

**Project Assignment 1 – Database Design**

Bezpečnost databázových systémů

Vypracovali: Martin Nečas, Natálie Klimková

# **Úvod**

V našem projektu jsme se snažili vytvořit databázi potenciální aplikace pro rozvoz jídla. Proč? Protože máme rádi jídlo. A co je lepší, než když vám ho někdo uvaří a doveze až domů. Při plánování projektu jsme vycházeli z již existujících aplikací Dáme jídlo a Wolt, díky tomu jsme si zjednodušili fázi vymýšlení jednotlivých tabulek, které by měla databáze obsahovat. Databáze funguje docela jednoduše. Našim úkolem je dostat jídlo k uživateli. K tomu potřebujeme poměrně dost informací, jako třeba adresu uživatele, kontakty pro doručení a co hlavně, jaké jídlo si uživatel vybral a v jaké restauraci. Stejně jako u adresy, tak i u samotné objednávky má uživatel možnost napsat vlastní požadavky. Jakmile tyto informace zjistíme, můžeme přejít k části, která vyžaduje trochu logistických schopností. My se toho ale nebojíme, protože tabulky nám pomůžou. Nejen, že se z nich dozvíme, který řidič má zrovna jakou směnu, ale třeba i to, jakým jezdí autem. Všechny zásadní informace potřebné k doručení jídla se spojí v tabulce delivery, která mimo jiné ukazuje třeba čas doručení. U toho ale nekončíme! Zpětná vazba je pro nás velice důležitá, a proto jsme se rozhodli dát uživatelům možnost vyjádřit své dojmy a připomínky nejen u samotných jídel, ale i u celé restaurace. Samozřejmostí je také hodnocení řidičů, kteří rádi uslyší pozitivní zprávy, a ty negativní je snad někam v životě posunou.

# **Vysvětlení jednotlivých tabulek:**

Pro vše, co obsahuje \_id jsme zvolili typ INT. Práce s čísly je rychlejší a jednodušší než práce s řetězcem.

**Obsah obrázku text

Popis byl vytvořen automatickyperson:**

Tato tabulka je určena pro zadávání informací o uživatelích nebo řidičích. Pro first\_name, last\_name, password a username jsme zvolili typ VARCHAR, protože zde očekáváme, že uživatel zadá znakový řetězec. U date\_of\_birth je typ DATE, protože očekávaným vstupem je datum.

**Obsah obrázku text

Popis byl vytvořen automatickycontact:**

Tabulka bude obsahovat kontakty uživatelů a řidičů z tabulky person. Pro type a value jsme zvolili VARCHAR, protože zde očekáváme vložení čísla/mailu a označení daného kontaktu.

**Obsah obrázku text

Popis byl vytvořen automatickyperson\_type:**

Tabulka dělí tabulku person na řidiče a uživatele. Pro rozdělení jsme zvolili typ VARCHAR.

Obsah obrázku text

Popis byl vytvořen automaticky**shift, driver\_has\_shift:**

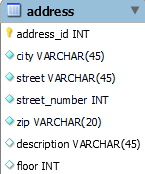
Obsah obrázku text

Popis byl vytvořen automatickyTyto tabulky používáme na přehled směn jednotlivých řidičů. V shift tabulce jsou vypsané jen směny, proto jsme pro start\_time a end\_time zvolili typ DATETIME. Tabulka driver\_has\_shift přiřadí řidiče k dané směně.

**Obsah obrázku text

Popis byl vytvořen automatickycar:**

Tabulka se používá na evidenci aut. Pro plate, type a brand jsme zvolili VARCHAR, protože očekáváme na vstupu znakový řetězec.

**address:**

Tabulka zaznamenává jednotlivé údaje, které souvisí s adresou uživatele. Pro city, street, zip a description jsme zvolili typ VARCHAR, protože očekáváme na vstupu znakový řetězec. V případě street\_number a floor jsme zvolili INT, protože všude očekáváme jako vstup číslo.

**user\_has\_address:**

**Text

Description automatically generated**Tabulka spojuje adresy s uživateli. Type je požito pro pojmenování typu adresy (doma, práce, …), proto je použito VARCHAR.

Obsah obrázku text

Popis byl vytvořen automaticky**restaurant:**

Tuto tabulku používáme pro jednotlivé restaurace. Name je VARCHAR, protože očekáváme na vstupu znakový řetězec (jméno restaurace).

**Obsah obrázku text

Popis byl vytvořen automatickycuisine:**

Tabulka je spojená s restaurací a určuje typ kuchyně, kterou restaurace vaří. U cuisine\_name je VARCHAR, protože očekáváme na vstupu znakový řetězec (název kuchyně).

**dish:**

**Text

Description automatically generated**Tabulka je věnovaná hlavnímu důvodu našeho projektu, a to jednotlivým jídlům. U name je VARCHAR, protože očekáváme na vstupu znakový řetězec (název jídla).

Obsah obrázku text

Popis byl vytvořen automaticky**dish\_has\_restaurant:**

Tabulka spojující jednotlivá jídla a restaurace. Najdeme v ní taky ceny, pro které používáme typ FLOAT.

**Obsah obrázku text

Popis byl vytvořen automatickydelivery:**

Tabulka delivery se týká samotné objednávky. Zjistíme zde, který řidič komu objednávku doveze a také můžeme zjistit čas příjezdu (jde o čas a datum, proto DATETIME) a cenu doručení (INT, protože počítáme s číselným vstupem).

Obsah obrázku text

Popis byl vytvořen automaticky**delivery\_has\_dish\_has\_restaurant:**

Tabulka spojující tabulku delivery, dish a restaurant. Plus má možnost uživatel dopsat poznámky k jídlu (například „bez koriandru“). Pro user\_requirements jsme zvolili typ TEXT s délkou 200, kdyby těch požadavků bylo hodně.

Obsah obrázku text

Popis byl vytvořen automaticky

**review:**

Poslední ale ne méně důležitá je tabulka review. Dělá přesně to, co by od ní člověk očekával. Díky ní se dozvíme, jestli je vybrané jídlo přesolené, jestli je restaurace tak dobrá, jak se o ní říká, nebo jaké chování má ten hezký řidič, co nám jídlo přivezl. Rating je typ TINYINT, protože nám to stačí. Text je typu TEXT s délkou 500, aby se člověk mohl pořádně rozepsat.

**Github repozitáře**

<https://github.com/mnecas/bds-project>

<https://github.com/natalieklimek/bds-db-design>

**3. normálová forma:** Myslíme, že databáze je v třetí normálové formě, protože všechny sloupce v jednotlivých tabulkách jsou závislé na primárních klíčích, a ne na jiném jiném sloupci.

**Celý návrh** databáze vypadá takto:

