**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 28. 04. 2024

Obsah

[Úvod a cíl práce 3](#_Toc165016214)

[1 Primární analýza 3](#_Toc165016215)

[2 Modelování dat 4](#_Toc165016216)

[2.1 Konceptuální model 4](#_Toc165016217)

[2.2 Transformace na relační model 6](#_Toc165016218)

[2.3 Formalizace dotazů v relačním kalkulu 6](#_Toc165016219)

[3 Realizace v prostředí SQL 7](#_Toc165016220)

[3.1 Návrh tabulek, včetně datových typů (CREATE TABLE) 7](#_Toc165016221)

[3.2 Integritní omezení 8](#_Toc165016222)

[3.3 Formulace dotazů v jazyce SQL 11](#_Toc165016223)

[Závěr 12](#_Toc165016224)

[Poděkování 12](#_Toc165016225)

Seznam obrázků

[Obr. 1: ER model školícího centra 6](#_Toc165013523)

[Obr. 2: Integritní omezení entity `kompetence` 9](#_Toc165013524)

[Obr. 3: Neúspěšné vkládání do `kompetence` - neexistující záznam v tabulce `lektor` 9](#_Toc165013525)

[Obr. 4 Úspěšné vložení unikátních dat do `kompetence` 9](#_Toc165013526)

[Obr. 5: Nespěšné vložení duplicitních dat do `kompetence` 9](#_Toc165013527)

[Obr. 6 Integritní omezení entity `kurz` 10](#_Toc165013528)

[Obr. 7: Neúspěšné vkládání do `kurz` - Nelze vytvořit kurz s datem v minulosti. 10](#_Toc165013529)

[Obr. 8: Neúspěšné vkládání do `kurz` - Lektor s id 44 nemá kompetenci 10](#_Toc165013530)

[Obr. 9: Neúspěšné vkládání do `kurz` - chybný stav kurzu s id 4 10](#_Toc165013531)

[Obr. 10: Integritní omezení entity `registrace` 11](#_Toc165013532)

[Obr. 11: Neúspěšné vkládání do `rezervace` - Zájemce nemůže stejný kurz rezervovat vícekrát. 11](#_Toc165013533)

[Obr. 12: Neúspěšné vkládání do `rezervace` -Zájemce neexistuje 11](#_Toc165013534)

[Obr. 13: Integritní omezení vazební entity `ucast` 11](#_Toc165013535)

# 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 vypsané termíny kurzů, 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 KURZ 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 kurzu; tato funkcionalita, řešitelná jednou přidanou relací typu (timestamp, odkazovaný\_vztah(student <-> kurz), 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 kurzy jsou aktuálně vypsány (a ještě se nerealizovaly)?
2. Kdo jsou účastníci té které konkrétní instance kurzu? (parametr – ID kurzu)

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

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

1. Které ještě nevypsané kurzy mají více než deset zájemců, kteří se ještě nezapsali na žádný konkrétní termín (a kurzu je tedy potřeba vybrat lektora a vypsat termín)?
2. Který konkrétní ještě neuskutečněný kurz 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)

KURZ\_Stav – Stav\_kurzu (přidělené unikátní identifikační číslo v číselníku, zamýšlený klíč)

### 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 kurzu

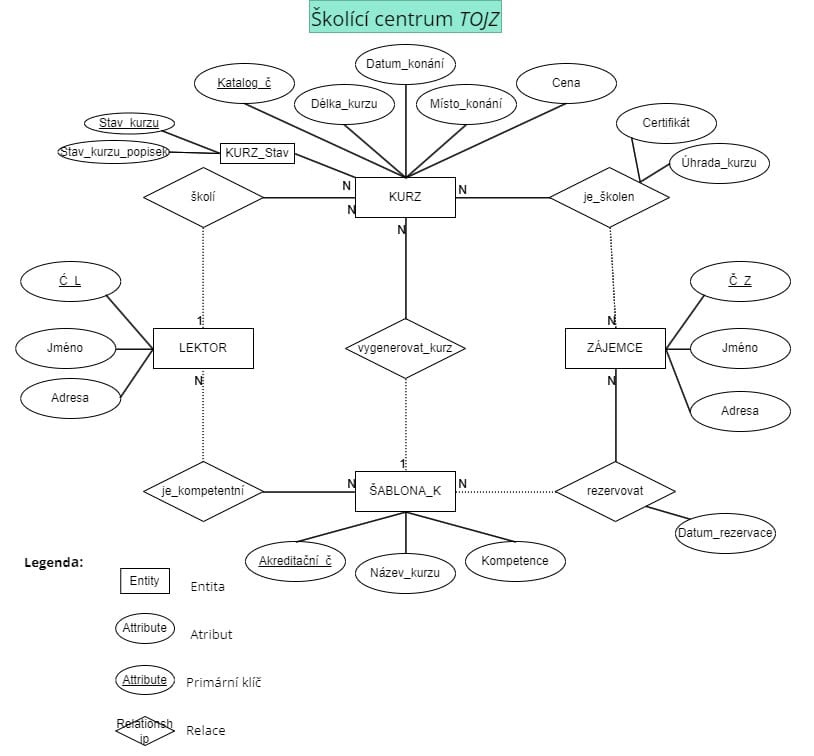
„školí“ mezi entitami LEKTOR a KURZ. Jednoznačná identifikace vedoucího kurzu.

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

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

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

### Vizualizace



Obr. 1: ER model školícího centra

(zdroj: vlastní práce „O“)

### 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: je\_školen (KURZ vs ZÁJEMCE), rezervace (ZÁJEMCE vs ŠABLONA) a je\_kompetentní (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 je\_školen (Certifikát, Úhrada kurzu)

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

[K]-KURZ (K.ID\_Kurs, K.Delka, K.Datum, K.Misto, K.Cena, *S.ID\_Sablona, L.ID\_Lektor, KS.ID*)   
[KS]- STAV\_K(KS.ID\_Stav, KS.popis)

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

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

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

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

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

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

## Formalizace dotazů v relačním kalkulu

Dotaz č. 1:

SABLONA\*(KURZ(Stav\_kurzu in (KURZ\_Stav(Stav\_kurzu\_popisek='Připravováno')[Stav\_kurzu]

Dotaz č. 2:

Var a) (ZAJEMCE\*UCAST) (kurz\_id=1 and uhrada !=0) [jmeno,prijmeni]

Var b) (ZAJEMCE\*UCAST) (kurz\_id=1 and certifikat !=0) [jmeno,prijmeni]

Dotaz č. 3:

….(TODO)

Dotaz č. 4:  
SABLONA\*(KURZ(Stav\_kurzu in (KURZ\_Stav(Stav\_kurzu\_popisek='Připravováno')[Stav\_kurzu])) - KURZ<\*((Katalog\_č **Γ** count(Č\_Z) (je\_školen))(count(Č\_Z)>=5)))[Katalog\_č,Datum\_konání, Akreditační\_č, Název\_kurzu]

Dotaz č. 5:  
LEKTOR \* KURZ \* UCAST

# Realizace v prostředí SQL

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

### Tvorba tabulek

Všechny tabulky jsou tvořeny podle jednotné sytaxe:

:- CREATE TABLE table*name* (  
*POLOZKA* [GENERATED ALWAYS AS IDENTITY]  
[,*POLOZKA….*],  
PRIMARY KEY (*keyname*));

*POLOZKA :- itemname itemtype* [[NOT] NULL] [DEFAULT *value*]

Takto byly vytvořeny tabulky: lektor, sablona, kurz, kompetence, stav\_k, zajemce, rezervace, ucast

### Zavedení omezení a indexů

Dále jsou na každou vytvořenou tabulku doplněna integritní omezení, zejména cizí klíče a unikátní omezení, s užitím syntaxe:

:- ALTER TABLE table*name OMEZENI* [, *OMEZENI*…];

OMEZENI :- ADD CONSTRAINT *name* FOREIGN KEY (*own\_key*) REFERENCES *foreign\_table*(*foreign\_key*)

případně i OMEZENI :- ADD CONSTRAINT *name* UNIQUE (*key*[,*key*])

nebo i OMEZENI :- ADD CONSTRAINT name CHECK (*PODMINKA*)

a pokud to bylo nutné, doplněny indexy:

:- CREATE INDEX *indexname* ON *tablename*(*key*[,*key*]));

### Definice integritních omezení funkcí

Dále byly vytvořeny integritní funkce:

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

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

### Zadání vzorových dat

Tabulky byly následně naplněny daty podle vzoru:

:- INSERT INTO *tablename* (*key*[,*key*]) VALUES (*value*[,*value*]);

