

Projekt

Temat: Kino

Autor: Rafał Kostrzewa

1. Treść polecenia

Utworzyć bazę danych na dowolnie wybrany temat. Baza powinna składać się z:

- Minimum 10 tabel
- Minimum 3 perspektyw (w tym co najmniej jedna zmaterializowana i funkcyjna)
- Minimum 3 triggerów / procedur wyzwolonych
- Minimum 3 funkcji (w tym co najmniej jedna z wykorzystaniem kursora oraz co najmniej jedna z wykorzystaniem transakcji)
- Minimum 3 procedur (w tym co najmniej jedna z wykorzystaniem kursora oraz co najmniej jedna z wykorzystaniem transakcji)
- Mechanizmu autoinkrementacji

2. Temat projektu

Temat bazy to Kino.

W bazie znajdują się informacje o różnych placówkach kin(MultiKino oraz CinemaCity). Dokładniej mamy informacje o znajdujących się tam salach, miejscach (Fotelach) i biletach. Również mamy podgląd do repertuaru filmowego wraz z obsadą.

Mamy możliwość w bazie m.in. :

- Sprawdzenia cen i historii utworzonych biletów.
- Wyświetlenie repertuaru filmowych seansów jak i podgląd wszystkich aktorów.
- Wyświetlenie zarobionych pieniędzy przez każdy z filmów.
- Dodawanie nowych filmów, aktorów, miejsc, biletów itd.
- Także edytowanie jak i usuwanie już istniejących bytów.

2. Model konceptualny relacyjnej bazy danych

Model konceptualny relacyjnej bazy danych to abstrakcyjny model przedstawiający strukturę danych i ich powiązania w bazie danych. Model ten składa się z trzech głównych elementów: encji, atrybutów i relacji.

Encje są przedstawieniem obiektów w rzeczywistym świecie. Każda encja ma swój unikalny identyfikator, który służy do odwoływania się do niej w innych częściach bazy danych.

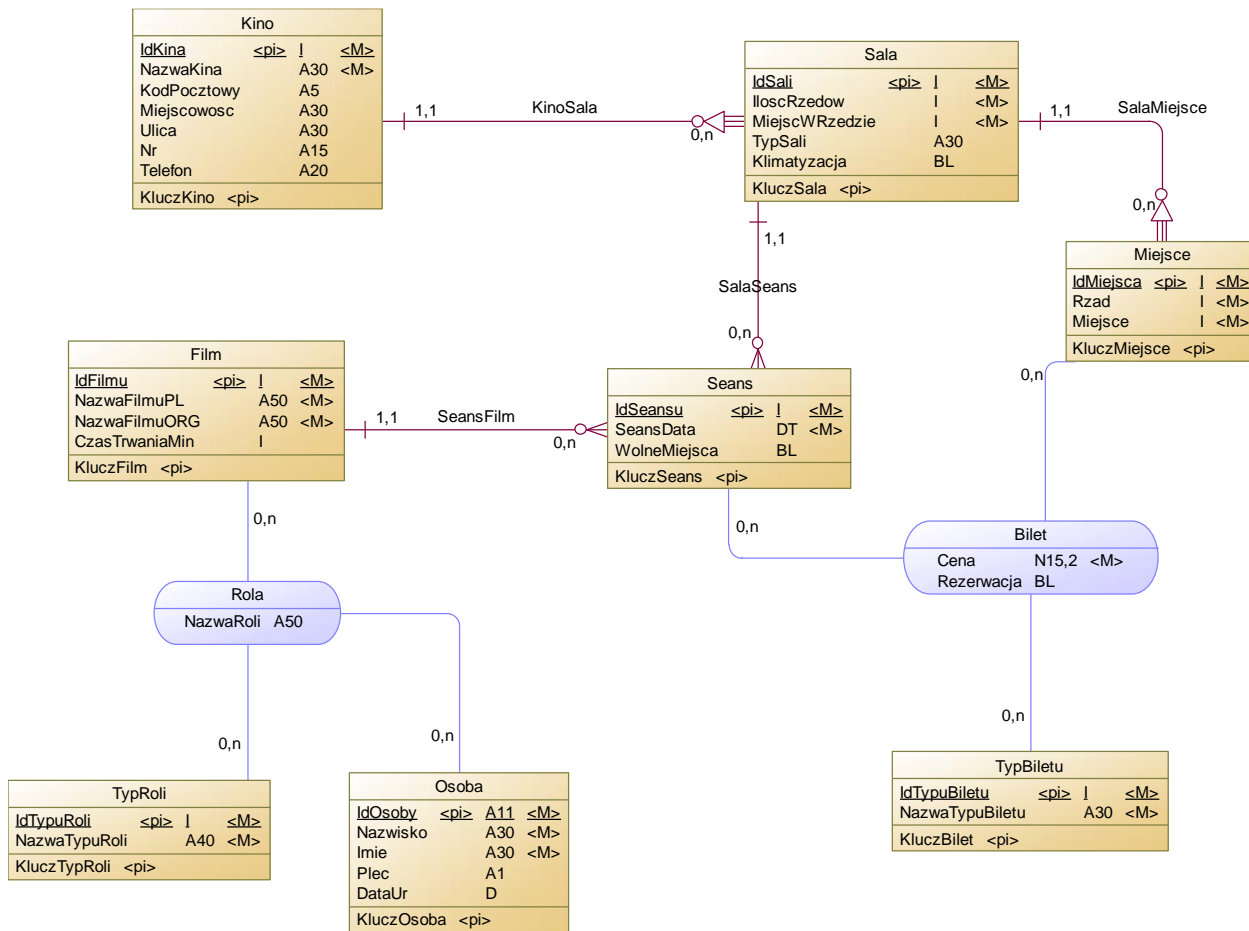
Atrybuty to cechy każdej encji. Każda encja może mieć wiele atrybutów, a każdy z nich może być przechowywany jako wartość.

