

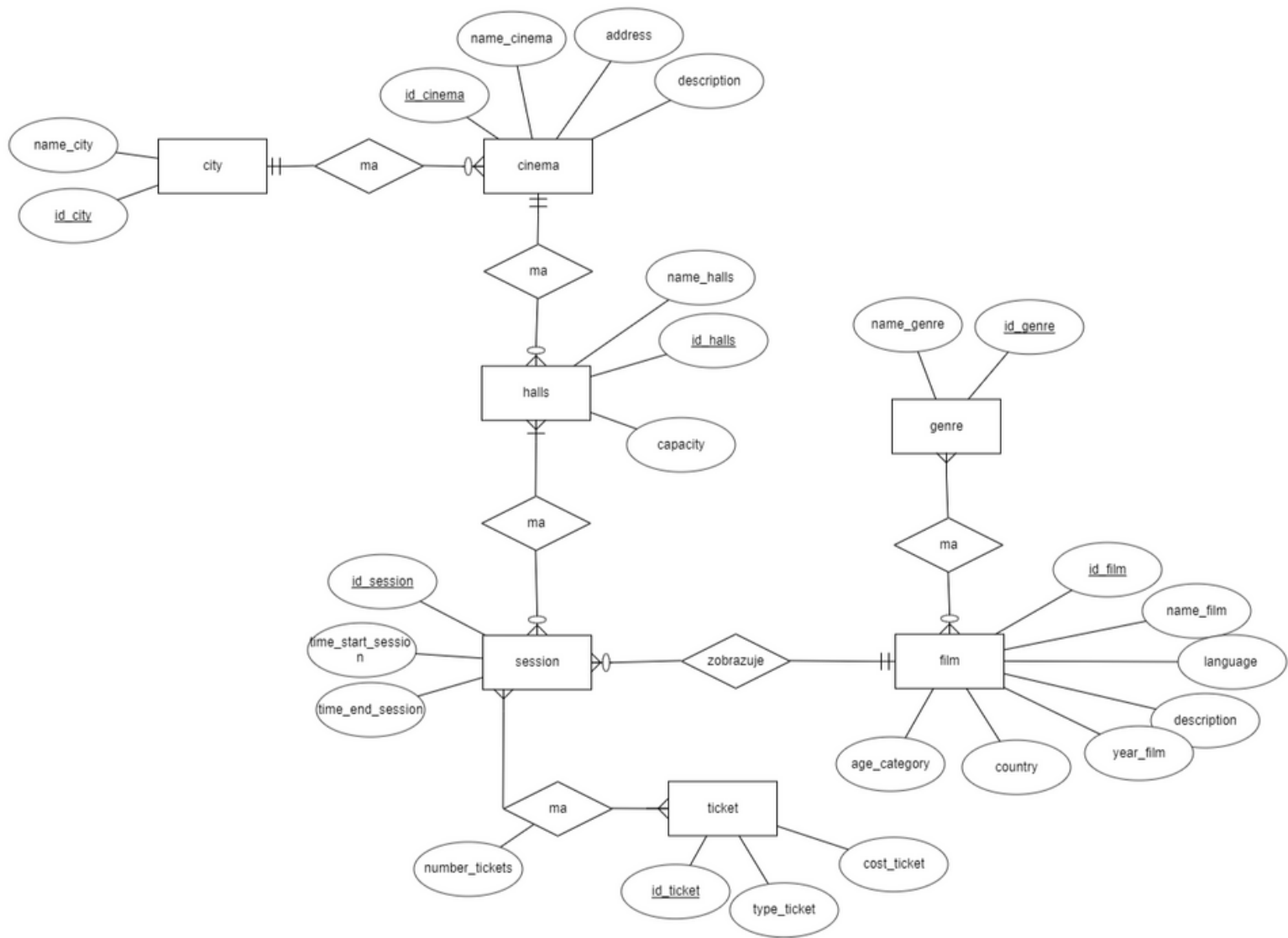
TECHNICKÁ UNIVERZITA KOŠICE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

