|  |  |
| --- | --- |
| https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcSvOMNhnhHEC_NSs_Hc-XDeg4xTn7I6xau8rdOznQz5hCCji-HwpA  **DATABASE MANAGEMENT SYSTEMS ASSIGNMENT**  December 2014 | **Radu Ciprian Rotaru**  Student ID:29223  https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcTcv07u6XFib0pzxN07Pmm0kC_AhS-ELiu0p7k3h2CXaVfAz2vPeQ |

Contents

[Introduction 8](#_Toc405553113)

[Finding and Analysis 8](#_Toc405553114)

[Tables development and relations (Entities, Attributes and Identifiers) 8](#_Toc405553115)

[‘Patients’ table 8](#_Toc405553116)

[‘NextOfKin’ Table 9](#_Toc405553117)

[‘Medical history’ table 10](#_Toc405553118)

[‘Staff’ table 11](#_Toc405553119)

[‘Roles’ table 12](#_Toc405553120)

[‘Appointments’ table 13](#_Toc405553121)

[‘Appointment\_event’ table 15](#_Toc405553122)

[‘Prescription’ table 16](#_Toc405553123)

[‘Drugs’ table 16](#_Toc405553124)

[‘Hospital\_refferals’ table 17](#_Toc405553125)

[‘Hospitals’ table 17](#_Toc405553126)

[‘Prescribed-drugs’ table 18](#_Toc405553127)

[E-R Diagram 20](#_Toc405553128)

[Triggers 21](#_Toc405553129)

[‘Before insert’ trigger from ‘Appointments’ table 21](#_Toc405553130)

[‘After insert’ trigger from ‘Appointments’ table 23](#_Toc405553131)

[‘Before delete’ trigger from ‘Appointments’ table 23](#_Toc405553132)

[‘Before delete’ trigger from ‘Appointment\_event’ table 23](#_Toc405553133)

[‘Before delete’ trigger from ‘Prescription’ table 24](#_Toc405553134)

[Views 24](#_Toc405553135)

[‘Appointment by patient’ view 24](#_Toc405553136)

[‘Prescription drugs by patient’ 25](#_Toc405553137)

[Security features and advice 27](#_Toc405553138)

[Conclusion 27](#_Toc405553139)

[Bibliography 28](#_Toc405553140)

[Appendix 28](#_Toc405553141)

[Appendix 1 29](#_Toc405553142)

[29](#_Toc405553143)

[Appendix 2 29](#_Toc405553144)

[Appendix 3 29](#_Toc405553145)

[Appendix 4 30](#_Toc405553146)

[30](#_Toc405553147)

[Appendix 5 30](#_Toc405553148)

[30](#_Toc405553149)

[Appendix 6 30](#_Toc405553150)

[Appendix 7 31](#_Toc405553151)

[31](#_Toc405553152)

[Appendix 8 31](#_Toc405553153)

[32](#_Toc405553154)

[Appendix 9 32](#_Toc405553155)

[Appendix 10 32](#_Toc405553156)

[Appendix 11 32](#_Toc405553157)

[33](#_Toc405553158)

[Appendix 12 33](#_Toc405553159)

[33](#_Toc405553160)

[Appendix 13 33](#_Toc405553161)

[34](#_Toc405553162)

[Appendix 14 34](#_Toc405553163)

[34](#_Toc405553164)

[Appendix 15 34](#_Toc405553165)

[35](#_Toc405553166)

[Appendix 16 35](#_Toc405553167)

[Appendix 17 35](#_Toc405553168)

[35](#_Toc405553169)

[Appendix 18 35](#_Toc405553170)

[35](#_Toc405553171)

[Appendix 19 35](#_Toc405553172)

[36](#_Toc405553173)

[Appendix 20 36](#_Toc405553174)

[36](#_Toc405553175)

[Appendix 21 36](#_Toc405553176)

# Introduction

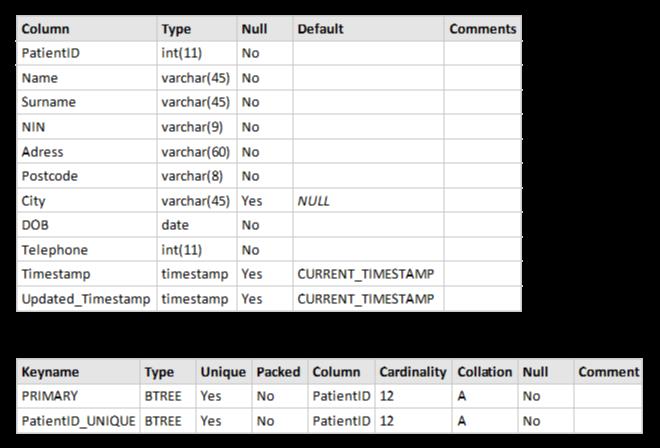
This report will explain step-by-step the database development for a large General Practitioner situated in UK. They have required an information system to help record the data in order to assist management in decision making for both routine and life-threatening ailments. A few tables have been created In order to fulfill their requirements. This database is in 3rd normal form. Each step of the development it will be followed by a code example that has been used. A few user views have been created in order to help the users in their findings. Also triggers have been used to manipulate data more easy. Views and triggers will be discussed more in depth in the content of this report.

