**Bazy danych - sprawozdanie z projektu**

# Podstawowe informacje

* **Autorzy projektu:** Natalia Adamczyk (124329), Kacper Roczniak (124315),

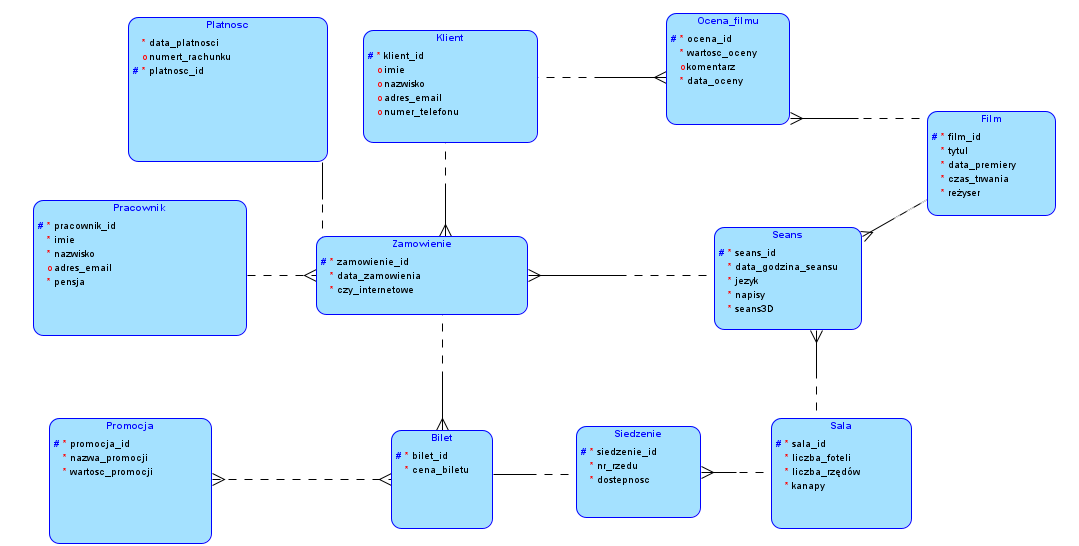
Aleksandra Kwaśniewska (124265)

* **Nazwa grupy:** K(i)NO
* **Temat:** Baza danych dla kina
* **Wybrany zakres (ocena):** Ocena bardzo dobra
* **Baza danych:** Projekt\_kino, Username: ie83831, Password: NA83831