# DATABÁZA

## Kino

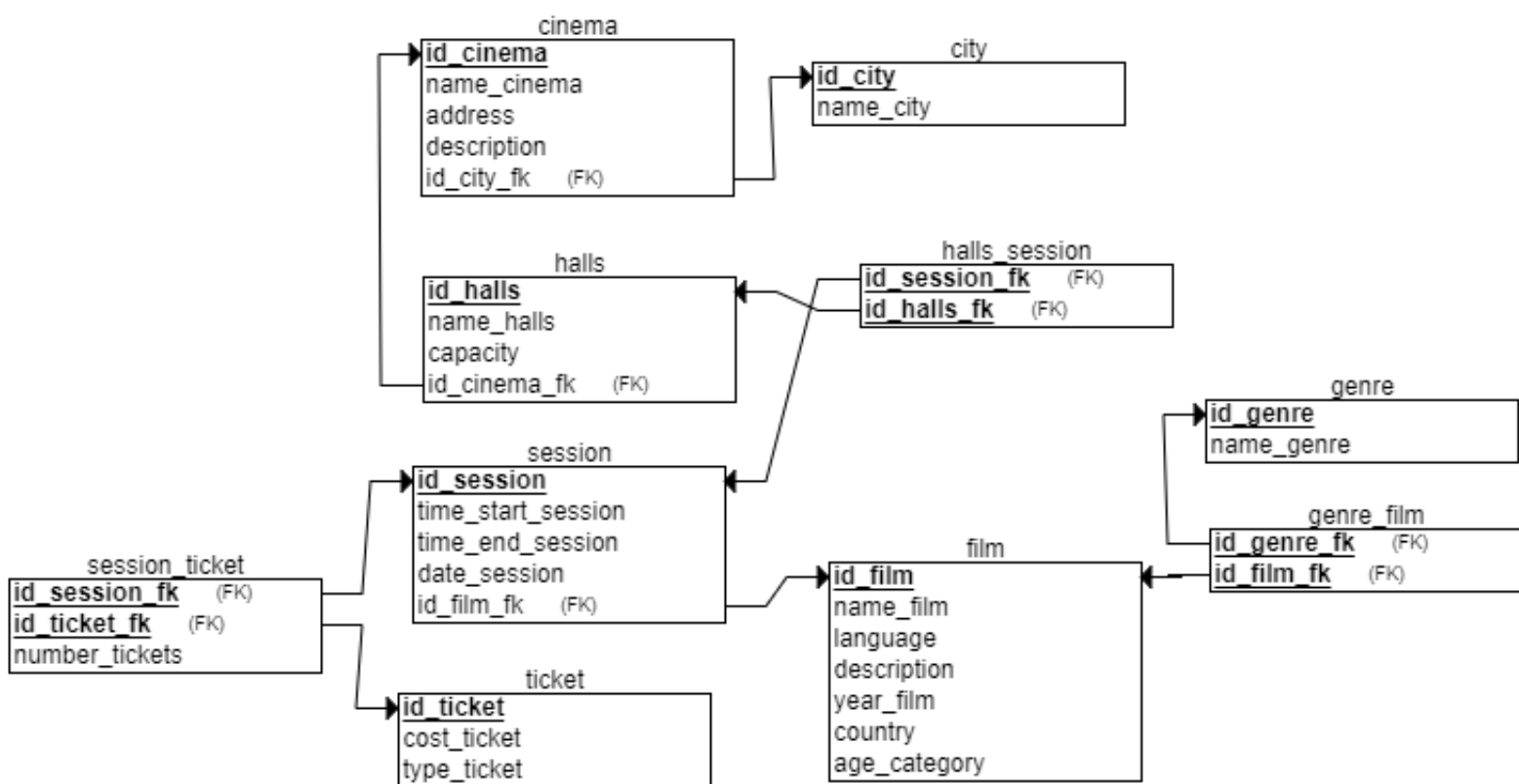
Erik Chytil

# Entitno-relačný model



Entities	Link type	Link content
city-cinema	1-M	jedno miesto môže mať veľa kín
halls-cinema	M-1	jedno kino môže mať veľa halls
halls-session	M-M	veľa halls môže mať veľa sessions
session-film	M-1	jeden film môže byť na viacerých sessions
film-gener	M-M	veľa filmov môže mať viacero žánrov
session-ticket	M-M	veľa session môže mať veľa tiketov

# Logický relačný model



# 1 časť

--Ako prvú si vytvorím tabuľku s názvom city a nastavím id\_city ako unique

```
create table city
```

```
(  
    id_city    int not null unique primary key,  
    name_city  varchar(50) not null
```

```
);
```

--Vytvorím tabuľku cinema vložím do nej hodnoty pre danú tabuľku, nastavím hodnotu povolenej dĺžky slova pre daný údaj

```
create table cinema
```

```
(  
    id_cinema  int not null unique primary key,  
    name_cinema varchar(100) not null,  
    address    varchar(100) not null,  
    description varchar(255) not null,  
    id_city_fk int not null,  
    foreign key (id_city_fk) references city(id_city)
```

```
);
```

