

# Portál stolní RPG hry

Dokumentace zkouškového příkladu pro předmět 4IT218 Databáze

Denisova Ekaterina, Bc.

*LS 2017/2018*

Uživatelské jméno: dene00

## Obsah

1	Popis zvolené výseče světa – zadání .....	3
2	Konceptuální schéma reality .....	4
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 .....	12
4.4	Definice přístupových práv.....	16
4.5	Definice dalších databázových objektů .....	16
5	Obsah databáze .....	18
5.1	SQL příkazy pro naplnění databáze daty .....	18
5.2	Opis vložených dat.....	20

## 1 Popis zvolené výšeče světa – zadání

1. září 2010 byl založen portál stolní RPG hry Dungeons and Dragons. Každý zájemce se může registrovat na portálu. Pro registraci hráč musí vyplnit dotazník, kde je třeba vymyslet svůj unikátní login, napsat svoje jméno, email a také pro každého hráče bude zapsáno datum registrace. Každý hráč může vytvořit postavu nebo více postav, pokud bude chtít zahrát jiným stylem. Pro postavu musí být vymyšleno svoje unikátní jméno. Jelikož majitel portálu je fanoušek klasických pravidel, bylo rozhodnuto povolit hráčům vytvářet postavy, které mají určité rasy a povolání. Například, hráč může vybrat rasu Člověk, Elf, Půlelf, Trpaslík, Půlčík nebo Gnóm a jedno z přípustných povolání: Bojovník, Paladin, Hraničář, Mág, Specialista, Klerik, Druid, Zloděj nebo Bard. Jistě každá postava má nějaký úroveň (0-99).

Každá postava se může zúčastnit dobrodružství. A v každém dobrodružství hráč může hrát několika svými postavami zároveň. U dobrodružství musí být vymyšlen unikátní název, ale pro naše pohodlí budeme používat unikátní číslo dobrodružství. Každé dobrodružství se koná na nějaké lokalitě, má obtížnost (snadné, průměrné, obtížné, epické) a cíl. Hráč se může zúčastnit dobrodružství ne jenom jako obyčejný hráč, ale ještě může být game masterem. Ale když nikdo nezatouží být game masterem, dobrodružství se nebude konat.

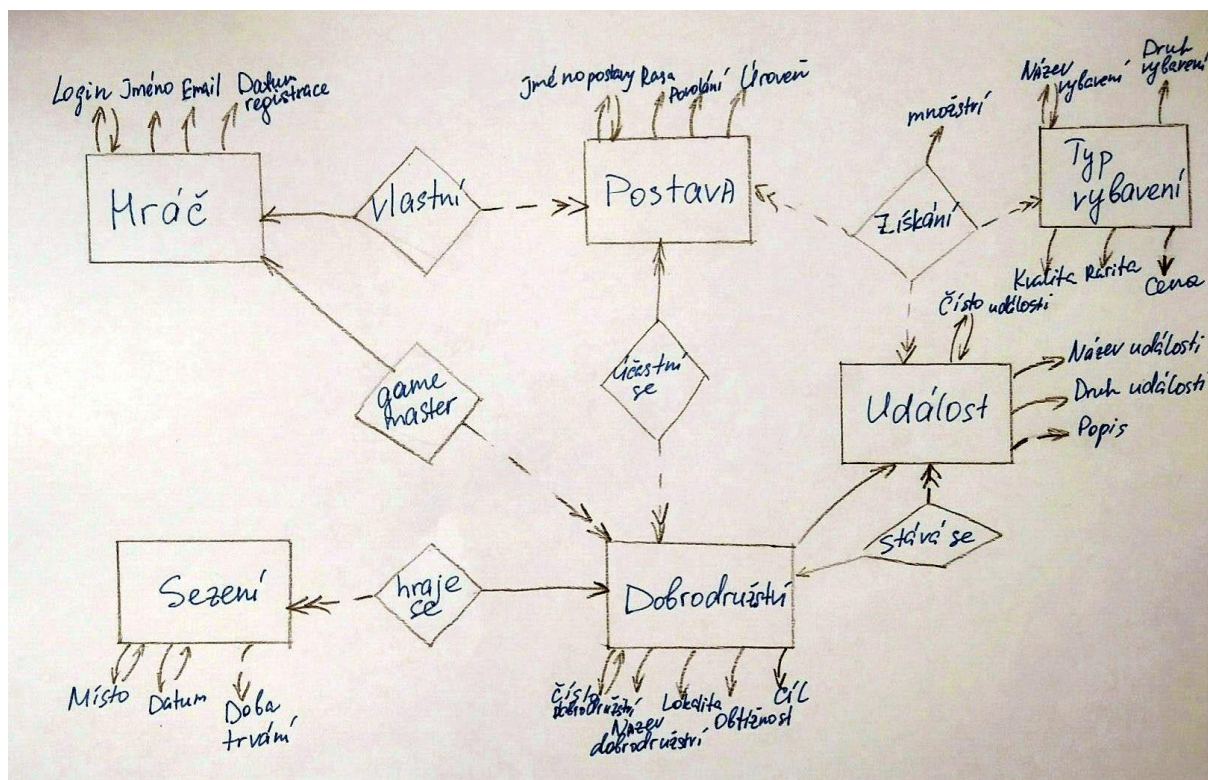
Jestli hráči se rozhodli setkat se pro hru, musí vybrat místo a datum pro sezení a určit dobu trvání v minutách a musí být kratší, než 300 minut. Protože už několikrát se stávalo, že hráči hráli celý den bez přestávky a potom měli problémy se zdravím.

Během dobrodružství vždycky se něco stává. A pro nás je důležité sledovat ty události, protože během události postava může získat nějaké vybavení.