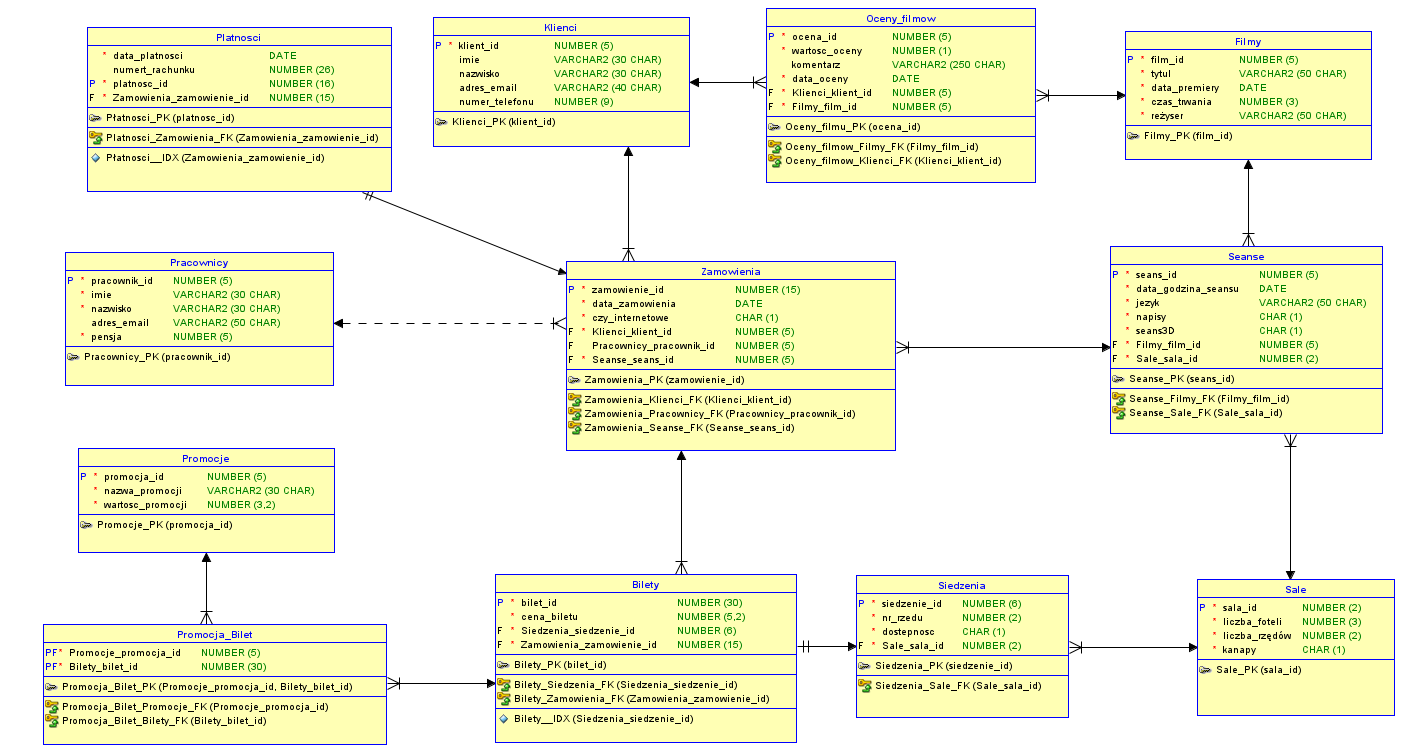
# Wprowadzenie – opis studium przypadku

Stworzyliśmy bazę danych dla kina, uwzględniliśmy wszystkie obiekty świata biznesowego związane z działalnością kina. Najważniejszą encją w naszym modelu jest zamówienie, które klient może złożyć internetowo lub w kasie. Jeżeli zamówienie zostało złożone w kasie przy zamówieniu jest dołączone id pracownika, jeżeli internetowo mamy informacje o numerze rachunku klienta. Zamówienie może dotyczyć tylko jednego seansu, ale może zawierać bilety objęte różnymi promocjami. Promocje sumują się. Każdy bilet ma przypisane miejsce w sali. Klient może wystawić opinię na temat filmu.

# Model związków encji



# Model relacyjnej bazy danych



# Schemat bazy - tabele

## --Siedzenia

CREATE TABLE siedzenia (

siedzenie\_id NUMBER(6) NOT NULL,

nr\_rzedu NUMBER(2) NOT NULL,

dostepnosc CHAR(1) NOT NULL,

sale\_sala\_id NUMBER(2) NOT NULL

);

ALTER TABLE siedzenia ADD CONSTRAINT siedzenia\_pk PRIMARY KEY ( siedzenie\_id );

ALTER TABLE siedzenia

ADD CONSTRAINT siedzenia\_sale\_fk FOREIGN KEY ( sale\_sala\_id )

REFERENCES sale ( sala\_id );

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 1);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 1);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 1);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 1);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 1);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 1);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 1);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 1);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 1);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 1);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 2);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 2);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 2);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 2);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 2);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 2);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 2);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 2);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 2);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 2);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 3);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 3);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 3);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 3);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 3);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 3);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 3);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 3);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 3);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 3);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 4);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 4);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 4);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 4);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 4);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 4);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 4);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 4);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 4);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 4);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 5);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 5);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 5);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 5);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 1, 'T' , 5);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 5);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 5);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 5);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 5);

INSERT INTO siedzenia (siedzenie\_id, nr\_rzedu, dostepnosc, sale\_sala\_id) VALUES (sek\_siedzenie\_id.nextval, 2, 'T' , 5);

## --Sale

CREATE TABLE sale (

sala\_id NUMBER(2) NOT NULL,

liczba\_foteli NUMBER(3) NOT NULL,

liczba\_rzędów NUMBER(2) NOT NULL,

kanapy CHAR(1) NOT NULL

);

ALTER TABLE sale ADD CONSTRAINT sale\_pk PRIMARY KEY ( sala\_id );

ALTER TABLE sale RENAME COLUMN liczba\_rzędów TO liczba\_rzedow;

INSERT INTO sale (sala\_id, liczba\_foteli, liczba\_rzedow, kanapy) VALUES (1, 10, 2, 'T');

INSERT INTO sale (sala\_id, liczba\_foteli, liczba\_rzedow, kanapy) VALUES (2, 10, 2, 'T');

INSERT INTO sale (sala\_id, liczba\_foteli, liczba\_rzedow, kanapy) VALUES (3, 10, 2, 'N');

INSERT INTO sale (sala\_id, liczba\_foteli, liczba\_rzedow, kanapy) VALUES (4, 10, 2, 'T');

INSERT INTO sale (sala\_id, liczba\_foteli, liczba\_rzedow, kanapy) VALUES (5, 10, 2, 'N');

## --Filmy

CREATE TABLE filmy (

film\_id NUMBER(5) NOT NULL,

tytul VARCHAR2(50 CHAR) NOT NULL,

data\_premiery DATE NOT NULL,

czas\_trwania NUMBER(3) NOT NULL,

reżyser VARCHAR2(50 CHAR) NOT NULL

);

ALTER TABLE filmy ADD CONSTRAINT filmy\_pk PRIMARY KEY ( film\_id );

ALTER TABLE filmy RENAME COLUMN reżyser TO rezyser;

INSERT INTO filmy (film\_id, tytul, data\_premiery, czas\_trwania, rezyser)

VALUES (1,'Shrek', TO\_DATE('2001-07-13'), 90, 'Vicky Jenson');

INSERT INTO filmy (film\_id, tytul, data\_premiery, czas\_trwania, rezyser)

VALUES (2,'Igrzyska smierci', '2012-02-23', 144, 'Gary Ross');

INSERT INTO filmy (film\_id, tytul, data\_premiery, czas\_trwania, rezyser)

VALUES (3,'Jestem legendą', '2008-01-11', 101, 'Francis Lawrence');

INSERT INTO filmy (film\_id, tytul, data\_premiery, czas\_trwania, rezyser)

VALUES (4,'Kot w butach', '2011-01-05', 97, 'Chris Miller');

INSERT INTO filmy (film\_id, tytul, data\_premiery, czas\_trwania, rezyser)

VALUES (5,'Avatar: Istota wody', '2022-12-16', 190, 'James Cameron');

## --Pracownicy

CREATE TABLE pracownicy (

pracownik\_id NUMBER(5) NOT NULL,

imie VARCHAR2(30 CHAR) NOT NULL,

nazwisko VARCHAR2(30 CHAR) NOT NULL,

adres\_email VARCHAR2(50 CHAR),

pensja NUMBER(5) NOT NULL

);

ALTER TABLE pracownicy ADD CONSTRAINT pracownicy\_pk PRIMARY KEY ( pracownik\_id );