Relacje określają powiązania między encjami. Mogą one być jedno- lub wielokierunkowe, a ich rodzaj zależy od sposobu, w jaki encje są ze sobą powiązane. Przykładowe relacje to relacja jeden do jednego, jeden do wielu i wiele do wielu.

Model konceptualny opiera się na zasadzie klucza głównego, który jest unikalnym identyfikatorem dla każdej encji. Klucz ten może być wykorzystywany do łączenia różnych encji i tworzenia relacji między nimi.

W modelu konceptualnym RBD encje, atrybuty i relacje są przedstawiane za pomocą diagramów ERD. Diagramy te służą do wizualizacji struktury bazy danych.

Model konceptualny relacyjnej bazy danych



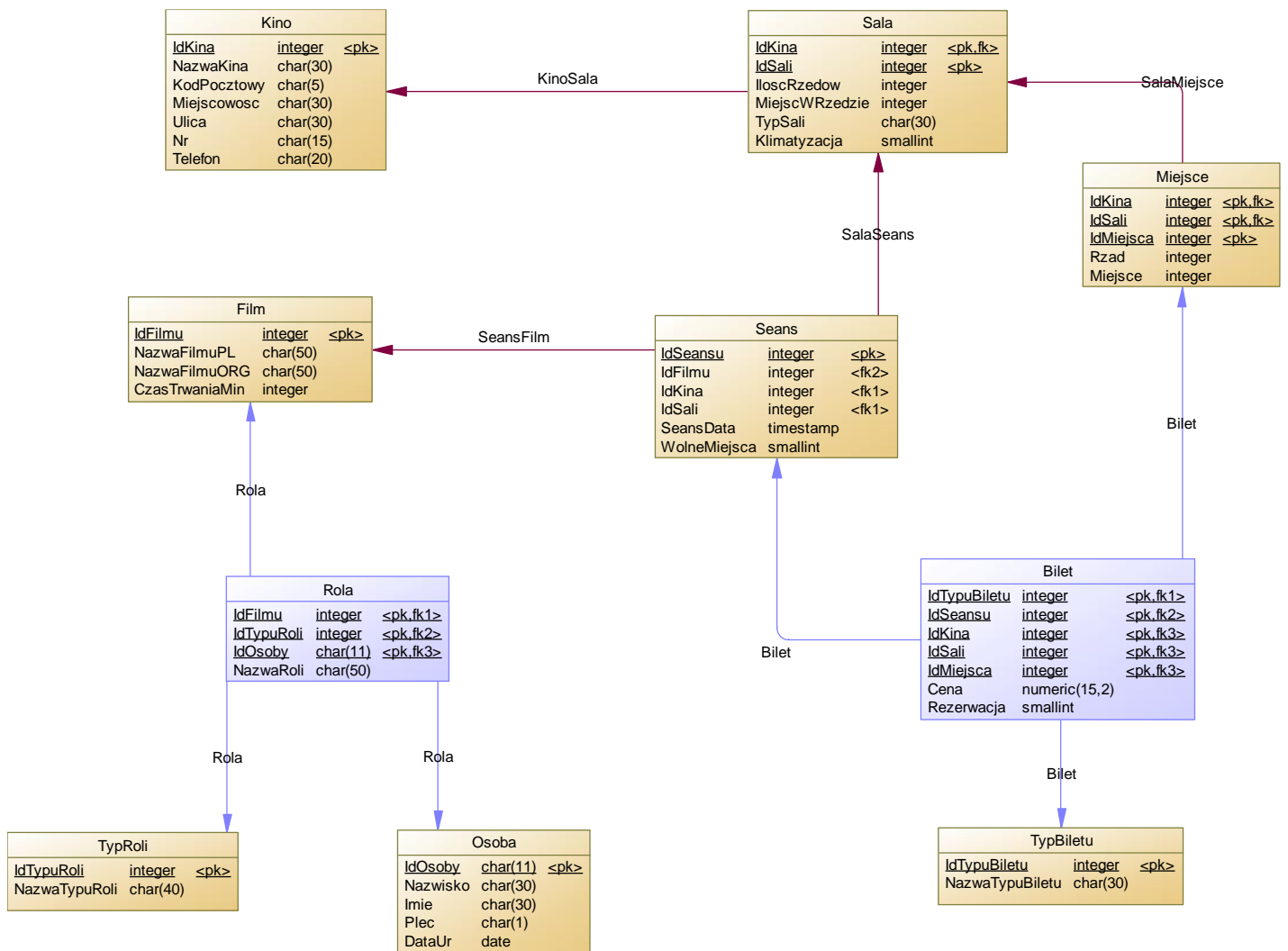
4. Model fizyczny relacyjnej bazy danych

