

# Databáze jazykové školy

Dokumentace semestrální práce pro předmět 4IT218 Databáze

Daria Pesotckaia

*LS 2021/2022*

Uživatelské jméno: pesd05

***Do odevzdávnice v InSIS uložte dokumentaci semestrální práce jako soubor ve formátu PDF. Dokumentaci nekomprimujte (nekládejte soubory .zip, .rar apod.) Do odevzdávnice v InSIS nekládejte žádné jiné soubory.***

## Obsah

1	Popis zvolené výseče světa – zadání .....	3
2	Konceptuální schéma reality .....	3
3	Konceptuální datový model .....	5
4	Dokumentace databáze .....	6
4.1	Fyzický datový model .....	6
4.2	Definice relačních tabulek a souvisejících objektů .....	7
4.3	Integritní omezení .....	7
4.4	Definice přístupových práv .....	13
5	Obsah databáze .....	15
5.1	SQL příkazy pro naplnění databáze daty .....	15
5.2	Opis vložených dat .....	17

## 1 Popis zvolené výšeče světa – zadání

Jazyková škola „Baker Street“ potřebuje databázi zákazníků a zaměstnanců.

U každého zaměstnance evidujeme ID (osobní číslo), jméno, datum narození, funkce a kontaktní telefon. Každý zaměstnanec má pouze jednu funkci.

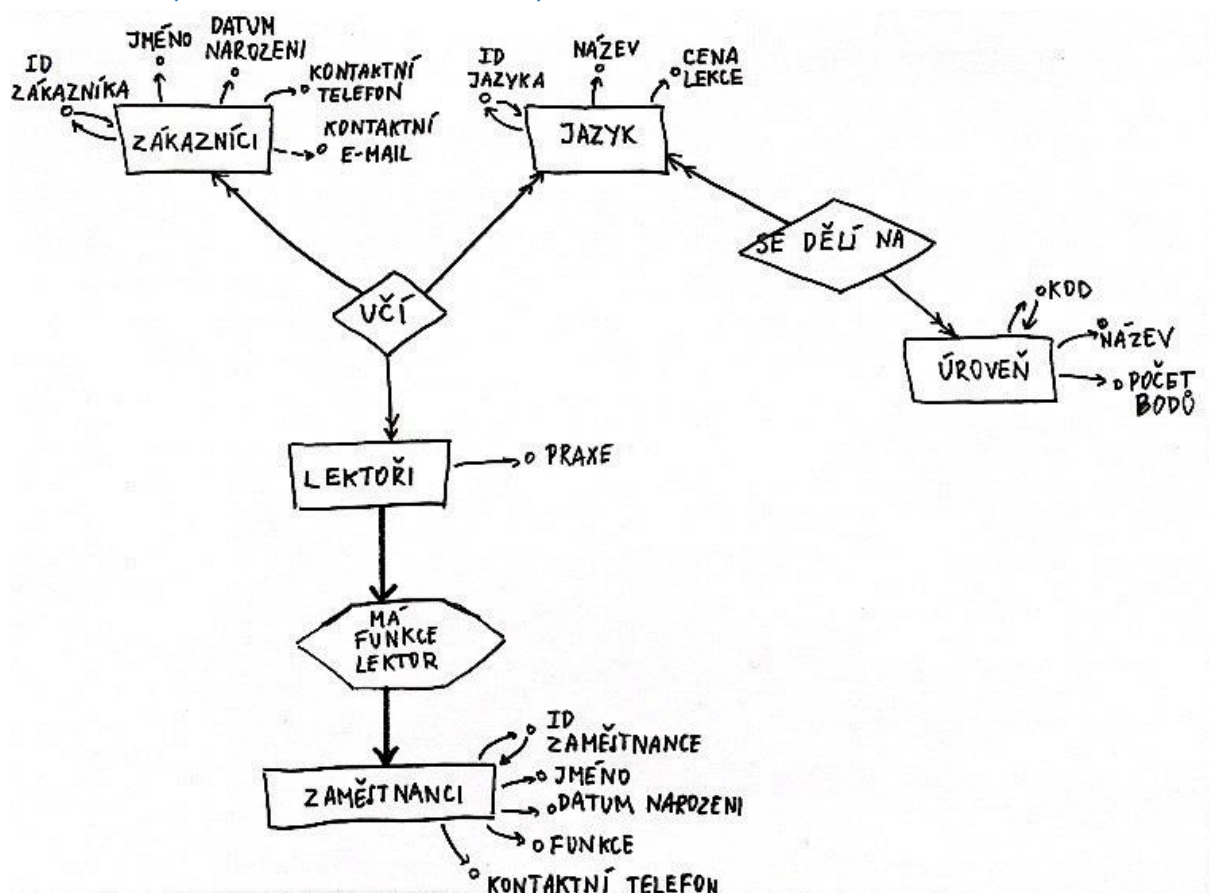
Do tabulky lektori patří každý zaměstnanec s funkcí lektor. U každého lektora evidujeme také počet roků v praxi.

