

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

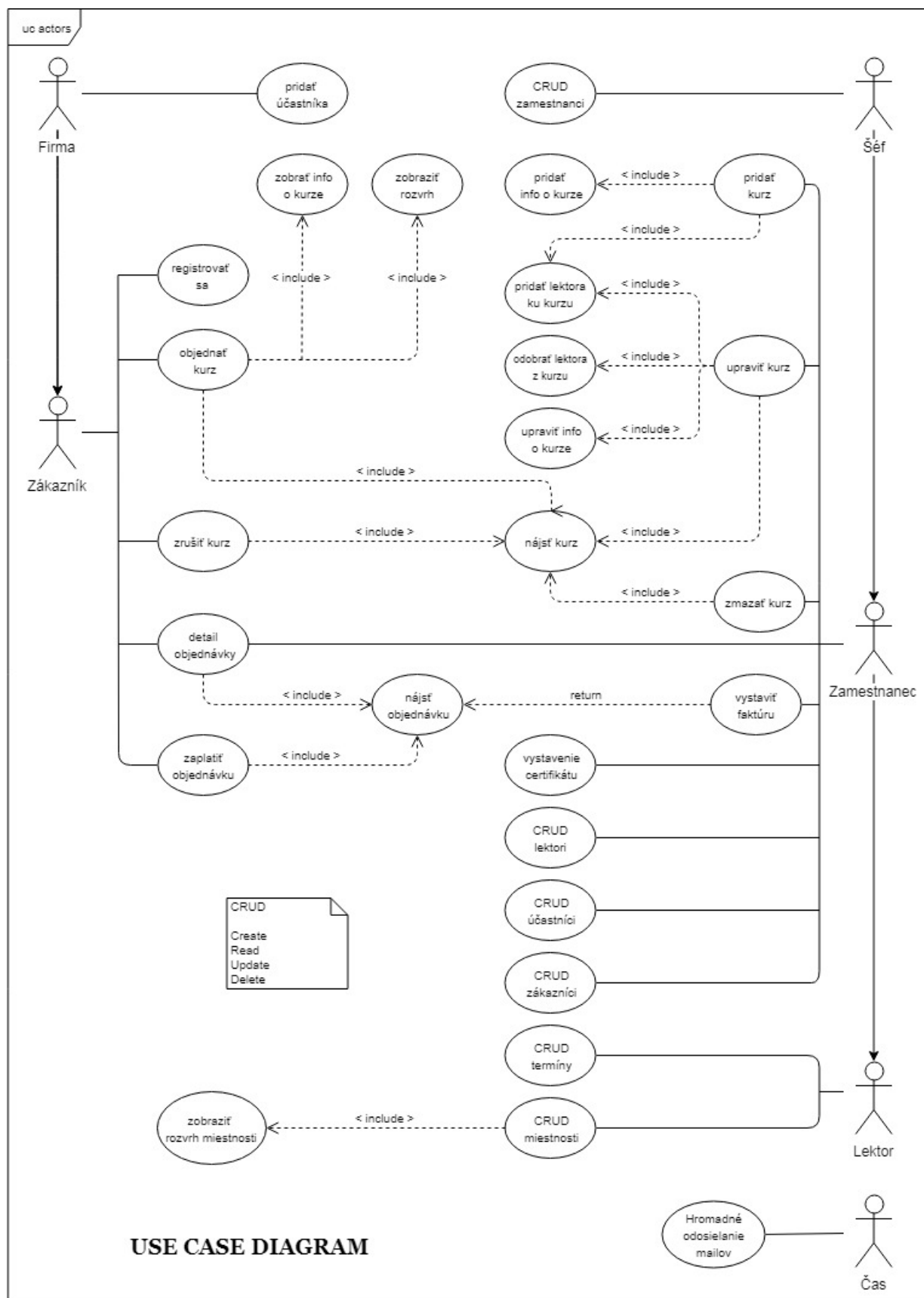
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

