

Kontext

Cílem seminární práce je přiblížit tvorbu databáze v praxi.

Zadání

1. Business zadání databáze IS dle vlastního výběru.
Odevzdání v dokumentu OpenOffice/ Word - název souboru RDB17S_business_jmeno_prijmeni
2. Vytvořit konceptuální model databáze dle zadání
Konceptuální model bude vytvořen v nástroji uuBML draw/MS Visio, výstupem bude čitelný obrázek - název souboru RDB17S_business_jmeno_prijmeni
3. Vytvořit logický model databáze
Logický model bude vytvořen v nástroji Oracle Data Modeler/MS Visio/podle dohody, výstupem bude čitelný obrázek - název souboru RDB17S_logmodel_jmeno_prijmeni
4. Transformace logického modelu na fyzický (platforma RDBMS - PostgreSQL, Oracle)
Fyzický model bude uložen pomocí nástroje Oracle Data Modeler/MS Visio/podle dohody, výstupem bude čitelný obrázek, - název souboru RDB17S_fyzmodel_jmeno_prijmeni
5. Vytvoření základních skriptů databáze (příkazy DDL)
Odevzdání v textovém souboru - název souboru RDB17S_DDL_jmeno_prijmeni
6. Naplnění databáze testovacími daty
Odevzdání v textovém souboru - název souboru RDB17S_testdata_jmeno_prijmeni
7. Sestavení dotazů pro systémové UC (vyhledávání, report,)
Odevzdání v textovém souboru - název souboru RDB17S_selects_jmeno_prijmeni
8. Nastínění možnosti optimalizace, ukázka postupu optimalizace u vybraného dotazu
Odevzdání v textovém souboru/ dokumentu OpenOffice - název souboru RDB17S_optimalization_jmeno_prijmeni

Rozsah práce

Rozsah práce není explicitně určen, musí být však v takové podobě, aby obsáhl všechny prvky, které byly v rámci předmětu probírány:

- Datový model minimálně s 5 tabulkami.
- V datovém modelu se musí vyskytovat všechny typy intergritních omezení relačních modelů:
primární klíče,
unikátní klíče,
not null,
check,
cizí klíče
- Minimální počet tupů vazeb mezi tabulkami:
1x vazba na tabulku číselníku,
1x vazba typu 1:N,
1x vazba typu M:N
- Testovací data - data musí být náležitě vložena (např. správná práce s datumy)
- Dotazy - použití všech druhů joinů, korelovaný i nekorelovaný poddotaz, seskupení (GROUP BY, HAVING), různé WHERE podmínky (s různými operátory jako je BETWEEN, LIKE), třídění (ORDER BY)
- Optimalizace musí být včetně důkazu úspěšnosti (doba běhu, exekuční plán, ...)

