

report

by Silvia Stoianova Settle

FILE	5780_SILVIA_STOIANOVA_SETTLE_REPORT_315925_1948508344.DOCX (455.13K)		
TIME SUBMITTED	06-FEB-2021 09:50PM (UTC+0000)	WORD COUNT	3270
SUBMISSION ID	144318656	CHARACTER COUNT	17361

Beverly Holland Module ACCA5026

Databases and Application Development

The Smile Factory Case Study

Silvia Settle, 1708386

Contents

Database Creation.....	2
Background	2
Project Aims	2
Context.....	2
Requirements for the system:	2
Design.....	3
Database ERD Diagram	3
Table Design with datatypes and constraints	3
Scope of the project and solution.....	9
Scope.....	9
Constraints and data integrity	10
Inserting data	11
Performance Tuning.....	12
Indexes	12
Triggers.....	12
Security Considerations	12
Backup and Restore	13
Distributed Database Solution	13
Suitable Fragmentation Schema	13
Sources:.....	15

Database Creation

Background

Project Aims

To create an application to allow the organisation and booking of dental services. Different type of dental treatments will require different appointment duration and depending on the status of the patient (NHS or private), different payment will be due. The system should allow for invoicing to be produced and revenue calculated. The system is to be used by the receptionist/admin staff. extension option could be the possibility of linking the database to a front-desk online service to allow the patient to book appointments online and dentists to receive communication by email.

Context

The Smile Factory has 4 dentists, 6 dental hygienists and 6 dental nurses. It offers 4 morning **surgeries** (rooms) and 2 afternoon surgeries. The surgery is open Monday to Friday. Morning surgeries are 8.30am till 11.30am and afternoon surgeries are 2pm till 5pm.

The patients make appointments for treatment. Dentists, hygienists and nurses must be scheduled into a surgery slot. Appointments are min of 15 mins long and are determined by the dental treatment. Patients are issued invoices for treatment and who pays for this treatment depends on the patient status: NHS or private. Private patients are charged.

Requirements for the system:

- It must record the invoices issued and payment made in order to keep track of the income generated from the appointments.
- It will automatically print out a schedule for each dentist
- Will print out invoices and keep record of the invoices.
- It should be able to keep a record of the income generated from patients' appointments.
- It should be able to produce a query for patients seeking appointments (views, filtering through status)
- It will print out reports on time schedules.
- It will print out reports on current booking levels of rooms and dental surgery staff.