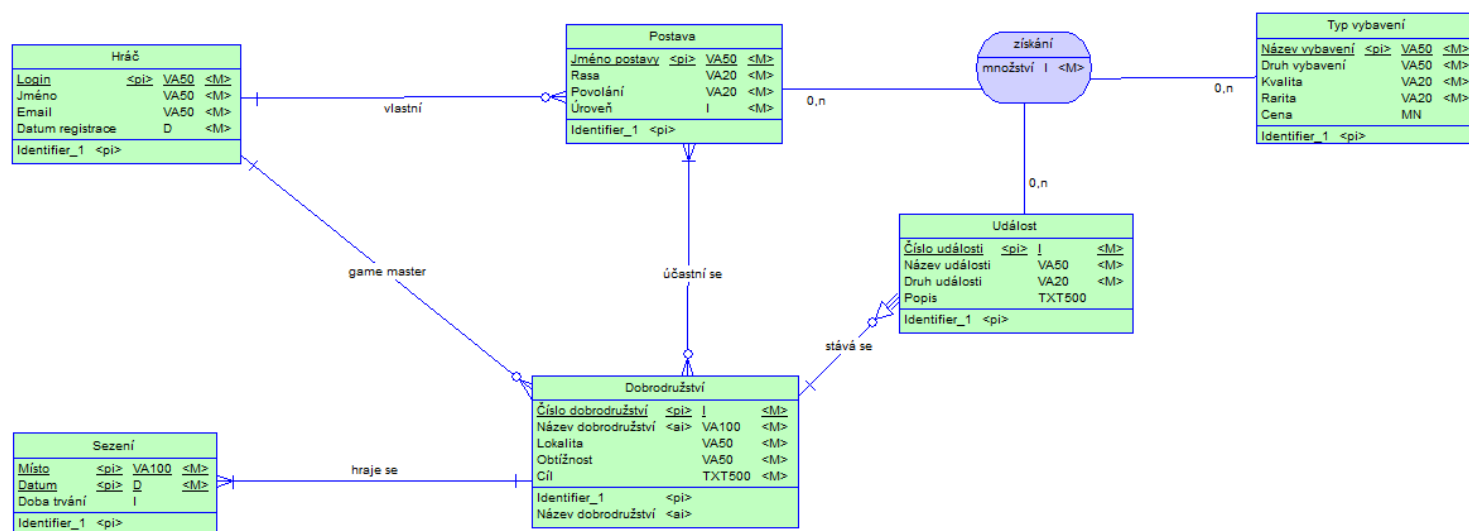
Událost může se konat v různých dobrodružství, například, postavy můžou zabít vlka v každém dobrodružství, kde se s ním potkají. Ale událost má svoje pořadové číslo, název události, druh události a nepovinný popis. Většinou události můžeme rozdělit do skupin Nákup, Přijetí úlohy, Odevzdání úlohy a Vražda. Jestli se stane něco mimořádného, jako druh události stačí napsat „Jiné“.

Postava může získat vybavení během události. U každého typu vybavení jsou určité vlastnosti: název vybavení, druh, kvalita, rarita a cena, kterou je také možno zapsat. Postava může dostat nějakou zbraň, brnění, jídlo, svitek, věci pro úkoly nebo něco jiného. Věc může být rozbitou, starou, obyčejnou nebo zlepšenou. A postava může získat vybavení s různou raritou: obyčejnou, vzácnou, legendární nebo epickou.

## 2 Konceptuální schéma reality

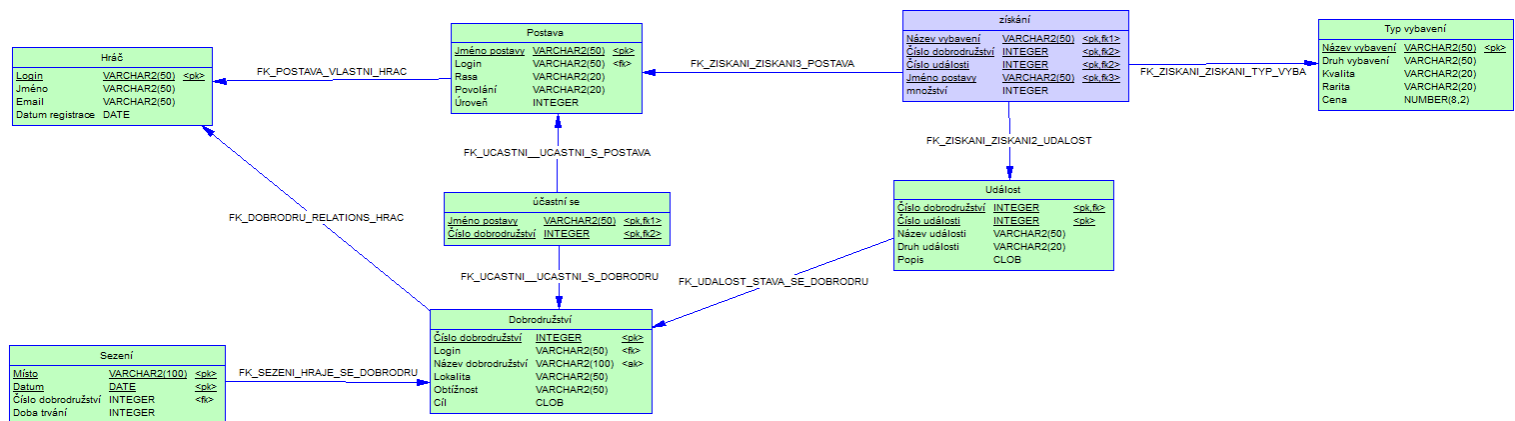


### 3 Konceptuální datový model



## 4 Dokumentace databáze

### 4.1 Fyzický datový model



## 4.2 Definice relačních tabulek a souvisejících objektů

```
alter table DOBRODRUZSTVI
  drop constraint FK_DOBRODRU_RELATIONS_HRAC;

alter table POSTAVA
  drop constraint FK_POSTAVA_VLASTNI_HRAC;

alter table SEZENI
  drop constraint FK_SEZENI_HRAJE_SE_DOBRODRU;

alter table UCASTNI_SE
  drop constraint FK_UCASTNI__UCASTNI_S_POSTAVA;

alter table UCASTNI_SE
  drop constraint FK_UCASTNI__UCASTNI_S_DOBRODRU;

alter table UDALOST
  drop constraint FK_UDALOST_STAVA_SE_DOBRODRU;

alter table ZISKANI
  drop constraint FK_ZISKANI_ZISKANI_TYP_VYBA;

alter table ZISKANI
  drop constraint FK_ZISKANI_ZISKANI2_UDALOST;

alter table ZISKANI
  drop constraint FK_ZISKANI_ZISKANI3_POSTAVA;

drop index RELATIONSHIP_5_FK;

drop index VLASTNI_FK;

drop index HRAJE_SE_FK;

drop index UCASTNI_SE2_FK;

drop index UCASTNI_SE_FK;

drop index STAVA_SE_FK;

drop index ZISKANI2_FK;

drop index ZISKANI3_FK;

drop index ZISKANI_FK;

drop table DOBRODRUZSTVI cascade constraints;

drop table HRAC cascade constraints;

drop table POSTAVA cascade constraints;

drop table SEZENI cascade constraints;

drop table TYP_VYBAVENI cascade constraints;
```

drop table UCASTNI\_SE cascade constraints;

drop table UDALOST cascade constraints;

