Data Base Management Systems Term Project

Project title and team info including name, surname, student ID, class information of the team members:

Comment Database

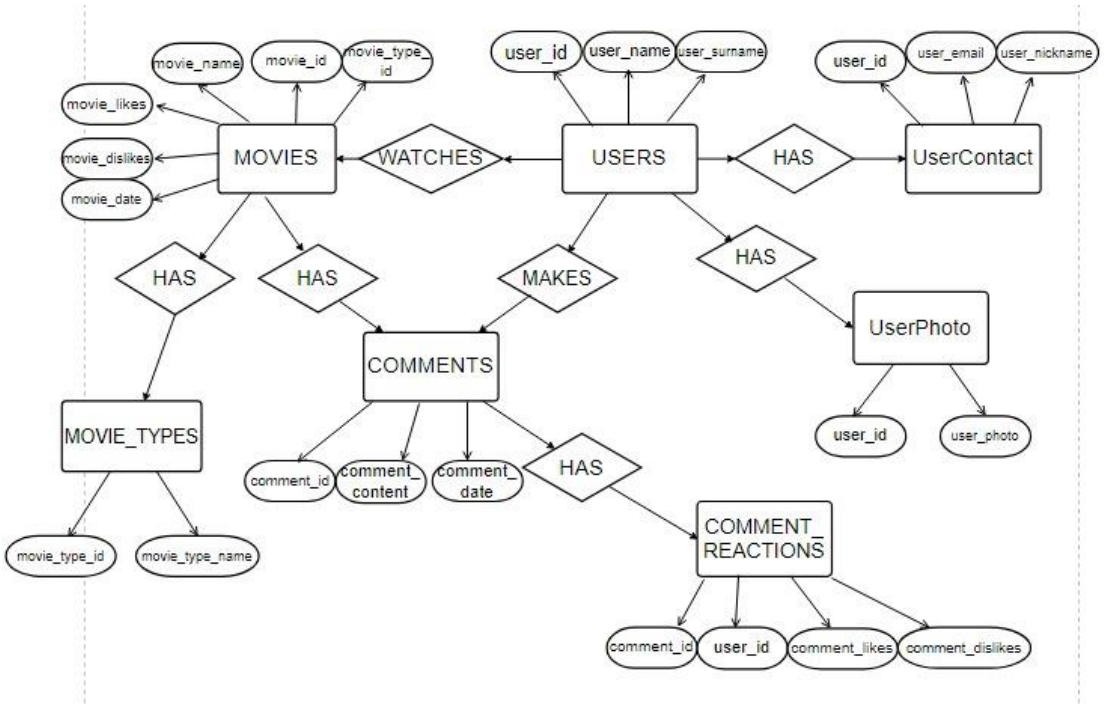
Özde Selen Akın 201504045 2th grade

Eylülnaz Patlı 2111504210 2th grade

Project Objective and Scope:

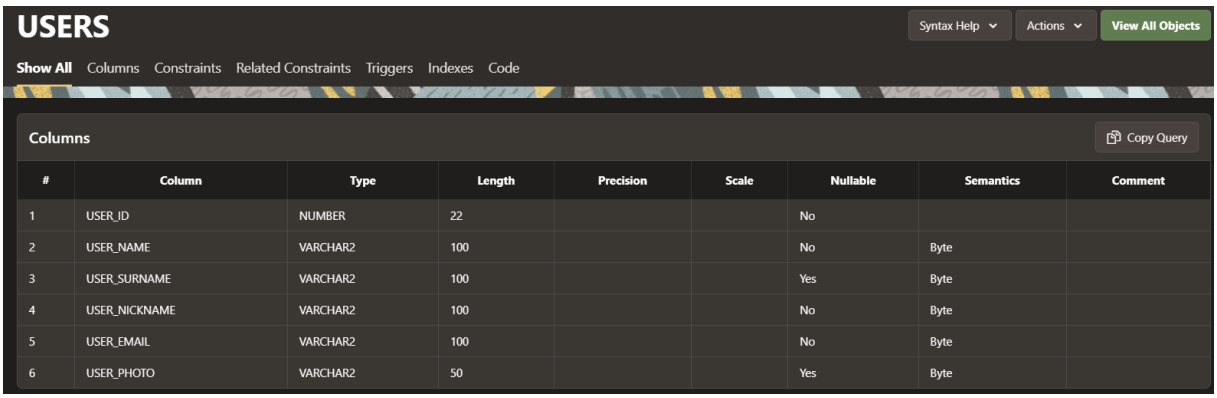
Codes, er diagram and queries necessary to keep the data of the people who comment on the movies on a site, to keep the data of the movies and to keep the data of the comments.

