

# Praktická maturitní zkouška

## Databázové systémy

Radek Vinařský  
C4c

Obsah:

1. Úvod
2. ER model
3. Integrita
4. Indexy
5. Pohledy
6. Triggery
7. Procedury
8. Uživatel
9. Import a export databáze
10. Závěr

# Úvod

Projekt byl vyvinut na MSSQL servru v prostředí Microsoft SQL Server Management Studio. Pro návrh jsem použil Oracle Datamodeler.

Přístupové údaje k servru, kde byla databáze zrealizovaná, jsou uloženy v souboru config.txt.

## Analýza

### Cíle

- Ukládání a správa informací o spotřebním materiálu (např. elektronice) dostupném v obchodě.
- Ukládání a správa informací o zaměstnancích obchodu.
- Ukládání a správa informací o položkách zahrnutých v každé objednávce.
- Ukládání a správa informací o zákaznících obchodu.
- Ukládání a správa informací o fyzických pobočkách obchodu a jejich umístění.
- Ukládání a správa informací o produktech ve skladu v každém obchodě.

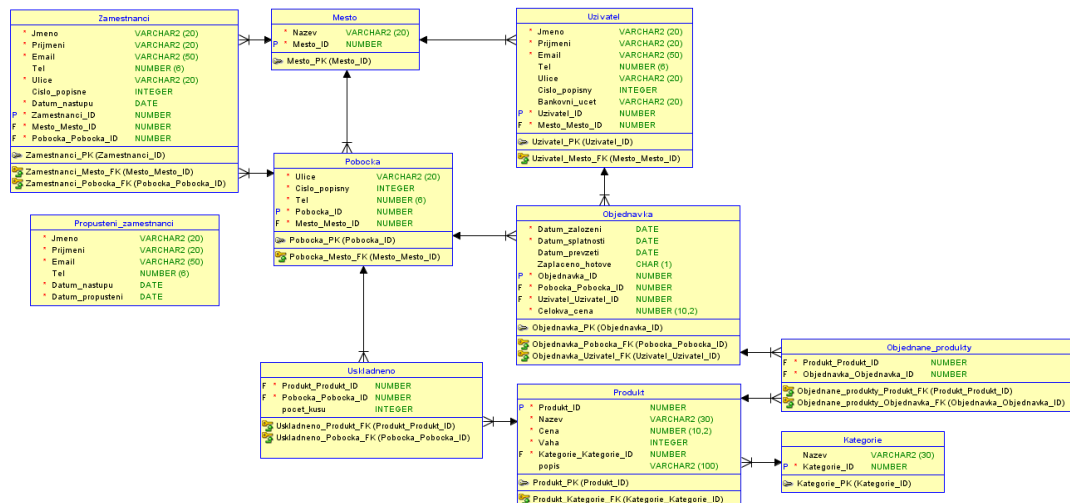
### Problémy

- Potřeba ukládat a spravovat velké množství dat o spotřebním materiálu, zaměstnancích, objednávkách, zákaznících, obchodech a zásobách.
- Potřeba efektivně získávat a analyzovat tato data, aby bylo možné činit informovaná obchodní rozhodnutí.

Databáze byla navržena tak, aby tyto problémy vyřešila vytvořením strukturovaného a organizovaného systému pro ukládání a správu dat. Různé tabulky v databázi umožňují efektivní ukládání a načítání dat a použití relací mezi tabulkami (cizí klíče) umožňuje vytváření složitých dotazů, které lze použít k analýze dat různými způsoby.

## ER model

Model lze nalézt v /img. Celkově se model skládá z 10 entit, z toho jsou 2 číselníky, město a kategorie produktu.



## Seznam entit:

### 1. Zaměstnanec

Uchovává data o všech zaměstnancích, kteří jsou momentálně zaměstnáni.

### 2. Propuštění zaměstnanci

Uchovává data o propuštěných zaměstnancích. Po delete z tabulky Zaměstnanec se data přesunou sem.

### 3. Město

Číselník s názvy měst. Lokaci chceme vědět u třech entit, zaměstnanec, pobočka, uživatel. To by byly tři atributy s názvem města, které by se často opakovali. Takto mám jen cizí klíč odkazující na název města.

### 4. Uživatel

Data o registrovaných uživateli.

### 5. Pobočka

Data o pobočkách. Pobočka slouží jako skladiště produktů a taky jako místo pro vyzvednutí objednávky.

### 6. Uskladněno

Vazební tabulka mezi pobočkou a produktem. Ukazuje aktuální stav počtu daného produktu v dané pobočce.

### 7. Objednávka

Entita, která drží data o všech objednávkách. Celková cena, uživatel, košík s produkty, pobočka vyzvednutí, datum založení, datum splatnosti, datum vyzvednutí a zda objednávka byla zaplacená hotově. Pomocí cizích klíčů spojuje data z jiných tabulek.

### 8. Objednané produkty

Vazební tabulka mezi objednávkou a produkty. Slouží jako košík pro objednávku. Umožňuje k objednávce přidat více produktů.

