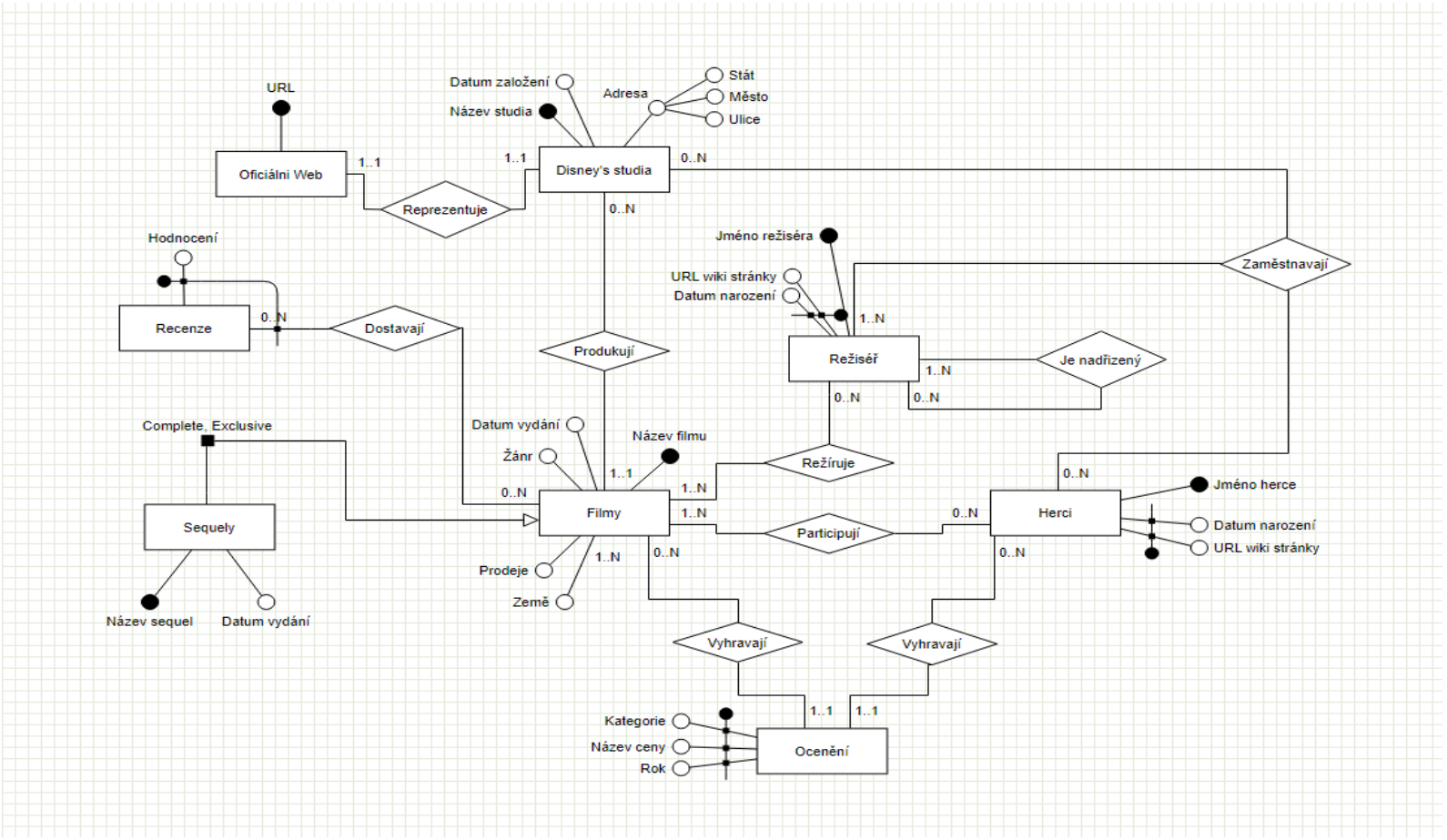
SQL documentation - filmy od Disney Studios

Konceptuální model:



Relační model:

- Disney's studia (<u>NazevFilmu</u>, <u>NazevStudio</u>, <u>URL</u>, DatumZalozeni, Adresa, Stat, Mesto, Ulice)
  - FK: NazevFilmu ⊆ Film(NazevFilmu)
- Film (<u>NazevFilmu</u>, DatumVydani, Zanr, Prodeje)
- Zeme (<u>NazevFilmu</u>, <u>Zeme</u>)
  - FK: NazevFilmu ⊆ Film(NazevFilmu)
- Herci (<u>JmenoHerce</u>, <u>DatumNarozeni</u>, <u>URLWikiStranky</u> , NazevFilmu)
  - FK: NazevFilmu ⊆ Film(NazevFilmu)
- Reziser (<u>JmenoRezisera</u>, <u>DatumNarozeni</u>, <u>URLWikiStranky</u> , NazevFilmu)
  - FK: NazevFilmu ⊆ Film(NazevFilmu)
- Je_nadrizeny (<u>Reziser</u>, Nadrizeny)
- Recenze (<u>NazevFilmu</u>, <u>Hodnoceni</u>)
  - FK: NazevFilmu ⊆ Film(NazevFilmu)
- Sequely (<u>NazevFilmu</u>, <u>NazevSequel</u>, DatumVydani)
  - FK: NazevFilmu ⊆ Film(NazevFilmu)
- Oceneni (<u>NazevFilmu</u>, <u>JmenoHerce</u>, <u>Kategorie</u>, <u>Rok</u>, <u>NazevCeny</u>)
  - FK: NazevFilmu ⊆ Film(NazevFilmu)
  - FK: JmenoHerce ⊆ Herci(JmenoHerce)