Model fizyczny relacyjnej bazy danych to sposób przedstawienia struktury i organizacji danych w bazie danych relacyjnej. Ten model opiera się na koncepcji tabeli, która składa się z wierszy i kolumn, z każdym wierszem reprezentującym rekord danych, a każdą kolumną reprezentującą atrybut danych.

Model fizyczny określa szczegółowe informacje o strukturze tabel, w tym typach danych, długości, ograniczeniach, kluczach głównych, kluczach obcych i indeksach. W modelu fizycznym każda tabela ma swoją własną fizyczną reprezentację na dysku twardym, z wierszami i kolumnami przechowywanymi w odpowiednich plikach.

Model fizyczny relacyjnej bazy danych zapewnia wysoką wydajność i optymalizację zapytań, umożliwiając szybki dostęp do danych. Umożliwia również łatwe utrzymanie i zarządzanie danymi, dzięki czemu baza danych może być łatwo zintegrowana z innymi systemami i aplikacjami.

Model fizyczny relacyjnej bazy danych



5. Widoki

W zadaniu należało zaimplementować minimum 3 widoki w tym jeden zmaterializowany

5.1 *BiletyNaPrzyszleSeansyW*

```
ALTER VIEW "DBA"."BiletyNaPrzyszleSeansyW"  
AS  
SELECT Seans.SeansData, Kino.NazwaKina ,TypBiletu.NazwaTypuBiletu,  
       Bilet.Cena ,Seans.IdSeansu,Film.NazwaFilmuPL, Bilet.Rezerwacja  
FROM Seans  
      INNER JOIN Sala ON Sala.IdSali=Seans.IdSali AND Sala.IdKina=Seans.IdKina  
      INNER JOIN Kino ON Kino.IdKina=Sala.IdKina  
      INNER JOIN Film ON Film.IdFilmu=Seans.IdFilmu  
      INNER JOIN Bilet ON Seans.IdSeansu=Bilet.IdSeansu  
      INNER JOIN TypBiletu ON TypBiletu.IdTypuBiletu=Bilet.IdTypuBiletu  
WHERE Seans.SeansData>getdate()
```

```
SELECT * FROM BiletyNaPrzyszleSeansyW
```

Results

	SeansData	NazwaKina	NazwaTypuBiletu	Cena	IdSeansu	NazwaFilmuPL	Rezerwacja
1	2023-02-22 10:00:00.000	MultiKino	Ulgowy	10,00	3	Człowiek ze stali	1
2	2023-02-22 17:00:00.000	MultiKino	Normalny	20,00	6	Piraci z Karaibów 1	0
3	2023-02-22 17:00:00.000	MultiKino	Ulgowy	10,00	6	Piraci z Karaibów 1	1
4	2023-02-22 17:00:00.000	MultiKino	Ulgowy	10,00	6	Piraci z Karaibów 1	1
5	2023-02-22 17:00:00.000	MultiKino	Normalny	20,00	6	Piraci z Karaibów 1	0
6	2023-02-22 10:00:00.000	MultiKino	Normalny	20,00	7	Harry Potter i Kamień Filozoficzny	0
7	2023-02-23 12:00:00.000	CinemaCity	Ulgowy	10,00	11	Szklana pułapka	0

