

# Big Data Analytics 2024 Project 2

## Welcome to MiniNet

**MiniNet** offers an online TV show streaming service similar to Netflix. It allows users to subscribe to various subscription plans, watch TV shows and videos, and rate/review their watched content. The database for MiniNet needs to manage user data, subscription plans, TV shows, videos, user reviews, watch favourites, and the details of the actors involved in the shows.

Your task is to develop the entity-relationship (ER) model and analyse the relationships using your reasoning skills. Then, you must build the SQL database based on your ER model and develop queries to demonstrate your understanding.

Good luck 🚀

## Part 1: Project 2 Description

- Consider the following use case text.

You must design an entity-relationship model for the given use case. You will need to identify the appropriate tables and their relationships. You will also need to identify attributes per table. **Make sure you explore Task 4 queries before designing your attributes.** Feel free to improvise when identifying the fields; the more, the better.


At MiniNet, users subscribe to watch a wide variety of movies and TV shows. Each user can have only one subscription, choosing between two options: HD or UHD. The content is organized into categories like comedy and drama, as well as by type, such as TV shows and movies.

Users can watch an unlimited number of movies and can maintain a single list of favorite movies. Each movie in the favorite list includes a user-assigned score. Additionally, MiniNet keeps detailed records of the actors in each movie, including their names, cities of residence, and dates of birth.

Base your cardinality design on logical arguments while at the same time considering the following requirements:

- Each user can have only one subscription.
- There are two subscription types: HD (£12) and UHD (£18).
- For each user, in addition to your desired attributes, you need to store the `user city`, `country`, and `age`.
- Each user can only have one favourite list.
- Actors should have an age.
- Actors are associated with more than one movie.
- Each movie belongs to a category, e.g. comedy, drama, etc.
- There should be at least one movie starting with the keyword `The`, e.g., `The Lord of the Rings`.
- Each user can give only one review per video.

- Each review can be any number between 0 to 5.


 You can make assumptions for the rest of the relationships.

## Part 2: Project 2 Tasks

### Task 1 [30 marks]

Design an ER model to include the following:

- a) Entities according to the use case.
- b) Relationships between entities.
- c) Cardinality of relationships.
- d) Attributes (2-5 per entity) and their data types.
- e) An ER diagram in your preferred format. The diagram can be a screenshot of your in-paper design or can be designed in your preferred software, e.g., PowerPoint.

 Feel free to improvise when deciding the data types of the attributes. You can base your decision on logical arguments, such as a first name should be a character variable of up to 30 characters, etc.

### Task 2 [10 marks]

You must provide the scripts to create the database model in SQL. Consider the following:

- Make sure that you identify the appropriate relationships using primary and foreign keys.
- Make sure you identify the appropriate constraints when linking the tables.
- You need to enforce referential integrity when creating the tables, for example, you should not be able to delete a user if there is an active subscription.


Provide the SQL scripts for creating the database and **at least five tables, including movies, users, reviews, actors, and subscriptions**. Provide your scripts in a report.

### Task 3 [5 marks]

Provide the SQL scripts for inserting data into the database tables. You should insert at least five records per table. Explore **Part 3** on example data.

### Task 4 [25 total marks]

Create queries for extracting data. **Feel free to adjust the queries if your database does not include these values.**

 The queries could be customised according to your requirements, e.g., show an actor's first name, last name, date of birth, etc.

The following queries demonstrate example outputs for your reference. **Note that your outputs could be different, based on your database structure, and in case you insert different data.**

4.1 Export all data about users in the  subscriptions.

UserID	Username	Email	Password	SubscriptionType	City	Country	Age
1	john_doe	<a href="mailto:john@example.com">john@example.com</a>	password1	HD	New York	USA	28
2	alice_smith	<a href="mailto:alice@example.com">alice@example.com</a>	password2	HD	Los Angeles	USA	34

4.2 Export all data about actors and their associated movies.

ActorID	Name	City	DateOfBirth	MovieID	Title	Genre	ReleaseDate
1	Millie Bobby Brown	Los Angeles	2004-02-19	1	Stranger Things	Sci-Fi	2016-07-15
2	Bryan Cranston	Hollywood	1956-03-07	2	Breaking Bad	Drama	2008-01-20

