

# Masážní salon – Relax

## Table of Contents

Popis, entity a vztahy.....	2
V takovém masážním salonu by přibližně existovaly následující entity: .....	2
S následujícími přibližnými vztahy:.....	2
ER model.....	3
Relační model to SQL.....	4
Vytvoření tabulek.....	6
Osoba.....	6
Zamestnanec.....	7
Oficialni_stranka.....	7
Zakaznik .....	7
Doporucil.....	8
Masaz.....	8
Objednavka .....	8
Druh.....	9
Druh_masaze .....	9
Zamestnanec_Dela_Masaz .....	9
Recenze.....	9
Dotazy.....	10
Vnější spojení tabulek.....	10
Vnitřní spojení tabulek.....	10
Podmínku na data.....	10
Agregaci a podmínku na hodnotu agregační funkce .....	11
Řazení a stránkování.....	11
Množinové operace .....	12
Vnořený SELECT.....	12
Diagram .....	13

## Popis, entity a vztahy

V důsledku sedavého způsobu života a stresu jsme se rozhodli navrhnout databázový systém pro masážní salon. Toto téma nás zajímá nejen proto, že nás často bolí záda a přemýšlíme, kdy bychom mohli navštívit maséra, ale také proto, že efektivní správa rezervací a zákaznických dat může výrazně zlepšit provoz salonu.

Navrhovaná databáze bude uchovávat informace o klientech, masérech, nabízených službách a rezervacích. Díky tomu umožní automatizaci objednávek klientů, sledování historie návštěv a analýzu vytíženosti masérů, což přispěje k efektivnějšímu chodu podniku.

V takovém masážním salonu by přibližně existovaly následující entity:

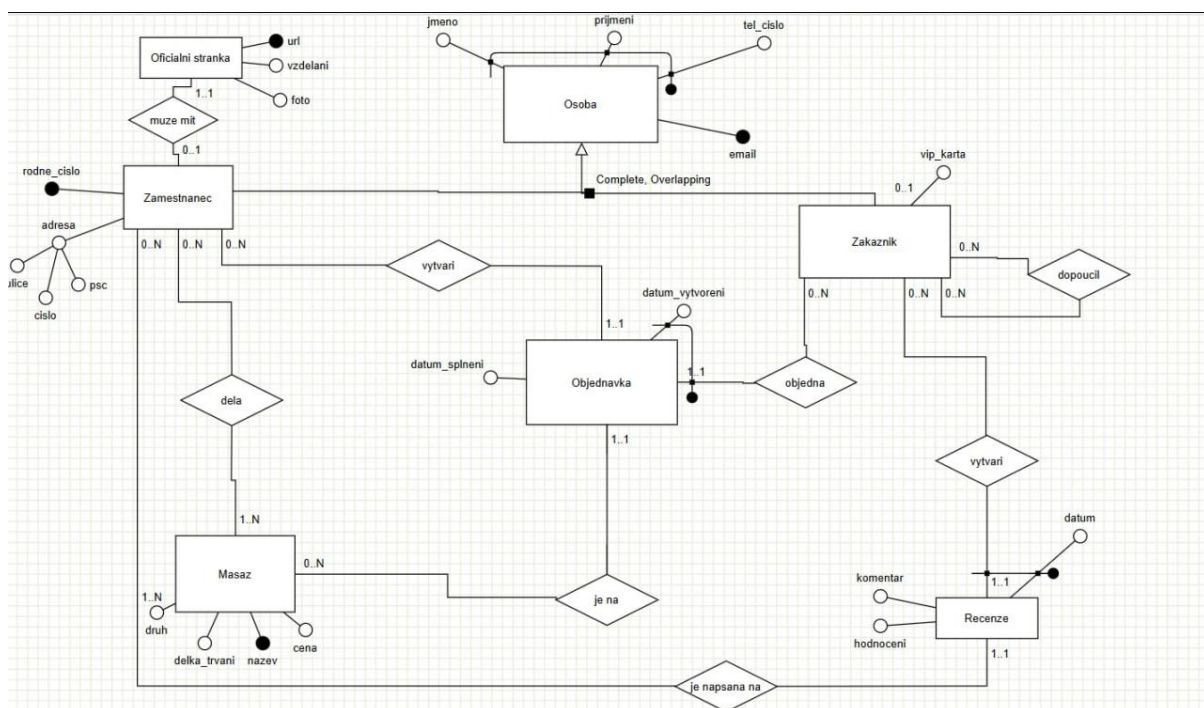
- **Osoba** (jméno, příjmení, telefonní číslo, email)
- **Zaměstnanec** (rodné číslo, adresa)
- **Oficiální stránka** (url, vzdělání, foto)
- **Zákazník** (vip\_karta)
- **Objednávka** (datum\_vytvoření, datum\_splnění)
- **Masáž** (název, druh, délka\_trvání, cena)
- **Recenze** (datum, komentář, hodnocení)

