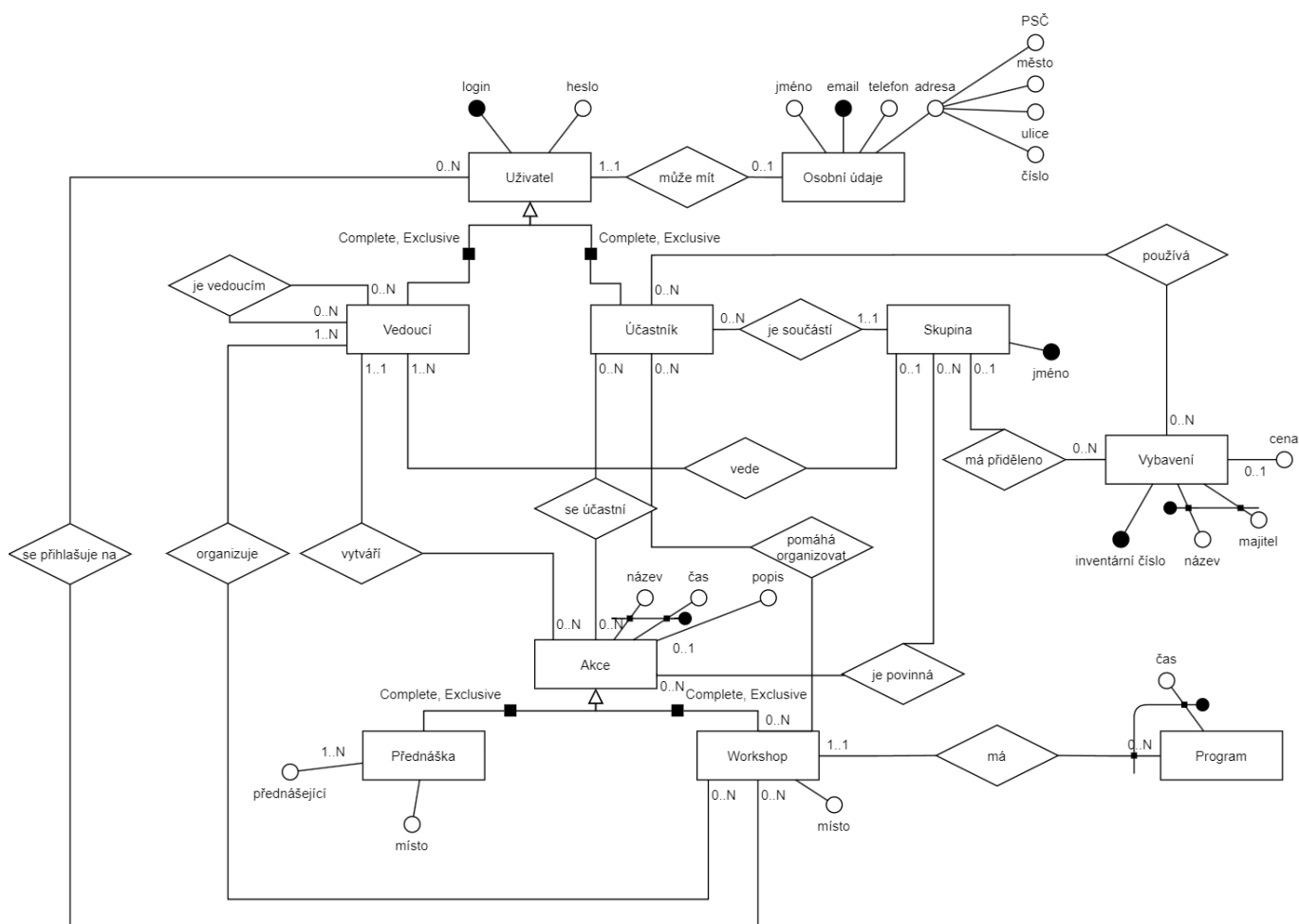


Konceptuální model



Relační model

- Uživatel (login, heslo)
- Osobní Údaje (login, jméno, email, telefon, PSC, město, ulice, číslo)
 - FK: (login) \subseteq Uživatel(login)
- Vedoucí (login)
 - FK: (login) \subseteq Uživatel(login)
- Účastník (login)
 - FK: (login) \subseteq Uživatel(login)
- Skupina (jméno)
- Akce (název, čas)
- AkcePopis (název, čas, popis)