Design

Database ERD Diagram

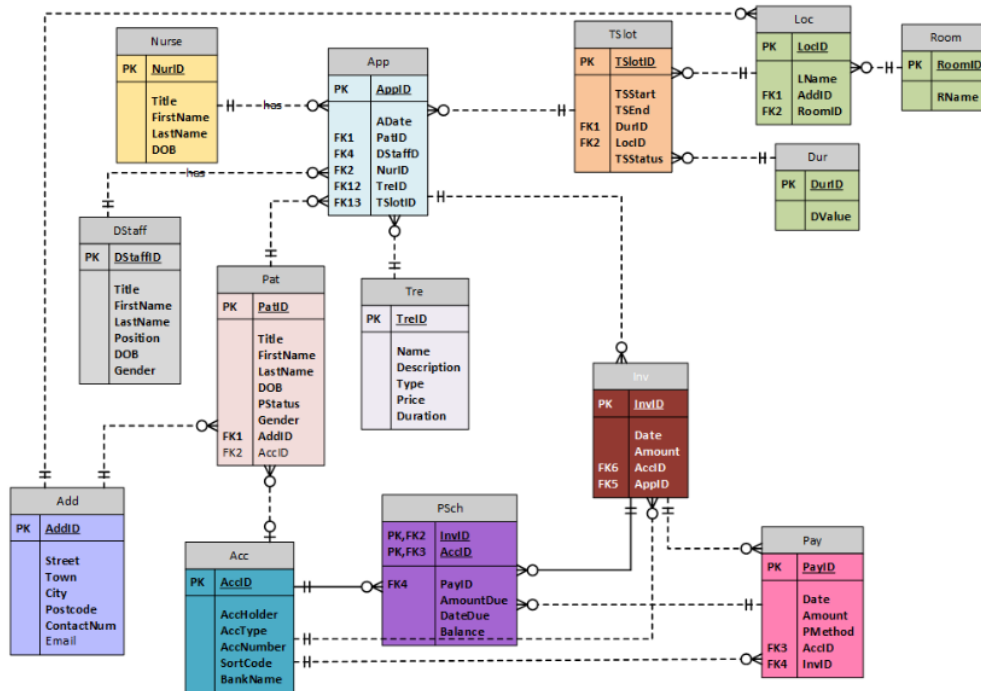


Table Design with datatypes and constraints

Acc=Accounts Receivable

Table Name	Acc		
Fields	Data Type	Constraints	Sample Data
AccID	int IDENTITY(100000,1) NOT NULL	PK_AccID	100001 100002
AccHolder	varchar(30) NOT NULL	UNIQUE	John Taylor, NHS
AccType	varchar(10) NOT NULL	CHK_Acc_AccType (NHS or PRP) Default_Acc_AccType (PRP)	NHS or PRP

AccNumber	varchar(8) NOT NULL		1002510 or 22245321
SortCode	varchar(8) NOT NULL		55-25-25
BankName	varchar(30) NOT NULL		Natwest

Add=Address

Table Name	Add		
Fields	Data Type	Constraints	Sample data
AddID	int IDENTITY(1,1) NOT NULL	PK_AddID	1, 55, 14
Street	varchar(30) NOT NULL		23 Meadow Lane
Town	varchar(30) NOT NULL		Clydach
City	varchar(20) NOT NULL		Swansea
Postcode	varchar(8) NOT NULL		SA88RR
ContactNum	varchar(12) NOT NULL		01792544123
Email	varchar(40) NULL		silly_socks@gmail.com

App=Appointments

Table Name	App		
Fields	Data Type	Constraints	Sample Data
AppID	int IDENTITY(1,1) NOT NULL	PK_AppID	1005, 5488
ADate	datetime NOT NULL	CHK_App_ADate (cannot be in the past)	12/05/2004
<i>PatID</i>	int NOT NULL	FK_App_Pat_PatID	100001, 511115
<i>DStaffID</i>	varchar(4) NOT NULL	FK_App_DStaff_DStaffID	DS05, DS08
<i>NurID</i>	varchar(4) NOT NULL	FK_App_Nur_NurID	NU08, NU01
<i>Tried</i>	varchar(4) NOT NULL	FK_App_Tre_TreID	TR01, TR04
<i>TSlotID</i>	varchar(4) NOT NULL	FK_App_TSlot_TSlotID	TS01

TSlot = TimeSlot

Table Name	TSlot		
Fields	Data Type	Constraints	Sample Data
<u>TSlotID</u>	varchar(4) NOT NULL	PK_TSlotID	TS01-TS24
<u>TSSStart</u>	datetime NOT NULL		2021-04-14 8:00
<u>TSEnd</u>	datetime NOT NULL		2021-04-14 8:30
<u>DurID</u>	varchar(1) NOT NULL	FK_TSlot_Dur_DurID	A, B, or C
<u>LocID</u>	varchar(4) NOT NULL	FK_TSlot_Loc_LocID	SU01
TSSStatus	varchar(1) NOT NULL	CHK_TSlot_TSSStatus (A or B)	A, B

Dur - Duration

Table Name	Dur		
Fields	Data Type	Constraints	Sample Data
<u>DurID</u>	varchar(1) NOT NULL	PK_DurID	A, B, C
<u>DValue</u>	int NOT NULL		15, 30, 45

Room

Table Name	Room		
Fields	Data Type	Constraints	Sample Data
<u>RoomID</u>	varchar(2)	PK_DurID	R1, R2
<u>RName</u>	varchar(10)		Room 1 – Room 3

Nurse

Table Name	Nurse		
Fields	Data Type	Constraints	Sample Data
<u>NurID</u>	varchar(4) NOT NULL	PK_NurID	NU01 to NU06
Title	varchar(10) NOT NULL	CHK_Nur_Title (Mr or Mrs or Ms or Miss)	Mr, Ms
Firstname	varchar(30) NOT NULL		Helena

LastName	varchar(50) NOT NULL		Carter
DOB	date NOT NULL	CHK_Nur_Date (cannot be in the future)	1954-05-14