S následujícími přibližnými vztahy:

- **Osoba** může být dvou typů: **Zaměstnanec** a **Zákazník**.
  - **Zaměstnanec** může být zároveň **Zákazníkem**.
- **Zaměstnanec** vytváří **Objednávky**.
  - Ne každý **Zaměstnanec** musí vytvářet **Objednávky** (např. uklízečka). **Zaměstnanec** může vytvořit více **Objednávek**.
- **Zaměstnanec** může mít **Oficiální stránku**.
  - **Zaměstnanec** nemusí mít **Oficiální stránku**. Pokud ji má, tak je pouze jedna.
- **Zaměstnanec** provádí **Masáže**.
  - Ne každý **Zaměstnanec** provádí **Masáže** (např. administrátor nebo uklízečka). **Zaměstnanec** může provádět více **Masáží**.
- **Zákazník** objednává **Objednávku**.
  - **Zákazník** může vytvořit více **Objednávek**.
- **Zákazník** může napsat **Recenze**.
  - **Zákazník** může hodnotit vícekrát, ale nemusí.
- **Zákazník** může doporučit jiného **Zákazníka**.
  - Ne každý **Zákazník** musí někoho doporučit. Jeden **Zákazník** může doporučit více **Zákazníků**.
- **Oficiální stránka** může být u **Zaměstnance**.

- **Oficiální stránka** je pro každého **Zaměstnance**, který ji má, pouze jedna.
- **Recenze** je vytvořena **Zákazníkem**.
  - Každá **Recenze** musí být napsána jedním **Zákazníkem**.
- **Objednávka** je na **Masáž**.
  - Každá **Objednávka** je právě na jednu **Masáž**.
- **Objednávka** je vytvořena **Zaměstnancem**.
- **Objednávka** je objednána **Zákazníkem**.
- **Masáž** dělá **Zaměstnanec**.
  - Jednu **Masáž** může dělat více **Zaměstnanců**.
- **Masáž** je právě na jednu **Objednávku**.

## ER model



- **Osoba**(jmeno, prijmeni, tel\_cislo, email)
- **Zaměstnanec**(jmeno, prijmeni, tel\_cislo, rodne\_cislo, ulice, cislo, psc)
  - FK: (jmeno, prijmeni, tel\_cislo) ⊆ Osoba(jmeno, prijmeni, tel\_cislo)
- **Oficialni\_stranka**(url, rodne\_cislo, foto, vzdelani)
  - FK: (rodne\_cislo) ⊆ Zaměstnanec(rodne\_cislo)
- **Zakaznik**(email)

- FK: (email)  $\subseteq$  Osoba(email)
- **Vip\_karta**(email, vip\_karta)
  - FK: (email)  $\subseteq$  Zakaznik(email)
- **Doporucil**(zakaznik, doporuceny)
  - FK: (zakaznik)  $\subseteq$  Zakaznik(email)
  - FK: (doporuceny)  $\subseteq$  Zakaznik(email)
- **Objednavaka**(datum\_vytvoreni, email, datum\_splneni, rodne\_cislo, nazev)
  - FK: (email)  $\subseteq$  Zakaznik(email)
  - FK: (rodne\_cislo)  $\subseteq$  Zamestnanec(rodne\_cislo)
  - FK: (nazev)  $\subseteq$  Masaz(nazev)
- **Masaz**(nazev, delka\_trvani, cena)
- **Druh**(nazev, druh\_masaze)
  - FK: (nazev)  $\subseteq$  Masaz(nazev)
- **Dela\_Masaz**(rodne\_cislo, nazev)
  - FK: (rodne\_cislo)  $\subseteq$  Zamestnanec(rodne\_cislo)
  - FK: (nazev)  $\subseteq$  Masaz(nazev)
- **Recenze**(datum, jmeno, prijmeni, tel\_cislo, komentar, hodnoceni, rodne\_cislo)
  - FK: (jmeno, prijmeni, tel\_cislo)  $\subseteq$  Zakaznik(jmeno, prijmeni, tel\_cislo)
  - FK: (rodne\_cislo)  $\subseteq$  Zamestnanec(rodne\_cislo)