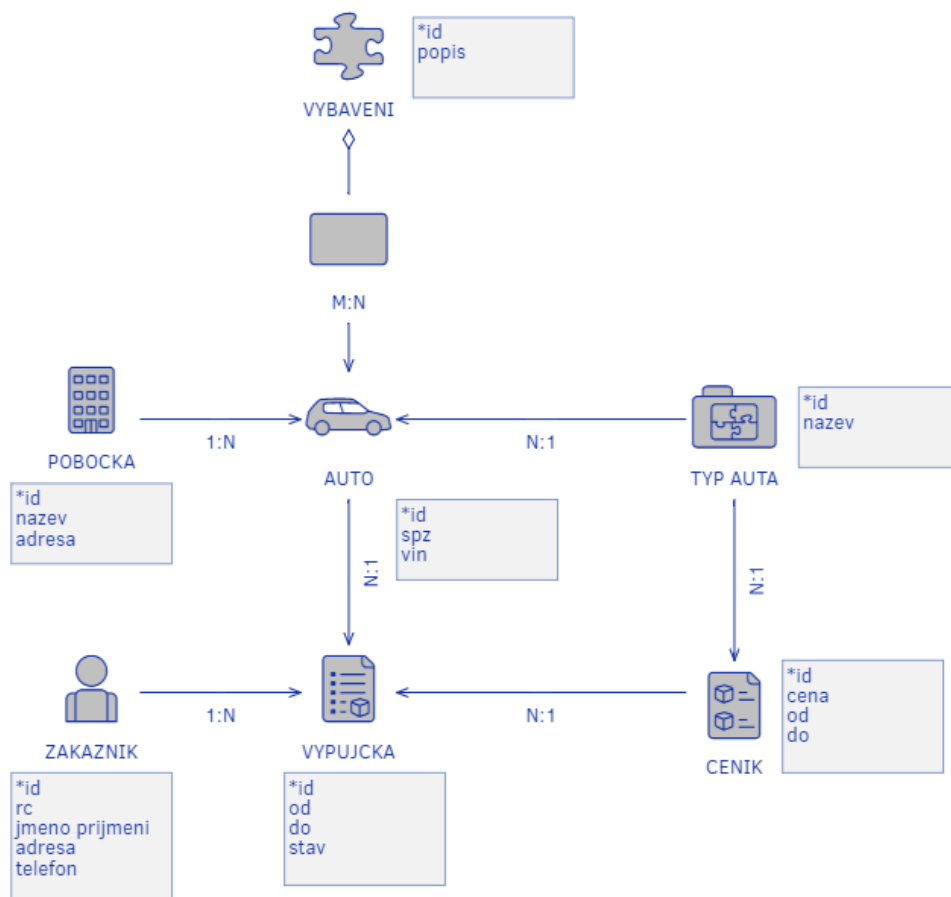
Ukázka jednotlivých bodů zadání

Nástin jednotlivých bodů zadání (v seminární práci se samozřejmě očekává rozsáhlejší vypracování)

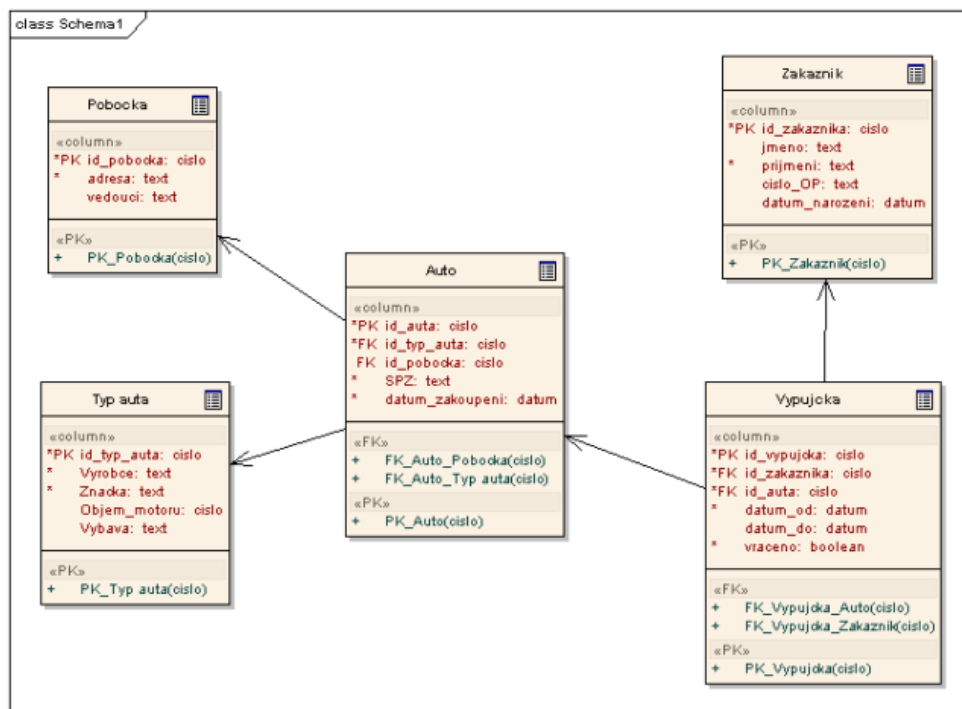
1. Business zadání

Auto se vždy vyskytuje na pobočce, kde si ho může zákazník po registraci vypůjčit. Vedou se informace o jednotlivých výpůjčkách, ať už aktuálních, tak i historických. ...