--Vytváram halls, odkazujem sa na primárny kľúč v predchádzajúcej tabuľke cinema

```
create table halls
```

```
(  
    id_halls    int not null unique primary key,  
    name_halls  varchar(50) not null,  
    capacity    int not null,  
    id_cinema_fk int not null,  
    foreign key (id_cinema_fk) references cinema(id_cinema)
```

```
);
```

--Vytváram tabuľku film, zadávam hodnoty, ktoré sa v nej majú nachádzať

```
create table film
```

```
(  
    id_film      int not null unique primary key,  
    name_film    varchar(100) not null,  
    language     varchar(50) not null,  
    description   varchar(255) not null,  
    year_film    int not null,  
    country      varchar(100) not null,  
    age_category int not null
```

```
);
```

--vytvorenie žánrov

```
create table genre
```

```
(  
    id_genre    int not null unique primary key,  
    name_genre   varchar(50) unique not null
```

```
);
```

--Vytvorenie tabuľky genre\_film a vytvorenie odkazu na fk v tabuľke genre a film

```
create table genre_film
```

```
(  
    id_genre_fk int not null,  
    foreign key (id_genre_fk) references genre(id_genre),
```

```
;
```

```
    id_film_fk int not null,  
    foreign key (id_film_fk) references film(id_film)
```

```
);
```

--vytvorenie tabuľky session, každá session musí niekedy začínať a končiť a mať svoj dátum

```
create table session
```

```
(  
    id_session      int not null unique primary key,  
    time_start_session time not null,  
    time_end_session time not null,  
    date_session    date not null,
```

```
;
```

```
    id_film_fk int not null,  
    foreign key (id_film_fk) references film(id_film)
```

```
);
```

--Vytvorenie halls\_session pridanie fk k tabuľke session a halls

```
create table halls_session
```

```
(  
    id_session_fk int not null,  
    foreign key (id_session_fk) references session(id_session),
```

```
;
```

```
    id_halls_fk int not null,  
    foreign key (id_halls_fk) references halls(id_halls)
```

```
);
```

# 1 časť

--Tvorím tabuľku ticket, kde pridám 3 parametre id,cena,typ ticketu

create table ticket

```
(
  id_ticket      int not null unique primary key,  -- 1
  cost_ticket    float not null,                  -- 5.50
  type_ticket     varchar(20) not null             --'vip ticket'
```

);

--vytvorenie session\_ticket tvorím odkaz na predošlú tabuľku session a ticket

create table session\_ticket

```
(
  id_session_fk  int not null,
  foreign key (id_session_fk) references session(id_session),

  id_ticket_fk   int not null,
  foreign key (id_ticket_fk) references ticket(id_ticket),
```

```

  number_tickets int not null
);
```

--vkladám hodnoty do genre

insert into genre (id\_genre, name\_genre)

```
values (1, 'Actions'),
       (2, 'Adventure'),
       (3, 'Animated'),
       (4, 'Comedy'),
       (5, 'Drama'),
       (6, 'Fantasy'),
       (7, 'Horror');
```

--vkladám hodnoty do id\_city a name\_city

insert into city (id\_city, name\_city)

```
values (1, 'Bratislava'),
       (2, 'Košice'),
       (3, 'Poprad'),
       (4, 'Michalovce'),
       (5, 'Banská Bystrica'),
       (6, 'Nitra'),
       (7, 'Trnava');
```

--Vkladanie údajov do tabuľky cinema pre daný column

insert into cinema (id\_cinema, name\_cinema, address, description, id\_city\_fk)

```
values (1, 'CINEMAX 2', 'Moldavská cesta 32', 'Naša sieť kín CINEMAX vznikla v roku 2006', 2),
       (2, '5D', 'Moldavská cesta 32', 'Najlepse kino v meste', 2),
       (3, 'Poprad Cinema', 'Poštová 2', 'kino v obchodnom centre MAX', 3),
       (4, 'Michalovce', 'Komenskeho 56', 'Velmi drahe kino', 4),
       (5, 'Multiplex', 'Jesenského 4', 'Nice fine kino', 1);
```

--vkladanie do halls pre daný column

insert into halls(id\_halls, name\_halls, capacity, id\_cinema\_fk)

```
values (1, 'hall 1', 150, 1),
       (2, 'hall 2', 100, 1),
       (3, 'hall 3', 90, 1),
       (4, 'hall water effect', 20, 2),
       (5, 'hall wind effect', 20, 2),
       (6, 'hall 1', 200, 3),
       (7, 'hall 2', 250, 3),
       (8, 'hall 3', 150, 3),
       (9, 'hall 4', 150, 3),
       (10, 'hall modern', 150, 4),
       (11, 'hall private', 250, 4),
       (12, 'hall velky', 300, 5),
       (13, 'hall luxury', 200, 5);
```