ER Diagram:



Normalization:

1- First we created the users table and we entered the user data. Then we normalized the users table to USERS, UserContact, UserPhoto and we adapted the data according to these three tables.



USERS

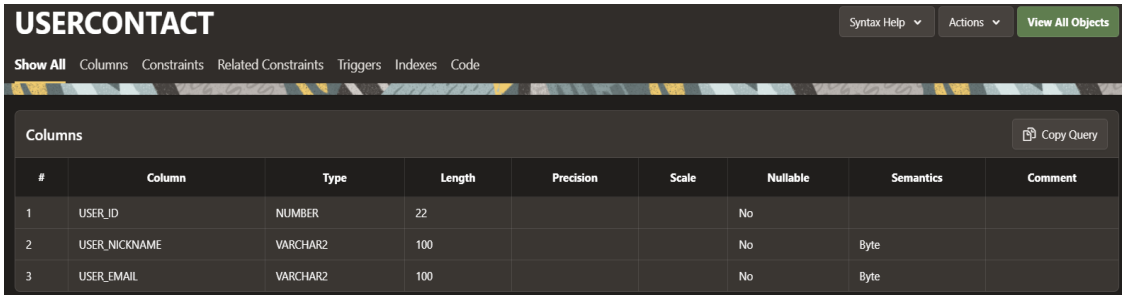
Syntax Help | Actions | View All Objects

Show All | Columns | Constraints | Related Constraints | Triggers | Indexes | Code

Columns — Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	USER_ID	NUMBER	22			No		
2	USER_NAME	VARCHAR2	100			No	Byte	
3	USER_SURNAME	VARCHAR2	100			Yes	Byte	
4	USER_NICKNAME	VARCHAR2	100			No	Byte	
5	USER_EMAIL	VARCHAR2	100			No	Byte	
6	USER_PHOTO	VARCHAR2	50			Yes	Byte	

A-



USERCONTACT

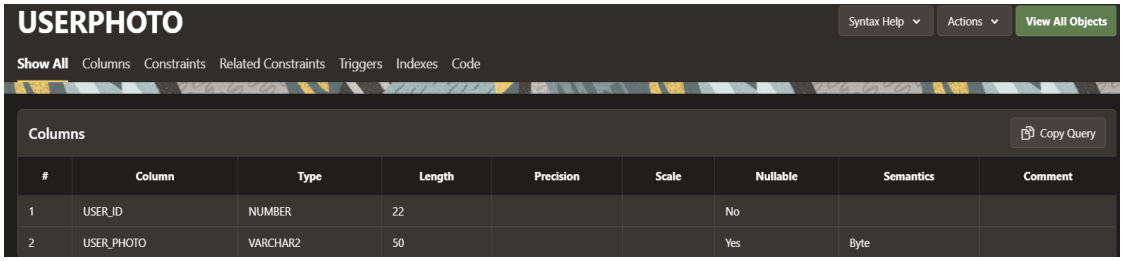
Syntax Help | Actions | View All Objects

Show All | Columns | Constraints | Related Constraints | Triggers | Indexes | Code

Columns — Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	USER_ID	NUMBER	22			No		
2	USER_NICKNAME	VARCHAR2	100			No	Byte	
3	USER_EMAIL	VARCHAR2	100			No	Byte	

B-



USERPHOTO

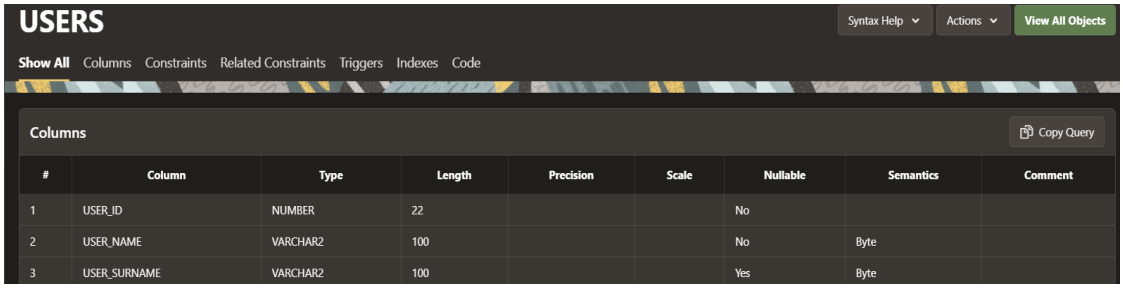
Syntax Help | Actions | View All Objects

