UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

*Informační systém květinářství*

Seminární práce z předmětu Databázové systémy 1

František Boura

V Pardubicích dne 25. 9. 2023

Obsah

[Analýza 10](#_Toc148218118)

[Kompletní scénář 10](#_Toc148218119)

[Strukturální pravidla 10](#_Toc148218120)

[Procedurální pravidla 10](#_Toc148218121)

[Jednoduchá analýza vyplývající ze scénáře 11](#_Toc148218122)

[CRUD analýza 12](#_Toc148218123)

[Diskuze smyček 12](#_Toc148218124)

[Integritní omezení 12](#_Toc148218125)

[ERD 8](#_Toc148218126)

[ERDish věty 8](#_Toc148218127)

[Relační model dat 8](#_Toc148218128)

[Postup transformace a mapování 8](#_Toc148218129)

[Fyzický model dat 9](#_Toc148218130)

[Výčet SQL dotazů 10](#_Toc148218131)

[Závěr 14](#_Toc148218132)

[Přílohy 15](#_Toc148218133)

Analýza

## Kompletní scénář

Firma "Květinka" prodává **květiny** a různé zahradnické **potřeby** prostřednictvím internetového obchodu. Internetový obchod umožňuje registraci **uživatele**, který může podávat **objednávky**. V objednávce může uživatel zvolit doručovací **adresu**, **typ platby**, **způsob doručení** a samozřejmě si může vybrat **zboží**. Zboží se dělí na výše zmíněné **květiny** a **potřeby**. Veškeré zboží firmě dodávají **dodavatelé**. Každý uživatel ve firmě má přiřazenou **roli**, od prostého zákazníka až po admina. Tato role definuje pravomoce uživatele zasahovat do systému.

## Strukturální pravidla

1. Uživatel může zastávat pouze jednu roli v danou chvíli.
2. Součástí objednávky může být pouze jeden uživatel.
3. Heslo uživatele je v databázi uloženo jako bcrypt hash.

## Procedurální pravidla

1. Každý údaj o adrese nesmí být neúplný.
2. Uživatel se musí nejprve registrovat před vytvořením objednávky.
3. Objednávku může zrušit pouze uživatel s oprávněním admin.

## Jednoduchá analýza vyplývající ze scénáře

|  |  |
| --- | --- |
| **Entita** | **Atributy** |
| Zboží | id, cena, počet, typ *kategorie\_id* |
| Zboží – Rostliny | zboží\_id, rostlina\_id, počet\_květů, délka\_stonku *barva\_id, druh\_rostliny\_id* |
| Zboží - Potřeby | zboží\_id, potřeba\_id, hmotnost |
| Kategorie | id, název |
| Druhy rostlin | id, název |
| Barvy | id, název |
| Role | id, název, úroveň\_oprávnění |
| Adresy | id, ulice, čp, psč, město, země |
| Dodavatelé | id, název, email *adresa\_id* |
| Dodané zboží | dodavatel\_id, zboží\_id, množství, cena\_za\_kus, datum\_dodání |
| Uživatelé | id, username, email, password\_hash, benefit\_body, příjmení, jméno *adresa\_id, role\_id* |
| Způsoby doručení | id, název |
| Stavy objednávky | id, název |
| Způsoby platby | id, název |
| Objednávky | id, celková\_cena, datum\_podání *doručovací\_adresa\_id, uživatel\_id, stav\_objednávky\_id, způsob\_doručení\_id, způsob\_platby\_id* |
| Objednané zboží | zboží\_id, objednávka\_id, množství, cena\_za\_kus |

1. Primární klíče jsou v uvedené tabulce vždy na prvním místě, v případě spojových tabulek M:N relací se jedná o první dva atributy.
2. Atributy zahrnují i cizí klíče zvýrazněny *kurzívou*.
3. Supertyp a subtyp:  
   Supertyp - Subtyp
4. M:N relace mezi entitami
   1. Zboží a Objednávka – tabulka Objednané zboží
   2. Dodavatelé a Zboží – tabulka Dodané zboží
5. Sloupec typ v tabulce Zboží je diskriminátor, značí exkluzivní vztah supertypu Zboží se subtypy Rostliny a Potřeby. Jedna instance Zboží tedy nemůže být zároveň instancí tabulky Rostliny a tabulky Potřeby.

