



ČVUT

ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

Definice databáze

Databázové systémy 2023/24





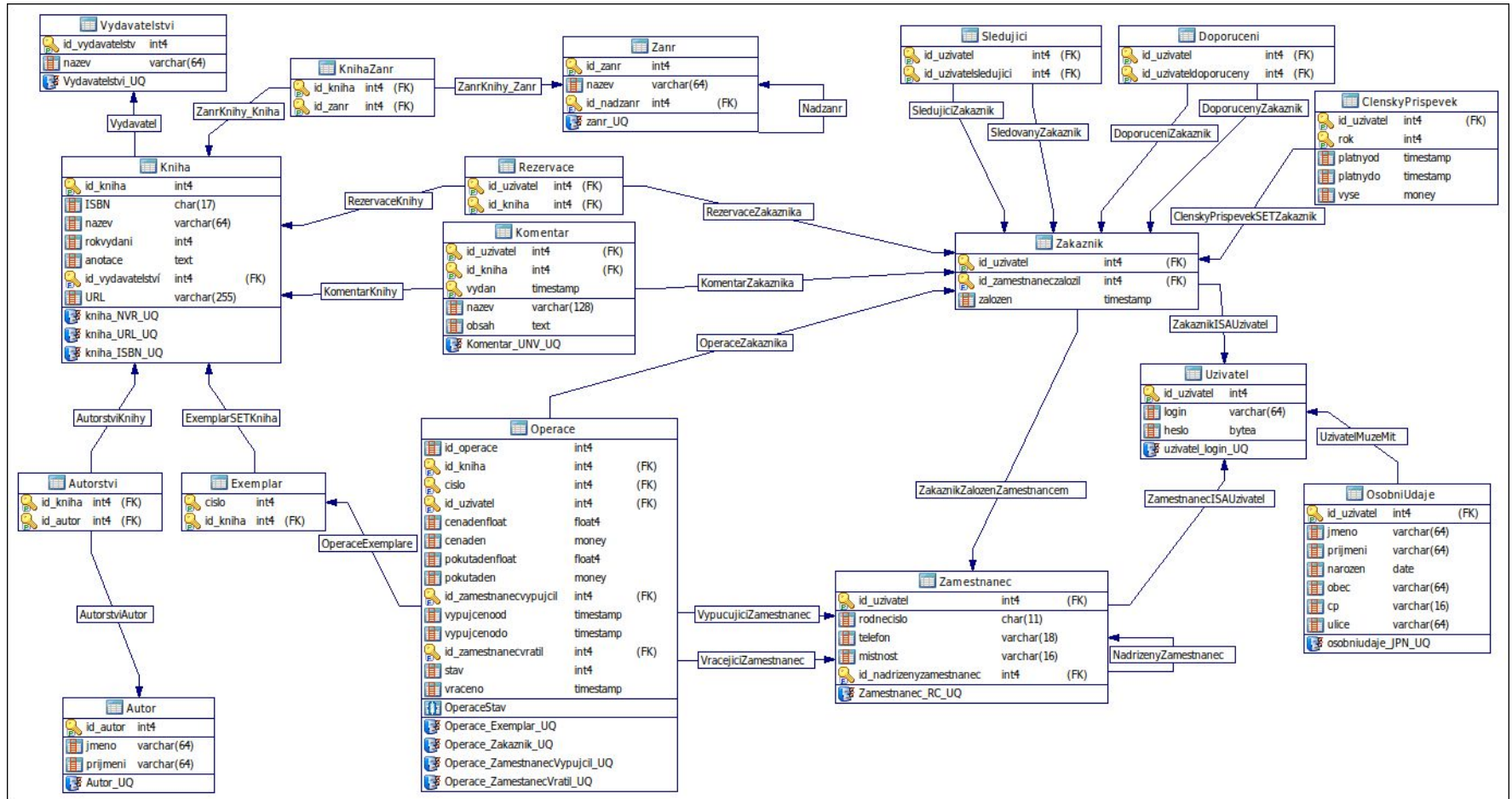
Definice databáze ER diagram

Cíl:

- Vytvořit z relačního modelu schéma implementovatelné v konkrétním řídicím databázovém systému
 - Výběr systému:
 - NULL hodnoty
 - podpora vhodných datových typů
 - textové klíče
 - procedurální zpracování (chytrá databáze, “hloupá” aplikace)








Použití:

- Založení tabulek v databázi
- Definice typů atributů (dle zvoleného řídicího databázového systému)
- Definice atributových a tabulkových omezení
 - Včetně referenční integrity (cizí klíče)
- Zavedení defaultních hodnot atributů
- Zavedení umělých klíčů



Každá relace je transformována do tabulky

- Definice tabulky - atributy:
 - jméno atributu
 - typ atributu
 - integritní omezení - atributové (implicitní)
 - integritní omezení - tabulkové

Kniha		
	id_kniha	int4
	ISBN	varchar(16)
	nazev	varchar(64)
	rokvydani	int4
	anotace	text
	id_vydavatelství	int4 (FK)
	kniha-nazev-vydavatelstvi-rok-UQ	

Kniha(id_kniha, ISBN, název, id_vydavatel, rok vydání, anotace)

```
CREATE TABLE kniha (
  id_kniha          serial          PRIMARY KEY,
  ISBN              char(17)        NOT NULL UNIQUE,
  nazev             varchar(127)    NOT NULL,
  id_vydavatel      int4            NOT NULL REFERENCES vydavatel,
  rokvydani         smallint        NOT NULL
                    CHECK (rok BETWEEN 1445 AND EXTRACT(YEAR FROM CURRENT_DATE)) ,
  anotace           text            NULL,
  UNIQUE (id_vydavatel, nazev, rok)
);
```

Definice databáze

Ukázka transformace z relačního modelu

PK	UQ	UQ			
id_kniha	ISBN	Název	Vydavatel (FK)	Rok vydání	Anotace
1	978-80-859-2617-0	To kalné ráno	Dita	1997	Černobílé fotografie...
2	978-80-754-6131-5	Válka s Mloky	Fortuna Libri	2017	
3	978-80-879-5041-8	Válka s Mloky	Československý spisovatel	1981	Toto dílo se pohybuje ...
4	978-80-7459-175-4	Stopy mých otců	Československý spisovatel	2018	Helena Šmahelová zpracovala...
5	978-80-7390-988-8	Rituál	Fobos	2020	
6	978-80-7303-433-7	Rituál	Domino	2009	Policejní potápěčka ...
7	978-80-253-4395-1	Rituál	Fragment	2019	
8	978-80-7577-821-5	Rituál	Host	2019	Rituál — tajemný obřad...
9	978-16-8369-417-5	The Auctioneer	Quirk Publishing	2020	Harrowing tensions...
10	978-80-277-1210-6	Dobří sousedé	Fobos	2023	

Definice databáze

Atributy - volba typů atributů

Typ atributu má výhradně odpovídat významu(doméně) atributu, viz např:

- **Řetězcové**
 - **char(n)** řetězec pevné délky n
 - **varchar(n)** řetězec proměnné délky 0..n
 - **text** řetězec neomezené délky
- **Číselné**
 - **int, smallint** celočíselné (různého rozsahu)
 - **float, double** reálné s pohyblivou řádovou čárkou a omezením přesnosti.
 - **numeric (n,m)** číslo s n platnými číslicemi, z toho m za desetinou tečkou.
- **Temporální**
 - **date, time, timestamp, timestamp with time zone**
- **Logické**
 - **boolean**
- **Binární**
 - **bytea**

Relační model:

Kniha(ISBN, název, vydavatel, rok vydání)

AnotaceKnihy (ISBN, anotace)

FK: (ISBN) \subseteq Kniha(ISBN)

Upravený relační model:

Kniha(ISBN, název, vydavatel, rok vydání, anotace)

SQL:

```
CREATE TABLE kniha (  
    ...  
    anotace text NULL,  
    ...  
);
```

Definice databáze

Modifikace relačního modelu - ENUM

Relační model:

Kniha(ISBN, název, vydavatel, rok vydání, anotace)

Upravený relační model:

Vydavatel (id_vydavatel, název)

Kniha(ISBN, název, id_vydavatel, rok vydání, anotace)

FK: (id_vydavatel) \subseteq Vydavatel(id_vydavatel)

Poznámka:

Implementace ENUM pomocí CHECK (vydavatel IN ('Dita', 'Fortuna',...)) jest přípustná, avšak změna v tomto typu znamená změnu integritního omezení (zásah do schématu databáze). Zároveň může znamenat větší paměťové nároky.

