Ronja Larsson
Final Assignment
SQL and databases

Documentation for Ronja's Freediving Database Design

Background
I am currently in Tenerife for freediving training. It is my biggest passion in life, and I have wanted to spend some time dedicated to this activity. I stay in a dive center, and I was inspired to create a database for the dive center for this assignment.

The Database Design
There are 10 tables in the database and most of them are in some way connected, as shown in graph *2_database_design_diagram* (see separate .png file). There are several 1-to-many relationships, two many-to-many relationships and one one-to-one relationship. I will go through each table now with my motivation for my choice of structure. In the diagram you find an illustration of how different tables are connected and what data types of each attribute has.

- *Table 1 Education Systems*

Table 1 contains information about different freediving education systems. The entities here are education systems, such as PADI, SSI and other organizations that certify freediving educations. The primary key in the table is auto-generated education system id, and the other attribute is the name of the education system. Table 1 is one of the parent tables for two other tables.

- *Table 2 Instructors*

Table 2 contains information about the instructors stored in my database. The entities here are freediving instructors. The primary key here is auto-generated instructor id. The other attributes include first name, last name, birthdate, e-mail and nationality. Table 2 is one of the parent tables for two other tables.

- *Table 3 Education Systems and Instructors*

Table 1 and table 2 have many-to-many relationship, which means one instructor can teach for several education systems and one education system can be taught by many different instructors. To facilitate for the unlimited possibility here, I created an intermediate table to connect table 1 and table 2. The primary key here is auto-generated esi id. The foreign keys here are instructor id from table 2 and education system id from table 1. An instructor can be registered with several different education systems and the esi id is the key that differentiates each instructor and the connected education systems.

- *Table 4 Level*

In freediving, most education systems give out courses in three levels. Each level has different requirements of how deep the diving student should dive to, in order to be certified. In this table, the primary key, level, is entered manually by humans and the other attributes include level description, requirements for the diving depth of four freediving disciplines: FIM, CWT, DNY and STA. Table 4 is a parent table for another table.

Happy new year ☺

- *Table 5 Freedivers*

Table 5 contains the information about the freedivers in my database. The entities here are freedivers. The primary key is auto-generated freediver id. The other attributes include first name, last name, birthdate, e-mail and nationality. Table five is a parent tables for three other tables.

- *Table 6 Certification*

Table 6 is a child table with four parent tables: table 1, 3, 4 and 5. Table 6 contains information about certifications of a freediver in table 5. A freediver can have several certifications from different levels and from different education systems. An instructor can issue many certifications. On the other hand, a freediving certification with a certification number can only be linked to one issuing instructor and one freediver the certification is issued to. Similarly, each certification can only contain information about one level and from one education system. A freediver who has completed three levels of courses will obtain three certifications with three different certification id and number, with information specific to the certification. The attributes here are certification number which is the primary key, four foreign keys; freediver id, instructor id, education system id and level, as well as issue date and certification id. Certification table basically links the education information about a freediver with the instructor, level and education system. This is done to achieve the third normal form, where each table only contains the information about the entity.

- *Table 7 Personal Best*

Table 7 contains information about personal records of a freediver on four freediving disciplines. It has a one-to-one relationship with table 5 and is direct linked to table five with the same id; freediver_id in table 5 should match freediver in table 7. Please see my code in .sql file *1_create_db_freedive* to see the code for linking these two tables with the needful constrains.

- *Table 8 Product*

Table 8 contains information about the products the dive center sells. The entities here are products for freediving, such as fins and snorkels. The primary key in the table is auto-generated product id, and the other attribute is the product brand, product name and price in SEK. Table 8 is one of the parent tables for one other table.

- *Table 9 Order*

Table 9 contains information about the purchase orders in my database. The entities here are purchase orders. The primary key here is auto-generated iorder id. The other attributes include order date and a foreign key freediver id that links the freediver placing the purchase order. Table 9 is one of the parent tables for one other table.

- *Table 10 Order and Product*

Table 8 and table 9 have many-to-many relationship, which means one purchase order can contain many different products and a kind of product can be bought by several customers. To facilitate for the unlimited possibility here, I created an intermediate table to connect table 8 and table 9. The primary key here is auto-generated transaction id. The foreign keys here are order id from table 9 and product id from table 8.

Happy new year ☺

Ronja Larsson
Final Assignment
SQL and databases

The design stage of the database took the most time. I first drew out different tables which will be in the database and drew the relationships between different tables. I want to achieve the third normalization so I further divided the tables into different tables. This is the reason why I have 10 tables instead of 4.

Once the relationships between the tables are established in the diagram, I started by coding the database structure with CREATE statements. The code for creating the database is saved in one file, *1_create_db_freedive*, this is done purposefully so that I will always have the structure of the database before any data is entered. The tables with primary keys are created before these primary keys will be coded as the foreign keys in other tables.

In my design, I organized the order of each table according to their parent and child relationships:

- Table 1 and 2 are the parent tables to table 3
- Table 1, 2, 4 and 5 are the parent tables to table 6
- Table 7 is connected with table 6 in a one-to-one relationship
- Table 8 and 9 are the parent tables to table 10

Tables:
1. Education systems
2. Instructors
3. Education systems and instructors
4. Level
5. Freedivers
6. Certification
7. Personal best
8. Product
9. Order
10. Order and product

This is also the exact same order when I insert the data with INSERT statements. The errors can be in this way avoided in the coding process. I only put the minimum number of rows required, which is five, because I determined that the amount of data is sufficient for the purpose of this assignment. The code of inserting the data in the tables is saved in file *2_insert_data_freedive*. This is to achieve the most basic required code in my database, to facilitate for the data manipulations.

The third SQL file, *3_query_data*, is the file with at least five queries that altogether cover CRUD operations and JOIN statements, nested queries, two math functions of my choice. To give the grade VG an attempt, in this file, you will also find queries with GROUP BY and @variables. In this documentation, I do not go into details about my code. I have written clear comments in my SQL files where I walk through my examinator on what I want to do and why I am doing it.

I hope my documentation, together with the comments in my code files, give you a clear picture on the motivation behind my choices when creating the database as well as the queries.

Happy new year ☺