## CRUD analýza

**A white background with black text

Description automatically generated**

## Diskuze smyček

Schéma neobsahuje žádné cyklické vztahy.

## Integritní omezení

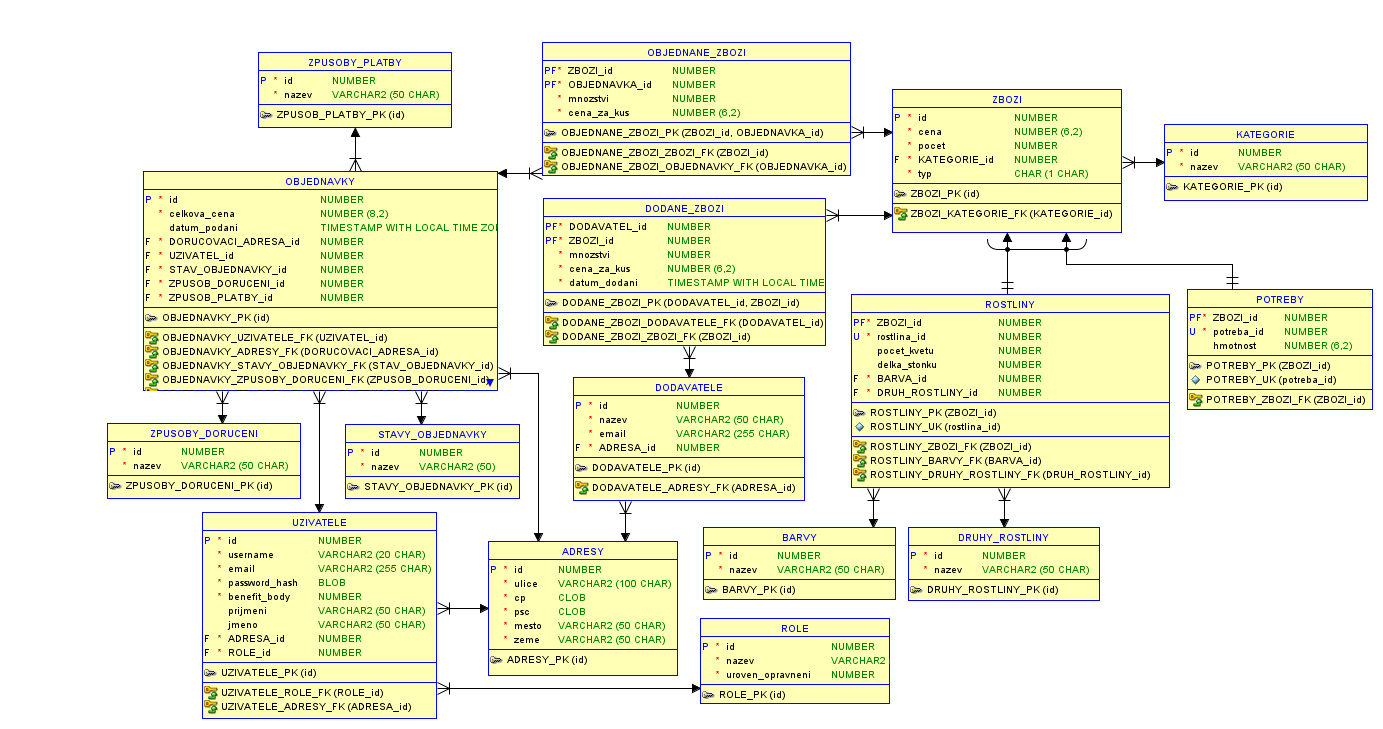
1. Každá entita má v databázi svůj unikátní identifikátor ID, který slouží jako primární klíč.
2. Hodnota sloupce úroveň\_oprávnění v tabulce ROLE může mít pouze hodnoty 0, 1, 2, žádné jiné.
3. Datum podání objednávky nemůže být dřívější než aktuální datum.
4. Cena zboží nemůže být menší než 0 a nemůže být větší než 9999.99 Kč.
5. Celková cena objednávky nemůže být menší než 0 a nemůže být větší než 999999.99 Kč.

## ERDA diagram of a computer flowchart Description automatically generated

## ERDish věty

1. Každý způsob platby může patřit jedné nebo více objednávkám.
2. Každá objednávka musí být placena právě jedním způsobem platby.
3. Každý způsob doručení může patřit jedné nebo více objednávkám.
4. Každá objednávka musí být doručena právě jedním způsobem doručení.
5. Každý stav objednávky může patřit jedné nebo více objednávkám.
6. Každá objednávka může být v právě jednom stavu objednávky.
7. Každá role může patřit jednomu nebo více uživatelům.
8. Každý uživatel musí zastávat právě jednu roli.
9. Každý uživatel může objednávat jednu nebo více objednávek.
10. Každá objednávka musí být objednána právě jedním uživatelem.
11. Každá adresa může patřit jednomu nebo více uživatelům.
12. Každý uživatel musí bydlet na právě jedné adrese.
13. Každá adresa může být součástí jedné nebo více objednávek.
14. Každá objednávka musí být doručena na právě jednu adresu.
15. Každá kategorie může vymezovat jedno nebo více zboží.
16. Každé zboží musí patřit do právě jedné kategorie.
17. Každý dodavatel může dodat jedno nebo více zboží.
18. Každé zboží musí být dodáno jedním nebo více dodavateli.
19. Každá objednávka musí zahrnovat jedno nebo více zboží.
20. Každé zboží může být součástí jedné nebo více objednávek.
21. Každá adresa může patřit jednomu nebo více dodavatelům.
22. Každý dodavatel musí sídlit na právě jedné adrese.
23. Každý druh rostliny může patřit jedné nebo více rostlinám.
24. Každá rostlina musí být právě jedním druhem rostliny.
25. Každá barva může patřit jedné nebo více rostlinám.
26. Každá rostlina musí být právě jednou barvou.

## Relační model dat



## Postup transformace a mapování

1. V nastavení entity ZBOZI jsem zvolil „Table for each entity“
2. Zvolil jsem možnost „Engineer to Relational model.“
3. Názvy tabulek a názvy constraints jsem převedl na množné číslo.
4. Do tabulky ZBOZI jsem přidal sloupec typ, který slouží jako diskriminátor, a může mít pouze hodnoty R jako Rostlina, nebo P jako Potřeba.
5. Vztahu mezi ZBOZI a ROSTLINY jsem přiřadil hodnotu R do sloupce typ a vztahu mezi ZBOZI a POTREBY jsem přiřadil hodnotu P do sloupce typ.
6. Pro každý umělý identifikátor byla vytvořena sekvence a trigger, které společně umožňují funkci autoincrement začínající na hodnotě 1.

Fyzický model dat

A close-up of a document

Description automatically generatedA screenshot of a computer

Description automatically generatedA close-up of a form

Description automatically generatedA close-up of a document

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA close-up of a document

Description automatically generatedA close-up of a document

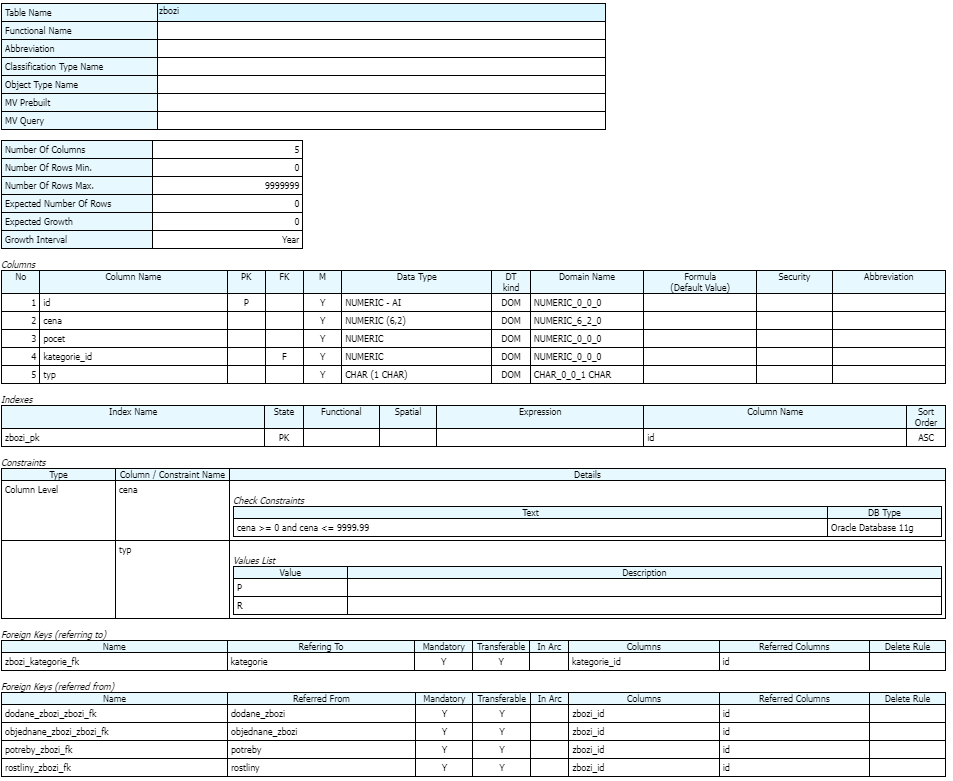
Description automatically generatedA close-up of a document

