



Dokumentace

Zadání č. 2. – Marketingová (reklamní) firma [IUS]

Databázové systémy

16. dubna 2021

Jakub Komárek (xkomar33)
Lacko Dávid (xlacko09)

Obsah

1	Zadání	2
2	Diagram případů užití	3
3	Datový model (ERD)	4
4	Implementace	5
4.1	Index a EXPLAIN PLAIN	5

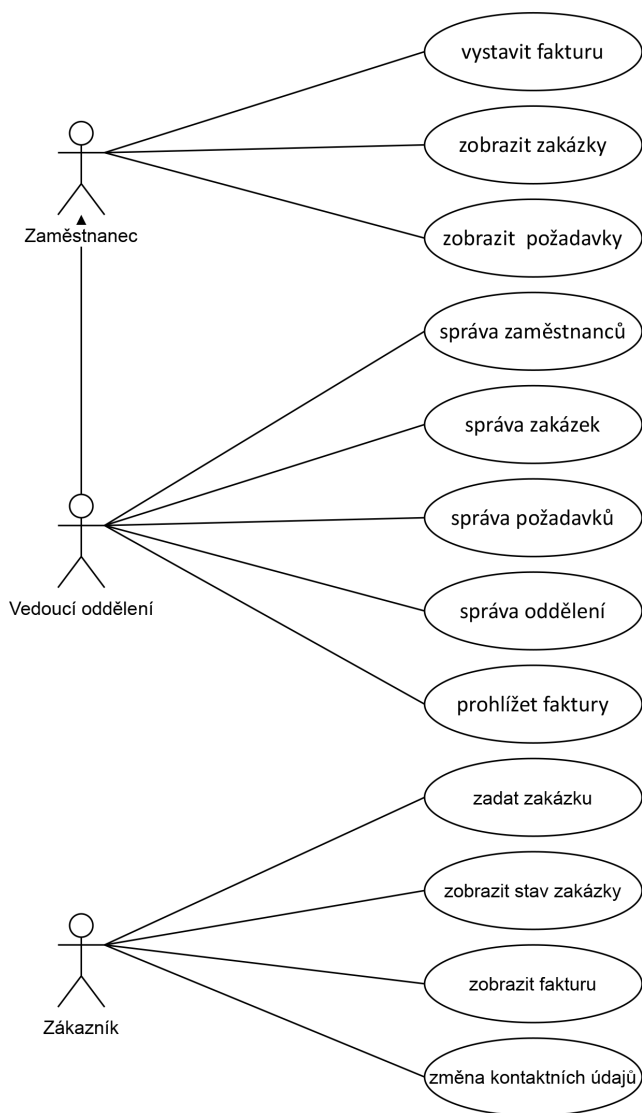
1 Zadání

Vytvořte jednoduchý IS pro malou marketingovou firmu, která řeší různé zakázky. Firma je členěna na několik oddělení. U každého oddělení je třeba uchovávat jeho název, stručný popis, místnost, zaměstnance a vedoucího. Zaměstnanci pracují na jednom oddělení, je u nich uchována informace o jejich mzdě, titulu, specializaci, atp. Předpokládejte že vedoucí má na starost pouze jedno oddělení. Zakázka sestává z jednoho či více požadavků, přičemž u každého požadavku je evidován jeho typ (např. letáková kampaň, reklama v médiích, billboard, ...), cena, doba platnosti marketingové akce, počet potřebných brigádníků (ty v systému nemusíte modelovat). Požadavek může zpracovávat i více oddělení. U každé zakázky eviduje informační systém zadavatele (klienta) s jeho základními informacemi. Dále je o zakázce evidováno datum zadání, celková cena (součet cen jednotlivých požadavků), termín dodání. Každá zakázka má zodpovědného zaměstnance, který vystavuje faktury na požadavky k zakázce. Jedna faktura může být i na více požadavků. Evidujte zároveň i stav zakázky a jednotlivých požadavků (návrh, příprava, řešení, konec).