- FK: (název, čas) \subseteq Akce(název, čas)
- Přednáška (název, čas, místo)
 - FK: (název, čas) \subseteq Akce(název, čas)
- Přednášející (jméno, název, čas)
 - FK: (název, čas) \subseteq Přednáška(název, čas)
- Workshop (název, čas, místo)
 - FK: (název, čas) \subseteq Akce(název, čas)
- Program (čas, položky, název, čas)
 - FK: (název, čas) \subseteq Workshop(název, čas)
- Vybavení (inventární číslo, název, majitel)
- VybaveníCena (inventární číslo, cena)
 - FK: (inventární číslo) \subseteq Vybavení(inventární číslo)
- Má přiděleno (skupina, vybavení)
 - FK: (skupina) \subseteq Skupina(jméno)
 - FK: (vybavení) \subseteq Vybavení(inventární číslo)
- Používá (účastník, vybavení)
 - FK: (účastník) \subseteq Účastník(login)
 - FK: (vybavení) \subseteq Vybavení(inventární číslo)
- Je součástí (skupina, účastník)
 - FK: (skupina) \subseteq Skupina(jméno)
 - FK: (účastník) \subseteq Účastník(login)
- Vede (skupina, vedoucí)
 - FK: (skupina) \subseteq Skupina(jméno)
 - FK: (vedoucí) \subseteq Vedoucí(login)
- Se přihlašuje na (uživatel, workshop)
 - FK: (uživatel) \subseteq Uživatel(login)
 - FK: (workshop) \subseteq Workshop(akce)
- Organizuje (vedoucí, workshop)
 - FK: (vedoucí) \subseteq Vedoucí(login)
 - FK: (workshop) \subseteq Workshop(akce)
- Vytváří (vedoucí, akce)

- FK: (vedoucí) \subseteq Vedoucí(login)
- FK: (akce) \subseteq Akce(název, čas)
- Pomáhá organizovat (účastník, workshop)
 - FK: (účastník) \subseteq Účastník(login)
 - FK: (workshop) \subseteq Workshop(akce)
- Je povinná (akce, skupina)
 - FK: (akce) \subseteq Akce(název, čas)
 - FK: (skupina) \subseteq Skupina(jméno)
- Je vedoucím (nadřízený, podřízený)
 - FK: (nadřízený) \subseteq Vedoucí(login)
 - FK: (podřízený) \subseteq Vedoucí(login)
- Se účastní (účastník, akce)
 - FK: (účastník) \subseteq Účastník(login)
 - FK: (akce) \subseteq Akce(název, čas)



