**Темa 11**

**Проект по база от данни за ТВ Сериали**

**Изготвил: Фросина Мулачка, ф.н. 1MI8000006**

1. **Обхват на модела. Дефиниране на задачата.**

Информационна система съхранява информация за ТВ сериали.

Съхранява се информация за сериала с данни за уникален номер на сериала(ид), име на сериала, начална дата на сериала, брой сезони, жанр на сериала. Съхранява се и информация за сезони с данни за номер на сезон, име на продукция, начална дата на сезона, крайна дата на сезона и брой епизоди в сезона. Съхранява се също и информация за епизоди с данни за номер на епизод, дата на излъчване на епизода и продължителност на епизода. Номерът на сезона определя уникално всеки сезон. Номерът на епизода е уникален в рамките на сезона. В информационната система се съхранява и информация за герои с данни за номер на герой — цяло положително число, уникално за всеки герой, име на герой, роля на герой и име на актьора, който играе или озвучава образа на героя. Един герой може да участва в много епизоди и в един епизод могат да участват много герои.

1. **Множества от същности и техните атрибути**

* **Сериали –** ид на сериала, име на сериала, начална дата, брой сезони, жанр
* **Сезони –** номер на сезон, начална дата на сезона, крайна дата на сезона, брой епизоди
* **Епизоди –** номер на епизод, дата на излъчване на епизода, продължителност
* **Герой –** номер на герой, име, роля, име на актьор

1. **Домейн на атрибутите**

* **Сериали –** ид на сериала: цяло положително число, име на сериала: низ, начална дата: дата, брой сезони: цяло положително число, жанр: низ
* **Сезони –** номер на сезон: цяло положително число, начална дата на сезона: дата, крайна дата на сезона: дата, брой епизоди: цяло положително число
* **Епизоди –** номер на епизод: цяло положително число, дата на излъчване на епизода: дата, продължителност: реално положително число
* **Герой –** номер на герой: цяло положително число, име: низ, роля: низ, име на актьор: низ

1. **Връзки**

* Един герой може да участва в много епизоди и в един епизод могат да участват много герои.
* Един сезон може да има много епизоди, а един епизод може да принадлежи само на един сезон.
* Един сериал може да има много сезони, а един сезон може да принадлежи само на един сериал.

***Поддържащи връзки (слаби м-ва от същности)***

* М-вото от същности Сезони се счита за слабо м-во в случая на тази информационна система, защото Сезон не може да бъде идентифициран еднозначно, без да се вземе предвид връзката му със Сериала.
* М-вото от същности Епизоди се счита за слабо м-во в случая на тази информационна система, защото Епизод не може да бъде идентифициран еднозначно, без да се вземе предвид връзката му със Сезона.

1. **Ограничения по единствена стойност, референтна цялостност и друг тип ограничения**

* **Сериали –** ид на сериала: еднозначно определя сериала
* **Сезони –** номер на сезон и сериала: еднозначно определят сезона
* **Епизоди –** номер на епизод и сезона: еднозначно определят епизода
* **Герой –** номер на герой: еднозначно определя героя

1. **Правила и проверки**

* За всички положителни числа, проверка за > 0;
* За датите, проверка дали са валидни дати;
* За датите на сезона, проверка дали начална дата < крайна дата;

1. **E/R модел на данни**

**Diagram

Description automatically generated**

1. **Релационен модел на данни**

В Е/R модела нямаме isa-йерархии, така, че първо преобразуваме множествата от

същности, после слабите множества, после връзките много-много и накрая връзките

много-един. Връзките много-един оптимизираме.

Множества същности, слаби множества и връзки:

-Series (serie\_id, serie\_name, start\_date, number\_of\_seasons, genre)

-Seasons (serie\_id, season\_number, number\_of\_episodes, from\_date, to\_date)

-Episodes (serie\_id, season\_number, episode\_number, duration, date)

-Heroes (hero\_number, hero\_name, actor\_name)

-ParticipatesIn (episode\_number, hero\_number)

Забележка: Връзките много-един в Е/R модела са поддържащи връзки и за тях не се

прави релация.

 Нямаме връзки много-един, които да оптимизираме.

Релационните схеми:

-Seasons (serie\_id, season\_number, number\_of\_episodes, from\_date, to\_date)

-Episodes (serie\_id, season\_number, episode\_number, duration, date)

имат съставен първичен ключ, който се състои от два или повече атрибута.