2 Diagram případů užití

Z diagramu vyplývá, že zaměstnanec nebude mít přístup k žádným úpravám databáze, jediná výjimka je vystavování faktur. Při práci může zobrazovat zakázky a požadavky, aby je poté zpracoval. Jeho vedoucí poté může práci průběžně monitorovat a podle toho editovat či jinak upravovat zakázky a požadavky. Vedoucí také může spravovat zaměstnance a nahlížet do faktur.

Zákazník může vytvořit zakázku a poté dohlížet na její zpracování. Může také zobrazovat faktury pro něj učené. Kdyby se zákazník přestěhoval nebo si např. chtěl změnit email, je mu umožněna změna jeho kontaktních údajů.

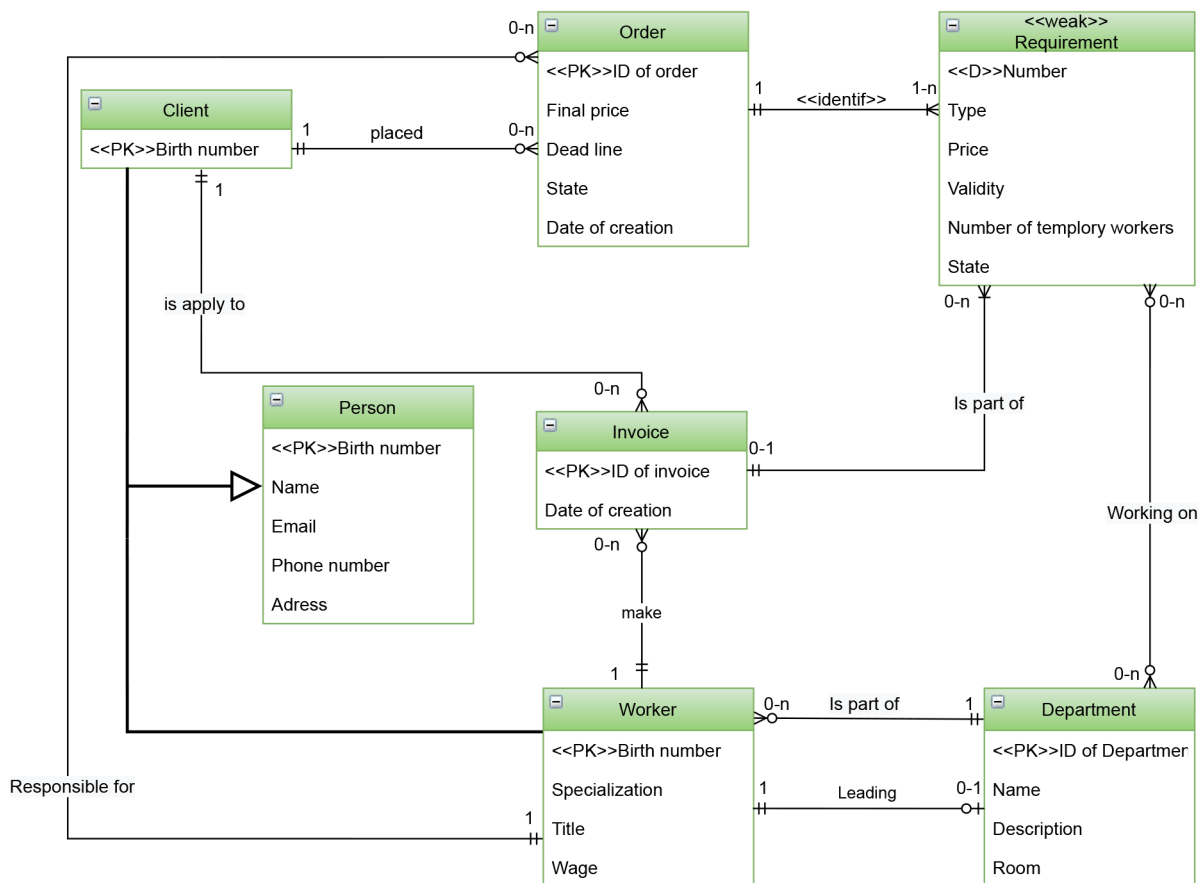


Obrázek 1: Model případů užití

3 Datový model (ERD)

Entita osoba má jako primární klíč rodné číslo (je vždy unikátní) a uchovává se u ní základní údaje pro kontaktování osoby. Tato entita má dvě specializace klient a pracovník. U klienta není potřeba uchovávat další informace navíc. U pracovníka jsou navíc uchovány údaje o specializaci, titulu a mzdy. Pracovník musí být součástí jednoho oddělení a zároveň může být ve vztahu s oddělením jako vedoucí (může vést pouze jedno). Oddělení musí mít unikátní idedntifikátor.

Klient může být ve vztahu s 0-n objednávkami. S objednávkou je spřažen jeden odpovědný pracovník, který ji má na starost. Objednávka se skládá z jednoho či více požadavků a má svůj unikátní identifikátor. Požadavky jsou slabou entitou a pro jednoznačnou identifikaci musí být spojeny s objednávkou. Jednotlivé požadavky objednávky jsou odlišeny diskriminátorem. Faktura je spojena s pracovníkem, který ji vytvořil. Její P.K. je číslo faktury (musí se vygenerovat unikátní číslo). Faktura je přiřazena jednomu klientovy a je propojena s jedním či více požadavků.



Obrázek 2: Datový model (ERD)

4 Implementace

4.1 Index a EXPLAIN PLAIN

Vybraný SELECT příkaz.

```
SELECT COUNT(*) noOfPeople, p.city FROM PERSON p, "Client" c
WHERE p.birthNum=c.birthNum GROUP BY p.city
```

Před vytvořením vlastního indexu.

Id	Operation	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT	4	128	4 (25)	00:00:01
1	HASH GROUP BY	4	128	4 (25)	00:00:01
2	NESTED LOOPS	4	128	3 (0)	00:00:01
3	TABLE ACCESS FULL	9	225	3 (0)	00:00:01
* 4	INDEX UNIQUE SCAN	1	7	0 (0)	00:00:01

Predicate Information (identified by operation id):

```
" 4 - access("P"."BIRTHNUM"="C"."BIRTHNUM") "
```

Note

- dynamic statistics used: dynamic sampling (level=2)

Vzhľadom na to, že spojenie JOIN je pomocou primárnych kľúčov, ktoré majú implicitne vygenerovaný index pri vytváraní tabuľky, tak je toto spojenie vykonané rýchlo. Avšak problém sa ukazuje pri zoskupení GROUP BY, kde sa musí pristupovať do samotnej tabuľky pre prístup k hodnote stĺpca city. Preto vytvoríme index pre tento stĺpec.

[fontsize=\small]

Id	Operation	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT	4	128	2 (0)	00:00:01
1	SORT GROUP BY NOSORT	4	128	2 (0)	00:00:01
2	NESTED LOOPS	4	128	2 (0)	00:00:01
3	TABLE ACCESS BY INDEX ROWID	9	225	2 (0)	00:00:01
4	INDEX FULL SCAN	9		1 (0)	00:00:01
* 5	INDEX UNIQUE SCAN	1	7	0 (0)	00:00:01

Predicate Information (identified by operation id):

```
" 5 - access("P"."BIRTHNUM"="C"."BIRTHNUM") "
```

Note

- dynamic statistics used: dynamic sampling (level=2)

Ako si môžeme všimnúť, tak po vytvorení indexu sa pri zoskupovaní používajú hodnoty z neho a pribudol na riadku 4 prístup do indexu, miesto prístupu do samotnej tabuľky. Toto znížilo CPU náročnosť.