## Relační model to SQL

- **Osoba**(id\_osoba, jmeno, prijmeni, tel\_cislo, email)
  - id\_osoba serial **primary key**
  - jmeno varchar not null
  - prijmeni varchar not null
  - tel\_cislo varchar not null
  - email varchar not null **unique**
  - **unique** (jmeno, prijmeni, tel\_cislo)
- **Zamestnanec**(id\_osoba, rodne\_cislo, ulice, cislo, psc)
  - FK: (id\_osoba)  $\subseteq$  Osoba(id\_osoba)
  - id\_osoba int **primary key**
  - rodne\_cislo (char/int)? not null **unique**
  - ulice varchar not null

- cislo int not null
- psc int not null
- **Oficialni\_stranka**(id\_oficialni\_stranka, url, id\_osoba, foto, vzdelani) ??
  - FK: (id\_osoba)  $\subseteq$  Zamestnanec(id\_osoba)
  - id\_oficialni\_stranka serial **primary key**
  - url char not null **unique**
  - id\_osoba int not null **unique** (ref Zamestnanec)
  - foto varchar not null
  - vzdelani text not null
- **Zakaznik**(id\_osoba, vip\_karta)
  - FK: (id\_osoba)  $\subseteq$  Osoba(id\_osoba)
  - Id\_osoba int **primary key**
  - vip\_karta char(10) null
- **Doporucil**(id\_zakaznik, id\_doporuceny)
  - FK: (id\_zakaznik)  $\subseteq$  Zakaznik(id\_osoba)
  - FK: (id\_doporuceny)  $\subseteq$  Zakaznik(id\_osoba)
  - Id\_zakaznik int not null (ref Zakaznik)
  - Doporiceny int not null (ref Zakaznik)
  - **Primary key**(id\_zakaznik, doporuceny)
- **Objednavka**(id\_objednavka, datum\_vytvoreni, id\_zakaznik, datum\_splneni, id\_zamestnanec, id\_masaz)
  - FK: (id\_zakaznik)  $\subseteq$  Zakaznik(id\_osoba)
  - FK: (id\_zamestnanec)  $\subseteq$  Zamestnanec(id\_zamestnanec)
  - FK: (id\_masaz)  $\subseteq$  Masaz(id\_masaz)
  - Id\_objednavka serial **primary key**
  - Id\_zakaznik int not null ref Zakaznik
  - Datum\_vytvoreni datetime not null
  - Datum\_splneni datetime not null
  - id\_zamestnanec int not null ref zamestnanec
  - Id\_masaz int not null ref Masaz
  - **Unique** (Id\_zakaznik, datum\_vytvoreni)
- **Masaz**(id\_masaz, nazev, delka\_trvani, cena)

- id\_masaz serial **primary key**
- nazev char not null **unique**
- delka\_trvani enum not null
- cena int not null
- **Druh**(id\_druh, druh\_masaze)
  - Id\_druh serial **primary key**
  - Druh\_masaz varchar not null
- Druh\_masaze(id\_masaz, id\_druh)
  - Id\_masaz int not null ref(masaz)
  - Id\_druh int not null ref(druh)
  - **Primary key** (id\_masaz, id\_druh)
- **Zamestnanec\_Dela\_Masaz**(id\_osoba, id\_masaz)
  - FK: (id\_osoba)  $\subseteq$  Zamestnanec(id\_osoba)
  - FK: (id\_masaz)  $\subseteq$  Masaz(id\_masaz)
  - Id\_osoba int not null (ref Zamestnanec)
  - Id\_masaz int not null (ref Masaz)
  - **Primary key**(id\_osoba, id\_masaz)
- **Recenze**(id\_recenze, id\_zakaznik, datum, komentar, hodnoceni, id\_zamestnanec)
  - FK: (id\_zakaznik)  $\subseteq$  Zakaznik(id\_osoba)
  - FK: (id\_zamestnanec)  $\subseteq$  Zamestnanec(id\_osoba)
  - Id\_recenze serial **primary key**
  - Id\_zakaznik int not null (ref Zakaznik)
  - Datum datetime not null
  - Komentar char not null
  - Hodnoceni int not null
  - Id\_zamestnanec int ( ref Zamestnanec)
  - **Unique** (Id\_zakaznik, datum)

