

# Slovník

*Oracle + Java*

Datum: 24. května 2006  
Předmět: IT\_360 – Databáze  
Autor: František Kučera

## Úvod

Toto je dokumentace k datovému modelu a aplikaci – k jednoduchému slovníku.

### **Popis reality**

Cílem je vytvoření jednoduchého vícejazyčného slovníku. Slova mohou být překládána mezi různými jazyky. U slov můžeme sledovat výslovnost a můžeme si je zařadit do nějaké kategorie (hovorová, odborná...). U překladu slova zaznamenáváme i v jakém smyslu slovo překládáme (slovní druh), protože podle toho se může lišit význam v druhém jazyce. Slovo jednoho jazyka může mít v druhém více odpovídajících překladů. U překladů zaznamenáváme také prioritu – jestli mají slova totožný význam, nebo vzdálenější. Jednotliví uživatelé slovníku si mohou označovat slova jako oblíbená.

### **Zpřesnění popisu**

Sledované vlastnosti:

- Jazyk: kód jazyka, název, příslovce, druhý pád
- Kategorie: zkratka, název, popis
- Oblíbené: uživatel, slovo, jazyk
- Překlad: zdrojové slovo, cílové slovo, zdrojový jazyk, cílový jazyk, slovní druh, priorita
- Slovní druh: číslo, český název, latinský název
- Slovo: text, jazyk, kategorie, výslovnost
- Uživatel: přihlašovací jméno, heslo, jméno, příjmení

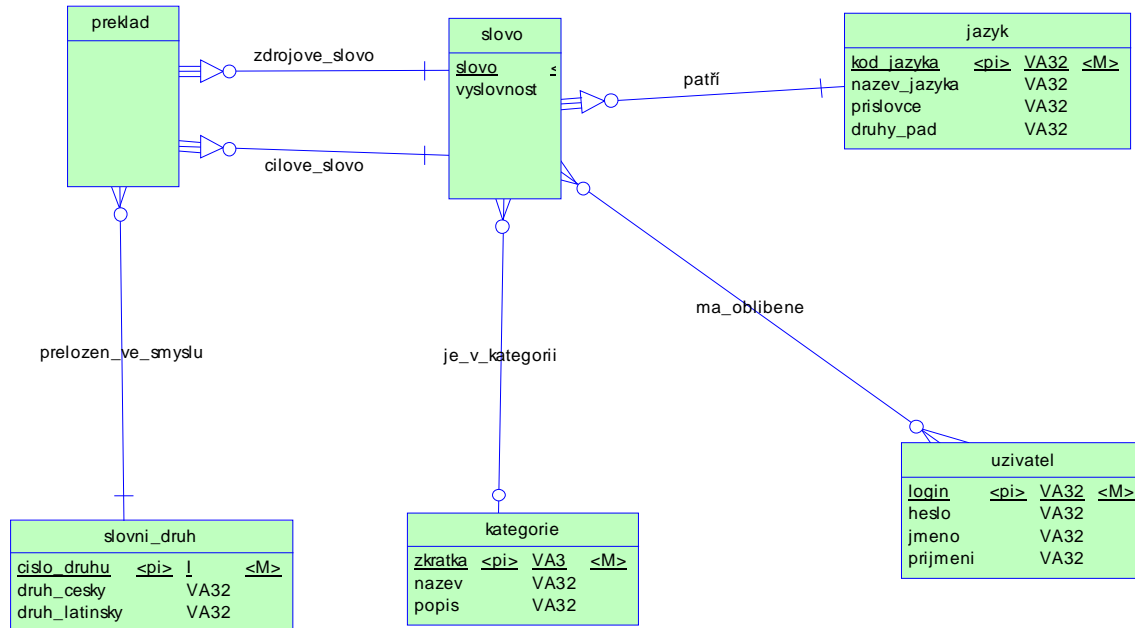
Každé slovo patří právě do jednoho jazyka. Do jednoho jazyka může patřit více slov. Slovo může ale nemusí patřit do jedné kategorie, kategorie může být přiřazena žádnému nebo více slovům. Uživatel může mít oblíbeno více slov, ale taky žádné. Jedno slovo může být oblíbeno více uživateli, ale nemusí být žádným. Slovo jednoho jazyka může být přeloženo do více jiných jazyků. Může existovat i slovo bez překladu. U překladu musí být uveden slovní druh (některá slova mají různý překlad podle toho, jestli jsou to např. slovesa nebo podstatná jména, i když se stejně píšou). Překlady se mohou lišit v závislosti na směru překladu, pokud přeložíme slovo např. z češtiny do angličtiny a potom zpátky, nemusíme dostat totéž slovo (a nemusíme dostat vůbec žádné). Nelze přeložit slovo z jednoho jazyka do téhož (např. překlad z češtiny do češtiny).

Slovník je jednoduchý, proto neřeší některé věci:

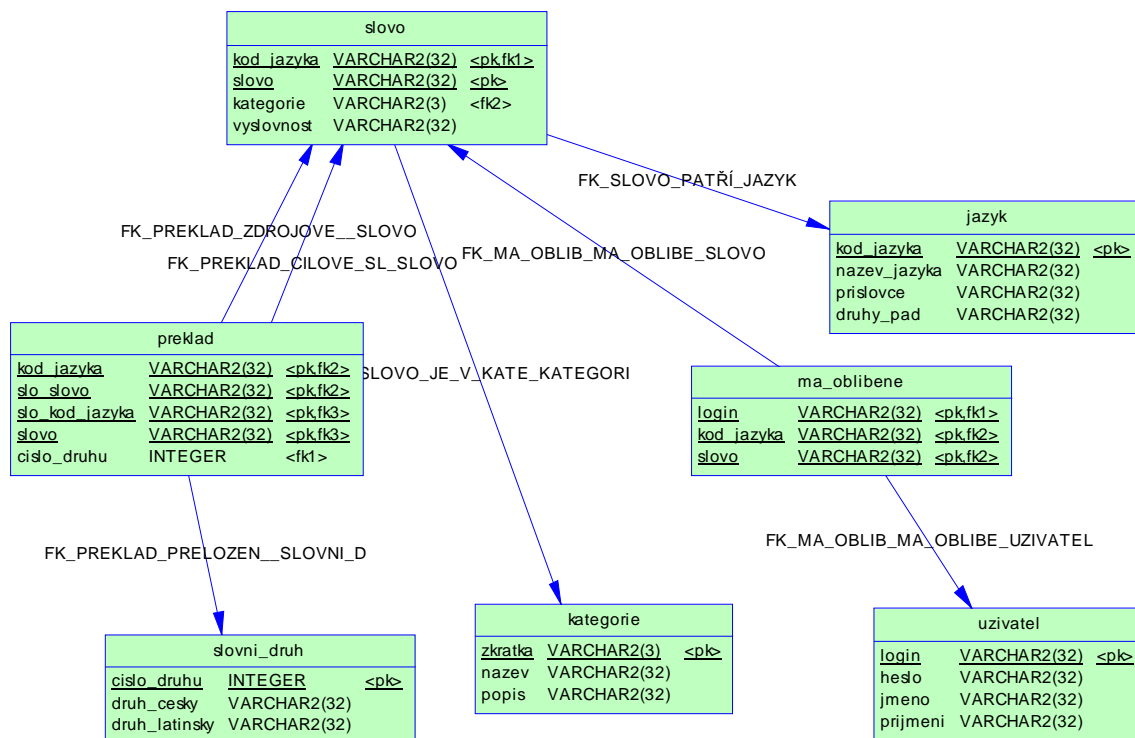
- Předpokládá se, že jedno slovo (skupina písmen) v jazyce se vyslovuje vždy stejně
- Nesledujeme životnost, rody podstatných jmen, stupně příslovcí a přídavných jmen

# Datové modely

## Konceptuální model



## Fyzický datový model



### **Integritní omezení**

- Jazyk – primárním klíčem je kód (např. cz, de, fr, en)
- Kategorie – primárním klíčem je zkratka
- Oblíbené – je to vazební tabulka, primární klíč je složen ze tří sloupců a jsou to zároveň cizí klíče: login uživatele, kód jazyka a slovo
- Překlad – druhá vazební tabulka, primární klíč je složen ze čtyř sloupců, které jsou zároveň cizími klíči: první slovo, druhé slovo, první jazyk a druhý jazyk
- Slovní druh – primárním klíčem je číslo (1-10)
- Slovo – primárním klíčem je slovo a kód jazyka. Kód jazyka je zároveň cizím klíčem. Druhým cizím klíčem je zkratka kategorie
- Uživatel – primárním klíčem je login – přihlašovací jméno
- U překladu se kontroluje, zda není zdrojový jazyk stejný jako cílový.
- Při smazání slova nebo uživatele se smaží i odpovídající oblíbená slova
- Při smazání slova se smažou i jeho překlady
- Nelze smazat slovní druh, dokud jsou v něm nějaká slova

## **Vytvoření databáze**

### **Tabulky**

```
CREATE TABLE "XKUCF03"."JAZYK"
(
  "KOD_JAZYKA" CHAR(2 CHAR) NOT NULL ENABLE,
  "NAZEV_JAZYKA" VARCHAR2(32 BYTE) NOT NULL ENABLE,
  "PRI_SLOVCE" VARCHAR2(32 BYTE) NOT NULL ENABLE,
  "DRUHY_PAD" VARCHAR2(32 BYTE) NOT NULL ENABLE,
  CONSTRAINT "JAZYK_PK" PRIMARY KEY ("KOD_JAZYKA") ENABLE
);

CREATE TABLE "XKUCF03"."KATEGORIE"
(
  "ZKRATKA" VARCHAR2(16 BYTE),
  "NAZEV" VARCHAR2(32 BYTE) NOT NULL ENABLE,
  "POPI" VARCHAR2(128 BYTE),
  CONSTRAINT "KATEGORIE_PK" PRIMARY KEY ("ZKRATKA") ENABLE
);

CREATE TABLE "XKUCF03"."OBLIBENE"
(
  "LOGIN" VARCHAR2(32 BYTE) NOT NULL ENABLE,
  "SLOVO" VARCHAR2(32 BYTE) NOT NULL ENABLE,
  "KOD_JAZYKA" CHAR(2 BYTE) NOT NULL ENABLE,
  CONSTRAINT "OBLIBENE_PK" PRIMARY KEY ("LOGIN", "SLOVO",
  "KOD_JAZYKA") ENABLE,
  CONSTRAINT "OBLIBENE_LOGIN" FOREIGN KEY ("LOGIN")
  REFERENCES "XKUCF03"."UZIVATEL" ("LOGIN") ON DELETE CASCADE ENABLE,
  CONSTRAINT "OBLIBENE_SLOVO" FOREIGN KEY ("SLOVO", "KOD_JAZYKA")
  REFERENCES "XKUCF03"."SLOVO" ("SLOVO", "KOD_JAZYKA") ON DELETE
  CASCADE ENABLE
);

CREATE TABLE "XKUCF03"."PREKLAD"
(
  "SLOVO_ZDROJ" VARCHAR2(32 BYTE) NOT NULL ENABLE,
  "SLOVO_CIL" VARCHAR2(32 BYTE) NOT NULL ENABLE,
  "JAZYK_ZDROJ" CHAR(2 BYTE) NOT NULL ENABLE,
  "JAZYK_CIL" CHAR(2 BYTE) NOT NULL ENABLE,
  "SLOVNI_DRUH" NUMBER NOT NULL ENABLE,
  "PRIORITA" NUMBER(2, 0) NOT NULL ENABLE,
  CONSTRAINT "PREKLAD_PK" PRIMARY KEY ("SLOVO_ZDROJ", "SLOVO_CIL",
  "JAZYK_ZDROJ", "JAZYK_CIL", "SLOVNI_DRUH") ENABLE,
```

```

        CONSTRAINT "PREKLAD_RUZNE_JAZYKY" CHECK (JAZYK_ZDROJ != JAZYK_CIL)
ENABLE,
        CONSTRAINT "PREKLAD_ZDROJ" FOREIGN KEY ("SLOVO_ZDROJ",
"JAZYK_ZDROJ")
        REFERENCES "XKUCF03"."SLOVO" ("SLOVO", "KOD_JAZYKA") ON DELETE
CASCADE ENABLE,
        CONSTRAINT "PREKLAD_CIL" FOREIGN KEY ("SLOVO_CIL", "JAZYK_CIL")
        REFERENCES "XKUCF03"."SLOVO" ("SLOVO", "KOD_JAZYKA") ON DELETE
CASCADE ENABLE,
        CONSTRAINT "PREKLAD_SLOVNI_DRUH" FOREIGN KEY ("SLOVNI_DRUH")
        REFERENCES "XKUCF03"."SLOVNI_DRUH" ("CISLO_DRUHU") ENABLE
    );

CREATE TABLE "XKUCF03"."SLOVNI_DRUH"
(
    "CISLO_DRUHU" NUMBER NOT NULL ENABLE,
    "DRUH_CESKY" VARCHAR2(32 BYTE) NOT NULL ENABLE,
    "DRUH_LATINSKY" VARCHAR2(32 BYTE) NOT NULL ENABLE,
    CONSTRAINT "SLOVNI_DRUH_PK" PRIMARY KEY ("CISLO_DRUHU") ENABLE
);

CREATE TABLE "XKUCF03"."SLOVO"
(
    "SLOVO" VARCHAR2(32 BYTE),
    "KOD_JAZYKA" CHAR(2 CHAR) NOT NULL ENABLE,
    "KATEGORIE" VARCHAR2(16 BYTE),
    "VYSLOVNOST" VARCHAR2(32 BYTE),
    CONSTRAINT "SLOVO_PK" PRIMARY KEY ("SLOVO", "KOD_JAZYKA") ENABLE,
    CONSTRAINT "SLOVO_KOD_JAZYKA" FOREIGN KEY ("KOD_JAZYKA")
        REFERENCES "XKUCF03"."JAZYK" ("KOD_JAZYKA") ENABLE,
    CONSTRAINT "SLOVO_KATEGORIE" FOREIGN KEY ("KATEGORIE")
        REFERENCES "XKUCF03"."KATEGORIE" ("ZKRATKA") ENABLE
);

CREATE TABLE "XKUCF03"."UZIVATEL"
(
    "LOGIN" VARCHAR2(32 BYTE) NOT NULL ENABLE,
    "HESLO" VARCHAR2(64 BYTE) NOT NULL ENABLE,
    "JMENO" VARCHAR2(32 BYTE) NOT NULL ENABLE,
    "PRIJMENI" VARCHAR2(32 BYTE) NOT NULL ENABLE,
    CONSTRAINT "UZIVATEL_PK" PRIMARY KEY ("LOGIN") ENABLE
);

```

### ***Integritní omezení***

Integritu se podařilo zajistit prostředky na úrovni databáze, proto nebylo nutné na úrovni aplikace implementovat žádná omezení. Data je tak možné vkládat a upravovat přímo zásahy do databáze.

```

CREATE UNIQUE INDEX "XKUCF03"."JAZYK_PK" ON "XKUCF03"."JAZYK"
("KOD_JAZYKA");

CREATE UNIQUE INDEX "XKUCF03"."KATEGORIE_PK" ON "XKUCF03"."KATEGORIE"
("ZKRATKA");

CREATE UNIQUE INDEX "XKUCF03"."OBLIBENE_PK" ON "XKUCF03"."OBLIBENE"
("LOGIN", "SLOVO", "KOD_JAZYKA");

CREATE UNIQUE INDEX "XKUCF03"."PREKLAD_PK" ON "XKUCF03"."PREKLAD"
("SLOVO_ZDROJ", "SLOVO_CIL", "JAZYK_ZDROJ", "JAZYK_CIL", "SLOVNI_DRUH")

CREATE UNIQUE INDEX "XKUCF03"."SLOVNI_DRUH_PK" ON "XKUCF03"."SLOVNI_DRUH"
("CISLO_DRUHU")

CREATE UNIQUE INDEX "XKUCF03"."SLOVO_PK" ON "XKUCF03"."SLOVO" ("SLOVO",
"KOD_JAZYKA")