#### 9. Produkt

Všechny produkty, které obchod prodává.

#### 10. Kategorie

Číselník s názvy kategorií pro produkt.

## Integrita

### Entitní integrita

Každá entita obsahuje jednoduchý umělý primární klíč označen jako id.

### Doménová a referenční integrita

#### 1. Zaměstnanec

id primární klíč

Jmeno varchar(20) not null

Prijmeni varchar(20) not null

Email varchar(20) not null, musí obsahovat @

Tel numeric(9)

Ulice varchar(20)

Cislo\_popisny int

Datum\_nastupu date

Mesto\_id cizí klíč odkazující na instanci v tabulce město, not null

Pobocka\_id cizí klíč odkazující na pobočku v tabulce Pobocka, kde zaměstnanec pracuje, not null

#### 2. Propuštění zaměstnanci

id primární klíč

Jmeno varchar(20) not null

Prijmeni varchar(20) not null

Email varchar(20) not null, musí obsahovat @

Tel numeric(9)

Datum\_nastupu date, not null

Datum\_propusteni date, not null

### 3. Město

id primární klíč

Nazev varchar(20), not null

### 4. Uživatel

id primární klíč

Jmeno varchar(20) not null

Prijmeni varchar(20) not null

Email varchar(20) not null, musí obsahovat @

Tel numeric(9)

Ulice varchar(20)

Cislo\_popisny int

Mesto\_id cizí klíč odkazující na instanci v tabulce město, not null

Bankovni\_ucet varchar(50)

### 5. Pobočka

id primární klíč

Ulice varchar(20), not null

Cislo\_popisny int, not null

Tel numeric(9)

Mesto\_id cizí klíč odkazující na instanci v tabulce město, not null

### 6. Uskladněno

id primární klíč

Produkt\_id cizí klíč odkazující na instanci v tabulce produkt, not null

Pobocka\_id cizí klíč odkazující na instanci v tabulce pobočka, not null

Pocet\_kusu int, not null

## 7. Objednávka

id primární klíč

Datum\_zalozeni date, not null

Datum\_splatnosti date, not null

Datum\_prevzeti date

Zaplaceno\_hotove char(1), pokud je hodnota 1, objednávka byla zaplacená hotově, pokud je hodnota 0, objednávka byla zaplacená přes účet, který má uživatel uložený v atributu

Bankovni\_ucet, not null

## 8. Objednané produkty

id primární klíč

Produkt\_id cizí klíč odkazující na instanci v tabulce produkt, not null

Objednavka\_id cizí klíč odkazující na instanci v tabulce objednavka, not null

## 9. Produkt

id primární klíč

Nazev varchar(30), not null

Cena numeric(10,2), not null

Vaha int, not null

Popis varchar(100)

Kategorie\_id cizí klíč odkazující na instanci v tabulce kategorie, not null

## 10. Kategorie

id primární klíč

Nazev varchar(30), not null

# Indexy

Databáze má pro každou entitu indexy vytvořené pro primární klíče. Pro zrychlení vyhledávání jsem vytvořil vlastní indexy.

1. uskladneno\_index

Index na tabulku uskladěno, zahrnuje atributy pobočka\_id, produkt\_id. Index jsem vytvořil, protože v databázi je spousta where podmínek, kde kontrolovaný atributem jsou pobočka\_id, produkt\_id. Například procedúra Prevoz\_zbozi\_na\_jinou\_pobocku má několik těchto podmínek.

## 2. objednávka\_uzivatel\_index

Index na tabulku objednávka, zahrnuje atributy uzivatel\_id. Index jsem vytvořil, aby se rychleji vyhledávaly informace o uživateli, kterému objednávka patří.

## 3. objednane\_produkty\_index

Index na tabulku objednane\_produkty, zahrnuje atributy objednávka\_id, produkt\_id.

# Pohledy

## 1. Pobočka\_s\_nejvice\_produkty

Pohled vypíše lokaci pobočky s nejvíce naplněným skladem.

	id	produkt_id	pobocka_id	pocet_kusu
1	41	13	30	6
2	42	4	27	9
3	43	11	27	10
4	44	4	22	10
5	45	1	23	1
6	46	9	28	7
7	47	2	23	10
8	48	4	26	4
9	49	8	28	10
10	50	7	24	3
11	53	8	30	6
12	54	11	26	11
13	55	11	30	3
14	56	9	21	5
15	57	7	29	10
16	58	5	26	4
17	59	13	29	4
18	60	3	27	2
19	61	5	27	1

Výstup:

	id	mnozstvi_produkту	Mesto_pobocky	ulice	tel
1	27	22	Ostrava	Hallows	329262520

Na původním screenshotu je vidět pobočka 27 čtyřikrát. Jsou to řádky 2, 3, 18, 19. Součet hodnot atributu pocet\_kusu je 22, což je nejvyšší ze všech poboček.