Dstaff = Dental Staff (Dentists and Dental Hygienists)

Table Name	DStaff		
Fields	Data Type	Constraints	
DStaffID	varchar(4) NOT NULL	PK_DStaffID	DS01 – DS10
Title	varchar(10) NOT NULL	CHK_DStaff_Title (Mr or Mrs or Ms or Miss or Dr or Prof)	Dr, Prof
FirstName	varchar(30) NOT NULL		Julian
LastName	varchar(50) NOT NULL		Watson
Position	varchar(2) NOT NULL	CHK_DStaff_Position (DS or DH)	DS (dentist), DH (dental hygienist)
DOB	date NOT NULL	CHK_DStaff_DOB (cannot be in the future)	1956-12-15
Gender	varchar(1) NOT NULL	CHK_DStaff_Gender (M or F)	M, F

Inv = Invoices

Table Name	Inv		
Fields	Data Type	Constraints	Sample Data
InvID	int IDENTITY(1,1) NOT NULL	PK_InvID	1, 56
Date	datetime NOT NULL	Default_Inv_Date (current date and time)	2019-04-17
Amount	decimal(10,2) NOT NULL		
<i>AccID</i>	int NOT NULL	FK_Inv_Acc_AccID	100002, 854141

<i>AppID</i>	int NOT NULL	<i>FK_Inv_App_AppID</i>	2, 59
--------------	--------------	-------------------------	-------

Loc = Location/Surgeries

Table Name	Loc		
Fields	Data Type	Constraints	Sample Data
<u>LocID</u>	varchar(4) NOT NULL	PK_LocID	SU01-SU05
LName	varchar(30) NOT NULL		Alexandra Road Surgery, Wern Surgery
<i>AddID</i>	int NOT NULL	<i>FK_Loc_Add_AddID</i>	1, 2
<i>RoomIS</i>	varchar(2)	<i>FK_Loc_Room_RoomID</i>	R1, R2, R3

Pat = Patients

Table Name	Pat		
Fields	Data Type	Constraints	Sample Data
<u>PatID</u>	int IDENTITY(100000,1) NOT NULL	PK_PatID	100000-999999
Title	varchar(5) NOT NULL	CHK_Pat_Title (Mr or Mrs or Ms or Miss or Master or Dr or Prof)	Mr, Mrs
FirstName	varchar(30) NOT NULL		John
LastName	varchar(40) NOT NULL		Talyor
DOB	date NOT NULL	CHK_Pat_DOB (cannot be in the future)	2000-05-12
PStatus	varchar(3) NOT NULL	CHK_Pat_PStatus (NHS or PRP) Default_Pat_PStatus (NHS)	NHS, PRP
Gender	varchar(1) NOT NULL	CHK_Pat_Gender (M or F)	F, M
<i>AddID</i>	int NOT NULL	<i>FK_Pat_Add_AddID</i>	1,2
<i>AccID</i>	int NULL	<i>FK_Pat_Acc_AccID</i>	100004

Pay = Payments

Table Name	Pay		
Fields	Data Type	Constraints	Sample Data

<u>PayID</u>	int IDENTITY(1,1) NOT NULL	PK_PayID	1, 58, 21
Date	date NOT NULL	Default_Pay_Date (today's date)	2020-01-01
Amount	decimal(10,2) NOT NULL		45.00
PMethod	varchar(2) NOT NULL	CHK_Pay_PMethod (DD or CD)	CD, DD
<i>InvID</i>	int NOT NULL	FK_Pay_Inv_InvID	14, 2
<i>AccID</i>	int NOT NULL	FK_Pay_Acc_AccID	8,12,5

PaySch = Payments Schedule

Table Name	PSch		
Fields	Data Type	Constraints	Sample Data
<i>InvID</i>	int NOT NULL	PK_PaySchID FK_PaySch_Inv_InvID	4,89
<i>AccID</i>	int NOT NULL	PK_PaySchID FK_PaySch_Acc_AccID	100001
<i>PayID</i>	int NOT NULL	FK_PaySch_Pay_PayID	1
AmountDue	decimal(10,2) NOT NULL		£45
DateDue	date NOT NULL		01/02/2012
Balance	decimal(10,2) NOT NULL		£45