SQL Tabulky

```
CREATE TABLE Uzivatel (  
    id_uzivatel SERIAL PRIMARY KEY,  
    login VARCHAR(64) NOT NULL,  
    heslo VARCHAR(128) NOT NULL,  
    salt VARCHAR(32) NOT NULL,  
    jmeno VARCHAR(64),  
    email VARCHAR(64),  
    telefon VARCHAR(23),  
    psc INTEGER CHECK (psc >= 10000 AND psc <= 99999),  
    mesto VARCHAR(64),  
    ulice VARCHAR(64),  
    cislo VARCHAR(16),  
    UNIQUE (login)  
);  
  
CREATE TABLE Skupina (  
    id_skupina SERIAL PRIMARY KEY,  
    jmeno VARCHAR(64) NOT NULL,  
    UNIQUE (jmeno)  
);  
  
/* ON DELETE CASCADE je použito, protože pokud smažeme uživatele, tak  
záznam v tabulce vedoucích ztrácí smysl. */  
/* ON UPDATE CASCADE je použito, protože pokud změníme skupinu kterou  
vedoucí vede, musíme také změnit skupinu v tabulce vedoucích. */  
CREATE TABLE Vedouci (  
    id_uzivatel INTEGER PRIMARY KEY,  
    vede INT,  
    FOREIGN KEY (id_uzivatel) REFERENCES Uzivatel ON DELETE CASCADE,  
    FOREIGN KEY (vede) REFERENCES Skupina(id_skupina) ON UPDATE CASCADE  
);  
  
/* ON DELETE a ON UPDATE CASCADE je podobné jako u tabulky Vedouci. */  
CREATE TABLE Ucastnik (  
    id_uzivatel INTEGER PRIMARY KEY,  
    soucasti_skupiny INT NOT NULL,  
    FOREIGN KEY (id_uzivatel) REFERENCES Uzivatel ON DELETE CASCADE,  
    FOREIGN KEY (soucasti_skupiny) REFERENCES Skupina(id_skupina) ON UPDATE  
CASCADE  
);
```

```
/* Po smazání uživatele smažeme akce které vytvořil. */
CREATE TABLE Akce (
    id_akce SERIAL PRIMARY KEY,
    created_by INTEGER NOT NULL,
    nazev VARCHAR(64) NOT NULL,
    cas TIMESTAMP NOT NULL,
    popis TEXT,
    UNIQUE (nazev, cas),
    FOREIGN KEY (created_by) REFERENCES Vedouci(id_uzivatel) ON DELETE
CASCADE
);
/* Při smazání akce, smažeme i záznam v tabulce Prednaska. */
CREATE TABLE Prednaska (
    id_akce INTEGER PRIMARY KEY,
    FOREIGN KEY (id_akce) REFERENCES Akce ON DELETE CASCADE,
    misto VARCHAR(64) NOT NULL
);
/* Při smazání akce, smažeme i záznam v tabulce Prednasejici. */
CREATE TABLE Prednasejici (
    id_prednasejici SERIAL PRIMARY KEY,
    jmeno VARCHAR(64) NOT NULL,
    id_akce INTEGER NOT NULL,
    FOREIGN KEY (id_akce) REFERENCES Prednaska ON DELETE CASCADE,
    UNIQUE (jmeno, id_akce)
);
/* Při smazání akce, smažeme i záznam v tabulce Workshop. */
CREATE TABLE Workshop (
    id_akce INTEGER PRIMARY KEY,
    FOREIGN KEY (id_akce) REFERENCES Akce ON DELETE CASCADE,
    misto VARCHAR(64) NOT NULL
);
/* Při smazání akce, smažeme i záznam v tabulce Program. */
CREATE TABLE Program (
    id_program SERIAL PRIMARY KEY,
    cas_polozky TIMESTAMP NOT NULL,
    nazev VARCHAR(64) NOT NULL,
    id_akce INTEGER NOT NULL,
    FOREIGN KEY (id_akce) REFERENCES Workshop ON DELETE CASCADE,
    UNIQUE (cas_polozky, id_akce)
);

CREATE TABLE Vybaveni (
    id_vybaveni SERIAL PRIMARY KEY,
    inventarni_cislo VARCHAR(64) NOT NULL,
    nazev VARCHAR(64) NOT NULL,
    majitel VARCHAR(64) NOT NULL,
    cena DECIMAL(10,2),
    UNIQUE (inventarni_cislo),
    UNIQUE (nazev, majitel)
);
```

```
/* Při změně / smazání skupiny nebo vybavení, musíme smazat i záznam v
MaPrirazeno */
CREATE TABLE MaPrirazeno (
    id_skupina INTEGER NOT NULL,
    id_vybaveni INTEGER NOT NULL,
    FOREIGN KEY (id_skupina) REFERENCES Skupina ON DELETE CASCADE,
    FOREIGN KEY (id_vybaveni) REFERENCES Vybaveni ON DELETE CASCADE,
);
/* Při smazání uživatele nebo vybavení, smažeme i záznam v tabulce Pouziva.
*/
CREATE TABLE Pouziva (
    id_uzivatel INTEGER NOT NULL,
    id_vybaveni INTEGER NOT NULL,
    FOREIGN KEY (id_uzivatel) REFERENCES Uzivatel ON DELETE CASCADE,
    FOREIGN KEY (id_vybaveni) REFERENCES Vybaveni ON DELETE CASCADE
);
/* Při smazání akce nebo uživatele, smažeme i záznam v tabulce
SePrihlasujeNa. */
CREATE TABLE SePrihlasujeNa (
    id_uzivatel INTEGER NOT NULL,
    id_akce INTEGER NOT NULL,
    FOREIGN KEY (id_uzivatel) REFERENCES Ucastnik ON DELETE CASCADE,
    FOREIGN KEY (id_akce) REFERENCES Workshop ON DELETE CASCADE,
);
/* Při smazání vedoucího nebo akce, smažeme i záznam v tabulce Organizuje.
*/
CREATE TABLE Organizuje (
    id_uzivatel INTEGER NOT NULL,
    id_akce INTEGER NOT NULL,
    FOREIGN KEY (id_uzivatel) REFERENCES Vedouci ON DELETE CASCADE,
    FOREIGN KEY (id_akce) REFERENCES Workshop ON DELETE CASCADE,
    UNIQUE (id_uzivatel, id_akce)
);
/* Při smazání účastníka nebo akce, smažeme i záznam v tabulce
PomahaOrganizovat. */
CREATE TABLE PomahaOrganizovat (
    id_uzivatel INTEGER NOT NULL,
    id_akce INTEGER NOT NULL,
    FOREIGN KEY (id_uzivatel) REFERENCES Ucastnik ON DELETE CASCADE,
    FOREIGN KEY (id_akce) REFERENCES Workshop ON DELETE CASCADE,
    UNIQUE (id_uzivatel, id_akce)
);
/* Při smazání akce nebo skupiny smažeme i záznam v tabulce JePovinna. */
CREATE TABLE JePovinna (
    id_akce INTEGER NOT NULL,
    id_skupina INTEGER NOT NULL,
    FOREIGN KEY (id_akce) REFERENCES Akce ON DELETE CASCADE,
    FOREIGN KEY (id_skupina) REFERENCES Skupina ON DELETE CASCADE,
    UNIQUE (id_akce, id_skupina)
);
```

```
/* Při smazání vedoucího - podřízeného nebo nadřízeného, smažeme i záznam v
tabulce JeVedoucim. */
CREATE TABLE JeVedoucim (
    nadrizeny INTEGER NOT NULL,
    podrizeny INTEGER NOT NULL,
    FOREIGN KEY (nadrizeny) REFERENCES Vedouci(id_uzivatel) ON DELETE
CASCADE,
    FOREIGN KEY (podrizeny) REFERENCES Vedouci(id_uzivatel) ON DELETE
CASCADE,
    UNIQUE (nadrizeny, podrizeny)
);

/* Při smazání účastníka nebo akce, smažeme i záznam v tabulce SeUcastni.
*/
CREATE TABLE SeUcastni (
    id_uzivatel INTEGER NOT NULL,
    id_akce INTEGER NOT NULL,
    FOREIGN KEY (id_uzivatel) REFERENCES Ucastnik ON DELETE CASCADE,
    FOREIGN KEY (id_akce) REFERENCES Akce ON DELETE CASCADE,
    UNIQUE (id_uzivatel, id_akce)
);
```

