UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

*Evidence obyvatelstva zdravotní pojišťovny*

Seminární práce z předmětu Databázové systémy I

Jméno a příjmení:

Tomáš Kvapil

V Chrudimi dne 9.12. 2022

Obsah

[Analýza 10](#_Toc70355807)

[ERD 13](#_Toc70355808)

[Relační model dat 16](#_Toc70355809)

[Fyzický model dat 17](#_Toc70355810)

[Výčet SQL dotazů 18](#_Toc70355811)

[Závěr 22](#_Toc70355812)

[Přílohy 23](#_Toc70355813)

Analýza

**1) Scénář**

* Zásadní a nejdůležitější ohledně které se všechno bude odehrávat je **osoba**. Musí obsahovat *křestní jméno, příjmení, telefon, pohlaví,* dále *datum narození a úmrtí*, a poslední jestli je osoba *fyzicky/mentálně postižená*. **Osoba** může být **studentem** nebo **zaměstanancem**. **Student** bude obsahovat *datum* *nástupu* na školu, *obor* *studia* a *typ* *studia*. **Zaměstnanec** bude obsahovat *datum* *nástupu* do **zaměstnání**, kolik *pracuje* *hodin* *týdně*, jestli má *smlouvu* *na* *určitou* *dobu* *nebo* *ne*, a dále v jakém *oddělení* *pracuje*. **Student** navštěvuje **školu**. U **Školy** se bude evidovat pouze *název* a *adresu*. **Zaměstnání** evidujeme *název* dané firmy, *telefonní* *kontakt*, případně *poznámku*. **Zaměstnanec** chodí do **zaměstnání**. Entita **Adresa** bude obsahovat *ulici*, *číslo* *popisné*, *poštovné* *směrovací* *číslo*, *obec*, *kraj* a *stát*. **Lékař** bude obsahovat jaký *typ* *doktora* je a kolik mu *měsíčně* *proplácíme* za danou práci. **Lékař** musí mít **Adresu**. **Platba** bude obsahovat kolik *posíláme* *peněz* dané **osobě**, jaké jsou *měsíční* *náklady* na osobu a kolik osoba *platí* *měsíčně* za pojištění. **Osoba** platí za **pojištění**. **Zápis** bude evidovat *datum* *změny*, do kdy je daná *změna* *platná* a *popis* dané *změny*. **Osoba** může mít **zapisy**. U **osoby** budou evidovány **občanské** **průkazy**, **řidičský** **průkaz**, **karta** **pojičtěnce**. **Doklad** bude evidovat *datum* *vydáni*, do kdy daný *doklad* *platí*, *místo* *vydání* a *číslo* *průkazu*. **Řidičský** **průkaz** bude obsahovat navíc *řidičské* *oprávnění*. **Karta** **pojištěnce** bude navíc obsahovat *číslo* *pojištěnce*.

**2) Strukturální a procedurální pravidla**

**Procedurální**

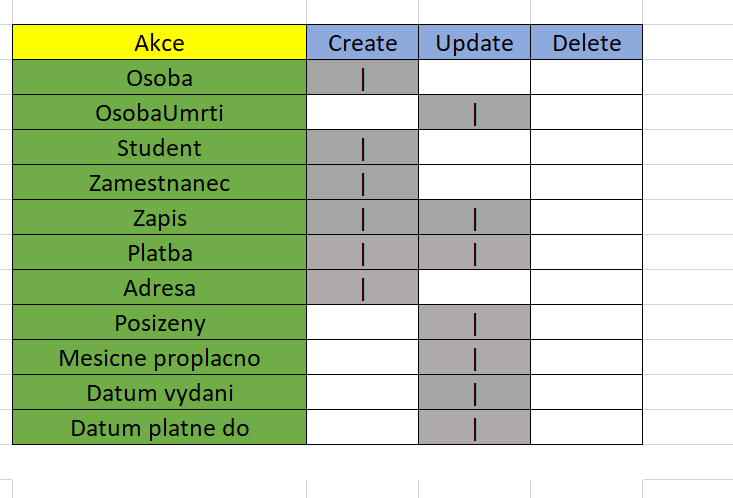
1. Datum narození u osoby musí být menší než datum nástupu do školy.
2. Datum narození u osoby musí být menší než datum nástupu do zaměstnání.

