

## **Databázové systémy**

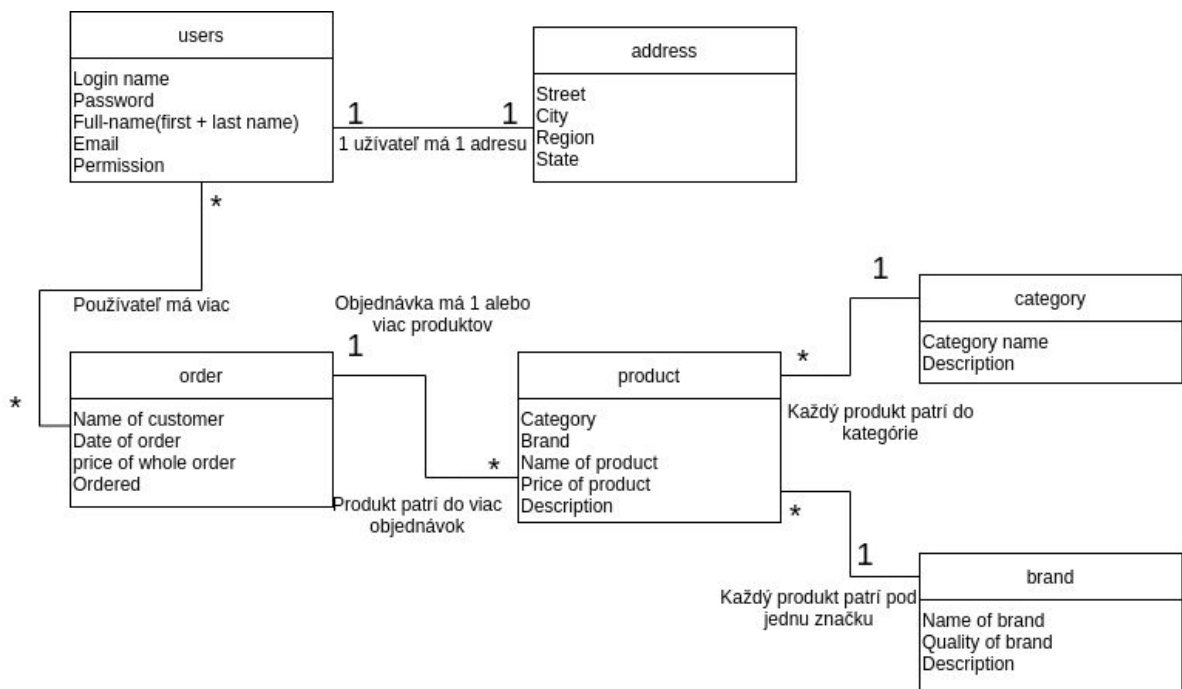
### **Projekt - ESHOP**

Popis:

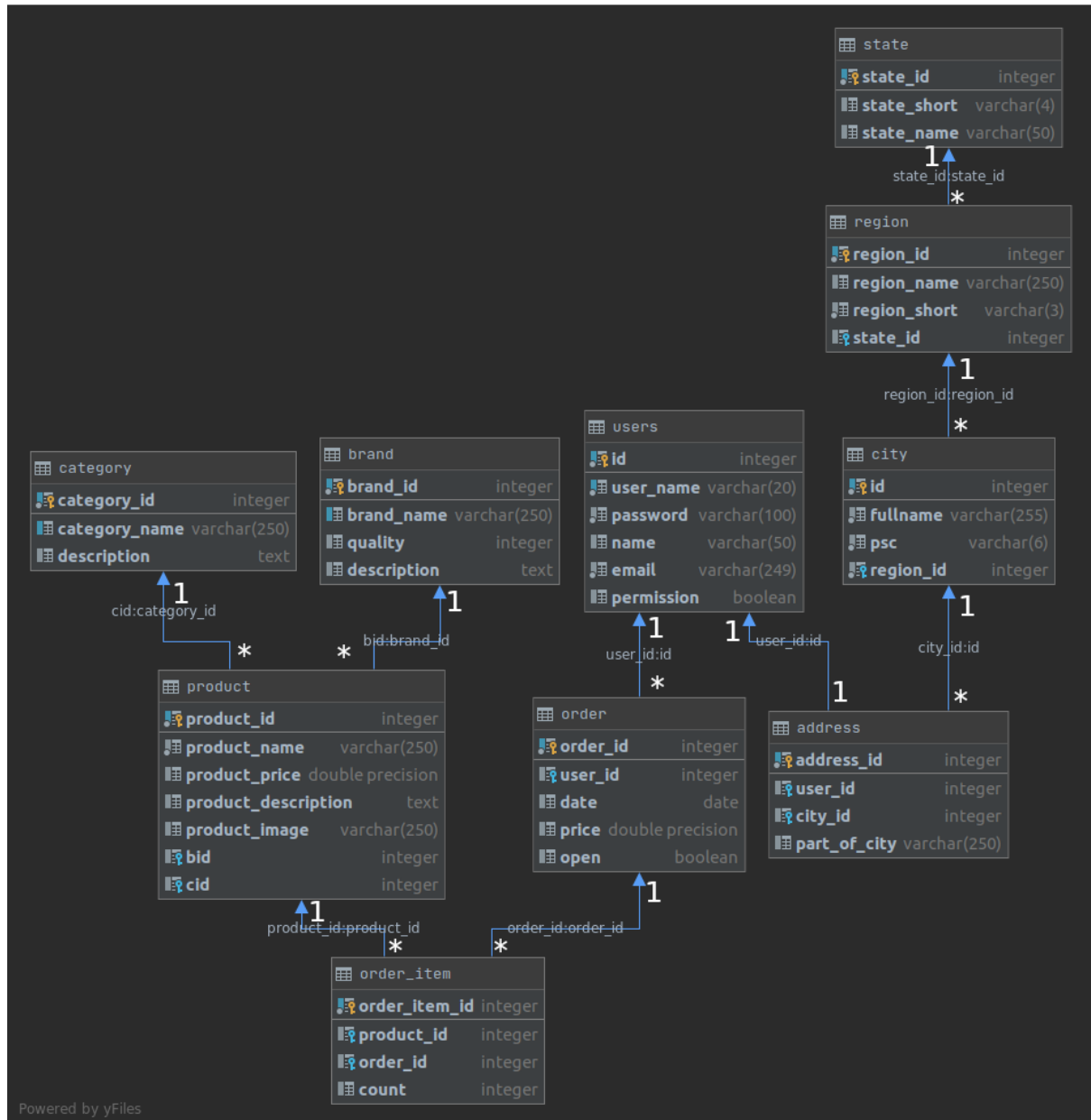
Počas projektu pracujem na webovej aplikácii eshopu. Bude možné sa registrovať, čím sa zaručia informácie o kupujúcom. Admin(správca) bude môcť pridávať a meniť údaje o produktoch, ktoré budú v eshope. Užívateľ bude môcť pridávať produkty do košíka a následne vytvoriť objednávku. Všetky objednávky s informáciami o produktoch a o osobe, ktorá objednávku vytvorila budú prístupne administrátorovi v jeho časti webu(dashboard). Aplikáciu implementujem pomocou frameworku Flask a databázového systému PostgreSQL.

Dátové modely:

## Logický model



## Fyzický model



## Špecifikácia

### **Databáza**

#### Zákazník

Názov tabuľky: users

Predstavuje zákazníkov. Pomocou tejto tabuľky sa ukladajú základné informácie o kupujúcom. Nový záznam sa vytvára pri registrácii.

Možno dobre spomenúť, že stlpec password je zahashovaný pomocou sha256.

#### Produkt

Názov tabuľky: product

Predstavuje produkty. Do tejto tabuľky sa teda ukladajú produkty, ich informácie. Nový záznam vytvára administrátor v sekcii dashboard. A taktiež ho v tejto sekcii aj môže meniť, či vymazať.

#### Objednávka

Názov tabuľky: order

Predstavuje objednávky. Záznam sa vytvára pri prvom pridaní produktu do košíka.

#### Objednaný produkt

Názov tabuľky: order\_item

Predstavuje objednaný produkt patriaci do objednávky vyššie. Záznam sa vytvára pri transakcii vytvárania objednávky(ak ešte neexistuje)

#### Značka

Názov tabuľky: brand

Predstavuje značku. Záznam sa vytvára v administrácii(dashboard). Bude slúžiť na filtráciu kvality produktov či filtráciu na základe značky.

#### Kategória

Názov tabuľky: category

Predstavuje objednaný kategóriu produktu. Taktiež sa vytvára v administrácii a jej hlavný zmysel je vo filtrovaní produktov podľa záujmu kupujúceho.

#### Adresa

Adresa je špecifický prvok/prvky tabuľky. Je rozdelená do viacerých tabuliek aby nevznikala zbytočná redundancia.

Názov tabuľky: address

Do tejto tabuľky sa ukladá ulica zákazníka a id mesta, ktoré pri registrácii zákazník vybral pomocou PSČ. Záznam vzniká pri registrácii.

Názov tabuľky: city

V tabuľke je uložené meno mesta, PSČ a fk regiónu, do ktorého mesto patrí.