Každý zákazník má přiřazeného lektora. U zákazníků takto evidujeme ID (osobní číslo), jméno, datum narození, kontaktní telefon a kontaktní email (volitelně).

Každý zákazník učí jeden nebo více jazyků. Pro každý jazyk evidujeme ID (osobní číslo), název a cenu lekce. Jeden jazyk může být přednášen více lektory a jeden lektor může přednášet více jazyků.

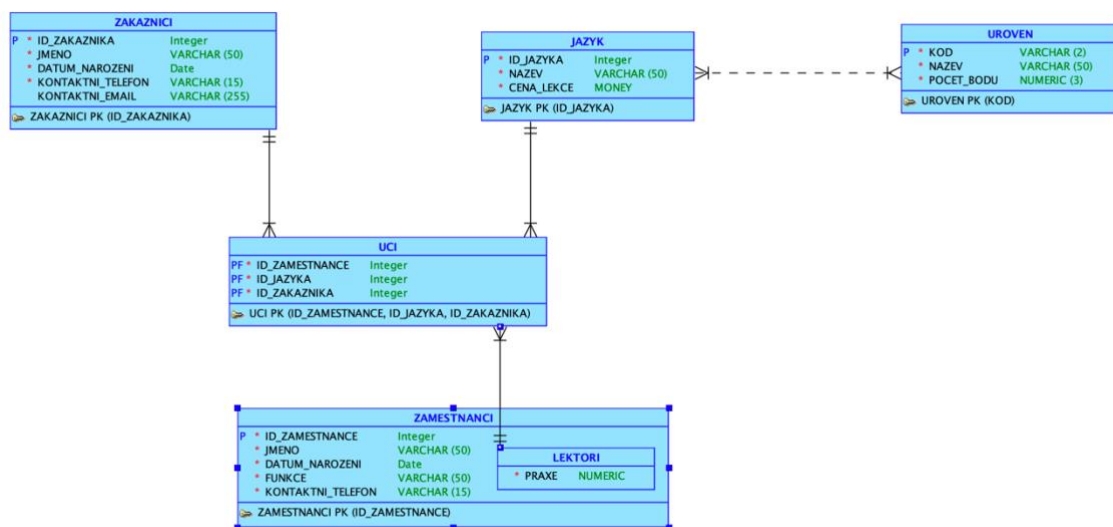
Jazyky se dělí na úrovně, u každé úrovně evidujeme kód a název. Před zahájením výuky každý zákazník píše test a podle počtu bodů je mu přiřazena úroveň. Proto v tabulce úrovní také zaznamenáváme minimální dosažený počet bodů potřebný pro danou úroveň.

## 2 Konceptuální schéma reality



Obrázek 1 Jazyková škola - konceptuální schéma reality, zdroj: autor

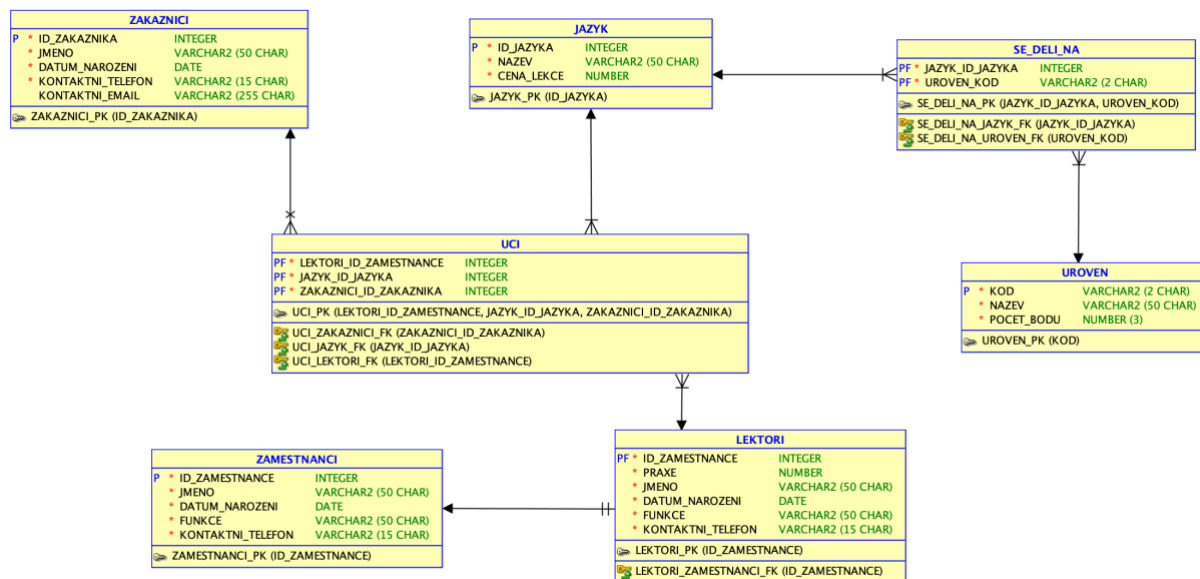
### 3 Konceptuální datový model



Obrázek 2 Jazyková škola - konceptuální datový model, zdroj: autor

## 4 Dokumentace databáze

### 4.1 Fyzický datový model



Obrázek 3 - Jazyková škola - fyzický datový model, zdroj: autor

#### 4.2 Definice relačních tabulek a souvisejících objektů

```
CREATE TABLE jazyk (  
    id_jazyka INTEGER NOT NULL,  
    nazev VARCHAR2(50 CHAR) NOT NULL,  
    cena_lekce NUMBER DEFAULT 0 NOT NULL  
);
```

```
ALTER TABLE jazyk ADD CONSTRAINT jazyk_pk PRIMARY KEY ( id_jazyka );
```

```
ALTER TABLE jazyk ADD CONSTRAINT cena_lekce_chc CHECK ( cena_lekce > 0);
```

```
GRANT SELECT ON jazyk TO STUDENT;
```

```
GRANT SELECT, DELETE, INSERT, UPDATE ON jazyk TO DB4IT218;
```

```
CREATE TABLE lektori (  
    id_zamestnance INTEGER NOT NULL,  
    praxe NUMBER DEFAULT 0 NOT NULL,  
    jmeno VARCHAR2(50 CHAR) NOT NULL,  
    datum_narozeni DATE NOT NULL,  
    funkce VARCHAR2(50 CHAR) NOT NULL,  
    kontaktni_telefon VARCHAR2(15 CHAR) NOT NULL  
);
```

```
ALTER TABLE lektori ADD CONSTRAINT lektori_pk PRIMARY KEY ( id_zamestnance );
```

```
ALTER TABLE lektori ADD CONSTRAINT praxe_chc CHECK ( praxe > 0);
```

```
ALTER TABLE lektori ADD CONSTRAINT datum_narozeni_chc CHECK (datum_narozeni < sysdate  
);
```

```
GRANT SELECT ON lektori TO STUDENT;
```

```
GRANT SELECT, DELETE, INSERT, UPDATE ON lektori TO DB4IT218;
```

```
CREATE TABLE se_deli_na (  
    jazyk_id_jazyka INTEGER NOT NULL,  
    uroven_kod VARCHAR2(2 CHAR) NOT NULL  
);
```

```
ALTER TABLE se_deli_na ADD CONSTRAINT se_deli_na_pk PRIMARY KEY ( jazyk_id_jazyka,  
    uroven_kod );
```

```
GRANT SELECT ON se_deli_na TO STUDENT;
```

```
GRANT SELECT, DELETE, INSERT, UPDATE ON se_deli_na TO DB4IT218;
```

```
CREATE TABLE uci (  
    lektori_id_zamestnance INTEGER NOT NULL,  
    jazyk_id_jazyka      INTEGER NOT NULL,  
    zakaznici_id_zakaznika INTEGER NOT NULL  
);
```

```
ALTER TABLE uci  
    ADD CONSTRAINT uci_pk PRIMARY KEY ( lektori_id_zamestnance,  
                                         jazyk_id_jazyka,  
                                         zakaznici_id_zakaznika );
```

```
GRANT SELECT ON uci TO STUDENT;
```

```
GRANT SELECT, DELETE, INSERT, UPDATE ON uci TO DB4IT218;
```

```
CREATE TABLE uroven (  
    kod    VARCHAR2(2 CHAR) NOT NULL,  
    nazev  VARCHAR2(50 CHAR) NOT NULL,  
    pocet_bodu NUMBER(3) DEFAULT 0 NOT NULL  
);
```

```
ALTER TABLE uroven ADD CHECK ( pocet_bodu BETWEEN 0 AND 100 );
```

```
ALTER TABLE uroven ADD CONSTRAINT uroven_pk PRIMARY KEY ( kod );
```

```
GRANT SELECT ON uroven TO STUDENT;
```

```
GRANT SELECT, DELETE, INSERT, UPDATE ON uroven TO DB4IT218;
```

```
CREATE TABLE zakaznici (  
    id_zakaznika    INTEGER NOT NULL,  
    jmeno           VARCHAR2(50 CHAR) NOT NULL,  
    datum_narozeni  DATE NOT NULL,  
    kontaktni_telefon VARCHAR2(15 CHAR) NOT NULL,  
    kontaktni_email VARCHAR2(255 CHAR) NOT NULL  
);
```

```
ALTER TABLE zakaznici ADD CONSTRAINT datum_narozeni_chc CHECK( datum_narozeni <
```



```
sysdate);
```

```
ALTER TABLE zakaznici ADD CONSTRAINT zakaznici_pk PRIMARY KEY ( id_zakaznika );
```

```
GRANT SELECT ON zakaznici TO STUDENT;
```

```
GRANT SELECT, DELETE, INSERT, UPDATE ON zakaznici TO DB4IT218;
```

```
CREATE TABLE zamestnanci (  
    id_zamestnance    INTEGER NOT NULL,  
    jmeno             VARCHAR2(50 CHAR) NOT NULL,  
    datum_narozeni    DATE NOT NULL,  
    funkce            VARCHAR2(50 CHAR) NOT NULL,  
    kontaktni_telefon VARCHAR2(15 CHAR) NOT NULL  
);
```

```
ALTER TABLE zamestnanci ADD CONSTRAINT zamestnanci_pk PRIMARY KEY ( id_zamestnance );
```

```
ALTER TABLE zamestnanci ADD CONSTRAINT datum_narozeni_chc CHECK (datum_narozeni <  
sysdate);
```

```
GRANT SELECT ON zamestnanci TO STUDENT;
```

```
GRANT SELECT, DELETE, INSERT, UPDATE ON zamestnanci TO DB4IT218;
```

```
ALTER TABLE lektori  
    ADD CONSTRAINT lektori_zamestnanci_fk FOREIGN KEY ( id_zamestnance )  
        REFERENCES zamestnanci ( id_zamestnance );
```

```
ALTER TABLE lektori ADD CONSTRAINT funkce_chc CHECK ( funkce = 'lektor');
```

```
CREATE OR REPLACE TRIGGER funkce_lektori  
BEFORE INSERT OR UPDATE ON lektori FOR EACH ROW  
AS BEGIN  
    IF :new.funkce != 'lektor' THEN  
        raise_application_error(-20001, 'Funkce musí být lektor!!');  
    END IF;  
END;
```

```
ALTER TABLE se_deli_na  
    ADD CONSTRAINT se_deli_na_jazyk_fk FOREIGN KEY ( jazyk_id_jazyka )  
        REFERENCES jazyk ( id_jazyka );
```

```
ALTER TABLE se_deli_na  
ADD CONSTRAINT se_deli_na_uroven_fk FOREIGN KEY ( uroven_kod )  
REFERENCES uroven ( kod );
```

```
ALTER TABLE uci  
ADD CONSTRAINT uci_jazyk_fk FOREIGN KEY ( jazyk_id_jazyka )  
REFERENCES jazyk ( id_jazyka );
```

```
ALTER TABLE uci  
ADD CONSTRAINT uci_lektori_fk FOREIGN KEY ( lektori_id_zamestnance )  
REFERENCES lektori ( id_zamestnance );  
ON DELETE SET NULL
```

```
ALTER TABLE uci  
ADD CONSTRAINT uci_zakaznici_fk FOREIGN KEY ( zakaznici_id_zakaznika )  
REFERENCES zakaznici ( id_zakaznika )  
ON DELETE CASCADE;
```

### 4.3 Integritní omezení

Tabulka ZAKAZNICI

*Entitní integrita*

Atributy tvořící primární klíč: ID\_ZAKAZNIKA

SQL kód pro definici primárního klíče:

```
CONSTRAINT zakaznici_pk PRIMARY KEY ( id_zakaznika );
```

*Doménová integrita*

**Sloupec DATUM\_NAROZENI nemůže být uveden v budoucnu**

SQL příslušného omezení:

```
ALTER TABLE zakaznici ADD CONSTRAINT datum_narozeni_chc CHECK( datum_narozeni <  
sysdate);
```

Tabulka JAZYK

*Entitní integrita*

Atributy tvořící primární klíč: ID\_JAZYKA

SQL kód pro definici primárního klíče:

```
CONSTRAINT jazyk_pk PRIMARY KEY ( id_jazyka );
```

*Doménová integrita*

**Sloupec CENA\_LEKCE nemůže mít negativní hodnoty**

Cena jednotlivé lekce má být kladná.

SQL příslušného omezení:

```
ALTER TABLE jazyk ADD CONSTRAINT cena_lekce_chc CHECK ( cena_lekce > 0);
```

Tabulka UROVEN

*Entitní integrita*

Atributy tvořící primární klíč: KOD

SQL kód pro definici primárního klíče:

```
ALTER TABLE uroven ADD CONSTRAINT uroven_pk PRIMARY KEY ( kod );
```

*Doménová integrita*

**Sloupec POČET BODU má hodnoty v rozmezí od 0 do 100**

Před zahájením výuky každý zákazník píše test a podle počtu bodů je mu přiřazena úroveň. Proto v tabulce úroven také zaznamenáváme minimální dosažený počet bodu potřebný pro danou úroveň.

SQL příslušného omezení:

```
ALTER TABLE uroven ADD CHECK ( pocet_bodu BETWEEN 0 AND 100 );
```

Tabulka SE\_DELI\_NA

*Entitní integrita*

Atributy tvořící primární klíč: JAZYK\_ID\_JAZYKA, UROVEN\_KOD

SQL kód pro definici primárního klíče:

```
ALTER TABLE se_deli_na ADD CONSTRAINT se_deli_na_pk PRIMARY KEY ( jazyk_id_jazyka,  
                                                                    uroven_kod );
```

*Referenční integrita*

**Sloupce JAZYK\_ID\_JAZYKA a UROVEN\_KOD v tabulce SE\_DELI\_NA představují cizí klíč**

Vyučování každého jazyka se dělí na úrovně. Každý jazyk má stejné dělení na úrovni.

Druh použitého řešení referenční integrity pro operaci DELETE: RESTRICT

SQL kód příslušného omezení:

```
ALTER TABLE se_deli_na
```

```
ADD CONSTRAINT se_deli_na_jazyk_fk FOREIGN KEY ( jazyk_id_jazyka )
```

```
REFERENCES jazyk ( id_jazyka );
```

```
ALTER TABLE se_deli_na
```

```
ADD CONSTRAINT se_deli_na_uroven_fk FOREIGN KEY ( uroven_kod )
```

```
REFERENCES uroven ( kod );
```

Tabulka ZAMESTNANCI

*Entitní integrita*

Atributy tvořící primární klíč: ID\_ZAMESTNANCE

SQL kód pro definici primárního klíče:

```
ALTER TABLE zamestnanci ADD CONSTRAINT zamestnanci_pk PRIMARY KEY ( id_zamestnance );
```

*Doménová integrita*

**Sloupec DATUM NAROZENI nemůže být uveden v budoucnu**

SQL příslušného omezení:

```
ALTER TABLE zamestnanci ADD CONSTRAINT datum_narozeni_chc CHECK( datum_narozeni < sysdate);
```

Tabulka LEKTORI

*Entitní integrita*

Atributy tvořící primární klíč: ID\_ZAMESTNANCE

SQL kód pro definici primárního klíče:

```
ALTER TABLE lektori ADD CONSTRAINT lektori_pk PRIMARY KEY ( id_zamestnance );
```

*Doménová integrita*

**Sloupec DATUM\_NAROZENI nemůže být uveden v budoucnu**

SQL příslušného omezení:

```
ALTER TABLE lektori ADD CONSTRAINT datum_narozeni_chc CHECK( datum_narozeni < sysdate);
```

**Sloupec PRAXE nemůže mít záporné hodnoty**

SQL příslušného omezení:

```
ALTER TABLE lektori ADD CONSTRAINT praxe_chc CHECK ( praxe > 0);
```

*Referenční integrita*

**Sloupec ID\_ZAMESTNANCE v tabulce LEKTORI představuje cizí klíč**

Tabulka LEKTORI je odvozena z tabulky ZAMESTNANCE, každý lektor je zaměstnancem s funkcí ,lektor', takže musí mít ID\_ZAMESTNANCE.