4.3 Export all data to group actors from a specific city, showing also the average age (per city).

City	NumberOfActors	AverageAge
Los Angeles	2	34.5
Hollywood	1	68

4.4 Export all data to show the favourite `comedy` movies for a specific user.

Username	MovieID	Title	Genre	ReleaseDate
john_doe	3	The Office	Comedy	2005-03-24
john_doe	4	Parks and Recreation	Comedy	2009-04-09

4.5 Export all data to count how many subscriptions are in the database per country.

Country	SubscriptionCount
USA	150
Canada	50

4.6 Export all data to find the movies that start with the keyword `The`.

MovieID	Title	Genre	ReleaseDate
5	The Godfather	Crime	1972-03-24
6	The Lord of the Rings	Action	2008-07-18

4.7 Export data to find the number of subscriptions per movie category.

Category	SubscriptionCount
Sci-Fi	80
Drama	120
Comedy	70

4.8 Export data to find the username and the city of the youngest customer in the `UHD` subscription category.

UserID	Username	Email	Password	SubscriptionType	City	Country	Age
3	jane_doe	<a href="mailto:jane@example.com">jane@example.com</a>	password3	UHD	Chicago	USA	21

4.9 Export data to find users between `22 - 30` years old (including `22` and `30`).

UserID	Username	Email	Password	SubscriptionType	City	Country	Age
1	john_doe	<a href="mailto:john@example.com">john@example.com</a>	password1	HD	New York	USA	28
4	bob_jones	<a href="mailto:bob@example.com">bob@example.com</a>	password4	UHD	Boston	USA	30

4.10 Export data to find the average age of users with low score reviews (less than 3). Group your data for users under 20, 21-40, and 41 and over.

AgeGroup	AverageAge
Under 20	18.5
21-40	30.2
41 and over	45.7

### Part 3: Using Python

You will need to use Python to extract data and demonstrate interactions with the database.

#### Task 5 [10 marks]

Provide Python scripts to run the Task 4 queries 1-5. These queries need to request user input and match the appropriate data from the database.

#### Task 6 [10 marks]

Create the `users` table in Apache Cassandra and generate the following queries in CQL and Python.

6.1 Provide the `CREATE` statement in CQL.

6.2 Export all users.

6.3 Export all users from a specific country.

6.4 Export data to find users between 22-30 years old (including 22 and 30).

6.5 Count how many users exist per specific city.

Task 7 [10 marks]

Provide a report to organise your tasks, including the ER model, SQL, CQL and Python scripts, with appropriate descriptions. Feel free to organise the report as you prefer.

🚩 For this assignment, there is no word limit.

Part 3: Supporting material

Feel free to use your design. These tables do not include the primary and foreign keys. You can adjust the following tables as needed.

Users Table

Username	Email	Password	SubscriptionType	City	Country	Age
john_doe	<a href="mailto:john@example.com">john@example.com</a>	password1	HD	New York	USA	28
alice_smith	<a href="mailto:alice@example.com">alice@example.com</a>	password2	HD	Los Angeles	USA	34
jane_doe	<a href="mailto:jane@example.com">jane@example.com</a>	password3	UHD	Chicago	USA	21
bob_jones	<a href="mailto:bob@example.com">bob@example.com</a>	password4	UHD	Boston	USA	30
emma_johnson	<a href="mailto:emma@example.com">emma@example.com</a>	password5	HD	San Francisco	USA	25

Actors Table

Name	City	DateOfBirth
Millie Bobby Brown	Los Angeles	2004-02-19
Bryan Cranston	Hollywood	1956-03-07
Winona Ryder	New York	1971-10-29
Aaron Paul	Boise	1979-08-27
David Harbour	White Plains	1975-04-10

Movies Table

Title	Genre	ReleaseDate
Stranger Things	Sci-Fi	2016-07-15
Breaking Bad	Drama	2008-01-20
The Office	Comedy	2005-03-24
Parks and Recreation	Comedy	2009-04-09
The Godfather	Crime	1972-03-24

Subscriptions Table

PlanName	Price	Duration (Months)
HD	9.99	1
UHD	14.99	1

FavoriteMovies Table

Username	MovieTitle	Score
john_doe	The Office	5
john_doe	Parks and Recreation	4
alice_smith	The Godfather	3
jane_doe	Stranger Things	5
bob_jones	Breaking Bad	4

Reviews Table

Username	MovieTitle	Score	Comment
john_doe	Stranger Things	5	Amazing show!
alice_smith	Breaking Bad	3	Good show
jane_doe	The Office	4	Funny and smart
bob_jones	Parks and Recreation	2	Not my taste
emma_johnson	The Godfather	5	A classic!