5.2 ObsadaFilmowW

```
ALTER VIEW "DBA"."ObsadaFilmowW"
```

```
AS
```

```
Select Film.NazwaFilmuPL,Film.NazwaFilmuORG,TypRoli.NazwaTypuRoli ,  
Rola.NazwaRoli , Osoba.Nazwisko, Osoba.Imie
```

```
FROM TypRoli
```

```
    INNER JOIN Rola ON TypRoli.IdTypuRoli=Rola.IdTypuRoli
```

```
    INNER JOIN Osoba ON Osoba.IdOsoby=Rola.IdOsoby
```

```
    INNER JOIN Film ON Film.IdFilmu=Rola.IdFilmu
```

```
ORDER BY Film.IdFilmu ASC , TypRoli.IdTypuRoli ASC
```

ements

```
SELECT * FROM ObsadaFilmowW
```

Results						
	NazwaFilmuPL	NazwaFilmuORG	NazwaTypuRoli	NazwaRoli	Nazwisko	Imie
1	Harry Potter i Kamień Filozoficzny	Harry Potter and the Philosopher's Stone	Pierwszoplanowa	Harry Potter	Radcliffe	Daniel
2	Harry Potter i Kamień Filozoficzny	Harry Potter and the Philosopher's Stone	Drugoplanowa	Hermiona	Watson	Emma
3	Harry Potter i Kamień Filozoficzny	Harry Potter and the Philosopher's Stone	Drugoplanowa	Ron	Grint	Rupert
4	Harry Potter i Komnata Tajemnic	Harry Potter and the Chamber of Secrets	Pierwszoplanowa	Harry Potter	Radcliffe	Daniel
5	Harry Potter i Komnata Tajemnic	Harry Potter and the Chamber of Secrets	Drugoplanowa	Hermiona	Watson	Emma
6	Harry Potter i Komnata Tajemnic	Harry Potter and the Chamber of Secrets	Drugoplanowa	Ron	Grint	Rupert
7	Człowiek ze stali	Man of Steel	Pierwszoplanowa	SuperMan	Cavil	Henry
8	Człowiek ze stali	Man of Steel	Epizodyczna	Lios Lane	Adams	Amy
9	Piraci z Karaibów 1	Pirates of the Caribbean 1	Drugoplanowa	Elizabeth	Knightley	Keira
10	Piraci z Karaibów 1	Pirates of the Caribbean 1	Pierwszoplanowa	Jack Sparrow	Depp	Johnny
11	Szklana pułapka	Die hard	Pierwszoplanowa	John McClane	Wills	Bruce
12	Harry Potter i Kamień Filozoficzny	Harry Potter and the Philosopher's Stone	Pierwszoplanowa	Lord Voldemort	Binka	Mateusz

5.3 RozpiskaSaliFilmowW

```
ALTER VIEW "DBA"."RozpiskaSaliFilmowW"  
AS  
SELECT Sala.IdSali, Kino.NazwaKina, Film.NazwaFilmuPL  
FROM Sala  
    INNER JOIN Kino ON Kino.IdKina=Sala.IdKina  
    INNER JOIN Seans ON Seans.IdSali=Sala.IdSali  
    INNER JOIN Film ON Film.IdFilmu=Seans.IdFilmu  
  
GROUP BY Sala.IdSali, Kino.NazwaKina, Film.NazwaFilmuPL  
ORDER BY Kino.NazwaKina
```

```
SELECT * FROM RozpiskaSaliFilmowW
```

	IdSali	NazwaKina	NazwaFilmuPL
1	1	CinemaCity	Człowiek ze stali
2	1	CinemaCity	Harry Potter i Komnata Tajemnic
3	1	CinemaCity	Harry Potter i Kamień Filozoficzny
4	2	CinemaCity	Szklana pułapka
5	2	CinemaCity	Piraci z Karaibów 1
6	1	MultiKino	Człowiek ze stali
7	1	MultiKino	Harry Potter i Komnata Tajemnic
8	1	MultiKino	Harry Potter i Kamień Filozoficzny
9	2	MultiKino	Szklana pułapka
10	2	MultiKino	Piraci z Karaibów 1
11	3	MultiKino	Harry Potter i Komnata Tajemnic
12	3	MultiKino	Harry Potter i Kamień Filozoficzny