INSERT INTO pracownicy (pracownik\_id, imie, nazwisko, adres\_email, pensja)

VALUES (1,'Aleksandra', 'Sobiło', '[aleksandra.sobilo@gmail.com](mailto:aleksandra.sobilo@gmail.com)', 2900);

INSERT INTO pracownicy (pracownik\_id, imie, nazwisko, adres\_email, pensja)

VALUES (2,'Karolina', 'Cudziło', '[karolina.cudzilo@gmail.com](mailto:karolina.cudzilo@gmail.com)', 2950);

INSERT INTO pracownicy (pracownik\_id, imie, nazwisko, adres\_email, pensja)

VALUES (3,'Michał', 'Biernacki', '[michal.biernacki@gmail.com](mailto:michal.biernacki@gmail.com)', 2850);

INSERT INTO pracownicy (pracownik\_id, imie, nazwisko, adres\_email, pensja)

VALUES (4,'Mateusz', 'Zygmunt', '[mateusz.zygmunt@gmail.com](mailto:mateusz.zygmunt@gmail.com)', 3100);

INSERT INTO pracownicy (pracownik\_id, imie, nazwisko, adres\_email, pensja)

VALUES (5,'Remigiusz', 'Pałka', '[remigiusz.palka@gmail.com](mailto:remigiusz.palka@gmail.com)', 3050);

## --Klienci

CREATE TABLE klienci (

klient\_id NUMBER(5) NOT NULL,

imie VARCHAR2(30 CHAR),

nazwisko VARCHAR2(30 CHAR),

adres\_email VARCHAR2(40 CHAR),

numer\_telefonu NUMBER(9)

);

ALTER TABLE klienci ADD CONSTRAINT klienci\_pk PRIMARY KEY ( klient\_id );

INSERT INTO klienci (klient\_id, imie, nazwisko, adres\_email, numer\_telefonu)

VALUES (1,'Oliwia', 'Kata', '[oliwia.kata@gmail.com](mailto:oliwia.kata@gmail.com)', 750923738);

INSERT INTO klienci (klient\_id, imie, nazwisko, adres\_email, numer\_telefonu)

VALUES (2,'Natalia', 'Stasiak', '[natalia.stasiak@gmail.com](mailto:natalia.stasiak@gmail.com)', 964852741);

INSERT INTO klienci (klient\_id, imie, nazwisko, adres\_email, numer\_telefonu)

VALUES (3,'Krzysztof', 'Siembida', '[krzysztof.siembida@gmail.com](mailto:krzysztof.siembida@gmail.com)', 142536479);

INSERT INTO klienci (klient\_id, imie, nazwisko, adres\_email, numer\_telefonu)

VALUES (4,'Magdalena', 'Krawiec', '[magdalena.krawiec@gmail.com](mailto:magdalena.krawiec@gmail.com)', 875647123);

INSERT INTO klienci (klient\_id, imie, nazwisko, adres\_email, numer\_telefonu)

VALUES (5,'Wiktor', 'Zbroński', '[wiktor.zbronski@gmail.com](mailto:wiktor.zbronski@gmail.com)', 987564257);

## --Promocje

CREATE TABLE promocje (

promocja\_id NUMBER(5) NOT NULL,

nazwa\_promocji VARCHAR2(30 CHAR) NOT NULL,

wartosc\_promocji NUMBER(3, 2) NOT NULL

);

ALTER TABLE promocje ADD CONSTRAINT promocje\_pk PRIMARY KEY ( promocja\_id );

INSERT INTO promocje (promocja\_id, nazwa\_promocji, wartosc\_promocji)

VALUES (1, 'ulgowy', 0.15);

INSERT INTO promocje (promocja\_id, nazwa\_promocji, wartosc\_promocji)

VALUES (2, 'grupowy', 0.3);

INSERT INTO promocje (promocja\_id, nazwa\_promocji, wartosc\_promocji)

VALUES (3, 'tani seans', 0.05);

INSERT INTO promocje (promocja\_id, nazwa\_promocji, wartosc\_promocji)

VALUES (4, 'duza rodzina', 0.2);

INSERT INTO promocje (promocja\_id, nazwa\_promocji, wartosc\_promocji)

VALUES (5, 'dla seniorow', 0.2);

## --Seanse

CREATE TABLE seanse (

seans\_id NUMBER(5) NOT NULL,

data\_godzina\_seansu DATE NOT NULL,

jezyk VARCHAR2(50 CHAR) NOT NULL,

napisy CHAR(1) NOT NULL,

seans3d CHAR(1) NOT NULL,

filmy\_film\_id NUMBER(5) NOT NULL,

sale\_sala\_id NUMBER(2) NOT NULL

);

ALTER TABLE seanse ADD CONSTRAINT seanse\_pk PRIMARY KEY ( seans\_id );

ALTER TABLE seanse

ADD CONSTRAINT seanse\_filmy\_fk FOREIGN KEY ( filmy\_film\_id )

REFERENCES filmy ( film\_id );

ALTER TABLE seanse

ADD CONSTRAINT seanse\_sale\_fk FOREIGN KEY ( sale\_sala\_id )

REFERENCES sale ( sala\_id );

--film1

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-01 17:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'polski', 'N', 'N', 1, 1);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-01 9:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'polski', 'N', 'N', 1, 1);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-02 17:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'polski', 'N', 'N', 1, 1);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-02 9:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'polski', 'N', 'N', 1, 1);

--film2

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-01 13:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'angielski', 'T', 'N', 2, 2);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-01 10:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'angielski', 'T', 'N', 2, 2);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-02 13:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'angielski', 'T', 'N', 2, 2);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-02 10:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'angielski', 'T', 'N', 2, 2);

