**Vysoká škola polytechnická Jihlava**

Aplikovaná informatika

**Implementace databázového systému**

Seminární práce

Autoři práce: Dvořáková J., Frydrýn Z., Komžáková O., Trégl T.

Předmět: Úvod do databázových systémů

Vyučující: doc. Ing. Zbyněk Bureš, Ph.D.

Jihlava 23. 04. 2024

Obsah

[1 Úvod a cíl práce 4](#__RefHeading___Toc804_1868499303)

[2 Primární analýza 4](#__RefHeading___Toc806_1868499303)

[3 Modelování dat 5](#__RefHeading___Toc808_1868499303)

[3.1 Konceptuální model 5](#__RefHeading___Toc810_1868499303)

[3.2 Transformace na relační model 7](#__RefHeading___Toc812_1868499303)

[4 Realizace v prostředí SQL 8](#__RefHeading___Toc814_1868499303)

[4.1 Návrh tabulek, včetně datových typů (CREATE TABLE) 8](#__RefHeading___Toc816_1868499303)

[4.2 Integritní omezení 11](#__RefHeading___Toc818_1868499303)

[4.3 Formulace dotazů v jazyce SQL 14](#__RefHeading___Toc820_1868499303)

[5 Obrázky a tabulky v dokumentu 14](#__RefHeading___Toc822_1868499303)

[5.1 Obrázky a grafy 14](#__RefHeading___Toc824_1868499303)

[5.2 Tabulky 15](#__RefHeading___Toc826_1868499303)

[Závěr 16](#__RefHeading___Toc828_1868499303)

[Společně… 16](#__RefHeading___Toc830_1868499303)

# Úvod a cíl práce

Účelem tohoto projektu je nejprve analyzovat data potřebná k provozu webu školícího centra, a pak navrhnout a implementovat databázový systém podporující jeho správu a poskytující data pro web. Východiskem byl stávající web <https://www.infracz.cz/vzdelavani>. Předmětem analýzy byly především úvahy, jak v databázi uchovávat informaci o nabídce centra (témata), o jeho lektorech, jak vypisovat termíny školení, jak doplnit model o online registraci zájemců (jak na vypsaná termíny kursů tak předregistraci na témata bez konkrétního termínu), zajištění evidence proběhlých školení včetně seznamu účastníků. Jednou z klíčových funkcionalit systému má být schopnost vystavit certifikáty účastníkům po úspěšném absolvování kurzu. Tímto způsobem se usnadní správa a provoz školícího centra, což povede k větší efektivitě a kvalitě poskytovaných služeb.

# Primární analýza

Prvotním náhledem na zadané prostředí byly určeny čtyři primární entity. ZAJEMCE a LEKTOR popisují osoby účastnící se projektu, SABLONA a KURS potom předmět školení a jeho konkrétní (termínovanou) instanci. Vztahy mezi entitami byly bouřlivě diskutovány a ustálily se v podobě prezentované konceptuálním grafem.

Pro zajištění souladu s „ideálním“ řešením se nabízí možnost logování historie těch úkonů, které se týkají účasti studenta na kursu; tato funkcionalita, řešitelná jednou přidanou relací typu (timestamp, odkazovaný\_vztah(student<->kurs), typ\_změny), však do primárního řešení nebyla zahrnuta, protože se předpokládá pevný počet kroků (registrace/realizace/absolvování) a ty lze plně postihnout atributy vztahu.

Na data pak budou kladeny například následující dotazy

1. Které kursy jsou aktuálně vypsány (a ještě se nerealizovaly)?
2. Kdo jsou účastníci té které konkrétní instance kursu? (parametr – ID kursu)

Var.1a) před kursem - kdo má zaplaceno?

Var.1b) po kursu – kdo úspěšně absolvoval?

