**Introduction to Database Technology TMA**

Mindaugas Pranaitis

Mprana01

Jerry Smallwood

12.06.2019

Table of Contents

[1.Introduction 3](#_Toc11235307)

[2. ER Diagram 3](#_Toc11235308)

[3. Database Table description 4](#_Toc11235309)

[4. Data types 5](#_Toc11235310)

[5. Relationships 7](#_Toc11235311)

[6.Summary 8](#_Toc11235312)

[7.Reference 9](#_Toc11235313)

**See-It-Snap-It-Store-It Database Design**

# 1.Introduction

See-It-Snap-It-Store-It is an app that will help people to devise their photographs. Relational database has been used to systemize information about user’s photographs such as location, image quality and also organize user images in to albums.

Database requirements has been represented in the entity relationship diagram and the table description following summary of datatypes used and relationship implemented. Data types I have chosen with an idea that database will scale up.

# 2. ER Diagram

What’s is ERD? It’s a diagram which graphically represents entities and their relationships with each other. An entity (object or a thing) is represented as rectangular in diagram and relationships denoted using Crow’s foot - . Refer to section 4 to find out about relationships.

Person

Album

Location

Photo

Personphotos

Photoalbums

Figure ER Diagram

INT

# 

# 3. Database Table description

The database table is where all data in the database is stored. Data in tables arranged in columns and rows. In my database (Figure 2) we see 6 tables. Each table has column, which holds single piece of data for the subject of the record. In contrast, row (record) will hold all the data about the subject. Looking at the Figure 2, Album table we can identify 3 columns: album\_id, description and image, all together will be row or record.

../dtphoto_tables.pdf

Figure Table description

# 4. Data types

MySQL supports the SQL standard integer types INTEGER/INT and SMALLINT. As an extension to the standard, MYSQL also supports the integer types MEDIUMINT and BIGINT. Figure 3 shows the requires storage and range for each integer type. Mysqlcom. 2019.Numeric Data types [Online].

| **Type** | **Storage (Bytes)** | **Minimum Value Signed** | **Minimum Value Unsigned** | **Maximum Value Signed** | **Maximum Value Unsigned** |
| --- | --- | --- | --- | --- | --- |
| TINYINT | 1 | -128 | 0 | 127 | 255 |
| SMALLINT | 2 | -32768 | 0 | 32767 | 65535 |
| MEDIUMINT | 3 | -8388608 | 0 | 8388607 | 16777215 |
| INT | 4 | -2147483648 | 0 | 2147483647 | 4294967295 |
| BIGINT | 8 | -263 | 0 | 263-1 | 264-1 |

Figure 3 MySQL supported Integer types

I have used the INT type in my database Album table, column album\_id, which is primary key (Figure 2). Primary key always need to be unique, therefore most accurate data type would be INT, which also lets us to ensure that all possible rows will be stored without limitations and headless calculations could be performed (imagine counting one and twos). Now SMALLINT would give better performance and use less space in memory, however if Albums number exceeds 32767 data storage issues will arise.

Next data type supported by SQL are String types (see Figure 4). Most commonly used are CHAR AND VARCHAR. Both are similar but differ in the way they stored and retrieved. Mysqlcom. String data types 2019 [Online].

In my database, Person table, column person name I have used VARCHAR versus CHAR due to CHAR length limitations and VARCHAR abilities to have variable length.

| **Data Type** | **Storage Required** |
| --- | --- |
| CHAR(***M***) | The compact family of InnoDB row formats optimize storage for variable-length character sets. See [COMPACT Row Format Storage Characteristics](https://dev.mysql.com/doc/refman/8.0/en/innodb-row-format.html#innodb-compact-row-format-characteristics). Otherwise, ***M*** × ***b***ytes, <= ***M*** <= 255, where ***w*** is the number of bytes required for the maximum-length character in the character set. |
| BINARY(***M***) | ***M*** bytes, 0 <= ***M*** <= 255 |
| VARCHAR(***M***), VARBINARY(***M***) | ***L*** + 1 bytes if column values require 0 − 255 bytes, ***L*** + 2 bytes if values may require more than 255 bytes |
| [TINYBLOB](https://dev.mysql.com/doc/refman/8.0/en/blob.html), [TINYTEXT](https://dev.mysql.com/doc/refman/8.0/en/blob.html) | ***L*** + 1 bytes, where ***L*** < 28 |
| [BLOB](https://dev.mysql.com/doc/refman/8.0/en/blob.html), [TEXT](https://dev.mysql.com/doc/refman/8.0/en/blob.html) | ***L*** + 2 bytes, where ***L*** < 216 |
| [MEDIUMBLOB](https://dev.mysql.com/doc/refman/8.0/en/blob.html), [MEDIUMTEXT](https://dev.mysql.com/doc/refman/8.0/en/blob.html) | ***L*** + 3 bytes, where ***L*** < 224 |
| [LONGBLOB](https://dev.mysql.com/doc/refman/8.0/en/blob.html), [LONGTEXT](https://dev.mysql.com/doc/refman/8.0/en/blob.html) | ***L*** + 4 bytes, where ***L*** < 232 |
| ENUM('***value1***','***value2***',...) | 1 or 2 bytes, depending on the number of enumeration values (65,535 values maximum) |
| SET('***value1***','***value2***',...) | 1, 2, 3, 4, or 8 bytes, depending on the number of set members (64 members maximum) |