**Strukturální**

1. Zaměstnanec nemůže pracovat více jak 100 hodin týdně.
2. Měsíčně posíláno peněz pojišťovnou může být rovno buď nule nebo musí být větší než nula.
3. Měsíční náklady na osobu můžou být rovno buď nule nebo musí být větší než nula.
4. Měsíčně posíláno pojišťovně osobou musí být buď rovno nule nebo větší než nula.

**3) Jednoduchá analýza vyplývající ze scénáře, která zahrnuje označení entit (např. tučně), označení atributů (kurzívou).**

**4) CRUD analýza**



**5) Diskuze smyček -** [**diskuse možných datových nekonzistencí v důsledku smyček ve schématu**](https://dbs.fit.cvut.cz/sw/editor/loop-discussion/)**.**

* Nemám smyčky

**6) Integritní omezení (IO)**

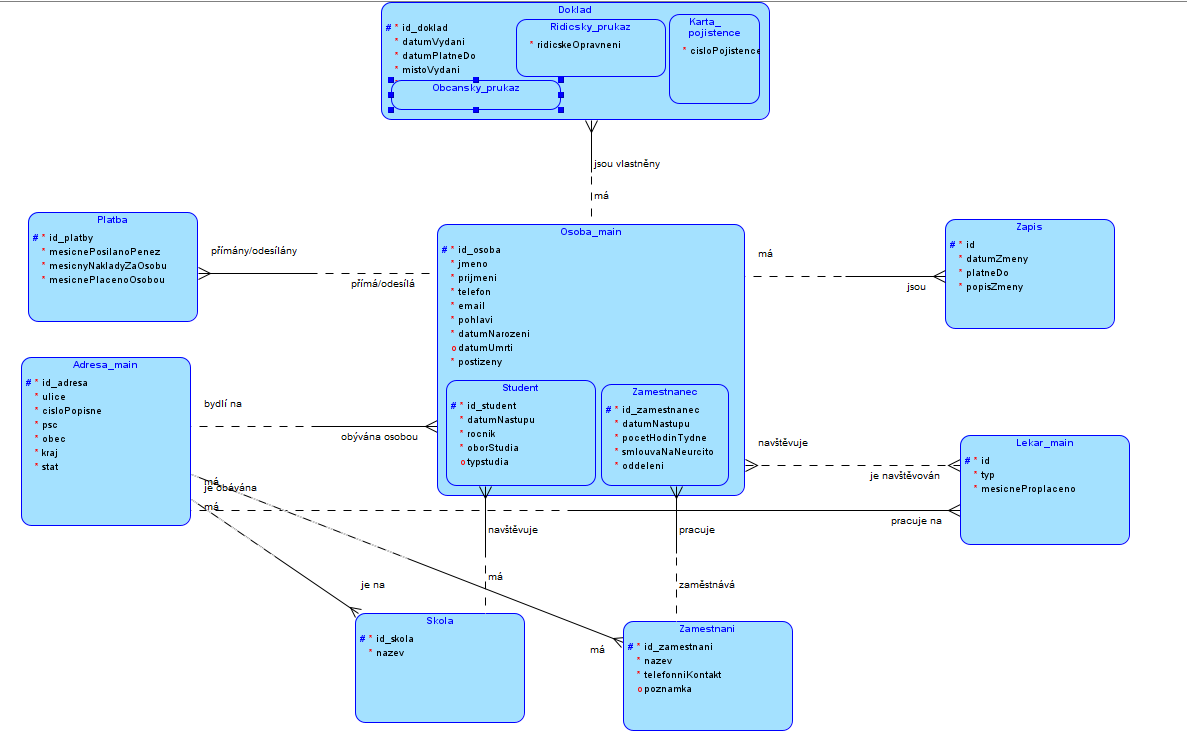
IO1: Měsíčně posíláno peněz pojišťovnou může být rovno buď nule nebo musí být větší než nula.

IO2: Měsíční náklady na osobu můžou být rovno buď nule nebo musí být větší než nula.