1. Které ještě nevypsané kursy mají více než deset zájemců, kteří se ještě nezapsali na žádný konkrétní termín (a kursu je tedy potřeba vybrat lektora a vypsat termín)?
2. Který konkrétní ještě neuskutečněný kurs nemá ani pět zájemců (hrozí že bude zrušen)?
3. Který lektor má nejvyšší účast/úspěšnost? (resp. seřaď lektory dle účasti/úspěšnosti studentů)?

# Modelování dat

## Konceptuální model

Vzhledem k diskrepanci mezi teorií relačních databází (kde jsou všechny atributy napříč celou relační strukturou identifikovány unikátně (pokud neslouží ke spojování) a primárně se pro dotazy používá přirozené spojení na jedné straně), a praxí tvorby SQL databází (kde jsou z praktických důvodů atributy identifikovány unikátně pouze v rámci jedné entity a to identifikátorem bez diakritiky, pro jejich globální identifikaci se používají prefixy subrelací v „dot“ režimu, a ke spojování se používají řízená spojení s uvedením spojovacích atributů) na straně druhé, bylo nutno přijmout duplicitu pojmenování relací a atributů. V konceptuálním modelu se přidržíme teoretického přístupu, při transformaci relačního modelu na SQL předefinujeme všechny identifikátory dle potřeb praxe.

### Identifikace entitních typů a jejich klíčů:

ZÁJEMCE – Č\_Z (číslo zájemce je vhodný kandidát na klíč, je unikátní)

KURZ – Katalog\_č (katalogové číslo je vhodný kandidát na klíč, je unikátní)

LEKTOR – **Č\_L** (číslo lektora je vhodný kandidát na klíč, je unikátní)

ŠABLONA\_K – Akreditační\_č (akreditační číslo je vhodný kandidát na klíč, je unikátní)

### Identifikace vztahů:

„je\_školen“ mezi entitami KURZ a ZÁJEMCE. - Tento vztah popisuje celou genezi vztahu mezi entitami, od registrace, přes platbu, účast až po hodnocení kursu

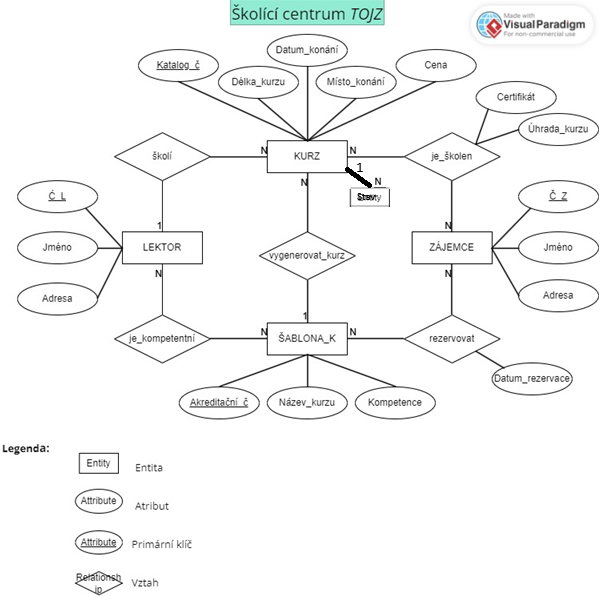
„školí“ mezi entitami LEKTOR a KURZ. Jednoznačná identifikace vedoucího kurzu.

„je\_kompetentní“ mezi entitami LEKTOR a ŠABLONA\_K – Způsobilost vést kurs daného typu

„rezervovat“ mezi entitami ŠABLONA\_K a ZÁJEMCE – Předběžné vyjádření zájmu o daný kurs, bez znalosti konkrétního data a lektora.

„vygenerovat\_kurz“ mezi entitami ŠABLONA \_K a KURZ – vypsání konkrétního termínu

### Vizualizace



### Identifikace integritních omezení:

Vztah „je\_školen“ je typu M:N, jelikož se každý zájemce může (a nemusí) zúčastnit více kurzů a každý kurz může (a nemusí) studovat více zájemců.

Vztah „školí“ je typu 1:N, jeho determinantem je entitní typ KURZ, neboť u kurzu lze jednoznačně říct, kdo ho vede.