Show All | Columns | Constraints | Related Constraints | Triggers | Indexes | Code

Columns — Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	USER_ID	NUMBER	22			No		
2	USER_PHOTO	VARCHAR2	50			Yes	Byte	

C-



USERS

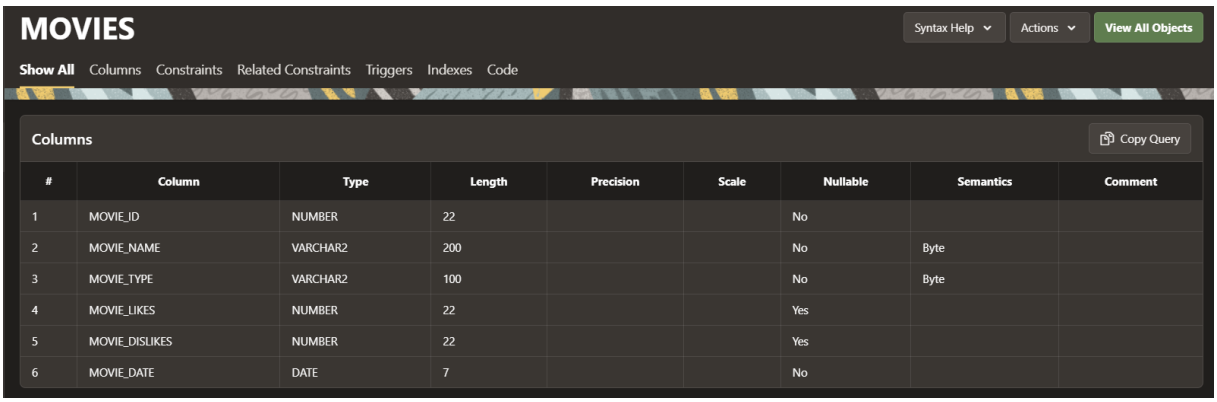
Syntax Help | Actions | View All Objects

Show All | Columns | Constraints | Related Constraints | Triggers | Indexes | Code

Columns — Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	USER_ID	NUMBER	22			No		
2	USER_NAME	VARCHAR2	100			No	Byte	
3	USER_SURNAME	VARCHAR2	100			Yes	Byte	

2- Secondly, we created the movie table and entered the data of the movies. Then we normalized the movies table to MOVIE_TYPES, MOVIES and we adapted the data according to these two tables.



MOVIES

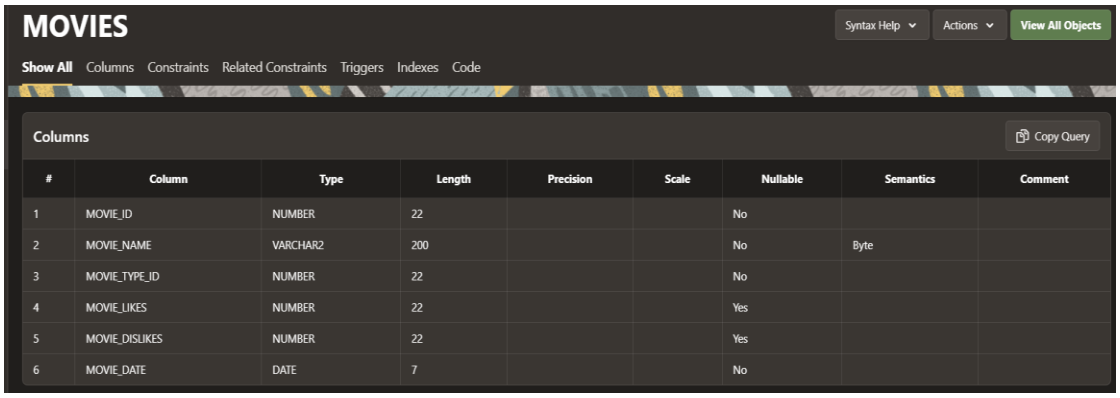
Syntax Help | Actions | View All Objects