IO3: Měsíčně posíláno pojišťovně osobou musí být buď rovno nule nebo větší než nula.

IO4: Zaměstnanec nemůže pracovat více jak 100 hodin týdně.

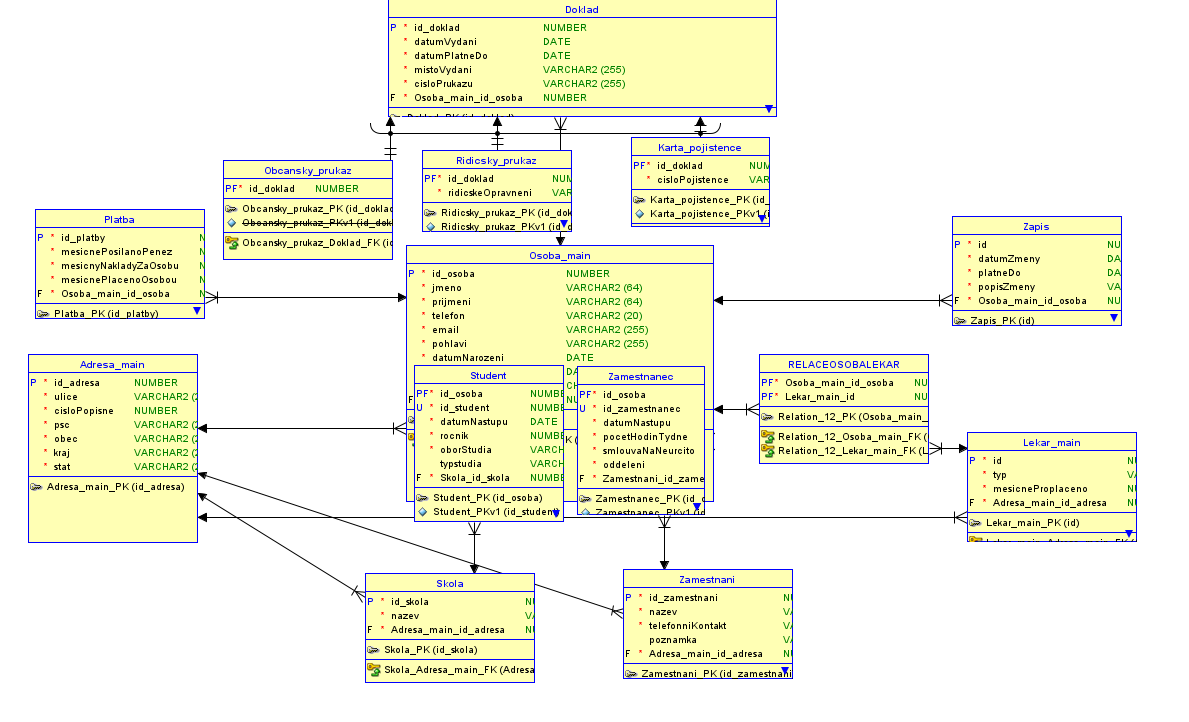
## ERD



1. Každá osoba může mít jeden nebo více zápisů a každý zápis musí mít jednu a pouze jednu osobu.
2. Každá osoba může přijímat/odesílat jednu nebo více plateb a každá platba musí mít jednu a pouze jednu osobu.
3. Každá osoba může mít jeden a více dokladů a každý doklad musí mít jednu a pouze jednu osobu.
4. Každá osoba může bydlet pouze a pouze na jedné adrese a každá adresa může být obývána jednou nebo více osob
5. Každý student musí navštěvovat pouze a pouze jednu školu a každá škola může mít jednoho nebo více studentů.
6. Každý zaměstnanec musí pracovat pouze a pouze v jednom zaměstnání a každé zaměstnání může zaměstnávat jednoho nebo více zaměstnanců.
7. Každá škola je na pouze a pouze jedné adrese a každá adresa může mít jednu nebo více škol.
8. Každé zaměstnání je na pouze a pouze jedné adrese a každá adresa může mít jednu nebo více zaměstnání.
9. Každá osoba může navštěvovat jednoho nebo více lékařů a každý lékař může být navštěvován jednou nebo více osobami.
10. Každý lékař musí pracovat na pouze a pouze jedné adrese a každá adresa může být obývána jedním nebo více lékaři.