Názov tabuľky: region

V tabuľke je uložené meno regiónu/kraju a id štátu (v mojom prípade som štát nechal pre zložitosť s PSČ len jeden a to štát, tabuľka je ale pripravená na pridávanie iných štátov)

Názov tabuľky: state

V tabuľke je uložené meno štátu a jeho skratka.

## **Scenáre:**

### **Scenár č. 1:**

Registrácia a prihlasovanie zákazníkov.

V prvom scenári som vytvoril tabuľku users, kde som uložil základné údaje o užívateľoch ako napríklad prihlasovacie meno a heslo. Návštevník webu je schopný sa registrovať (vytvoriť zápis do tabuľky) a následne sa prihlásiť (skontrolovať či existuje zápis v tabuľke). Pred insertom do tabuľky sa kontroluje, či vybraný user name už v tabuľke nie je, respektíve to databáza kontroluje za mňa a ja už len chytím výnimku o chybe kvôli argumentu UNIQUE. Pri registrácii sa vytvára aj adresa, ktorá je spomenutá v špecifikácii.

### **Scenár č. 2, 3, 4:**

Dashboard - administrácia.

Tento scenár bol rozdelený na viac častí, podľa zadania. Počas procesu som to všetko vložil do administrácie a tak to pokladám za jednotný scenár. V tomto scenári admin spravuje viacmenej všetky tabuľky (zatiaľ nemôže pridať/vytvoriť ani meniť objednávku). V tejto sekcii dashboard má prístup k pridaniu (INSERT), aktualizovaniu (UPDATE) a mazaniu (DELETE) záznamov v tabuľkách. Môže meniť, odstraňovať a pridávať značky, kategórie, produkty či upravovať profily, informácie o zákazníkoch. Tiež má prístup k zmene ich hesla. Sekcia dashboard je vlastne sekcia tabuliek. Každá položka menu vypíše jednu tabuľku (niektoré tabuľky, ktoré obsahujú FK vypíšu údaje aj z druhej tabuľky. Takouto sekciou je napríklad výpis užívateľov, kde sa pri výpise zobrazí aj mesto z ich adresy, toto mesto je referované cez tabuľku address, ktorá má FK usera - celkovo sa využívajú 3 tabuľky)

V tomto scenári využívam netriviálne dopyty v dashboard sekcii na základne štatistiky. Jednou z nich je výpis nadpriemerných zákazníkov, t.j. zákazník, ktorý utratil nadpriemernú cenu v eshope. V tomto dopyte využívam agregáčné funkcie COUNT, SUM, AVG a klauzuly INNER JOIN - pre výpis mena zákazníka, GROUP - na rozdelenie zákazníkov, HAVING - na odfiltrovanie tých, ktorí majú celkovú sumu väčšiu ako je priemer ostatných zákazníkov, ORDER BY - na usporiadanie od toho najlepšieho. Klauzulu WHERE som okrem iného použil aj pri štatistike výberu dní, ktoré mali zisk väčší ako 15 000. Tam som vďaka WHERE vybral len tie objednávky, ktoré už boli uzavreté. Pri tomto dopyte som taktiež využil klauzuly GROUP BY - roztriedenie podľa dátumu, HAVING - pre sumu ceny viac ako 15000 a ORDER BY - pre zoradenie od najziskovejšieho dňa. Po naplnení tabuliek som sa rozhodol vyčleniť tieto štatistiky na prvých 50 riadkov.

Scenár 5 => ešte nie je hotový na 100%

Filtrácia produktov

V tomto scenári sa v hlavnom menu zobrazia všetky produkty. Užívateľ si však pomocou menu môže vybrať, ktoré produkty chce vidieť. Bude si môcť vybrať na základe kategórie, značky, ceny alebo vyfiltrovať produkt podľa mena - nájsť produkt podľa časti názvu. Napríklad ak zadá phone, vypíšu sa mu všetky produkty, ktorých názov obsahuje časť "phone"

### **Napĺňanie tabuliek**

Tabuľky som naplnil pomocou fadera (hlavne používateľov, ale aj pár tisíc produktov). Pre naplnenie tabuľky produktov som si stiahol dataset, ktorý je priložený v zdrojových súboroch. Z tohto datasetu som si vybral iba dáta, ktoré sa mi hodili. Na vytváranie objednávok som už použil dáta, ktoré boli v ostatných tabuľkách a len som ich náhodne generoval podľa dátumu. Väčšina z týchto funkcií naplnenia tabuliek sú ešte v zdrojovom kóde pre prípad nutnosti pridania ďalších dát.

SQL a XML súbory - export databázy

<https://drive.google.com/open?id=1zocCHGcM2qBwdabHbcljc7vXRWb60qJv>