```

```
CREATE UNIQUE INDEX "XKUCF03"."UZI VATEL_PK" ON "XKUCF03"."UZI VATEL"
("LOGIN")
```

+ kontrola jestli se neshoduje zdrojový a cílový jazyk – viz vytvoření tabulky „preklad“:  
 CHECK (JAZYK\_ZDROJ != JAZYK\_CIL)

### **Přístupová práva**

```
GRANT SELECT ON "XKUCF03"."JAZYK" TO "STUDENT";
GRANT SELECT ON "XKUCF03"."KATEGORIE" TO "STUDENT";
GRANT SELECT ON "XKUCF03"."OBLIBENE" TO "STUDENT";
GRANT SELECT ON "XKUCF03"."PREKLAD" TO "STUDENT";
GRANT SELECT ON "XKUCF03"."SLOVNI_DRUH" TO "STUDENT";
GRANT SELECT ON "XKUCF03"."SLOVO" TO "STUDENT";
GRANT SELECT ON "XKUCF03"."UZI VATEL" TO "STUDENT";

GRANT SELECT, INSERT, UPDATE, DELETE ON "XKUCF03"."JAZYK" TO "IT_360";
GRANT SELECT, INSERT, UPDATE, DELETE ON "XKUCF03"."KATEGORIE" TO "IT_360";
GRANT SELECT, INSERT, UPDATE, DELETE ON "XKUCF03"."OBLIBENE" TO "IT_360";
GRANT SELECT, INSERT, UPDATE, DELETE ON "XKUCF03"."PREKLAD" TO "IT_360";
GRANT SELECT, INSERT, UPDATE, DELETE ON "XKUCF03"."SLOVNI_DRUH" TO
"IT_360";
GRANT SELECT, INSERT, UPDATE, DELETE ON "XKUCF03"."SLOVO" TO "IT_360";
GRANT SELECT, INSERT, UPDATE, DELETE ON "XKUCF03"."UZI VATEL" TO "IT_360";

GRANT EXECUTE ON zkontroluj_heslo TO student;
GRANT EXECUTE ON pridej_oblibene TO student;
GRANT EXECUTE ON odeber_oblibene TO student;

GRANT EXECUTE ON zkontroluj_heslo TO it_360;
GRANT EXECUTE ON pridej_oblibene TO it_360;
GRANT EXECUTE ON odeber_oblibene TO it_360;

GRANT SELECT ON "XKUCF03"."UZI VATEL_BEZ_HESLA" TO "IT_360";
GRANT SELECT ON "XKUCF03"."UZI VATEL_BEZ_HESLA" TO "STUDENT";
```

### **Další objekty**

Databáze obsahuje také dvě procedury a jeden pohled (viz kapitola bezpečnost):

```
create or replace PROCEDURE odeber_oblibene(login_uzivatel
VARCHAR2, heslo_uzivatel VARCHAR2, slovo_nove VARCHAR2, jazyk CHAR)
AS
BEGIN
  DECLARE pocet_uzivatelu NUMBER;
  BEGIN
    SELECT COUNT(*)
    INTO pocet_uzivatelu
    FROM
      (SELECT *
      FROM uzivatel
      WHERE login = login_uzivatel
      AND heslo = heslo_uzivatel)
    ;

    IF pocet_uzivatelu = 1 THEN

      DELETE FROM oblibene
      WHERE login = login_uzivatel
      AND slovo = slovo_nove
      AND kod_jazyka = jazyk;
    ELSE
      raise_application_error(-20001, 'Špatné heslo. Co to zkoušíš???');
    END IF;

  END;
END;
```

```

create or replace PROCEDURE pridej_oblibene(login_uzivatele VARCHAR2,
heslo_uzivatele VARCHAR2, slovo VARCHAR2, jazyk CHAR) AS
BEGIN
    DECLARE pocet_uzivatelu NUMBER;
    BEGIN
        SELECT COUNT(*)
        INTO pocet_uzivatelu
        FROM
            (SELECT *
            FROM uzivatel
            WHERE login = login_uzivatele
            AND heslo = heslo_uzivatele)
        ;

        IF pocet_uzivatelu = 1 THEN
            INSERT
            INTO oblibene
            VALUES(login_uzivatele, slovo, jazyk);
        ELSE
            raise_application_error(-20001, 'Špatné heslo. Co to zkoušíš???');
        END IF;

    END;
END;

create or replace PROCEDURE zkontroluj_heslo(login_uzivatele IN VARCHAR2,
heslo_uzivatele IN VARCHAR2, jmeno_uzivatele OUT VARCHAR2,
prijmeni_uzivatele OUT VARCHAR2) AS
r NUMBER;
type ref_cur IS ref CURSOR;
u ref_cur;
BEGIN
    SELECT COUNT(*)
    INTO r
    FROM uzivatel
    WHERE login = login_uzivatele
    AND heslo = heslo_uzivatele;

    IF r = 1 THEN

        OPEN u FOR --- já vím, šlo by to napsat i líp :-)
        SELECT jmeno,
        prijmeni
        FROM uzivatel
        WHERE login = login_uzivatele
        AND heslo = heslo_uzivatele;
        FETCH u
        INTO jmeno_uzivatele,
        prijmeni_uzivatele;

        CLOSE u;
    ELSE
        raise_application_error(-20001, 'Špatné heslo nebo neexistující
        uživatel');
    END IF;

END;

CREATE OR REPLACE FORCE VIEW "XKUCF03"."UZIVATEL_BEZ_HESLA" ("LOGIN",
"JMENO", "PRIJMENI") AS
select login, jmeno, prijmeni from uzivatel
WITH READ ONLY;

```

### **Vložení dat**

```

-- INSERTING into JAZYK
Insert into JAZYK ("KOD_JAZYKA", "NAZEV_JAZYKA", "PRI SLOVCE", "DRUHY_PAD")
values ('de', 'němčina', 'německy', 'němčiny');

```

```

Insert into JAZYK ("KOD_JAZYKA", "NAZEV_JAZYKA", "PRI_SLOVCE", "DRUHY_PAD")
values ('fr', 'francouzština', 'francouzsky', 'francouzštiny');
Insert into JAZYK ("KOD_JAZYKA", "NAZEV_JAZYKA", "PRI_SLOVCE", "DRUHY_PAD")
values ('en', 'angličtina', 'anglicky', 'angličtiny');
Insert into JAZYK ("KOD_JAZYKA", "NAZEV_JAZYKA", "PRI_SLOVCE", "DRUHY_PAD")
values ('cz', 'čeština', 'česky', 'češtiny');

-- INSERTING into KATEGORIE
Insert into KATEGORIE ("ZKRATKA", "NAZEV", "POPI_S") values
('bás', 'básnický', 'básnický výraz');
Insert into KATEGORIE ("ZKRATKA", "NAZEV", "POPI_S") values
('krim', 'kriminalistický', 'kriminalistika, kriminalistický');
Insert into KATEGORIE ("ZKRATKA", "NAZEV", "POPI_S") values
('cit', 'citový', 'citový výraz');
Insert into KATEGORIE ("ZKRATKA", "NAZEV", "POPI_S") values
('hovor', 'hovorový', 'hovorově');
Insert into KATEGORIE ("ZKRATKA", "NAZEV", "POPI_S") values
('zast', 'zastaralý', 'dnes málo používaný');
Insert into KATEGORIE ("ZKRATKA", "NAZEV", "POPI_S") values
('kniž', 'knižní', 'používaný převážně v knížkách');
Insert into KATEGORIE ("ZKRATKA", "NAZEV", "POPI_S") values
('slang', 'slangový', 'slangový výraz');

-- INSERTING into OBLIBENE
Insert into OBLIBENE ("LOGIN", "SLOVO", "KOD_JAZYKA") values
('host', 'chat', 'fr');
Insert into OBLIBENE ("LOGIN", "SLOVO", "KOD_JAZYKA") values
('host', 'chien', 'fr');
Insert into OBLIBENE ("LOGIN", "SLOVO", "KOD_JAZYKA") values
('host', 'kočka', 'cz');
Insert into OBLIBENE ("LOGIN", "SLOVO", "KOD_JAZYKA") values
('host', 'pes', 'cz');

-- INSERTING into PREKLAD
Insert into PREKLAD
("SLOVO_ZDROJ", "SLOVO_CIL", "JAZYK_ZDROJ", "JAZYK_CIL", "SLOVNI_DRUH", "PRIORITA")
values ('Kneipe', 'hospoda', 'de', 'cz', '1', '0');
Insert into PREKLAD
("SLOVO_ZDROJ", "SLOVO_CIL", "JAZYK_ZDROJ", "JAZYK_CIL", "SLOVNI_DRUH", "PRIORITA")
values ('Kneipe', 'putyka', 'de', 'cz', '1', '2');
Insert into PREKLAD
("SLOVO_ZDROJ", "SLOVO_CIL", "JAZYK_ZDROJ", "JAZYK_CIL", "SLOVNI_DRUH", "PRIORITA")
values ('Kneipe', 'pajzl', 'de', 'cz', '1', '3');
Insert into PREKLAD
("SLOVO_ZDROJ", "SLOVO_CIL", "JAZYK_ZDROJ", "JAZYK_CIL", "SLOVNI_DRUH", "PRIORITA")
values ('pes', 'dog', 'cz', 'en', '1', '0');
Insert into PREKLAD
("SLOVO_ZDROJ", "SLOVO_CIL", "JAZYK_ZDROJ", "JAZYK_CIL", "SLOVNI_DRUH", "PRIORITA")
values ('kočka', 'chat', 'cz', 'fr', '1', '0');
Insert into PREKLAD
("SLOVO_ZDROJ", "SLOVO_CIL", "JAZYK_ZDROJ", "JAZYK_CIL", "SLOVNI_DRUH", "PRIORITA")
values ('dům', 'maison', 'cz', 'fr', '1', '0');
Insert into PREKLAD
("SLOVO_ZDROJ", "SLOVO_CIL", "JAZYK_ZDROJ", "JAZYK_CIL", "SLOVNI_DRUH", "PRIORITA")
values ('dog', 'pes', 'en', 'cz', '1', '1');

-- INSERTING into SLOVNI_DRUH