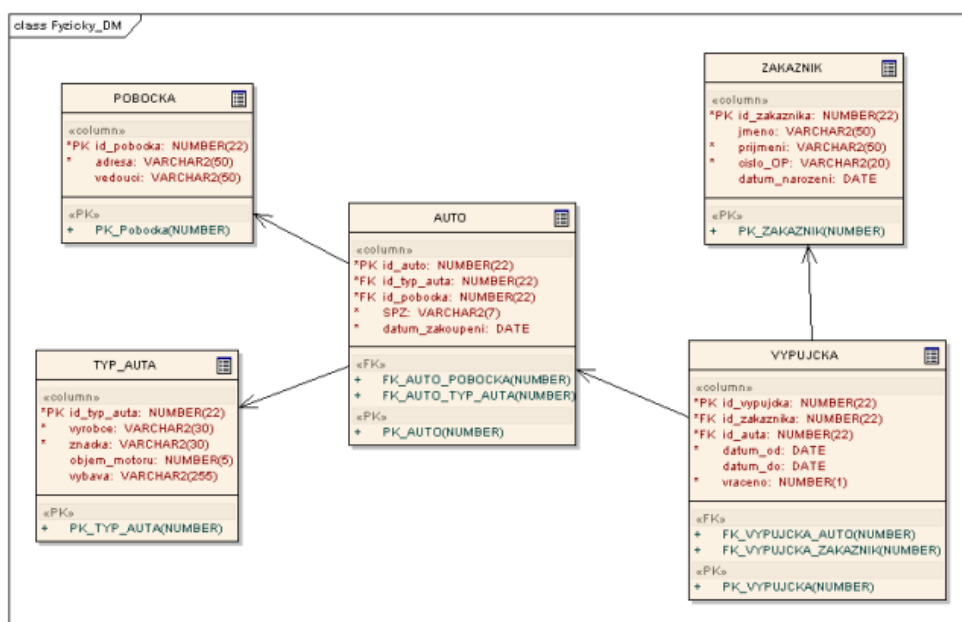
2. Konceptuální model



3. Logický datový model



4. Fyzický datový model



5. Základní skripty

```

CREATE TABLE AUTO (
  id_auta NUMBER(22) NOT NULL,
  id_typ_auta NUMBER(22) NOT NULL,
  id_pobocka NUMBER(22) NOT NULL,
  SPZ VARCHAR2(7) NOT NULL,
  datum_zakoupeni DATE NOT NULL);
  
```

```

CREATE TABLE POBOCKA (
  id_pobocka NUMBER(22) NOT NULL,
  adresa VARCHAR2(50) NOT NULL,
  vedouci VARCHAR2(50));
  
```

```

CREATE TABLE TYP_AUTA (
  id_typ_auta NUMBER(22) NOT NULL,
  
```

```
vyrobce VARCHAR2(30) NOT NULL,
znacka VARCHAR2(30) NOT NULL,
objem_motoru NUMBER(5),
vybava VARCHAR2(255));
```

```
CREATE TABLE VYPUJCKA (
id_vypujcka NUMBER(22) NOT NULL,
id_zakaznika NUMBER(22) NOT NULL,
id_auta NUMBER(22) NOT NULL,
datum_od DATE NOT NULL,
datum_do DATE,
vraceno NUMBER(1) NOT NULL);
```

```
CREATE TABLE ZAKAZNIK (
id_zakaznika NUMBER(22) NOT NULL,
jmeno VARCHAR2(50),
prijmeni VARCHAR2(50) NOT NULL,
cislo_OP VARCHAR2(20) NOT NULL,
datum_narozeni DATE);
```

```
ALTER TABLE AUTO ADD CONSTRAINT PK_AUTO PRIMARY KEY (id_auta);
```

```
ALTER TABLE POBOCKA ADD CONSTRAINT PK_Pobocka PRIMARY KEY (id_pobocka);
```

```
ALTER TABLE TYP_AUTA ADD CONSTRAINT PK_TYP_AUTA PRIMARY KEY (id_typ_auta);
```

```
ALTER TABLE VYPUJCKA ADD CONSTRAINT PK_VYPUJCKA PRIMARY KEY (id_vypujcka);
```

```
ALTER TABLE ZAKAZNIK ADD CONSTRAINT PK_ZAKAZNIK PRIMARY KEY (id_zakaznika);
```

```
ALTER TABLE AUTO ADD CONSTRAINT FK_AUTO_POBOCKA
FOREIGN KEY (id_pobocka) REFERENCES POBOCKA (id_pobocka);
```

```
ALTER TABLE AUTO ADD CONSTRAINT FK_AUTO_TYP_AUTA
FOREIGN KEY (id_typ_auta) REFERENCES TYP_AUTA (id_typ_auta);
```

```
ALTER TABLE VYPUJCKA ADD CONSTRAINT FK_VYPUJCKA_AUTO
FOREIGN KEY (id_auta) REFERENCES AUTO (id_auta);
```

```
ALTER TABLE VYPUJCKA ADD CONSTRAINT FK_VYPUJCKA_ZAKAZNIK
FOREIGN KEY (id_zakaznika) REFERENCES ZAKAZNIK (id_zakaznika);
```