-Series (serie\_id, serie\_name, start\_date, number\_of\_seasons, genre)

-Seasons (serie\_id, season\_number, number\_of\_episodes, from\_date, to\_date)

-Episodes (serie\_id, season\_number, episode\_number, duration, date)

-Heroes (hero\_number, hero\_name, actor\_name)

-ParticipatesIn (episode\_id, hero\_number)

С червено са маркирани първичните ключове, със синьо външните ключове, а със  
зелено проверките и те са както следва:

Primary Keys:

-Series: PK(serie\_id)

-Seasons: PK(season\_number)

-Episodes: PK(episode\_number)

-Heroes: PK(hero\_number)

-ParticipatesIn: PK(episode\_number, hero\_number)

Foreign Keys:

FK: Seasons (serie\_id) -> Series (serie\_id)

FK: Episodes (serie\_id) -> Series (serie\_id)

FK: Episodes (season\_number) -> Seasons (season\_number)

FK: ParticipatesIn (episode\_number) -> Episodes (episode\_number)

FK: ParticipatesIn (hero\_number) -> Heroes (hero\_number)

Check:

* CK: Series (start\_date –> valid date)
* CK: Series (number\_of\_seasons > 0 –> 1,2,3…. INTEGER NUMBER)
* CK: Seasons (number\_of\_episodes > 0 –> INTEGER NUMBER)
* CK: Seasons (from\_date < to\_date –> valid dates)
* CK: Episodes (duration > 0)
* CK: Episodes (date –> valid date)

1. **Схема на базата от данни**

**Graphical user interface, text

Description automatically generated**

1. **Изгледи**

/\*read-only view\*/

CREATE VIEW V\_SERIE\_SEASON\_INFO

AS SELECT SER.SERIE\_NAME, SEA.SEASON\_NUMBER, SEA.FROM\_DATE

FROM SERIES SER, SEASONS SEA

WHERE SER.SERIE\_ID = SEA.SERIE\_ID;

SELECT \*

FROM V\_SERIE\_SEASON\_INFO

WHERE FROM\_DATE > '2017-01-01';

/\*updatable view\*/

CREATE VIEW V\_HEROES\_ALL

AS

SELECT HERO\_NAME,ACTOR\_NAME

FROM HEROES

WHERE HERO\_NAME LIKE 'J%';

INSERT INTO  V\_HEROES\_ALL(HERO\_NAME, ACTOR\_NAME)

VALUES ('Walter White Jr','RJ Mitte');

SELECT \* FROM V\_HEROES\_ALL;

INSERT INTO  V\_HEROES\_ALL(HERO\_NAME, ACTOR\_NAME)

VALUES ('Jenna Carie','Jenna Ortega');

SELECT \* FROM V\_HEROES\_ALL;

/\*check option view\*/

CREATE VIEW V\_SEASONS\_ALL\_WITH\_CK

AS

SELECT SERIE\_ID, SEASON\_NUMBER, NUMBER\_OF\_EPISODES, FROM\_DATE, TO\_DATE

FROM SEASONS

WHERE NUMBER\_OF\_EPISODES > 8

WITH CHECK OPTION;

INSERT INTO V\_SEASONS\_ALL\_WITH\_CK(SERIE\_ID, SEASON\_NUMBER, NUMBER\_OF\_EPISODES, FROM\_DATE, TO\_DATE)

VALUES ('S00005','S03',10,'2022-11-13',NULL); --OK

INSERT INTO V\_SEASONS\_ALL\_WITH\_CK(SERIE\_ID, SEASON\_NUMBER, NUMBER\_OF\_EPISODES, FROM\_DATE, TO\_DATE)

VALUES ('S00005','S04',7,'2022-11-13',NULL); --NOT OK , EPISODES < 7

SELECT \* FROM V\_SEASONS\_ALL\_WITH\_CK;