5.4 ZarobkiKinZDniaW

```
CREATE MATERIALIZED VIEW "DBA"."ZarobkiKinZDniaW"  
AS  
SELECT Kino.NazwaKina,DATE(Seans.SeansData) AS  
ZyskiZDnia,TypBiletu.NazwaTypuBiletu, SUM(Bilet.Cena) AS SumaCenBiletow  
FROM Kino  
      INNER JOIN Sala ON Kino.IdKina=Sala.IdKina  
      INNER JOIN Seans ON Seans.IdSali=Sala.IdSali AND Sala.IdKina=Seans.IdKina  
      INNER JOIN Film ON Film.IdFilmu=Seans.IdFilmu  
      INNER JOIN Bilet ON Seans.IdSeansu=Bilet.IdSeansu  
      INNER JOIN TypBiletu On Bilet.IdTypuBiletu=TypBiletu.IdTypuBiletu  
WHERE Seans.SeansData< '2023-02-20 00:00:00.000'  
AND Seans.SeansData> '2023-02-19 00:00:00.000'  
GROUP BY Kino.NazwaKina,Seans.SeansData,  
TypBiletu.NazwaTypuBiletu ,Bilet.Cena
```

```
SELECT * FROM ZarobkiKinZDniaW
```

	NazwaKina	ZyskiZDnia	NazwaTypuBiletu	SumaCenBiletow
1	CinemaCity	2023-02-19	Ulgowy	10,00
2	CinemaCity	2023-02-19	Normalny	60,00

6. Triggery

Należy stworzyć minimum 3 triggery / procedury wyzwolone

6.1 CenyBiletow

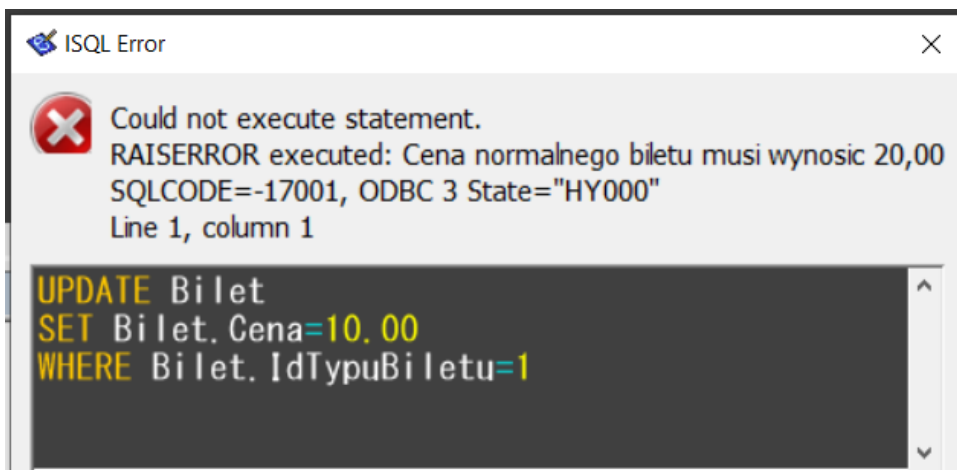
```
ALTER TRIGGER "CenyBiletow" BEFORE INSERT, UPDATE
ORDER 1 ON "DBA"."Bilet"
REFERENCING NEW AS Bilet
FOR EACH ROW
BEGIN
    DECLARE Cena NUMERIC(15,2);

    SET Cena=(SELECT TOP 1 Bilet.Cena FROM Bilet
ORDER BY Bilet.Seq DESC);

    CASE
    WHEN Bilet.IdTypuBiletu=1 THEN
        IF Bilet.Cena!=20.00
        THEN RAISERROR 17001 'Cena normalnego biletu musi wynosic 20,00';
        ENDIF;

    WHEN Bilet.IdTypuBiletu=2 THEN
        IF Bilet.Cena!=10.00
        THEN RAISERROR 17002 'Cena ulgowego biletu musi wynosic 10,00';
        ENDIF;
    END
END
```

```
1 UPDATE Bilet
2 SET Bilet.Cena=10.00
3 WHERE Bilet.IdTypuBiletu=1
```