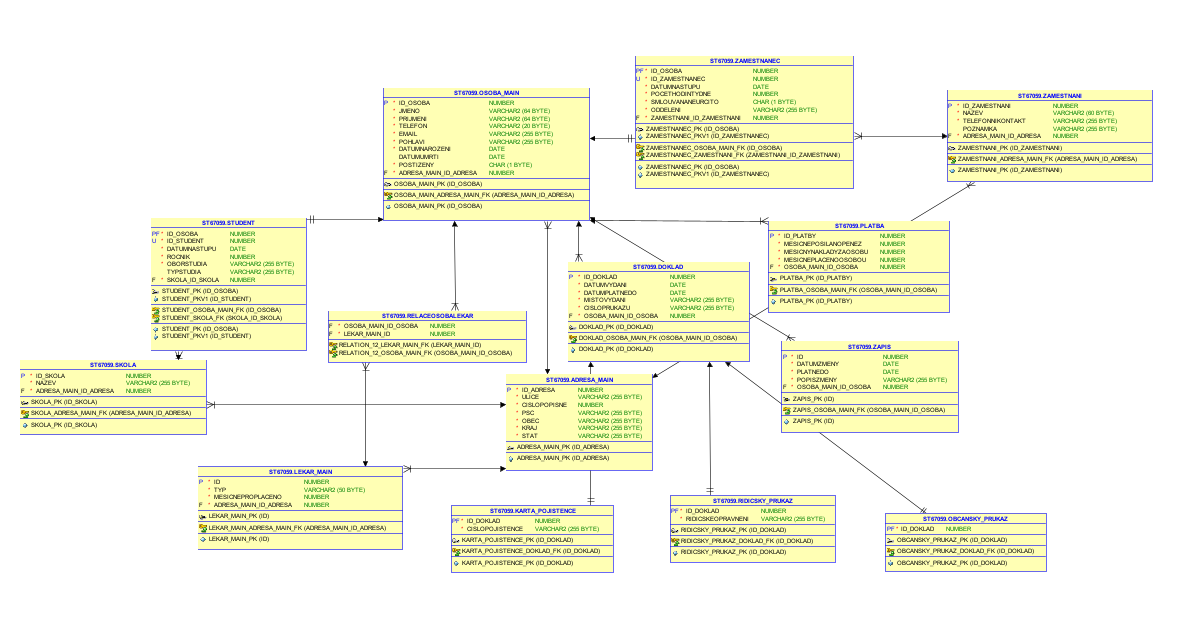
## 

## Relační model dat



1. SuperTyp Osoba\_main
   1. id\_osoba je povinný numerický primární klíč
   2. jmeno je povinný varchar s omezením 64 charakterů
   3. prijmeni je povinný varchar s omezením 64 charakterů
   4. telefon je povinný varchar s omezením 20 charakterů
   5. email je povinný varchar s omezením 255 charakterů
   6. pohlavi je povinný varchar s omezením 20 charakterů
   7. datumNarozeni je povinné datum
   8. datumUmrti je nepovinné datum
   9. postizeny je povinný boolean s omezením na 1 znak
   10. Adresa\_main\_id\_adresa je numerické id entity Adresa\_main
2. Podtyp Student
   1. id\_osoba je povinný numerický primární klíč děděný z osoby
   2. id\_student je povinný numerický primární klíč entity Student
   3. datumNastupu je povinné datum
   4. rocnik je povinný numerický atribut
   5. oborStudia je povinný varchar s omezením 255
   6. typStudia je nepovinný varchar s omezením 255
   7. id\_skola je numerické id entity Skola
3. Podtyp Zamestnanec
   1. id\_osoba je povinný numerický primární klíč děděný z osoby
   2. id\_zamestnanec je povinný numerický primární klíč entity Zamestnanec
   3. datumNastupu je povinné datum
   4. pocetHodinTydne je povinný numerický atribut
   5. smlouvaNaNeurcito je povinný boolean s omezením 1
   6. oddeleni je povinný varchar s omezením 255
   7. id\_zamestnani je numerické id entity Zamestnani
4. Skola
   1. id\_skola je povinný numerický primární klíč
   2. nazev je povinný varchar s omezením 255
   3. Adresa\_main\_id\_adresa je numerické id entity Adresa\_main
5. Zamestnani
   1. id\_zamestnani je povinný numerický primární klíč
   2. nazev je povinný varchar s omezením 255
   3. telefoniKontakt je povinný varchar s omezením 255
   4. poznámka je nepovinný varchar s omezením 255
   5. Adresa\_main\_id\_adresa je numerické id entity Adresa\_main
6. Adresa\_main
   1. id\_adresa je povinný numerický primární klíč
   2. ulice je povinný varchar s omezením 255
   3. cisloPopisne je povinný numerický atribut
   4. psc je povinný varchar s omezením 255
   5. obec je povinný varchar s omezením 255
   6. kraj je povinný varchar s omezením 255
   7. stat je povinný varchar s omezením 255
7. Platba
   1. id\_platba je povinný numerický primární klíč
