



Databázové systémy

2015/2016

Projekt 4. - Dokumentácia

Zadanie č. 25 – Pracovná agentúra

Michal Ondrejó

[xondre08]

Július Kéry

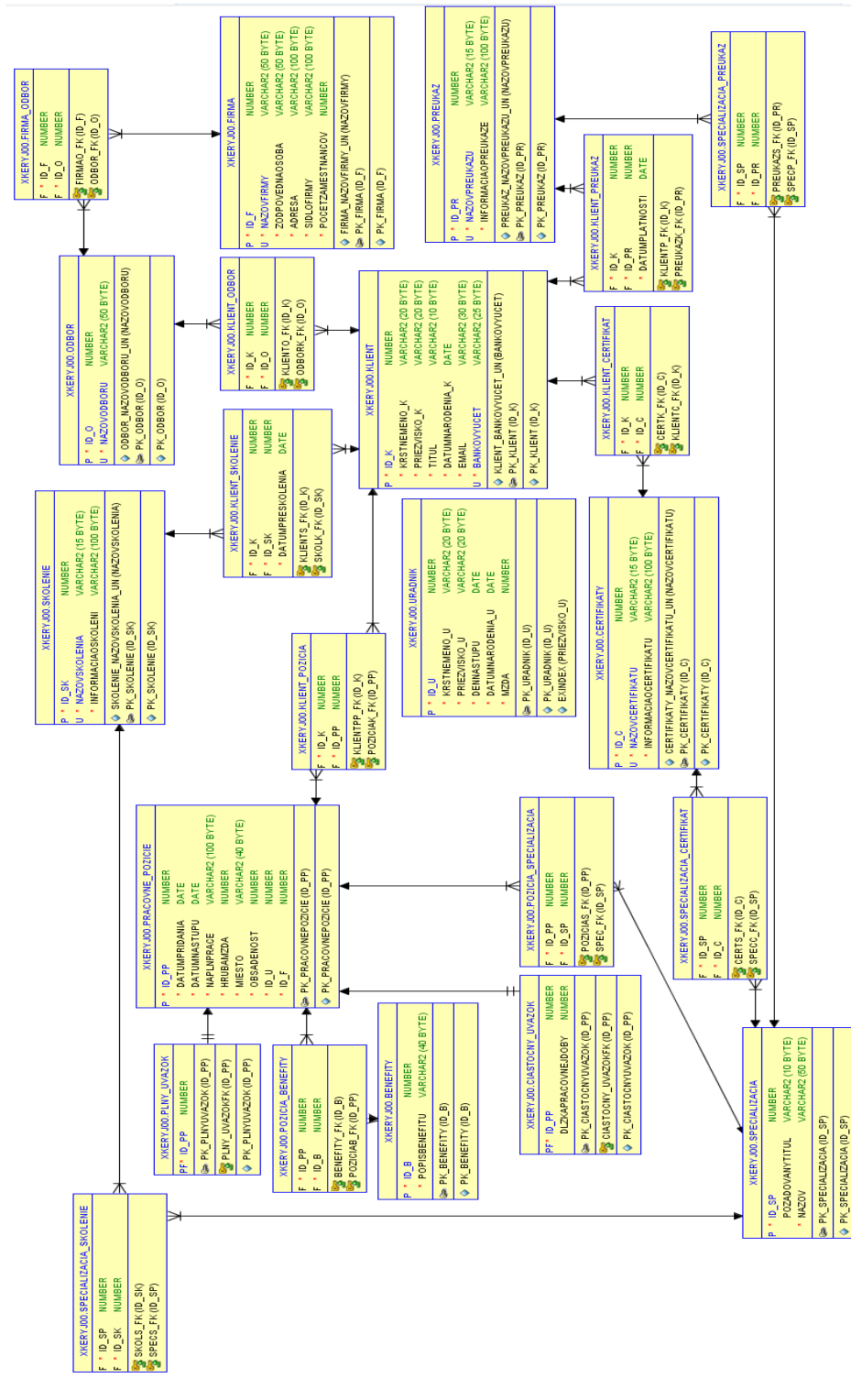
[xkeryj00]

Brno, 1.5. 2016

1. Zadanie

- Vypracujete návrh nového informačního systému pro malou pracovní agenturu.
- Agentura zpracovává nabídky od různých zaměstnavatelů/firem, o kterých samozřejmě má základní informace jako název/odpovědná osoba, adresu, sídlo firmy, počet zaměstnanců, roční obrat, obor ve kterém působí (IT, reklama, účetnictví, zdravotnictví, atd.).
- Jednotlivé firmy mohou mít v nabídce více pozic s určitým zaměřením/specializací pro kterou je pozice vypsána.
 - U jednotlivých nabídek pozic lze vypsát stručný popis náplně práce, hrubou mzdu, místo (lokalitu) působení, různé benefity, které pozice nabízí (služební auto, služební mobil, stravenky, atd.), zda se jedná o pozici na částečný/plný úvazek. Specializace (zaměření), do kterého nabízená pozice spadá, obsahuje obecné informace typu, do kterého oboru patří, všeobecně platné prerekvizity pro danou pozici (požadovaný titul, atd.), přičemž každá specializace může vyžadovat i různé certifikáty/školení/průkazy (řidičský průkaz, potravinářský průkaz, FCE certifikát, atd.).
- Klient pracovní agentury si může nabídky pouze prohlížet přes webové rozhraní nebo se přímo zaregistrovat a zadat informace o své osobě.
 - Pokud se klient zaregistruje, má možnost zadat o jaké nabídky (z jakého oboru) má zájem, vyplnit informace o svých certifikátech/školeních/průkazech a jakou má specializaci. Registrovaný uživatel může v IS požádat o zasílání emailů s novými nabídkami odpovídajícími jeho specializaci, jež jsou nabízeny v lokalitě jeho bydliště či podle toho v jakém oboru by chtěl pracovat.

2. Schéma databázy



3. Implementácia

3.1 – Generalizácia / Špecializácia

- Pre toto zadanie sme generalizáciu využili pre tabuľku PRACOVNE_POZICIE, kde pracovná pozícia mohla byť na plný alebo čiastočný úväzok.

3.2 – Trigger

- Prvým trigger, ktorý sme implementovali bola auto-inkrementácia ID pre tabuľku Klient.
 - Realizovaný pomocou sekvencie, kde si ukladalo číslo posledného pridaného ID.
- Druhý trigger slúži na overenie, či zadaný IBAN v tabuľke Klienta je správny.
 - 1. kontrola, či zadaný IBAN má dĺžku 24 znakov
 - 2. kontrola prvých 2 znakov, či sú to písmena (A - Z)
 - 3. kontrola či ostatné znaky sú iba čísla (0 - 9)
- Obidva trigger sa spúšťajú ešte pred vložením údaje do danej tabuľky.

3.3 – Procedúry

- Implementovali sme 2 procedúry:
 - pozad_Titul(SP_ID) *SP_ID => ID špecializácie
 - uradnici(U_ID) *U_ID => ID úradníka
- 1. procedúra (pozad_Titul) nám ukáže, či daná špecializácia, kde jej ID je argumentov tento procedúry, vyžaduje určitý titul.
 - Ošetrené máme stav, kde v tabuľke nie je zadaný požadovaný Titul a taktiež situáciu, že zadané ID nie je evidované v tabuľke.
- 2. procedúra (uradnici) ma ako argument ID úradníka, ktoré keď zadáme, tak prejde všetkých ostatných úradníkov a porovná ich platy.
 - Buď sa naskytne stav, že daný úradník má menší, väčší alebo rovnaký plat.
 - Ošetrené máme, ak sa porovná úradník sám so sebou(oznam o tom, že tento úradník bol zadaný do porovnania) a taktiež , že zadané ID úradníka nie je evidované v tabuľke.