Description automatically generatedA close-up of a form

Description automatically generatedA close-up of a document

Description automatically generatedA close-up of a document

Description automatically generatedA close-up of a document

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Výčet SQL dotazů

|  |  |
| --- | --- |
| **Kategorie** | **Charekteristika kategorie příkazu** |
| D1 | pozitivní dotaz nad spojením alespoň dvou tabulek například *„vyber seznam se jmény a adresami pacientů, kteří byli vyšetřováni MUDr. Čmelákem“* |
| D2 | negativní dotaz nad spojením alespoň dvou tabulek například: *„seznam se jmény a adresami pacientů, kteří nenavštívili lékaře MUDr. Čmeláka“* |
| D3 | Vyber ty, kteří mají vztah POUZE k …například „*seznam se jmény a adresami pacientů, kteří navštívili pouze MUDr. Čmeláka ~ navštívili doktora Čmeláka a nenavštívili žádného jiného lékaře“* |
| D4 | Vyber ty, kteří/které jsou ve vztahu se všemi - dotaz s univerzální kvantifikací například: *„seznam se jmény a adresami lékařů, které navštívili VŠICHNI pacienti, kteří někdy navštívili MUDr. Čmeláka“* |
| D5 | spojení - JOIN USING |
| D6 | spojení - JOIN ON |
| D7 | spojení - NATURAL JOIN |
| D8 | spojení - CROSS JOIN |
| D9 | polospojení (vnější) - LEFT OUTER JOIN |
| D10 | polospojení (vnější) - RIGHT OUTER JOIN |
| D11 | plné (vnější) spojení - FULL (OUTER) JOIN |
| D12 | vnořený dotaz v klauzuli WHERE |
| D13 | vnořený dotaz v klauzuli FROM |
| D14 | vnořený dotaz v klauzuli SELECT |
| D15 | vztažený vnořený dotaz (EXISTS | NOT EXISTS) |
| D16 | množinové sjednocení - UNION |
| D17 | množinový rozdíl – MINUS |
| D18 | množinový průnik - INTERSECT |
| D19 | Funkce pro práci s řetězci |
| D20 | Funkce pro práci s čísly |
| D21 | Funkce pro práci s datumy |
| D22 | agregační funkce (count | sum | min | max| avg) |
| D23 | agregační funkce nad seskupenými řádky - GROUP BY (HAVING) |
| D24 | stejný dotaz ve třech různých formulacích SQL |
| D25 | všechny klauzule - SELECT FROM WHERE GROUP BY HAVING ORDER BY |
| D26 | pohled VIEW |
| D27 | dotaz nad pohledem alespoň s podmínkou |
| D28 | Příkaz pro vložení množiny řádků - INSERT bez klauzule VALUES, např. „rezervuj pacientovi č. 6 v různých časech všechny lékaře (zubař, neurolog a praktický lékař) |
| D29 | UPDATE s vnořeným SELECT příkazem |
| D30 | DELETE s vnořeným SELECT příkazem |

D1: „Seznam všech uživatelů, kteří žijí ve městě Opava.“, HK: D1, VK: D6

select jmeno, prijmeni

from uzivatele

join adresy on uzivatele.adresa\_id = adresy.id

where adresy.mesto = 'Opava';