Druh použitého řešení referenční integrity pro operaci DELETE: RESTRICT

SQL kód příslušného omezení:

```
ALTER TABLE lektori
```

```
ADD CONSTRAINT lektori_zamestnanci_fk FOREIGN KEY ( id_zamestnance )  
REFERENCES zamestnanci ( id_zamestnance );
```

```
ALTER TABLE lektori ADD CONSTRAINT funkce_chc CHECK ( funkce = 'lektor');
```

```
CREATE OR REPLACE TRIGGER funkce_lektori
```

```
BEFORE INSERT OR UPDATE ON lektori FOR EACH ROW
```

```
AS BEGIN
```

```
IF :new.funkce != 'lektor' THEN
```

```
raise_application_error(-20001, 'Funkce musí být lektor!!');
```

```
END IF;
```

END;

#### Tabulka UCI

##### *Entitní integrita*

Atributy tvořící primární klíč: LEKTORI\_ID\_ZAMESTNANCE, JAZYK\_ID\_JAZYKA, ZAKAZNICI\_ID\_ZAKAZNIKA

SQL kód pro definici primárního klíče:

```
ALTER TABLE uci
  ADD CONSTRAINT uci_pk PRIMARY KEY ( lektori_id_zamestnance,
                                     jazyk_id_jazyka,
                                     zakaznici_id_zakaznika );
```

##### *Referenční integrita*

**Sloupce LEKTORI\_ID\_ZAMESTNANCE, JAZYK\_ID\_JAZYKA, ZAKAZNICI\_ID\_ZAKAZNIKA v tabulce UCI představují cizí klíč**

V tabulce UCI jsou představeny jednotlivé zákazníci, je každému přiřazen lektor a je tam informace o jazycích, které učí.

Druh použitého řešení referenční integrity pro operaci DELETE: RESTRICT pro JAZYK\_ID\_JAZYKA, SET NULL pro LEKTORI\_ID\_ZAMESTNANCE, CASCADE pro ZAKAZNICI\_ID\_ZAKAZNIKA

SQL kód příslušného omezení:

```
ALTER TABLE uci
  ADD CONSTRAINT uci_jazyk_fk FOREIGN KEY ( jazyk_id_jazyka )
  REFERENCES jazyk ( id_jazyka );
```

```
ALTER TABLE uci
```

```
  ADD CONSTRAINT uci_lektori_fk FOREIGN KEY ( lektori_id_zamestnance )
  REFERENCES lektori ( id_zamestnance );

  ON DELETE SET NULL
```

```
ALTER TABLE uci
```

```
  ADD CONSTRAINT uci_zakaznici_fk FOREIGN KEY ( zakaznici_id_zakaznika )
  REFERENCES zakaznici ( id_zakaznika )

  ON DELETE CASCADE;
```

#### 4.4 Definice přístupových práv

Definice přístupových práv pro STUDENT:

```
GRANT SELECT ON jazyk TO STUDENT;
```

```
GRANT SELECT ON lektori TO STUDENT;
```

GRANT SELECT ON se\_deli\_na TO STUDENT;

GRANT SELECT ON uci TO STUDENT;

GRANT SELECT ON uroven TO STUDENT;

GRANT SELECT ON zakaznici TO STUDENT;

GRANT SELECT ON zamestnanci TO STUDENT;

Definice přístupových práv pro DB4IT218:

GRANT SELECT, DELETE, INSERT, UPDATE ON jazyk TO DB4IT218;

GRANT SELECT, DELETE, INSERT, UPDATE ON lektori TO DB4IT218;

GRANT SELECT, DELETE, INSERT, UPDATE ON se\_deli\_na TO DB4IT218;

GRANT SELECT, DELETE, INSERT, UPDATE ON uci TO DB4IT218;

GRANT SELECT, DELETE, INSERT, UPDATE ON uroven TO DB4IT218;

GRANT SELECT, DELETE, INSERT, UPDATE ON zakaznici TO DB4IT218;

GRANT SELECT, DELETE, INSERT, UPDATE ON zamestnanci TO DB4IT218;

