

Spis treści

1	Baza danych	2
1.1	System zarządzania baza danych	2
1.2	Tabele	2
1.2.1	Pomocnicze tabele i pola. Indeksy	3
1.2.2	Listing	3

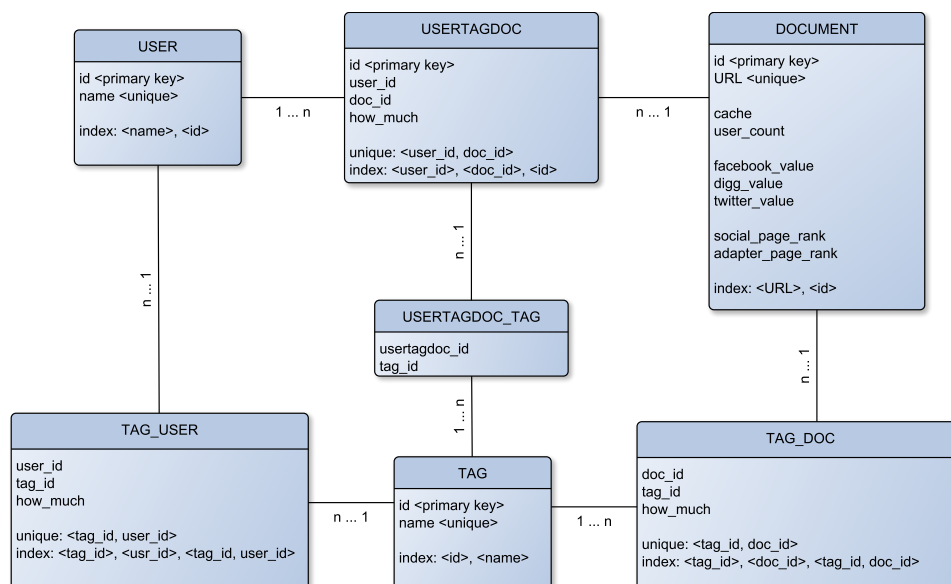
Rozdział 1

Baza danych

1.1 System zarządzania baza danych

Jako serwer bazy danych został użyty MySQL serwer z silnikiem bazy danych InnoDB. MySQL został wybrany głównie z powodu swojej szybkości działania jak również powodu prostoty używania. Dodatkowe funkcjonalności bazy danych rozbudowane w innych systemach zarządzania takich jak: PostgreSQL i Oracle, czyli na przykład: funkcje, triggerzy, nie są używane w tej aplikacji.

1.2 Tabele



Rysunek 1.1: Schemat bazy danych

Baza danych składa się z trzech głównych tabel: USER, DOCUMENT i TAG. Zawierają one informację na temat, odpowiednio, użytkowników, dokumentów i adnotacji pobrane z serwisu delicious. W bazie danych znajdują się też tabele: USERTAGDOC i USERTAGDOC_TAG, które służą do zapisania relacji między użytkownikami a dokumentami (USERTAGDOC) i adnotacjami ich opisującymi (USERTAGDOC_TAG). W tych tabelach zapisane są informacje na temat tego czy dany użytkownik dodał dokument do serwisu i jakimi tagami została dana strona opisana.

1.2.1 Pomocnicze tabele i pola. Indeksy

Dodatkowo w bazie danych (rys 1.1) znajdują się dwie tabele: TAG_DOC i TAG_USR. W tabelach tych zapisywane są dane wyliczone z pozostałych tabel. W tabeli TAG_DOC znajdują się informacje na temat tego ile użytkowników dodało dany dokument doc_i i opisało go tagiem tag_k . Odpowiednia w tabeli TAG_USR znajdują się informacje na temat ilości różnych dokumentów dodanych przez użytkownika usr_n i opisanych tagiem tag_m . Dane ilości różnych tagów którymi użytkownik usr_l opisał dokument doc_j przechowywane są w już istniejącej tabeli USERTAGDOC.

Tabele TAG_DOC, TAG_USR i pole `how_much` w tabeli USERTAGDOC zostają wypełnione w czasie preprocessingu. Dane te posłużą później do utworzenia macierzy, używanych przez algorytmy Social PageRank i Adapted PageRank.

W tabelach na różnych polach zostały dodane indeksy. Przyspieszają one działanie aplikacji, pozwalają na szybsze operacje przy często używanych polach. Główne indeksy zostały zaznaczone na rysunku 1.1

1.2.2 Listing

Poniżej znajduje się listing zapytania SQL tworzącego tabele w bazie danych.