Tre = Treatments/Dental Procedures

Table Name	Tre		
Fields	Data Type	Constraints	
<u>TreID</u>	varchar(4) NOT NULL	PK_TreID	TR01-TR04
Name	varchar(20) NOT NULL	CHK_Tre_Name (Band 1, Band 2, Band 3), preventative	Band 1, 2, 3 or preventative
Description	varchar(50) NOT NULL		Scale& Polish
Type	varchar(2) NOT NULL	CHK_Tre_Type (PC or HM)	PC (preventative care) or HM (health-management)
Price	decimal(10, 2) NOT NULL		Band A
Duration	int NOT NULL		15

Scope of the project and solution

Scope

Staff Table

The database is only dealing with the type of staff involved in the treatment. Although we would expect that a surgery would have other members of staff such as administrators, managers, accountants, these are left out of the scope of this project. The project is also not dealing with the employment and financial side of staff management.

Treatment Table

The project is focusing only on the services ordinarily provided by the NHS-i.e. preventative and health-maintenance dentalcare ([Health in Wales | Find an NHS Dentist](#)). For convenience, the solution's emphasis is on the revenue this treatment brings and therefore the detail of different types of treatment is left out. The project is not going to record detailed information on patients' treatment and history, and as a result the treatment was grouped into bands (band1, 2 and 3 according to the fee they incur as well as the time required – 15, 30 and 45 min respectively). Another presumption is that the dental hygienists are qualified to complete dental procedures from Band 1. For convenience, the dental hygienist treatments (polish and scaling) were grouped within band 1 too, both from the point of view of time required for treatment and the fee.

Specialist dental treatments which is classed as cosmetic by the NHS ([Health in Wales | Find an NHS Dentist](#)) is left out of the scope of the project as it is not usually within the list of NHS treatments.

Surgery

This is interpreted as two sites with several rooms offering treatment in the morning and/or afternoon. One of the surgeries is open only in the morning.

Patients

For this solution the presumption was made that the Smile Factory staff will determine the type of patients when their information is entered into the database. Although in real life there is a list of criteria that trigger the exemption from treatment – such as age (under 18 and over...), means tested (low income and benefits), pregnant (during pregnancy and for a year after), the project will not investigate this area. If the patient does not meet the criteria for exemption and is classed as “private”, a payment account is created that will hold their DD details. The system of payment is focused on the private patients only, while the NHS ones are only tracked to the

point of invoicing, where the presumption is that further processes are followed by the business to obtain the fees due for NHS treatment.

Constraints and data integrity

To ensure data integrity several measures were introduced into the database:

Tables and primary keys

The nurse, the dental staff, the treatment, and the location tables hold data that is relatively stable, and it is not very likely to change very often. For example, we expect the members of staff and the location to stay the same for a long time. This is the reason why the data type for the primary keys in those tables was chosen to be varchar and in a format that makes them look more like natural keys that will make sense to the humans looking at the data in the tables. For example, SU01 stands for Surgery 1 – the Alexandra Road Surgery, DS01 is the dentist Dr Joanne Jones, DH01 is the dental hygienist Dahlia Davadjikan and NU01 is the nurse Rosie Romford. The Dentists and the dental hygienists were grouped into a common table because of the presumption that they can both offer treatment from Band 1, and as such, their appointment allocation can follow the same process. The combination of different characters 'DH' for dental hygienist and 'DS' for a dentist makes it easy to distinguish what type of surgery staff they are - a dentist or a dental hygienist. The decision which one will be assigned for an appointment will depend on the treatment type required and using the primary keys in this format allows to distinguish which is which. It is possible to split these two in separate tables, however the decision was made to keep the dental staff in one table for convenience. The dental staff has a gender column for situations where the patient may prefer to be treated by a specialist from a specific gender, however this was omitted from the nurses table as their numbers limit the options available.

The data in the patients table is more prone to change in terms of a patient can change status, address, name, title, or address. The number of records is also expected to increase over time, therefore the primary key was chosen to be 'int IDENTITY' to make the process of key allocation automatic. Similarly, the appointments, the appointments schedule, the invoices, the payments, and the private patients' accounts tables hold more dynamic data so 'int IDENTITY' was chosen as the data type for their primary key too. For example, the combination of dates, timeslots and availability in the appointments schedule table varies throughout the lifespan of the system.