Vztah „je\_kompetentní“ je typu M:N, jelikož lektor může (a nemusí) být kompetentní k více šablonám a šablona může (a nemusí) mít více kompetentních lektorů.

Vztah „rezervovat“ je typu M:N, jelikož šablona kurzu může mít (a nemusí) více zájemců o rezervaci a jeden zájemce může (a nemusí) mít více rezervovaných šablon.

Vztah „vygenerovat\_kurz“ je typu 1:N, jeho determinantem je entitní typ KURZ, neboť u kurzu lze jednoznačně říct, ze které šablony vychází.

## Transformace na relační model

### Standardní relační transformace

KURZ(Katalog\_č, Délka\_kurzu, Datum\_konání, Místo\_konání, Cena, *Akreditační\_č, Č\_L, Stav\_kurzu*)

KURZ\_Stav (Stav\_kurzu, Stav\_kurzu\_popisek)

ZÁJEMCE(Č\_Z, Jméno, Adresa)

ŠABLONA\_k(Akreditační\_č, Název\_kurzu, Kompetence)

LEKTOR(Č\_L, Jméno, Adresa)

je\_kompetentní(*Č\_L*, *Akreditační\_č*)

rezervace*(Č\_Z*, *Akreditační\_č*, Datum\_rezervace)

je\_školen(*Katalog\_č, Č\_Z*, Certifikát, Úhrada\_kurzu)

### Použité postupy a integritní omezení

Ze vztahů M:N vznikly tři nové relace: ucast (KURZ vs ZÁJEMCE), rezervace (ZÁJEMCE vs ŠABLONA) a kompetence (ŠABLONA vs LEKTOR). Integritní omezení jsou částečně pokryta relačním modelem (zejména kardinalita), k dalšímu ošetření zbylo automatické vyplnění všech umělých klíčů, zajištění unikátnosti alternativních klíčových množin (jméno v entitách ZÁJEMCE a LEKTOR) a povinné vyplnění všech atributů, s výjimkou několika málo neklíčových: rezervace (datum\_rezervace) a ucast(Certifikát, Úhrada kurzu)

### Formalizace dotazů v relačním kalkulu

….(TODO)

### Transformace na SQL atributy (v [] jsou identifikační aliasy relací):

[K]-KURZ (K.ID\_Katalog, K.Delka, K.Datum, K.Misto, K.Cena, *S.ID\_Akreditace, L.ID\_Lektor, KS.ID*)   
[KS]-CC\_KURZ\_STAV(KS.ID, KS.descr)

[Z]-ZAJEMCE(Z.ID\_Zajemce, Z.Jmeno, Z.Prijmeni, Z.Adresa)

[S]-SABLONA (S.ID\_Akreditace, S.Nazev, S.Kompetence)

[L]-LEKTOR(L.ID\_Lektor, L.Jmeno, L.Prijmeni, L.Adresa)

[P]-KOMPETENCE(*L.ID\_Lektor*, *S.ID\_Akreditace*)

[R]-REZERVACE*(Z.ID\_Zajemce*, *S.ID\_Akreditace*, R.Datum)

[U]-UCAST(*K.ID\_Katalog,* *Z.ID\_Zajemce*, U.Certifikat, U.Uhrada)

# Realizace v prostředí SQL

## Návrh tabulek, včetně datových typů (CREATE TABLE)

**Tvorba tabulek**

CREATE TABLE lektor (

id\_lektor INT GENERATED ALWAYS AS IDENTITY,

jmeno VARCHAR(255) NOT NULL,

prijmeni VARCHAR(255) NOT NULL,

adresa VARCHAR(255) NOT NULL,

PRIMARY KEY(id\_lektor)

);

CREATE TABLE sablona (

id\_sablona INT GENERATED ALWAYS AS IDENTITY,

nazev VARCHAR(255) NOT NULL,

PRIMARY KEY(id\_sablona)

);

