



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

IDS - Databázové systémy

Semestrální projekt

Lucia Makaiová (xmakai00)
Tadeáš Kachyňa (xkachy00)

2. května 2022

Obsah

1	Zadání	2
1.1	Doplnění	2
2	Use Case Diagram	3
3	Návrh relační databáze	4
4	Implementace	5
4.1	Generalizace/Specializace	5
4.2	Triggery	5
4.3	Procedury	6
4.4	Explain Plan	6
4.5	Přístupová práva	6
4.6	Materializovaný pohled	6

Tato dokumentace popisuje projekt spočívající v implementaci relační databáze v jazyce **SQL** na základě převzatého zadání z předmětu IUS. Popisuje všechny části odevzdávaného projektu včetně provedených úprav na ERD diagramu v průběhu vypracovávání. Jako implementační prostředí jsme zvolili *DataGrip* a školní databázový server *Oracle*.

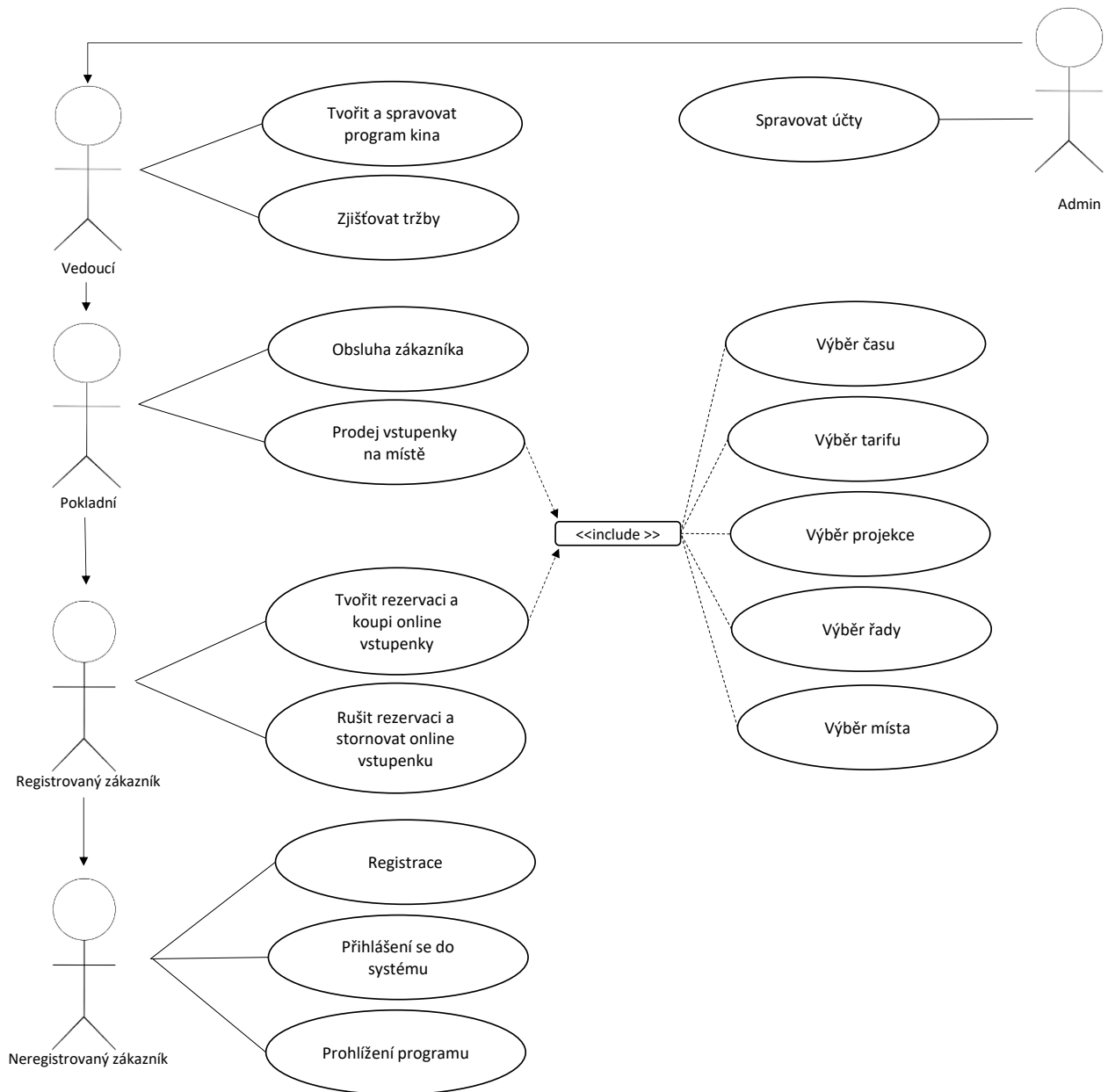
1 Zadání

Řetězec vlastní několik multikin a každé multikino obsahuje několik promítacích sálů. V sálech probíhají projekce filmů. Systém musí umožnit klientům připojeným přes webové rozhraní vyhledávání projekcí podle názvu filmu, žánru, kina, apod. Systém umožní klientovi zarezervovat si na zvolenou projekci konkrétní sedadla (jeden klient si může zarezervovat i více sedadel), či dříve zadanou rezervaci zrušit. Cena za vstupenku se liší podle toho, zda je divák dítě, mládež, dospělý, či důchodce, zda se jedná o dopolední, odpolední, či večerní představení, atd. Pokladní má možnost prodat vstupenku, ať už při koupi na místě, tak i na základě webové rezervace. V případě online rezervace je taky možnost zaplatit za vstupenky předem. V tom případě pokladní nebo automaty na to určeny vystaví zákazníkovi vstupenku. Proto je v případě online vstupenky nutno kromě všech původních dat vstupenky uchovat také informace o stavu platby. Vedoucí pracovníci jednotlivých poboček mají možnost zjistit tržby jednotlivých multikin.