D2: „Seznam všech rostlin, které nejsou červené.“, HK: D1, VK: D6

select rostliny.rostlina\_id, druhy\_rostliny.nazev, barvy.nazev

from rostliny

join barvy on barvy.id = rostliny.barva\_id

join druhy\_rostliny on druhy\_rostliny.id = rostliny.druh\_rostliny\_id

where barvy.nazev <> 'Červená';

D3: „Seznam všeho zboží, které nebylo nikdy součástí objednávky.", HK: D3, VK:D12

select zbozi.\*

from zbozi

where zbozi.id not in (select zbozi\_id from objednane\_zbozi);

D4: „Seznam všech uživatelů, kteří si objednali zboží s id 10.“, HK:D4

select uzivatele.jmeno, uzivatele.prijmeni, objednane\_zbozi.zbozi\_id

from objednavky

join objednane\_zbozi on objednane\_zbozi.objednavka\_id = objednavky.id

join uzivatele on uzivatele.id = objednavky.uzivatel\_id

where objednane\_zbozi.zbozi\_id = 10;

D5: „Seznam všech rostlin, které byly součástí objednávky.“, HK: D5

select rostliny.rostlina\_id, objednavky.id

from rostliny

join objednane\_zbozi using (zbozi\_id)

join objednavky on objednavky.id = objednane\_zbozi.objednavka\_id;

D6: „Seznam zákazníků, kteří nemají žádnou objednávku.“, HK:D6

select distinct uzivatele.id, uzivatele.jmeno, uzivatele.prijmeni, objednavky.id

from uzivatele

left join objednavky on uzivatele.id = objednavky.uzivatel\_id

where objednavky.uzivatel\_id is null;

D7: „Seznam zboží, které bylo součástí objednávky.“, HK: D7

select distinct zbozi.\*

from objednane\_zbozi

natural join zbozi

where zbozi.id in (select zbozi\_id from objednane\_zbozi);

D8: „Kartézský součin všech potřeb se zbožím.“, HK: D8

select zbozi.id, potreby.zbozi\_id

from zbozi

cross join potreby;

D9: „Seznam všech uživatelů a všech jejich objednávek.“, HK: D9

select uzivatele.jmeno, uzivatele.prijmeni, objednavky.\*

from uzivatele

left outer join objednavky on uzivatele.id = objednavky.uzivatel\_id;

D10: „Seznam všech objednávek se jmény uživatelů.“, HK:D10

select uzivatele.jmeno, uzivatele.prijmeni, objednavky.\*

from uzivatele

right outer join objednavky on objednavky.uzivatel\_id = uzivatele.id;

D11: „Seznam všech rostlin a všech barev.“ HK:D11

select rostliny.rostlina\_id, barvy.nazev, druhy\_rostliny.nazev

from rostliny

join druhy\_rostliny on druhy\_rostliny.id = rostliny.druh\_rostliny\_id

full join barvy on barvy.id = rostliny.barva\_id;

D12: „Seznam všech barev, které nejsou v nabídce.“ HK:D12

select barvy.nazev

from barvy

where barvy.nazev not in (select barvy.nazev from rostliny join barvy on barvy.id = rostliny.barva\_id);

D13: „Počet uživatelů, kteří si něco objednali.“ HK:D13

select count(\*)

from (select distinct uzivatele.id from uzivatele join objednavky on objednavky.uzivatel\_id = uzivatele.id);

D14: „Seznam objednávek, které mají nadprůměrnou celkovou cenu.“ HK:D14

select objednavky.id, objednavky.celkova\_cena, (select avg(objednavky.celkova\_cena) from objednavky)

from objednavky

where objednavky.celkova\_cena > (select avg(objednavky.celkova\_cena) from objednavky);

D15: „Seznam všech potřeb, které dodal dodavatel s ID 3.“ HK:D15

select potreby.\*

from potreby

where exists (select dodane\_zbozi.dodavatel\_id from dodane\_zbozi where potreby.zbozi\_id = dodane\_zbozi.zbozi\_id and dodane\_zbozi.dodavatel\_id = 3);

D16: „Spojení UNION všech barev a druhů rostlin.“ HK:D16

select nazev from barvy

union

select nazev from druhy\_rostliny;