*Tabulky implementující ENUM běžně nazýváme **číselníky**.*

Definice databáze

Zavedení umělých primárních klíčů (ID)

Umělý klíč definujeme **POUZE** pro případ, kdy:

- ***Klíč je takového typu, že jeho hodnotu nelze porovnat v $O(1)$***
 - Složitost porovnávání řetězců od délce n je až $O(n)$
- ***Klíč se skládá z více atributů***
 - Porovnání hodnoty jednoho atributu vs více atributů
 - Neplatí pro slabý entitní typ
- ***Použité klíče závisí na externích normách, které mohou být***
 - změněny (budoucnost)
 - dodrženy, avšak chybně přiřazeny (chyba vstupních dat)
 - např:
 - Různým knihám bylo přiřazeno stejné ISBN
 - Různým lidem bylo přiřazeno stejné rodné číslo
 - Možnost v dané (provizorní) situaci použít nadklíč
- **VŽDY nutno zachovat korespondenci mezi umělými a všemi (přirozenými) klíči**

Umělý klíč *id_tabulka* definujeme:

- *Pseudotypem SERIAL (Postgresql)*

```
CREATE TABLE kniha (  
    id_kniha serial PRIMARY KEY,  
    ...  
);
```

- *Integritním omezením **autoincrement** (mimo normu SQL - jiné systémy)*
- *Sekvencí a nastavením defaultní hodnoty*

```
CREATE SEQUENCE sq_kniha;  
CREATE TABLE kniha (  
    id_kniha int PRIMARY KEY DEFAULT nextval('sq_kniha'),  
    ...  
);
```



Definice databáze

Zavedení klíčů - UNIQUE NOT NULL

Pro každý identifikátor (který je v relačním modelu podtržen) musí být nastaveno omezení UNIQUE NOT NULL.

Kniha(id_kniha, ISBN, název, id_vydavatel, rok_vydání, anotace)

```
CREATE TABLE kniha (  
    id_kniha          serial          PRIMARY KEY,  
    ISBN             char(17)        NOT NULL UNIQUE,  
    nazev            varchar(127)    NOT NULL,  
    id_vydavatel     int4            NOT NULL REFERENCES vydavatel,  
    rokvydani        smallint        NOT NULL ...  
    anotace          text            NULL,  
    UNIQUE (id_vydavatel, nazev, rok)  
);
```



Definice databáze

Zavedení klíčů - UNIQUE NOT NULL

Některá atributová omezení je možné přepsat jako tabulková.

Kniha(id_kniha, ISBN, název, id_vydavatel, rok vydání, anotace)

```
CREATE TABLE kniha (  
    id_kniha          serial          ,  
    ISBN              char(17)        NOT NULL,  
    nazev             varchar(127)    NOT NULL,  
    id_vydavatel      int4            NOT NULL REFERENCES vydavatel...,  
    rokvydani         smallint        NOT NULL ...,  
    anotace           text            NULL,  
    PRIMARY KEY (id_kniha),  
    UNIQUE (ISBN),  
    UNIQUE (id_vydavatel, nazev, rok)  
);
```

Definice databáze

Zavedení klíčů - UNIQUE NOT NULL

Integritní omezení je možné pojmenovat (pokud není uvedeno, pojmenuje systém automaticky)

Kniha(id_kniha, ISBN, název, id_vydavatel, rok vydání, anotace)

```
CREATE TABLE kniha (  
  id_kniha          serial          ,  
  ISBN              char(17)        NOT NULL,  
  nazev             varchar(127)    NOT NULL,  
  id_vydavatel     int4             NOT NULL REFERENCES vydavatel,  
  rok               smallint        NOT NULL ...,  
  anotace           text            NULL,  
  CONSTRAINT kniha PK PRIMARY KEY (id_kniha),  
  CONSTRAINT kniha ISBN UQ UNIQUE (ISBN),  
  CONSTRAINT kniha VNR UQ UNIQUE (id_vydavatel, nazev, rok)  
);
```

Cizí klíče se implementují pomocí **referenční integrity** - zajišťuje korespondenci hodnot cizího klíče s hodnotami primárního klíče.