drop table ZISKANI cascade constraints;

```

/*=====*/
/* Table: DOBRODRUZSTVI */
/*=====*/
create table DOBRODRUZSTVI (
  CISLO_DOBRODRUZSTVI INTEGER          not null,
  LOGIN                VARCHAR2(50)    not null,
  NAZEV_DOBRODRUZSTVI VARCHAR2(100)    not null,
  LOKALITA             VARCHAR2(50)     not null,
  OBTIZNOST            VARCHAR2(50)     not null
  constraint CKC_OBTIZNOST_DOBRODRU check (OBTIZNOST in ('Snadny','Prumerny','Obtizny','Epicky')),
  CIL                  CLOB             not null,
  constraint PK_DOBRODRUZSTVI primary key (CISLO_DOBRODRUZSTVI),
  constraint CKC_CIS_DOB check (CISLO_DOBRODRUZSTVI>0),
  constraint AK_NAZEV_DOBRODRUZSTV_DOBRODRU unique (NAZEV_DOBRODRUZSTVI)
);

/*=====*/
/* Index: RELATIONSHIP_5_FK */
/*=====*/
create index RELATIONSHIP_5_FK on DOBRODRUZSTVI (
  LOGIN ASC
);

/*=====*/
/* Table: HRAC */
/*=====*/
create table HRAC (
  LOGIN                VARCHAR2(50)    not null,
  JMENO                VARCHAR2(50)    not null,
  EMAIL                VARCHAR2(50)    not null
  constraint CKC_EMAIL check (REGEXP_LIKE(EMAIL,'^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$')),
  DATUM_REGISTRACE    DATE             not null
  constraint CKC_DATUM_REGISTR check (DATUM_REGISTRACE>=to_date('2010-01-09', 'YYYY-DD-MM')),
  constraint PK_HRAC primary key (LOGIN)
);

/*=====*/
/* Table: POSTAVA */
/*=====*/
create table POSTAVA (
  JMENO_POSTAVY        VARCHAR2(50)    not null,
  LOGIN                VARCHAR2(50)    not null,
  RASA                 VARCHAR2(20)     not null
  constraint CKC_RASA_POSTAVA check (RASA in ('Clovek','Elf','Pulelf','Trpaslik','Pulcik','Gnom')),
  POVOLANI             VARCHAR2(20)     not null
  constraint CKC_POVOLANI_POSTAVA check (POVOLANI in ('Bojovnik','Paladin','Hranicar','Mag','Specialista','Klerik','Druid','Zlodej','Bard')),
  UROVEN               INTEGER          not null
  constraint CKC_UROVEN_POSTAVA check (UROVEN between 0 and 100),
  constraint PK_POSTAVA primary key (JMENO_POSTAVY)
);

```



```
/*=====*/
/* Index: VLASTNI_FK */
/*=====*/
create index VLASTNI_FK on POSTAVA (
  LOGIN ASC
);

/*=====*/
/* Table: SEZENI */
/*=====*/
create table SEZENI (
  MISTO      VARCHAR2(100)      not null,
  DATUM      DATE               not null,
  CISLO_DOBRODRUZSTVI INTEGER    not null,
  DOBA_TRVANI INTEGER,
  constraint PK_SEZENI primary key (MISTO, DATUM),
  constraint CKC_DOB_TR check (DOBA_TRVANI between 1 and 300),
  constraint CKC_DATUM_SEZ check (DATUM >=to_date('2010-01-09', 'YYYY-DD-MM'))
);

/*=====*/
/* Index: HRAJE_SE_FK */
/*=====*/
create index HRAJE_SE_FK on SEZENI (
  CISLO_DOBRODRUZSTVI ASC
);

/*=====*/
/* Table: TYP_VYBAVENI */
/*=====*/
create table TYP_VYBAVENI (
  NAZEV_VYBAVENI VARCHAR2(50)      not null,
  DRUH_VYBAVENI  VARCHAR2(50)      not null
  constraint CKC_DRUH_VYBAVENI_TYP_VYBA check (DRUH_VYBAVENI in ('Zbran','Brneni','Jidlo','Svitek','Veci
pro ukoly','Jine')),
  KVALITA        VARCHAR2(20)      not null
  constraint CKC_KVALITA_TYP_VYBA check (KVALITA in ('Rozbity','Stary','Obycejny','Zlepseny')),
  RARITA         VARCHAR2(20)      not null
  constraint CKC_RARITA_TYP_VYBA check (RARITA in ('Obycejny','Vzacny','Legendarni','Epicky')),
  CENA           NUMBER(8,2)
  constraint CKC_CENA_TYP_VYBA check (CENA is null or (CENA >= 0)),
  constraint PK_TYP_VYBAVENI primary key (NAZEV_VYBAVENI)
);

/*=====*/
/* Table: UCASTNI_SE */
/*=====*/
create table UCASTNI_SE (
  JMENO_POSTAVY VARCHAR2(50)      not null,
  CISLO_DOBRODRUZSTVI INTEGER      not null,
  constraint PK_UCASTNI_SE primary key (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI)
);

/*=====*/
/* Index: UCASTNI_SE_FK */
/*=====*/
```

```
create index UCASTNI_SE_FK on UCASTNI_SE (
  JMENO_POSTAVY ASC
);

/*=====*/
/* Index: UCASTNI_SE2_FK                      */
/*=====*/
create index UCASTNI_SE2_FK on UCASTNI_SE (
  CISLO_DOBRODRUZSTVI ASC
);

/*=====*/
/* Table: UDALOST                             */
/*=====*/
create table UDALOST (
  CISLO_DOBRODRUZSTVI INTEGER          not null,
  CISLO_UDALOSTI      INTEGER          not null,
  NAZEV_UDALOSTI      VARCHAR2(50)     not null,
  DRUH_UDALOSTI       VARCHAR2(20)     not null
  constraint CKC_DRUH_UDALOSTI_UDALOST check (DRUH_UDALOSTI in ('Nakup','Prodej','Prijeti
ulohy','Odevzdani ulohy','Vrazda','Jine')),
  POPIS              CLOB,
  constraint PK_UDALOST primary key (CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI),
  constraint CKC_CIS_UDL check (CISLO_UDALOSTI>0)
);

/*=====*/
/* Index: STAVA_SE_FK                        */
/*=====*/
create index STAVA_SE_FK on UDALOST (
  CISLO_DOBRODRUZSTVI ASC
);

/*=====*/
/* Table: ZISKANI                             */
/*=====*/
create table ZISKANI (
  NAZEV_VYBAVENI      VARCHAR2(50)     not null,
  CISLO_DOBRODRUZSTVI INTEGER          not null,
  CISLO_UDALOSTI      INTEGER          not null,
  JMENO_POSTAVY       VARCHAR2(50)     not null,
  MNOZSTVI            INTEGER          not null
  constraint CKC_MNOZSTVI_ZISKANI check (MNOZSTVI >= 1),
  constraint PK_ZISKANI primary key (CISLO_DOBRODRUZSTVI, NAZEV_VYBAVENI, CISLO_UDALOSTI,
JMENO_POSTAVY)
);

/*=====*/
/* Index: ZISKANI_FK                        */
/*=====*/
create index ZISKANI_FK on ZISKANI (
  NAZEV_VYBAVENI ASC
);

/*=====*/
/* Index: ZISKANI2_FK                      */
/*=====*/
```

```
create index ZISKANI2_FK on ZISKANI (
  CISLO_DOBRODRUZSTVI ASC,
  CISLO_UDALOSTI ASC
);

/*=====*/
/* Index: ZISKANI3_FK                      */
/*=====*/
create index ZISKANI3_FK on ZISKANI (
  JMENO_POSTAVY ASC
);
alter table DOBRODRUZSTVI
  add constraint FK_DOBRODRU_RELATIONS_HRAC foreign key (LOGIN)
    references HRAC (LOGIN);

alter table POSTAVA
  add constraint FK_POSTAVA_VLASTNI_HRAC foreign key (LOGIN)
    references HRAC (LOGIN)
    on delete cascade;

alter table SEZENI
  add constraint FK_SEZENI_HRAJE_SE_DOBRODRU foreign key (CISLO_DOBRODRUZSTVI)
    references DOBRODRUZSTVI (CISLO_DOBRODRUZSTVI)
    on delete cascade;

alter table UCASTNI_SE
  add constraint FK_UCASTNI__UCASTNI_S_POSTAVA foreign key (JMENO_POSTAVY)
    references POSTAVA (JMENO_POSTAVY)
    on delete cascade;

alter table UCASTNI_SE
  add constraint FK_UCASTNI__UCASTNI_S_DOBRODRU foreign key (CISLO_DOBRODRUZSTVI)
    references DOBRODRUZSTVI (CISLO_DOBRODRUZSTVI)
    on delete cascade;

alter table UDALOST
  add constraint FK_UDALOST_STAVA_SE_DOBRODRU foreign key (CISLO_DOBRODRUZSTVI)
    references DOBRODRUZSTVI (CISLO_DOBRODRUZSTVI)
    on delete cascade;

alter table ZISKANI
  add constraint FK_ZISKANI_ZISKANI_TYP_VYBA foreign key (NAZEV_VYBAVENI)
    references TYP_VYBAVENI (NAZEV_VYBAVENI)
    on delete cascade;

alter table ZISKANI
  add constraint FK_ZISKANI_ZISKANI2_UDALOST foreign key (CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI)
    references UDALOST (CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI)
    on delete cascade;

alter table ZISKANI
  add constraint FK_ZISKANI_ZISKANI3_POSTAVA foreign key (JMENO_POSTAVY)
    references POSTAVA (JMENO_POSTAVY)
    on delete cascade;
```