Data

Pro generování dat jsem použil knihovnu `faker`, data jsem vygeneroval do `.sql` souborů, které pak za pomoci `psql` nahrávám na server.

Data vkládám pomocí

```
INSERT INTO Tabulka
    (sloupce, ..., ...) VALUES
    ('val1', 'val2', 'val3'),
    ('val4', 'val5', 'val6');
```

SQL Dotazy

Vylistování všech vedoucích (a skupin které vedou), kteří nemají vyplněno telefonní číslo.

```
SELECT Uzivatel.login, Uzivatel.jmeno, Skupina.jmeno AS skupina_jmeno FROM
Uzivatel
    INNER JOIN Vedouci ON Uzivatel.id_uzivatel = Vedouci.id_uzivatel
    INNER JOIN Skupina ON Vedouci.vede = Skupina.id_skupina
WHERE Uzivatel.telefon IS NULL
ORDER BY Uzivatel.login ASC LIMIT 10;
```



```

jakub@NitroN50-620:~/ctu/dbs/hw3$ make test_vedouci
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f tests/vedouci.sql
      login      |      jmeno      |      skupina_jmeno
-----+-----+-----
 ctiborhavlicek_743. | Ctibor Havlíček | Analogová astrofotografie
 nadezdavackova_742c |                 | Deep sky
 otobarta_839f      | Oto Bárta      | Digitální astrofotografie
 tomaskucera_309y    | Tomáš Kučera    | Deep sky
(4 rows)