# 1 časť

--Vkladanie údajov do tabuľky film pre každý stĺpec

insert into film(id\_film, name\_film, language, description, year\_film, country, age\_category)

values (1,'The Dark Knight', 'En', 'When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.', 2008, 'United States', 14),

(2,'The Lord of the Rings: The Return of the King', 'En, Sk', 'Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring.', 2003, 'New Zealand, United States', 12),

(3,'The Godfather', 'En, Sk', 'The aging patriarch of an organized crime dynasty in postwar New York City transfers control of his clandestine empire to his reluctant youngest son.', 1972, 'United States', 16),

(4,'Inception', 'En, Sk', 'A thief who steals corporate secrets through the use of dream-sharing technology.', 2010, 'United States, Great Britain', 12),

(5,'The Matrix', 'En, Sk', 'When a beautiful stranger leads computer hacker Neo to a forbidding underworld', 1999, 'United States, Australia', 16),

(6,'The Green Mile', 'En', 'The lives of guards on Death Row are affected by one of their charges', 1999, 'United States', 14),

(7,'Star Wars', 'En, Sk', 'Luke Skywalker joins forces with a Jedi Knight, a cocky pilot', 1977, 'United States', 0),

(8,'Lucy', 'En', 'A woman, accidentally caught in a dark deal, turns the tables on her captors and transforms into a merciless warrior evolved beyond human logic.', 2014, 'United States', 18),

(9,'Interstellar', 'En, Sk', 'A team of explorers travel through a wormhole in space in an attempt to ensure humanity's survival.', 2014, 'United States, Great Britain', 12),

(10,'Home Invasion', 'En', 'Intruders shoot Chloe's friend and break into Chloe's home on an isolated island on a rainy night.', 2016, 'United States', 13);

--Vložíím údaje pre genre\* každý film má svoje jedinečné id a žáner

insert into genre\_film( id\_film\_fk, id\_genre\_fk)

values (1,1),

(1,2),

(2,6),

(2,3),

(3,5),

(4,1),

(4,6),

(5,1),

(5,6),

(6,5),

(7,1),

(7,2),

(7,6),

(8,1),

(8,5),

(9,6),

(9,2),

(10,2),

(10,7);

--Vložíím do tabuľky ticket hodnoty aký typ lístka, cena a jeho id

insert into ticket(id\_ticket, cost\_ticket, type\_ticket)

values (1, 2, 'normal'),

(2, 4.5, 'vip'),

(3, 8, 'luxury');

--Vkladám pre každú session časy kedy sa má daná session začať, končiť, konkrétne id filmu a dátum

insert into session(id\_session, time\_start\_session, time\_end\_session, id\_film\_fk, date\_session)

values (1, '09:30:00', '11:30:00', 1, '2020-04-05'),

(2, '09:30:00', '11:30:00', 2, '2020-04-05'),

(3, '19:00:00', '21:30:00', 3, '2020-04-05'),

(4, '18:30:00', '20:30:00', 4, '2020-04-05'),

(5, '09:30:00', '11:30:00', 5, '2020-04-06'),

(6, '11:30:00', '13:30:00', 6, '2020-04-06'),

(7, '17:30:00', '19:30:00', 1, '2020-04-06'),

(8, '10:00:00', '11:50:00', 10, '2020-04-07'),

(9, '15:20:00', '17:30:00', 4, '2020-04-07'),

(10, '10:00:00', '11:50:00', 6, '2020-04-08');

--Vkladám do session\_ticket údaje ako id\_session, id\_lístka, počet lístkov

insert into session\_ticket(id\_session\_fk, id\_ticket\_fk, number\_tickets)

values (1, 1, 100),

(1, 2, 20),

(2, 1, 100),

(2, 3, 5), (3, 1, 200),

(4, 2, 100),

(5, 1, 150),

(5, 2, 15),

(5, 3, 15),

(6, 1, 150), (7, 1, 100),

(7, 2, 10),

(8, 1, 130),

(9, 1, 100),

(10, 2, 20);

# 1 časť

```
--Vkladám údaje do stlpcov id session a id danej halls
insert into halls_session(id_session_fk, id_halls_fk)
values (1,1),
       (1,2),
       (2,6),
       (3,8),
       (4,12),
       (5,6),
       (6,7),
       (7,8),
       (8,9),
       (9,10),
       (4,11),
       (7,12),
       (9,13),
       (10,5);
```