Insert into SLOVNI_DRUH ("CI_SLO_DRUHU", "DRUH_CESKY", "DRUH_LATI_NSKY") values
('10', 'citoslovce', 'interjekce');
Insert into SLOVNI_DRUH ("CI_SLO_DRUHU", "DRUH_CESKY", "DRUH_LATI_NSKY") values
('9', 'částice', 'partikule');
Insert into SLOVNI_DRUH ("CI_SLO_DRUHU", "DRUH_CESKY", "DRUH_LATI_NSKY") values
('8', 'spojky', 'konjunkce');
Insert into SLOVNI_DRUH ("CI_SLO_DRUHU", "DRUH_CESKY", "DRUH_LATI_NSKY") values
('7', 'předložky', 'prepozice');
Insert into SLOVNI_DRUH ("CI_SLO_DRUHU", "DRUH_CESKY", "DRUH_LATI_NSKY") values
('6', 'příslovce', 'adverbia');
Insert into SLOVNI_DRUH ("CI_SLO_DRUHU", "DRUH_CESKY", "DRUH_LATI_NSKY") values
('5', 'slovesa', 'verba');

```



```

Insert into SLOVNI_DRUH ("CI SLO_DRUHU", "DRUH_CESKY", "DRUH_LATI NSKY") val ues
(' 4', 'čí slovky', 'numeralia');
Insert into SLOVNI_DRUH ("CI SLO_DRUHU", "DRUH_CESKY", "DRUH_LATI NSKY") val ues
(' 3', 'zájmena', 'pronomina');
Insert into SLOVNI_DRUH ("CI SLO_DRUHU", "DRUH_CESKY", "DRUH_LATI NSKY") val ues
(' 2', 'přídavná jména', 'adjektiva');
Insert into SLOVNI_DRUH ("CI SLO_DRUHU", "DRUH_CESKY", "DRUH_LATI NSKY") val ues
(' 1', 'podstatná jména', 'substantiva');

-- INSERTING into SLOVO
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' Knei pe', 'de', 'slang', 'Knei pe');
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' paj zl', 'cz', 'hovor', null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' putyka', 'cz', null, null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' hospoda', 'cz', null, null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' dům', 'cz', null, null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' pes', 'cz', null, null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' kočka', 'cz', null, null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' Haus', 'de', null, null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' Katze', 'de', null, null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' Hund', 'de', null, null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' cat', 'en', null, 'ket');
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' dog', 'en', null, 'dog');
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' house', 'en', null, 'haus');
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' chien', 'fr', null, null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' chat', 'fr', null, null);
Insert into SLOVO ("SLOVO", "KOD_JAZYKA", "KATEGORIE", "VYSLOVNOST") val ues
(' mai son', 'fr', null, null);

-- INSERTING into UZIVATEL
Insert into UZIVATEL ("LOGIN", "HESLO", "JMENO", "PRI JMENI") val ues
(' xkucf03', 'heslo', 'František', 'Kučera');
Insert into UZIVATEL ("LOGIN", "HESLO", "JMENO", "PRI JMENI") val ues
(' host', 'host', 'Zkušební', 'Uživatel');
Insert into UZIVATEL ("LOGIN", "HESLO", "JMENO", "PRI JMENI") val ues
(' ni kdo', 'ni c', 'Druhý', 'Uživatel');