Figure . String type storage requirements

In addition to numeric and string data types SQL supports date and time types(Figure5) Mysqlcom. 2019. String data types. Looking at the Figure 2 table Photo, column photo\_date I used DATE type. Now DATE will grant me an easy way to store and manipulate data, hence DATETIME provides precision of a second, however is not relevant for my database, knowing that user need standard date.

| **Data Type** | **“Zero” Value** |
| --- | --- |
| [DATE](https://dev.mysql.com/doc/refman/5.7/en/datetime.html) | '0000-00-00' |
| [TIME](https://dev.mysql.com/doc/refman/5.7/en/time.html) | '00:00:00' |
| [DATETIME](https://dev.mysql.com/doc/refman/5.7/en/datetime.html) | '0000-00-00 00:00:00' |
| [TIMESTAMP](https://dev.mysql.com/doc/refman/5.7/en/datetime.html) | '0000-00-00 00:00:00' |
| [YEAR](https://dev.mysql.com/doc/refman/5.7/en/year.html) | 0000 |

Figure . Date and Time types

To summarize, each data type serves different purpose in database and has its advantages and disadvantages. Choosing right data is very important.

Fixed numeric data type like SMALLINT might perform faster than INT when manipulating data, because fixed size allocated in memory for fixed data type, therefore uses less space, but database scalability will be lost comparing to INT.

# 5. Relationships

According to (Database guide, 2019) there are 3 types relationships in relational database:

* One-to-one. Not used in my database.
* One-to-many.
* Many-to-many.

To identify relationships used in my database we need to refer to Figure 1. There are 6 entities in total. Starting with Album entity, we see crows foot pointing (shows many side of a relationship) to Photoalbums from it and from Photo entity. Since photos can be in many albums and many albums can have many pictures (many-to-may), I have created linking table to avoid data de-normalization. In addition to it, two foreign keys were created (See figure 2, Photoalbums table) which references to Album table, column album\_id and Photo table photo\_id.

Same approach applies to Person, Personphotos and photo tables. Linking table Personphotos was created with two foreign keys (Figure 2, Personphotos table). Each foreign key references their table primary key.

Looking at Location entity in ER diagram, we see crows foot pointing to Photo. This type of relationship called one-to-many and in ER diagram, means that one location can have many pictures and definitely not many-to-many. In addition, foreign key photo\_location (see Figure 2, table Photo table) was created to reference table Location, location\_id.

# 6.Summary

Documenting database is beneficial not only for business owners, but also staff who will work with it. Imagine, new database administrator joins your company. Without documentation, it might take long time for individual to understand approaches taken for DB implementation. More than that, database ERD (Entity relationship diagram) and Table description gives logical view of the system, so non-technical personal can understand concepts.

# 7.Reference

1.Mysqlcom. 2019. [Online]. Numeric data types [11 June 2019]. Available from: <https://dev.mysql.com/doc/refman/5.7/en/integer-types.html>

2. Mysqlcom. 2019. String data types. [Online]. [11 June 2019]. Available from: <https://dev.mysql.com/doc/refman/5.7/en/char.html>

3. Mysqlcom. 2019. String data types. [Online]. [11 June 2019]. Available from: https:/dev.mysql.com/doc/refman/5.7/en/date-and-time-types.htm

4. Database guide. 2019. The 3 Types of Relationships in Database Design. [Online]. [11 June 2019]. Available from: https://database.guide/the-3-types-of-relationships-in-database-design/)