## Vytvoření tabulek

Osoba

```
DROP TABLE IF EXISTS Osoba CASCADE;
CREATE TABLE Osoba (
    id_osoba SERIAL PRIMARY KEY,
    jmeno VARCHAR(60) NOT NULL,
    prijmeni VARCHAR(60) NOT NULL,
```

```

tel_cislo VARCHAR(20) NOT NULL,
email VARCHAR(100) NOT NULL UNIQUE,

CONSTRAINT chk_email_format

        CHECK (email ~* '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'),

CONSTRAINT chk_tel_cislo_format

        CHECK (tel_cislo ~ '^\(?\d[\d-]*)$'),
        UNIQUE (jmeno, prijmeni, tel_cislo)
);

```

#### Zamestnanec

```

DROP TABLE IF EXISTS Zamestnanec CASCADE;
CREATE TABLE Zamestnanec (
        id_osoba INT PRIMARY KEY,
        rodne_cislo VARCHAR(15) NOT NULL UNIQUE,
        ulice VARCHAR(60) NOT NULL,
        cislo INT NOT NULL,
        psc INT NOT NULL,
        FOREIGN KEY (id_osoba) REFERENCES Osoba(id_osoba)
        ON UPDATE CASCADE ON DELETE CASCADE
);

```

#### Officialni\_stranka

```

DROP TABLE IF EXISTS Officialni_stranka CASCADE;
CREATE TABLE Officialni_stranka (
        id_oficialni_stranka SERIAL PRIMARY KEY,
        url VARCHAR(250) NOT NULL UNIQUE,
        id_osoba INT NOT NULL unique,
        foto VARCHAR(100) NOT NULL,
        vzdeleni TEXT NOT NULL,

        CHECK (url ~* '^https?://'),
        CHECK (foto ~* '\.(jpg|png)$'),

        FOREIGN KEY (id_osoba) REFERENCES Zamestnanec(id_osoba)
        ON UPDATE CASCADE ON DELETE CASCADE
);

```

#### Zakaznik

```

DROP TABLE IF EXISTS Zakaznik CASCADE;
CREATE TABLE Zakaznik (
        id_osoba INT PRIMARY KEY,
        vip_karta VARCHAR(10),
        FOREIGN KEY (id_osoba) REFERENCES Osoba(id_osoba)
        ON UPDATE CASCADE ON DELETE CASCADE
);

```

### Doporucil

```
DROP TABLE IF EXISTS Doporucil CASCADE;
CREATE TABLE Doporucil (
    id_zakaznik INT NOT NULL,
    id_doporuceny INT NOT NULL,
    PRIMARY KEY (id_zakaznik, id_doporuceny),
    FOREIGN KEY (id_zakaznik) REFERENCES Zakaznik(id_osoba)
        ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (id_doporuceny) REFERENCES Zakaznik(id_osoba)
        ON UPDATE CASCADE ON DELETE SET NULL
-- Pokud je id_zakaznik odstraněn, je nám jedno, zda někoho doporučil, můžeme odstranit id_doporuceny
-- Pokud je id_doporuceny odstraněn, je důležité vědět, že id_zakaznik někoho doporučil.
);
```

### Masaz

```
DROP TABLE IF EXISTS Masaz CASCADE;
CREATE TYPE masaz_delka_enum AS ENUM ('30', '45', '60', '90'); -- enum
CREATE TABLE Masaz (
    id_masaz SERIAL PRIMARY KEY,
    nazev VARCHAR(100) NOT NULL UNIQUE,
    delka_trvani masaz_delka_enum NOT NULL,
    cena INT NOT NULL
);
```

### Objednavka

```
DROP TABLE IF EXISTS Objednavka CASCADE;
CREATE TABLE Objednavka (
    id_objednavka SERIAL PRIMARY KEY,
    id_zakaznik INT NOT NULL,
    datum_vytvoreni TIMESTAMP NOT NULL,
    datum_splneni TIMESTAMP NOT NULL,
    id_zamestnanec INT NOT NULL,
    id_masaz INT NOT NULL,

    UNIQUE (id_zakaznik, datum_vytvoreni),
    CHECK (datum_vytvoreni <= datum_splneni),

    FOREIGN KEY (id_zakaznik) REFERENCES Zakaznik(id_osoba)
        ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (id_zamestnanec) REFERENCES Zamestnanec(id_osoba)
        ON UPDATE CASCADE ON DELETE SET NULL,
    FOREIGN KEY (id_masaz) REFERENCES Masaz(id_masaz)
        ON UPDATE CASCADE ON DELETE SET NULL
-- Pokud je smazán id_zamestnanec, id_masaz není potřeba mazat celý záznam s tou objednavkou.
-- Pokud je smazán id_zakaznik, lze smazat i jeho objednávku.
);
```