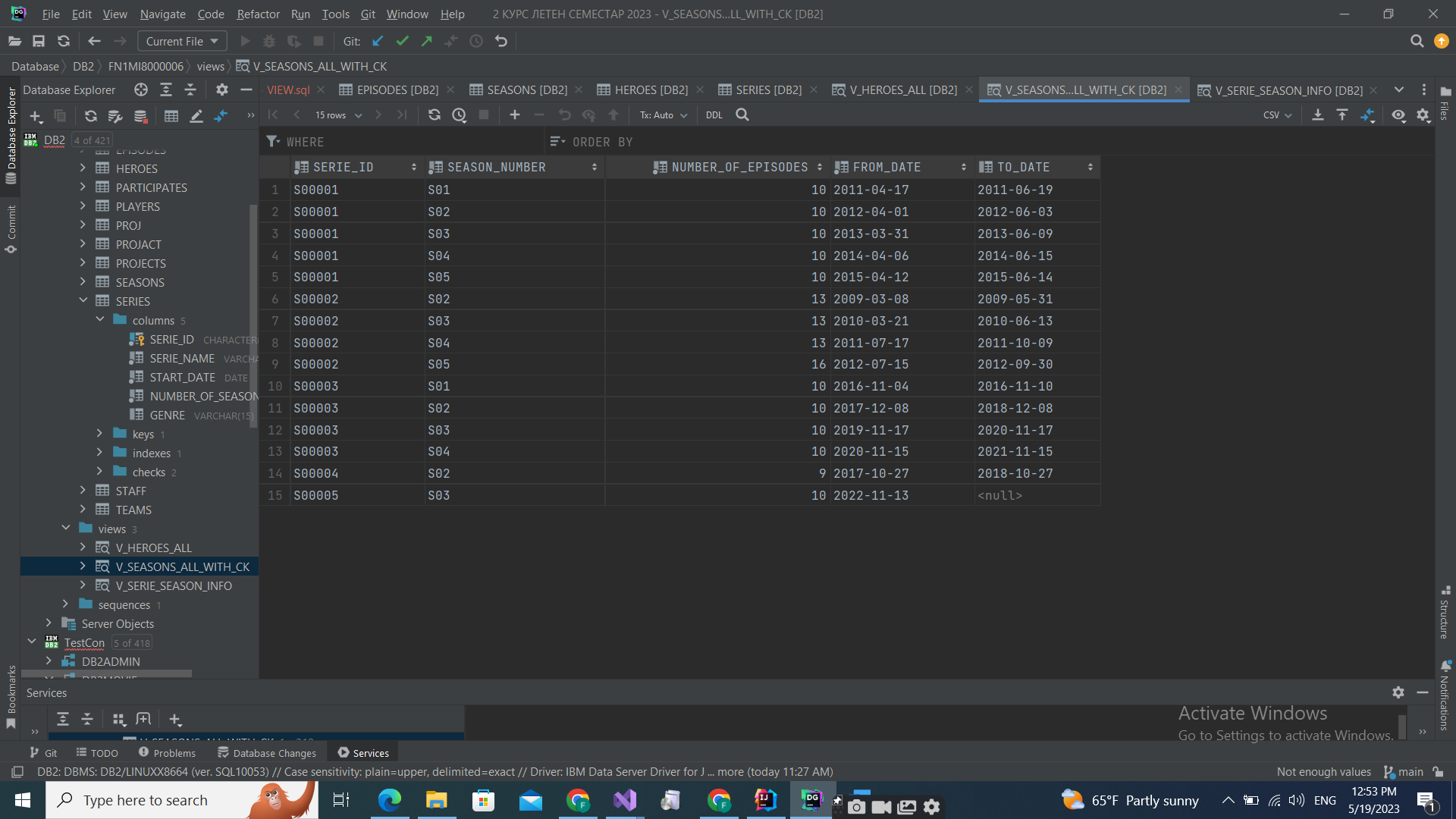
*Резултат от изгледите:*

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



1. **Приложение за достъп до базата**

public static void main(String[] args) {  
  
 DB2Test db2Obj = new DB2Test();  
 String stmnt = "";  
  
 db2Obj.openConnection();  
  
 Scanner scanner = new Scanner(System.*in*);  
 int input = 0;  
 while(input != -1)  
 {  
 System.*out*.printf("%s:\n%s\n%s\n%s\n", "Please, enter", "0 to select data","1 to insert data","2 to delete data","from SERIES table.");  
 System.*out*.printf("%s\n","Enter -1 to exit the program.");  
 input = scanner.nextInt();  
 if(input == 0)  
 {  
 stmnt = "SELECT SERIE\_ID,SERIE\_NAME,START\_DATE,NUMBER\_OF\_SEASONS,GENRE FROM FN1MI8000006.SERIES";  
 db2Obj.select(stmnt, 5);  
 }  
 else if(input == 1)  
 {  
 System.*out*.println("Enter serie id, serie name, start date, number of seasons and genre for the serie:");  
 String id = "";  
 String title = "";  
 String date = "";  
 int seasons = 0;  
 String genre = "";  
 id = scanner.next();  
 title = scanner.next();  
 date = scanner.next();  
 seasons = scanner.nextInt();  
 genre = scanner.next();  
 if(id != "" && title != "" && date != "" && genre != "")  
 {  
 stmnt = " INSERT INTO FN1MI8000006.SERIES(SERIE\_ID, SERIE\_NAME,START\_DATE,NUMBER\_OF\_SEASONS,GENRE)"  
 + " VALUES ('" + id + "','" + title + "','" + "2019-10-10" + "','" + seasons + "','" + genre + "')";  
  
 db2Obj.insert(stmnt);  
 }  
 else if(genre == "")  
 {  
 stmnt = " INSERT INTO FN1MI8000006.SERIES(SERIE\_ID, SERIE\_NAME,START\_DATE,NUMBER\_OF\_SEASONS)"  
 + " VALUES ('" + id + "','" + title + "','" + date + "','" + seasons + "')";  
  
 db2Obj.insert(stmnt);  
 }  
 else  
 {  
 System.*out*.println("Invalid data for the serie entered! Please, try again...");  
 System.*out*.println(id + title + date + seasons + genre);  
 }  
 }  
 else if(input == 2)  
 {  
 System.*out*.println("Please, enter valid id for the serie you want to delete:");  
 String id;  
 id = scanner.next();  
 stmnt = "DELETE FROM FN1MI8000006.SERIES WHERE SERIE\_ID = " + "'" + id + "' ";  
 db2Obj.delete(stmnt);  
 }  
 else if(input == -1)  
 {  
 System.*out*.println("Exiting program...");  
 db2Obj.closeConnection();  
 }  
 else {  
 System.*out*.println("Invalid input! Please, try again...");  
 }  
 }  
  
 }  
  
}

*Резултат:*

*A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated*