



Vysoká škola ekonomická

Technická podpora datového centra

Dokumentace zkouškového příkladu pro předmět 4IT218 Databáze

Eliška Peníšková

Letní semestr 2017/2018

Uživatelské jméno: pene00

Body z testu SQL: 11

Body z testu DM: 18

Obsah

1	Popis zvolené výseče světa – zadání	4
2	Konceptuální schéma reality	5
3	Konceptuální datový model	6
4	Dokumentace databáze.....	7
4.1	Fyzický datový model	7
4.2	Definice relačních tabulek a souvisejících objektů.....	8
4.3	Integritní omezení	15
4.4	Definice přístupových práv.....	18
5	Obsah databáze	19
5.1	Vložení dat do databáze	19
5.2	Opis vložených dat.....	21

1 Popis zvolené výšeče světa – zadání

Pro potřeby technické podpory datového centra je nutné evidovat klienty, jimi využívané hostingové služby, tickety přiřazené těmto službám. Službou se rozumí fyzický server (DCH – dedikovaný server, CH- single rack server, DCF – dedicated farm, CF – farm housing, CR – rack housing), a/nebo virtuální server VS.

Každý klient využívá minimálně jednu nebo více hostingových služeb najednou. Každá služba musí mít přiřazeného právě jednoho konkrétního klienta. Všechny služby jsou monitorovány všemi zaměstnanci datacentra. Pokud klient má ke službě nějaký požadavek založí ticket. K jedné službě může klient založit více ticketů. Ticket se vždy týká jedné konkrétní služby. Služba nemusí mít vždy založen ticket. Dále evidujeme údaje o zaměstnancích. Zaměstnanci řeší tickety. Jeden ticket je vždy řešen jedním zaměstnancem.

Zaměstnanci jsou identifikováni loginem, který je ve formátu *jmeno.prijmeni* zaměstnance. Dále evidujeme údaje o zaměstnancích – jméno a příjmení, email, telefonní číslo je nepovinné.

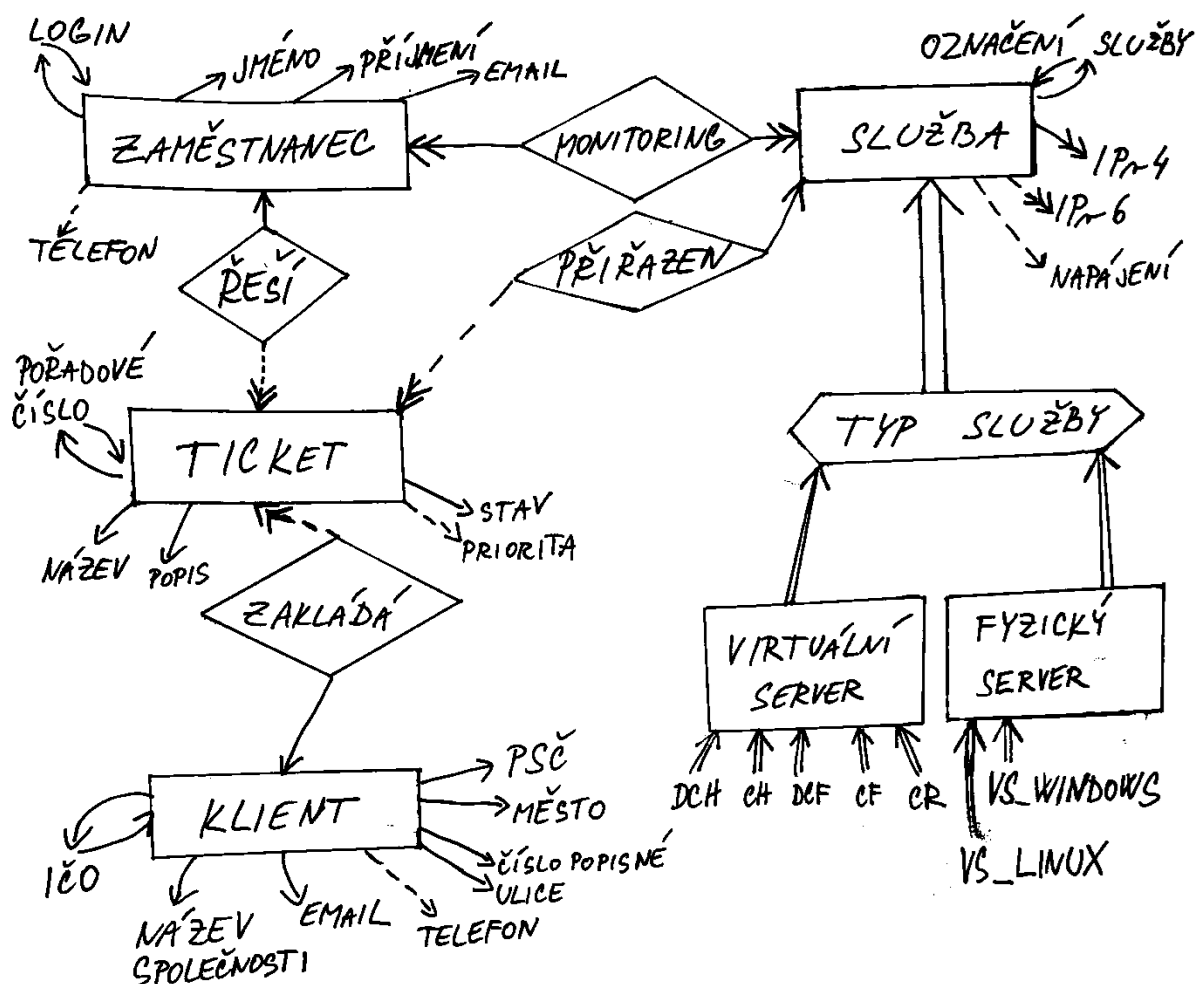
Identifikátor klienta je *IČO*, kterou daný člověk zastupuje. U klienta dále známe jeho email a adresu, telefonní kontakt je nepovinný.

Identifikátor služby je unikátní označení služby ve formátu *typ serveru číslo/rok spuštění serveru* – tedy např. VS2058/2016. Služba má minimálně jednu až několik IPv4 adres, služba může mít i IPv6 adresy. Služba může mít uvedené napájení.

Identifikátorem ticketu je pořadové číslo. Ticket má vždy název, popis, zadavatele, stav. Ticket může mít prioritu a řešitele. Řešitelem ticketu je zaměstnanec. Zadavatelem ticketu je klient.

2 Konceptuální schéma reality

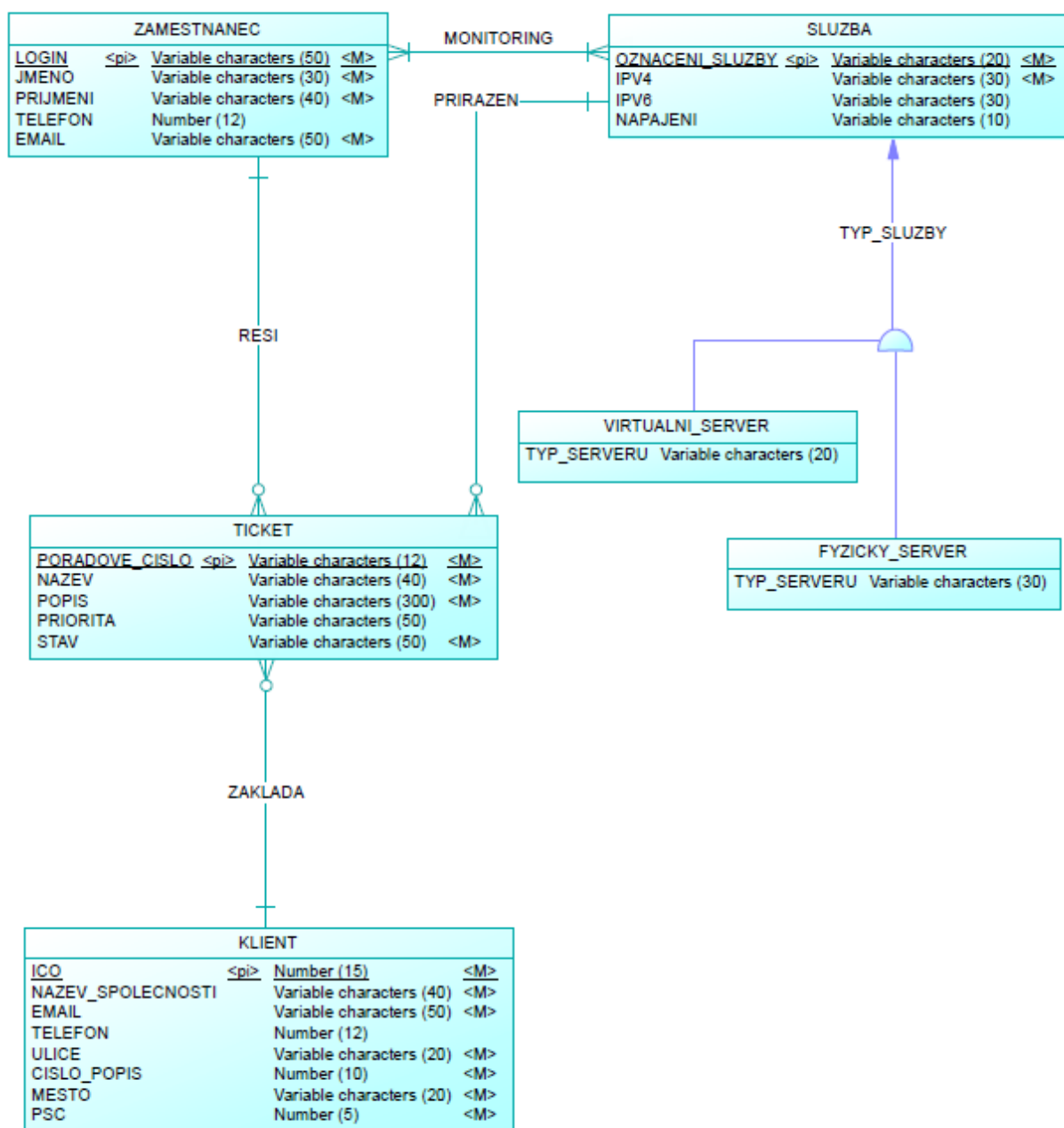
Na obrázku 1 je konceptuální schéma reality výšece světa nastíněné v kapitole 1 Popis zvolené výšece světa – zadání. Je použita notace dle přednášek předmětu 4IT218 Databáze v letním semestru akademického roku 2017/2018.



Obrázek 1: Konceptuální schéma reality zakreslené pomocí přednáškové notace

3 Konceptuální datový model

Na Obrázek 2 je konceptuální datový model vygenerovaný CASE nástrojem PowerDesigner®.



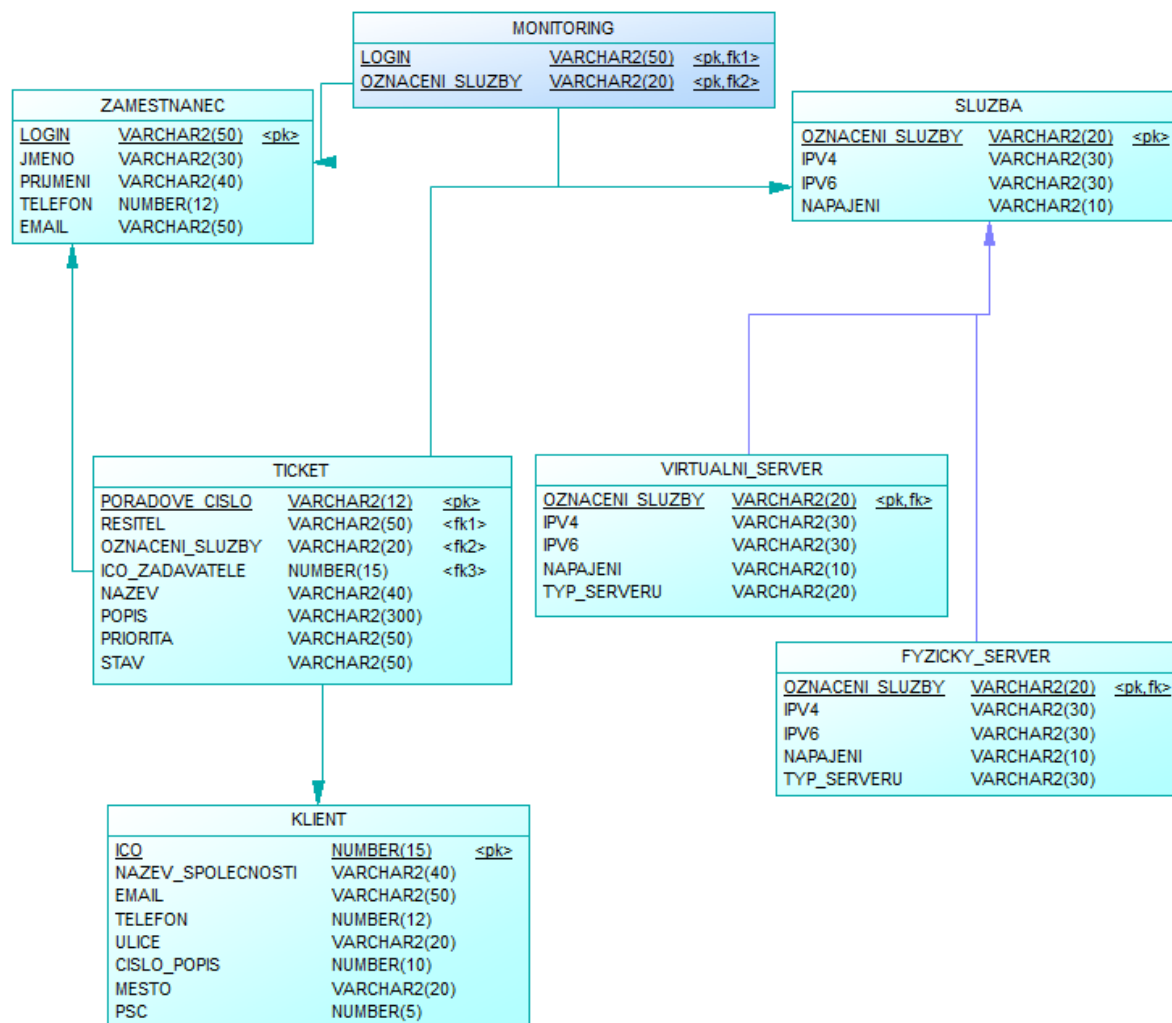
Obrázek 2: Konceptuální datový model vygenerovaný CASE nástrojem PowerDesigner®

4 Dokumentace databáze

V následující sekci bude postupně dokumentována technická stránka databáze. Od fyzického datového modelu bude probírána definice relačních tabulek a přidružených objektů, integritních omezení, přístupových práv, eventuálně dalších databázových objektů.

4.1 Fyzický datový model

Na Obrázek 3 je fyzický datový model vygenerovaný CASE nástrojem PowerDesigner®.



Obrázek 3: Fyzický datový model vygenerovaný CASE nástrojem PowerDesigner®

4.2 Definice relačních tabulek a souvisejících objektů

Zde je uveden veškerý SQL kód nutný pro definování databázových objektů a relací.

```
/*=====*/
/* DBMS name:   ORACLE Version 10g           */
/* Created on:   13. 5. 2018 13:04:32          */
/*=====*/

alter table FYZICKY_SERVER

drop constraint FK_FYZICKY__TYP_SLUZB_SLUZBA;

alter table MONITORING

drop constraint FK_MONITORI_MONITORIN_ZAM;

alter table MONITORING

drop constraint FK_MONITORI_MONITORIN_SLUZBA;

alter table TICKET

drop constraint FK_TICKET_PRIRAZEN_SLUZBA;

alter table TICKET

drop constraint FK_TICKET_RESI_ZAM;

alter table TICKET

drop constraint FK_TICKET_ZAKLADA_KLIENT;

alter table VIRTUALNI_SERVER

drop constraint FK_VIRTUALN_TYP_SLUZB_SLUZBA;

drop table FYZICKY_SERVER cascade constraints;

drop table KLIENT cascade constraints;

drop index MONITORING2_FK;

drop index MONITORING_FK;
```



```
drop table MONITORING cascade constraints;
drop table SLUZBA cascade constraints;
drop index ZAKLADA_FK;
drop index PRIRAZEN_FK;
drop index RESI_FK;
drop table TICKET cascade constraints;
drop table VIRTUALNI_SERVER cascade constraints;
drop table ZAM cascade constraints;
```

```
/*=====*/
/* Table: FYZICKY_SERVER */
/*=====*/
create table FYZICKY_SERVER (
    OZ_SLUZ      VARCHAR2(20)      not null,
    IPV4         VARCHAR2(30)      not null,
    IPV6         VARCHAR2(30),
    NPAJENI      VARCHAR2(10),
    TYP_SERVERU  VARCHAR2(30)
    constraint CKC_TYP_SERVERU_FYZICKY_ check (TYP_SERVERU is null or (TYP_SERVERU in
('DCH','CH','DCF','CF','CR'))),
    constraint PK_FYZICKY_SERVER primary key (OZ_SLUZ)
);
```

```
/*=====*/
/* Table: KLIENT */
/*=====*/
create table KLIENT (
    ICO          NUMBER(15)        not null,
    NAZEV_SPOLEC VARCHAR2(40)      not null,
    EMAIL        VARCHAR2(50)      not null
    constraint CKC_EMAIL_KLIENT check (EMAIL like '%_@_%._%'),
```

```
TELEFON      NUMBER(12),
ULICE        VARCHAR2(20)      not null,
CIS_POPIIS   NUMBER(10)        not null,
MESTO        VARCHAR2(20)      not null,
PSC          NUMBER(5)         not null,
constraint PK_KLIENT primary key (ICO)
);

/*=====*/
/* Table: MONITORING          */
/*=====*/

create table MONITORING (
  LOGIN       VARCHAR2(50)      not null
    constraint CKC_LOGIN_MONITORI check (LOGIN in like '%.%'),
  OZ_SLUZ     VARCHAR2(20)      not null
    constraint CKC_OZ_SLUZ_MONITORI check (OZ_SLUZ like '%/%'),
  constraint PK_MONITORING primary key (LOGIN, OZ_SLUZ)
);

/*=====*/
/* Index: MONITORING_FK          */
/*=====*/

create index MONITORING_FK on MONITORING (
  LOGIN ASC
);

/*=====*/
/* Index: MONITORING2_FK          */
/*=====*/

create index MONITORING2_FK on MONITORING (
  OZ_SLUZ ASC
);
```

```
/*=====*/
```

```
/* Table: SLUZBA */
```

```
/*=====*/
```

```
create table SLUZBA (
```

```
    OZ_SLUZ      VARCHAR2(20)      not null
    constraint CKC_OZ_SLUZ_SLUZBA check (OZ_SLUZ like '%/%'),
```

```
    IPV4         VARCHAR2(30)      not null,
```

```
    IPV6         VARCHAR2(30),
```

```
    NPAJENI      VARCHAR2(10),
```

```
    constraint PK_SLUZBA primary key (OZ_SLUZ)
```

```
);
```

```
/*=====*/
```

```
/* Table: TICKET */
```

```
/*=====*/
```

```
create table TICKET (
```

```
    PORAD_CIS    VARCHAR2(12)      not null,
```

```
    RESITEL      VARCHAR2(50),
```

```
    OZ_SLUZ      VARCHAR2(20)      not null,
```

```
    ICO_ZADAVATELE  NUMBER(15)      not null,
```

```
    NAZEV        VARCHAR2(40)      not null,
```

```
    POPIS        VARCHAR2(300)     not null,
```

```
    PRIORITA     VARCHAR2(50)
```

```
    constraint CKC_PRIORITA_TICKET check (PRIORITA is null or (PRIORITA in ('NIZKA','STREDNI','VYSOKA'))),
```

```
    STAV         VARCHAR2(50)      not null
```

```
    constraint CKC_STAV_TICKET check (STAV in ('NOVY','OTEVRENY','VYRESENÝ','ODLOZENÝ')),
```

```
    constraint PK_TICKET primary key (PORAD_CIS)
```

```
);
```

```
/*=====*/
/* Index: RESI_FK */
/*=====*/
create index RESI_FK on TICKET (
    RESITEL ASC
);
/*=====*/
/* Index: PRIRAZEN_FK */
/*=====*/
create index PRIRAZEN_FK on TICKET (
    OZ_SLUZ ASC
);
/*=====*/
/* Index: ZAKLADA_FK */
/*=====*/
create index ZAKLADA_FK on TICKET (
    ICO_ZADAVATELE ASC
);
/*=====*/
/* Table: VIRTUALNI_SERVER */
/*=====*/
create table VIRTUALNI_SERVER (
    OZ_SLUZ      VARCHAR2(20)      not null,
    IPV4        VARCHAR2(30)      not null,
    IPV6        VARCHAR2(30),
    NPAJENI     VARCHAR2(10),
    TYP_SER     VARCHAR2(20)      not null
    constraint CKC_TYP_SER_VIRTUALN check (TYP_SER in ('VS_WINDOWS','VS_LINUX')),
    constraint PK_VIRTUALNI_SERVER primary key (OZ_SLUZ)
);
```

```
/*=====*/
/* Table: ZAM */
/*=====*/
create table ZAM (
  LOGIN      VARCHAR2(50)      not null
    constraint CKC_LOGIN_ZAM check (LOGIN like '%.%') ENABLE,
  JMENO      VARCHAR2(30)      not null,
  PRIJMENI   VARCHAR2(40)      not null,
  TELEFON    NUMBER(12),
  EMAIL      VARCHAR2(50)      not null
    constraint CKC_EMAIL_ZAM check (EMAIL like '%_@_%._%') ENABLE,
  constraint PK_ZAM primary key (LOGIN)
);
```

```
alter table FYZICKY_SERVER
  add constraint FK_FYZICKY__TYP_SLUZB_SLUZBA foreign key (OZ_SLUZ)
    references SLUZBA (OZ_SLUZ)
    on delete cascade;
```

```
alter table MONITORING
  add constraint FK_MONITORI_MONITORIN_ZAM foreign key (LOGIN)
    references ZAM (LOGIN)
    on delete cascade;
```

```
alter table MONITORING
  add constraint FK_MONITORI_MONITORIN_SLUZBA foreign key (OZ_SLUZ)
    references SLUZBA (OZ_SLUZ)
    on delete cascade;
```

alter table TICKET

add constraint FK_TICKET_PRIRAZEN_SLUZBA foreign key (OZ_SLUZ)

references SLUZBA (OZ_SLUZ)

on delete cascade;

alter table TICKET

add constraint FK_TICKET_RESI_ZAM foreign key (RESITEL)

references ZAM (LOGIN)

on delete set null;

alter table TICKET

add constraint FK_TICKET_ZAKLADA_KLIENT foreign key (ICO_ZADAVATELE)

references KLIENT (ICO)

on delete cascade;

alter table VIRTUALNI_SERVER

add constraint FK_VIRTUALN_TYP_SLUZH_SLUZBA foreign key (OZ_SLUZ)

references SLUZBA (OZ_SLUZ)

on delete cascade;

4.3 Integritní omezení

V následujících nadpisech je ponechána diakritika kvůli jejich snazšímu čtení. Příslušné relační tabulky jsou však ve vytvořené databázi uloženy bez diakritiky.

4.3.1 Tabulka Zaměstnanec

Entitní integrita

Primární klíč tvoří atribut LOGIN.

SQL skript pro definici primárního klíče je: `constraint PK_ZAM primary key (LOGIN)`

Doménová integrita

Email musí obsahovat alespoň jeden znak, poté „@“, poté alespoň dva znaky, poté tečku a poté alespoň dva znaky.

SQL skript pro tuto definici: `constraint CKC_EMAIL_ZAM check (EMAIL like '%_@__%.__%')`

Login musí obsahovat nějaký počet znaků, tečku, nějaký počet znaků.

SQL skript pro tuto definici: `constraint CKC_LOGIN_ZAM check (LOGIN like '%.%')`

Referenční integrita

Zde není definována.

4.3.2 Tabulka Ticket

Entitní integrita

Primární klíč tvoří atribut PORADOVE_CISLO.

SQL skript pro definici primárního klíče je: `constraint PK_TICKET primary key (PORAD_CIS)`

Doménová integrita

Stav ticketu musí nabývat hodnot NOVY, OTEVRENY, VYRESENY, ODLOZENY.

SQL skript pro definici tohoto omezení je: `constraint CKC_STAV_TICKET check (STAV in ('NOVY','OTEVRENY','VYRESENY','ODLOZENY'))`

Priorita musí nabývat hodnot NIZKA, STREDNI, VYSOKA nebo je ponechána prázdná.

SQL skript pro definici tohoto omezení je: `constraint CKC_PRIORITA_TICKET check (PRIORITA is null or (PRIORITA in ('NIZKA','STREDNI','VYSOKA')))`

Referenční integrita

Tabulka je propojená pomocí klíče OZ_SLUZ s tabulkou SLUZBA. Pokud je smazán záznam v tabulce SLUZBA, všechny související tickety jsou smazány.

```
alter table TICKET
```

```
add constraint FK_TICKET_PRIRAZEN_SLUZBA foreign key (OZ_SLUZ)
references SLUZBA (OZ_SLUZ)
on delete cascade;
```

Tabulka je propojená pomocí klíče RESITEL s tabulkou ZAM. Pokud je smazán záznam v tabulce ZAM, související hodnota ve sloupci RESITEL je nahrazena prázdnou hodnotou.

```
alter table TICKET
```

```
add constraint FK_TICKET_RESI_ZAM foreign key (RESITEL)
references ZAM (LOGIN)
on delete set null;
```

Tabulka je propojená pomocí klíče ICO_ZADAVATELE s tabulkou KLIENT. Pokud je smazán záznam v tabulce KLIENT, všechny související tickety jsou smazány.

```
alter table TICKET
add constraint FK_TICKET_ZAKLADA_KLIENT foreign key (ICO_ZADAVATELE)
references KLIENT (ICO)
on delete cascade;
```

4.3.3 Tabulka Klient

Entitní integrita

Primární klíč tvoří atribut ICO.