The accounts table contains a field for account name. It was decided that this is going to be a field to hold one account name that combines the first and last name of the account holder to ensure uniqueness. The UNIQUE constraint was added to the field to ensure this and to prevent

situations where different patients with the same first and last name try to open an account and be considered by the system as already having one. The business can decide how they deal with situations like this, i.e. whether to introduce a middle initial or add an integer at the end of the account name to ensure uniqueness.

The Pay Schedule table has a composite/compound key having the foreign key columns from the invoices and the accounts tables. This decision was made to ensure that the combinations of invoices and payees accounts is unique. The design for the payments table used a surrogate key in contrast for ease of use at the stage of tracking payments and calculating income.

Any long-term considerations for the system operation were left out of the scope of the current design due to prioritising requirements and considering time and resources constraints. For example, scenarios where a patient dies or moves away and is not able to complete the direct debit (DD) payments for their treatment were not considered within the parameters of this solution. Similarly, another example of a rejected consideration was the use of a foreign key constraint 'ON DELETE' for the addresses table to allow for situations where if we delete a record from the patients, the dental staff, or the nurses table, we will want to delete the address associated with this table as well. This was not implemented due to not being considered critical to the current operational functionality of the system.

CHECK Constraints

Accounts table – account type has a default value of PHP. This was design with a view to be able to hold the details for NHS accounts that are used to pay for treatments for NHS patients. Later on, the design was changed, and the account was limited to hold private patients bank account details only. The feature was left in case of changes in the business requirements in the future.

Naming conventions

General naming conventions were used in the design and the implementation of the project. Hungarian notation as used for the naming of all constraints (primary keys, foreign keys, default values, check constraints), for example: FK for foreign key, CHK for check constraints. Table and column names used Camel Case, while any SQL key words were in capitals.

Inserting data

Stored Procedures were used to facilitate the process of database management, including DDL and DML commands. For example, the tables creation, insertion of foreign keys and check constraints.

Performance Tuning

Indexes

Indexes were considered for the tables that would be queried most often. Non-clustered Indexes were considered for the two main junction tables: appointments and payments. For example, an index on the date, invoiceNo and amount paid/payment method will allow for quicker retrieval and data aggregation. Indexed fields in the appointments table would allow for faster search on appointments by date, patient, dental staff/nurse, treatment type.



A clustered index was considered for the payment schedule table where larger volumes of data are expected to be stored and the composite key makes it a suitable candidate to organise the data by account number. Indexing was planned but not implemented in the database as it was deemed more appropriate to investigate this at the implementation stage, where the presence of data will make it easier to observe queries behaviour.

Triggers

SQL-triggers were considered for implementation to handle the complex queries at two points of the system – the booking of appointments and the payments. Their purpose is to automatically validate the data being processed in the tables handling the data manipulation, to ensure domain integrity of the business rules at these points and to handle post-processing procedures within the following tables: appointments, invoice, payments, and the payment-schedule tables. These were not implemented as the database was not developed at the advanced stages of operation. These are to be considered carefully and not overused as to not deter the performance of the database queries.