Show All | Columns | Constraints | Related Constraints | Triggers | Indexes | Code

Columns — Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	MOVIE_ID	NUMBER	22			No		
2	MOVIE_NAME	VARCHAR2	200			No	Byte	
3	MOVIE_TYPE	VARCHAR2	100			No	Byte	
4	MOVIE_LIKES	NUMBER	22			Yes		
5	MOVIE_DISLIKES	NUMBER	22			Yes		
6	MOVIE_DATE	DATE	7			No		

A-



MOVIES

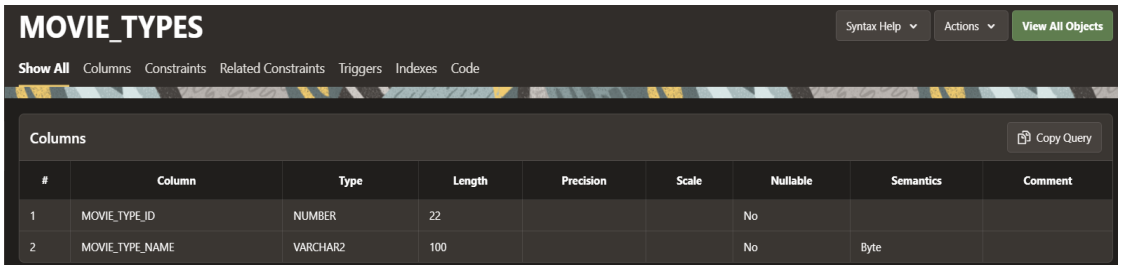
Syntax Help | Actions | View All Objects

Show All | Columns | Constraints | Related Constraints | Triggers | Indexes | Code

Columns — Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	MOVIE_ID	NUMBER	22			No		
2	MOVIE_NAME	VARCHAR2	200			No	Byte	
3	MOVIE_TYPE_ID	NUMBER	22			No		
4	MOVIE_LIKES	NUMBER	22			Yes		
5	MOVIE_DISLIKES	NUMBER	22			Yes		
6	MOVIE_DATE	DATE	7			No		

B-



MOVIE_TYPES

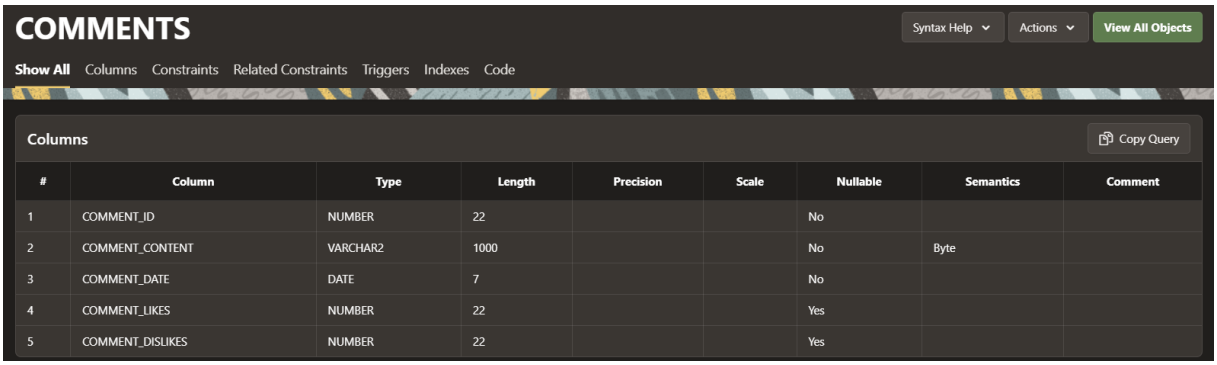
Syntax Help | Actions | View All Objects

Show All | Columns | Constraints | Related Constraints | Triggers | Indexes | Code

Columns — Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	MOVIE_TYPE_ID	NUMBER	22			No		
2	MOVIE_TYPE_NAME	VARCHAR2	100			No	Byte	

3- Third, we created the comments table and then entered the data for the comments. Then we normalized the users table to COMMENTS, COMMENT_REACTIONS and we adapted the data according to these two tables.