```

Výpis všech akcí, které se konaly v prvním týdnu v dubnu (vždy mezi půlnocí a 8:00), s uživateli, kterými byly vytvořeny.

```

SELECT Akce.nazev, Akce.cas, Uzivatel.jmeno AS zalozil FROM Akce
  INNER JOIN Vedouci ON Akce.created_by = Vedouci.id_uzivatel
  INNER JOIN Uzivatel ON Vedouci.id_uzivatel = Uzivatel.id_uzivatel
 WHERE Akce.cas >= '2024-04-01' AND Akce.cas <= '2024-04-07' AND
EXTRACT(HOUR FROM Akce.cas) >= 0 AND EXTRACT(HOUR FROM Akce.cas) < 8
 ORDER BY Akce.cas ASC;

```

```

jakub@NitroN50-620:~/ctu/dbs/hw3$ make test_akce
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f tests/akce.sql
      nazev      |      cas      |      zalozil
-----+-----+-----
 skleněný | 2024-04-02 03:11:13 | Adéla Musilová
 čtvrtý | 2024-04-02 05:42:25 | Oto Bárta
 cyklus | 2024-04-03 01:09:45 | Štěpánka Brožová
 snídaně | 2024-04-03 01:17:21 | Zdislava Bednářová
 položit | 2024-04-04 04:41:11 | Ctibor Havlíček
 učení | 2024-04-04 06:35:49 | Adéla Musilová
 adresa | 2024-04-05 01:18:12 |
 dojit | 2024-04-05 04:32:28 | Iveta Strnadová Th.D.
 skutečný | 2024-04-05 05:52:46 |
(9 rows)

```

Zobrazí uživatele, kteří se přihlásili na nejvíce workshopů.

```

SELECT Uzivatel.login, Uzivatel.jmeno, COUNT(*) AS pocet FROM Uzivatel
  INNER JOIN SePrihlasujeNa ON Uzivatel.id_uzivatel =
SePrihlasujeNa.id_uzivatel
 GROUP BY Uzivatel.jmeno, Uzivatel.login
 ORDER BY pocet DESC
 LIMIT 10;

```

```

jakub@NitroN50-620:~/ctu/dbs/hw3$ make test_max_workshops
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f tests/max_workshops.sql

```

login	jmeno	pocet
radovancechph.d._206_	Radovan Čech Ph.D.	42
anastazieberanova_654g	Anastázie Beranová	42
jonasriha_590#	Jonáš Říha	39
pavlakrausova_807n		38
irenavlckova_452y	Irena Vlčková	38
jaroslavamachova_516y	Jaroslava Machová	38
alicevavrova_694m	Alice Vávrová	37
ludekhruska_325.	Luděk Hruška	37
romanaduskova_41h		37
hynekpólak_866v	Hynek Polák	37

(10 rows)

Vypíše přes vnořený SELECT všechny kusy vybavení (název, majitele, cenu, inventární číslo), která mají cenu, která je vyšší než 10000 korun, a vybavení (název) začíná na Fujifilm, DJI nebo RED Digital Cinema.

```

SELECT nazev, majitel, cena, inventarni_cislo FROM Vybaveni
WHERE cena > 10000 AND nazev IN (
    SELECT nazev FROM Vybaveni
    WHERE nazev LIKE 'Fujifilm%' OR nazev LIKE 'DJI%' OR nazev LIKE
'RED Digital Cinema%'
)
ORDER BY cena DESC, nazev ASC
LIMIT 10;

```

```

jakub@NitroN50-620:~/ctu/dbs/hw3$ make test_select_select
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f tests/select_select.sql

```

nazev	majitel	cena	inventarni_cislo
Fujifilm X-T2	Leoš Horáček	49997.70	0260943331061
Fujifilm X-H1	Sára Moravcová	49997.68	0562769127560
Fujifilm X-T100	Apolena Vítková	49950.39	1356060251883
DJI Osmo Pocket	Jozef Beneš	49941.83	0759598265895
DJI Inspire 2	Radim Šmíd	49929.98	0445795721523
DJI 136mm f/2.6	Věra Pokorná	49912.03	0559252790776
Fujifilm 247mm f/2.2	Marie Černá	49903.40	0692680311707
Fujifilm 215mm f/1.1	Boleslav Staněk	49856.91	0339676642468
RED Digital Cinema Raven	Hubert Kopecký	49839.07	0925389941471
DJI Osmo Action	Jaromír Beneš	49819.90	1366975042322

(10 rows)