CREATE TABLE kompetence (

lektor\_id INT NOT NULL,

sablona\_id INT NOT NULL,

CONSTRAINT fk\_lektor

Foreign Key (lektor\_id) REFERENCES lektor(id\_lektor) ON DELETE CASCADE,

CONSTRAINT fk\_akreditace

Foreign Key (sablona\_id) REFERENCES sablona(id\_sablona) ON DELETE CASCADE,

CONSTRAINT uq\_lektor\_sablona UNIQUE (lektor\_id, sablona\_id)

);

CREATE Table stav\_k (

id\_stav INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

popis VARCHAR(255) NOT NULL

);

CREATE TABLE kurz (

id\_kurz INT GENERATED ALWAYS AS IDENTITY,

delka INT NOT NULL,

datum DATE NOT NULL,

misto VARCHAR(255) NOT NULL,

cena DECIMAL(8, 2) NOT NULL,

sablona\_id INT NOT NULL DEFAULT 1,

lektor\_id INT NOT NULL DEFAULT 1,

stav\_id INT NOT NULL DEFAULT 1,

PRIMARY KEY(id\_kurz),

CONSTRAINT fk\_akreditace

FOREIGN KEY (sablona\_id)

REFERENCES sablona(id\_sablona)

ON DELETE SET DEFAULT, -- akreditace id = 1 bude mit hodnotu "ceka na vytvoreni"

CONSTRAINT fk\_lektor

FOREIGN KEY (lektor\_id)

REFERENCES lektor(id\_lektor)

ON DELETE SET DEFAULT, -- lektor id = 1 bude mit hodnotu "suplujici"

CONSTRAINT fk\_stav

FOREIGN KEY (stav\_id)

REFERENCES stav\_k(id\_stav)

ON DELETE RESTRICT,

CONSTRAINT ch\_kompetence

CHECK ( fn\_check\_kompetence (lektor\_id, sablona\_id) ),

CONSTRAINT ch\_date

CHECK (datum > CURRENT\_DATE)

);

CREATE TABLE zajemce (

id\_zajemce INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

jmeno VARCHAR(255) NOT NULL,

prijmeni VARCHAR(255) NOT NULL,

adresa VARCHAR(255) NOT NULL,

CONSTRAINT uq\_customer

UNIQUE (jmeno, prijmeni, adresa)

);

CREATE Table rezervace (

zajemce\_id INT NOT NULL,

sablona\_id INT NOT NULL,

datum DATE DEFAULT CURRENT\_DATE,

CONSTRAINT fk\_zajemce

Foreign Key (zajemce\_id)

REFERENCES zajemce(id\_zajemce)

ON DELETE CASCADE,

CONSTRAINT fk\_sablona

Foreign Key (sablona\_id)

REFERENCES sablona(id\_sablona)

ON DELETE CASCADE,

CONSTRAINT uq\_reservation

UNIQUE (zajemce\_id, sablona\_id, datum)

);

CREATE Table ucast (

kurz\_id INT NOT NULL,

zajemce\_id INT NOT NULL,

certifikat VARCHAR(255) NOT NULL,

uhrada DATE NOT NULL, -- constraint nez zacne kurz

CONSTRAINT fk\_kurz

Foreign Key (kurz\_id)

REFERENCES kurz(id\_kurz)

ON DELETE RESTRICT,

CONSTRAINT fk\_zajemce

Foreign Key (zajemce\_id)

REFERENCES zajemce(id\_zajemce)

ON DELETE CASCADE,

CONSTRAINT uq\_zajemce\_kurz

UNIQUE (zajemce\_id, kurz\_id),

CONSTRAINT ch\_paid

CHECK ( uhrada < fn\_kurz\_started(kurz\_id))

);

**Funkce**

**Kontrola kompetence lektora učit zvolený kurz**

CREATE OR REPLACE FUNCTION fn\_check\_kompetence(fn\_lektor\_id INT, fn\_sablona\_id INT)