6. Testovací data

-- TYP_AUTA

```
Insert into TYP_AUTA (ID_TYP_AUTA,OBJEM_MOTORU,VYBAVA,VYROBCE,ZNACKA) values (1,2500,'základní','BMW','525');
Insert into TYP_AUTA (ID_TYP_AUTA,OBJEM_MOTORU,VYBAVA,VYROBCE,ZNACKA) values (2,2000,'luxusní','AUDI','A6');
Insert into TYP_AUTA (ID_TYP_AUTA,OBJEM_MOTORU,VYBAVA,VYROBCE,ZNACKA) values (3,1200,'ambiente','Škoda','Fabia');
Insert into TYP_AUTA (ID_TYP_AUTA,OBJEM_MOTORU,VYBAVA,VYROBCE,ZNACKA) values (4,1600,'HPE','Lancia','Delta');
Insert into TYP_AUTA (ID_TYP_AUTA,OBJEM_MOTORU,VYBAVA,VYROBCE,ZNACKA) values (5,1400,'základní','Seat','Toledo');
Insert into TYP_AUTA (ID_TYP_AUTA,OBJEM_MOTORU,VYBAVA,VYROBCE,ZNACKA) values (6,650,'minimální','Fiat','500');
```

-- POBOCKA

```
Insert into POBOCKA (ADRESA,ID_POBOCKA,VEDOUCI) values ('Polní 123, Praha',1,'Smutný');
Insert into POBOCKA (ADRESA,ID_POBOCKA,VEDOUCI) values ('Zahradní 55, Hradec Králové',2,'Šťastný');
```

-- AUTO

```
Insert into AUTO (DATUM_ZAKOUPENI,ID_AUTO,ID_POBOCKA,ID_TYP_AUTA,SPZ) values
(to_date('01.03.2000','DD.MM.YYYY'),1,1,1,'1234567');
Insert into AUTO (DATUM_ZAKOUPENI,ID_AUTO,ID_POBOCKA,ID_TYP_AUTA,SPZ) values
(to_date('25.06.2008','DD.MM.YYYY'),2,1,2,'7654321');
Insert into AUTO (DATUM_ZAKOUPENI,ID_AUTO,ID_POBOCKA,ID_TYP_AUTA,SPZ) values
(to_date('01.01.2009','DD.MM.YYYY'),3,1,3,'3456712');
Insert into AUTO (DATUM_ZAKOUPENI,ID_AUTO,ID_POBOCKA,ID_TYP_AUTA,SPZ) values
(to_date('16.03.2009','DD.MM.YYYY'),4,2,4,'4567123');
```

```
Insert into AUTO (DATUM_ZAKOUPENI,ID_AUTO,ID_POBOCKA,ID_TYP_AUTA,SPZ) values
(to_date('05.05.1999','DD.MM.YYYY'),5,2,5,'7612345');
```

```
Insert into AUTO (DATUM_ZAKOUPENI,ID_AUTO,ID_POBOCKA,ID_TYP_AUTA,SPZ) values
(to_date('12.06.2003','DD.MM.YYYY'),6,2,6,'7777777');
```

```
-- ZAKAZNIK
```

```
Insert into ZAKAZNIK (CISLO_OP,DATUM_NAROZENI,ID_ZAKAZNIKA,JMENO,PRIJMENI) values
('12345678',to_date('01.09.1978','DD.MM.YYYY'),1,'Pavel','Nový');
```

```
Insert into ZAKAZNIK (CISLO_OP,DATUM_NAROZENI,ID_ZAKAZNIKA,JMENO,PRIJMENI) values
('987654321',to_date('24.05.1965','DD.MM.YYYY'),2,'Zuzana','Krátká');
```

```
Insert into ZAKAZNIK (CISLO_OP,DATUM_NAROZENI,ID_ZAKAZNIKA,JMENO,PRIJMENI) values
('55544442',to_date('30.11.1987','DD.MM.YYYY'),3,'Jan','Novák');
```

```
...
```