Vydavatel (id_vydavatel, název)

Kniha (id_kniha, ISBN, název, id_vydavatel, rok vydání, anotace)

FK: (id_vydavatel) \subseteq Vydavatel(id_vydavatel)

```
CREATE TABLE kniha (  
...  
  id_vydavatel      int4  
                    NOT NULL  
                    REFERENCES vydavatel,  
...  
);
```



Definice databáze

Zavedení cizích klíčů - 1:N - Referenční integrita

Cizí klíče se implementují pomocí **referenční integrity** - zajišťuje korespondenci hodnot cizího klíče s hodnotami primárního klíče.

Vydavatel (id_vydavatel, název)

Kniha (id_kniha, ISBN, název, id_vydavatel, rok vydání, anotace)

FK: (id_vydavatel) \subseteq Vydavatel(id_vydavatel)

```
CREATE TABLE kniha (  
...  
    id_vydavatel      int4  
                        NOT NULL  
                        REFERENCES vydavatel ON UPDATE CASCADE ON DELETE NOACTION  
);
```

ON UPDATE CASCADE - při aktualizaci hodnoty primárního klíče referencovaného záznamu dojde ke změně hodnot cizích klíčů ve zdrojové tabulce

ON DELETE NOACTION - při mazání referencovaného záznamu dojde na aplikaci referenční integrity a mazání bude odmítnuto.

Další možnosti: SET NULL, SET DEFAULT, RESTRICT



Definice databáze

Zavedení klíčů - Slabý entitní typ

Slabý entitní typ z relačního modelu udává existenční závislost jednoho entitního typu na druhém.

Kniha (id_kniha, ISBN, název, id_vydavatel, rok_vydání, anotace)

Exemplář (id_kniha, číslo)

FK: (id_kniha) \subseteq Kniha(id_kniha)

```
CREATE TABLE exemplar (  
  id_kniha int4 NOT NULL  
  REFERENCES kniha ON UPDATE CASCADE ON DELETE CASCADE,  
  cislo smallint NOT NULL,  
  PRIMARY KEY (id_kniha, cislo)  
);
```

ON DELETE CASCADE - při smazání knihy musí dojít ke smazání všech exemplářů (jsou existenčně závislé na knize)

Tip: pro generování čísel rozlišující entity slabého entitního typu je vhodné použít funkce `generate_series(min, max)`



Definice databáze

Zavedení klíčů - Slabý entitní typ - Alternativa

Pokud to vyžadují jiné okolnosti a nevyžadují dotazy na tabulky referencující slabý entitní typ, je možné vytvořit umělý primární klíč i slabého entitního typu. Pak

Kniha(id_kniha, ISBN, název, id_vydavatel, rok vydání, anotace)

Exemplář(id_kniha, číslo)

FK: (id_kniha) \subseteq Kniha(id_kniha)

```
CREATE TABLE exemplar (  
  id_exemplar  serial,  
  id_kniha     int4          NOT NULL  
               REFERENCES kniha ON UPDATE CASCADE ON DELETE CASCADE,  
  cislo       smallint     NOT NULL,  
  UNIQUE (id_kniha, cislo),  
  PRIMARY KEY (id_exemplar)  
);
```

Definice databáze

Zavedení doménových omezení - CHECK

Pokud modelovaná realita vyžaduje omezení domény - omezte ji. Například kniha mohla být vydána pouze od roku 1445 dodnes, ISBN je omezeno na `xxx-xx-xxx-xxxx-x`.

```
CREATE TABLE kniha (  
...  
    ISBN          char(17)  
                NOT NULL  
                CHECK (isbn LIKE '____-__-____-____-__'),  
    rokvydani    smallint  
                NOT NULL  
                CHECK (rok BETWEEN 1445 AND EXTRACT(YEAR FROM CURRENT_DATE)),  
...  
);
```

Poznámka:

Integritní omezení se vyhodnocuje v okamžiku zápisu (viz temporální atributy).
Atributové integritní omezení se nevyhodnocuje, pokud je hodnota atributu NULL.