# Finding and Analysis

## Tables development and relations (Entities, Attributes and Identifiers)

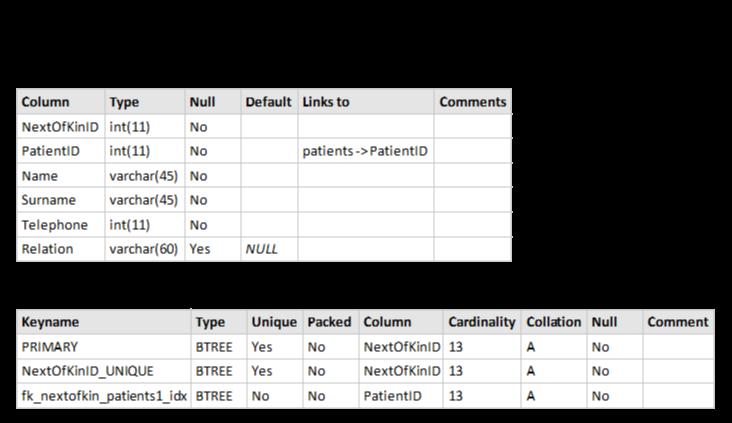
### ‘Patients’ table

The first table created was “Patients”. This table will include all the details needed for the patients. The first entity is ‘PatientID’ which will also be the primary key in this table and it will be Not Null, Unique and also Auto Incremented by the system. The rest of the columns are :’Name’, ‘Surname’, ‘National Insurance Number(NIN)’, ‘Address’, ‘Postcode’,’City’,’Date of birth (DOB)’ and ‘Telephone’. Also this table will include two extra columns named ‘Timestamp’ (which will automatically set the time of the created row) and ‘Updated\_timestamp’ (last time when a specific row has been updated). [Appendix 1](#_Appendix_1)



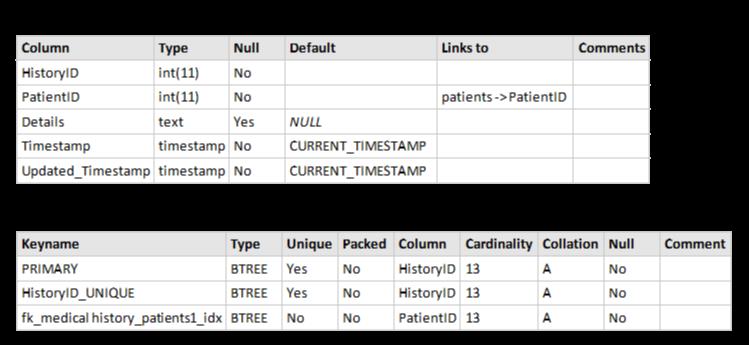
### ‘NextOfKin’ Table

The second table will be ‘NextOfKin’. This table will include for normalization standards of the database, the next of kin persons for each patient and also contact telephone number and the relationship with the patient. The first entity in this table will be ‘NextOfKinID’ which will also be the primary key. It will be set as Not Null, Unique and again, Auto Increment. In this table the second entity it is ‘PatientID’ which will also be the foreign key from the ‘Patients’ table. This foreign key has been used in order to assign to every each patient a next of kin. The rest of the entities are: ‘Name’, ‘Surname’, ‘Telephone’ and ‘Relation’. Those are only meant to store the next of kin’s details without having a big impact on the database[. Appendix 2](#_Appendix_2)