   2. mesicnePosilanoPenez je povinný numerický atribut
   3. mesicnyNakladyZaOsobu je povinný numerický atribut
   4. mesicnePlacenoOsobou je povinný numerický atribut
   5. Osoba\_main\_id\_osoba je numerické id entity Osoba\_main
8. Zapis
   1. id\_zapis je povinný numerický primární klíč
   2. datumZmeny je povinné datum
   3. platneDo je povinné datum
   4. popisZmeny je povinný varchar s omezním 255
   5. Osoba\_main\_id\_osoba je numerické id entity Osoba\_main
9. Lekar\_main
   1. id je povinný numerický primární klíč
   2. typ je povinný varchar s omezením 50
   3. mesicneProplaceno je povinný numerický atribut
   4. Adresa\_main\_id\_adresa je numerické id entity Adresa\_main
10. SuperTyp Doklad
    1. id\_doklad je povinný numerický primární klíč
    2. datumVydani je povinné datum
    3. datumPlatneDo je povinné datum
    4. mistoVydani je povinný varchar s omezením 255
    5. cisloPrukazu je povinný varchar s omezením 255
    6. Osoba\_main\_id\_osoba je numerický id entity Osoba\_main
11. PodTyp Obcansky\_prukaz
    1. id\_doklad je povinný numerický primární klíč děděný z entity Doklad
12. PodTyp Ridicsky\_prukaz
    1. id\_doklad je povinný numerický primární klíč děděný z entity Doklad
    2. ridicskeOpravneni je povinný varchar s omezením 255
13. PodTyp Karta\_pojistence
    1. id\_doklad je povinný numerický primární klíč děděný z entity Doklad
    2. cisloPojistence je povinný varchar s omezením 255

Fyzický model dat



V Entitě Osoba\_main, jsou uchovávány data o osobě. Často atributy v této entitě jsou omezeny. U Varcharu 255, 64, 20. U Entity Adresa\_main je všude varchar omezen na 255. U Karty\_pojistenec je cislopojistence též varchar 255, neboť tam se vkládá /. U typu studia v entitě Student je typ studia povoleno na nepovinné neboť to není zásadní údaj.

Sekvence:

CREATE SEQUENCE SEQUENCE1 INCREMENT BY 1 START WITH 1 MAXVALUE 100 MINVALUE 0;

CREATE SEQUENCE Lekar\_SEQUENCE INCREMENT BY 1 START WITH 1 MAXVALUE 100 MINVALUE 0;

* sekvence jsou na zvyšování id u entity Lekar\_main, Student

Triggry:

create or replace RELACEOSOBALEKAR\_trigger

AFTER INSERT ON RELACEOSOBALEKAR

REFERENCING OLD AS OLD NEW AS NEW

BEGIN

delete from RELACEOSOBALEKAR where RELACEOSOBALEKAR.osoba\_id\_osoba = (select o.id\_osoba from osoba\_main o where o.datumumrti IS NOT NULL);