RETURNS BOOL

AS

$$

BEGIN

RETURN EXISTS (

SELECT 1 FROM kompetence

JOIN lektor ON fn\_lektor\_id = lektor.id\_lektor

JOIN sablona ON fn\_sablona\_id = sablona.id\_sablona

WHERE fn\_lektor\_id = lektor.id\_lektor

AND fn\_sablona\_id = sablona.id\_sablona

);

END;

$$

LANGUAGE PLPGSQL;

**Kontrola při tvorbě účasti na kurzu. Zajemce je přidán, pokud má zaplaceno před začátkem kurzu.**

CREATE OR REPLACE FUNCTION fn\_kurz\_started(fn\_kurz\_id INT)

RETURNS DATE

AS

$$

DECLARE

kurz\_date DATE;

BEGIN

SELECT datum INTO kurz\_date FROM kurz

WHERE fn\_kurz\_id = id\_kurz;

RETURN kurz\_date;

END

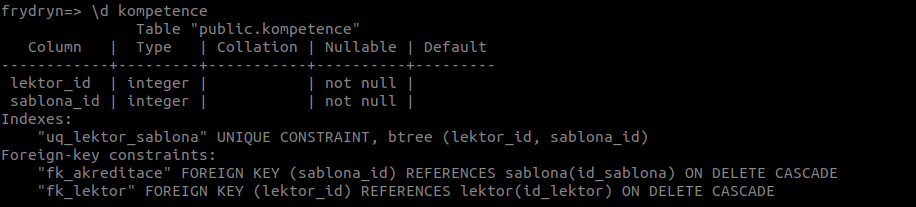
$$

LANGUAGE PLPGSQL;

Popis a zdůvodnění principu zvoleného řešení, popis integritních omezení + formulace všech požadovaných dotazů v SQL: vložit vždy i snímek obrazovky (terminálu), z něhož bude patrné správné fungování dotazu. …(TODO)

## Integritní omezení

### Kompetence



**Obr. 1: Integritní omezení entity `kompetence`**

**Unikátnost**

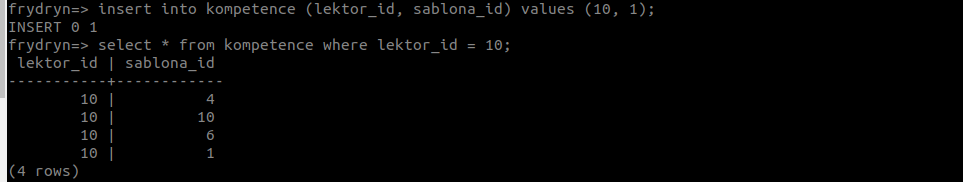
Určuje, který lektor je kompetentní k učení kurzů. Kombinace sloupců **lektor\_id** a **sablona\_id** tvoří kompozitní klíč, jednoznačně určující řádek tabulky. Klíčové slovo UNIQUE zajišťuje unikátní kombinaci.

**Cizí klíče**

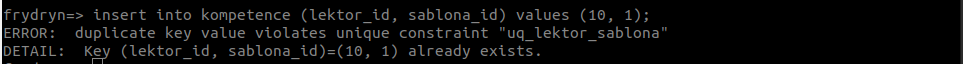
**fk\_akreditace** a **fk\_lektor** odkazují na primární klíče v tabulkách **lektor** a **sablona**.



