



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Databázové systémy
Dokumentace k projektu IDS

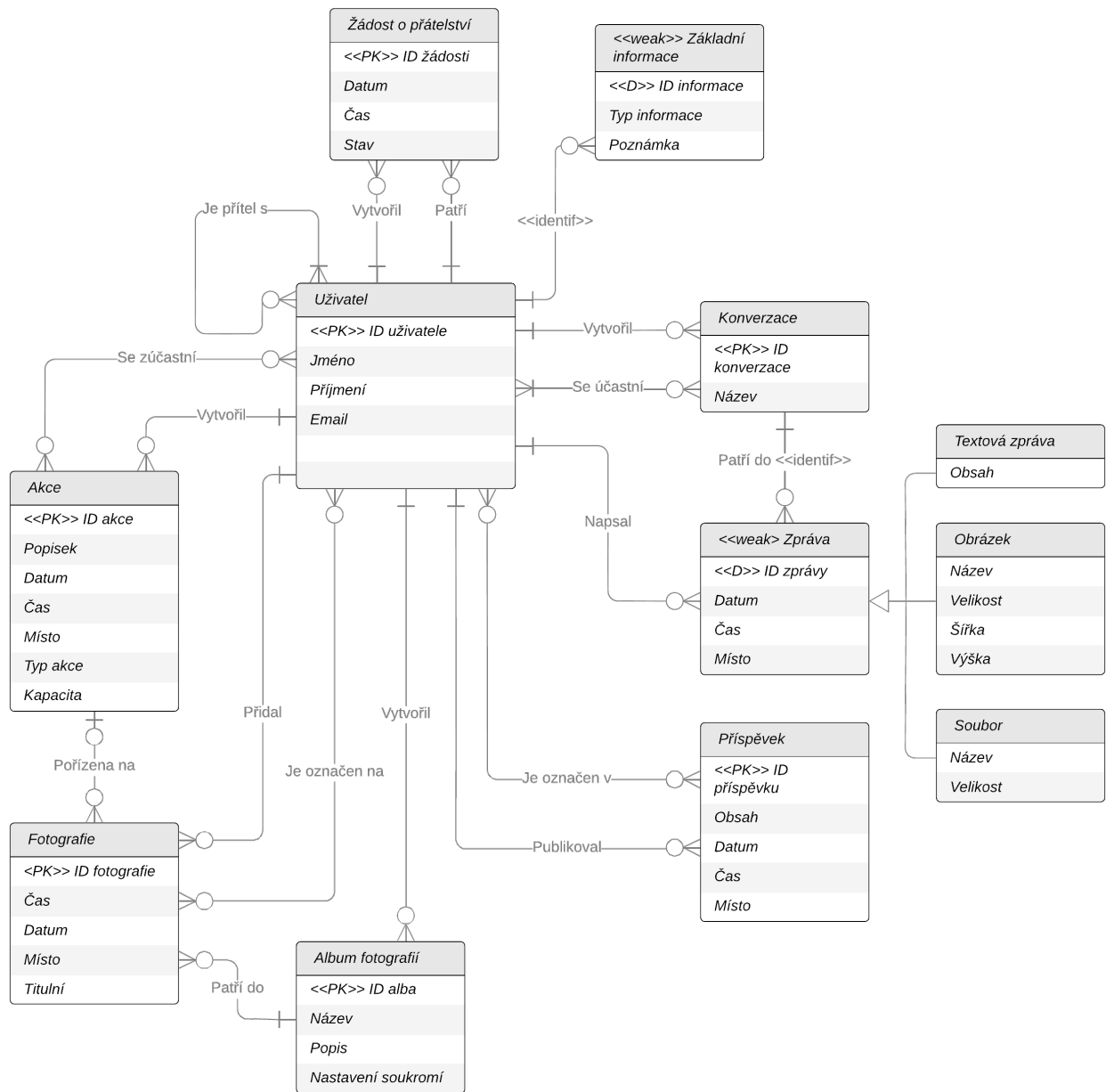
Obsah

1	Zadání	2
2	Databázový model a model případů užití	3
3	Vytvoření základních objektů schématu databáze	4
4	SQL dotazy SELECT	4
4.1	Spojení dvou tabulek	4
4.2	Spojení tří tabulek	5
4.3	Dotazy s GROUP BY	5
4.4	Dotaz s predikátem EXISTS	5
4.5	Dotaz s predikátem IN	5
5	Pokročilé objektů schématu databáze	6
5.1	Triggery	6
5.2	Procedury	6
5.3	Index	6
5.4	EXPLAIN PLAN	7
5.5	Matezializovaný pohled	7