# 2 časť

```
--ODOVZDAVKA 2
--2 pohľady s netriviálnym selektom nad jednou tabuľkou
--zobrazí všetky filmy krajina ktorých je United States
select f.name_film, f.description, concat(f.country, ' - ', f.language) as country_lang, f.age_category
from film f
where f.country like '%United States%'
order by age_category asc; --veková kategória
--príkaz asc zoradzuje všetky údaje od najmenšieho po najväčší concat pridáva reťazce dohromady
--zobrazí všetky halls podľa kapacity
select h.capacity, string_agg(h.name_halls, ' ') as halls --string_agg je agregáčna funkcia, spája riadok reťazca do jedného
from halls h
group by h.capacity
order by h.capacity desc; --desc mi tuna zoradi capacity od najväčšej po najmenšiu hodnotu

-- 3 pohľady so spájaním tabuliek (1x spojenie 3 tabuliek, 1x outer join, 1x spojenie aspoň 2 tabuliek);
--zobrazí všetky films, genre akých je action
select f.name_film, f.country, g.name_genre
from genre g
inner join genre_film gf on g.id_genre = gf.id_genre_fk --vyberá záznam, ktoré majú zhodné tabuľky v oboch hodnotách
inner join film f on f.id_film = gf.id_film_fk
where g.name_genre = 'Actions' --filtruje actions_genre
order by f.name_film asc;

--zobrazí všetky city a kino v týchto miestach
select c.name_city, coalesce(c2.name_cinema, '---') as name_cinema --vracia 1 nenulovú hodnotu v zozname
from city c
left outer join cinema c2 on c.id_city = c2.id_city_fk;

--načítava všetky riadky z ľavej tabuľky ak sa zhoduje z pravej tabuľky .Výsledok je null
-- ak neexistuje žiadny spoločný riadok kt. by sa uhoďoval medzi ľavou a pravou tabuľkou výsledok je NULL
--vypíše všetky session z kino Michalovce medzi dátami '2020-04-05' a '2020-04-07'
select c.name_cinema, concat(h.name_halls, ' - ', h.capacity) as halls_capacity,
       concat(s.time_start_session, ' - ', s.time_end_session) as time_session,
       f.name_film
from cinema c
left outer join halls h on c.id_cinema = h.id_cinema_fk
left outer join halls_session hs on h.id_halls = hs.id_halls_fk
left outer join session s on s.id_session = hs.id_session_fk
left outer join film f on f.id_film = s.id_film_fk
where name_cinema = 'Michalovce' and s.date_session
       between '2020-04-05' and '2020-04-07';

--2 pohľady s použitím agregáčnych funkcií alebo zoskupenia
--zobrazí počet miest na sedenie v kine CINEMAX 2
select c.name_cinema, sum(h.capacity)
from cinema c
inner join halls h on c.id_cinema = h.id_cinema_fk
where c.name_cinema = 'CINEMAX 2'
group by c.name_cinema;
```

# 2 časť

```
--Zobrazí stredný cenu ticketu za '2020-04-05';
select round( avg(t.cost_ticket)::numeric, 2) || ' €' as avg_cost
from ticket t
left join session_ticket st on t.id_ticket = st.id_ticket_fk
left join session s on s.id_session = st.id_session_fk
where s.date_session = '2020-04-05';
```

```
-- 1 pohľad s použitím množinových operácií
--zobrazí všetky animované filmy pre vekovú kategóriu do 14 rokov
-- a všetky ostatné žánre pre vekovú kategóriu < viac ako 14 rokov
select f.name_film, f.age_category, string_agg(g.name_genre, ', ')
from film f
join genre_film gf on f.id_film = gf.id_film_fk
join genre g on g.id_genre = gf.id_genre_fk
where f.age_category <14 and g.name_genre = 'Animated'
group by f.age_category, f.name_film
union
select f.name_film, f.age_category, string_agg(g.name_genre, ', ')
from film f
join genre_film gf on f.id_film = gf.id_film_fk
join genre g on g.id_genre = gf.id_genre_fk
where f.age_category >=14
group by f.age_category, f.name_film
order by age_category asc ;
```