SQL skript pro definici primárního klíče je: constraint PK_KLIENT primary key (ICO)

Doménová integrita

Email klienta musí obsahovat alespoň jeden znak, poté „@“, poté alespoň dva znaky, poté tečku a poté alespoň dva znaky.

SQL skript pro tuto definici: constraint CKC_EMAIL_KLIENT check (EMAIL like '%_@__%.__%')

Referenční integrita

Zde není definována.

4.3.4 Služba

Entitní integrita

Primární klíč tvoří atribut OZNACENI_SLUZBY.

SQL skript pro definici primárního klíče je: constraint PK_SLUZBA primary key (OZ_SLUZ)

Doménová integrita

OZNACENI_SLUZBY je ve formátu nějaký počet znaků / nějaký počet znaků.

SQL skript pro tuto definici: constraint CKC_OZ_SLUZ_SLUZBA check (OZ_SLUZ like '%/%')

Referenční integrita

Zde není definována.

4.3.5 Tabulka Virtuální server

Entitní integrita

Primární klíč tvoří atribut PORADOVE_CISLO.

SQL skript pro definici primárního klíče je: constraint PK_TICKET primary key (PORAD_CIS)

Doménová integrita

Typ serveru může nabývat pouze hodnot VS_WINDOWS, VS_LINUX nebo null.

SQL skript pro definici tohoto omezení je: constraint CKC_TYP_SER_VIRTUALN check (TYP_SER in ('VS_WINDOWS','VS_LINUX'))

Referenční integrita

Tabulka je propojená pomocí klíče OZ_SLUZ s tabulkou SLUZBA. Pokud je smazán záznam v tabulce SLUZBA, všechny související virtuální servery jsou smazány.

SQL skript pro definici tohoto omezení je:

```
alter table VIRTUALNI_SERVER
add constraint FK_VIRTUALN_TYP_SLUZB_SLUZBA foreign key (OZ_SLUZ)
references SLUZBA (OZ_SLUZ)
on delete cascade;
```

4.3.6 Tabulka Fyzický server

Entitní integrita

Primární klíč tvoří atribut PORADOVE_CISLO.

SQL skript pro definici primárního klíče je: constraint PK_TICKET primary key (PORAD_CIS)

Doménová integrita

Typ serveru může nabývat pouze hodnot DCH, CH, DCF, CF nebo CR.

SQL skript pro definici tohoto omezení je: constraint CKC_TYP_SERVERU_FYZICKY_ check (TYP_SERVERU is null or (TYP_SERVERU in ('DCH','CH','DCF','CF','CR')))

Referenční integrita

Tabulka je propojená pomocí klíče OZ_SLUZ s tabulkou SLUZBA. Pokud je smazán záznam v tabulce SLUZBA, všechny související fyzické servery jsou smazány.

SQL skript pro definici tohoto omezení je:

```
alter table FYZICKY_SERVER
add constraint FK_FYZICKY__TYP_SLUZB_SLUZBA foreign key (OZ_SLUZ)
references SLUZBA (OZ_SLUZ)
on delete cascade;
```

4.3.7 Tabulka Monitoring

Entitní integrita

Primární klíče tvoří atributy LOGIN a OZ_SLUZ.

SQL skript pro definici primárního klíče je: constraint PK_MONITORING primary key (LOGIN, OZ_SLUZ)

Doménová integrita

Login musí obsahovat nějaký počet znaků, tečku, nějaký počet znaků.

SQL skript pro tuto definici: constraint CKC_LOGIN_MONITORI check (LOGIN like '%.%')

OZNACENI_SLUZBY je ve formátu nějaký počet znaků / nějaký počet znaků.

SQL skript pro tuto definici: constraint CKC_OZ_SLUZ_MONITORI check (OZ_SLUZ like '%/%')

Referenční integrita

Tabulka je propojená pomocí klíče OZ_SLUZ s tabulkou SLUZBA. Pokud je smazán záznam v tabulce SLUZBA, všechny související záznamy jsou smazány.

SQL skript pro tuto definici:

```
alter table MONITORING
add constraint FK_MONITORI_MONITORIN_SLUZBA foreign key (OZ_SLUZ)
references SLUZBA (OZ_SLUZ)
on delete cascade;
```

Tabulka je propojená pomocí klíče LOGIN s tabulkou ZAM. Pokud je smazán záznam v tabulce ZAM, všechny související záznamy jsou smazány.