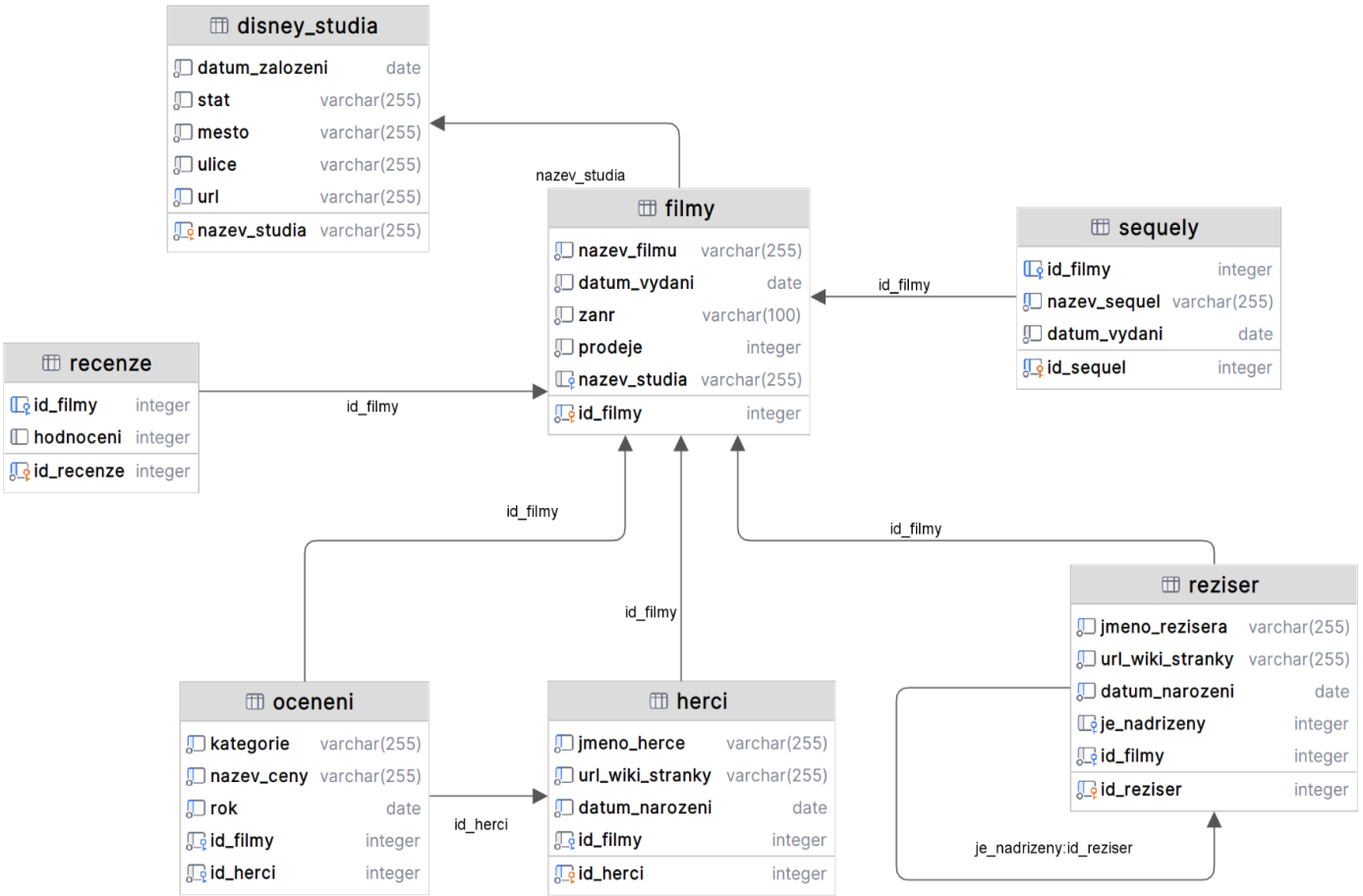
Tables:

```
CREATE TABLE Filmy (
	id_filmy SERIAL PRIMARY KEY,
	nazev_filmu VARCHAR (255) NOT NULL UNIQUE,
	datum_vydani DATE NOT NULL CHECK (datum_vydani > '1900-01-01'),
	zanr VARCHAR (100) NOT NULL,
	prodeje INT NOT NULL
);

CREATE TABLE Sequely (
   id_sequel SERIAL PRIMARY KEY,
	id_filmy INT UNIQUE REFERENCES Filmy ON UPDATE CASCADE ON DELETE
CASCADE,
	nazev_sequel VARCHAR (255) NOT NULL UNIQUE,
	datum_vydani DATE NOT NULL CHECK (datum_vydani > '1900-01-01')
);

CREATE TABLE Recenze (
   id_recenze SERIAL PRIMARY KEY,
	id_filmy INT UNIQUE REFERENCES Filmy ON UPDATE CASCADE ON DELETE
CASCADE,
	hodnoceni INT CHECK (hodnoceni BETWEEN 1 and 10)
);

CREATE TABLE Disney_studia (
	nazev_studia VARCHAR (255) NOT NULL UNIQUE,
	datum_zalozeni DATE NOT NULL CHECK (datum_zalozeni > '1900-01-01'),
	stat VARCHAR (255) NOT NULL,
	mesto VARCHAR (255) NOT NULL,
	ulice VARCHAR (255) NOT NULL,
	URL VARCHAR (255) NOT NULL UNIQUE
);

CREATE TABLE Reziser (
	id_reziser SERIAL PRIMARY KEY,
	jmeno_rezisera VARCHAR (255) NOT NULL UNIQUE,
	URL_wiki_stranky VARCHAR (255) NOT NULL,
	datum_narozeni DATE NOT NULL CHECK (datum_narozeni > '1800-01-01'),
	UNIQUE (URL_wiki_stranky, datum_narozeni),
	je_nadrizeny INT REFERENCES Reziser(id_reziser),
	id_filmy INT NOT NULL REFERENCES Filmy(id_filmy) ON UPDATE CASCADE ON
DELETE CASCADE
);
```