## 4.3 Integritní omezení

### Tabulka HRAC

#### Entitní integrita

- Atributy tvořící primární klíč: LOGIN.
- SQL kód pro definici primárního klíče:  
constraint PK\_HRAC primary key (LOGIN)

#### Doménová integrita

- Email hráče musí být ve formátu 'xxx@xxx.xxx', povolené znaky - písmena, číslice, tečka, pomlčka, podtržítka.
- constraint CKC\_EMAIL check (REGEXP\_LIKE(EMAIL,'^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}\$'))
- Registrace se musí konat později než vytvoření portálu
- constraint CKC\_DATUM\_REGISTR check (DATUM\_REGISTRACE>=to\_date('2010-01-09', 'YYYY-DD-MM'))

### Tabulka POSTAVA

#### Entitní integrita

- Atributy tvořící primární klíč: JMENO\_POSTAVY
- SQL kód pro definici primárního klíče:  
constraint PK\_POSTAVA primary key (JMENO\_POSTAVY)

#### Doménová integrita

- Máme přesně definované rasy: Člověk, Elf, Půlelf, Trpaslík, Půlčik, Gnóm
- constraint CKC\_RASA\_POSTAVA check (RASA in ('Clovek','Elf','Pulelf','Trpaslik','Pulcik','Gnom'))
- Postava může mít jedno z následujících povolání: Bojovník, Paladin, Hraničář, Mág, Specialista, Klerik, Druid, Zloděj, Bard
- constraint CKC\_POVOLANI\_POSTAVA check (POVOLANI in ('Bojovnik','Paladin','Hranicar','Mag','Specialista','Klerik','Druid','Zlodej','Bard'))
- Úroveň postavy je číslo mezi 0 a 100
- constraint CKC\_UROVEN\_POSTAVA check (UROVEN between 0 and 100)

#### Referenční integrita

- Pokud bude smazán některý hráč, automaticky se mají smazat jeho postavy.
- Vztah POSTAVA – HRAC - cascade
- alter table POSTAVA  
add constraint FK\_POSTAVA\_VLASTNI\_HRAC foreign key (LOGIN)  
references HRAC (LOGIN)  
on delete cascade;

### Tabulka TYP\_VYBAVENI

#### Entitní integrita

- Atributy tvořící primární klíč: NAZEV\_VYBAVENI

- SQL kód pro definici primárního klíče:  
constraint PK\_TYP\_VYBAVENI primary key (NAZEV\_VYBAVENI)

#### *Doménová integrita*

- Jako vybavení během hry je možno dostat zbraň, brnění, jídlo, svitek, věci pro úkoly nebo něco jiného
- constraint CKC\_DRUH\_VYBAVENI\_TYP\_VYBA check (DRUH\_VYBAVENI in ('Zbran','Brneni','Jidlo','Svitek','Veci pro ukoly','Jine'))
- Věc může být rozbitou, starou, obyčejnou nebo zlepšenou a to zapíšeme do atributu Kvalita
- constraint CKC\_KVALITA\_TYP\_VYBA check (KVALITA in ('Rozbity','Stary','Obycejny','Zlepseny'))
- Rarita věci: Obyčejný, Vzácný, Legendární, Epický
- constraint CKC\_RARITA\_TYP\_VYBA check (RARITA in ('Obycejny','Vzacny','Legendarni','Epicky'))
- Věc může mít nějakou cenu a ta cena musí být nezáporná
- constraint CKC\_CENA\_TYP\_VYBA check (CENA is null or (CENA >= 0))

#### Tabulka DOBRODRUZSTVI

##### *Entitní integrita*

- Atributy tvořící primární klíč: CISLO\_DOBRODRUZSTVI
- SQL kód pro definici primárního klíče:  
constraint PK\_DOBRODRUZSTVI primary key (CISLO\_DOBRODRUZSTVI).

##### *Doménová integrita*

- Číslem dobrodružství může být pouze kladné číslo
- constraint CKC\_CIS\_DOBR check (CISLO\_DOBRODRUZSTVI>0)
- Názvy dobrodružství jsou unikátní
- constraint AK\_NAZEV\_DOBRODRUZSTV\_DOBRODRU unique (NAZEV\_DOBRODRUZSTVI)
- Obtížnost dobrodružství má následující stupnici: snadný, průměrný, obtížný, epický
- constraint CKC\_OBTIZNOST\_DOBRODRU check (OBTIZNOST in ('Snadny','Prumerny','Obtizny','Epicky'))

##### *Referenční integrita*

- Vždy musíme vědět game mastera, proto nesmíme ho smazat.
- Vztah DOBRODRUZSTVI – HRAC – restrict
- alter table DOBRODRUZSTVI  
add constraint FK\_DOBRODRU\_RELATIONS\_HRAC foreign key (LOGIN)  
references HRAC (LOGIN);

#### Tabulka SEZENI

##### *Entitní integrita*

- Atributy tvořící primární klíč: MISTO, DATUM

- SQL kód pro definici primárního klíče.  
constraint PK\_SEZENI primary key (MISTO, DATUM)

#### *Doménová integrita*

- Doba trvání sezení v minutách musí být kladným číslem a nemůže překročit 960 min
- constraint CKC\_DOB\_TR check (DOBA\_TRVANI between 1 and 300)
- Sezení nemůže být dříve, než byl založen portál
- constraint CKC\_DATUM\_SEZ check (DATUM >=to\_date('2010-01-09', 'YYYY-DD-MM'))

#### *Referenční integrita*

- Pokud dobrodružství bude smazáno, záznam v tabulce SEZENI taky musí zmizet
- Vztah SEZENI-DOBRODRUZSTVI - cascade
- alter table SEZENI  
add constraint FK\_SEZENI\_HRAJE\_SE\_DOBRODRU foreign key (CISLO\_DOBRODRUZSTVI)  
references DOBRODRUZSTVI (CISLO\_DOBRODRUZSTVI)  
on delete cascade;

### Tabulka UCASTNI\_SE

#### *Entitní integrita*

- Atributy tvořící primární klíč: JMENO\_POSTAVY, CISLO\_DOBRODRUZSTVI
- SQL kód pro definici primárního klíče:  
constraint PK\_UCASTNI\_SE primary key (JMENO\_POSTAVY, CISLO\_DOBRODRUZSTVI)

#### *Referenční integrita*

- Pokud bude smazána postava, záznam o její účasti v dobrodružství se taky má smazat
- Vztah UCASTNI\_SE-POSTAVA – cascade
- alter table UCASTNI\_SE  
add constraint FK\_UCASTNI\_\_UCASTNI\_S\_POSTAVA foreign key (JMENO\_POSTAVY)  
references POSTAVA (JMENO\_POSTAVY)  
on delete cascade;
- Pokud bude smazáno dobrodružství, záznamy v tabulce UCASTNI\_SE musí také zmizet
- Vztah UCASTNI\_SE-DOBRODRUZSTVI - cascade
- alter table UCASTNI\_SE  
add constraint FK\_UCASTNI\_\_UCASTNI\_S\_DOBRODRU foreign key (CISLO\_DOBRODRUZSTVI)  
references DOBRODRUZSTVI (CISLO\_DOBRODRUZSTVI)  
on delete cascade;

### Tabulka UDALOST

#### *Entitní integrita*

- Atributy tvořící primární klíč: CISLO\_DOBRODRUZSTVI, CISLO\_UDALOSTI
- SQL kód pro definici primárního klíče:  
constraint PK\_UDALOST primary key (CISLO\_DOBRODRUZSTVI, CISLO\_UDALOSTI)

#### *Doménová integrita*

- Číslo události musí být kladným
- constraint CKC\_CIS\_UDL check (CISLO\_UDALOSTI>0)

#### Referenční integrita

- Po smazání dobrodružství, událost se musí automatické smazat
- Vztah UDALOST-DOBRODRUZSTVI - cascade
- alter table UDALOST  
add constraint FK\_UDALOST\_STAVA\_SE\_DOBRODRU foreign key (CISLO\_DOBRODRUZSTVI)  
references DOBRODRUZSTVI (CISLO\_DOBRODRUZSTVI)  
on delete cascade;

#### Tabulka ZISKANI

##### Entitní integrita

- Atributy tvořící primární klíč: CISLO\_DOBRODRUZSTVI, NAZEV\_VYBAVENI, CISLO\_UDALOSTI, JMENO\_POSTAVY
- SQL kód pro definici primárního klíče:  
constraint PK\_ZISKANI primary key (CISLO\_DOBRODRUZSTVI, NAZEV\_VYBAVENI, CISLO\_UDALOSTI, JMENO\_POSTAVY)

##### Doménová integrita

- Jestli postava dostane nějaké vybavení během událostí, musíme napsat ho množství, které se rovná alespoň 1
- constraint CKC\_MNOZSTVI\_ZISKANI check (MNOZSTVI >= 1)

#### Referenční integrita

- Pokud smažeme nějaké vybavení, musíme smazat záznam o jeho získání
- Vztah ZISKANI-TYP\_VYBAVENI - cascade
- alter table ZISKANI  
add constraint FK\_ZISKANI\_ZISKANI\_TYP\_VYBA foreign key (NAZEV\_VYBAVENI)  
references TYP\_VYBAVENI (NAZEV\_VYBAVENI)  
on delete cascade;
- Pokud smažeme nějakou událost, musíme smazat záznamy, že postava během ní získala nějaké vybavení
- Vztah ZISKANI-UDALOST - cascade
- alter table ZISKANI  
add constraint FK\_ZISKANI\_ZISKANI2\_UDALOST foreign key (CISLO\_DOBRODRUZSTVI, CISLO\_UDALOSTI)  
references UDALOST (CISLO\_DOBRODRUZSTVI, CISLO\_UDALOSTI)  
on delete cascade;
- Pokud smažeme nějakou postavu, musí se automaticky smazat všechny záznamy v tabulce ZISKANI
- Vztah ZISKANI-ZISKANI – cascade
- alter table ZISKANI  
add constraint FK\_ZISKANI\_ZISKANI3\_POSTAVA foreign key (JMENO\_POSTAVY)  
references POSTAVA (JMENO\_POSTAVY)  
on delete cascade;