SQL skript pro tuto definici :

```
alter table MONITORING
add constraint FK_MONITORI_MONITORIN_ZAM foreign key (LOGIN)
references ZAM (LOGIN)
on delete cascade;
```

4.4 Definice přístupových práv

Zde uvádím SQL příkazy, kterými byla definována přístupová práva k vloženým datům pro uživatele:

a) STUDENT: práva pro operaci SELECT

```
grant select on ZAM to STUDENT;
grant select on TICKET to STUDENT;
grant select on KLIENT to STUDENT;
grant select on MONITORING to STUDENT;
grant select on SLUZBA to STUDENT;
grant select on VIRTUALNI_SERVER to STUDENT;
grant select on FYZICKY_SERVER to STUDENT;
```

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

```
grant select, insert, update, delete on ZAM to IT218;
grant select, insert, update, delete on TICKET to IT218;
grant select, insert, update, delete on KLIENT to IT218;
grant select, insert, update, delete on MONITORING to IT218;
grant select, insert, update, delete on SLUZBA to IT218;
grant select, insert, update, delete on VIRTUALNI_SERVER to IT218;
grant select, insert, update, delete on FYZICKY_SERVER to IT218;
```

5 Obsah databáze

5.1 Vložení dat do databáze

V této podkapitole jsou uvedeny SQL příkazy, kterými byli vloženy data do jednotlivých tabulek.

Tabulka KLIENT:

```
INSERT INTO "PENE00"."KLIENT" (ICO, NAZEV_SPOLEC, EMAIL, TELEFON, ULICE, CIS_POPIS, MESTO, PSC) VALUES ('23456782', 'Abc', 'abc@abc.cz', '752585369', 'Sokolovská', '5', 'Praha', '18005')
```

```
INSERT INTO "PENE00"."KLIENT" (ICO, NAZEV_SPOLEC, EMAIL, TELEFON, ULICE, CIS_POPIS, MESTO, PSC) VALUES ('15158458', 'Fenix', 'fenix@fenix.de', '741258741', 'Vysoká', '8', 'Brno', '25802')
```

```
INSERT INTO "PENE00"."KLIENT" (ICO, NAZEV_SPOLEC, EMAIL, TELEFON, ULICE, CIS_POPIS, MESTO, PSC) VALUES ('87842664', 'Cisco', 'cisco@cisco.com', '789652854', 'Okružní', '1', 'Ostrava', '12050')
```

Tabulka ZAMESTNANEC:

```
INSERT INTO "PENE00"."ZAM" (LOGIN, JMENO, PRIJMENI, TELEFON, EMAIL) VALUES ('eliska.peniskova', 'Eliška', 'Peníšková', '721445854', 'eliska@gmail.com')
```

```
INSERT INTO "PENE00"."ZAM" (LOGIN, JMENO, PRIJMENI, TELEFON, EMAIL) VALUES ('petr.freitag', 'Petr', 'Freitag', '722333444', 'petr@gmail.com')
```

```
INSERT INTO "PENE00"."ZAM" (LOGIN, JMENO, PRIJMENI, TELEFON, EMAIL) VALUES ('michal.novotny', 'Michal', 'Novotný', '722333443', 'michal@gmail.com')
```

Tabulka SLUZBA:

```
INSERT INTO "PENE00"."SLUZBA" (OZ_SLUZ, IPV4, IPV6, NAPAJENI) VALUES ('VS1515/2016', '182.144.87.233', '51515a2651651aa', 'A52')
```

```
INSERT INTO "PENE00"."SLUZBA" (OZ_SLUZ, IPV4, IPV6, NAPAJENI) VALUES ('VS2522/2017', '147.122.36.211', '21851a1511a1198', 'B15')
```

```
INSERT INTO "PENE00"."SLUZBA" (OZ_SLUZ, IPV4, IPV6, NAPAJENI) VALUES ('VS2511/2018', '133.111.1.221', '1981a19181a1981', 'E81')
```

```
INSERT INTO "PENE00"."SLUZBA" (OZ_SLUZ, IPV4, IPV6, NAPAJENI) VALUES ('DCH151/2015', '147.131.36.211', '54a18181a1181a18', 'A26')
```

```
INSERT INTO "PENE00"."SLUZBA" (OZ_SLUZ, IPV4, IPV6, NAPAJENI) VALUES ('CH252/2018', '147.135.36.211', '189181a51958a195', 'B58')
```

```
INSERT INTO "PENE00"."SLUZBA" (OZ_SLUZ, IPV4, IPV6, NAPAJENI) VALUES ('CF3954/2017', '133.221.1.221', '4981a191515a1915', 'A56')
```

Tabulka TICKET:

```
INSERT INTO "PENE00"."TICKET" (PORAD_CIS, RESITEL, OZ_SLUZ, ICO_ZADAVATELE, NAZEV, POPIS, PRIORITA, STAV) VALUES ('1915151', 'eliska.peniskova', 'VS1515/2016', '23456782', 'Problém', 'Eviduji problém se serverem.', 'NIZKA', 'NOVY')
```

```
INSERT INTO "PENE00"."TICKET" (PORAD_CIS, RESITEL, OZ_SLUZ, ICO_ZADAVATELE, NAZEV, POPIS, PRIORITA, STAV) VALUES ('1915152', 'petr.freitag', 'CH252/2018', '15158458', 'Restart', 'Prosim o restart serveru.', 'STREDNI', 'OTEVRENY')
```

```
INSERT INTO "PENE00"."TICKET" (PORAD_CIS, RESITEL, OZ_SLUZ, ICO_ZADAVATELE, NAZEV, POPIS, PRIORITA, STAV) VALUES ('1915153', 'michal.novotny', 'CF3954/2017', '87842664', 'Nefunguje', 'Server nefunguje.', 'VYSOKA', 'VYRESENY')
```