--film3

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2020-01-02 14:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'angielski', 'T', 'N', 3, 3);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2020-01-02 14:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'polski', 'N', 'N', 3, 4);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2020-01-04 14:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'angielski', 'T', 'N', 3, 3);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2020-01-04 14:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'polski', 'N', 'N', 3, 4);

--film4

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-02 9:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'polski', 'N', 'T', 4, 4);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-01 9:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'polski', 'N', 'T', 4, 4);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-01 12:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'polski', 'N', 'T', 4, 4);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-01 19:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'polski', 'N', 'T', 4, 4);

--film5

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-01 13:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'angielski', 'T', 'T', 5, 5);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-01 18:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'angielski', 'T', 'T', 5, 5);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-02 13:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'angielski', 'T', 'T', 5, 5);

INSERT INTO seanse (seans\_id, data\_godzina\_seansu, jezyk, napisy, seans3d, filmy\_film\_id, sale\_sala\_id) VALUES (sek\_seans\_id.nextval, to\_date('2023-02-02 18:00:00', 'YYYY-MM-DD HH24:MI:SS') , 'angielski', 'T', 'T', 5, 5);

## --Zamowienia

CREATE TABLE zamowienia (

zamowienie\_id NUMBER(15) NOT NULL,

data\_zamowienia DATE NOT NULL,

czy\_internetowe CHAR(1) NOT NULL,

klienci\_klient\_id NUMBER(5) NOT NULL,

pracownicy\_pracownik\_id NUMBER(5),

seanse\_seans\_id NUMBER(5) NOT NULL

);

ALTER TABLE zamowienia ADD CONSTRAINT zamowienia\_pk PRIMARY KEY ( zamowienie\_id );

ALTER TABLE zamowienia

ADD CONSTRAINT zamowienia\_klienci\_fk FOREIGN KEY ( klienci\_klient\_id )

REFERENCES klienci ( klient\_id );

ALTER TABLE zamowienia

ADD CONSTRAINT zamowienia\_pracownicy\_fk FOREIGN KEY ( pracownicy\_pracownik\_id )

REFERENCES pracownicy ( pracownik\_id );

ALTER TABLE zamowienia

ADD CONSTRAINT zamowienia\_seanse\_fk FOREIGN KEY ( seanse\_seans\_id )

REFERENCES seanse ( seans\_id );

INSERT INTO zamowienia (zamowienie\_id, data\_zamowienia, czy\_internetowe, klienci\_klient\_id, pracownicy\_pracownik\_id, seanse\_seans\_id)

VALUES (sek\_zamowienie\_id.nextval, SYSDATE , 'N', 1,2,1); --klient 1 seans 1

INSERT INTO zamowienia (zamowienie\_id, data\_zamowienia, czy\_internetowe, klienci\_klient\_id, pracownicy\_pracownik\_id, seanse\_seans\_id)

VALUES (sek\_zamowienie\_id.nextval, SYSDATE, 'T', 1, NULL,12); --klient 1 seans 12

INSERT INTO zamowienia (zamowienie\_id, data\_zamowienia, czy\_internetowe, klienci\_klient\_id, pracownicy\_pracownik\_id, seanse\_seans\_id)

VALUES (sek\_zamowienie\_id.nextval, SYSDATE , 'T', 2,NULL,1); --klient 2 seans 1

INSERT INTO zamowienia (zamowienie\_id, data\_zamowienia, czy\_internetowe, klienci\_klient\_id, pracownicy\_pracownik\_id, seanse\_seans\_id)

VALUES (sek\_zamowienie\_id.nextval, SYSDATE , 'T', 3,NULL,1); --klient 3 seans 1

INSERT INTO zamowienia (zamowienie\_id, data\_zamowienia, czy\_internetowe, klienci\_klient\_id, pracownicy\_pracownik\_id, seanse\_seans\_id)

VALUES (sek\_zamowienie\_id.nextval, SYSDATE , 'T', 4,NULL,12); --klient 4 seans 12

INSERT INTO zamowienia (zamowienie\_id, data\_zamowienia, czy\_internetowe, klienci\_klient\_id, pracownicy\_pracownik\_id, seanse\_seans\_id)

VALUES (sek\_zamowienie\_id.nextval, SYSDATE , 'T', 5,NULL,11); --klient 5 seans 11

INSERT INTO zamowienia (zamowienie\_id, data\_zamowienia, czy\_internetowe, klienci\_klient\_id, pracownicy\_pracownik\_id, seanse\_seans\_id)

VALUES (sek\_zamowienie\_id.nextval, SYSDATE , 'T', 2,NULL,16); --klient 2 seans 16

INSERT INTO zamowienia (zamowienie\_id, data\_zamowienia, czy\_internetowe, klienci\_klient\_id, pracownicy\_pracownik\_id, seanse\_seans\_id)

VALUES (sek\_zamowienie\_id.nextval, SYSDATE , 'T', 3,NULL,20); --klient 3 seans 20

INSERT INTO zamowienia (zamowienie\_id, data\_zamowienia, czy\_internetowe, klienci\_klient\_id, pracownicy\_pracownik\_id, seanse\_seans\_id)

VALUES (sek\_zamowienie\_id.nextval, SYSDATE , 'T', 5,NULL,10); --klient 5 seans 10

## --Bilety

CREATE TABLE bilety (

bilet\_id NUMBER(30) NOT NULL,

cena\_biletu NUMBER(5, 2) NOT NULL,

siedzenia\_siedzenie\_id NUMBER(6) NOT NULL,

zamowienia\_zamowienie\_id NUMBER(15) NOT NULL

);