## 5 Obsah databáze

### 5.1 SQL příkazy pro naplnění databáze daty

```
INSERT INTO ZAKAZNICI (ID_ZAKAZNIKA, JMENO, DATUM_NAROZENI, KONTAKTNI TELEFON) VALUES ('1', 'Anastasia Levchenko', TO_DATE('2002-04-18', 'YYYY-MM-DD'), '778908898')
```

```
INSERT INTO ZAKAZNICI (ID_ZAKAZNIKA, JMENO, DATUM_NAROZENI, KONTAKTNI TELEFON, KONTAKTNI_EMAIL) VALUES ('2', 'Robin Jaensch', TO_DATE('1995-03-02', 'YYYY-MM-DD'), '788164655', 'rjaefa@gmail.com' )
```

```
INSERT INTO ZAKAZNICI (ID_ZAKAZNIKA, JMENO, DATUM_NAROZENI, KONTAKTNI TELEFON, KONTAKTNI_EMAIL) VALUES ('3', 'Felix Donaubauer', TO_DATE('1998-11-11', 'YYYY-MM-DD'), '777029916', 'alter.egon@gmail.com' )
```

```
INSERT INTO JAZYK (ID_JAZYKA, NAZEV, CENA_LEKCE) VALUES ('01', 'Angličtina', '350')
```

```
INSERT INTO JAZYK (ID_JAZYKA, NAZEV, CENA_LEKCE) VALUES ('02', 'Němčina', '300')
```

```
INSERT INTO JAZYK (ID_JAZYKA, NAZEV, CENA_LEKCE) VALUES ('03', 'Francouzština', '300')
```

```
INSERT INTO JAZYK (ID_JAZYKA, NAZEV, CENA_LEKCE) VALUES ('04', 'Španělština', '300')
```

```
INSERT INTO UROVEN (KOD, NAZEV, POČET_BODU) VALUES ('A1', 'Začátečník', '0')
```

```
INSERT INTO UROVEN (KOD, NAZEV, POČET_BODU) VALUES ('A2', 'Pokročilý začátečník', '20')
```

```
INSERT INTO UROVEN (KOD, NAZEV, POČET_BODU) VALUES ('B1', 'Mírně pokročilý', '40')
```

```
INSERT INTO UROVEN (KOD, NAZEV, POČET_BODU) VALUES ('B2', 'Pokročilý', '65')
```

```
INSERT INTO UROVEN (KOD, NAZEV, POČET_BODU) VALUES ('C1', 'Velmi pokročilý', '90')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('1', 'A1')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('1', 'A2')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('1', 'B1')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('1', 'B2')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('1', 'C1')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('2', 'A1')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('2', 'A2')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('2', 'B1')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('2', 'B2')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('2', 'C1')
```

```
INSERT INTO SE_DELI_NA (JAZYK_ID_JAZYKA, UROVEN_KOD) VALUES ('3', 'A1')
```

INSERT INTO SE\_DELI\_NA (JAZYK\_ID\_JAZYKA, UROVEN\_KOD) VALUES ('3', 'A2')

INSERT INTO SE\_DELI\_NA (JAZYK\_ID\_JAZYKA, UROVEN\_KOD) VALUES ('3', 'B1')

INSERT INTO SE\_DELI\_NA (JAZYK\_ID\_JAZYKA, UROVEN\_KOD) VALUES ('3', 'B2')

INSERT INTO SE\_DELI\_NA (JAZYK\_ID\_JAZYKA, UROVEN\_KOD) VALUES ('3', 'C1')

INSERT INTO SE\_DELI\_NA (JAZYK\_ID\_JAZYKA, UROVEN\_KOD) VALUES ('4', 'A1')

INSERT INTO SE\_DELI\_NA (JAZYK\_ID\_JAZYKA, UROVEN\_KOD) VALUES ('4', 'A2')

INSERT INTO SE\_DELI\_NA (JAZYK\_ID\_JAZYKA, UROVEN\_KOD) VALUES ('4', 'B1')

INSERT INTO SE\_DELI\_NA (JAZYK\_ID\_JAZYKA, UROVEN\_KOD) VALUES ('4', 'B2')

INSERT INTO SE\_DELI\_NA (JAZYK\_ID\_JAZYKA, UROVEN\_KOD) VALUES ('4', 'C1')

INSERT INTO ZAMESTNANCI (ID\_ZAMESTNANCE, JMENO, DATUM\_NAROZENI, FUNKCE, KONTAKTNI TELEFON) VALUES ('101', 'Kamila Srp', TO\_DATE('1978-10-16', 'YYYY-MM-DD'), 'účetní', '767778532')

INSERT INTO ZAMESTNANCI (ID\_ZAMESTNANCE, JMENO, DATUM\_NAROZENI, FUNKCE, KONTAKTNI TELEFON) VALUES ('102', 'Florian Fuerstenberg', TO\_DATE('1998-05-04', 'YYYY-MM-DD'), 'lektor', '777778097')

INSERT INTO ZAMESTNANCI (ID\_ZAMESTNANCE, JMENO, DATUM\_NAROZENI, FUNKCE, KONTAKTNI TELEFON) VALUES ('103', 'Diego Johns', TO\_DATE('1988-12-01', 'YYYY-MM-DD'), 'lektor', '766788582')

INSERT INTO ZAMESTNANCI (ID\_ZAMESTNANCE, JMENO, DATUM\_NAROZENI, FUNKCE, KONTAKTNI TELEFON) VALUES ('104', 'Anna Zázvorková', TO\_DATE('1996-10-28', 'YYYY-MM-DD'), 'administrátor', '769878242')