END;

create or replace TRIGGER Platba\_trigger

AFTER INSERT ON Platba

REFERENCING OLD AS OLD NEW AS NEW

BEGIN

delete from Platba where Platba.osoba\_main\_id\_osoba = (select o.id\_osoba from osoba\_main o where o.datumumrti IS NOT NULL);

END;

create or replace TRIGGER Doklad\_trigger

AFTER INSERT ON Doklad

REFERENCING OLD AS OLD NEW AS NEW

BEGIN

delete from Doklad where Doklad.osoba\_main\_id\_osoba = (select o.id\_osoba from osoba\_main o where o.datumumrti IS NOT NULL);

END;

create or replace TRIGGER Zamestnanec\_trigger

AFTER INSERT ON Zamestnanec

REFERENCING OLD AS OLD NEW AS NEW

BEGIN

delete from Zamestnanec where Zamestnanec.id\_osoba = (select o.id\_osoba from osoba\_main o where o.datumumrti IS NOT NULL);

END;

create or replace TRIGGER Student\_trigger

AFTER INSERT ON Student

REFERENCING OLD AS OLD NEW AS NEW

BEGIN

delete from Student where Student.id\_osoba = (select o.id\_osoba from osoba\_main o where o.datumumrti IS NOT NULL);

END;

* všechny triggery zachycují pokud se snažíme vytvořit nový záznam u lékaře, u plateb, dokladů, zamestnance, studenta pokud má daná osoba vyplněný datum úmrtí.

Pf je složený klíč

Výčet SQL dotazů

A1: Jmeno, prijmeni, email, ulice, obec, kde email končí na %gmail.com HK: D1

select o.jmeno, o.prijmeni, o.email, a.ulice, a.obec from osoba\_main o, adresa a where o.email like '%@gmail.com';

A2: Jmeno, prijmeni, email, ulice, obec, kde email končí na cokoli jiného než %gmail.com HK: D2

select o.jmeno, o.prijmeni, o.email, a.ulice, a.obec from osoba\_main o, adresa a where o.email not like '%@gmail.com';

A3: Jmeno, prijmeni, typ lékaře, kde pohlaví je žena HK: D3, VK: D1

select o.jmeno, o.prijmeni, l.typ from osoba\_main o, lekar l where o.pohlavi = 'žena';

A5: Všechno z osoba\_main spojeno se skolou pomocí adresa\_id\_adresa HK: D5

select \* from osoba\_main o join skola a using(adresa\_id\_adresa);

A6: ulice, obec, jmeno, telefon, kde id adresy se rovná id osoby HK: D6

select a.ulice, a.obec, o.jmeno, o.telefon from adresa\_main a join osoba\_main o on a.id\_adresa = o.adresa\_id\_adresa;

A7: Všechno ze zaměstnáni spojeno s adresou HK: D7

select \* from zamestnani natural join adresa\_main;

A8: Všechno z osoby a všechno z adresy HK:D8

select \* from osoba\_main cross join adresa\_main;

A9: Nazev zamestnani, jmeno oddleni, kde id zamestnani v zamestnancovi se shoduje s id zamestnani HK: D9

select zamestnani.nazev, za.oddeleni from zamestnani left join zamestnanec za on zamestnani.id\_zamestnani = za.zamestnani\_id\_zamestnani;

A10: Jmeno, email, rocnik, kde id osoby ve studentovi se shoduje s id osoby. HK: D10

select o.jmeno, o.email, st.rocnik from student st right join osoba\_main o on o.id\_osoba = st.id\_osoba;

A11: Nazev skoly, typ studia, rocnik, kde id skoly ve studentovi se shoduje s id skoly

HK: D11

select sk.nazev, st.typstudia, st.rocnik from skola sk full outer join student st on sk.id\_skola = st.skola\_id\_skola;

A12: Jmeno, prijmeni, kde id osoby je nejvyšší a zároveň je pohlavím muž HK:D12 VK:D22

select jmeno, prijmeni from osoba\_main where id\_osoba = (select max(id\_osoba) from osoba\_main where pohlavi = 'muž');