CREATE UNIQUE INDEX bilety\_\_idx ON

bilety (

siedzenia\_siedzenie\_id

ASC );

ALTER TABLE bilety ADD CONSTRAINT bilety\_pk PRIMARY KEY ( bilet\_id );

ALTER TABLE bilety

ADD CONSTRAINT bilety\_siedzenia\_fk FOREIGN KEY ( siedzenia\_siedzenie\_id )

REFERENCES siedzenia ( siedzenie\_id );

ALTER TABLE bilety

ADD CONSTRAINT bilety\_zamowienia\_fk FOREIGN KEY ( zamowienia\_zamowienie\_id )

REFERENCES zamowienia ( zamowienie\_id );

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (1, 30, 1, 1);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (2, 30, 2, 1);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (3, 30, 3, 2);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (4, 30, 4, 3);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (5, 30, 5, 3);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (6, 30, 3, 4);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (7, 30, 20, 5);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (8, 30, 19, 5);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (9, 30, 18, 6);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (9, 30, 3, 7);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (10, 30, 11, 8);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (11, 30, 12, 8);

INSERT INTO bilety (bilet\_id, cena\_biletu, siedzenia\_siedzenie\_id, zamowienia\_zamowienie\_id) VALUES (12, 30, 13, 8);

## --Promocja\_bilet

CREATE TABLE promocja\_bilet (

promocje\_promocja\_id NUMBER(5) NOT NULL,

bilety\_bilet\_id NUMBER(30) NOT NULL

);

ALTER TABLE promocja\_bilet ADD CONSTRAINT promocja\_bilet\_pk PRIMARY KEY ( promocje\_promocja\_id, bilety\_bilet\_id );

ALTER TABLE promocja\_bilet

ADD CONSTRAINT promocja\_bilet\_bilety\_fk FOREIGN KEY ( bilety\_bilet\_id )

REFERENCES bilety ( bilet\_id );

ALTER TABLE promocja\_bilet

ADD CONSTRAINT promocja\_bilet\_promocje\_fk FOREIGN KEY ( promocje\_promocja\_id )

REFERENCES promocje ( promocja\_id );

INSERT INTO promocja\_bilet (promocje\_promocja\_id, bilety\_bilet\_id) VALUES (1 ,1 );

INSERT INTO promocja\_bilet (promocje\_promocja\_id, bilety\_bilet\_id) VALUES (4 ,1 );

INSERT INTO promocja\_bilet (promocje\_promocja\_id, bilety\_bilet\_id) VALUES (4 , 2);

INSERT INTO promocja\_bilet (promocje\_promocja\_id, bilety\_bilet\_id) VALUES (3 , 3);

INSERT INTO promocja\_bilet (promocje\_promocja\_id, bilety\_bilet\_id) VALUES ( 3, 4);

INSERT INTO promocja\_bilet (promocje\_promocja\_id, bilety\_bilet\_id) VALUES ( 3, 5);

----

## --Platnosci

CREATE TABLE platnosci (

data\_platnosci DATE NOT NULL,

numert\_rachunku NUMBER(26),

platnosc\_id NUMBER(16) NOT NULL,

zamowienia\_zamowienie\_id NUMBER(15) NOT NULL

);

CREATE UNIQUE INDEX płatnosci\_\_idx ON

platnosci (

zamowienia\_zamowienie\_id

ASC );

ALTER TABLE platnosci RENAME COLUMN numert\_rachunku TO numer\_rachunku;

ALTER TABLE platnosci ADD CONSTRAINT płatnosci\_pk PRIMARY KEY ( platnosc\_id );

ALTER TABLE platnosci

ADD CONSTRAINT platnosci\_zamowienia\_fk FOREIGN KEY ( zamowienia\_zamowienie\_id )

REFERENCES zamowienia ( zamowienie\_id );

INSERT INTO platnosci(data\_platnosci, numer\_rachunku, platnosc\_id, zamowienia\_zamowienie\_id) VALUES (SYSDATE, NULL, 1, 1);

INSERT INTO platnosci VALUES (SYSDATE, 55417055137136742260434241, 2, 2);

INSERT INTO platnosci VALUES (SYSDATE, 18753118497323209949572453, 3, 3);

INSERT INTO platnosci VALUES (SYSDATE, 61717901745563054295678312, 4, 4);

INSERT INTO platnosci VALUES (SYSDATE, 83661068149521355453357385, 5, 5);

INSERT INTO platnosci VALUES (SYSDATE, 81899313124691774713708765, 6, 6);

INSERT INTO platnosci VALUES (SYSDATE, 86599718383997704178132876, 7, 7);

INSERT INTO platnosci VALUES (SYSDATE, 86599718383997704178132876, 8, 8);

## --Oceny\_filmow

CREATE TABLE oceny\_filmow (

ocena\_id NUMBER(5) NOT NULL,

wartosc\_oceny NUMBER(1) NOT NULL,

komentarz VARCHAR2(250 CHAR),

data\_oceny DATE NOT NULL,

klienci\_klient\_id NUMBER(5) NOT NULL,

filmy\_film\_id NUMBER(5) NOT NULL

);

ALTER TABLE oceny\_filmow ADD CONSTRAINT oceny\_filmu\_pk PRIMARY KEY ( ocena\_id );

ALTER TABLE oceny\_filmow

ADD CONSTRAINT oceny\_filmow\_filmy\_fk FOREIGN KEY ( filmy\_film\_id )

REFERENCES filmy ( film\_id );

ALTER TABLE oceny\_filmow

ADD CONSTRAINT oceny\_filmow\_klienci\_fk FOREIGN KEY ( klienci\_klient\_id )

REFERENCES klienci ( klient\_id );