CREATE TABLE Herci (
id_herci SERIAL PRIMARY KEY,
jmeno_herce VARCHAR (255) NOT NULL UNIQUE,
URL_wiki_stranky VARCHAR (255) NOT NULL,
datum_narozeni DATE NOT NULL CHECK (datum_narozeni > '1800-01-01'),
UNIQUE (URL_wiki_stranky, datum_narozeni),
id_filmy INT NOT NULL REFERENCES Filmy(id_filmy) ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE Oceneni (
kategorie VARCHAR (255) NOT NULL,
nazev_ceny VARCHAR (255) NOT NULL,
rok DATE NOT NULL CHECK (rok > '1900-01-01'),
UNIQUE (kategorie, nazev_ceny, rok),
id_filmy INT NOT NULL REFERENCES Filmy(id_filmy) ON UPDATE CASCADE ON DELETE CASCADE,
id_herci INT NOT NULL REFERENCES Herci(id_herci) ON UPDATE CASCADE ON DELETE CASCADE
);

ER Model:



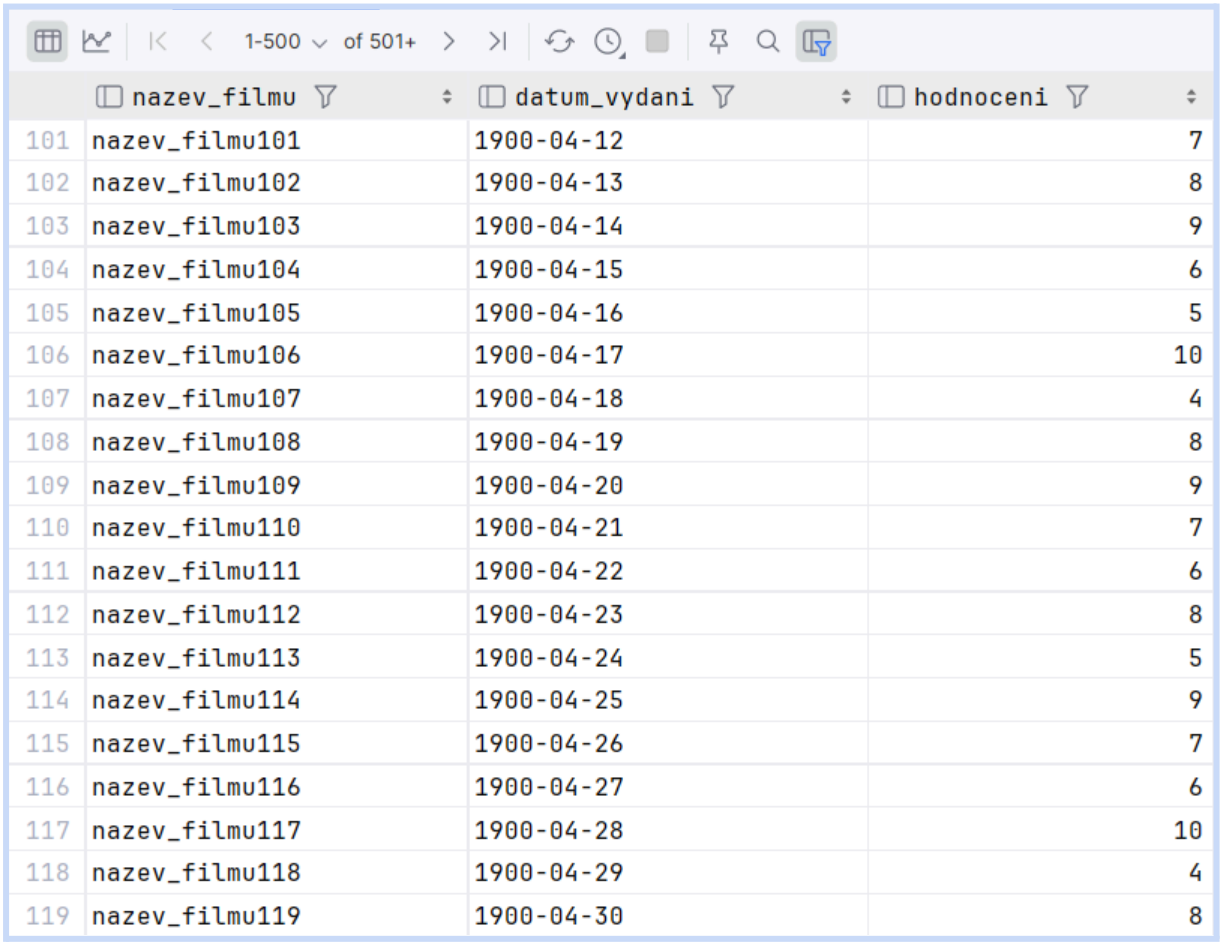
SQL dotazy pro získání údajů z databáze:

1.Vnější spojení tabulek

```
SELECT f.nazev_filmu, f.datum_vydani, r.hodnoceni
FROM Filmy f
LEFT OUTER JOIN Recenze r ON f.id_filmy = r.id_filmy;
```

```
1 ✓ SELECT f.nazev_filmu, f.datum_vydani, r.hodnoceni
2   FROM Filmy f
3   LEFT OUTER JOIN Recenze r ON f.id_filmy = r.id_filmy;
```

Tento dotaz provádí levé vnější spojení mezi tabulkami Filmy a Recenze, aby zahrnul všechny filmy, zobrazuje jejich názvy, data vydání a hodnocení recenzí, pokud jsou k dispozici.

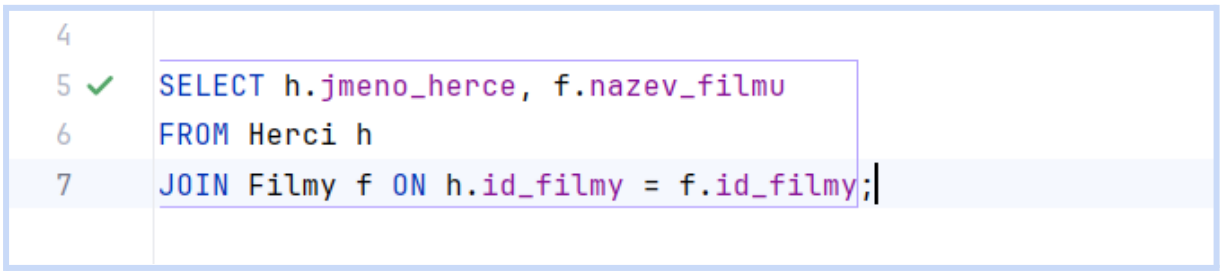


1-500 of 501+

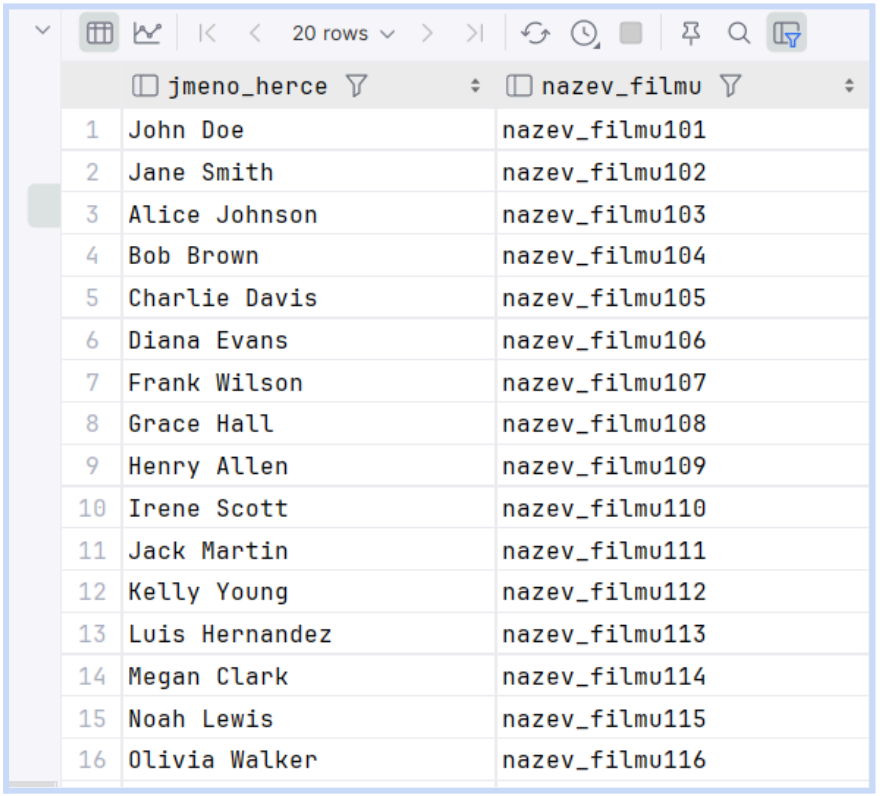
	nazev_filmu	datum_vydani	hodnoceni
101	nazev_filmu101	1900-04-12	7
102	nazev_filmu102	1900-04-13	8
103	nazev_filmu103	1900-04-14	9
104	nazev_filmu104	1900-04-15	6
105	nazev_filmu105	1900-04-16	5
106	nazev_filmu106	1900-04-17	10
107	nazev_filmu107	1900-04-18	4
108	nazev_filmu108	1900-04-19	8
109	nazev_filmu109	1900-04-20	9
110	nazev_filmu110	1900-04-21	7
111	nazev_filmu111	1900-04-22	6
112	nazev_filmu112	1900-04-23	8
113	nazev_filmu113	1900-04-24	5
114	nazev_filmu114	1900-04-25	9
115	nazev_filmu115	1900-04-26	7
116	nazev_filmu116	1900-04-27	6
117	nazev_filmu117	1900-04-28	10
118	nazev_filmu118	1900-04-29	4
119	nazev_filmu119	1900-04-30	8

2. Vnitřní spojení tabulek

```sql
SELECT h.jmeno_herce, f.nazev_filmu
FROM Herci h
JOIN Filmy f ON h.id_filmy = f.id_filmy;
```



Tento dotaz načítá pouze ty herce a filmy, kde existuje přímá asociace, tj. obě entity musí mít záznamy.