A13: Název zaměstnání, počet hodiny týdně zaměstnance, datum nástupu a jméno zaměstnance, kde id zamestnani v zaměstnanci je rovno s id zaměstnáni HK:D13, VK: D12, D14

select zam.nazev, zames.pocet\_hodin\_tydne, zames.datumNastupuDoFirmy, zames.jmeno from

(select zamestnanec.pocethodintydne pocet\_hodin\_tydne,zamestnanec.zamestnani\_id\_zamestnani as idZam, zamestnanec.datumnastupu as datumNastupuDoFirmy, (select o.jmeno from osoba\_main o where o.id\_osoba = zamestnanec.id\_osoba) as jmeno from zamestnanec) zames, zamestnani zam

where zam.id\_zamestnani = zames.idZam;

A14: Jmeno, datum narození, datum nástupu do práce, kde id osoby se shoduje s id zaměstnancem HK:D14, VK: D13

select jmeno, datumnarozeni, datum.datum\_nastupu\_do\_prace from (select datumnastupu datum\_nastupu\_do\_prace, id\_osoba from zamestnanec) datum, osoba\_main where osoba\_main.id\_osoba = datum.id\_osoba;

A15: Popis změny, id změny a id osoby, kde pohlavi osoby není muž HK:D15

-Not exists

select popiszmeny, id, osoba\_id\_osoba from zapis where not exists(select datumumrti from osoba\_main where osoba\_main.id\_osoba = zapis.osoba\_id\_osoba and pohlavi = 'muž');

-exists

- oddeleni a nazev zaměstnání, kde je poznámka je nenulová a oddelení patří do zaměstnání HK: D15, VK: D1

select oddeleni, firma.nazev from zamestnanec, zamestnani firma where exists(select poznamka from zamestnani where zamestnani.id\_zamestnani = zamestnanec.zamestnani\_id\_zamestnani and poznamka is not null) and firma.id\_zamestnani = zamestnanec.zamestnani\_id\_zamestnani;

A16: Datumy nástupů studentů do škol a zaměstnanců do zaměstnání HK:D16

select datumNastupu from student union select datumNastupu from zamestnanec;

A17: Adresy kde bydlí osoby a nepracujou lékaři HK:D17

select adresa\_id\_adresa from osoba\_main minus select adresa\_id\_adresa from lekar;

A18: Jmeno, prijmeni, telefon, pohlavi kde id osoby se shoduje s id v zamestnancovi HK:D18

select jmeno, prijmeni, telefon, pohlavi from osoba\_main left join zamestnanec on zamestnanec.id\_osoba = osoba\_main.id\_osoba

intersect select jmeno, prijmeni, telefon, pohlavi from osoba\_main right join zamestnanec on zamestnanec.id\_osoba = osoba\_main.id\_osoba;

A19: Jmeno bude aktualizováno na `Pepa`, kde délka jména je větší než 6 HK:D19

update osoba\_main set jmeno = 'Pepa' where length(jmeno) > 6;

A20:Oddeleni, datum nastupu, počet hodin týdně kde počet hodin týdně je sudé číslo HK:D20

select oddeleni, datumnastupu, pocethodintydne from zamestnanec where mod(pocethodintydne, 2) = 0;

A21: Místo vydání, datum vydání, kde datum je vetší nebo rovno HK:D21

select mistovydani, datumvydani from doklad where datumvydani >= TO\_DATE('2020-01-01','YYYY-MM-DD');

A22:

-count: počet hodin týdně, kde jsou vyšší než 40

select count(pocethodintydne) from zamestnanec where pocethodintydne >= 40;

-max: nejvyšší studující ročník

select max(rocnik) as nejvyssi\_rocnik from student;

-min: nejnižžší studovaný ročník

select min(rocnik) as nejmensi\_rocnik from student;

-avg: průměr pracujících hodin ve všech zaměstnání

select avg(za.pocethodintydne) as avg\_pocet\_hodin\_tydne\_vsema\_zamestnanima from zamestnanec za;

-sum: počet lidí s postížením

select sum(postizeny) pocet\_postizenych\_osoba from osoba\_main;

A23: Jmeno, prijmeni, datum narození, kde datum narození je větší ne 1.1.1970 HK:D17, VK:D22

select jmeno, prijmeni, max(datumnarozeni) from osoba\_main group by jmeno, prijmeni having max(datumnarozeni) > TO\_DATE('1970-01-01','YYYY-MM-DD');