3.4 – Explain Plan + Index

- Túto časť projektu sme implementovali s jednoduchým SELECT dotazom.
 - Za 1. sme spustili EXPLAIN PLAN (plán spracovania daného select dotazu) bez indexu.
 - Ako ďalší krok sme si práve zadefinovali INDEX a EXPLAIN PLAN zopakovali.
- Výstup: 1. obrázok – Bez indexu // 2. obrázok – s indexom

Id	Operation	Name	Rows	Bytes	Cost	(%CPU)	Time
0	SELECT STATEMENT		7	357	4	(25)	00:00:01
1	HASH GROUP BY		7	357	4	(25)	00:00:01
2	NESTED LOOPS		7	357	3	(0)	00:00:01
3	NESTED LOOPS		7	357	3	(0)	00:00:01
4	TABLE ACCESS FULL	PRACOVNE_POZICIE	7	182	3	(0)	00:00:01
* 5	INDEX UNIQUE SCAN	PK_URADNIK	1		0	(0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID	URADNIK	1	25	0	(0)	00:00:01

Id	Operation	Name	Rows	Bytes	Cost	(%CPU)	Time
0	SELECT STATEMENT		7	357	6	(17)	00:00:01
1	HASH GROUP BY		7	357	6	(17)	00:00:01
* 2	HASH JOIN		7	357	5	(0)	00:00:01
3	TABLE ACCESS BY INDEX ROWID BATCHED	URADNIK	4	100	2	(0)	00:00:01
4	INDEX FULL SCAN	EXINDEX	4		1	(0)	00:00:01
5	TABLE ACCESS FULL	PRACOVNE_POZICIE	7	182	3	(0)	00:00:01

- EXPLAIN PLAN:
 - SELECT STATEMENT -> dotaz select.
 - HASH GROUP BY -> zgrupovanie podľa hashovacieho kľúča.
 - NESTED LOOPS -> tabuľky spojené naivne, čo značí, že prechádza po 1 tabuľke a pre každú položku 1. tabuľky sa prechádzajú všetky riadky 2. tabuľky.
 - **WITHOUT INDEX:**
 - TABLE ACCESS FULL -> tabuľka sa prechádza celá bez indexov.
 - INDEX UNIQUE SCAN -> prístup k tabuľkám cez B-strom(vypadne nám 1 jedinečný riadok podľa kľúča /**/).
 - **WITH INDEX:**
 - TABLE ACCESS BY INDEX ROWID -> prístup do tabuľky cez konkrétny riadok(index).
 - INDEX FULL SCAN s indexom -> index použitý na výpis, bez nazerania do tabuľky.

3.5 – Prístupové práva

- Prístupové práva k tabuľkám a taktiež k procedúram má druhý kolega tímu plný prístup, čo mu umožňuje vkladať do všetkých tabuliek ako aj spúšťať procedúry.

3.6 – Materializovaný pohľad

- Pre materializovaný pohľad druhého člena tímu, využívajúci tabuľky definované prvým členom, sa najprv vytvorili materializované logy(uchované zmeny hlavnej tabuľky), kvôli zmenám.
- Používa sa FAST REFRESH ON COMMIT -> z dôvodu, aby sa nemusel celý pohľad od začiatku spúšťať.
- CACHE -> optimalizácia toho, čo sa načítalo z pohľadu.
- BUILD IMMEDIATE -> hneď po vytvorení dôjde k naplneniu.
- ENABLE QUERY REWRITE -> MP používaný optimalizátorom, v prípade vyhodnotenia dotazu.

4. Záver

- Tento skript bol testovaný v prostredí Oracle
- Školský server: Oracle12c
- Skript bol tvorený v text. editore: Notepad++
- Skript bol testovaný cez program: Oracle Sql Developer
- Zdroje: Školské materiály, StackOverflow, w3schools