

YR - SEM 1 - BEV DATABASE - ASSIGNMENT 2

by Jericho Jermaine Uzzell

FILE	12014_JERICO_JERMAINE_UZZELL_YR_-_SEM_1_-_BEV_DATABASE_-_ASSIGNMENT_2_315925_161044202.DOCX (135.3K)		
TIME SUBMITTED	14-JAN-2021 07:07PM (UTC+0000)	WORD COUNT	1829
SUBMISSION ID	142135215	CHARACTER COUNT	10055

University of Wales Trinity Saint David
School of Computing
Databases and Application Development
ACCA5026
Jericho Uzzell
1702201
09/01/2021



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Abstract

This document will assess Smile Factory requirements for a new system, this report will then develop a database, how the database tables are connected and potential issues from the new design, this could be missing tables or tables not thought out correctly.

1. Introduction; [page 2]
2. Database creation; [page 3-6]
 - 2.1. Requirements for creating the database and tables; [page 3]
 - 2.2. Database created; [4-5]
 - 2.2.1. Administration; [page 4]
 - 2.2.2. Surgeons; [page 4]
 - 2.2.3. Surgeons address; [page 4]
 - 2.2.4. Surgeons email; [page 4]
 - 2.2.5. Customers; [page 4]
 - 2.2.6. Customers address; [page 4]
 - 2.2.7. Treatment; [page 5]
 - 2.2.8. Treatment type; [page 5]
 - 2.2.9. Transactions; [page 5]
 - 2.3. Entity relationship diagram; [page 6]
3. Transactions; [page 6]
4. Tuning; [page 7]
5. Reference; [page 7]
6. Bibliography; [page 7]

1. Introduction

This document will create database, entity relationship diagram (ERD) and ... for a dental practice known as Smile Factory. This document will go over the requirements for making databases, tables and things to consider when these are put into practice – how exactly the database tables react with other tables, why you should label foreign keys (FK) and why you should have their relationships stated.

2. Database creation

2.1. Requirements for creating the database and tables

According to [1], they state that tables that have second normal form (2nf) are dependent on the primary key (PK), for example, columns for age and date of birth (DOB) can cause issue where the data is inconsistent.

This is where third normal form (3nf) should be used. According to [1], they state that for a table to follow 3nf, the table must already conform to 2nf and the columns within it are non-transitively dependent on the primary key.

What does this mean in practice? If you have column within your table that could be transferred to their own table for consistency, you should. An example of this could be, if you had a phone line within a company that told others your extension number, should this be jotted down on your personal record?

For this example, as it is one to one database relationship, you could get away with it. However, this table could have other links for line management extensions, floor details, department details etc. You essentially lose that ability to add that detail later on.

tbl_staff			
staffid	firstname	surname	extension
55889	John	Doe	009
55890	Barry	John	002
55891	Jessica	Hughes	003

tbl_staff			
staffid	firstname	surname	extensionid
55889	John	Doe	2

tbl_extension			
extensionid	Manager Extension	Dept	extension
2	15	IT	009

The two images above, created by the student demonstrates how you can get more information about the column by making it a foreign key (FK) to another column without adding more columns to the staff table. This eliminates inconsistency between the staff and the extension database.

2.2. Databases created

2.2.1. Administrators

Administrators would be the main overseer of the surgeons and customers. They are mainly receptionist who would add customers to the database and a few receptionists would be administrators to add, remove and edit surgeons.

The table does not have nor does it require to have any links with such as FK to the customers and surgeons' tables. As for logging who creates customers and surgeons, an audit log would be a benefit.

2.2.2. Surgeons

The surgeons within the dental practice have customers they oversee and preform check-ups and other types of treatment on customers. The surgeon could request treatment for a certain date and time, which will be inputted by the reception staff. If treatment is carried out on the same date, the treatment table would be amended.

2.2.5. Surgeon addresses

This table is linked with the surgeon table, the surgeon identification field on both identify the address for who it is. It is a one-to-many table, meaning there can only be one surgeon to many addresses. The reason why it is one-to-many is, you could have two address however, an argument against this is that there should only be one address as it can cause confusion. This is a thing to consider.