D17: „Seznam potřeb a množství objednání kromě potřeby s názvem ‘Nůžky‘.“ HK:D17

select potreby.nazev, objednane\_zbozi.mnozstvi

from potreby

join objednane\_zbozi on objednane\_zbozi.zbozi\_id = potreby.zbozi\_id

minus

select potreby.nazev, objednane\_zbozi.mnozstvi

from potreby

join objednane\_zbozi on objednane\_zbozi.zbozi\_id = potreby.zbozi\_id

where potreby.nazev = 'Nůžky';

D18: „Seznam rostlin, které byly objednány, kromě zboží s id 10.“ HK:D18

select rostliny.zbozi\_id

from rostliny

intersect

select objednane\_zbozi.zbozi\_id

from objednane\_zbozi

where objednane\_zbozi.zbozi\_id <> 10;

D19: „Seznam názvů potřeb velkým písmem.“ HK:D19

select distinct upper(potreby.nazev)

from potreby;

D20: „Seznam objednaného zboží společně s jeho cenou vynásobenou množstvím zaokrouhlenou nahoru pro objednávku s ID 5.“ HK:D20

select objednane\_zbozi.zbozi\_id, ceil(objednane\_zbozi.mnozstvi \* objednane\_zbozi.cena\_za\_kus)

from objednane\_zbozi

where objednane\_zbozi.objednavka\_id = 5;

D21: „Seznam datumů všech dodávek zboží ve formátu ‘Month DD, YYYY’.“ HK:D21

select to\_char(dodane\_zbozi.datum\_dodani, 'Month DD, YYYY')

from dodane\_zbozi;

D22: „Počet objednávek v systému.“ HK:D22

select count(\*)

from objednavky;

D23: „Seznam ID uživatelů, kteří si někdy něco objednali, a počet jejich objednávek.“ HK:D23

select objednavky.uzivatel\_id, count(\*)

from objednavky

group by objednavky.uzivatel\_id;

D24: „Adresa s ID 2.“ HK:D24

select adresy.\*

from adresy

where adresy.id = 2;

select adresy.\*

from adresy

where adresy.id in (2);

select adresy.\*

from adresy

where adresy.id like 2;

D25: „Seznam všech jmen uživatelů a jejich počtů objednávek, kteří mají ID 1 nebo 5 a jejich počet objednávek je větší než 1 seřazený sestupně podle příjmení.“ HK:D25

select uzivatele.jmeno, uzivatele.prijmeni, count(objednavky.id)

from objednavky

join uzivatele on uzivatele.id = objednavky.uzivatel\_id

where uzivatele.id in (1, 5)

group by uzivatele.jmeno, uzivatele.prijmeni

having count(objednavky.id) > 1

order by uzivatele.prijmeni desc;

D26: „Pohled nad tabulkou objednávky.“ HK:D26

create or replace view v\_objednavky as

select \*

from objednavky;

D27: „Seznam všech objednávek z pohledu v\_objednavky vztahující se na všechny uživatele s ID větším než 5.“ HK:D27

select \*

from v\_objednavky

where uzivatel\_id > 5;

D28: „Vložení nového dodavatele.“ HK:D28

insert into dodavatele(id, nazev, email, adresa\_id)

select max(dodavatele.id) + 1, 'Pridany dodavatel', 'pridany@email.cz', 5 from dodavatele;

D29: „Update názvu města pro adresu, na které sídlí dodavatel ‘Zahrada Express’.“ HK:D29

update adresy

set mesto = 'Pardubice'

where adresy.id in (select dodavatele.adresa\_id from dodavatele where dodavatele.nazev = 'Zahrada Express');

D30: „Odstranění z tabulky rostliny se zboží ID 18.“ HK:D30

delete from rostliny

where rostliny.zbozi\_id in (select zbozi\_id from zbozi where zbozi\_id = 18);

Závěr

V této semestrální práci jsem se naučil, jak vytvořit databázi, naplnit ji daty a pracovat s ní. Přestože musela být několikrát předělána, věřím, že s trochou úprav by bylo možné ji použít v profesionálním prostředí. Jsem se svou prací spokojen.

Přílohy

1. tables.dll
   1. Soubor s DDL příkazy
2. inserts.dml
   1. Soubor s daty
3. queries.sql
   1. Soubor s výčtem příkazů
4. model.dmd + složka /model
   1. vymodelovaná databáze v aplikaci SQL Data Modeler
5. Relational.png
   1. Obrázek relačního modelu
6. Logical.png
   1. Obrázek ERD