#### 4.4 Definice přístupových práv

a) STUDENT: práva pro operaci SELECT

```
grant select on DOBRODRUZSTVI to STUDENT;  
grant select on HRAC to STUDENT;  
grant select on POSTAVA to STUDENT;  
grant select on SEZENI to STUDENT;  
grant select on TYP_VYBAVENI to STUDENT;  
grant select on UCASTNI_SE to STUDENT;  
grant select on UDALOST to STUDENT;  
grant select on ZISKANI to STUDENT;
```

b) IT218: práva na operace SELECT, INSERT, UPDATE, DELETE

```
grant select, insert, update, delete on DOBRODRUZSTVI to IT218;  
grant select, insert, update, delete on HRAC to IT218;  
grant select, insert, update, delete on POSTAVA to IT218;  
grant select, insert, update, delete on SEZENI to IT218;  
grant select, insert, update, delete on TYP_VYBAVENI to IT218;  
grant select, insert, update, delete on UCASTNI_SE to IT218;  
grant select, insert, update, delete on UDALOST to IT218;  
grant select, insert, update, delete on ZISKANI to IT218;
```

#### 4.5 Definice dalších databázových objektů

```
/*POCET_POSTAV_V_DOBRODRUZSTVI*/
```

```
/*V dobrodružství se musí zúčastnit alespoň dvě postavy*/
```

```
create or replace view POCET_POSTAV_V_DOBRODRUZSTVI as  
select CISLO_DOBRODRUZSTVI, count(JMENO_POSTAVY) as POCET_POSTAV  
from DOBRODRUZSTVI left join UCASTNI_SE using (CISLO_DOBRODRUZSTVI)  
group by CISLO_DOBRODRUZSTVI  
having count(JMENO_POSTAVY)<2;
```

```
/*DATUM_REGIS_SEZEN*/
```

```
/*Registrace hráče nemůže být později než sezení*/
```

```
create or replace view DATUM_REGIS_SEZEN as  
select UCASTNI_SE.JMENO_POSTAVY, UCASTNI_SE.CISLO_DOBRODRUZSTVI, POSTAV_REG.LOGIN,  
POSTAV_REG.DATUM_REGISTRACE, SEZ_POSTAV.DATUM  
from UCASTNI_SE left join  
(select JMENO_POSTAVY, LOGIN, DATUM_REGISTRACE  
from POSTAVA join HRAC using (LOGIN)) POSTAV_REG  
on UCASTNI_SE.JMENO_POSTAVY=POSTAV_REG.JMENO_POSTAVY  
left join  
(select MISTO, DATUM, CISLO_DOBRODRUZSTVI, JMENO_POSTAVY  
from SEZENI join UCASTNI_SE using (CISLO_DOBRODRUZSTVI)) SEZ_POSTAV  
on POSTAV_REG.JMENO_POSTAVY=SEZ_POSTAV.JMENO_POSTAVY  
where POSTAV_REG.DATUM_REGISTRACE>SEZ_POSTAV.DATUM;
```

```
/*ZISKANI_BEZ_SEZENI*/
```

```
/*Postava nemůže získat vybavení během události, jestli událost byla v dobrodružství, pro které ne  
bylo zaznamenáno sezení*/
```



```
create or replace view ZISKANI_BEZ_SEZENI as
select  ZISKANI.NAZEV_VYBAVENI,  ZISKANI.CISLO_DOBRODRUZSTVI,  ZISKANI.CISLO_UDALOSTI,
ZISKANI.JMENO_POSTAVY, DOBR_SEZ.MISTO, DOBR_SEZ.DATUM
from ZISKANI left join
(select  DOBRODRUZSTVI.CISLO_DOBRODRUZSTVI,  DOBRODRUZSTVI.NAZEV_DOBRODRUZSTVI,
SEZENI.MISTO as MISTO, SEZENI.DATUM as DATUM
from      DOBRODRUZSTVI      left      join      SEZENI      on
DOBRODRUZSTVI.CISLO_DOBRODRUZSTVI=SEZENI.CISLO_DOBRODRUZSTVI) DOBR_SEZ
on ZISKANI.CISLO_DOBRODRUZSTVI=DOBR_SEZ.CISLO_DOBRODRUZSTVI
where DOBR_SEZ.MISTO is null;
```