#### Druh

```
DROP TABLE IF EXISTS Druh CASCADE;
CREATE TABLE Druh (
    id_druh SERIAL PRIMARY KEY,
    druh_masaz VARCHAR(100) NOT NULL
);
```

#### Druh\_masaze

```
DROP TABLE IF EXISTS Druh_masaze CASCADE;
CREATE TABLE Druh_masaze (
    id_masaz INT NOT NULL,
    id_druh INT NOT NULL,
    PRIMARY KEY (id_masaz, id_druh),
    FOREIGN KEY (id_masaz) REFERENCES Masaz(id_masaz)
        ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (id_druh) REFERENCES Druh(id_druh)
        ON UPDATE CASCADE ON DELETE CASCADE
);
```

#### Zamestnanec\_Dela\_Masaz

```
DROP TABLE IF EXISTS Zamestnanec_Dela_Masaz CASCADE;
CREATE TABLE Zamestnanec_Dela_Masaz (
    id_osoba INT NOT NULL,
    id_masaz INT NOT NULL,
    PRIMARY KEY (id_osoba, id_masaz),

    FOREIGN KEY (id_osoba) REFERENCES Zamestnanec(id_osoba)
        ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (id_masaz) REFERENCES Masaz(id_masaz)
        ON UPDATE CASCADE ON DELETE CASCADE
);
```

#### Recenze

```
DROP TABLE IF EXISTS Recenze CASCADE;
CREATE TABLE Recenze (
    id_recenze SERIAL PRIMARY KEY,
    id_zakaznik INT NOT NULL,
    datum TIMESTAMP NOT NULL,
    komentar VARCHAR(255) NOT NULL,
    hodnoceni INT NOT NULL,
    id_zamestnanec INT,
    UNIQUE (id_zakaznik, datum),
    FOREIGN KEY (id_zakaznik) REFERENCES Zakaznik(id_osoba)
        ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (id_zamestnanec) REFERENCES Zamestnanec(id_osoba)
        ON UPDATE CASCADE ON DELETE SET NULL
);
```

Pokud je smazán id\_zamestnanec, nechceme u něj mazat všechny recenze.

Pokud je smazán id\_zakaznik, můžeme smazat všechny jeho recenze.

## Dotazy

Při novém vytvoření tabulek se u vás některé výsledky mohou lišit kvůli generování dat.

Vnější spojení tabulek

Zobrazit všechny zaměstnance a jejich oficiální stránky – konkrétně číslo zaměstnance, URL a foto.

```
select id_osoba as cislo_zamestnance, url, foto  
from zamestnanec left join oficialni_stranka using (id_osoba);
```

	cislo_zamestnance	url	foto
1	1	https://maser-janek.cz	janek.jpg
2	2	https://admin-novakova.cz	novakova.png
3	3	https://info-kratochvil.cz	kratochvil.jpg
4	4	https://masaze-petra.cz	petra.png
5	5	<null>	<null>
6	6	<null>	<null>
7	7	https://maser-martina.cz	martina.jpg
8	8	<null>	<null>
9	9	<null>	<null>
10	10	https://touchofrelax.cz	relax.jpg
11	41	<null>	<null>
12	42	<null>	<null>
13	43	<null>	<null>

Vnitřní spojení tabulek

"Zobrazit zaměstnance, kteří mají oficiální stránky – konkrétně jejich číslo zaměstnance, URL a vzdělání.

```
select id_osoba as cislo_zamestnance, url, vzdelani from zamestnanec  
inner join oficialni_stranka using(id_osoba);
```

	cislo_zamestnance	url	vzdelani
1	1	https://maser-janek.cz	Certifikovaný masér, kurz rekondiční a sportovní masáže.
2	2	https://admin-novakova.cz	Střední odborná škola - administrativní směr.
3	3	https://info-kratochvil.cz	Ekonomická škola, praxe v recepci.
4	4	https://masaze-petra.cz	Zdravotní škola, kurz lymfatické a relaxační masáže.
5	7	https://maser-martina.cz	Certifikace v oblasti reflexní a thajské masáže.
6	10	https://touchofrelax.cz	Wellness akademie, aromaterapie a celotělová masáž.