INSERT INTO LEKTORI (ID\_ZAMESTNANCE, PRAXE, JMENO, DATUM\_NAROZENI, FUNKCE, KONTAKTNI TELEFON) VALUES ('102', '2', 'Florian Fuerstenberg', TO\_DATE('1998-05-04', 'YYYY-MM-DD'), 'lektor', '777778097')

INSERT INTO LEKTORI (ID\_ZAMESTNANCE, PRAXE, JMENO, DATUM\_NAROZENI, FUNKCE, KONTAKTNI TELEFON) VALUES ('103', '7', 'Diego Johns', TO\_DATE('1988-12-01', 'YYYY-MM-DD'), 'lektor', '766788582')

INSERT INTO UCI (LEKTORI\_ID\_ZAMESTNANCE, JAZYK\_ID\_JAZYKA, ZAKAZNICI\_ID\_ZAKAZNIKA) VALUES ('102', '02', '1')

INSERT INTO UCI (LEKTORI\_ID\_ZAMESTNANCE, JAZYK\_ID\_JAZYKA, ZAKAZNICI\_ID\_ZAKAZNIKA) VALUES ('102', '01', '1')

INSERT INTO UCI (LEKTORI\_ID\_ZAMESTNANCE, JAZYK\_ID\_JAZYKA, ZAKAZNICI\_ID\_ZAKAZNIKA) VALUES ('103', '04', '2')

INSERT INTO UCI (LEKTORI\_ID\_ZAMESTNANCE, JAZYK\_ID\_JAZYKA, ZAKAZNICI\_ID\_ZAKAZNIKA) VALUES ('103', '03', '3')



```
INSERT INTO UCI (LEKTORI_ID_ZAMESTNANCE, JAZYK_ID_JAZYKA, ZAKAZNICI_ID_ZAKAZNIKA)
VALUES ('103', '04', '3')
```

## 5.2 Opis vložených dat

V této části uveďte opis dat uložených ve Vámi vytvořené databázi. Je třeba, aby zde byl uveden opis dat všech relačních tabulek ve Vaší databázi. Pro každou relační tabulku uveďte samostatnou podkapitulu. V této podkapitole pak uveďte tabulku, která bude svojí strukturou a obsahem odpovídat příslušné relační tabulce.

Tabulka ZAKAZNICI

ID_ZAKAZNIKA	JMENO	DATUM_NAROZENI	KONTAKTNI_TELEFON	KONTAKTNI_EMAIL
1	Anastasia Levchenko	2002-04-18	778908898	NULL
2	Robin Jaensch	1995-03-02	788164655	rjaefa@gmail.com
3	Felix Donaubauer	1998-11-11	777029916	alter.egon@gmail.com

Tabulka JAZYK

ID_JAZYKA	NAZEV	CENA_LEKCE
01	Angličtina	350
02	Němčina	300
03	Francouzština	300
04	Španělština	300

Tabulka UROVEN

KOD	NAZEV	POČET_BODU
A1	Začátečník	0
A2	Pokročilý začátečník	20
B1	Mírně pokročilý	40
B2	Pokročilý	65
C1	Velmi pokročilý	90

Tabulka SE\_DELI\_NA

JAZYK_ID_JAZYKA	UROVEN_KOD
01	A1
01	A2
01	B1
01	B2
01	C1
02	A1
02	A2
02	B1
02	B2
02	C1
03	A1
03	A2
03	B1

03	B2
03	C1
04	A1
04	A2
04	B1
04	B2
04	C1

Tabulka ZAMESTNANCI

ID_ZAMESTNANCE	JMENO	DATUM_NAROZENI	FUNKCE	KONTAKTNI_TELEFON
101	Kamila Srp	1978-10-16	účetní	767778532
102	Florian Fuerstenberg	1998-05-04	lektor	777778097
103	Diego Johns	1988-12-01	lektor	766788582
104	Anna Zázvorková	1996-10-28	administrátor	769878242

Tabulka LEKTORI

ID_ZAMESTNANCE	PRAXE	JMENO	DATUM_NAROZENI	FUNKCE	KONTAKTNI_TELEFON
102	2	Florian Fuerstenberg	1998-05-04	lektor	777778097
103	7	Diego Johns	1988-12-01	lektor	766788582

Tabulka UCI

LEKTORI_ID_ZAMESTNANCE	JAZYK_ID_JAZYKA	ZAKAZNICI_ID_ZAKAZNIKA
102	02	1
102	01	1
103	04	2
103	03	3
103	04	3