1 Zadání

Na sociální síti bude možné uchovat veškeré základní informace o uživatelích (včetně škol, bydliště, zaměstnání, kontaktu, vztahů, . . .). Uživatelé si mohou mezi sebou vytvářet (imaginární) přátelství pomocí žádosti. Každý uživatel má svoji zeď, kde může on i jeho přátelé publikovat příspěvky, které budou mít obsah, datum, místo a čas publikování a můžou v nich být označeni i jiní uživatelé. Aby si uživatelé mohli sdílet nejen své příspěvky, ale také fotky, mohou vytvářet i alba fotek, které budou mít svůj název, nastavení soukromí a popis. Na jednotlivých fotkách mohou být označení různí uživatelé a bude u nich uveden čas, datum a místo pořízení a jedna z fotek bude vždy titulní fotka alba. Navíc může být fotka pořízena v rámci nějaké akce. Uživatelé si mohou prostřednictvím konverzací s jistým názvem, do níž může být zapojen jeden (on sám) a více uživatelů, vyměňovat zprávy, které budou mít svůj obsah, datum, čas a místo zaslání. Uživatelé mohou vytvářet akce, které se konají na určitém místě, v určitý čas a den a mohou mít nastavenou kapacitu (pokud není nastavena, kapacita je neomezená). Účastníci akce by měli znát, o jakou akci se jedná a pokud se jim akce zalíbí, tak se mohou akce, ať už jen virtuálně, či skutečně zúčastnit.

2 Databázový model a model případů užití



Obrázek 1: ERD Diagram



Obrázek 2: Use Case Diagram

3 Vytvoření základních objektů schématu databáze

4 SQL dotazy SELECT

4.1 Spojení dvou tabulek

```

1  -- Jake prispevky publikoval uzivatel
2  SELECT jmeno, prijmeni, obsah
3  FROM uzivatel U,
4       prispevek P
5  WHERE U.id = P.autor;
6
7  -- Dalsi dotaz
8  SELECT U.jmeno, U.prijmeni, I.vzdelani
9  FROM uzivatel U,
10       zakladni_informace I
11  WHERE U.id = I.uzivatel
12        AND I.vzdelani IS NOT NULL;

```

4.2 Spojení tří tabulek

```
1  -- Kteri uzivatele se zucastnili akce 'Koncert skupiny XYZ'
2  SELECT CONCAT(CONCAT(U.jmeno, ' '), U.prijmeni) jmeno
3  FROM  uzivatel U,
4         akce A,
5         ucastnici_akce UA
6  WHERE U.id = UA.uzivatel
7         AND UA.akce = A.id
8         AND A.popisek LIKE 'Koncert skupiny XYZ';
9
10 -- Na jak ch fotk ch jsou uzivatele oznaceni
11 SELECT CONCAT(CONCAT(U.jmeno, ' '), U.prijmeni) jmeno, F.cesta foto
12 FROM  uzivatel U,
13         foto_oznaceni FO,
14         fotografie F
15 WHERE U.id = FO.uzivatel
16        AND FO.fotografie = F.id
17 ORDER BY jmeno;
```

4.3 Dotazy s GROUP BY

```
1  -- Kolik fotek patri do daneho alba
2  SELECT A.nazev, COUNT(*) AS pocet_fotek_v_albumu
3  FROM  album A,
4         fotografie F
5  WHERE A.id = F.album
6  GROUP BY A.nazev;
7
8  -- Kolik fotek bylo porizeno na akci
9  SELECT popisek, COUNT(*) pocet_fotek
10 FROM  akce A,
11        fotografie F
12 WHERE A.id = F.akce
13 GROUP BY popisek;
```

4.4 Dotaz s predikátem EXISTS

```
1  -- Kteri uzivatele maji na profilu informaci o vzdelani a nemaji informaci o
   povolani
2  SELECT DISTINCT jmeno, prijmeni
3  FROM  uzivatel U,
4         zakladni_informace I
5  WHERE U.id = I.uzivatel
6         AND vzdelani IS NOT NULL
7         AND EXISTS(SELECT *
8                     FROM  zakladni_informace I
9                     WHERE U.id = I.uzivatel
10                    AND povolani IS NULL);
```

4.5 Dotaz s predikátem IN

```
1  -- Kteri uzivatele zalozili alespon jednu konverzaci
2  SELECT U.jmeno, U.prijmeni
3  FROM  uzivatel U
4  WHERE U.id IN (SELECT zakladatel FROM konverzace);
```

5 Pokročilé objektů schématu databáze

5.1 Triggery

- Trigger `trigger_valid_conversation` má za úkol hlídat, aby do dané konverzace mohl přidávat zprávy pouze uživatel, který je členem konverzace (v tabulce `uzivatel_v_konverzaci`).
- Trigger `trigger_capacity_check` zabraňuje, aby se po naplnění kapacity akce nemohl už nikdo další na akci přihlásit.