Spočítá průměrnou cenu vybavení, které je buďto Canon EOS 5D, Canon EOS 6D nebo Canon EOS 7D.

```
SELECT AVG(cena) FROM Vybaveni
WHERE nazev IN ('Canon EOS 5D', 'Canon EOS 6D', 'Canon EOS 7D') AND
cena IS NOT NULL;
```

```
jakub@NitroN50-620:~/ctu/dbs/hw3$ make test_average
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f tests/average.sql
      avg
-----
28347.509264705882
(1 row)
```

Zobrazí skupiny, které mají průměrnou cenu vybavení vyšší než 25 500.

```
SELECT Skupina.jmeno, AVG(Vybaveni.cena) AS prumerna_cena FROM Skupina
JOIN MaPrirazeno ON Skupina.id_skupina = MaPrirazeno.id_skupina
JOIN Vybaveni ON MaPrirazeno.id_vybaveni = Vybaveni.id_vybaveni
GROUP BY Skupina.jmeno
HAVING AVG(Vybaveni.cena) > 25500 ORDER BY prumerna_cena DESC;
```

```
jakub@NitroN50-620:~/ctu/dbs/hw3$ make test_average_groups
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f tests/average_groups.sql
      jmeno                | prumerna_cena
-----+-----
Analogová astrofotografie | 26990.873104575163
Digitální astrofotografie | 26504.852956810631
CCD                        | 25565.674576271186
(3 rows)
```

Zobrazí druhých 10 vybavení s cenou v rozmezí 10-15k zaokrouhlenou na celá čísla, seřazených podle ceny, jména a jména majitele.

```
SELECT Vybaveni.nazev, Vybaveni.majitel, ROUND(Vybaveni.cena) AS cena FROM
Vybaveni
WHERE Vybaveni.cena >= 10000 AND Vybaveni.cena <= 15000
ORDER BY cena, Vybaveni.nazev, Vybaveni.majitel
LIMIT 10 OFFSET 10;
```

```

jakub@NitroN50-620:~/ctu/dbs/hw3$ make test_offset
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f tests/offset.sql

```

nazev	majitel	cena
Olympus PEN-F	Lubomír Blažek	10013
Orion refractor 806/147 f5.48	Marek Ježek	10017
Orion refractor 928/234 f3.97	Mahulena Králová	10018
Sigma 156mm f/1.7	Robin Navrátil	10018
Blackmagic Design 113mm f/2.1	Bohuslav Hruška	10020
Sky-Watcher reflector 877/167 f5.25	Lenka Šmídová	10020
Takahashi reflector 1112/230 f4.83	Štěpánka Bednářová	10020
Canon 18mm f/1.7	Eduard Havlíček	10022
Sky-Watcher refractor 1236/136 f9.09	Karolína Fialová	10023
Hasselblad X1D-50c	Andrea Bártová	10024

(10 rows)

Vypíše login, jméno, id skupiny a jméno skupiny uživatelů, kteří jsou v nějaké skupině. Pokud uživatel není v žádné skupině, vypíše se NULL.

```

SELECT U.login, U.jmeno, S.jmeno AS skupina_jmeno
FROM Uzivatel U
LEFT OUTER JOIN Ucastnik UC ON U.id_uzivatel = UC.id_uzivatel
LEFT OUTER JOIN Skupina S ON UC.soucasti_skupiny = S.id_skupina
ORDER BY U.id_uzivatel ASC LIMIT 15;

```

```

jakub@NitroN50-620:~/ctu/dbs/hw3$ make test_outer
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f tests/outer.sql

```

login	jmeno	skupina_jmeno
marianhorvath_296#	Marian Horváth	
stanislavmusil_686d	Stanislav Musil	Analogová astrofotografie
gabrieladostalova_50a	Gabriela Dostálová	
bohuslavstejskal_575o		
tomaskucera_309y	Tomáš Kučera	
jaroslavamachova_516y	Jaroslava Machová	Deep sky
hynekpolar_866v	Hynek Polák	Digitální astrofotografie
janbenes_793@	Jan Beneš	CCD
romanaduskova_41h		Digitální astrofotografie
jonasriha_590#	Jonáš Říha	Deep sky
irenavlckova_452y	Irena Vlčková	Digitální astrofotografie
zdenkacermakova_316j		
rostislavkovar_490v		CCD
stanislavaslavikova_32s	Stanislava Slavíková	Analogová astrofotografie
alicevavrova_694m	Alice Vávrová	Digitální astrofotografie

(15 rows)