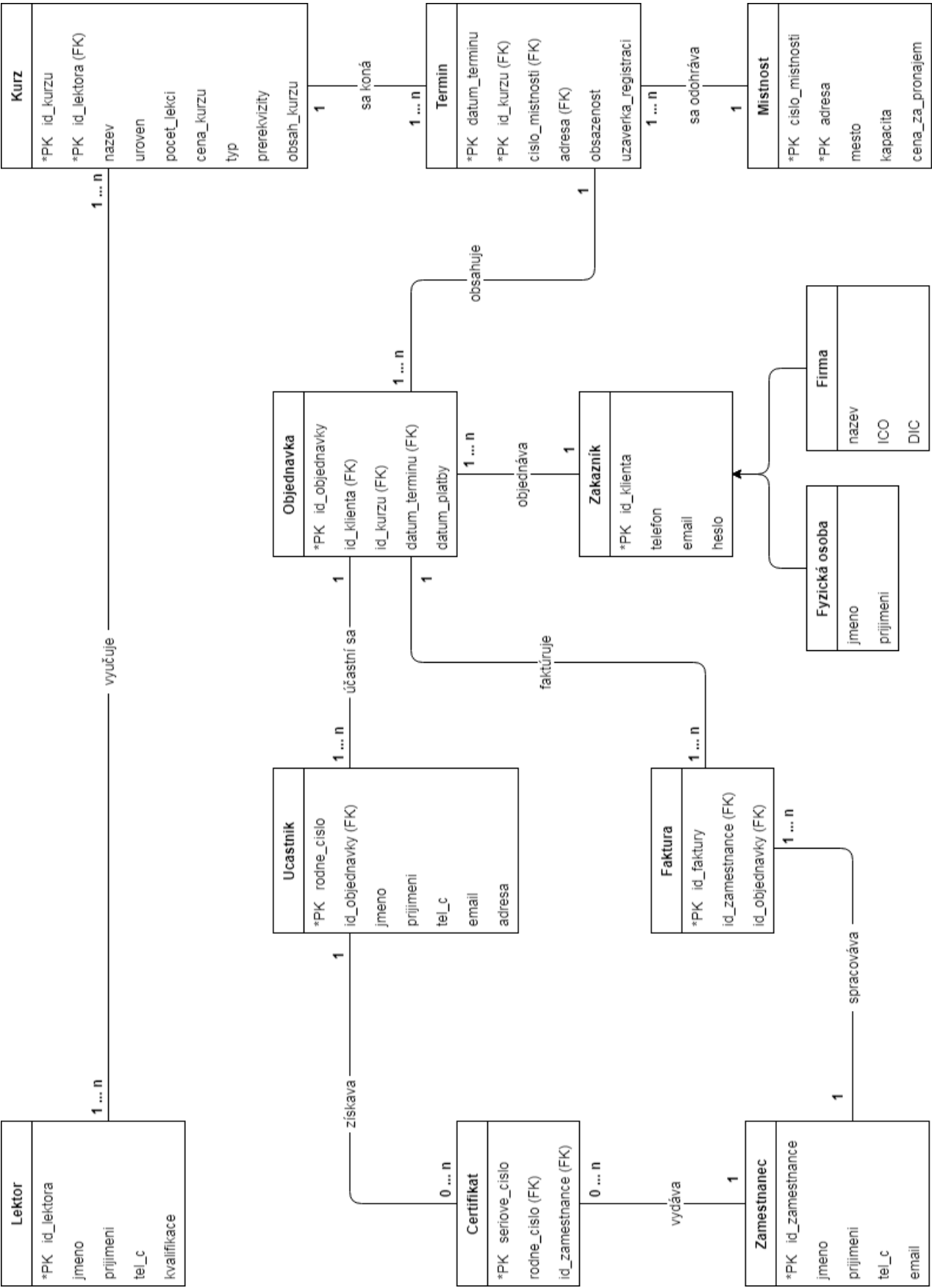
IDS - Databázové systémy
Školicí středisko

Róbert Ďurovič [xdurov01], Marek Poláček [xpolac30]