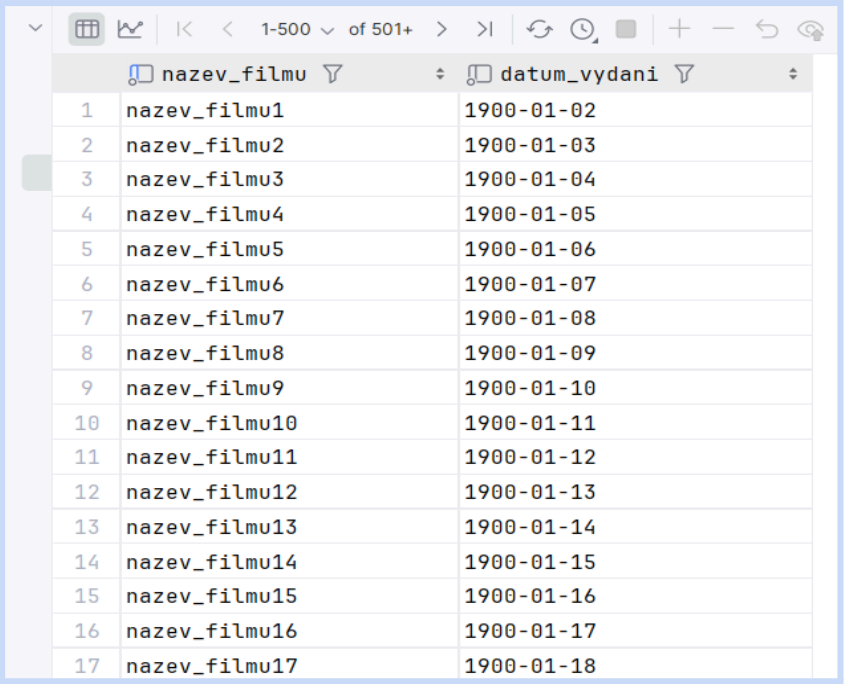
	jmeno_herce	nazev_filmu
1	John Doe	nazev_filmu101
2	Jane Smith	nazev_filmu102
3	Alice Johnson	nazev_filmu103
4	Bob Brown	nazev_filmu104
5	Charlie Davis	nazev_filmu105
6	Diana Evans	nazev_filmu106
7	Frank Wilson	nazev_filmu107
8	Grace Hall	nazev_filmu108
9	Henry Allen	nazev_filmu109
10	Irene Scott	nazev_filmu110
11	Jack Martin	nazev_filmu111
12	Kelly Young	nazev_filmu112
13	Luis Hernandez	nazev_filmu113
14	Megan Clark	nazev_filmu114
15	Noah Lewis	nazev_filmu115
16	Olivia Walker	nazev_filmu116

3. Podmínka na data

```sql
SELECT nazev_filmu, datum_vydani
FROM Filmy
WHERE datum_vydani > '1900-01-01';
```

```sql

SELECT nazev_filmu, datum_vydani
FROM Filmy
WHERE datum_vydani > '1900-01-01';

```

Tento SQL filtruje filmy tak, aby zobrazil pouze ty vydané po 1. lednu 1900.



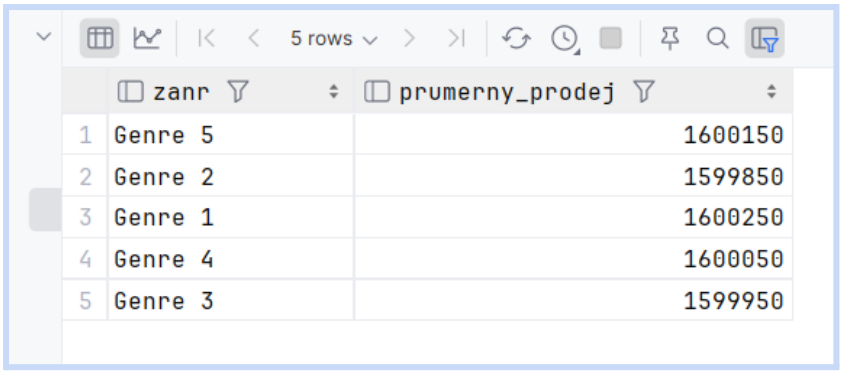
	nazev_filmu	datum_vydani
1	nazev_filmu1	1900-01-02
2	nazev_filmu2	1900-01-03
3	nazev_filmu3	1900-01-04
4	nazev_filmu4	1900-01-05
5	nazev_filmu5	1900-01-06
6	nazev_filmu6	1900-01-07
7	nazev_filmu7	1900-01-08
8	nazev_filmu8	1900-01-09
9	nazev_filmu9	1900-01-10
10	nazev_filmu10	1900-01-11
11	nazev_filmu11	1900-01-12
12	nazev_filmu12	1900-01-13
13	nazev_filmu13	1900-01-14
14	nazev_filmu14	1900-01-15
15	nazev_filmu15	1900-01-16
16	nazev_filmu16	1900-01-17
17	nazev_filmu17	1900-01-18