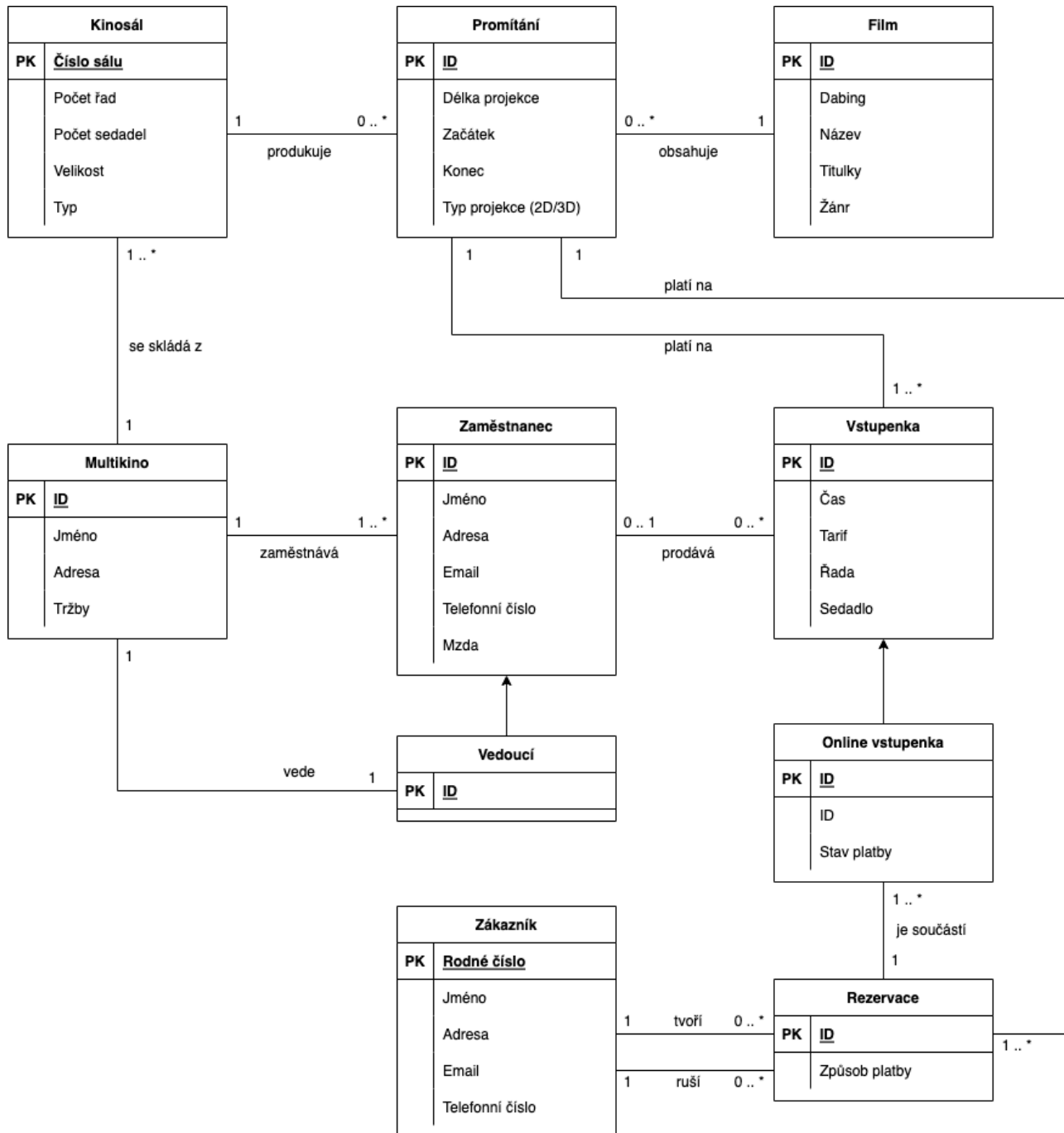
1.1 Doplnění

Zadání jsme v průběhu vypracovávání mírně měnili. Po prvním odevzdání jsme dostali několik připomínek k ERD diagramu. Na jejich základech jsme zmíněné chyby v diagramu odstranili. A to konkrétně entitu *Program*, protože nebyla jednoduše implementovatelná, jak jsme si ve skutečnosti mysleli, že by byla. Následně došlo k přidání atributu k entitě *Film* - *název filmu* a *Zaměstnanec* - *mzda*.

2 Use Case Diagram

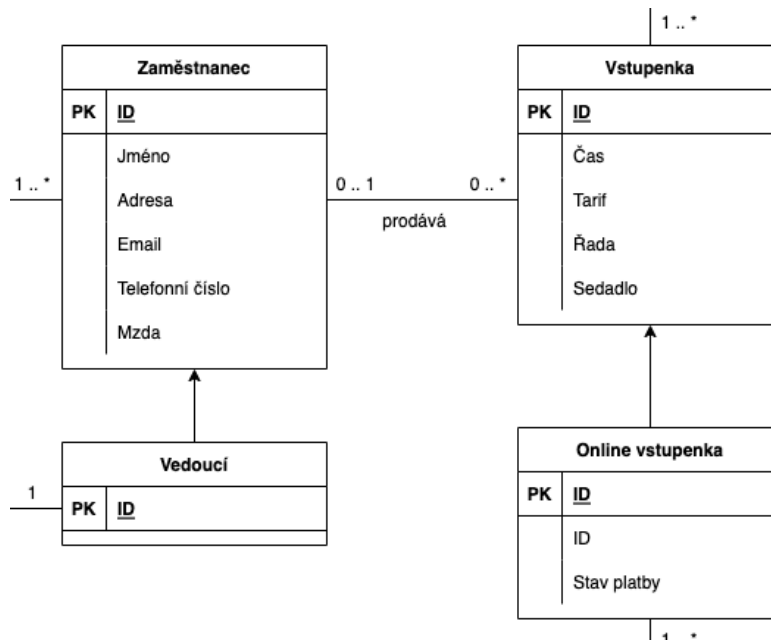


3 Návrh relační databáze



4 Implementace

4.1 Generalizace/Specializace



Vztah generalizace/specializace jsme v návrhu aplikovali na dvě entity. Ze zaměstnance se specializuje entita *Vedoucí*. Ze Vstupenky *Online Vstupenka*. Na oba vztahy jsme aplikovali čtvrtý typ transformace zmiňovaný na přednášce.

Implementaci SQL skriptu jsme započali vytvořením databázových tabulek, které jsme si již předtím definovali v relačním modelu. Taktéž jsme implementovali několik různých omezení, např. na format telefonních čísel nebo e-mailů za pomoci **CHECK**. Další fáze spočívala v napsání několika jednoduchých dotazů **SELECT**, kde jsme podle zadání museli použít klauzuli **GROUP BY**, predikáty **EXISTS**, **IN**, spojení dvou či tří tabulek nebo zakonponování vnořených dotazů. Následně bylo implementováno několik dalších částí skriptu, které jsou popsány detailněji v jednotlivých podkapitolkách níže.