INSERT INTO oceny\_filmow (ocena\_id, wartosc\_oceny, komentarz, data\_oceny, klienci\_klient\_id, filmy\_film\_id) VALUES (1, 3, 'bardzo slaby film',to\_date( '2022-12-12 09:23:12', 'YYYY-MM-DD HH24:MI:SS'), 1, 3);

INSERT INTO oceny\_filmow (ocena\_id, wartosc\_oceny, komentarz, data\_oceny, klienci\_klient\_id, filmy\_film\_id) VALUES (2, 5, 'SUPER film', to\_date( '2022-12-12 11:34:12', 'YYYY-MM-DD HH24:MI:SS'), 1, 2);

INSERT INTO oceny\_filmow (ocena\_id, wartosc\_oceny, komentarz, data\_oceny, klienci\_klient\_id, filmy\_film\_id) VALUES (3, 3, 'slaby film', to\_date( '2022-12-15 09:23:12', 'YYYY-MM-DD HH24:MI:SS'), 2, 3);

INSERT INTO oceny\_filmow (ocena\_id, wartosc\_oceny, komentarz, data\_oceny, klienci\_klient\_id, filmy\_film\_id) VALUES (4, 4, 'polecam', to\_date( '2022-12-15 10:33:02', 'YYYY-MM-DD HH24:MI:SS'), 2, 4);

INSERT INTO oceny\_filmow (ocena\_id, wartosc\_oceny, komentarz, data\_oceny, klienci\_klient\_id, filmy\_film\_id) VALUES (5, 2, 'nie polecam', to\_date( '2022-12-16 07:23:12', 'YYYY-MM-DD HH24:MI:SS'), 2, 1);

INSERT INTO oceny\_filmow (ocena\_id, wartosc\_oceny, komentarz, data\_oceny, klienci\_klient\_id, filmy\_film\_id) VALUES (6, 1, NULL, to\_date( '2023-01-02 09:43:52', 'YYYY-MM-DD HH24:MI:SS'), 2, 1);

INSERT INTO oceny\_filmow (ocena\_id, wartosc\_oceny, komentarz, data\_oceny, klienci\_klient\_id, filmy\_film\_id) VALUES (7, 5, 'swietny film', to\_date( '2023-01-03 07:23:12', 'YYYY-MM-DD HH24:MI:SS'), 3, 1);

INSERT INTO oceny\_filmow (ocena\_id, wartosc\_oceny, komentarz, data\_oceny, klienci\_klient\_id, filmy\_film\_id) VALUES (8, 3, 'ok ', to\_date( '2023-01-07 07:23:12', 'YYYY-MM-DD HH24:MI:SS'), 5, 5);

INSERT INTO oceny\_filmow (ocena\_id, wartosc\_oceny, komentarz, data\_oceny, klienci\_klient\_id, filmy\_film\_id) VALUES (9, 5, 'Rewelacyjny film. Warto obejrzec', to\_date( '2023-01-10 12:23:22', 'YYYY-MM-DD HH24:MI:SS'), 3, 4);

INSERT INTO oceny\_filmow (ocena\_id, wartosc\_oceny, komentarz, data\_oceny, klienci\_klient\_id, filmy\_film\_id) VALUES (10, 2, 'nudny film', to\_date('2023-01-22 12:23:22', 'YYYY-MM-DD HH24:MI:SS'), 1, 5);

# Przypadki użycia bazy danych

Opis przypadków użycia bazy danych wraz z poleceniami SELECT.

1. Ile sprzedano biletów na każdy z filmów?

SELECT filmy.tytul, COUNT(bilety.bilet\_id) as "Sprzedane bilety" FROM filmy

RIGHT OUTER JOIN seanse ON seanse.filmy\_film\_id = filmy.film\_id

FULL OUTER JOIN zamowienia ON zamowienia.seanse\_seans\_id = seanse.seans\_id

FULL OUTER JOIN bilety ON bilety.zamowienia\_zamowienie\_id = zamowienia.zamowienie\_id

GROUP BY filmy.tytul

ORDER BY COUNT(bilety.bilet\_id) DESC;

1. Ile zarobiło kino na sprzedaży biletów w poszczególnych dniach. Wyświetl datę seansu iw wartość przychody kina tak aby data była w typie char a wartość przychodu była zaokrąglona do 2 miejsc po przecinku.

SELECT TO\_CHAR(S.DATA\_GODZINA\_SEANSU, 'YY/MM/DD') AS "Data seansu", ROUND(SUM((1-NVL(P.WARTOSC\_PROMOCJI,0))\*b.CENA\_BILETU),2) AS "Zysk" from bilety b

INNER JOIN zamowienia z ON z.zamowienie\_id = b.ZAMOWIENIA\_ZAMOWIENIE\_ID

INNER JOIN seanse s ON s.SEANS\_ID = z.SEANSE\_SEANS\_ID

FULL OUTER JOIN PROMOCJA\_BILET pb ON pb.BILETY\_BILET\_ID=b.bilet\_id

LEFT OUTER JOIN promocje p ON p.promocja\_id = pb.PROMOCJE\_PROMOCJA\_ID

GROUP BY S.DATA\_GODZINA\_SEANSU;

1. Którzy klienci byli w kinie przynajmniej 2 razy?

SELECT CONCAT(k1.IMIE||' ' , k1.NAZWISKO) "Klient", COUNT(z.zamowienie\_id) "Liczba filmów" FROM klienci k1  
JOIN zamowienia z ON z.klienci\_klient\_id = k1.klient\_id  
GROUP BY k1.klient\_id, CONCAT(k1.IMIE||' ' , k1.NAZWISKO)  
HAVING COUNT(z.zamowienie\_id) >= 2;

1. Wyświetl imiona, nazwiska oraz adresu mailowe klientów, którzy zapłacili z tego samego rachunku bankowego. Dodatkowo wyświetl tytuły filmów na które dokonali zakupu.

