Shows the movies that are for rent of the 'PG' rating

```
SELECT c.last_name, a.stage_name, m.title, m.rating, rh.return_date  
FROM actors a, movies m, rental_history rh, star_billings sb, media me,  
customers c  
WHERE a.actor_id = sb.actor_id  
      and m.title_id = sb.title_id  
      and me.media_id = rh.media_id  
      and m.title_id = me.title_id  
      and c.customer_id = rh.customer_id  
      and m.rating = 'PG';
```

MySQL Diagram:



Create User:

To create a user, we have to create the database as long as it has not been created before, in this case we have to create the user along with its password to avoid the problem that we use Flix as username and password, then we have to grant the user permission to execute the required operation.

Create tables:

Para poder crear nuestras tablas deberemos tomar en cuenta las llaves primarias y secundarias, al igual que debemos observar cuáles tablas no son dependientes para que se puedan crear primero esas tablas y así no nos afecte a medida que vayamos avanzando y creando nuevas tablas, en este caso como llaves primarias tenemos:

- Movie
- Actor
- Customers

Las cuales nos ayudarán a unir las tablas dependientes apoyándonos en llaves secundarias como lo puede ser

- Star_billings
- Rental_history
- Media

Insert Data:

When inserting data into a MySQL database, using that database in the clients, media, actors, and movies tables, they handle the auto-increment field in Oracle, call the sequence, and tell it to increment the next sequence of amount, in MySQL because there is no sequence, so the tables need to be changed to start with the desired number, and when inserting data, you don't need to put an autoincrement column, because when you declare an autoincrement in MySQL, when you insert a value , the autoincrement takes the next value.

Sequences:

is a sequence of integers: as a data table can have increment primary keys only one field, if you want to implement additional fields they are automatically incremented. In the case of MySQL the simplest method is to use a sequence to define MySQL's AUTO_INCREMENT column usage.

Index, synonym and view:

Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

The synonym: In the case of MySQL it is practically the same information that we must place with small syntax variations due to the type of language that is used, however we must consider that MySQL does not support synonyms, for which this step is omitted as shown in the following information obtained from a reliable source page such as IBM and is immediately left

<https://www.ibm.com/docs/es/rtw/9.1.1?topic=overview-synonyms>

In MySQL, a view is a virtual table based on the result set of an SQL statement, just like in Oracle. A view contains rows and columns, just like a real table. The fields of a view are fields from one or more actual tables in the database.

Conclusion:

For this project we had to have prior knowledge of a database such as tables, basic commands, etc. This in order to be able to perform all the necessary operations to connect tables, generate queries and be able to identify both primary and secondary keys.

Having developed this database on two different platforms shows that there is no problem in using different platforms as long as you have the basic structure of what you are going to do. Likewise, knowing how to solve failures within the database for its optimal performance and functionality, which will greatly facilitate some work in the future.