7. Dotazy

Přehled aktuálně vypůjčených aut a kdo je má vypůjčené:

```
SELECT
    zakaznik.jmeno,
    zakaznik.prijmeni,
    typ_auta.vyrobce,
    typ_auta.znacka,
    vypujcka.datum_od
FROM vypujcka
    JOIN auto
        ON vypujcka.id_auta = auto.id_auto
    JOIN typ_auta
        ON typ_auta.id_typ_auta = auto.id_typ_auta
    JOIN zakaznik
        ON vypujcka.id_zakaznika = zakaznik.id_zakaznika
WHERE vypujcka.vraceno = 0;
```

8. Optimalizace, indexy

Nejčastější dotaz - kdo má vypůjčené auto s danou SPZ - trvá dlouho:

```
SELECT zakaznik.jmeno, zakaznik.prijmeni
FROM vypujcka
    JOIN auto
        ON vypujcka.id_auta = auto.id_auto
    JOIN zakaznik
        ON zakaznik.id_zakaznika = vypujcka.id_zakaznika
WHERE vypujcka.vraceno = 0
    AND auto.spz = 'ABC1234';
```

Návrh řešení optimalizace:

Přidat index nad sloupec SPZ tabulky AUTO - unikátní hodnoty, dle kterých se často vyhledává:

```
CREATE INDEX idx_auto_spz ON auto (SPZ);
```

Termíny a řešení

1. Formulujte své zadání a odevzdejte jako samostatný soubor (nejpozději do **26.03.2017 23:59**)
2. CHECKPOINT - odevzdání první části semestrální práce (nejpozději do **23.04.2017 23:59**)
3. FINAL - odevzdání celé vypracované semestrální práce (nejpozději do **28.05.2017 23:59**)

Řešení

Domácí úkol (body 1, 2 a 3) odešlete k ohodnocení tlačítkem níže. Dodržujte pokyny pro vyplnění požadavku, například formát přílohy či název požadavku, pokud vyučující takové pokyny uvedl v zadání domácího úkolu.

Odeslat řešení domácího úk...

toto tlačítko je určeno pro skupiny T1 a T2, výuka v tutoriálech prvního ročníku, vyučující Kastner Vojtěch

Odeslat řešení domácího úk...

toto tlačítko je určeno pro skupinu T3, výuka v tutoriálech druhého ročníku, vyučující Přech Karel

Termíny odevzdání jsou závazné. Na později odevzdaná řešení nebude brán zřetel. Svá řešení můžete odevzdávat i dříve.

Pokyny pro vyplnění požadavku

Vyplňte "**Název**" - **CHECKPOINT 1, CHECKPOINT 2 nebo FINAL**. (Jméno odesílatele a předmět jsou doplněny automaticky školním systémem.)


Soubory u každého bodu zadání odevzdejte do přílohy požadavku dle uvedeného názvu. Rozsah jednotlivých bodů musí být takový, aby dostatečně předvedl pochopení probírané látky v rámci celého semestru (tzn. např. ve fyzickém datovém modelu se musí objevit cizí klíče, různé datové typy, ...)

Jak odevzdat domácí úkol

Vyplňte "Název" - někteří vyučující uvádějí požadovaný název v zadání domácího úkolu. (Jméno odesílatele a předmět jsou doplněny automaticky školním systémem.)

Pokud chcete do požadavku **vložit přílohy**, klikněte na tlačítko "Vložit přílohu". Zobrazí se dialogové okno, kde zadáte cestu k příslušnému souboru. Po kliknutí na tlačítko "OK" se zobrazí seznam příloh, následně zavřete dialogové okno.

Domácí úkol odevzdáte kliknutím na tlačítko "Odeslat". V záhlaví se zobrazí lišta s informací "Požadavek byl úspěšně odeslán ke zpracování." *Založené ale neodeslané požadavky se hlásí ve vašem úkolovníku aktivitou "Neodesláno!" O akceptaci (či zamítnutí) vyučujícím Vás bude systém informovat aktivitou a požadavek s domácím úkolem se následně odkáže na Vaší kartu studenta v předmětu.*

Podrobná nápověda k odevzdání domácího úkolu v  [Jak odevzdat domácí úkol](#)