SELECT CONCAT(k1.IMIE||' ' , k1.NAZWISKO) AS "Imie i nazwisko klieanta", k1.ADRES\_EMAIL AS "Adres mailowy", k1.tytul FROM (SELECT \* FROM klienci k

FULL OUTER JOIN zamowienia z ON z.KLIENCI\_KLIENT\_ID=k.KLIENT\_ID

FULL OUTER JOIN platnosci p ON p.ZAMOWIENIA\_ZAMOWIENIE\_ID=z.ZAMOWIENIE\_ID

LEFT OUTER JOIN seanse s ON z.SEANSE\_SEANS\_ID = s.SEANS\_ID

FULL OUTER JOIN filmy f ON f.film\_id = s.FILMY\_FILM\_ID

WHERE p.NUMER\_RACHUNKU IN (SELECT NVL(p2.NUMER\_RACHUNKU, 0) FROM platnosci p2 GROUP BY p2.NUMER\_RACHUNKU HAVING COUNT(\*)>=2)) k1

ORDER BY ADRES\_EMAIL;

1. Planowane są podwyżki dla pracowników. Wyświetl nazwiska z inicjałem imienia (Nazwisko I.) wraz z rozważaną nową kwotą wynagrodzenia 20% większą od obecnej.

SELECT nazwisko || ' ' || SUBSTR(imie, 1, 1) || '.' AS pracownik, 1.2\*pensja AS nowa\_pensja FROM pracownicy;

1. Wyświetl najniższą wartość oceny w każdym roku, dla każdej takiej oceny wyświetl komentarz. Jeśli komentarz jest pusty, wyświetl “Brak komentarza”

SELECT EXTRACT(YEAR FROM o.data\_oceny) AS rok, NVL(o.komentarz, 'Brak komentarza') AS tresc\_komentarza, o.wartosc\_oceny AS najnizsza\_ocena FROM oceny\_filmow o

LEFT JOIN (SELECT EXTRACT(YEAR FROM k.data\_oceny) AS rok, MIN(k.wartosc\_oceny) AS min\_ocena FROM oceny\_filmow k

GROUP BY EXTRACT(YEAR FROM data\_oceny)) d ON (((EXTRACT(YEAR FROM o.data\_oceny))= d.rok))

WHERE o.wartosc\_oceny = d.min\_ocena

ORDER BY EXTRACT(YEAR FROM o.data\_oceny);

1. Wyświetl średnią ocenę dla każdej z promocji

SELECT nazwa\_promocji, ROUND(AVG(o.wartosc\_oceny),2) AS srednia\_ocena FROM promocje

JOIN promocja\_bilet ON (promocje\_promocja\_id = promocja\_id)

JOIN bilety ON (bilety\_bilet\_id = bilet\_id)

JOIN zamowienia z ON (z.zamowienie\_id=zamowienia\_zamowienie\_id)

JOIN klienci k ON (z.klienci\_klient\_id = k.klient\_id)

JOIN oceny\_filmow o ON (o.klienci\_klient\_id=k.klient\_id)

GROUP BY nazwa\_promocji;

1. Wyświetl id, nazwisko, pensje pracowników, którzy obsłużyli jakieś zamówienie. Dodatkowo wyświetl id obsłużonego zamówienia

SELECT p.pracownik\_id, p.nazwisko, p.pensja, (SELECT zamowienie\_id FROM zamowienia z WHERE z.pracownicy\_pracownik\_id=p.pracownik\_id) AS zamowienie\_id

FROM pracownicy p

WHERE p.pracownik\_id IN

(SELECT z1.pracownicy\_pracownik\_id FROM pracownicy p1

INNER JOIN zamowienia z1 ON z1.pracownicy\_pracownik\_id=p1.pracownik\_id

GROUP BY z1.pracownicy\_pracownik\_id

HAVING COUNT(\*)=1);

1. Którzy klienci obejrzeli jakiś film więcej niż jeden raz i jaki był to film?

SELECT k.nazwisko, f.tytul, COUNT(s.filmy\_film\_id) AS ile\_razy FROM zamowienia z

JOIN seanse s ON z.seanse\_seans\_id = s.seans\_id

JOIN klienci k ON z.klienci\_klient\_id = k.klient\_id

JOIN filmy f ON s.filmy\_film\_id = f.film\_id

GROUP BY k.nazwisko, f.tytul

HAVING COUNT(s.filmy\_film\_id)>1;

# Pozostałe obiekty bazy danych

## --Sekwencer

CREATE SEQUENCE sek\_zamowienie\_id

START WITH 1

INCREMENT BY 1

MAXVALUE 999999;

CREATE SEQUENCE sek\_siedzenie\_id

START WITH 1

INCREMENT BY 1

MAXVALUE 99999;

CREATE SEQUENCE sek\_seans\_id

START WITH 1

INCREMENT BY 1

MAXVALUE 99999;

## --Perspektywy

CREATE OR REPLACE VIEW sprzedane\_bilety AS ( SELECT filmy.tytul, COUNT(bilety.bilet\_id) as "Sprzedane bilety" FROM filmy

RIGHT OUTER JOIN seanse ON seanse.filmy\_film\_id = filmy.film\_id

FULL OUTER JOIN zamowienia ON zamowienia.seanse\_seans\_id = seanse.seans\_id

FULL OUTER JOIN bilety ON bilety.zamowienia\_zamowienie\_id = zamowienia.zamowienie\_id