**Obr 2: Dotaz nelze vykonat. Neexistuje záznam v tabulce lektor s id\_lektor = 100.**



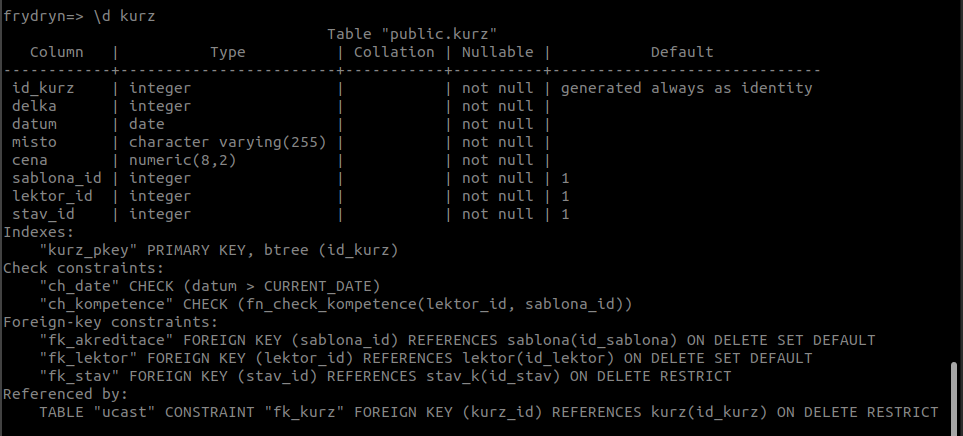
**Obr 3: Dotaz `Insert into kompetence (lektor\_id, sablona\_id) values (10, 1)` je úspěšně proveden.**



**Obr 4: Demostrace unikátnosti. Byl použit stejný dotaz, jako v předchozím případě.**

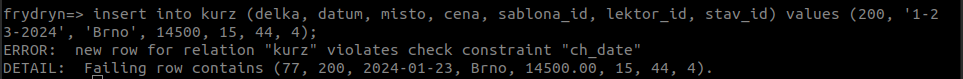
### Kurz

**ch\_date**



**Obr 5: Integritní omezení entity `kurz`**

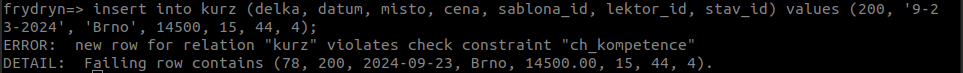
Nelze vytvořit kurz s datem v minulosti.



**Obr. 6: Chybně zadaný dotaz s datem starším než dnešním**

**ch\_kompetence**

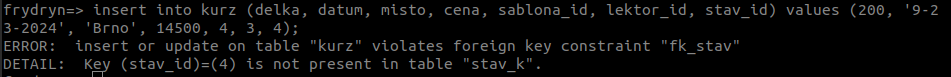
Kontrola kempetence lektora učit daný kurz. Funkce **fn\_check\_kompetence(lektor\_id, sablona\_id)** [10]vyhledá v databázi lektora a jeho kompetence, porovná vyhledané hodnoty s hodnotami vkládanými do tabulky `kurz`.



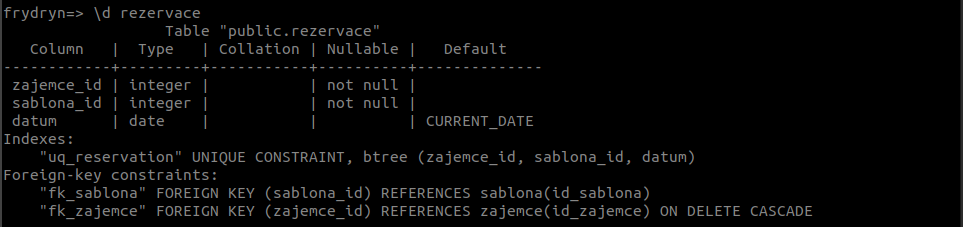
**Obr. 7: Lektor s id 44 nemá kompetenci učit kurz s id 15**

**Cizí klíče**

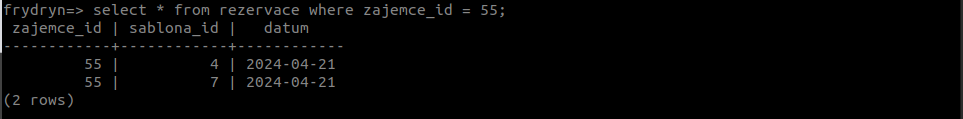
**fk\_akreditace –** odkazuje na primární klíč šablony  
**fk\_lektor** - odkazuje na primární klíč lektora  
**fk\_stav** - odkazuje na primární klíč stavu kurzu

**Obr. 8: V dotazu je chybně odkazováno na stav kurzu s id 4**

### Rezervace

**Obr. 9: Integritní omezení entity `registrace`**

**Unikátnost**



**Obr. 10: Kompozitní klíč je tvořen unikátní kombinací sloupců zajemce\_id, sablona\_id a datum.**



**Obr. 11: Zájemce nemůže stejný kurz ve stejný den rezervovat vícekrát.**

**Cizí klíče**

fk\_sablona odkazuje na primární klíč šablony. fk\_zajemce odkazuje na primární klíč zájemce.