4. Agregace a podmínka na hodnotu agregační funkce

```sql
SELECT zanr, AVG(prodeje) AS prumerny_prodej
FROM Filmy
GROUP BY zanr
HAVING AVG(prodeje) > 1000;
```

```sql

SELECT zanr, AVG(prodeje) AS prumerny_prodej
FROM Filmy
GROUP BY zanr
HAVING AVG(prodeje) > 1000;

```

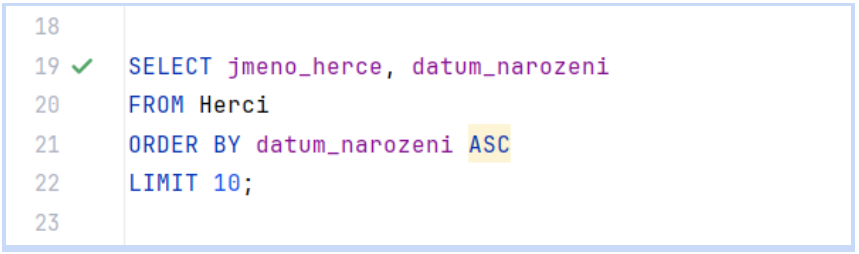
Tento dotaz seskupuje filmy podle žánru a vypočítává průměrné tržby, vrací pouze ty žánry, kde průměrné tržby přesahují 1000 USD.



	zanr	prumerny_prodej
1	Genre 5	1600150
2	Genre 2	1599850
3	Genre 1	1600250
4	Genre 4	1600050
5	Genre 3	1599950

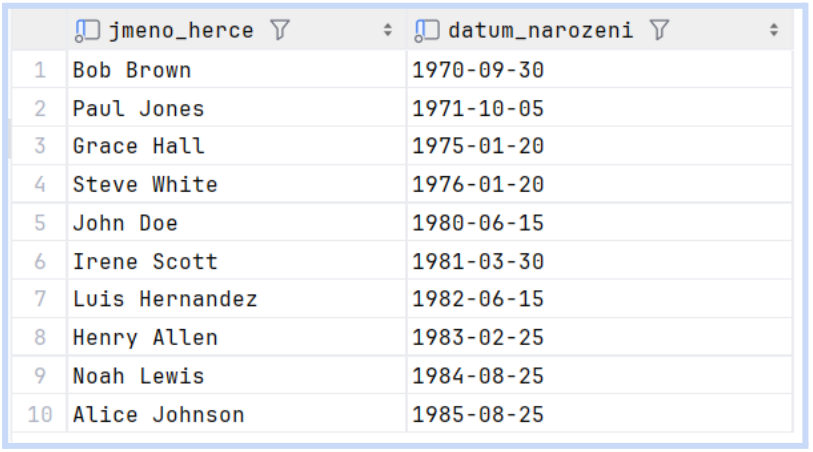
5. Řazení a stránkování

```sql
SELECT jmeno_herce, datum_narozeni
FROM Herci
ORDER BY datum_narozeni ASC
LIMIT 10;
```



```sql
18
19 ✓ SELECT jmeno_herce, datum_narozeni
20   FROM Herci
21   ORDER BY datum_narozeni ASC
22   LIMIT 10;
23
```

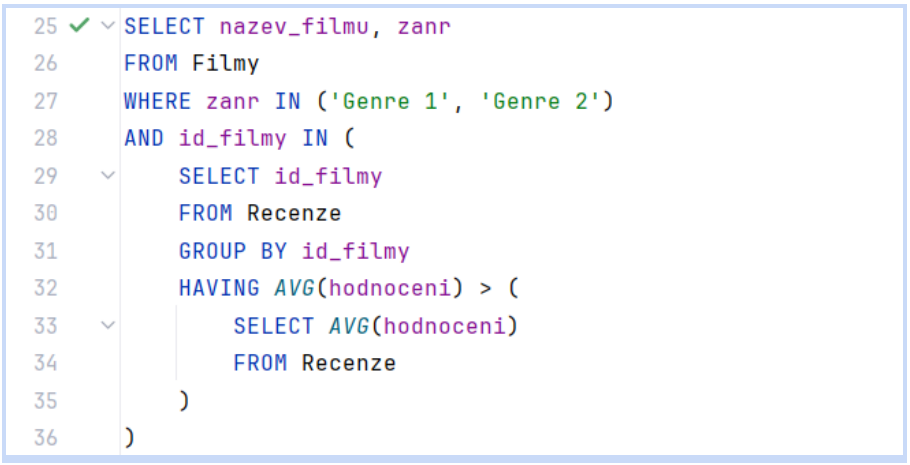
Tento dotaz řadí herce podle data jejich narození vzestupně a omezuje výstup na prvních 10 záznamů.



	jmeno_herce	datum_narozeni
1	Bob Brown	1970-09-30
2	Paul Jones	1971-10-05
3	Grace Hall	1975-01-20
4	Steve White	1976-01-20
5	John Doe	1980-06-15
6	Irene Scott	1981-03-30
7	Luis Hernandez	1982-06-15
8	Henry Allen	1983-02-25
9	Noah Lewis	1984-08-25
10	Alice Johnson	1985-08-25

6. Množinové operace a vnořený SELECT

```
SELECT nazev_filmu, zanr
FROM Filmy
WHERE zanr IN ('Genre 1', 'Genre 2')
AND id_filmy IN (
    SELECT id_filmy
    FROM Recenze
    GROUP BY id_filmy
    HAVING AVG(hodnoceni) > (
        SELECT AVG(hodnoceni)
        FROM Recenze
    )
)
```