GROUP BY filmy.tytul

);

CREATE OR REPLACE VIEW przychod\_wg\_dni AS (SELECT TO\_CHAR(S.DATA\_GODZINA\_SEANSU, 'YY/MM/DD') AS "Data seansu", ROUND(SUM((1-NVL(P.WARTOSC\_PROMOCJI,0))\*b.CENA\_BILETU),2) AS "Zysk" from bilety b

INNER JOIN zamowienia z ON z.zamowienie\_id = b.ZAMOWIENIA\_ZAMOWIENIE\_ID

INNER JOIN seanse s ON s.SEANS\_ID = z.SEANSE\_SEANS\_ID

FULL OUTER JOIN PROMOCJA\_BILET pb ON pb.BILETY\_BILET\_ID=b.bilet\_id

LEFT OUTER JOIN promocje p ON p.promocja\_id = pb.PROMOCJE\_PROMOCJA\_ID

GROUP BY S.DATA\_GODZINA\_SEANSU );

CREATE OR REPLACE VIEW stali\_klienci AS ( SELECT CONCAT(k1.IMIE||' ' , k1.NAZWISKO) "Klient", COUNT(z.zamowienie\_id) "Liczba filmów" FROM klienci k1  
JOIN zamowienia z ON z.klienci\_klient\_id = k1.klient\_id  
GROUP BY k1.klient\_id, CONCAT(k1.IMIE||' ' , k1.NAZWISKO)  
HAVING COUNT(z.zamowienie\_id) >= 2);

CREATE OR REPLACE VIEW powiazani\_klienci AS (SELECT CONCAT(k1.IMIE||' ' , k1.NAZWISKO) AS "Imie i nazwisko klieanta", k1.ADRES\_EMAIL AS "Adres mailowy", k1.tytul FROM (SELECT \* FROM klienci k

FULL OUTER JOIN zamowienia z ON z.KLIENCI\_KLIENT\_ID=k.KLIENT\_ID

FULL OUTER JOIN platnosci p ON p.ZAMOWIENIA\_ZAMOWIENIE\_ID=z.ZAMOWIENIE\_ID

LEFT OUTER JOIN seanse s ON z.SEANSE\_SEANS\_ID = s.SEANS\_ID

FULL OUTER JOIN filmy f ON f.film\_id = s.FILMY\_FILM\_ID

WHERE p.NUMER\_RACHUNKU IN (SELECT NVL(p2.NUMER\_RACHUNKU, 0) FROM platnosci p2 GROUP BY p2.NUMER\_RACHUNKU HAVING COUNT(\*)>=2)) k1

);

CREATE OR REPLACE VIEW proponowane\_podwyzki AS (SELECT nazwisko || ' ' || SUBSTR(imie, 1, 1) || '.' AS pracownik, 1.2\*pensja AS nowa\_pensja FROM pracownicy);

CREATE OR REPLACE VIEW min\_ocena\_wg\_roku AS (SELECT EXTRACT(YEAR FROM o.data\_oceny) AS rok, NVL(o.komentarz, 'Brak komentarza') AS tresc\_komentarza, o.wartosc\_oceny AS najnizsza\_ocena FROM oceny\_filmow o

LEFT JOIN (SELECT EXTRACT(YEAR FROM k.data\_oceny) AS rok, MIN(k.wartosc\_oceny) AS min\_ocena FROM oceny\_filmow k

GROUP BY EXTRACT(YEAR FROM data\_oceny)) d ON (((EXTRACT(YEAR FROM o.data\_oceny))= d.rok))

WHERE o.wartosc\_oceny = d.min\_ocena);

CREATE OR REPLACE VIEW srednia\_ocena\_wg\_promocji AS (SELECT nazwa\_promocji, ROUND(AVG(o.wartosc\_oceny),2) AS srednia\_ocena FROM promocje

JOIN promocja\_bilet ON (promocje\_promocja\_id = promocja\_id)

JOIN bilety ON (bilety\_bilet\_id = bilet\_id)

JOIN zamowienia z ON (z.zamowienie\_id=zamowienia\_zamowienie\_id)

JOIN klienci k ON (z.klienci\_klient\_id = k.klient\_id)

JOIN oceny\_filmow o ON (o.klienci\_klient\_id=k.klient\_id)

GROUP BY nazwa\_promocji);

CREATE OR REPLACE VIEW pracownicy\_na\_kasie AS ( SELECT p.pracownik\_id, p.nazwisko, p.pensja, (SELECT zamowienie\_id FROM zamowienia z WHERE z.PRACOWNICY\_PRACOWNIK\_ID=p.pracownik\_id) AS zamowienie\_id

FROM pracownicy p

WHERE P.PRACOWNIK\_ID IN

(SELECT z1.PRACOWNICY\_PRACOWNIK\_ID FROM pracownicy p1

INNER JOIN zamowienia z1 ON z1.PRACOWNICY\_PRACOWNIK\_ID=p1.pracownik\_id

GROUP BY z1.PRACOWNICY\_PRACOWNIK\_ID

HAVING COUNT(\*)=1));

CREATE OR REPLACE VIEW filmy\_warte\_powtorki AS ( SELECT k.nazwisko, f.tytul, COUNT(s.filmy\_film\_id) AS ile\_razy FROM zamowienia z

JOIN seanse s ON z.seanse\_seans\_id = s.seans\_id

JOIN klienci k ON z.klienci\_klient\_id = k.klient\_id

JOIN filmy f ON s.filmy\_film\_id = f.film\_id

GROUP BY k.nazwisko, f.tytul

HAVING COUNT(s.filmy\_film\_id)>1);