## 5 Obsah databáze

### 5.1 SQL příkazy pro naplnění databáze daty

```
/*INSERT INTO HRAC*/
```

```
INSERT INTO "DENE00"."HRAC" (LOGIN, JMENO, EMAIL, DATUM_REGISTRACE) VALUES ('Ekdenka',
'Denisova Ekaterina', 'ekden@gmail.com', TO_DATE('10.9.2010', 'DD.MM.RR'));
INSERT INTO "DENE00"."HRAC" (LOGIN, JMENO, EMAIL, DATUM_REGISTRACE) VALUES ('Jenek',
'Evgenia Boyarentseva', 'boy@gmail.com', TO_DATE('10.09.10', 'DD.MM.RR'));
INSERT INTO "DENE00"."HRAC" (LOGIN, JMENO, EMAIL, DATUM_REGISTRACE) VALUES ('AntiGeroy',
'Vladislav Ermolaev', 'ermol@seznam.cz', TO_DATE('20.12.2010', 'DD.MM.RR'));
INSERT INTO "DENE00"."HRAC" (LOGIN, JMENO, EMAIL, DATUM_REGISTRACE) VALUES ('Grey', 'Václav
Kupka', 'kupo@seznam.cz', TO_DATE('20.12.10', 'DD.MM.RR'));
INSERT INTO "DENE00"."HRAC" (LOGIN, JMENO, EMAIL, DATUM_REGISTRACE) VALUES ('Gideon',
'Franz Havel', 'frank@gmail.cz', TO_DATE('18.1.2011', 'DD.MM.RR'));
INSERT INTO "DENE00"."HRAC" (LOGIN, JMENO, EMAIL, DATUM_REGISTRACE) VALUES ('Luntik', 'Luna
Lovegood', 'polumna@seznam.cz', TO_DATE('2.2.2011', 'DD.MM.RR'));
```

```
/*INSERT INTO POSTAVA*/
```

```
INSERT INTO "DENE00"."POSTAVA" (JMENO_POSTAVY, LOGIN, RASA, POVOLANI, UROVEN) VALUES
('Fer', 'Ekdenka', 'Elf', 'Klerik', '20');
INSERT INTO "DENE00"."POSTAVA" (JMENO_POSTAVY, LOGIN, RASA, POVOLANI, UROVEN) VALUES
('Kela', 'Jenek', 'Pulelf', 'Druid', '22');
INSERT INTO "DENE00"."POSTAVA" (JMENO_POSTAVY, LOGIN, RASA, POVOLANI, UROVEN) VALUES
('Qwiz', 'AntiGeroy', 'Clovek', 'Paladin', '1');
INSERT INTO "DENE00"."POSTAVA" (JMENO_POSTAVY, LOGIN, RASA, POVOLANI, UROVEN) VALUES
('Selom', 'Grey', 'Trpaslik', 'Specialista', '20');
INSERT INTO "DENE00"."POSTAVA" (JMENO_POSTAVY, LOGIN, RASA, POVOLANI, UROVEN) VALUES
('Vila', 'Gideon', 'Elf', 'Druid', '2');
INSERT INTO "DENE00"."POSTAVA" (JMENO_POSTAVY, LOGIN, RASA, POVOLANI, UROVEN) VALUES
('Nurik', 'Luntik', 'Clovek', 'Paladin', '15');
INSERT INTO "DENE00"."POSTAVA" (JMENO_POSTAVY, LOGIN, RASA, POVOLANI, UROVEN) VALUES
('Steroj', 'Luntik', 'Elf', 'Bard', '1');
INSERT INTO "DENE00"."POSTAVA" (JMENO_POSTAVY, LOGIN, RASA, POVOLANI, UROVEN) VALUES
('Terra', 'Gideon', 'Pulcik', 'Hranicar', '7');
INSERT INTO "DENE00"."POSTAVA" (JMENO_POSTAVY, LOGIN, RASA, POVOLANI, UROVEN) VALUES
('Ruin', 'AntiGeroy', 'Trpaslik', 'Mag', '25');
```

```
/*INSERT INTO TYP_VYBAVENI*/
```

```
INSERT INTO "DENE00"."TYP_VYBAVENI" (NAZEV_VYBAVENI, DRUH_VYBAVENI, KVALITA, RARITA,
CENA) VALUES ('Mec obycejny', 'Zbran', 'Stary', 'Obycejny', '5');
INSERT INTO "DENE00"."TYP_VYBAVENI" (NAZEV_VYBAVENI, DRUH_VYBAVENI, KVALITA, RARITA,
CENA) VALUES ('Jablko', 'Jidlo', 'Zlepseny', 'Obycejny', '5');
INSERT INTO "DENE00"."TYP_VYBAVENI" (NAZEV_VYBAVENI, DRUH_VYBAVENI, KVALITA, RARITA,
CENA) VALUES ('Recept lektvaru neviditelnosti', 'Svitek', 'Rozbity', 'Vzacny', '300');
INSERT INTO "DENE00"."TYP_VYBAVENI" (NAZEV_VYBAVENI, DRUH_VYBAVENI, KVALITA, RARITA,
CENA) VALUES ('Stit dreveny', 'Brneni', 'Obycejny', 'Obycejny', '45');
INSERT INTO "DENE00"."TYP_VYBAVENI" (NAZEV_VYBAVENI, DRUH_VYBAVENI, KVALITA, RARITA,
CENA) VALUES ('Mango', 'Jidlo', 'Stary', 'Obycejny', '3');
INSERT INTO "DENE00"."TYP_VYBAVENI" (NAZEV_VYBAVENI, DRUH_VYBAVENI, KVALITA, RARITA,
CENA) VALUES ('Mapa zamku Orgrima', 'Jine', 'Obycejny', 'Epicky', '1000');
```

/\*INSERT INTO DOBRODRUZSTVI\*/

```
INSERT INTO "DENE00"."DOBRODRUZSTVI" (CISLO_DOBRODRUZSTVI, NAZEV_DOBRODRUZSTVI,
LOGIN, LOKALITA, OBTIZNOST, CIL) VALUES ('1','Co se deje v kralovstvi?', 'AntiGeroy', 'Velika pustina',
'Snadny', 'Najit vez maga');
INSERT INTO "DENE00"."DOBRODRUZSTVI" (CISLO_DOBRODRUZSTVI, NAZEV_DOBRODRUZSTVI,
LOGIN, LOKALITA, OBTIZNOST, CIL) VALUES ('2','Obrana tajemstvi maga ', 'Jenek', 'Tmavy les',
'Prumerny', 'Schovat stary svitek');
INSERT INTO "DENE00"."DOBRODRUZSTVI" (CISLO_DOBRODRUZSTVI, NAZEV_DOBRODRUZSTVI,
LOGIN, LOKALITA, OBTIZNOST, CIL) VALUES ('3','Starmalitit', 'Luntik', 'Stary utes', 'Obtizny',
'Prozkoumat jeskyni');
INSERT INTO "DENE00"."DOBRODRUZSTVI" (CISLO_DOBRODRUZSTVI, NAZEV_DOBRODRUZSTVI,
LOGIN, LOKALITA, OBTIZNOST, CIL) VALUES ('4','Ve jmenu krále', 'Ekdenka', 'Kralovstvi Ferefia',
'Snadny', 'Vybojovat zamek');
```

/\*INSERT INTO SEZENI\*/

```
INSERT INTO "DENE00"."SEZENI" (MISTO, DATUM, CISLO_DOBRODRUZSTVI, DOBA_TRVANI) VALUES
('Herni zona 1, stul 1', TO_DATE('15.5.11', 'DD.MM.RR'), '1', '240');
INSERT INTO "DENE00"."SEZENI" (MISTO, DATUM, CISLO_DOBRODRUZSTVI, DOBA_TRVANI) VALUES
('Herni zona 1, stul 1', TO_DATE('16.5.11', 'DD.MM.RR'), '1', '100');
INSERT INTO "DENE00"."SEZENI" (MISTO, DATUM, CISLO_DOBRODRUZSTVI, DOBA_TRVANI) VALUES
('Herni zona 1, stul 1', TO_DATE('17.6.11', 'DD.MM.RR'), '2', '100');
INSERT INTO "DENE00"."SEZENI" (MISTO, DATUM, CISLO_DOBRODRUZSTVI, DOBA_TRVANI) VALUES
('Herni zona 1, stul 2', TO_DATE('15.2.12', 'DD.MM.RR'), '3', '240');
```