--2 pohľady s použitím vnorených selektov;

```
--zobrazí kino podľa počtu session
select c.name_cinema
from cinema c
-- kontrolujem id_cinema z vnutorneho selektu
where id_cinema in (
    --vyberem len id_cinema z vnutorneho selektu
    select t.id_cinema_fk
    from (
        --vypise id_cinema vsetkych halls, ktory je v tejto cinema a sucet session v tejto cinema
        select h.id_cinema_fk, string_agg(h.name_halls, ', '), count(hs.id_session_fk) as countt
        from halls h
        join halls_session hs on h.id_halls = hs.id_halls_fk
        --zoskupujem podľa cinema
        group by h.id_cinema_fk
        --zoradujem sucet session od najväcsieho po najmensi
        order by countt desc
    ) as t
);
```

```
--zobrazí všetky filmy, a time session pre type ticket luxury
select concat(s.time_start_session, ' - ', s.time_end_session) as time_session, f.name_film
from session s
join film f on f.id_film = s.id_film_fk
where exists(
    select *
    from ticket t
    join session_ticket st on t.id_ticket = st.id_ticket_fk
    where t.type_ticket ='luxury' and s.id_session = st.id_session_fk
);
```

--vytvorte 1 sekvenciu na generovanie primárnych kľúčov a triggre, ktorá bude vkladať hodnoty do príslušných tabuliek;

--vytvorím trigger pre tabuľku genre v ,ktorej id\_genre bude samostané generované

create or replace function func\_generateKey() returns trigger as

\$\$

```
declare
    max_id int;
begin
    select max(id_genre) into max_id
    from genre;
    max_id = max_id +1;
    new.id_genre = max_id;
    return new;
end;
```

\$\$

language plpgsql;



# 2 časť

```
create trigger generateKey before insert on genre
for each row
execute procedure func_generateKey();
```

```
insert into genre (name_genre)
values ('robot'),
('animal');
```

```
select *
from genre ;
```

--vytvorte 1 ľubovoľný trigger okrem typu triggera uvedeného vyššie

--trigger pre zmenu cenu lístka, cena ticketu nemôže byť viac ako 15€ a v maine menšie ako 1€

```
create trigger udpCost after update on ticket
for each row
execute procedure func_udpCost();
```

create or replace function func\_udpCost() returns trigger as

```
$$
begin
    if(new.cost_ticket)<1 then
        raise exception 'cena ticketu musi byt vacsia alebo = ako 1 €';
    end if;
    if(new.cost_ticket)>15 then
        raise exception 'cena ticketu musi byt mensia alebo = ako 15 €';
    end if;
    update ticket set cost_ticket = new.cost_ticket where cost_ticket = old.cost_ticket;
    return new;
end;
$$
language plpgsql;
```

update ticket set cost\_ticket = 16.9 where type\_ticket = 'normal';

--zobrazí všetky filmy krajina ktorých je United States

```
select f.name_film, f.description, concat(f.country, ' - ', f.language) as country_lang, f.age_category
from film f
where f.country like 'United States%'
order by age_category asc; --veková kategória
```

--príkaz asc zoradzuje všetky údaje od najmenšieho po najväčší concat pridáva reťazce dohromady

name_film	description	country_lang
1 Star Wars	Luke Skywalker joins forces with a Jedi Knight...	United States - En, SK
2 The Lord of the Rings: The Return of the King	Gandalf and Aragorn lead the World of Men agai...	New Zealand, United States - En, SK
3 Inception	A thief who steals corporate secrets through t...	United States, Great Britain - En, SK
4 Interstellar	A team of explorers travel through a wormhole...	United States, Great Britain - En, SK
5 Home Invasion	Intruders shoot Chloe's friend and break into...	United States - En
6 The Green Mile	The lives of guards on Death Row are affected...	United States - En
7 The Dark Knight	When the menace known as the Joker wreaks havo...	United States - En
8 The Matrix	When a beautiful stranger leads computer hacke...	United States, Australia - En, SK
9 The Godfather	The aging patriarch of an organized crime dyna...	United States - En, SK
10 Lucy	A woman, accidentally caught in a dark deal, t...	United States - En

```
--zobrazí všetky halls podľa kapacity
select h.capacity, string_agg(h.name_halls, ', ') as halls --string_agg
from halls h
group by h.capacity
order by h.capacity desc; --desc mi tuna zoradi capacity od najväčšej po na
```

Output príkaz asc zoradzuje...áva reťazce dohromady

capacity	halls
1 300	hall velky
2 250	hall 2, hall private
3 200	hall 1, hall luxury
4 150	hall 1, hall 3, hall 4, hall modern
5 100	hall 2
6 90	hall 3
7 20	hall water effect, hall wind effect

```

243 -- 3 pohľady so spájaním tabuliek (1x spojenie 3 tabuliek, 1x outer join, 1x spojenie aspoň
244 --zobrazí všetky films, genre akých je action
245 ✓ select f.name_film, f.country, g.name_genre
246 from genre g
247 inner join genre_film gf 1<->1..n: on g.id_genre = gf.id_genre_fk --vyberá záznam, ktoré maj
248 inner join film f 1..n<->1: on f.id_film = gf.id_film_fk
249 where g.name_genre = 'Actions' --filtruje actions_genre
250 order by f.name_film asc;
251