Celý SQL code je pro větší přehlednost a snazší implementaci (a kontrolu) uveden jako Příloha č.1

## Integritní omezení

### Kompetence

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obr. 2: Integritní omezení entity `kompetence`

(zdroj: vlastní práce „Z“)

**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. 3: Neúspěšné vkládání do `kompetence` - neexistující záznam v tabulce `lektor`

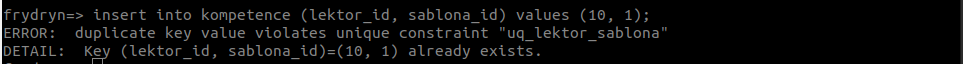
(zdroj: vlastní práce „Z“)

Obsah obrázku text, snímek obrazovky

Popis byl vytvořen automaticky

Obr. 4 Úspěšné vložení unikátních dat do `kompetence`

(zdroj: vlastní práce „Z“)



Obr. 5: Nespěšné vložení duplicitních dat do `kompetence`

(zdroj: vlastní práce „Z“)

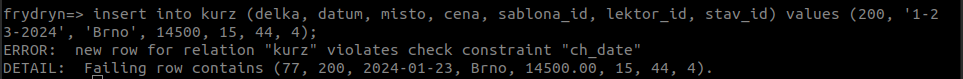
### Kurz

Obsah obrázku text, snímek obrazovky

Popis byl vytvořen automaticky

Obr. 6 Integritní omezení entity `kurz`

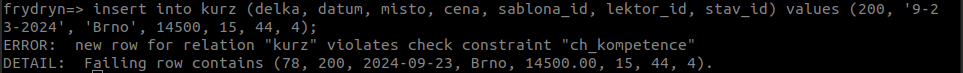
(zdroj: vlastní práce „Z“)



Obr. 7: Neúspěšné vkládání do `kurz` - Nelze vytvořit kurz s datem v minulosti.

(zdroj: vlastní práce „Z“)

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

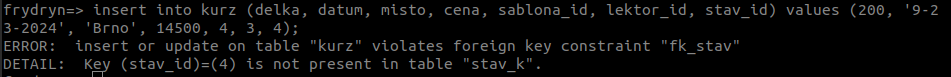


Obr. 8: Neúspěšné vkládání do `kurz` - Lektor s id 44 nemá kompetenci

(zdroj: vlastní práce „Z“)

**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. 9: Neúspěšné vkládání do `kurz` - chybný stav kurzu s id 4

(zdroj: vlastní práce „Z“)

### Rezervace

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obr. 10: Integritní omezení entity `registrace`

(zdroj: vlastní práce „Z“)

Unikátnost:



Obr. 11: Neúspěšné vkládání do `rezervace` - Zájemce nemůže stejný kurz rezervovat vícekrát.

(zdroj: vlastní práce „Z“)

**Cizí klíče**

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



Obr. 12: Neúspěšné vkládání do `rezervace` -Zájemce neexistuje

(zdroj: vlastní práce „Z“)

### 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`**

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automaticky

Obr. 13: Integritní omezení vazební entity `ucast`

(zdroj: vlastní práce „Z“)

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

**ch\_paid**Vytvoření záznamu o účasti na kurzu je podmíněna platbou předem. Zajištěno funkcí **fn\_kurz\_started**, 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

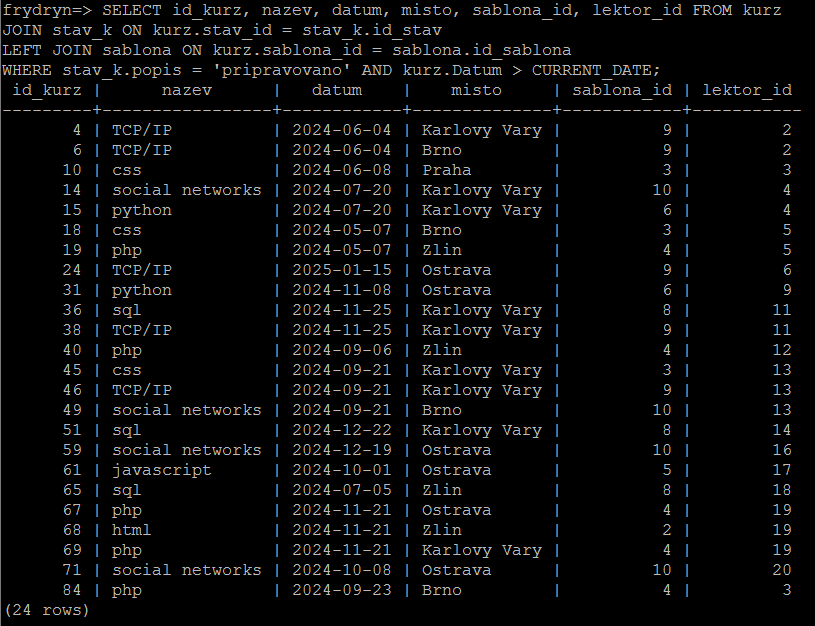
Dotaz č.1:

SELECT id\_kurz, nazev, datum, misto, sablona\_id, lektor\_id FROM kurz

JOIN stav\_k ON kurz.stav\_id = stav\_k.id\_stav

LEFT JOIN sablona ON kurz.sablona\_id = sablona.id\_sablona

WHERE stav\_k.popis = 'pripravovano' AND kurz.Datum > CURRENT\_DATE;



Obr. 14 Výsledek dotazu č.1

(zdroj: vlastní práce „O“)

Dotaz č.2 var a) (parametr se předává nastavením proměnné „\set P1=‘ID\_cílového kursu;‘“ před vlastní dotaz)

SELECT jmeno, prijmeni FROM zajemce

JOIN ucast ON (zajemce.id\_zajemce=ucast.zajemce\_id)

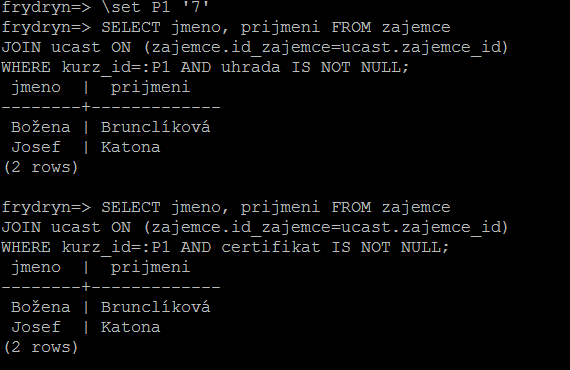
WHERE kurz\_id=:P1 AND uhrada IS NOT NULL

Dotaz č.2 var b)

SELECT jmeno, prijmeni FROM zajemce

JOIN ucast ON (zajemce.id\_zajemce=ucast.zajemce\_ID)

WHERE kurz\_id=:P1 AND certifikat IS NOT NULL



Obr. 15: Výsledek dotazu 2

(zdroj: vlastní práce „J“)

Dotaz č.3:  
…(TODO)

Dotaz č.4:  
SELECT \* FROM (

SELECT K.ID\_KURZ,K.Datum, K.MISTO, S.Nazev,

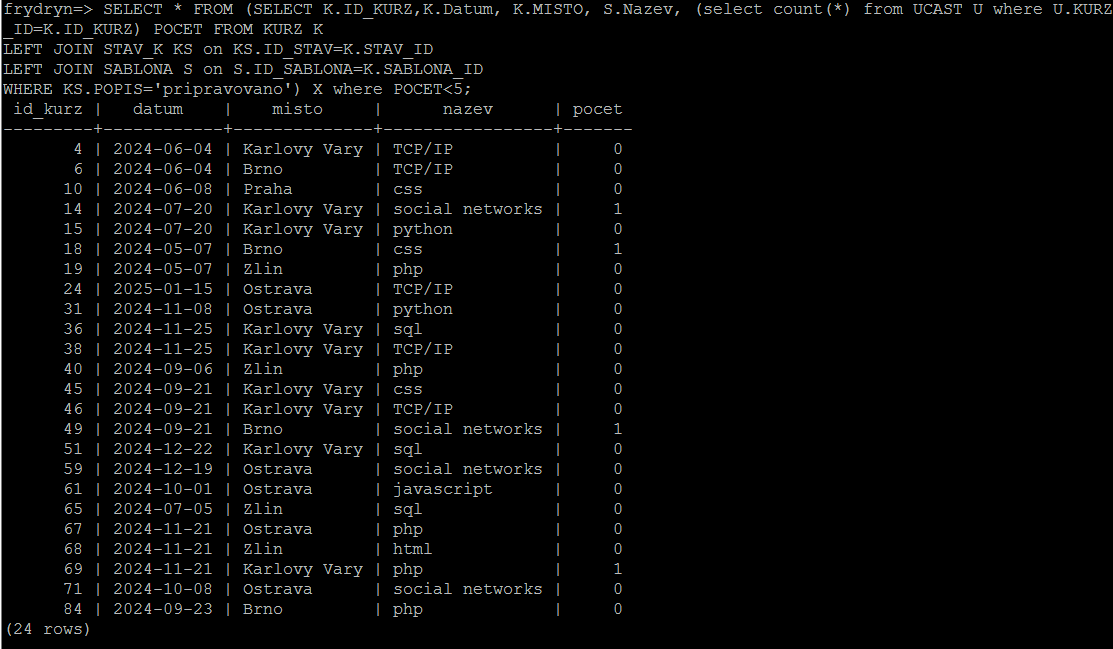
(select count(\*) from UCAST U where U.KURZ\_ID=K.ID\_KURZ) POCET FROM KURZ K