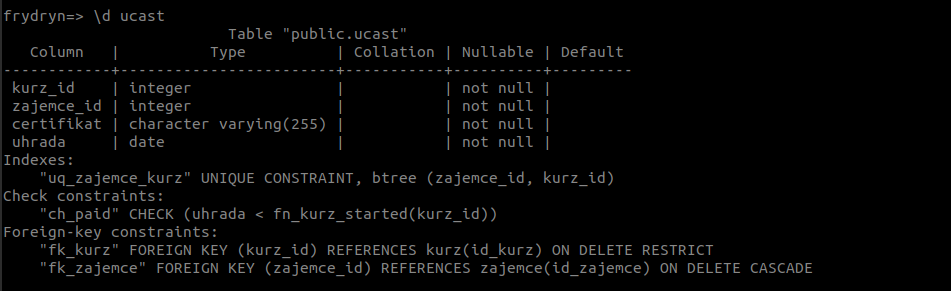


**Obr. 12: Zájemce s id 999 neexistuje**

### Účast

Vazební tabulka mezi entitami kurz a zájemce. Lze přidat pouze zájemce, kteří existují v tabulce **`zajemce`** a lze přidat pouze údaj o kurzu s existujícím id v tabulce **`kurz`**

**Unikátnost**Kombinace cizích klíčů tvoří kompozitní klíč, jednoznačně definující řádek tabulky. Tato kombinace musí být unikátní.



**Obr. 13: Integritní omezení vazební entity `ucast`**

**ch\_paid**Vytvoření záznamu o účasti na kurzu je podmíněna platbou předem. Zajištěno funkcí **fn\_kurz\_started**[10], která vrátí datum zahájení kurzu, který je porovnán s datem úhrady. Datum úhrady musí být dřívějšího data, než je datum zahájení.

**Cizí klíče**

**fk\_kurz** odkazuje na primární klíč v tabulce **`kurz`  
fk\_zajemce** odkazuje na primární klíč v tabulce **`zajemce`**

## Formulace dotazů v jazyce SQL

…(TODO)

# Obrázky a tabulky v dokumentu

## Obrázky a grafy

Obr. 1: ER model školícího centra

Zdroj: vlastní práce

## Tabulky

Číslo a název tabulky se uvádějí nad tabulkou (styl Titulek). Zdroje dat, které se v tabulce vyskytují, se uvádějí pod tabulkou (styl Zdroj). Tabulky jsou centrované (zarovnané na střed). Ohraničení tabulky je ponecháno na volbě autora. Pro text v tabulce je použitý styl Tabulka.

Tab. 1: Název (popis) tabulky

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Styl*** | ***Písmo*** | ***Velikost*** | ***Tučné*** | ***Zarovnání*** | ***Mezera před*** | ***Mezera za*** |
| **Nadpis 1** | Calibri Light | 18 | Ano | Na střed | 18 | 6 |
| **Nadpis 2** | Calibri Light | 15 | Ano | Vlevo | 18 | 6 |

Zdroj: vlastní zpracování dle Borůvková a kol.(2021)

# Závěr

Kapitola závěr obsahuje shrnutí zjištěných poznatků – odpověď na položenou hlavní výzkumnou otázku, zhodnocení stanovených hypotéz, přínos autora k řešení dané problematiky = co je v práci původního, zhodnocení využitelnosti dosažených výsledků, nastínění dalších směrů bádání apod.

# Společně…