6.2 CzyFotelIstnieje

```
ALTER TRIGGER "CzyFotelIstnieje" BEFORE INSERT, UPDATE
ORDER 1 ON "DBA"."Miejsce"
REFERENCING NEW AS Miejsce
FOR EACH ROW
BEGIN

    DECLARE rzadow INTEGER;
    DECLARE miejsc INTEGER;
    DECLARE tmp1 INTEGER;
    DECLARE tmp2 INTEGER;
    DECLARE tmp3 INTEGER;

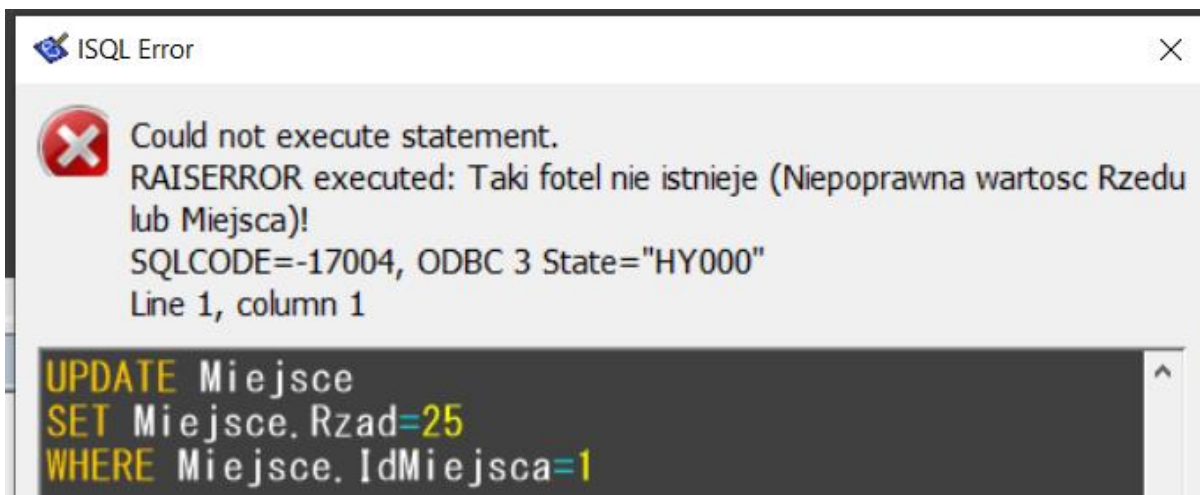
    SET tmp1=(SELECT TOP 1 Miejsce.Rzad FROM Miejsce
        ORDER BY Miejsce.IdMiejscas DESC);
    SET tmp2=(SELECT TOP 1 Miejsce.Miejscas FROM Miejsce
        ORDER BY Miejsce.IdMiejscas DESC);
    SET tmp3=(SELECT TOP 1 Miejsce.IdSali FROM Miejsce
        ORDER BY Miejsce.IdMiejscas DESC);

    SET rzadow=(SELECT TOP 1 Sala.IloscRzadow FROM Sala
        WHERE Sala.IdSali=tmp3);
    SET miejsc=(SELECT TOP 1 Sala.MiejscasWRzedzie FROM Sala
        WHERE Sala.IdSali=tmp3);

    IF (tmp1>rzadow OR tmp2>miejsc)
    THEN RAISERROR 17004 'Taki fotel nie istnieje (Niepoprawna wartosc Rzedu lub Miejsca)!'
    ENDIF;

END
```

```
UPDATE Miejsce
SET Miejsce.Rzad=25
WHERE Miejsce.IdMiejscas=1
```



6.3 WolneMiejsca

```
ALTER TRIGGER "WolneMiejsca" BEFORE INSERT, UPDATE
ORDER 2 ON "DBA"."Bilet"
REFERENCING NEW AS Bilet
FOR EACH ROW
BEGIN
    DECLARE status1 INTEGER;
    DECLARE status2 INTEGER;

    SET status1=(SELECT TOP 1 Bilet.IdSeansu FROM Bilet
    ORDER BY Bilet.Seq DESC);
    SET status2=(SELECT Seans.WolneMiejsca FROM Seans
    WHERE Seans.IdSeansu=status1);

    IF status2=0
    THEN RAISERROR 17003 'Na ten seans nie ma juz wolnych miejsc!';
    ENDIF;
END
```

```
INSERT INTO Bilet (IdTypuBiletu, IdSeansu, IdKina, IdSali, IdMiejsca, Cena, Rezerwacja)
VALUES (1, 5, 2, 2, 16, 20.00, 0)
```

ISQL Error



Could not execute statement.
RAISERROR executed: Na ten seans nie ma juz wolnych miejsc!
SQLCODE=-17003, ODBC 3 State="HY000"
Line 1, column 1

```
INSERT INTO Bilet (IdTypuBiletu, IdSeansu, IdKina, IdSali, IdMiejsca, Cena, Rezerwacja)
VALUES (1, 5, 2, 2, 16, 20.00, 0)
```

Columns	Constraints	Referencing Constraints	Indexes	Text Indexes	Triggers	Permissions	D
	IdSeansu	IdFilmu	IdKina	IdSali	SeansData	WolneMiejsca	
1	1	1	1	1	2023-02-19 11:40:00.000	1	
2	2	2	1	1	2023-02-22 15:40:00.000	1	
3	3	3	2	1	2023-02-22 10:00:00.000	1	
4	4	4	2	2	2023-02-22 19:00:00.000	1	
5	5	5	2	2	2023-02-22 21:00:00.000	0	
6	6	4	2	2	2023-02-22 17:00:00.000	1	