```

## Bezpečnost

Vlastníkem všech objektů aplikace je XKUCF03 (já). Pro provoz aplikace ale stačí (je to vhodnější) i méně oprávněný uživatel (třeba student), který není vlastníkem objektů a má práva jen pro čtení.

Aby bylo možné ověřit heslo k aplikaci (jiné než k DB, slouží pro rozpoznání uživatelů aplikace, kteří si mohou přidávat slova mezi oblíbená), je v databázi procedura zkontroluj\_heslo – ta vrací při úspěchu jméno a příjmení uživatele a při neúspěchu chybu „- 20001, 'Špatné heslo nebo neexistující uživatel'“

Díky tomu je možné odejmout uživateli právo čtení na tabulku uzivatel a zabránit mu tak ve čtení hesel ostatních uživatelů. Zatím ale toto právo má, protože to bylo v zadání. Lze ho odejmout: `REVOKE SELECT ON uzivatel FROM student;`

Aplikace zobrazuje informace o počtu vložených slov a překladů a také o počtu uživatelů. Aby tento počet mohl zjistit i obyčejný uživatel a nemusel mít právo pro čtení tabulky `uzivatel` (aby neviděl hesla ostatních), je v databázi pohled `UZIVATEL_BEZ_HESLA`, který obsahuje všechny sloupce kromě hesla. Pokud právo k pohledu chybí, aplikace bude fungovat, ale nebude zobrazovat informace o počtech slov a uživatelů.

Aby si i obyčejný uživatel DB (student), který má jméno a heslo k aplikaci (je v tabulce `uzivatel`), mohl přidávat a odebírat oblíbená slova, jsou v databázi další dvě procedury: `odeber_obli_bene` a `pri_dej_obli_bene`. Tyto procedury kontrolují jméno a heslo (podle tabulky `uzivatel`) a oprávněným uživatelům dovolují přidávat a odebírat jejich oblíbená slova. Pokud se procedurám nepodaří ověřit uživatele, dojde k výjimce „- 20001, ' Špatné heslo. Co to zkoušíš???' “. K této výjimce by ale nemělo docházet, protože uživatel se už jednou ověřil při spouštění aplikace. Výjimka tak může nastat pouze ve dvou případech:

- Po přihlášení k aplikaci se změnil záznam v tabulce `uzivatel` a uživatel tak přestal existovat nebo má nové heslo.
- Uživatel spouští proceduru jinak než z aplikace a tudíž tam mohl ručně zadat špatné jméno a heslo

Tabulka `obli_bene` je pro čtení přístupná, je tak možné vidět jaká slova mají ostatní mezi oblíbenými. Protože to není důvěrná informace, není to chyba.

## Závěr

Tento jednoduchý počítačový překladový slovník vzniknul jako semestrální práce pro předmět IT\_360 – Databáze. Aplikace nebyla testována ani optimalizována pro velké objemy dat. SQL výjimky jsou ošetřeny, ale z výukových důvodů nejsou „zaobaleny“ do uživatelsky přívětivé formy, uživatel se proto zobrazují přímo chybové hlášky databázového stroje, aby bylo vidět, na jaká omezení jsme narazili.