Definice databáze

Zavedení defaultní hodnoty - DEFAULT

Pokud není při vkládání dat uvedena hodnota atributu, je možné stanovit její defaultní hodnotu. Například aktuální rok u roku vydání knihy.

```
CREATE TABLE kniha (  
    ...  
    rokvydani smallint  
        NOT NULL  
        DEFAULT EXTRACT(YEAR FROM CURRENT_DATE)  
        CHECK (rok BETWEEN 1445 AND EXTRACT(YEAR FROM CURRENT_DATE)) ,  
    ...  
);
```

Definice databáze

Zavedení vazební tabulky - N:M

Při realizaci N:M vazby dochází k dekompozici na 2 vazby 1:N

Autor(id_autor, jmeno, prijmeni)

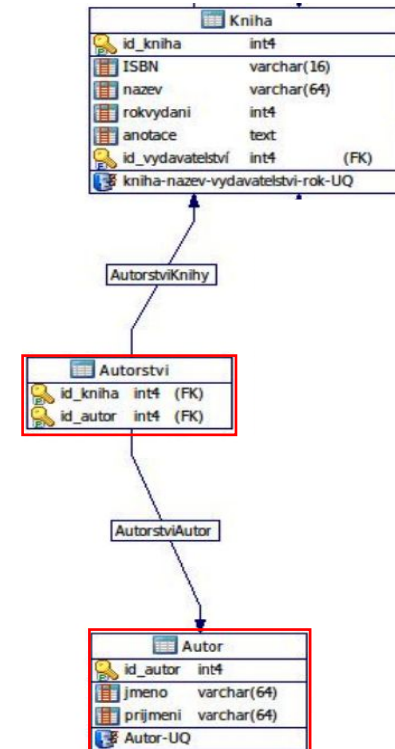
Autorstvi(id_kniha, id_autor)

FK: (id_kniha) \subseteq Kniha(id_kniha)

FK: (id_autor) \subseteq Autor(id_autor)

```
CREATE TABLE autor (
  id_autor      serial          PRIMARY KEY,
  jméno         varchar(64)     NOT NULL,
  prijmeni      varchar(64)     NOT NULL,
  UNIQUE (jmeno, prijmeni);
```

```
CREATE TABLE autorstvi (
  id_autor      int            NOT NULL REFERENCES autor
                                ON UPDATE CASCADE ON DELETE CASCADE,
  id_kniha      int            NOT NULL REFERENCES kniha
                                ON UPDATE CASCADE ON DELETE CASCADE,
  PRIMARY KEY (id_autor, id_kniha));
```





Definice databáze

Zavedení vazební tabulky - N:M

Kniha

id_kniha	ISBN	Název	Vydavatel	Rok vydání	Anotace
1	978-80-859-2617-0	To kalné ráno	Dita	1997	Černobílé fotografie...
2	978-80-754-6131-5	Válka s Mloky	Fortuna Libri	2017	
3	978-80-879-5041-8	Válka s Mloky	Československý spisovatel	1981	Toto dílo se pohybuje ...
4	978-80-7459-175-4	Stopy mých otců	Československý spisovatel	2018	Helena Šmahelová zpracovala...
5	978-80-7390-988-8	Rituál	Fobos	2020	
6	978-80-7303-433-7	Rituál	Domino		í potápěčka ...
7	978-80-253-4395-1	Rituál	Fragment		
8	978-80-7577-821-5	Rituál	Host		tajemný obřad...
9	978-16-8369-417-5	The Auctioneer	Quirk Pub		ng tensions...
10	978-80-277-1210-6	Dobří sousedé	Fobos		

Autor

id_autor	Jmeno	Prijmeni
1	Karel	Čapek
2	Jaroslav	Seifert
3	Helena	Šmahelová
4	Adam	Nevill
5	Mo	Hayder
6	Marina	Djačenko
7	Sergej	Djačenko
8	Lenka	Šíková
9	Jan Marek	Šík
10	Joan	Samson

id_kniha	id_autor
1	1
1	2
2	1
3	1
4	3
5	4
6	5
7	6
7	7
8	8
8	9
9	10
10	4

Autorstvi



- ***Pouze listové tabulky***

- Nevhodné, pokud se záznamy mohou překrývat - redundance/duplicita dat
- Výhoda - malý počet tabulek