2.2.6. Surgeon emails

As is with the surgeon addresses table, this table has a one-to-many design. It has a column where the surgeon identification can be stated (to link back to the surgeon table) and the email addresses can have more than one entity. The same concern can be raised reference emails, that the surgeon should only have one email linked with their account and that any more than one can cause confusion.

2.2.3. Customers

Customers table has an FK for surgeon identification as one surgeon can be and are commonly primary surgeon for a patient. This table consists of personal name, date of birth (DOB) and when they joined the surgery. The DOB is used to assistant whether the individual will pay or not for certain types of treatment.

For this database, Smile Factory stated that if the customer is under a certain age, treatment is free.

2.2.4. Customer addresses

It is imperative to have addresses for customers as letters for appointments and other information about the surgery can be received. This table has a one-to-many design, where the customer identification is used to link back to the customers table. As previously stated, having this one-to-many design can cause confusion.

2.2.7. Treatment

The treatment table consists of three FKs, these being customer identification, surgeon identification and treatment type. You might be wondering why exactly this table has a surgeon identification and that is simply due to the fact that a surgeon separate from the primary can perform treatment if the primary surgeon is not available or simply cannot for other reasons.

On the customers profile, having a primary surgeon listed can give information to receptionists and others, without guessing who might be that customers surgeon – within the application, if no surgeon is listed on the treatment report then the customers primary surgeon is selected.

Treatment type is an FK as it would cause inconsistencies with current or future treatment if the staff are allowed to enter the treatment type – having this table concatenates a lot more information.

Other information the treatment table holds is the time it was added, the time this treatment is for and the treatment details / notes. The time the treatment is for can be the same day if treatment had been performed without an appointment.

A note to mention, this table should have been named appointments and a separate table for completed treatments added to it as at the moment, the name can cause confusion.

2.2.8. Treatment types

Treatment types table is a one-to-many for the treatments table. The data stored in the table are the name and the charge for treatment. This table can have many entities, with different options and charges. Charges range from zero (free) to whatever the price is deemed.

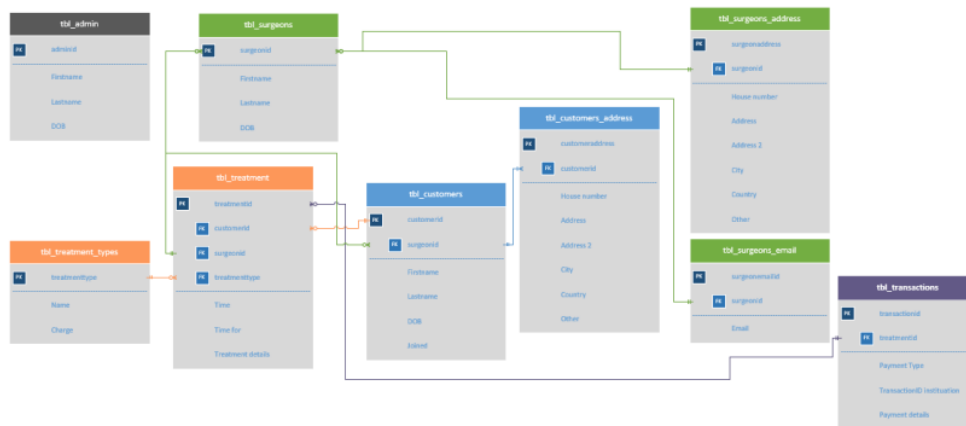
2.2.9. Transactions

The transaction table has an FK to the treatment table. This is because the transaction charge can track easily if administrators are doing checks. This table (transaction) has a one-to-one design, this is so you cannot have duplicate entries and double charges. The table has payment type, transaction identification details and payment details.

A design concern raised is, if you have many treatments on one day, does they get bundled into one or must you pay per treatment?

2.3. Entity relationship diagram (ERD)

The student created an entity relationship diagram for Smile Factory, how the databases link with each other, the foreign keys and more.



Entities are coloured to make it clear if one is linked to another. Each entity FK and PK have been correctly matched if they are in some way linked.

2.4. Most queried tables within Smile Factory

Upon assessment, you can just know that tables such as customers, treatment and surgeons are going to be queried the most which is why a good reason to separate them into other tables is to improve performance. Tables such as surgeons do not have all information about that surgeon on the main table, instead information like address and email are stored separately.