Listing 1.1: Skrypt tworzący tabele w bazie danych

```
CREATE TABLE 'DOCUMENT' (
  'id' bigint(20) NOT NULL AUTO_INCREMENT,
  'url' varchar(255) NOT NULL,
  'digg_value' int(11) DEFAULT '0',
  'facebook_value' int(11) DEFAULT '0',
  'twitter_value' int(11) DEFAULT '0',
  'page_fetch' tinyint(1) DEFAULT '0',
  'tag_count' bigint(20) DEFAULT NULL,
  'user_count' bigint(20) DEFAULT NULL,
```

```

    'cache' text,
    'adapted_page_rank' double DEFAULT NULL,
    'social_page_rank' double DEFAULT NULL,
    PRIMARY KEY ('id'),
    UNIQUE KEY 'url' ('url')
)

CREATE TABLE 'TAG' (
    'id' bigint(20) NOT NULL AUTO_INCREMENT,
    'doc_count' bigint(20) DEFAULT NULL,
    'doc_dist_count' bigint(20) DEFAULT NULL,
    'tag' varchar(255) NOT NULL,
    'user_count' bigint(20) DEFAULT NULL,
    'adapted_page_rank' double DEFAULT NULL,
    PRIMARY KEY ('id'),
    UNIQUE KEY 'tag' ('tag')
)

CREATE TABLE 'TAG_DOC' (
    'id' bigint(20) NOT NULL AUTO_INCREMENT,
    'doc_id' bigint(20) DEFAULT NULL,
    'tag_id' bigint(20) DEFAULT NULL,
    'how_much' int(11) DEFAULT '1',
    PRIMARY KEY ('id'),
    UNIQUE KEY 'tag_doc' ('doc_id', 'tag_id'),
    KEY 'doc_id' ('doc_id', 'tag_id'),
    KEY 'tag_doc_doc' ('doc_id'),
    KEY 'tag_doc_tag' ('tag_id')
)

CREATE TABLE 'TAG_USR' (
    'id' bigint(20) NOT NULL AUTO_INCREMENT,
    'user_id' bigint(20) DEFAULT NULL,
    'tag_id' bigint(20) DEFAULT NULL,
    'how_much' int(11) DEFAULT '1',
    PRIMARY KEY ('id'),
    UNIQUE KEY 'tag_user' ('user_id', 'tag_id'),
    KEY 'user_id' ('user_id', 'tag_id'),
    KEY 'tag_usr_doc' ('tag_id'),
    KEY 'tag_usr_usr' ('user_id')
)

CREATE TABLE 'USER' (
    'id' bigint(20) NOT NULL AUTO_INCREMENT,

```

```

    'doc_count' bigint(20) DEFAULT NULL,
    'name' varchar(255) DEFAULT NULL,
    'new_data' tinyint(1) DEFAULT '1',
    'tag_count' bigint(20) DEFAULT NULL,
    'tag_dist_count' bigint(20) DEFAULT NULL,
    'adapted_page_rank' double DEFAULT NULL,
    PRIMARY KEY ('id'),
    UNIQUE KEY 'name' ('name')
)

```

```

CREATE TABLE 'USERTAGDOC' (
    'id' bigint(20) NOT NULL AUTOINCREMENT,
    'doc_id' bigint(20) DEFAULT NULL,
    'user_id' bigint(20) DEFAULT NULL,
    'how_much' int(11) DEFAULT NULL,
    PRIMARY KEY ('id'),
    UNIQUE KEY 'user_id' ('user_id','doc_id'),
    KEY 'FKB30EFAE96714BC07' ('doc_id'),
    KEY 'FKB30EFAE9520DD4E4' ('user_id'),
    CONSTRAINT 'FKB30EFAE9520DD4E4' FOREIGN KEY ('user_id') REFERENCES
    'user' ('id'),
    CONSTRAINT 'FKB30EFAE96714BC07' FOREIGN KEY ('doc_id') REFERENCES
    'document' ('id')
)

```

```

CREATE TABLE 'USERTAGDOC.TAG' (
    'USERTAGDOC_id' bigint(20) NOT NULL,
    'tags_id' bigint(20) NOT NULL,
    KEY 'FK4C01D124CA702D03' ('tags_id'),
    KEY 'FK4C01D1243D83CB28' ('USERTAGDOC_id'),
    CONSTRAINT 'FK4C01D1243D83CB28' FOREIGN KEY ('USERTAGDOC_id')
REFERENCES 'usertagdoc' ('id'),
    CONSTRAINT 'FK4C01D124CA702D03' FOREIGN KEY ('tags_id') REFERENCES
    'tag' ('id')
)

```
