



Příklady k přednášce

PV003 – Architektura relačních databází

ZADÁNÍ PŘÍKLADU PRO PV003	1
<u>PŘÍKLAD 1</u>.....	2
<u>PŘÍKLAD 2</u>.....	3
<u>PŘÍKLAD 3</u>.....	4
PŘÍKLAD 4.....	6
PŘÍKLAD 5.....	9
PŘÍKLAD 6.....	10
PŘÍKLAD 7.....	11
PŘÍKLAD 8.....	11
PŘÍKLAD 10.....	11

ZADÁNÍ PŘÍKLADU PRO PV003 ¹

- 1) Katastr nemovitostí vede údaje o vlastnictví pozemků (parcel) a budov.
- 2) Nemovitost může vlastnit „oprávněný subjekt“ a to i částečně.
- 3) Oprávněný subjekt může vlastnit více nemovitostí.
- 4) Oprávněný subjekt je buď fyzická osoba, právnická osoba nebo společné jmění manželů.
- 5) Právnická osoba je identifikována IČOem, fyzická osoba je identifikována rodným číslem.
- 6) Rodné číslo má strukturu buď RRMMDccc nebo RRMMDcccc. Pokud je rodné číslo desetimístné, potom je dělitelné 11.
- 7) Parcela je identifikována kódem katastrálního území (státní číselník), typem parcely (stavební, nestavební), parcelním číslem, a podlomením.
- 8) Budova je identifikována kódem části obce (státní číselník) a číslem popisným, resp. evidenčním.
- 9) Obec se skládá z 1, nebo více částí obce.
- 10) Obec se skládá z 1, nebo více katastrálních území.
- 11) Části obce a katastrální území nejsou skladebné.
- 12) Vlastnické vztahy jsou vedeny na listech vlastnictví a to tak, že v části A jsou uvedeny oprávněné subjekty se podílem vlastnictví ke všem nemovitostem, které jsou uvedeny v části B. např:

Okres: 3209 Praha - východ
 Obec: 078 Říčany
 Kat. území 01 Říčany u Prahy
 Číslo LV 365

A Vlastník (pořadí jméno ,název a adresa)	Identifikátor	Podíl
1 Josef Novák Podlesná 11, Brno	581205/3256	1/2
2 Pavel Novák Sady 15, Olomouc	351231/454	1/2

B Parcelní číslo	Výměra m2	Část obce číslo. Budovy	Druh
1452/2	4584		louka		
1452/3	592		orná půda		

.

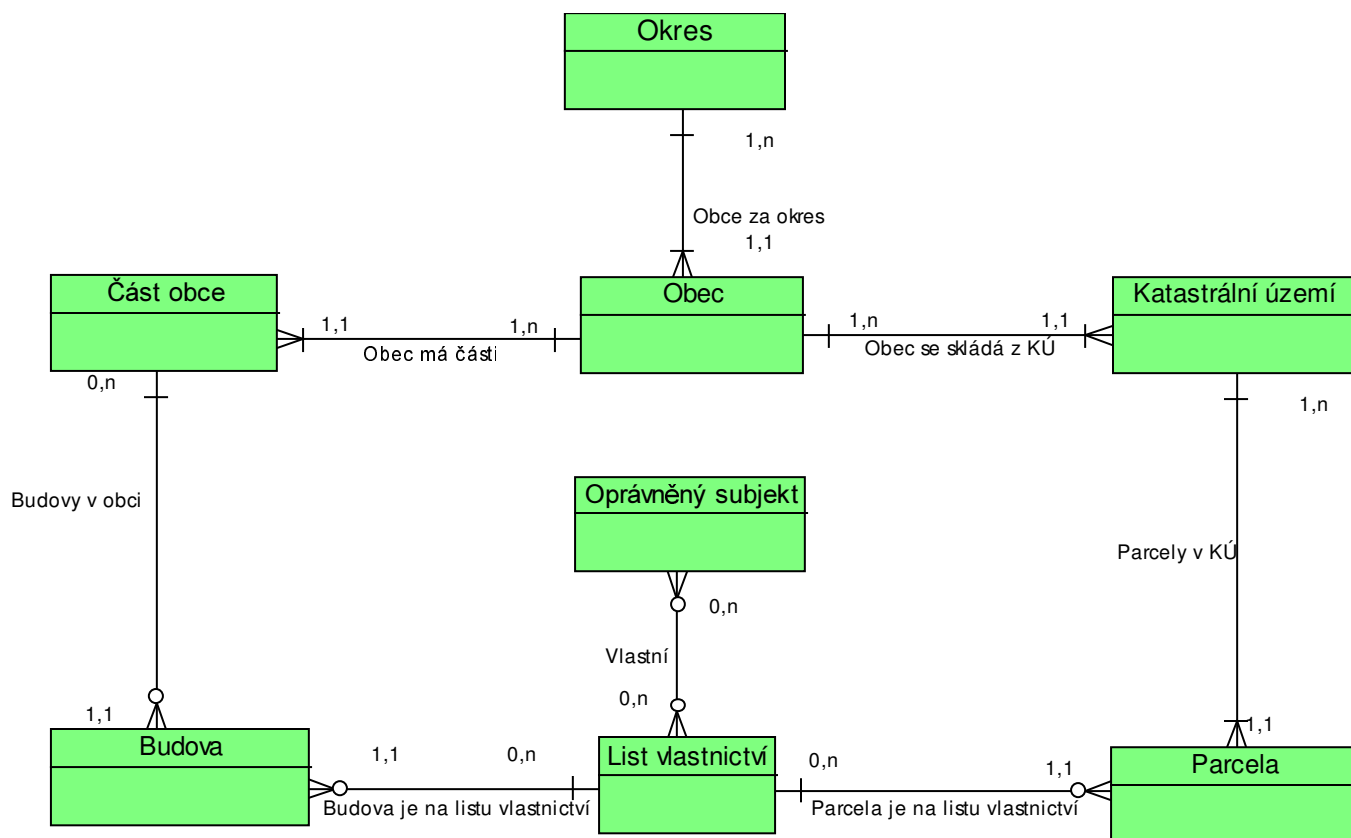
.

- 13) Součet podílů na listu vlastnictví je 1.
- 14) Na listu vlastnictví je uvedena alespoň jedna nemovitost.
- 15) Nemovitost je uvedena právě na jednom listu vlastnictví.

¹ Příklady jsou řešeny v rámci RDBMS ORACLE – pro jiné SQL databáze tato řešení nemusí platit.

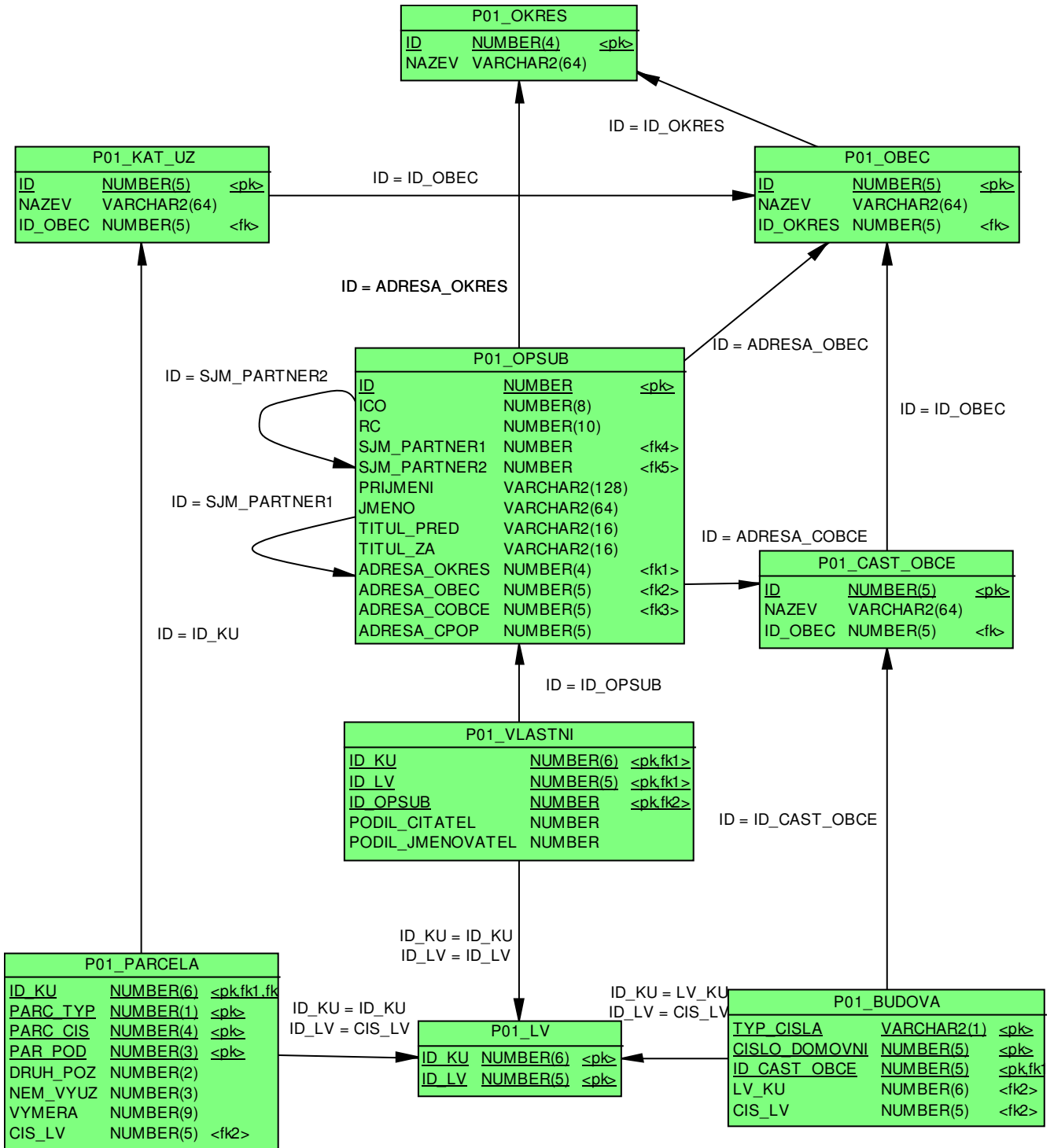
PŘÍKLAD 1.

Vytvořte konceptuální datový model reprezentovaný E-R diagramem. □



PŘÍKLAD 2.

Navrhnete logický datový model reprezentovaný PDM diagramem (tj. tabulky a referenční integrita). □



Návrh fyzického datového modelu bude kopíruje entity z Příklad 1, s jedinou výjimkou. Vazba „Vlastni“ (kardinality M:N) je realizována tabulkou.

PŘÍKLAD 3

Sestavte základací skript pro tabulky P01_OKRES, P01_OBEC, P01_CAST_OBCE, P01_KAT_UZ, P01_BUDOVA, P01_PARCELA. Dbejte na to, aby každá tabulka měla primární klíč a byly realizovány všechny vazby z diagramu v Příklad 2. □

```
DROP TABLE P01_OKRES cascade constraints;
```

```
create table P01_OKRES
(
  ID          number(4),
  NAZEV       varchar2(64),
  constraint P01_OKRES_PK primary key (id)
);
```

```
DROP TABLE P01_OBEC cascade constraints;
```

```
create table P01_OBEC
(
  ID          number(5),
  NAZEV       varchar2(64),
  ID_OKRES    number(5) NOT NULL,
  constraint P01_OBEC_PK
    primary key (ID),
  constraint P01_OBEC_FK01
    foreign key (ID_OKRES) references P01_OKRES(ID)
);
```

```
DROP TABLE P01_CAST_OBCE cascade constraints;
```

```
create table P01_CAST_OBCE
(
  ID          number(5),
  NAZEV       varchar2(64),
  ID_OBEC     number(5) NOT NULL,
  constraint P01_CAST_OBCE_PK
    primary key (ID),
  constraint P01_CAST_OBCE_FK01
    foreign key (ID_OBEC) references P01_OBEC(ID)
);
```

```
DROP TABLE P01_KAT_UZ cascade constraints;
```

```
create table P01_KAT_UZ
(
  ID          number(6),
  NAZEV       varchar2(64),
  ID_OBEC     number(5) NOT NULL,
  constraint P01_KAT_UZ_PK
    primary key (ID),
  constraint P01_KAT_UZ_FK01
    foreign key (ID_OBEC) references P01_OBEC(ID)
);
```

```
DROP TABLE P01_BUDOVA cascade constraints;
```

```
create table P01_BUDOVA
(
  TYP_CISLA          VARCHAR2(1), /* P - popisne, E -
evidencni */
  CISLO_DOMOVNI      number(5), /* cislo popisne nebo
evidencni */
  ID_CAST_OBCE       number(5) NOT NULL,
  CIS_LV             number(5),
  constraint P01_BUDOVA_PK
    primary key (ID_CAST_OBCE,TYP_CISLA,CISLO_DOMOVNI),
  constraint P01_BUDOVA_FK01
    foreign key (ID_CAST_OBCE) references P01_CAST_OBCE(ID),
  constraint P01_BUDOVA_CH01
    check (TYP_CISLA IN ('E','P')),
  constraint P01_BUDOVA_CH02
    check (CISLO_DOMOVNI>0)
);
```

```
drop table P01_PARCELA cascade constraints;
```

```
create table P01_PARCELA
(
  ID_KU           number(6),
  PARC_TYP        number(1),
  PARC_CIS        number(4),
  PAR_POD         number(3),
  DRUH_POZ        number(2),
  NEM_VYUZ        number(3),
  VYMERA          number(9),
  CIS_LV          number(5),
  constraint P01_PARCELA_PK
    primary key (ID_KU,PARC_TYP,PARC_CIS,PAR_POD),
  constraint P01_PARCELA_FK01
    foreign key (ID_KU) references P01_KAT_UZ(ID),
  constraint P01_PARCELA_CH01
    check (PARC_TYP IN ('1','2')));
```

PŘÍKLAD 4

Do základacího SQL skriptu doplň tabulky:

Oprávněných subjektů P01_OPSUB se sloupci:

- IČO
- RČ
- RČ – RČ u SJM
- Jméno
- Příjmení
- Tituly před jménem
- Tituly za jménem
- Adresa_okres
- Adresa obec
- Adresa část obce
- Adresa číslo popisné / orientační
- + nutné vazební sloupce

Listů vlastnictví P01_LV se sloupci:

- ID (identifikace)
- + nutné vazební sloupce

Vazba mezi Oprávněnými subjekty a LV P01_VLASTNÍ se sloupci

- Podíl – čítatel
- Podíl – jmenovatel
- + nutné vazební sloupce

Uprav v SQL skriptu chybějící vazby.□


```
create table P01_OPSSUB
(
  ID          int,
  ICO         number(8),
  RC          number(10),
  SJM_PARTNER1 int,
  SJM_PARTNER2 int,
  PRIJMENI   varchar2(128),
  JMENO      varchar2(64),
  TITUL_PRED varchar2(16),
  TITUL_ZA   varchar2(16),
  ADRESA_OKRES number(4) not null,
  ADRESA_OBEC number(5) not null,
  ADRESA_COBCE number(5) not null,
  ADRESA_CPOP number(5) not null,
  constraint P01_OPSSUB_PK primary key (ID),
  constraint P01_OPSSUB_FK1 foreign key
    (ADRESA_OKRES) references P01_OKRES(ID),
  constraint P01_OPSSUB_FK2 foreign key
    (ADRESA_OBEC) references P01_OBEC(ID),
  constraint P01_OPSSUB_FK3 foreign key (ADRESA_OBEC)
    references P01_OBEC(ID),
  constraint P01_OPSSUB_FK4 foreign key (ADRESA_COBCE)
    references P01_CAST_OBCE(ID),
  constraint P01_OPSSUB_FK5 foreign key (SJM_PARTNER1)
    references P01_OPSSUB(ID),
  constraint P01_OPSSUB_FK6 foreign key (SJM_PARTNER2)
    references P01_OPSSUB(ID)
);
```

```
create table P01_LV
(
  ID_KU      number(6),
  ID_LV      number(5),
  constraint P01_LV_PK primary key (ID_KU, ID_LV),
  constraint P01_LV_FK1 foreign key (ID_KU)
    references P01_KAT_UZ(ID)
);
```

```
alter table P01_PARCELA add
constraint P01_PARCELA_FK3
foreign key      (ID_KU, CIS_LV)
references P01_LV(ID_KU, ID_LV);
```

```
alter table P01_PARCELA modify CIS_LV not null;
```

```
alter table P01_BUDOVA add LV_KU number(6);
```

```
alter table P01_BUDOVA add
  constraint P01_BUDOVA_FK3
    foreign key      (LV_KU,CIS_LV)
    references P01_LV(ID_KU, ID_LV);

alter table P01_BUDOVA modify CIS_LV not null;

alter table P01_BUDOVA modify LV_KU not null;

create table P01_VLASTNI
(
  ID_KU          number(6),
  ID_LV          number(5),
  ID_OPSUB       int,
  PODIL_CITATEL int,
  PODIL_JMENOVA TEL int,
  constraint P01_VLASTNI_PK primary key
    (ID_KU, ID_LV, ID_OPSUB),
  constraint P01_VLASTNI_FK1 foreign key (ID_KU, ID_LV)
    references P01_LV(ID_KU, ID_LV),
  constraint P01_VLASTNI_FK2 foreign key (ID_OPSUB)
    references P01_OPSUB(ID)
);
```

PŘÍKLAD 5

Sestrojte dotazy, které vrátí:

- Jméno, příjmení a adresu oprávněného subjektu (ID=%d)
- Parcely a podíly ve vlastnictví oprávněného subjektu (ID=%d)
- Sestrojte dotaz, který vrátí všechny vlastníky parcely, podíly a jejich adresy, je-li vstupem její úplná identifikace (tj. její primární klíč).□

```
select
  OS.JMENO,
  OS.PRIJMENI,
  OK.NAZEV,
  OB.NAZEV,
  CO.NAZEV,
  OS.ADRESA_CPOP
from
  P01_OKRES      OK,
  P01_OBEC       OB,
  P01_CAST_OBCE CO,
  P01_OPSUB      OS
where
  OS.ID           =%d      AND
  OS.ADRESA_OKRES=OK.ID AND
  OS.ADRESA_OBEC =OB.ID AND
  OS.ADRESA_COBCE=CO.ID;
```

```
select
  KU.NAZEV,
  PA.PARC_TYP,
  PA.PARC_CIS,
  PA.PAR_POD,
  VL.PODIL_CITATEL || '/' || VL.PODIL_JMENOVA TEL
from
  P01_VLASTNI VL,
  P01_PARCELA PA,
  P01_KAT_UZ  KU
where
  VL.ID_OPSUB=%d      AND
  VL.ID_KU    =PA.ID_KU AND
  VL.ID_LV    =PA.CIS_LV AND
  PA.ID_KU    =KU.ID;
```

PŘÍKLAD 6

Zrušte v tabulce (resp. zakažte používání viz. alter table .. modify .. disable) integritní omezení primárního klíče P01_OPSUB_PK. Pokusně vložte do P01_OPSUB_PK několik řádků, z nichž některé mají stejná ID.

- a) Odstraňte duplicitní řádky P01_OPSUB_PK
- b) Znovu povolte integritní omezení P01_OPSUB_PK.□

Metoda #1:

- a) create sequence p;
- b) alter table P01_OPSUB add POM int;
- c) update P01_OPSUB set pom=p.nextval;
- d) delete from P01_OPSUB A where pom<>
(select max(pom) from P01_OPSUB B where a.id=b.id);
- e) alter table P01_OPSUB drop column POM;
- f) drop sequence p;

Metoda #2:

Každý SQL SELECT, UPDATE obsahuje fiktivní sloupec ROWNUM, který obsahuje zpracováváný řádek

- a) alter table P01_OPSUB add POM int;
- b) update P01_OPSUB set pom=rownum;
- d) delete from P01_OPSUB A where
pom<>(select max(pom) from P01_OPSUB B where a.id=b.id);

Metoda #3:

Fiktivní sloupec ROWID v každé tabulce obsahuje „adresu řádku tabulky“, je unikátní pro celou instanci databáze ORACLE.

```
delete from P01_OPSUB A where ROWID<>
(
  select max(ROWID) from P01_OPSUB b
  where a.id=b.id
);
```

PŘÍKLAD 7

Realizujte trigger, který na tabulce P01_OPSUB automaticky naplní primární klíč (ID) hodnotou se sekvence P01_OPSUB_SEQ a zkontroluje korektnost sloupce RC (rodné číslo). □

PŘÍKLAD 8

V tabulce P01_OPSUB jsou redundantní cizí klíče,

ADRESA_OKRES (→P01_OKRES)

ADRESA_OBEC (→P01_OBEC)

jejich hodnota lze odvodit z tabulek P01_CAST_OBCE a P01_OBEC.²

Mohou však být nekonzistentní, tj. v jednom řádku tabulky P01_OPSUB je uvedena obec, která není nadřizena části obce atd.

- Zformulujte SQL skript, který tyto klíče uvede do souladu.
- Zajistěte (triggerem), aby k takové situaci nemohlo dojít. □

PŘÍKLAD 10

Datový model doplňte tabulkou:

```
CREATE TABLE P01_REPORT
(
  ID          INT,
  LINE_NO    INT,
  LINE       VARCHAR2(1024),
  CONSTRAINT P01_REPORT_PK PRIMARY KEY (ID, LINE_NO)
)
```

Sestavte PL/SQL proceduru GENERATE_LV (v package P01)

```
package P01 is
  procedure GENERATE_LV (
    KU in INT, /* katastrální území */
    LV in INT, /* číslo listu vlastnictví */
    ID in int /* report id */
  );
end P01;
```

po jejímž provedení:

```
begin
  P01.GENERATE_LV(ku, lv, xx);
end;
/
```

² Nemusí se nutně jednat o analytickou chybu, důvodem může být zrychlování dotazů.

příkaz:

```
SELECT LINE FROM P01_REPORT WHERE ID=xx ORDER BY LINE_NO
```

vrátí výpis - list vlastnictví v této formě:

Okres: 3209 Praha - východ
 Obec: 078 Říčany
 Kat. území 01 Říčany u Prahy
 Číslo LV 365

A Vlastník (pořadí jméno/název a adresa)	Identifikátor	Podíl
--	---------------	-------

1 Josef Novák Podlesná 11, Brno	581205/3256	1/2
2 Pavel Novák Sady 15, Olomouc	351231/454	1/2

B Parcelní číslo	Výměra m2	Část číslo obce budovy	Druh
1452/2	4584		louka
1452/3	592		orná půda