```

Output 3 pohľady so spájaním...ie aspoň 2 tabuliek; X

5 rows

	name_film	country	name_genre
1	Inception	United States, Great Britain	Actions
2	Lucy	United States	Actions
3	Star Wars	United States	Actions
4	The Dark Knight	United States	Actions
5	The Matrix	United States, Australia	Actions

```

--zobrazí všetky city a kino v týchto miestach
✓ select c.name_city, coalesce(c2.name_cinema, '---') as name_cinema --
from city c
left outer join cinema c2 1<->0..n: on c.id_city = c2.id_city_fk;

```

Output Result 6 X

8 rows

	name_city	name_cinema
1	Bratislava	Multiplex
2	Košice	5D
3	Košice	CINEMAX 2
4	Poprad	Poprad Cinema
5	Michalovce	Michalovce
6	Banská Bystrica	---
7	Nitra	---
8	Trnava	---

```

--načítava všetky riadky z ľavej tabuľky ak sa zhoduje z pravej tabuľky .Výsledok je null
-- ak neexistuje žiadny spoločný riadok kt. by sa uhoďoval medzi ľavou a pravou tabuľkou výsledok je NULL
--vypíše všetky session z kina Michalovce medzi dátami '2020-04-05' a '2020-04-07'
select c.name_cinema, concat(h.name_halls, ' - ', h.capacity) as halls_capacity,
concat(s.time_start_session, ' - ', s.time_end_session) as time_session,
f.name_film
from cinema c
left outer join halls h 1<->0..n: on c.id_cinema = h.id_cinema_fk
left outer join halls_session hs 1<->0..n: on h.id_halls = hs.id_halls_fk
left outer join session s 1..n<->1: on s.id_session = hs.id_session_fk
left outer join film f 1..n<->1: on f.id_film = s.id_film_fk
where name_cinema = 'Michalovce' and s.date_session
between '2020-04-05' and '2020-04-07';

```

Output ak neexistuje žiadny...fkou výsledok je NULL X

2 rows

	name_cinema	halls_capacity	time_session	name_film
1	Michalovce	hall modern - 150	15:20:00 - 17:30:00	Inception
2	Michalovce	hall private - 250	18:30:00 - 20:30:00	Inception

```
--2 pohľady s použitím agregáčnych funkcií alebo zoskupenia
--zobrazí počet miest na sedenie v kine CINEMAX 2
select c.name_cinema, sum(h.capacity)
from cinema c
inner join halls h 1<->1..n: on c.id_cinema = h.id_cinema_fk
where c.name_cinema = 'CINEMAX 2'
group by c.name_cinema;
```

5

Output

2 pohľady s použitím...kcí alebo zoskupenia X

< >

1 row

> >|

↺

■

★

name\_cinema

÷

sum

÷

1

CINEMAX 2

340

```
--Zobrazí stredný cenu ticketu za '2020-04-05';
select round( avg(t.cost_ticket)::numeric, 2) || ' €' as avg_cost
from ticket t
left join session_ticket st 1<->0..n: on t.id_ticket = st.id_ticket_fk
left join session s 1..n<->1: on s.id_session = st.id_session_fk
where s.date_session = '2020-04-05';
```

Output

avg\_cost:text X

< >

1 row

> >|

↺

■

★

avg\_cost

÷

1

3.83 €

```
-- 1 pohľad s použitím množinových operácií
--zobrazí všetky animované filmy pre vekovú kategóriu do 14 rokov
-- a všetky ostatné žánre pre vekovú kategóriu < viac ako 14 rokov
select f.name_film, f.age_category, string_agg(g.name_genre, ', ')
from film f
join genre_film gf 1<->1..n: on f.id_film = gf.id_film_fk
join genre g 1..n<->1: on g.id_genre = gf.id_genre_fk
where f.age_category <14 and g.name_genre = 'Animated'
group by f.age_category, f.name_film
union
select f.name_film, f.age_category, string_agg(g.name_genre, ', ')
from film f
join genre_film gf 1<->1..n: on f.id_film = gf.id_film_fk
join genre g 1..n<->1: on g.id_genre = gf.id_genre_fk
where f.age_category >=14
group by f.age_category, f.name_film
order by age_category asc ;
```