Security Considerations

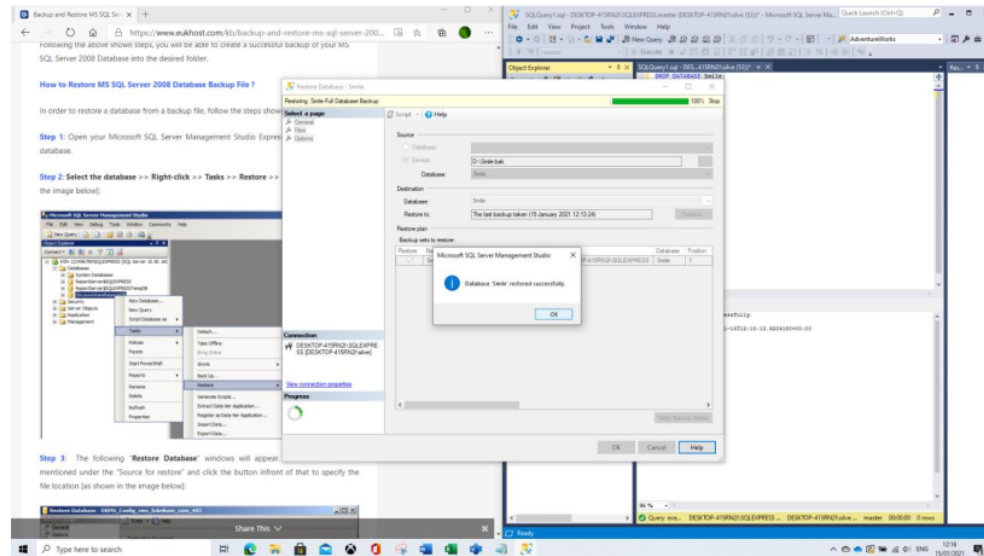
Encrypting Stored Procedures

Stored procedures were used extensively in the design and implementation of the database in order to:

- Improve reusability and makes changes easy to implement and trace;
- Reduce network traffic once the database is operational;
- To allow for code reusability and better maintainability if changes are required during the lifespan of the application.

Backup and Restore

A regular back up on an external device was considered important to ensure the data security as part of the business planned and non-planned contingencies.



Distributed Database Solution

Suitable Fragmentation Schema

The Smile database is operational across two sites with the possibility of new surgeries being opened at other locations. The need for working across several sites in parallel and with the potential of increased volume of data entering the system, the consideration for a distributed database design seems appropriate. Distributed databases are understood as “the combination of autonomous processes that are linked over a computer network” (Tamer Ozsu et al).

At the same time, the context of the current system and the lower level of need for efficient management of the data does eliminate a solution on such magnitude as the Scalable Distributed Data Structures (SDDSs) in which the logical and physical segmentation is fluid and scalable (Litwin and co).

Traditionally, three general approaches to the practical implementation of a database distribution can be taken: horizontal, vertical and hybrid fragmentation, in all of which the tables are split into smaller segments called fragments (Saeed, et al). The choice will depend on the need of the business and the logical organisation of the data. With the vertical fragmentation

the tables can be split into sub-sets by columns, while the horizontal fragmentation investigates sectioning the records within the tables by rows. The hybrid approach is a combination of the other two and seem to offer an opportunity the solution to be tailored to the needs of the client. The latter approach was considered for the Smile Factory. For example, the invoices and payments could be fragmented horizontally by dates or by financial year, while the physical need for the bookings and the payments to be made at different sites suggest that these parts of the database can be further segmented into bookings and payments systems by site. An integral part of the fragmentation process is the use of indexes, which were briefly discussed above, and which allow the practical implementation of database partitioning.

Among the benefits of distributed database systems are better overall database control (including authentication and access rights), semantic integrity control, query optimisation, controlling concurrency, deadlock handling, replication and transaction control, and security (Saeed, et al). All of these areas are crucial for the effective operation of any business, regardless of its type or size. And with view of more and more of the business operations being performed and stored online makes them even more dependent and in demand for distributed data solutions.

Sources:

Litwin, W., S. Sahri, T. Schwarz. Overview of Scalable Distributed Database System SD-SQL Server, Paris-Dauphine University 75016 Paris, France [Scalable Distributed Databases Design \(scu.edu\)](http://www.scu.edu)

NHS website, www.nhs.uk

Saeed K. Rahimi, Frank S. Haug Distributed Database Management Systems: A Practical Approach A John Wiley & Sons, Inc., Publication, 2010

SQL Shack website (How to identify and resolve SQL Server Index Fragmentation (sqlshack.com))

Tamer Ozsu, M., P. Valldurries, Principles of Distributed Database Systems, Springer New York Dordecht Heidelberg London

Tutorial Cup Data fragmentation in DBMS - Data fragmentation - SQL (tutorialcup.com)

Youtube videos by kudvenkat, Bhaskar Jogi, Caleb Curry and Lisa Ballbach

report

GRADEMARK REPORT

FINAL GRADE

83/100

GENERAL COMMENTS

Instructor

A very elegant design in your ERD however you could have placed some screen shots within the report to prove all tables worked correctly. Good analysis of the database tuning requirements.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13



Comment 1

Good analysis



Comment 2

very good considerations of requirements

PAGE 14