Podmínku na data

Zobrazit nejdražší masáže – tedy ty, jejichž cena je 1000 a více.

```
select * from masaz where cena >= 1000;
```

	id_masaz	nazev	delka_trvani	cena
1	3	Aromaterapeutická masáž	90	1200
2	5	Havajská masáž Lomi Lomi	60	1000
3	6	Masáž lávovými kameny	90	1300

### Agregaci a podmínku na hodnotu agregační funkce

Zobrazit jména zákazníků a počet jejich objednávek, seřazené od nejaktivnějších. Zákazníci jsou považováni za nejaktivnější, pokud mají více než 200 objednávek.

```
select o.jmeno, count(*) as pocet_objednavek
from objednavka ob
    join zakaznik z on ob.id_zakaznik = z.id_osoba
    join osoba o on z.id_osoba = o.id_osoba
group by o.jmeno
having count(*) > 200
order by pocet_objednavek desc;
```

(Tady výsledek u vás může být jiný kvůli generování pomocí scriptu)

	jmeno	pocet_objednavek
1	Tomas	304
2	Julianne	209
3	Audy	205
4	Lucie	204
5	Barnett	203

### Řazení a stránkování

Zobrazit osoby, jejichž id je mezi 200 a 300, seřazené abecedně podle jména. Výpis bude omezen na 10 výsledků s přeskočením prvních 20 záznamů.

```
select * from osoba
    where id_osoba between 200 and 300
    order by jmeno asc
    limit 10 offset 20;
```

	id_osoba	jmeno	prijmeni	tel_cislo	email
1	226	Damien	Vaar	+420 694 852 231	dvaar55@blinklist.com
2	202	Daniella	Warret	+420 734 097 098	dwarret4h@smh.com.au
3	235	Danila	Coke	+420 747 026 034	dcoke5e@un.org
4	208	Desirae	Windebank	+420 657 194 349	dwindebank4n@google.it
5	274	De witt	Pitney	+420 794 587 881	dpitney6h@vk.com
6	201	Dorian	Duxbarry	+420 773 552 979	dduxbarry4g@un.org
7	233	Eliot	Jaszczak	+420 611 571 705	ejaszczak5c@imgur.com
8	245	Eliot	Ridsdell	+420 747 193 844	eridsdell5o@dion.ne.jp
9	216	Enrique	Khomishin	+420 747 620 958	ekhomishin4v@bluehost.com
10	264	Eustacia	Vouls	+420 657 858 884	evouls67@imgur.com

### Množinové operace

Vypsát jen ty masáže, které jsou zároveň prováděné a mají přiřazený druh masáže.

```
select * from masaz where id_masaz in (
  select id_masaz
  from zamestnanec_dela_masaz
  intersect
  SELECT id_masaz
  FROM druh_masaze );
```

	id_masaz	nazev	delka_trvani	cena
1	3	Aromaterapeutická masáž	90	1200
2	5	Havajská masáž Lomi Lomi	60	1000
3	4	Klasická masáž	30	500
4	6	Masáž lávovými kameny	90	1300
5	2	Sportovní masáž	45	800
6	1	Relaxační masáž	60	900
7	8	Anticelulitidní masáž	60	950

### Vnořený SELECT

Jména a Příjmení zákazníků, kteří někoho doporučili, ale sami nebyli doporučeni.

```
select
  o1.jmeno as doporučil_jmeno,
  o1.prijmeni as doporučil_prijmeni,
  o2.jmeno as doporučený_jmeno,
  o2.prijmeni as doporučený_prijmeni
from doporučil d
  join zakaznik z1 on d.id_zakaznik = z1.id_osoba
  join osoba o1 on z1.id_osoba = o1.id_osoba
  join zakaznik z2 on d.id_doporučený = z2.id_osoba
  join osoba o2 on z2.id_osoba = o2.id_osoba
where d.id_zakaznik not in (
  select id_doporučený
  from doporučil)
```

);

	<input type="text" value="doporucil_jmeno"/>	<input type="text" value="doporucil_prijmeni"/>	<input type="text" value="doporuceny_jmeno"/>	<input type="text" value="doporuceny_prijmeni"/>
1	Ardelle	Loveguard	Eb	Dunston
2	Friedrich	Vaz	Rodrique	Escot
3	Willie	Dundon	Karia	O'Fielly
4	Willie	Dundon	Barnett	Dugan
5	Logan	Codd	Kenn	Wayman
6	Claudianus	Southorn	Yvon	Yurygyn
7	Claudianus	Southorn	Alwyn	Hyland

## Diagram