/\*INSERT INTO UCASTNI\_SE\*/

```
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES ('Fer', '1');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES ('Kela',
'1');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES ('Qwiz',
'1');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES ('Fer', '2');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES ('Kela',
'2');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES ('Qwiz',
'2');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES ('Selom',
'2');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES ('Vila',
'3');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES ('Nurik',
'3');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES ('Vila',
'4');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES
('Steroj', '4');
INSERT INTO "DENE00"."UCASTNI_SE" (JMENO_POSTAVY, CISLO_DOBRODRUZSTVI) VALUES
('Terra', '4');
```

```
/*INSERT INTO UDALOST*/
```

```
INSERT INTO "DENE00"."UDALOST" (CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI, NAZEV_UDALOSTI,
DRUH_UDALOSTI, POPIS) VALUES ('1', '1', 'Nakup jídla', 'Nakup', 'V hospodě Tri tlustochove');
INSERT INTO "DENE00"."UDALOST" (CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI, NAZEV_UDALOSTI,
DRUH_UDALOSTI, POPIS) VALUES ('1', '2', 'První ukol pro hrdiny', 'Přijetí ulohy', 'Kdo v noci ji mrkev na
zahradě hospody?');
INSERT INTO "DENE00"."UDALOST" (CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI, NAZEV_UDALOSTI,
DRUH_UDALOSTI, POPIS) VALUES ('4', '1', 'BipBip', 'Jine', 'Spatná uloha');
INSERT INTO "DENE00"."UDALOST" (CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI, NAZEV_UDALOSTI,
DRUH_UDALOSTI, POPIS) VALUES ('2', '1', 'Zabití zradce', 'Vražda','');
INSERT INTO "DENE00"."UDALOST" (CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI, NAZEV_UDALOSTI,
DRUH_UDALOSTI, POPIS) VALUES ('2', '2', 'Zradce je mrtvý', 'Odevzdání ulohy', 'Hura! Jsme vyhráli');
```

```
/*INSERT INTO ZISKANI*/
```

```
INSERT INTO "DENE00"."ZISKANI" (NAZEV_VYBAVENI, CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI,
JMENO_POSTAVY, MNOZSTVI) VALUES ('Jabko', '1', '1', 'Fer', '2');
INSERT INTO "DENE00"."ZISKANI" (NAZEV_VYBAVENI, CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI,
JMENO_POSTAVY, MNOZSTVI) VALUES ('Recept lektvaru neviditelnosti', '1', '1', 'Kela', '1');
INSERT INTO "DENE00"."ZISKANI" (NAZEV_VYBAVENI, CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI,
JMENO_POSTAVY, MNOZSTVI) VALUES ('Mapa zamku Orgrima', '1', '2', 'Qwiz', '1');
INSERT INTO "DENE00"."ZISKANI" (NAZEV_VYBAVENI, CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI,
JMENO_POSTAVY, MNOZSTVI) VALUES ('Stit dreveny', '2', '1', 'Fer', '2');
INSERT INTO "DENE00"."ZISKANI" (NAZEV_VYBAVENI, CISLO_DOBRODRUZSTVI, CISLO_UDALOSTI,
JMENO_POSTAVY, MNOZSTVI) VALUES ('Stit dreveny', '2', '1', 'Nurik', '1');
```

## 5.2 Opis vložených dat

Tabulka HRAC

LOGIN	JMENO	EMAIL	DATUM_REGISTRACE
Ekdenka	Denisova Ekaterina	ekden@gmail.com	10.09.10
Jenek	Evgenia Boyarentseva	boy@gmail.com	10.09.10
AntiGeroy	Vladislav Ermolaev	ermol@seznam.cz	20.12.10
Grey	Václav Kupka	kupo@seznam.cz	20.12.10
Gideon	Franz Havel	frank@gmail.cz	18.01.11
Luntik	Luna Lovegood	polumna@seznam.cz	02.02.11
Gapo	Harry Potter	potter@seznam.cz	03.03.11

Tabulka POSTAVA

JMENO_POSTAVY	LOGIN	RASA	POVOLANI	UROVEN
Fer	Ekdenka	Elf	Klerik	20
Kela	Jenek	Pulelf	Druid	22
Qwiz	AntiGeroy	Clovek	Paladin	1
Selom	Grey	Trpaslik	Specialista	20
Vila	Gideon	Elf	Druid	2
Nurik	Luntik	Clovek	Paladin	15
Steroj	Luntik	Elf	Bard	1
Terra	Gideon	Pulcik	Hranicar	7
Ruin	AntiGeroy	Trpaslik	Mag	25

Tabulka TYP\_VYBAVENI

NAZEV_VYBAVENI	DRUH_VYBAVENI	KVALITA	RARITA	CENA
Mec obycejny	Zbran	Stary	Obycejny	5
Jablko	Jidlo	Zlepseny	Obycejny	5
Recept lektvaru neviditelnosti	Svitek	Rozbity	Vzacny	300
Stit dreveny	Brneni	Obycejny	Obycejny	45
Mango	Jidlo	Stary	Obycejny	3
Mapa zamku Orgrima	Jine	Obycejny	Epicky	1000

Tabulka DOBRODRUZSTVI

CISLO_DOBRODRUZSTVI	LOGIN	NAZEV_DOBRODRUZSTVI	LOKALITA	OBTIZNOST	CIL
1	AntiGeroy	Co se deje v kralovstvi?	Velika pustina	Snadny	Najit vez maga
2	Jenek	Obrana tajemstvi maga	Tmavy les	Prumerny	Schovat stary svitek
3	Luntik	Starmalitit	Stary utes	Obtizny	Prozkoumat jeskyni
4	Ekdenka	Ve jmenu krále	Kralovstvi Ferefia	Snadny	Vybojovat zamek

Tabulka SEZENI

MISTO	DATUM	CISLO_DOBRODRUZSTVI	DOBA_TRVANI
Herni zona 1, stul 1	15.05.11	1	240
Herni zona 1, stul 1	16.05.11	1	100
Herni zona 1, stul 1	17.06.11	2	100
Herni zona 1, stul 2	15.02.12	3	240

Tabulka UCASTNI\_SE

JMENO_POSTAVY	CISLO_DOBRODRUZSTVI
Fer	1
Kela	1
Qwiz	1
Fer	2
Kela	2
Qwiz	2
Selom	2
Vila	3
Nurik	3
Vila	4
Steroj	4
Terra	4

Tabulka UDALOST

CISLO_ DOBRODRUZSTVI	CISLO_ UDALOSTI	NAZEV_ UDALOSTI	DRUH_ UDALOSTI	POPIS
1	1	Nakup jídla	Nakup	V hospode Tri tlustochove
1	2	Prvni ukol pro hrdiny	Prijeti ulohy	Kdo v noci ji mrkev na zahrade hospody?
4	1	BipBip	Jine	Spatna uloha
2	1	Zabiti zradce	Vrazda	(null)
2	2	Zradce je mrtvy	Odevzdani ulohy	Hura! Jsme vyhrali

Tabulka ZISKANI

NAZEV_VYBAVENI	CISLO_ DOBRODRUZSTVI	CISLO_ UDALOSTI	JMENO_POSTAVY	MNOZSTVI
Jablko	1	1	Fer	2
Recept lektvaru neviditelnosti	1	1	Kela	1
Mapa zamku Orgrima	1	2	Qwiz	1
Stit dreveny	2	1	Fer	2
Stit dreveny	2	1	Nurik	1