Output

1 pohľad s použitím množinových operácií X

< >

6 rows

> >|

↺

■

★

name\_film

÷

age\_category

÷

string\_agg

÷

1

The Lord of the Rings: The Return of the King

12

Animated

2

The Dark Knight

14

Actions, Adventure

3

The Green Mile

14

Drama

4

The Matrix

16

Actions, Fantasy

5

The Godfather

16

Drama

6

Lucy

18

Actions, Drama

```
--zobrazí kino podľa počtu session
select c.name_cinema
from cinema c
-- kontrolujem id_cinema z vnutorneho selektu
where id_cinema in (
    --vyberem len id_cinema z vnutorneho selektu
    select t.id_cinema_fk
    from (
        --vypise id_cinema vsetkych halls, ktorý je v tejto cinema a sučet session v tejto cinema
        select h.id_cinema_fk, string_agg(h.name_halls, ', '), count(hs.id_session_fk) as countt
        from halls h
        join halls_session hs on h.id_halls = hs.id_halls_fk
        --zoskupujem podľa cinema
        group by h.id_cinema_fk
        --zoradujem sučet session od najväčšieho po najmenší
        order by countt desc
    ) as t
);
```

Output 2 pohľady s použitím vnorených selektov, X

5 rows

name_cinema
1 Poprad Cinema
2 Multiplex
3 Michalovce
4 CINEMAX 2
5 50

```
--zobrazí všetky filmy, a time session pre type ticket luxury
select concat(s.time_start_session, ' - ', s.time_end_session) as time_session, f.name_film
from session s
join film f on f.id_film = s.id_film_fk
where exists(
    select *
    from ticket t
    join session_ticket st on t.id_ticket = st.id_ticket_fk
    where t.type_ticket = 'luxury' and s.id_session = st.id_session_fk
);
```

Output Result 12 X

2 rows

time_session	name_film
1 09:30:00 - 11:30:00	The Matrix
2 09:30:00 - 11:30:00	The Lord of the Rings: The Return of the King

```
--vytvorím 1 sekvenciu na generovanie primárnych kľúčov a trigger, ktorá bude vkladaf hodnoty do príslušných tá
--vytvorím trigger pre tabuľku genre v ,ktorej id_genre bude samostatné generované
create or replace function func_generateKey() returns trigger as
$$
declare
    max_id int;
begin
    select max(id_genre) into max_id
    from genre;
    max_id = max_id +1;
    new.id_genre = max_id;
    return new;
end;
$$
language plpgsql;

create trigger generateKey before insert on genre
for each row
execute procedure func_generateKey();

insert into genre (name_genre)
values ('robot'),
('animal');
```

Output postgres.kino.genre X

9 rows

id_genre	name_genre
8	robot
9	animal


```

--vytvorte 1 ľubovľný trigger okrem typu triggera uvedeného vyššie
--trigger pre zmenu cenu lístka, cena ticketu nemôže byť viac ako 15€ a v maine menšie ako 1€
create trigger udpCost after update on ticket
for each row
execute procedure func_udpCost();

create or replace function func_udpCost() returns trigger as
$$
begin
    if(new.cost_ticket)<1 then
        raise exception 'cena ticketu musi byt vacsia alebo = ako 1 €';
    end if;
    if(new.cost_ticket)>15 then
        raise exception 'cena ticketu musi byt mensia alebo = ako 15 €';
    end if;
    update ticket set cost_ticket = new.cost_ticket where cost_ticket = old.cost_ticket;
    return new;
end;
$$
language plpgsql;

update ticket set cost_ticket = 16.9 where type_ticket = 'normal';

```

 udpCost  
 ROR: cena ticketu musi byt mensia alebo = ako 15 €  
 pgSQL function func\_udpCost() line 7 at RAISE

Popis	Body
2 pohľady s netriviálnym selektom nad jednou tabuľkou (nestačí použiť iba vymenovanie stĺpcov, treba použiť "niečo navyše" napr.: vstavané funkcie);	2
3 pohľady so spájaním tabuliek (1x spojenie 3 tabuliek, 1x outer join, 1x spojenie aspoň 2 tabuliek);	3
2 pohľady s použitím agregáčnych funkcií alebo zoskupenia;	2
1 pohľad s použitím množinových operácií;	2
2 pohľady s použitím vnorených selektov;	2
vytvorte 1 sekvenciu na generovanie primárnych kľúčov a trigger, ktorá bude vkladať hodnoty do príslušných tabuliek;	2
vytvorte 1 ľubovoľný trigger okrem typu triggera uvedeného vyššie (trigger musí obsahovať buď podmienku, cyklus alebo výnimku).	2
Spolu	15