A24: Jmeno a pohlaví, kde id osoby se shoduje s id ve studentovi HK:D24, VK:D1, D15

select o.jmeno, o.pohlavi from osoba\_main o, student st where exists(select adresa\_id\_adresa from osoba\_main where o.id\_osoba = st.id\_osoba);

select o.jmeno, o.pohlavi from osoba\_main o where o.id\_osoba in (select id\_osoba from student st);

select o.jmeno, o.pohlavi from osoba\_main o, student st where o.id\_osoba = st.id\_osoba;

A25: Celkem pohlaví, pohlaví seřazeno od největšího čísla HK:D25, VK:D22

select count(pohlavi), pohlavi from osoba\_main group by pohlavi having count(pohlavi) > 1 order by count(pohlavi) desc;

A26: Jmeno, datum narozeni, datum nastupu, pocethodin tydně, kde id osoby se shoduje s id v zaměstnancovi HK:D26, VK: D1, D9

create or replace view pohled\_d26 as

select o.jmeno, o.datumnarozeni, z.datumnastupu, z.pocethodintydne from osoba\_main o left join zamestnanec z on o.id\_osoba = z.id\_osoba;

A27:Všechno z pohledu, kde datum je větší než 1.1. 1980 HK:D27, VK:D21

select \* from pohled\_d26 where datumnarozeni > TO\_DATE('1980-01-01','YYYY-MM-DD');

A28: Všechny osoby od lékaře 1 k lékaři 3. DK:28

insert into RELACEOSOBALEKAR select osoba\_id\_osoba,3 from RELACEOSOBALEKAR where lekar\_id = 1;

A29: Nastav nejvyšší počty hodin týdně, kde je zaměstnání s id 2, kde je počet hodin týdně 40. HK:D29, VK:D22, D12, D13, D14

update zamestnanec za set za.pocethodintydne = (select max(pocethodintydne) from zamestnanec where zamestnani\_id\_zamestnani = 2) where za.pocethodintydne = 40;

A30: Odstraň zaměstnance, kde datum nástupu je mezi lety 1.1. 1990 a 1.1. 1999 HK:D30

delete from zamestnanec where datumnastupu in (select datumnastupu from zamestnanec where datumnastupu between TO\_DATE('1990-01-01','YYYY-MM-DD') and TO\_DATE('1999-01-01','YYYY-MM-DD'));

Závěr

Databáze by mohla být provozu schopná pro menší pojišťovnu. Podle mě databáze obsahuje opravdu pouze zásadní informace o obyvatelech různých měst. Systém rozhodně není perfektní a snažil jsem se ho dělat, abych to zvládl. Kdybych měl větší znalosti databází, dal by se můj systém o dost vylepšit. Například pokud daný člověk zemřel, aby se automaticky o tom zapsal záznam do entity Zapis. To samé pokud by osoba nastoupila na školu, zaměstnání nebo změnila adresu bydliště. Osvoj sem si dělání návrhu databáze, základní skripty a hodně informacích o dané problematice. S Crud analýzou si nejsem vůbe jistý, jelikož v materilách jsem toho moc nenašel a ani na internetu toho moc není. Jelikož jsem Crud analýzu přehlížel a nechával si ji nakonec, už jsem se nemohl poradit s cvičícím ohledně mého problému. Dále co bych rád zmínil je, že u některých entit mám \_main, je to z důvodu, že jsem jsem nemohl dané entity dropnout a ani nic s nimi udělat a jediné řešení bylo podle stackoverflow killnout celou databázy a s tím i dané procesy. Jelikož na to nemám oprávnění a je blbost killnout celou databázy jenom kvůli mé práci. Jak dlouho to bude trvat jsem se nikde nedočetl.

Přílohy

1. DDL skript pro vytvoření tabulek, omezení, sekvencí, indexů, triggerů, apod.
2. DML skript pro naplnění tabulek daty.
3. Soubor dmd včetně složky (Datový model musí souhlasit s DDL skriptem)
4. Skript se všemi dotazy.