4.2 Triggery

Triggery jsme implementovali celkem tři. První z nich, nejdelší a také nejsložitější, slouží k validaci zákazníkova rodného čísla. Kontrolujeme ho podle aktuálně platných norem, které obsahují několik podmínek. Druhý trigger slouží k převedení tržeb multikin z českých korun na eura. Poslední trigger slouží ke generování primárního klíče kinosálu. Pro lepší orientaci, aby šlo na první pohled říct z primárního klíče ke kterému multikinu kinosál patří, tak ke 4místnému cizímu klíči multikina přičítáme dvojčíferné číslo, které je inkrementována po jednotkách. Nyní se dá snadno z prvního dvoučíslí odvodit do jakého kina sál patří a z posledního dvojčíslí jeho identifikační číslo.

4.3 Procedury

Ve skriptu implementujeme dvě procedury. První z nich filtruje a zobrazuje počet prodaných vstupenek dle tarifu specifikovaný uživatelem z celkového počtu prodaných vstupenek, taktéž zobrazuje procentuální zastoupení a poměr prodaných vstupenek online anebo na místě. Druhá procedura zobrazuje průměrný plat zaměstnanců multikin.

4.4 Explain Plan

Explain plan nám umožňuje zjistit postupnost kroků, které sa používají na přístup k údajům v systému správy relačních databází SQL. V našem projektu jsme zjistili postupnost kroků dotazu "Kolik vstupenek prodali zaměstnanci, kteří jsou z Brna". Následně jsme vykonali optimalizaci pomocí vytvoření indexu pre sloupec míst z tabulky zaměstnanců a taktéž index pre sloupec který v tabulce vstupenek uchovává informaci o tom, který zaměstnanec danou vstupenku prodal. Díky indexům sa urychlilo vyhledávání, protože nebylo v daných případech nutné přistupovat k celé tabulce. Úspěšnou optimalizací jsme nakonec ověřili opětovným použitím příkazu `EXPLAIN PLAN` na stejný dotaz.

Před optimalizací							Po optimalizaci						
Plan hash value: 1128639545							Plan hash value: 1206256284						
ID	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	ID	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6	2496	8 (25)	00:00:01	0	SELECT STATEMENT		6	2496	3 (34)	00:00:01
1	SORT ORDER BY		6	2496	8 (25)	00:00:01	1	SORT ORDER BY		6	2496	3 (34)	00:00:01
2	HASH GROUP BY		6	2496	8 (25)	00:00:01	2	SORT GROUP BY NOSORT		6	2496	3 (34)	00:00:01
3	HASH JOIN		6	2496	6 (0)	00:00:01	3	NESTED LOOPS		6	2496	2 (0)	00:00:01
4	NESTED LOOPS		3	1209	3 (0)	00:00:01	4	NESTED LOOPS		3	1209	2 (0)	00:00:01
5	TABLE ACCESS FULL	ZAMESTNANEC	3	1200	3 (0)	00:00:01	5	TABLE ACCESS BY INDEX ROWID	ZAMESTNANEC	3	1200	2 (0)	00:00:01
6	INDEX UNIQUE SCAN	SYS_C001414538	1	3	0 (0)	00:00:01	6	INDEX FULL SCAN	NAME_INDEX	4		1 (0)	00:00:01
7	TABLE ACCESS FULL	VSTUPENKA	8	104	3 (0)	00:00:01	7	INDEX UNIQUE SCAN	SYS_C001414538	1	3	0 (0)	00:00:01
8							8	INDEX RANGE SCAN	TOWN_INDEX	2	26	0 (0)	00:00:01
Predicate Information (identified by operation id):							Predicate Information (identified by operation id):						

4.5 Přístupová práva

Přístupová práva pro druhého člena týmu jsme vyřešili příkazem `GRANT ALL [TABLES|PROCEDURE|MATERIALIZED VIEW] TO [USER]` pro všechny tabulky definované v naše databázi, procedury a taktéž materializovaný pohled. Tím pádem je druhý uživatel dalším administrátorem, protože má defakto plný přístup.

4.6 Materializovaný pohled

Při tvorbě materializovaného pohledu pro druhého uživatele jsme vytvořili materializované logy. Tím pádem je nám umožněné používat `FAST REFRESH ON COMMIT`. Pohled můžeme aktualizovat bez nutnosti vytvářet nový. Pro demonstraci jsme zvolili tabulku *Zaměstnanec* ze které budeme importovat všechna neosobní data (např. kterými nejsou data o mzdě zaměstnanců). Následně tabulku aktualizujeme, provedeme příkaz `COMMIT` a můžeme zaznamenat, že se aktualizoval i pohled.