

1. progettare il modello ER per questa situazione nella seconda facciata del foglio protocollo (prima metà)
2. scrivere il modello logico corrispondente nella seconda facciata (seconda metà)

Entità:

Utente (ID, nome, cognome, data di nascita, nazione di residenza, sesso, inizio abbonamento, fine abbonamento)
 Episodio (ID, ID Serie TV, titolo, durata, stagione)
 Serie TV (ID, nome, trama, numero stagioni)
 Attore (ID, nome, cognome)
 Genere (ID, nome)

Relazioni:

Valutazione (Utente-Serie TV, ID Utente, ID Serie TV, valutazione Serie TV)
 Appartiene (Serie TV-Episodio)
 Partecipa (Episodio-Attore, ID Episodio, ID Attore, nome personaggio, cognome personaggio)
 Visualizzazione (Utente-Episodio, ID Utente, ID Episodio, data visualizzazione, valutazione episodio)
 Relazione (Serie TV-Genere, ID Serie TV, ID Genere)

3. scrivere nella terza facciata le istruzioni SQL per:
 - a. avere il gradimento medio dell'episodio 'La scatola del bravo ragazzo' della serie TV 'Dahmer'

```
SELECT avg(visualizzazione.valutazioneepisodio) as
gradimentomedio
from Visualizzazione inner join Episodio
on Visualizzazione.IDEpisodio = Episodio.ID
inner join SerieTV
on Episodio.IDSerieTV = SerieTV.ID
where SerieTV.nome = 'Dahmer' AND Episodio.titolo = 'La
scatola del bravo ragazzo'
```

```
SELECT avg(visualizzazione.valutazioneepisodio)
where IDEpisodio = (SELECT ID from Episodio where nome =
'La scatola del bravo ragazzo' AND IDSerieTV = (SELECT ID
from SerieTV where nome = 'Dahmer'))
```

- b. avere l'elenco delle serie TV con il numero di episodi di ogni serie

```
SELECT SerieTV.ID, SerieTV.nome, count (*) as
numeroepisodi
from SerieTV inner join Episodio
on SerieTV.ID = episodio.IDSerieTV
GROUP BY SerieTV.ID, SerieTV.nome
```

- c. avere gli utenti (nome e cognome) che hanno guardato più di 100 episodi

```
SELECT Utente.ID, Utente.nome, Utente.cognome, count(*)  
as episodiguardati  
from Utente inner join Visualizzazione  
on Utente.ID = Visualizzazione.IDUtente  
GROUP BY Utente.ID, Utente.nome, Utente.cognome  
HAVING count(*) > 100  
ORDER BY episodiguardati desc
```

- d. avere gli utenti che hanno guardato episodi di più di 10 serie TV diverse

```
SELECT Utente.ID, Utente.nome, Utente.cognome,  
count(SerieTV.ID) as numeroserie  
from Utente inner join Visualizzazione  
on Utente.ID = Visualizzazione.IDUtente  
inner join Episodio  
on Visualizzazione.IDEpisodio = Episodio.ID  
inner join SerieTV  
on Episodio.IDSerieTV = SerieTV.ID  
GROUP BY Utente.ID, Utente.nome, Utente.cognome  
HAVING numeroserie > 10
```

- e. avere l'elenco degli utenti che nell'ultimo mese non hanno guardato la serie TV 'Dahmer'

```
SELECT Utente.ID, Utente.nome, Utente.cognome from Utente  
where Utente.ID not in (SELECT Utente.ID  
from Utente inner join Visualizzazione  
on Utente.ID = Visualizzazione.IDUtente  
inner join Episodio  
on Visualizzazione.IDEpisodio = Episodio.ID  
inner join SerieTV  
on Episodio.IDSerieTV = SerieTV.ID  
where SerieTV.nome = 'Dahmer' AND Visualizzazione.data  
between '2022-09-20' AND '2022-10-20')  
OPPURE AND datediff (day, Visualizzazione.data,  
dateadd(day -30, getdate())) > 0)
```

4. scrivere nella quarta facciata le istruzioni SQL per:

- a. avere l'elenco delle serie TV di genere drammatico e fantascientifico.
Ordinare in ordine alfabetico visualizzando prima le serie di fantascienza e poi quelle drammatiche

```
SELECT * from SerieTV  
where genere = 'fantascienza' OR genere = 'drammatico'  
ORDER BY genere desc, SerieTV.nome
```

- b. avere l'elenco degli episodi (titolo e serie TV) in cui ha recitato Flavio Nedina.

```
SELECT Episodio.titolo, SerieTV.nome
from Episodio inner join SerieTV
on Episodio.IDSerieTV = SerieTV.ID
inner join partecipazione
on Episodio.ID = partecipazione.IDEpisodio
inner join Attore
on partecipazione.IDAttore = Attore.ID
where Attore.nome = 'Flavio' AND Attore.cognome = 'Nedina'
```

- c. avere l'elenco delle serie TV in cui compare la parola 'The' all'inizio del nome della serie TV

```
SELECT * from SerieTV
where SerieTV.nome LIKE 'The %'
```

- d. avere il numero di utenti che hanno guardato episodi nel settembre del 2022

```
SELECT count(Utente.ID) as settembre
from Utente inner join Visualizzazione
on Utente.ID = Visualizzazione.IDUtente
where Visualizzazione.data between '2022-09-01' AND
'2022-09-30'
```

5. scrivere nella prima facciata le istruzioni SQL per:

- a. creare una tabella che rappresenta una entità

```
CREATE TABLE Attore (
    ID int PRIMARY KEY IDENTITY (0,1),
    nome varchar(255),
    cognome varchar(255)
)
```

- b. creare una tabella che rappresenta una relazione collegata all'entità

```
CREATE TABLE Partecipazione (
    ID Attore FOREIGN KEY REFERENCES Attore(ID),
    IDEpisodio FOREIGN KEY REFERENCES Episodio(ID),
    nomepersonaggio varchar(255),
    cognomepersonaggio varchar(255)
)
```

- c. aggiungere il campo numero di telefono all'utente

```
ALTER TABLE Utente
ADD numeroditelefono varchar(15)
```

- d. eliminare l'attributo sesso dell'utente

```
ALTER TABLE Utente  
DROP COLUMN sesso
```

- e. eliminare tutti gli utenti a cui è scaduto l'abbonamento

```
DELETE from Utente  
where fineabbonamento < getdate()
```

- f. modificare la valutazione degli episodi passando da un voto da 1 a 10 ad un voto da 1 a 5 stelle secondo la seguente tabella:

voto	1	2	3	4	5	6	7	8	9	10
stelle	1	1	1.5	2	2.5	3	3.5	4	4.5	5

```
ALTER TABLE Visualizzazione  
ALTER COLUMN Visualizzazione decimal(2,1)  
UPDATE Visualizzazione  
SET valutazioneepisodio = valutazioneepisodio/2  
UPDATE Visualizzazione  
SET valutazioneepisodio = 1  
where valutazioneepisodio = 0.5
```