MONITORING

```
INSERT INTO "PENE00"."MONITORING" (LOGIN, OZ_SLUZ) VALUES ('eliska.peniskova', 'VS1515/2016')
```

```
INSERT INTO "PENE00"."MONITORING" (LOGIN, OZ_SLUZ) VALUES ('petr.freitag', 'CH252/2018')
```

```
INSERT INTO "PENE00"."MONITORING" (LOGIN, OZ_SLUZ) VALUES ('michal.novotny', 'CF3954/2017')
```

FYZICKY_SERVER

```
INSERT INTO "PENE00"."FYZICKY_SERVER" (OZ_SLUZ, IPV4, IPV6, NAPAJENI, TYP_SERVERU) VALUES ('DCH151/2015', '147.131.36.211', '54a18181a1181a18', 'A26', 'DCH')
```

```
INSERT INTO "PENE00"."FYZICKY_SERVER" (OZ_SLUZ, IPV4, IPV6, NAPAJENI, TYP_SERVERU) VALUES ('CH252/2018', '147.135.36.211', '189181a51958a195', 'B58', 'CH')
```

```
INSERT INTO "PENE00"."FYZICKY_SERVER" (OZ_SLUZ, IPV4, IPV6, NAPAJENI, TYP_SERVERU) VALUES ('CF3954/2017', '133.221.1.221', '4981a191515a1915', 'A56', 'CF')
```

VIRTUALNI_SERVER

```
INSERT INTO "PENE00"."VIRTUALNI_SERVER" (OZ_SLUZ, IPV4, IPV6, NAPAJENI, TYP_SER) VALUES ('VS1515/2016', '182.144.87.233', '51515a2651651aa', 'A52', 'VS_LINUX')
```

```
INSERT INTO "PENE00"."VIRTUALNI_SERVER" (OZ_SLUZ, IPV4, IPV6, NAPAJENI, TYP_SER) VALUES ('VS2522/2017', '147.122.36.211', '21851a1511a1198', 'B15', 'VS_LINUX')
```

```
INSERT INTO "PENE00"."VIRTUALNI_SERVER" (OZ_SLUZ, IPV4, IPV6, NAPAJENI, TYP_SER) VALUES ('VS2511/2018', '133.111.1.221', '1981a19181a1981', 'E81', 'VS_WINDOWS')
```

5.2 Opis vložených dat

Tabulka ZAMESTNANEC

LOGIN	JMENO	PRIJMENI	EMAIL	TELEFON
eliska.peniskova	Eliška	Peníšková	eliska@gmail.com	721445854
petr.freitag	Petr	Freitag	petr@gmail.com	722333444
michal.novotny	Michal	Novotný	michal@gmail.com	722333443

Tabulka TICKET

PORADOVE_CISLO	RESITEL	OZNACENI_SLUZBY	ICO_ZADAVATELE
1915151	eliska.peniskova	VS1515/2016	23456782
1915152	petr.freitag	CH252/2018	15158458
1915153	michal.novotny	CF3954/2017	87842664

NAZEV	POPIS	PRIORITA	STAV
Problém	Eviduji problém se serverem.	NIZKA	NOVY
Restart	Prosim o restart serveru.	STREDNI	OTEVRENY
Nefunguje	Server nefunguje.	VYSOKA	VYRESENÝ

Tabulka KLIENT

ICO_	NAZEV_SPOLECNOSTI	EMAIL	TELEFON	ULICE	CISLO_POPIS	MESTO	PSC
23456782	Abc	abc@abc.cz	752585369	Sokolovská	5	Praha	18005
15158458	Fenix	fenix@fenix.de	741258741	Vysoká	8	Brno	25802
87842664	Cisco	cisco@cisco.com	789652854	Okružní	1	Ostrava	12050

Tabulka SLUZBA

OZNACENI_SLUZBY	IPV4	IPV6	NAPAJENI
CF3954/2017	133.221.1.221	4981a191515a1915	A56
DCH151/2015	147.131.36.211	54a18181a1181a18	DCH
CH252/2018	147.135.36.211	189181a51958a195	B58
VS1515/2016	182.144.87.233	51515a2651651aa	A52
VS2511/2018	133.111.1.221	1981a19181a1981	E81
VS2522/2017	147.122.36.211	21851a1511a1198	B15

Tabulka VIRTUALNI_SERVER

OZNACENI_SLUZBY	IPV4	IPV6	NAPAJENI	TYP_SERVERU
VS1515/2016	182.144.87.233	51515a2651651aa	A52	VS_LINUX
VS2522/2017	147.122.36.211	21851a1511a1198	B15	VS_LINUX
VS2511/2018	133.111.1.221	1981a19181a1981	E81	VS_WINDOWS

Tabulka FYZICKY_SERVER

OZNACENI_SLUZBY	IPV4	IPV6	NAPAJENI	TYP_SERVERU
DCH151/2015	147.131.36.211	54a18181a1181a18	A26	DCH
CH252/2018	147.135.36.211	189181a51958a195	B58	CH
CF3954/2017	133.221.1.221	4981a191515a1915	A56	CF

Tabulka MONITORING

LOGIN	OZNACENI_SLUZBY
eliska.peniskova	VS1515/2016
petr.freitag	CH252/2018
michal.novotny	CF3954/2017