WatchHistory Table

Username	MovieTitle	WatchDate
john_doe	Stranger Things	2023-06-10
alice_smith	Breaking Bad	2023-06-11
jane_doe	The Office	2023-06-12
bob_jones	Parks and Recreation	2023-06-13
emma_johnson	The Godfather	2023-06-14

## MovieActors Table

MovieTitle	ActorName	Role
Stranger Things	Millie Bobby Brown	Eleven
Breaking Bad	Bryan Cranston	Walter White
Stranger Things	Winona Ryder	Joyce Byers
Breaking Bad	Aaron Paul	Jesse Pinkman
Stranger Things	David Harbour	Jim Hopper

## SQL Insert statements.

Similarly to previous examples, the scripts below do not include primary or foreign keys.

## Users Table

```
INSERT INTO Users (Username, Email, Password, SubscriptionType, City, Country, Age)
VALUES
('john_doe', 'john@example.com', 'password1', 'HD', 'New York', 'USA', 28),
('alice_smith', 'alice@example.com', 'password2', 'HD', 'Los Angeles', 'USA', 34),
('jane_doe', 'jane@example.com', 'password3', 'UHD', 'Chicago', 'USA', 21),
('bob_jones', 'bob@example.com', 'password4', 'UHD', 'Boston', 'USA', 30),
('emma_johnson', 'emma@example.com', 'password5', 'HD', 'San Francisco', 'USA', 25);
```

## Actors Table

```
INSERT INTO Actors (Name, City, DateOfBirth) VALUES
('Millie Bobby Brown', 'Los Angeles', '2004-02-19'),
('Bryan Cranston', 'Hollywood', '1956-03-07'),
('Winona Ryder', 'New York', '1971-10-29'),
('Aaron Paul', 'Boise', '1979-08-27'),
('David Harbour', 'White Plains', '1975-04-10');
```

## Movies Table

```
INSERT INTO Movies (Title, Genre, ReleaseDate) VALUES
('Stranger Things', 'Sci-Fi', '2016-07-15'),
('Breaking Bad', 'Drama', '2008-01-20'),
('The Office', 'Comedy', '2005-03-24'),
('Parks and Recreation', 'Comedy', '2009-04-09'),
('The Godfather', 'Crime', '1972-03-24');
```

## Subscriptions Table

```
INSERT INTO Subscriptions (PlanName, Price, Duration) VALUES
('HD', 9.99, 1),
('UHD', 14.99, 1);
```

## FavoriteMovies Table

```
INSERT INTO FavoriteMovies (Username, MovieTitle, Score) VALUES
('john_doe', 'The Office', 5),
('john_doe', 'Parks and Recreation', 4),
('alice_smith', 'The Godfather', 3),
('jane_doe', 'Stranger Things', 5),
('bob_jones', 'Breaking Bad', 4);
```

## Reviews Table

```
INSERT INTO Reviews (Username, MovieTitle, Score, Comment) VALUES
('john_doe', 'Stranger Things', 5, 'Amazing show!'),
('alice_smith', 'Breaking Bad', 3, 'Good show'),
('jane_doe', 'The Office', 4, 'Funny and smart'),
('bob_jones', 'Parks and Recreation', 2, 'Not my taste'),
('emma_johnson', 'The Godfather', 5, 'A classic!');
```

## MovieActors Table

```
INSERT INTO MovieActors (MovieTitle, ActorName, Role) VALUES
('Stranger Things', 'Millie Bobby Brown', 'Eleven'),
('Breaking Bad', 'Bryan Cranston', 'Walter White'),
('Stranger Things', 'Winona Ryder', 'Joyce Byers'),
('Breaking Bad', 'Aaron Paul', 'Jesse Pinkman'),
('Stranger Things', 'David Harbour', 'Jim Hopper');
```