1 Školící středisko

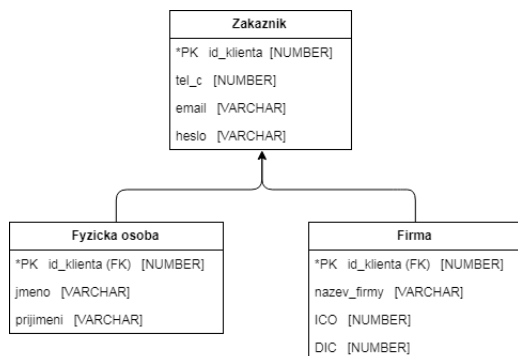
Navrhněte jednoduchý IS malého školícího střediska, které organizuje kurzy orientované na programové vybavení osobních počítačů. Kurzy probíhají v různých městech v pronajatých místnostech a jsou vypisovány přímo střediskem nebo na základě objednávky zákazníka (firma objednávací kurz pro své zaměstnance). Firma si může objednat určitý typ kurzu, u kterého jsou k dispozici informace o jeho úrovni, obtížnosti, prerekvizitách, ceně, počtu a obsahu lekcí. Konkrétní kurzy určitého typu, které si zákazník/firma objednává jsou vedeny jedním lektorem, mají maximální kapacitu účastníků a odehrávají se v určité místnosti v konkrétní čas a den v týdnu. U vypsaných kurzů lze dohledat jaké lekce obsahují a datum jejich konání. Školící středisko si může pronajímat místnosti v různých budovách (i v různých městech), přičemž každá místnost má stanovenou cenu za hodinu a maximální kapacitu. Zároveň předpokládejte, že jeden lektor může být vyškolen pro více typů kurzů, toto modelujte. Účastníci po absolvování kurzu obdrží certifikát se svým jménem na svou fyzickou adresu. Systém musí být na požádání schopen vypsát rozvrh pro jednotlivé místnosti. Zaměstnanec školícího střediska má možnost vložit do systému nové typy kurzu a konkrétní kurzy. Lektor může měnit čas a místnost konání konkrétního kurzu. Zákazník si může vypsát u kurzů, které platí, své přihlášené zaměstnance (zaměstnanec si však může kurz platit sám a tudíž by neměl být takto dohledatelný svým zaměstnavatelem, a také může změnit zaměstnavatele). Potenciální účastník kurzu si může přes webové rozhraní vypsát rozvrh vypisovaných kurzů, zjistit počet volných míst v kurzu a zaregistrovat se.





2 Generalizácia/specializácia

V našom zadaní sme problém generalizácie/specializácie riešili na prípade zákazníka. Zákazník môže byť ako fyzická osoba, tak aj firma. Typicky sa to rieši pomocou troch tabuliek: *Zákazník*, *Fyzická osoba* a *Firma*, kde v posledných dvoch tabuľkách by cudzím kľúčom bol primárny kľúč tabuľky *Zákazník*.



3 Implementácia SQL skriptu

Skript na začiatku zahodí pomocou príkazu *DROP* databázové objekty, aby sme predišli možným konfliktom. Následne sa vytvoria tabuľky, nastaví sa kľúče, sekvencie, trigger a procedúry. Skript nakoniec naplní tabuľky testovacími dátami, nad ktorými potom prevádzame na ukážku niekoľko *SELECT* príkazov a procedúr.

4 Triggery

Projekt obsahuje celkovo 3 trigger, pričom dva z nich sa spúšťajú po vložení alebo aktualizácii dát do tabuľky (*AFTER INSERT OR UPDATE*) a jeden sa spúšťa pred vložením (*BEFORE INSERT OR UPDATE*).

Prvý z týchto triggerov slúži na validáciu dátumu termínu kurzu. V prípade, že vkladáme do tabuľky dátum z minulosti alebo dátum konkrétného dňa, napr. 24/12/2018, na výstupe sa nám objaví chybové hlásenie.

Druhý trigger validuje rodné číslo účastníka kurzu. Kontrolujeme či je toto číslo deliteľné jedenástimi. Z čísla postupne extrahujeme deň, mesiac i rok. A na základe týchto údajov potom ešte zisťujeme či ide o ženu alebo nie v prípade, že mesiac >50. Ak nastane niekde chyba a číslo nie je validné, na výstupe sa objaví chybové hlásenie.