## 2. Nejdrazsi\_objednavka

Pohled vypíše vlastníka, datum založení a datum splatnosti objednávky s nejvyšší celkovou cenou.

Data v tabulce objednavak:

	id	datum_zalozeni	datum_splatnosti	datum_prevzeti	zaplaceno_hotove	pobocka_id	uzivatel_id	celkova_cena
1	2	2022-12-30	2023-01-13	NULL	0	26	41	40800.00
2	3	2022-12-30	2023-01-13	NULL	1	27	45	71800.00
3	4	2023-01-01	2023-01-15	NULL	0	28	46	0.00

Výstup:

	id	celkova_cena	datum_zalozeni	datum_splatnosti	jmeno	prijmeni
1	3	71800.00	2022-12-30	2023-01-13	Eberhard	Rowden

## Triggery

### 1. zam\_do\_propusteny\_zam

Při propouštění zaměstnanců se zavolá delete nad tabulkou zaměstnanců. Při delete se spustí trigger, který nahradí dml příkaz. Data o zaměstnanci k propuštění přesune do tabulky propuštěných zaměstnanců s datem propuštění. Zaměstnanec se potom smaže.

### 2. vlozeni\_do\_kosiku

Při spustění dml příkazu insert se spustí trigger, který nahradí dml příkaz. Podle dat z příkazu se zjistí, ke které objednávce se má produkt přidělit a změní celkovou cenu objednávky. Když např. do tabulky vložím zápis, že k objednávce 10 přidám produkt 5, tak u objednávky 10 se updatne celková cena, přičte se cena produktu 5.

## Procedury

### 1. Prevoz\_zbozi\_na\_jinou\_pobocku

Při založení objednávky si uživatel může zvolit, na jaké pobočce si chce zboží vyzvednout. Když na té pobočce zrovna to zboží není, tahle procedura umožní přesunout zboží z jedné pobočky do jiné. Vstupní parametry procedury jsou id pobočky, kam chci zboží převést, id pobočky, odkud chci zboží vyvézt, id produktu, který chci vyvézt, množství produktů, který chci vyvézt. Pokud množství k převozu, přesáhne množství zboží uložené na pobočce, odkud se má zboží vyvézt, procedura vrátí varování 'Počet produktu k převozu přesahuje uskladněný počet'.

### 2. Zalozeni\_objednavky

Procedura zjednodušuje vytváření objednávky. Dlouhý insert příkaz se zkrátí na pár slov. Na vstupu procedury je id uživatel, kterému objednávka patří, id pobočky, kde si chce zboží vyzvednout a zda se bude platit hotově nebo přes účet. V těle procedury se vytvoří objednávka, kde datum založení je aktuální datum, datum splatnosti je aktuální datum + 14 dní.



# Uživatelé

Databáze aktuálně obsahuje pouze jednoho uživatele, tím je manager.

Manager je hlavní uživatel, který má přístup ke každému objektu v databázi. Jeho role je db\_owner.

Návrh dalších uživatelů:

Skladník - práva pouze na dml příkazy v tabulkách uskladněno a produkt a spuštění proceduru Prevoz\_zbozi\_na\_jinou\_pobocku.

Personalista - manipulování s daty v tabulkách zaměstnanec a propuštění zaměstnanci.

Zakaznicka\_podpora - práva pro manipulaci s daty v tabulkách uživatelů, objednávek a objednaných produktů.

## Import a export databáze

Neočekává se velký provoz dat v databázi, takže ideální interval zálohování je jednou za týden. Při zvětšení obsahu některých tabulek se data můžou zálohovat do vedlejších tabulek. K tomuto může hlavně dojít v tabulce uživatelů a tabulce objednane\_produkty. Dlouho neaktivní zákazníci se můžou zálohovat a 'košíky' již vyřízených objednávek se též mohou přesunout, aby se zrychlilo vyhledávání a výkon databáze.

V případě potřeby nahrát databázi na jiný server, než je školní, veškeré soubory jsou k dispozici v src složce. Složka má export struktury databáze a testovací data. Ve složce sql se nachází full export a script, který byl použit pro vytváření databáze.

## Závěr

S výsledkem práce jsem spokojený a myslím, že jsem splnil cíl práce. Jediná část práce, která asi nejvíce zaostává je zpracování platby. Nad metodou ukládání informací o způsobu platby jsem přemýšlel dlouho a rozhodl jsem se pro tu nejjednodušší metodu, kdy objednávka má atribut zaplaceno\_hotově s datovým typem bool. Když je hodnota 1, objednávka byla zaplácena hotově při vyzvedávání. Když je hodnota 0, objednávka byla zaplácena přes účet u uživatele, který je uložen v atributu Bankovni\_ucet. Takto jsem se rozhodl v zájmu zachování jednoduchosti a srozumitelnosti.

Databáze neřeší reklamace, platy zaměstnanců, samostatné sklady zboží oddělené od poboček. Žádný z těchto nedostatků by neměl způsobit větší zásahy do existující struktury.