LEFT JOIN STAV\_K KS on KS.ID\_STAV=K.STAV\_ID

LEFT JOIN SABLONA S on S.ID\_SABLONA=K.SABLONA\_ID

WHERE KS.POPIS='pripravovano') X

where POCET<5;



Obr. 16 Výsledek dotazu 4

(zdroj: vlastní práce „T“)

Dotaz č.5

select lektor.jmeno, lektor.prijmeni, count(\*)

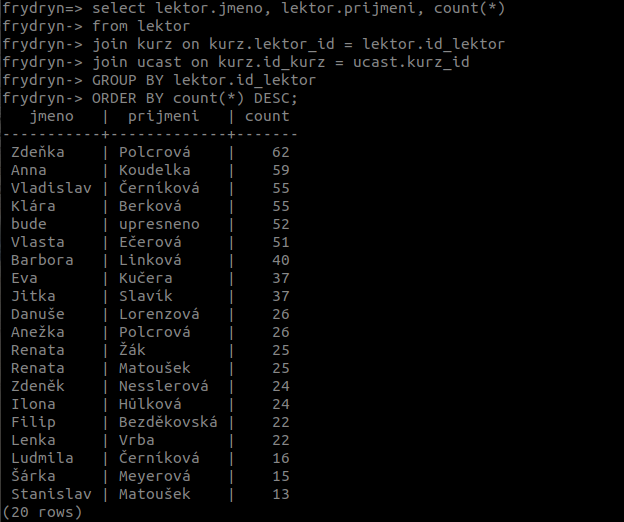
from lektor

join kurz on kurz.lektor\_id = lektor.id\_lektor

join ucast on kurz.id\_kurz = ucast.kurz\_id

GROUP BY lektor.id\_lektor

ORDER BY count(\*) DESC;

Obr. 17 Výsledek dotazu 5

(zdroj: vlastní práce „Z“)

# Závěr

Společným kolektivním úsilím všech členů teamu se podařilo namodelovat databázi dle zadání a ověřit v ní funkčnost dotazů. Domnívám se, že výsledný projekt by opravdu mohl sloužit za základ reálné implementace databáze pro zamýšlený účel, tedy jako informační systém školícího centra.

Přesto je třeba zmínit, že se v teamu vyskytly (všeobecně sdílené) pochybnosti, zda by praktický projekt využil všechny kroky, které byly dle zadání podniknuty; zejména byla (vzhledem k praxi) už na takto omezeném projektu diskutována vhodnost užití Chenovy notace, vhodnost vynucené unikátní identifikace atributů napříč celou databází, komplikace spojené s užitím přirozeného spojení a také možnost sloučení tvorby konceptuálního a relačního návrhu. Nicméně by bylo nespravedlivé neocenit přínos tohoto teoretického projektu k hlubšímu pochopení problematiky databází.

# Poděkování

Dovolte mi na tomto místě jako koordinátorovi projektu vyjádřit až nadšení nad zájmem všech členů o projekt jakož i radost ze společných konzultací, a ocenit efektivní podíl každého jednotlivě na celkovém výsledku. Olze dík za konkretizaci zadání a koncept, Janě za relační modelování, Zdendovi za návrh, četné revize a testování SQL a každému za zpracování vlastního relačního i databázového dotazu. Tomáš.

…(TODO) Literatura a citace. Kurňa mně se do toho nechce. Nechcete někdo poděkování navíc? :-D