7. Procedury

Należy stworzyć minimum 3 procedury (w tym co najmniej jedna z wykorzystaniem kursora oraz co najmniej jedna z wykorzystaniem transakcji)

7.1 DodajFilmProc

```
ALTER PROCEDURE "DBA"."DodajFilmProc"( IN NazwaPL VARCHAR(50),IN
NazwaORG CHAR(50),IN Czas INTEGER)

BEGIN

        INSERT INTO Film
(NazwaFilmuPL,NazwaFilmuORG,CzasTrwaniaMin)
VALUES (NazwaPL, NazwaORG, Czas);

MESSAGE 'Dodano nowy film -> Tytuł: ' || NazwaPL || ' Czas Trwania:' || Czas TO
CLIENT ;

END
```

```
CALL DodajFilmProc('ABC', 'DBA', 200)
```

Results

Dodano nowy film -> Tytuł: ABC Czas Trwania:200

1 row(s) affected

Execution time: 0.004 seconds

Procedure completed

Film (DBA)

Columns Constraints Referencing Constraints Indexes Text Indexes Triggers Permissions Dependent Views Data

	IdFilmu	NazwaFilmuPL	NazwaFilmuORG	CzasTrwaniaMin
1	1	Harry Potter i Kamień Filozoficzny	Harry Potter and the Philosopher's Stone	152
2	2	Harry Potter i Komnata Tajemnic	Harry Potter and the Chamber of Secrets	161
3	3	Człowiek ze stali	Man of Steel	143
4	4	Piraci z Karaibów 1	Pirates of the Caribbean 1	140
5	5	Szklana pułapka	Die hard	123
6	7	ABC	DBA	200

7.2 WyświetlOsobyRoleProc

```
ALTER PROCEDURE "DBA"."WyświetlOsobyRoleProc"( IN ImieAktora CHAR(30) )

BEGIN
    DECLARE licznik INTEGER;
    SET licznik=0;
    FOR petla1 AS kursor1 CURSOR FOR
        (SELECT Osoba.Imie, Osoba.Nazwisko , Osoba.DataUr,Rola.NazwaRoli,
        Film.NazwaFilmuPL
        FROM Osoba
        INNER JOIN Rola ON Rola.IdOsoby=Osoba.IdOsoby
        INNER JOIN Film ON Film.IdFilmu=Rola.IdFilmu
        WHERE Osoba.Imie=ImieAktora)
    DO
        SET licznik=licznik+1;
        MESSAGE 'Rola nr.' || licznik || ' Aktor: ' || Nazwisko ||
        ' ' || Imie || ' jako ' || NazwaRoli || ' w filmie ' || NazwaFilmuPL TO CLIENT ;

    END FOR;
END
```

```
CALL WyświetlOsobyRoleProc('Daniel')
```

Results

Rola nr.1 Aktor: Radcliffe Daniel jako Harry Potter w filmie Harry Potter i Kamień Filozoficzny
Rola nr.2 Aktor: Radcliffe Daniel jako Harry Potter w filmie Harry Potter i Komnata Tajemnic
Execution time: 0.003 seconds
Procedure completed

7.3 ZmianaCzasuFilmuProc

```
ALTER PROCEDURE "DBA"."ZmianaCzasuFilmuProc"( IN TytulFilmu CHAR(50), IN czas
INTEGER )
AS
BEGIN
BEGIN TRANSACTION zmiana
    UPDATE Film
    SET Film.CzasTrwaniaMin=czas
    WHERE Film.NazwaFilmuPL=TytulFilmu

    IF ( SELECT Film.CzasTrwaniaMin
    FROM Film WHERE Film.NazwaFilmuPL=TytulFilmu ) > 600
        BEGIN
            ROLLBACK TRANSACTION zmiana
            RETURN
        END

    IF( SELECT Film.CzasTrwaniaMin FROM Film
        WHERE Film.NazwaFilmuPL=TytulFilmu ) < 5
        BEGIN
            ROLLBACK TRANSACTION zmiana
            RETURN
        END

    ELSE
        COMMIT TRANSACTION zmiana
END
```

```
1 CALL DodajFilmProc('ABC', 'DBA', 200)
```

6	12	ABC	DBA	200
---	----	-----	-----	-----

```
1 CALL ZmianaCzasuFilmuProc('ABC', 300)
```