Tento dotaz filtruje filmy podle žánru a zároveň podle kvality hodnocení, kde se vybírají ty filmy, které mají nadprůměrné hodnocení ve srovnání s ostatními filmy v databázi. Tento přístup je užitečný pro zjištění kvalitativně vysoce hodnocených filmů v konkrétních žánrech.

	nazev_filmu	zanr
1	nazev_filmu106	Genre 2
2	nazev_filmu121	Genre 2

Inserts:

```
DO $$
DECLARE
    i INTEGER := 1;
    my_string VARCHAR;
```

```
    zanr VARCHAR;
    my_date date;
    local_nazev_studia VARCHAR;
    studio_count INTEGER;

BEGIN
    SELECT COUNT(*) INTO studio_count FROM disney_studia;

    WHILE i<=32000 LOOP
        my_string := CAST(i as VARCHAR);
        my_date := '1900-01-01' :: date + (i * INTERVAL '1
day');
        zanr := 'Genre ' || (1 + i%5);
        SELECT nazev_studia INTO local_nazev_studia FROM
disney_studia LIMIT 1 OFFSET (i % studio_count);

        INSERT INTO filmy (nazev_filmu, datum_vydani, zanr,
prodeje, nazev_studia) VALUES (
        'nazev_filmu' || i, my_date, zanr, i*100,
local_nazev_studia); -- Use the renamed variable

        i := i + 1;
    END LOOP;
END;
$$;


INSERT INTO Disney_studia (nazev_studia, datum_zalozeni, stat,
mesto, ulice, URL) VALUES
('Magic Studio', '1923-10-16', 'USA', 'Burbank', 'Walt Disney
Plaza', 'http://www.magicstudio.com'),
('Dream Animations', '1934-05-18', 'USA', 'Los Angeles',
'Dreamers Lane', 'http://www.dreamanimations.com'),
('Fantasy Works', '1940-06-25', 'USA', 'San Francisco',
'Imagination Road', 'http://www.fantasyworks.com'),
('Toon Town Studios', '1950-07-30', 'USA', 'Chicago',
'Animation Avenue', 'http://www.toontownstudios.com'),
('Fable Factory', '1960-08-15', 'USA', 'New York', 'Fairy Tale
Lane', 'http://www.fablefactory.com'),
('Mythos Media', '1970-11-11', 'USA', 'Austin', 'Legend
Boulevard', 'http://www.mythosmedia.com'),
('Legend Productions', '1980-12-05', 'USA', 'Seattle', 'Hero
Passage', 'http://www.legendproductions.com'),
```

```
('Epic Entertainment', '1991-04-10', 'USA', 'Atlanta', 'Epic
Street', 'http://www.epicentertainment.com'),
('Saga Studios', '2000-09-20', 'USA', 'Boston', 'Saga Avenue',
'http://www.sagastudios.com'),
('Tale Tales Media', '2010-02-14', 'USA', 'San Diego',
'Narrative Way', 'http://www.taletalesmedia.com'),
('Chronicle Creations', '2015-03-27', 'USA', 'Dallas',
'Chronicle Court', 'http://www.chroniclecreations.com'),
('Folklore Films', '2018-08-22', 'USA', 'Miami', 'Folklore
Freeway', 'http://www.folklorefilms.com');


INSERT INTO Herci (jmeno_herce, URL_wiki_stranky,
datum_narozeni, id_filmy) VALUES
('John Doe', 'http://en.wikipedia.org/wiki/John_Doe',
'1980-06-15', 101),
('Jane Smith', 'http://en.wikipedia.org/wiki/Jane_Smith',
'1990-07-20', 102),
('Alice Johnson',
'http://en.wikipedia.org/wiki/Alice_Johnson', '1985-08-25',
103),
('Bob Brown', 'http://en.wikipedia.org/wiki/Bob_Brown',
'1970-09-30', 104),
('Charlie Davis',
'http://en.wikipedia.org/wiki/Charlie_Davis', '1995-10-05',
105),
('Diana Evans', 'http://en.wikipedia.org/wiki/Diana_Evans',
'1992-11-10', 106),
('Frank Wilson', 'http://en.wikipedia.org/wiki/Frank_Wilson',
'1988-12-15', 107),
('Grace Hall', 'http://en.wikipedia.org/wiki/Grace_Hall',
'1975-01-20', 108),
('Henry Allen', 'http://en.wikipedia.org/wiki/Henry_Allen',
'1983-02-25', 109),
('Irene Scott', 'http://en.wikipedia.org/wiki/Irene_Scott',
'1981-03-30', 110),
('Jack Martin', 'http://en.wikipedia.org/wiki/Jack_Martin',
'1994-04-05', 111),
('Kelly Young', 'http://en.wikipedia.org/wiki/Kelly_Young',
'1986-05-10', 112),
('Luis Hernandez',
'http://en.wikipedia.org/wiki/Luis_Hernandez', '1982-06-15',
113),
```

```
('Megan Clark', 'http://en.wikipedia.org/wiki/Megan_Clark',
'1993-07-20', 114),
('Noah Lewis', 'http://en.wikipedia.org/wiki/Noah_Lewis',
'1984-08-25', 115),
('Olivia Walker',
'http://en.wikipedia.org/wiki/Olivia_Walker', '1989-09-30',
116),
