

MATEUSZ KUCAJ GRUPA 4 110953

PRZYKŁADOWE DOPISYWANIE DANYCH DO BAZY:

```
INSERT INTO egzemplarz VALUES(16,13,1,3,5);
```

```
INSERT INTO film(id_film,id_rezyser,id_kategoria,id_recenzja, id_recenzja,nazwa,opis,dlugosc)
VALUES(16,1,1,2,'Bekarty wojny','W okupowanej przez nazistów Francji oddział złożony z
Amerykanów żydowskiego pochodzenia planuje zamach na Hitlera.','2h3min');
```

```
INSERT INTO klient VALUES(16,'Mateusz','Kowalski','689232901','matkow@gmail.com');
```

```
UPDATE klient SET imie='Nowak' WHERE id_klient=16;
```

PRZYKŁADOWE ZAPYTANIA MODYFIKUJĄCE I USUWAJĄCE DANE Z BAZY:

```
UPDATE egzemplarz SET id_film=16 WHERE id_egzemplarz=16;
```

```
UPDATE film SET id_recenzja=3 WHERE id_film=16;
```

```
DELETE FROM egzemplarz WHERE id_film=16;
```

```
DELETE FROM film WHERE id_film=16;
```

ZAPYTANIA SELECT

```
SELECT p.id_pracownik, w.id_wypozyczenie FROM wypozyczenie AS w INNER JOIN pracownik AS p
ON w.id_pracownik=p.id_pracownik;
```

```
SELECT f.nazwa,k.nazwa FROM film as f LEFT OUTER JOIN kategoria as k ON
k.id_kategoria=f.id_kategoria;
```

```
SELECT nazwa FROM film WHERE id_film>10 GROUP BY nazwa;
```

```
SELECT imie,nazwisko,stanowisko FROM pracownik WHERE stanowisko='Sprzedawca' ORDER BY imie
ASC;
```

```
SELECT p.id_pracownik,(SELECT COUNT(id_klient) FROM wypozyczenie AS w WHERE  
p.id_pracownik=w.id_pracownik) AS LiczbaKlientow FROM pracownik AS p;
```

```
SELECT COUNT(id_wypozyczenie),data_wypozyczenia,data_oddania FROM wypozyczenie GROUP BY  
data_oddania,data_wypozyczenia HAVING data_oddania is null;
```

```
SELECT nazwa FROM film WHERE id_kategoria IN ('1','3','4');
```

```
SELECT nazwa FROM film WHERE id_film=ANY(SELECT id_film FROM egzemplarz WHERE  
id_oplata=6);
```

```
SELECT nazwisko FROM rezyser WHERE id_rezyser=ALL(SELECT id_rezyser FROM film WHERE  
id_film=3);
```

```
SELECT nazwa FROM nosnik WHERE EXISTS (SELECT id_nosnik FROM egzemplarz WHERE  
egzemplarz.id_nosnik=nosnik.id_nosnik AND id_egzemplarz=3);
```

```
SELECT imie,nazwisko FROM klient WHERE imie LIKE 'S%a';
```

```
SELECT p.imie,p.nazwisko,w.id_wypozyczenie FROM pracownik as p INNER JOIN wypozyczenie as w  
ON p.id_pracownik=w.id_pracownik WHERE p.imie LIKE '%a';
```

```
SELECT COUNT(id_egzemplarz) FROM egzemplarz WHERE id_nosnik=3;
```

```
SELECT film.id_film,oplata.cena FROM film INNER JOIN egzemplarz ON  
egzemplarz.id_film=film.id_film INNER JOIN oplata ON egzemplarz.id_oplata=oplata.id_oplata  
WHERE oplata.cena>10;
```

```
SELECT id_egzemplarz,(SELECT COUNT(id_wypozyczenie) FROM wypozyczenie WHERE  
egzemplarz.id_egzemplarz=wypozyczenie.id_egzemplarz) AS IloscWypozyczen FROM egzemplarz;
```

PROCEDUREY

```
SET TERM ^ ;

CREATE PROCEDURE WYPOZYCZ3 (

    I_IMIE varchar(50),
    I_NAZWISKO varchar(50),
    I_NR_TELEFONU varchar(50),
    I_E_MAIL varchar(100),
    I_ID_PRACOWNIK integer,
    I_ID_EGZ integer,
    I_DATA date )
AS

declare kID int;

BEGIN

INSERT INTO klient(imie,nazwisko,nr_telefonu,e_mail)
values(:i_imie,:i_nazwisko,:i_nr_telefonu,:i_e_mail);

SELECT id_klient FROM klient WHERE imie=:i_imie AND nazwisko=:i_nazwisko AND
nr_telefonu=:i_nr_telefonu AND e_mail=:i_e_mail INTO :kID;

INSERT INTO wypozyczenie(id_klient,id_pracownik,id_egzemplarz,data_wypozyczenia)
values(:kID,:i_id_pracownik,:i_id_egz,:i_data);

END^

SET TERM ; ^
```

```
SET TERM ^ ;

CREATE PROCEDURE EGZEM (

    IFILM varchar(100),
    IZAWIERA integer,
    INOSNIK varchar(10),
    IOPLATA integer )
AS

declare fID int;

declare nID int;

declare oID int;
```

```

BEGIN
SELECT id_film FROM film where nazwa=:iFILM INTO :fID;
SELECT id_oplata FROM oplata where cena=:iOPLATA INTO :oID;
SELECT id_nosnik FROM nosnik where nazwa=:iNOSNIK INTO :nID;
IF (:fID is NULL) THEN exception pusty;
IF (:nID is NULL) THEN exception nosn;
IF (:oID is NULL) THEN exception op;
INSERT INTO egzemplarz(id_oplata,id_nosnik,id_zawiera,id_film) values(:oID,:nID,:izawiera,:fID);
END^
SET TERM ; ^

```

WYZWALACZE

```

SET TERM ^ ;
CREATE TRIGGER TRIG_SPRAWDZ_STANOWISKO FOR PRACOWNIK ACTIVE
AFTER delete POSITION 0
AS
BEGIN
    if (old.stanowisko='Szef') then exception stanowisko;
END^
SET TERM ; ^

```

```

SET TERM ^ ;
CREATE TRIGGER TRIG_SPRAWDZ_KATEGORIE FOR KATEGORIA ACTIVE
BEFORE insert POSITION 0
AS
DECLARE
istnieje1 int;
BEGIN
SELECT COUNT(nazwa) FROM KATEGORIA where nazwa=new.nazwa INTO istnieje1;
    if (istnieje1>0) then exception istnieje;
END^
SET TERM ; ^

```

```
SET TERM ^ ;  
CREATE TRIGGER TRIG_SPRAWDZ_OCENE FOR RECENZJA ACTIVE  
BEFORE insert OR update POSITION 0  
AS  
BEGIN  
    if (new.ocena>10) then exception name;  
END^  
SET TERM ; ^
```