5.2 Procedury

- Procedura `naplnenost_akce` vypíše všechny akce a informace o zaplnění jejich kapacity v procentech (pokud kapacita akce chybí, tzn. je NULL, je kapacita akce neomezená, tudíž je naplněnost 0 %).
- Procedura `pocet_akci_uzivatele(id_uzivatele)` má jako parametr ID uživatele a po zavolání vypíše, na kolik akcí (proběhlých i neproběhlých) se uživatel přihlásil.

```
XOSINE00> BEGIN
        naplnenost_akce();
    END;
[2022-05-02 16:36:15] completed in 470 ms
Akce   |   Naplněnost
-----|-----
Koncert skupiny XYZ   |    0 %
FIT festival          |    2 %
Oslava narozenin Jardy |   100 %
Disco                 |    0 %
```

Obrázek 3: `naplnenost_akce`

```
XOSINE00> BEGIN
        pocet_akci_uzivatele(1);
    END;
[2022-05-02 16:36:25] completed in 103 ms
Uživatel s ID: 1 se účastní 2 akcí
```

Obrázek 4: `pocet_akci_uzivatele(id)`

5.3 Index

- Index `usersLastNameDesc` seřadí uživatele podle příjmení sestupně. To urychlí dotaz, kde vybíráme uživatele s příjmením začínajícím na písmeno "V". V se nachází na konci abecedy a procházet tabulku od začátku ke konci je náročné. INDEX nám v tomto případě zajistí, že se hledané hodnoty nebudou nacházet na konci tabulky, ale na začátku.

5.4 EXPLAIN PLAN

- Explain Plan `explain_plan` pro dotaz, který vybírá počet fotek, které patří do daného albumu. Dotaz se provede tak, že se nejprve zpracuje operace SELECT. Jelikož dotaz obsahuje agregační funkci COUNT, v dalším kroku se provede GROUP BY, protože kromě počtu fotografií vybíráme taky název albumu. V následujících krocích se přistoupí k tabulkám album a fotografie a provede se jejich spojení. GROUP BY je velmi náročná operace, a proto je vhodné tuto operaci zjednodušit INDEXem.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		4	312	7 (15)	00:00:01
1	HASH GROUP BY		4	312	7 (15)	00:00:01
* 2	HASH JOIN		4	312	6 (0)	00:00:01
3	TABLE ACCESS FULL	ALBUM	2	130	3 (0)	00:00:01
4	TABLE ACCESS FULL	FOTOGRAFIE	4	52	3 (0)	00:00:01

- Explain Plan `explain_plan_fast` je rychlejší verze. Dotaz jsme urychlili tím, že jsme vytvořili INDEX, který indexuje tabulku album podle názvu. To ulehčí práci při provádění dotazu, jelikož se nebude muset znovu vykonávat operace HASH GROUP BY. NESTED LOOPS je efektivní způsob spojení tabulek. Při porovnávání cen vykonávání jednotlivých operací je možné si všimnout, že jsme došli k našemu tíženému výsledku, nižší ceně.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		4	312	6 (0)	00:00:01
1	SORT GROUP BY NOSORT		4	312	6 (0)	00:00:01
2	NESTED LOOPS		4	312	6 (0)	00:00:01
3	TABLE ACCESS BY INDEX ROWID	ALBUM	2	130	2 (0)	00:00:01
4	INDEX FULL SCAN	POCET_FOTEK	2		1 (0)	00:00:01
* 5	TABLE ACCESS FULL	FOTOGRAFIE	2	26	2 (0)	00:00:01

5.5 Matezializovaný pohled

Materializovaný pohled dokáže urychlit dotazy tím, že uloží fyzickou tabulku danou příkazem SELECT. Pohled `ucastnici_akce_brno` slouží k vyhledávání osob, které se zúčastnili nebo zúčastní akce v městě Brno.

```
1 CREATE MATERIALIZED VIEW ucastnici_akce_brno AS
2   SELECT DISTINCT jmeno, prijmeni
3   FROM uzivatel U, akce A, ucastnici_akce I, misto M
4   WHERE U.id=I.uzivatel AND I.akce=A.id AND A.misto=M.id AND M.mesto='Brno';
5
6 -- Pro aktualizaci pohledu je pouzit prikaz
7 BEGIN
8   DBMS.MVIEW.REFRESH( list => 'ucastnici_akce_brno', method => '?');
9 END;
```