- ***Sloučení do jedné tabulky***

- Nepoužité atributy nabývají hodnoty NULL
- Výhoda - jedna tabulka
- Nevýhoda:
 - možná větší paměťová náročnost při řídkém vyplnění
 - není možné referencovat dílčí tabulky

- ***Hierarchicky uspořádané tabulky s 1:1 vazbou***

- Výhoda: návrhově čisté řešení bez redundancí, paměťově optimální
- Nevýhoda: nutnost při dotazech tabulky spojovat



Definice databáze

Dědičnost - Hierarchie tabulek

Uživatel(id_uživatel, login, heslo)

Zaměstnanec(id_uživatel, rodné číslo, interní telefon, místnost, id_nadrizeny)

FK: (id_uživatel) \subseteq Uživatel(id_uživatel)

FK: (id_nadrizeny) \subseteq Zaměstnanec(id_uživatel)

Zákazník(id_uživatel, id_zaměstnanec, čas)

FK: (id_uživatel) \subseteq Uživatel(id_uživatel)

FK: (id_zamestnanec) \subseteq Zaměstnanec(id_uživatel)

```
CREATE TABLE uzivatel (  
  id_uzivatel      serial          PRIMARY KEY ,  
  login            varchar(64)     NOT NULL UNIQUE ,  
  heslo            bytea           NOT NULL);  
  
CREATE TABLE zamestnanec (  
  id_uzivatel      int             PRIMARY KEY  
                        REFERENCES uzivatel ON UPDATE CASCADE ON DELETE CASCADE ,  
  rodnecislo       char(11)        NOT NULL UNIQUE ,  
  internitelefon   varchar(16)    NOT NULL ,  
  mistnost         varchar(16)    NOT NULL ,  
  id_nadrizeny     int             REFERENCES zamestnanec );  
  
CREATE TABLE zakaznik (  
  id_uzivatel      int             PRIMARY KEY REFERENCES uzivatel ON UPDATE  
  CASCADE ON DELETE CASCADE ,  
  ...);
```

Definice databáze

Dědičnost - Pouze listové tabulky

Zaměstnanec(id_uživatel, login, heslo, rodné číslo, interní telefon, místnost, id_nadrizeny)

FK: (id_nadrizeny) \subseteq Zaměstnanec(id_uživatel)

Zákazník(id_uživatel, login, heslo, id_zaměstnanec, čas)

FK: (id_zamestnanec) \subseteq Zaměstnanec(id_uživatel)

```
CREATE TABLE zamestnanec (  
  id zamestnanec      serial          PRIMARY KEY  
  login               varchar(64)    NOT NULL UNIQUE,  
  heslo               bytea         NOT NULL  
  rodnecislo         char(11)      NOT NULL UNIQUE,  
  internitelefon     varchar(16)   NOT NULL,  
  mistnost           varchar(16)   NOT NULL,  
  id_nadrizeny       int           REFERENCES zamestnanec );  
  
CREATE TABLE zakaznik (  
  id zakaznik        serial          PRIMARY KEY,  
  login               varchar(64)    NOT NULL UNIQUE,  
  heslo               bytea         NOT NULL );  
...);
```


Definice databáze

Dědičnost - Pouze listové tabulky

Zaměstnanec(id_uživatel, login, heslo, rodné číslo, interní telefon, místnost, id_nadrizeny)

FK: (id_nadrizeny) \subseteq Zaměstnanec(id_uživatel)

Zákazník(id_uživatel, login, heslo, id_zaměstnanec, čas)

FK: (id_zamestnanec) \subseteq Zaměstnanec(id_uživatel)

```
CREATE TABLE zamestnanec (  
  id_zamestnanec  serial          PRIMARY KEY  
  login           varchar(64)    NOT NULL UNIQUE,  
  heslo          bytea          NOT NULL  
  rodnecislo     char(11)       NOT NULL UNIQUE,  
  internitelefon varchar(16)    NOT NULL,  
  mistnost       varchar(16)    NOT NULL,  
  id_nadrizeny   int            REFERENCES zamestnanec );  
  
CREATE TABLE zakaznik (  
  id_zakaznik    serial          PRIMARY KEY,  
  login          varchar(64)    NOT NULL UNIQUE,  
  heslo          bytea          NOT NULL);  
...);
```

POZOR
Zde nevhodné řešení -
redundantní login!!!