Tuning a database, which we will talk more about later, however tuning does not only involve how well an SQL query performs (how efficiently) and also how the database tables are connected.

3. Transactions

Transactions are imperative to systems running databases. If you are updating a database, running a task which has not got a zero change of messing up, then transactions should be used. They are intended to be simple things a database administrator can do which allows for starting a change history, ability to commit a change(s) and the ability to revert change(s).

[2] States that a transaction can run multiple lines of code, a transaction must be atomic, consistent, isolated and durable. Atomic is defined as: transaction that is logical and work with all of its data modifications or none of them preformed; consistent: once data is finished it must be left in a consistent state; isolated: the modified data performed by the transaction must be independent else, the outcome from the transaction may be erroneous.

Lastly, durable: once a transaction is completed, the effects of it must be permanent to the system; [2] further states that the acronym is known as ACID.

4. Tuning

SQL Tuning according to [3] is ensuring SQL code that an application will use will run as fast as possible. [3] States that as is driving home, there are many different routes for you to choose and that every so often a faster one can be found. That is SQL tuning, finding an answer to a question. According to [4], the ways to tune an SQL statement is based off statistics.

5. Conclusion

From the above designs chosen, some things such as the email address and addresses table have had concerns raised regarding confusion over the ability to have multiple entries from one PK. A solution should be sought, one being to only allow for one entry to be entered (each time) or two, a column defining whether if that entry is the primary.

The basic structure is there however, certain things such as payment to staff table not in original design, confusion over the treatment table not being named appointments. It is unknown if multiple treatments can be carried out for one day, whether if it all can be bundled into one sum for the customer or if they are separated.

The way in which the database is connect however, in the sense that PK in one table can link in with an FK within another, the tracking on how tables link with each other and the separation of key columns such as emails being separated into different tables for faster query times.

6. References

- [1] K.Wenzel, *essentialsql*, "Database Third Normal Form Explained in Simple English", 2014, [Online], Available: <https://www.essentialsql.com/get-ready-to-learn-sql-11-database-third-normal-form-explained-in-simple-english/>, [Accessed: 09/01/2020]
- [2] *weresource*, "SQL Transaction", 2020 February 26, [Online], Available: [https://www.w3resource.com/sql/controlling-transactions.php#:~:text=A%20transaction%20is%20a%20sequence,\(undone%20from%20the%20database\).&text=End%20Transaction](https://www.w3resource.com/sql/controlling-transactions.php#:~:text=A%20transaction%20is%20a%20sequence,(undone%20from%20the%20database).&text=End%20Transaction), [Accessed: 09/01/2020]
- [3] *searchoacle*, "What is SQL tuning?", April 2001 18, [Online], Available: <https://searchoracle.techtarget.com/answer/What-is-SQL-tuning#:~:text=SQL%20tuning%20is%20the%20process,in%20the%20fastest%20possible%20time.&text=Tuning%20SQL%20statements%20is%20finding,route%20is%20not%20very%20intuitive>, [Accessed: 09/01/2020]
- [4] *synametrics*, "Top 10 performance tuning tips for relational databases", 2020, [Online], Available: <https://www.synametrics.com/SynametricsWebApp/WPTop10Tips.jsp>, [Accessed: 09/01/2020]

7. Bibliography

- <https://stackoverflow.com/questions/10153648/correct-use-of-transactions-in-sql-server>
- <https://docs.microsoft.com/en-us/sql/t-sql/language-elements/begin-transaction-transact-sql?view=sql-server-ver15>
- <https://www.withinweb.com/info/sql-transactions-in-phpmyadmin/>

YR - SEM 1 - BEV DATABASE - ASSIGNMENT 2

GRADEMARK REPORT

FINAL GRADE

47 / 100

GENERAL COMMENTS

Instructor

Your report was clear but when you are referencing please state the authors name if you are going to say "X stated that". You have not suggested or created and performance tuning enhancements. Your ERD is appropriate but you do not require the admin table it is not clear where or why you are using it. Be careful of your relationship lines they can change the way in which we store the data.

PAGE 1

PAGE 2

PAGE 3



Comment 1

state the name of the author that you are referring to

PAGE 4

PAGE 5

PAGE 6

PAGE 7