COMMENTS

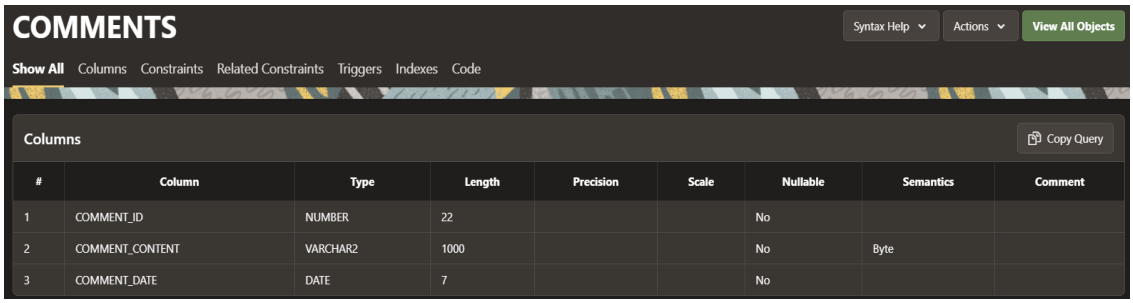
Syntax Help | Actions | View All Objects

Show All | Columns | Constraints | Related Constraints | Triggers | Indexes | Code

Columns — Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	COMMENT_ID	NUMBER	22			No		
2	COMMENT_CONTENT	VARCHAR2	1000			No	Byte	
3	COMMENT_DATE	DATE	7			No		
4	COMMENT_LIKES	NUMBER	22			Yes		
5	COMMENT_DISLIKES	NUMBER	22			Yes		

A-



COMMENTS

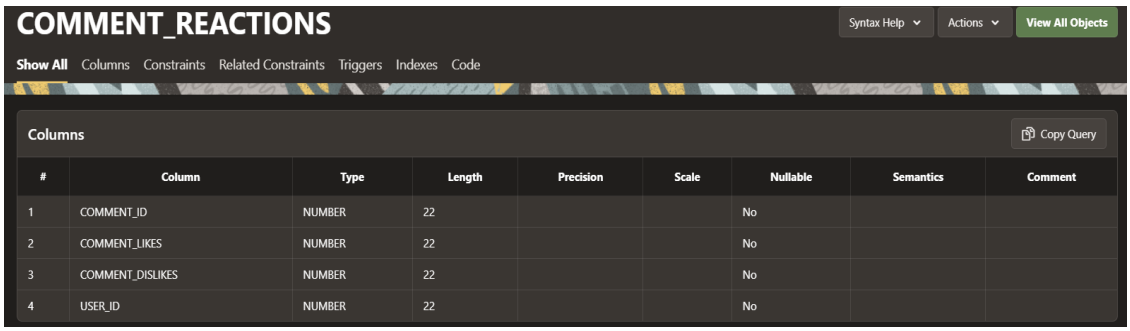
Syntax Help | Actions | View All Objects

Show All | Columns | Constraints | Related Constraints | Triggers | Indexes | Code

Columns — Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	COMMENT_ID	NUMBER	22			No		
2	COMMENT_CONTENT	VARCHAR2	1000			No	Byte	
3	COMMENT_DATE	DATE	7			No		

B-



COMMENT_REACTIONS

Syntax Help | Actions | View All Objects

Show All | Columns | Constraints | Related Constraints | Triggers | Indexes | Code

Columns — Copy Query

#	Column	Type	Length	Precision	Scale	Nullable	Semantics	Comment
1	COMMENT_ID	NUMBER	22			No		
2	COMMENT_LIKES	NUMBER	22			No		
3	COMMENT_DISLIKES	NUMBER	22			No		
4	USER_ID	NUMBER	22			No		

Queries:

1- Names and genres of all movies:

```
SELECT m.movie_name, mt.movie_type_name
FROM MOVIES m
JOIN MOVIE_TYPES mt ON m.movie_type_id = mt.movie_type_id;
```

2- A user's first name, last name, and contact information:

```
SELECT u.user_name, u.user_surname, uc.user_nickname, uc.user_email
FROM USERS u
JOIN UserContact uc ON u.user_id = uc.user_id
WHERE u.user_id = 1;
```

3- Number of likes and dislikes for a movie:

```
SELECT movie_likes, movie_dislikes
FROM MOVIES
WHERE movie_id = 1;
```

4- Profile photo of a user:

```
SELECT user_photo
FROM UserPhoto
WHERE user_id = 1;
```

5- Comments made by a user and their comment dates:

```
SELECT c.comment_content, c.comment_date
FROM COMMENTS c
JOIN USERS u ON c.user_id = u.user_id
WHERE u.user_id = 1;
```