('Paul Jones', 'http://en.wikipedia.org/wiki/Paul_Jones',
'1971-10-05', 117),
('Quinn Taylor', 'http://en.wikipedia.org/wiki/Quinn_Taylor',
'1990-11-10', 118),
('Rachel Moore', 'http://en.wikipedia.org/wiki/Rachel_Moore',
'1991-12-15', 119),
('Steve White', 'http://en.wikipedia.org/wiki/Steve_White',
'1976-01-20', 120);


INSERT INTO Oceneni (kategorie, nazev_ceny, rok, id_filmy,
id_herci) VALUES
('Best Picture', 'Golden Film Award', '2022-01-01', 101, 21),
('Best Actor', 'Golden Star Award', '2022-01-01', 102, 22),
('Best Director', 'Director's Guild Award', '2022-01-01', 103,
23),
('Best Screenplay', 'Writers Guild Award', '2022-01-01', 104,
24),
('Best Cinematography', 'Cinematography Society Award',
'2022-01-01', 105, 25),
('Best Actress', 'Golden Film Award', '2021-01-01', 106, 26),
('Best Supporting Actor', 'Silver Star Award', '2021-01-01',
107, 27),
('Best Supporting Actress', 'Silver Screen Award',
'2021-01-01', 108, 28),
('Best Visual Effects', 'Visual Effects Society Award',
'2021-01-01', 109, 29),
('Best Sound Editing', 'Sound Editors Guild Award',
'2021-01-01', 110, 30),
('Best Original Score', 'Composers Guild Award', '2020-01-01',
111, 31),
('Best Costume Design', 'Costume Designers Award',
'2020-01-01', 112, 32),
('Best Production Design', 'Production Design Guild Award',
'2020-01-01', 113, 33),
('Best Makeup and Hairstyling', 'Makeup Artists Award',
'2020-01-01', 114, 34),
```

```
('Best Animated Feature', 'Animation Guild Award',
'2020-01-01', 115, 35),
('Best Documentary Feature', 'Documentary Guild Award',
'2019-01-01', 116, 36),
('Best Foreign Language Film', 'International Film Award',
'2019-01-01', 117, 37),
('Best Editing', 'Editors Guild Award', '2019-01-01', 118,
38),
('Best Sound Mixing', 'Sound Mixers Guild Award',
'2019-01-01', 119, 39),
('Best Original Song', 'Composers Guild Award', '2019-01-01',
120, 40);

INSERT INTO Recenze (id_filmy, hodnoceni) VALUES
(101, 7),
(102, 8),
(103, 9),
(104, 6),
(105, 5),
(106, 10),
(107, 4),
(108, 8),
(109, 9),
(110, 7),
(111, 6),
(112, 8),
(113, 5),
(114, 9),
(115, 7),
(116, 6),
(117, 10),
(118, 4),
(119, 8),
(120, 7),
(121, 9),
(122, 6),
(123, 8),
(124, 5),
(125, 7),
(126, 6),
(127, 8),
(128, 9),
(129, 10),
(130, 5);
```

```sql
INSERT INTO Reziser (jmeno_rezisera, URL_wiki_stranky,
datum_narozeni, je_nadrizeny, id_filmy) VALUES
('Michael Bay', 'http://en.wikipedia.org/wiki/Michael_Bay',
'1965-02-17', NULL, 101),
('Quentin Tarantino',
'http://en.wikipedia.org/wiki/Quentin_Tarantino',
'1963-03-27', NULL, 102),
('Sofia Coppola',
'http://en.wikipedia.org/wiki/Sofia_Coppola', '1971-05-14',
NULL, 103),
('Christopher Nolan',
'http://en.wikipedia.org/wiki/Christopher_Nolan',
'1970-07-30', NULL, 104),
('Greta Gerwig', 'http://en.wikipedia.org/wiki/Greta_Gerwig',
'1983-08-04', NULL, 105),
('Wes Anderson', 'http://en.wikipedia.org/wiki/Wes_Anderson',
'1969-05-01', NULL, 106),
('Patty Jenkins',
'http://en.wikipedia.org/wiki/Patty_Jenkins', '1971-07-24',
NULL, 107),
('Ridley Scott', 'http://en.wikipedia.org/wiki/Ridley_Scott',
'1937-11-30', NULL, 108),
('Taika Waititi',
'http://en.wikipedia.org/wiki/Taika_Waititi', '1975-08-16',
NULL, 109),
('Bong Joon Ho', 'http://en.wikipedia.org/wiki/Bong_Joon_Ho',
'1969-09-14', NULL, 110);
```

```sql
INSERT INTO Sequely (id_filmy, nazev_sequel, datum_vydani)
VALUES
(101, 'Epic Journey: The Return', '2023-07-15'),
(102, 'Nightfall: Part II', '2023-05-21'),
(103, 'Beyond the Stars: Resurgence', '2024-03-10'),
(104, 'Whispering Shadows: Aftermath', '2023-12-01'),
(105, 'Warriors of Light: Reborn', '2023-10-23'),
(106, 'Silent Oceans: Depths Unknown', '2024-01-18'),
(107, 'Dreamers Awake: The Next Chapter', '2023-09-15'),
(108, 'Guardians of Tomorrow: Legacy', '2024-04-20'),
(109, 'Mystic Winds: The Awakening', '2023-11-03'),
(110, 'Frozen Thrones: The Lost Empire', '2024-02-28');
```