Definice databáze

Dědičnost - Spojení tabulek do jedné

Operace (id_kniha, číslo, čas, id_zaměstnanec, čas, id_zakazník, čas)

FK: (id_zakazník) \subseteq Zákazník(id_login)

FK: (číslo, id_kniha) \subseteq Exemplář(číslo, id_kniha)

FK: (id_zaměstnanec) \subseteq Zaměstnanec(login)

Výpůjčka (login, čas, do, cena / den, pokuta / den)

FK: (login, čas) \subseteq Operace(login, čas)

Vrácení (login, čas, stav)

FK: (login, čas) \subseteq Operace(login, čas)



Definice databáze

Dědičnost - Spojení tabulek do jedné

```
CREATE TABLE operace (  
  id_operace      serial          PRIMARY KEY,  
  id_kniha        int            NOT NULL,  
  cislo           int            NOT NULL,  
  vypujckacas    timestamp      NOT NULL DEFAULT now(),  
  id_zamestnanecvyp int          NOT NULL REFERENCES zamestnanec,  
  vypujckado     timestamp      NOT NULL,  
  cenaden        numeric (6,2)   NOT NULL DEFAULT 15,  
  pokutaden      numeric (6,2)   NOT NULL DEFAULT 50,  
  vracenicas     timestamp,  
  id_zamestnanecvrc int          REFERENCES zamestnanec,  
  stav           smallint,  
  UNIQUE (id_kniha, cislo, vypujckacas),  
  UNIQUE (id_zamestnanecvyp, vypujckacas),  
  UNIQUE (id_zamestnanecvrc, vracenicas),  
  FOREIGN KEY (id_kniha,cislo) REFERENCES kniha  
  (id_kniha,cislo));
```



Definice databáze

Vložení dat a kontrola integritních omezení

```
INSERT INTO kniha
  (ISBN, nazev, vydavatel, rok, anotace) VALUES
  ('978-80-859-2617-0', 'To kalné ráno',
   (SELECT id_vydavatel FROM vydavatel WHERE nazev = 'Dita'),
  1997, 'Černobílé fotografie..');
```

```
INSERT INTO kniha
  (ISBN, nazev, vydavatel, rok, anotace) VALUES
  ('978-80-754-6131-5', 'Válka s Mloky',
   (SELECT id_vydavatel FROM vydavatel WHERE nazev = 'Fortuna Libri'),
  2017, NULL);
```

```
INSERT INTO kniha
  (ISBN, nazev, vydavatel, rok, anotace) VALUES
  ('978-80-754-6131-5', 'Válka s Mloky',
   (SELECT id_vydavatel FROM vydavatel WHERE nazev='Fortuna Libri'),
  2017, NULL);
```

(Selže, protože kniha se stejným ISBN již existuje)



Definice databáze

Změna schématu

Přidejte do knihy nový klíčový atribut URL.

```
ALTER TABLE kniha ADD COLUMN url varchar(128);
```

```
ALTER TABLE kniha ALTER COLUMN url SET NOT NULL;
```

```
ALTER TABLE kniha  
    ADD CONSTRAINT kniha_url_UQ UNIQUE (url);
```



Definice databáze Změna schématu

Přidejte do knihy nový klíčový atribut URL.

```
ALTER TABLE kniha ADD COLUMN url varchar(128);
```

```
ALTER TABLE kniha ALTER COLUMN url SET NOT NULL;
```

(selže, protože existují záznamy,
které toto nové integritní omezení nesplňují).

```
UPDATE TABLE kniha  
  SET url='https://www.knihovna.cz/to-kalne-rano-osm-dni.html'  
  WHERE id_kniha = 1;
```

...

Doporučení:

Ověřte funkčnost databázového schématu na malém vzorku dat.