Posledný trigger slúži na autoinkrementáciu identifikátora v tabuľke zamestnanec v prípade, že pri vkladaní dát nie je id určené.

5 Procedúry

Projekt obsahuje celkovo 2 netriviálne procedúry s využitím *CURSORu* a dátovým typom odkazujúcim na riadok v tvare *nazov_tabulky%ROWTYPE*.

Prvá procedúra počíta priemernú cenu jednej lekcie z vybraného kurzu, ktorý nám bude slúžiť ako vstupný parameter. Nadefinujeme si pomocné premenné *num_lectures* (počet lekcií) a *overall_price* (celková cena kurzu), ktoré spočítavame v cykle. Po skončení cyklu spravíme aritmetický priemer delením týchto premenných. V prípade, že by sa delilo nulou, procedúra vypíše chybu. Nakoniec výsledok ešte zaokrúhlime, aby sme dostali číslo v priateľnom nedešatinnom formáte a hlásenie vypíšeme na výstup.

Druhá procedúra počíta celkovú percentuálnu vyťaženosť miestností v danom meste. Postup je podobný ako v prvej procedúre. Vstupný parametrom je mesto, v ktorom sa miestnosti nachádzajú. Máme nadefinované premenné *cap_classroom* (aktualna obsadenosť kurzu) a *capacity* (maximálna kapacita miestnosti). Cez cyklus spočítame tieto hodnoty pred dané mesto. Aby sme získali výsledok do premennej *overall_percentage* v percentách, musíme tie premenné podeliť a následne násobiť 100. Pre výsledok v priateľnom nedesatinnom formáte výsledok ešte zaokrúhlime a vyôfšeme na výstup. V prípade delenia nulou procedúra vypíše chybovú hlášku.

6 Explain plan

Vďaka *Explain plan* nám databáza ukáže ako spracováva určitý dotaz. Na základe toho môžeme sledovať spracovanie dotazu *SELECT* nad tabuľkami *Mistnost* a *Termin*, kde sa dotazujeme na maximálnu obsadenosť miestnosti v meste.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		5	425	7 (15)	00:00:01
1	HASH GROUP BY		5	425	7 (15)	00:00:01
* 2	HASH JOIN		5	425	6 (0)	00:00:01
3	TABLE ACCESS FULL	MISTNOST	5	205	3 (0)	00:00:01
4	TABLE ACCESS FULL	TERMIN	5	220	3 (0)	00:00:01

V znázornenej tabuľke vidíme niekoľko operácií, ktoré si vysvetlíme detailnejšie. *SELECT STATEMENT* znamená vykonaný *SELECT* dotaz. *HASH GROUP BY* nám grupuje položky podľa hashovacieho kľúča. *TABLE ACCESS FULL* značí prechod tabuľkou bez použitia indexov.

7 Materializovaný pohľad

V projekte sme implementovali materializovaný pohľad, ktorý patrí druhému členovi tímu. Ten používa tabuľky nadefinované prvým členom. Najprv sme teda vytvorili materializované logy, ktoré uchovávajú zmeny hlavnej tabuľky a použili *REFRESH FAST ON COMMIT*, aby sa nemusel dotaz spúšťať celý. *CACHE* nám zabezpečuje optimalizáciu čítania z pohľadu a vďaka *BUILD IMMEDIATE* sa nám materializovaný pohľad okamžite naplní. *ENABLE QUERY REWRITE* znamená použitie materializovaného pohľadu optimalizátorom. Pohľad sme potom otestovali na priloženom ukázkovom dotaze a cez *COMMIT* potvrdili zmeny.

8 Záver

Projekt sme spracovali a otestovali v prostredí Oracle na serveroch Oracle 12c pomocou softwaru SQL Developer. K projektu nám stačili vedomosti z projektu IDS nadobudnuté na prednáškach a demonštračných cvičeniach.