6	12	ABC	DBA	300
---	----	-----	-----	-----

Statements

```
1 CALL ZmianaCzasuFilmuProc('ABC', 1)
```

6	12	ABC	DBA	300
---	----	-----	-----	-----

8. Funkcje

Należy stworzyć minimum 3 funkcji (w tym co najmniej jedna z wykorzystaniem kursora oraz co najmniej jedna z wykorzystaniem transakcji)

8.1 ZarobkiFilmu

```
ALTER FUNCTION "DBA"."ZarobkiFilmu"( IN TytulFilmy CHAR(50) )
RETURNS INTEGER
NOT DETERMINISTIC
BEGIN
    DECLARE "Zyski" INTEGER;
    SET Zyski=(SELECT SUM(Bilet.Cena)
    FROM Kino
    INNER JOIN Sala ON Kino.IdKina=Sala.IdKina
    INNER JOIN Seans ON Seans.IdSali=Sala.IdSali AND Sala.IdKina=Seans.IdKina
    INNER JOIN Film ON Film.IdFilmu=Seans.IdFilmu
    INNER JOIN Bilet ON Seans.IdSeansu=Bilet.IdSeansu
    INNER JOIN TypBiletu On Bilet.IdTypuBiletu=TypBiletu.IdTypuBiletu
    WHERE Film.NazwaFilmuPL=TytulFilmy);

    RETURN "Zyski";
END
```

```
1 SELECT Film.IdFilmu, NazwaFilmuPL, CzasTrwaniaMin ,
2 ZarobkiFilmu('Piraci z Karaibów 1')
3 FROM Film
4 WHERE Film.NazwaFilmuPL='Piraci z Karaibów 1'
```

Results				
	IdFilmu	NazwaFilmuPL	CzasTrwaniaMin	ZarobkiFilmu('Piraci z Karaibów 1')
1	4	Piraci z Karaibów 1	140	60

8.2 CzyJestKlimatyzacja

```
ALTER FUNCTION "DBA"."CzyJestKlimatyzacja"( In NazwaK CHAR(30) )
RETURNS CHAR(1)
DETERMINISTIC
BEGIN
    DECLARE CzyKlima CHAR(1);
    DECLARE tmp CHAR(1);
    DECLARE kursor CURSOR FOR
    (SELECT Sala.Klimatyzacja
     FROM Sala
     INNER JOIN Kino ON Sala.IdKina=Kino.IdKina
     WHERE Kino.NazwaKina=NazwaK );

    SET CzyKlima='N';
    OPEN kursor;
    petla1: LOOP
        FETCH NEXT kursor INTO tmp;
        IF SQLCODE <> 0 THEN
            LEAVE petla1;
        ENDIF;

        IF tmp=1
            THEN SET CzyKlima='T';
        ENDIF;
    END LOOP;

    CLOSE kursor;
    RETURN CzyKlima;
END
```

	IdKina	IdSali	IloscRzedow	MiejscWRzedzie	TypSali	Klimatyzacja
1	1	1	10	5	Filmy 2D	1
2	1	2	15	10	Filmy 2D	1
3	2	1	5	5	Filmy 3D	0
4	2	2	3	3	VIP	0
5	2	3	10	5	Filmy 2D	0

statements

```
1 SELECT CzyJestKlimatyzacja('MultiKino')
```

Results

CzyJestKlimatyzacja('MultiKino')

1 N

```
1 SELECT CzyJestKlimatyzacja('CinemaCity')
```

Results

CzyJestKlimatyzacja('CinemaCity')

1 T

8.3 *UsuwanieSeansu*

```
ALTER FUNCTION "DBA"."UsuwanieSeansu"(IN @IdS INTEGER)
RETURNS INTEGER
NOT DETERMINISTIC

BEGIN
/*
DECLARE "@IdUseansu" INTEGER;
DECLARE "@tmp" INTEGER;

SET @IdUseansu=@IdS;

SET @tmp = (SELECT COUNT(Bilet.IdMiejsca)
            FROM Bilet
            INNER JOIN Seans ON Bilet.IdSeansu=Seans.IdSeansu
            WHERE Bilet.IdSeansu=@IdS);

BEGIN TRANSACTION zmiana

    DELETE FROM Seans WHERE Seans.IdSeans=@idS

        UPDATE Seans
        SET Seans.IdSeans=0 WHERE Seans.IdSeans=@idS

        IF @tmp !=0
        BEGIN
            ROLLBACK TRANSACTION zmiana
            RETURN
        END

    ELSE
        COMMIT TRANSACTION zmiana

        RETURN "@IdUseansu";
*/
END
```