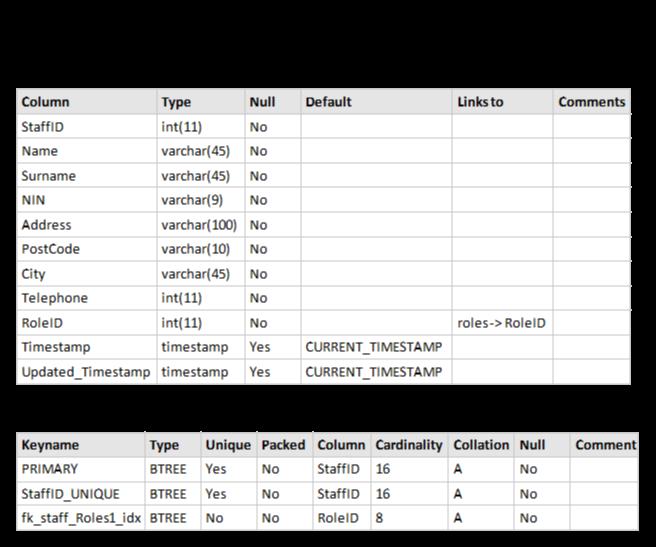
### ‘Medical history’ table

The third table it is ‘Medical history’. Here, again it will be a primary key named ‘HistoryID’ and it will be Not Null, Unique and also Auto Increment. Again it will be a foreign key named ‘PatientID’ which is meant to link a medical history to every patient in the database. In this table it is another entity name ‘Details’ which can be null in case will not be any medical history holded for a patient. Again at the end there are the two columns Timestamp and Updated\_timestamp. [Appendix 3](#_Appendix_3)



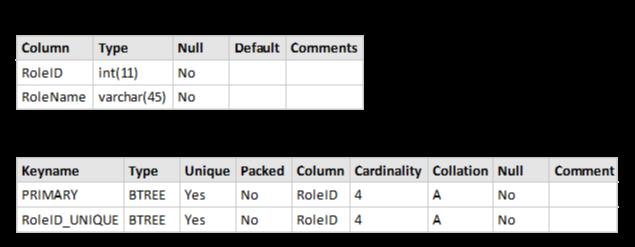
### ‘Staff’ table

The 4th table developed it is the ‘Staff’ table. Here it will be stored all the personnel working in this facility. The primary key it will be ‘StaffID’ which it has also been set as Not Null, Unique and Auto increment. The rest of the fields are: ‘Name’, ‘Surname’, ‘NIN’, ‘Address’, ‘Postcode’, ‘City’ and ‘Telephone’. In this table it is a foreign key named ‘RoleID’ which is imported from the ‘Roles’ table. All the entities have been set as Not Null being very important for the business to hold their staff details. At the end again ‘Timestamp’ and ‘Updated\_timestamp’ are present. [Appendix 4](#_Appendix_4)



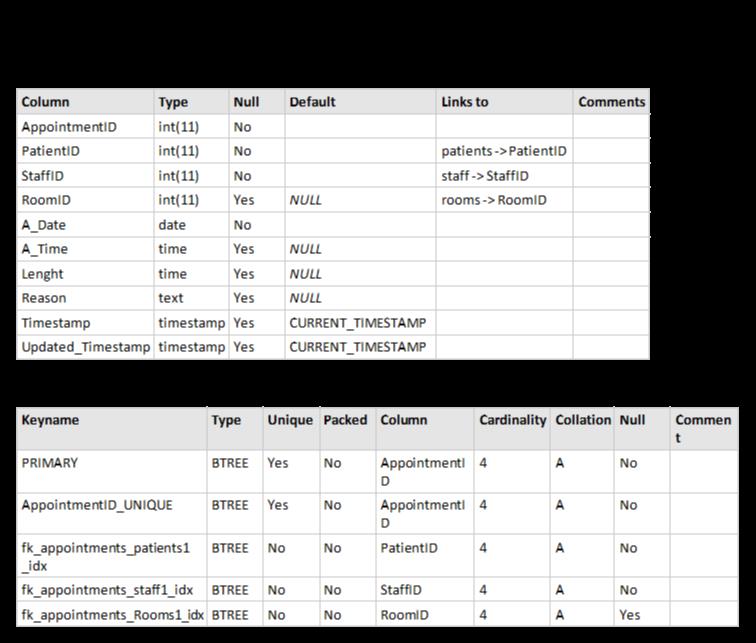
### ‘Roles’ table

The 5th table named ‘Roles’ have been created for normalization purposes and it will hold only the name of roles hold in this facility. A ‘RoleID’ it will be the primary key and also the Not Null, Unique and Auto increment has been assigned. It will help the database administrator to assign to every staff a certain role. [Appendix 5](#_Appendix_5)



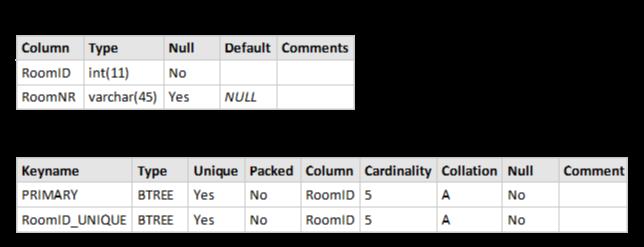
### ‘Appointments’ table

The next table developed was ‘Appointments’. This is a very important table because it holds the trigger for the booking system. Later in this report, this trigger will be discussed more in depth. In this table, ‘AppointmentsID’ has been set up as the primary key and it also be Not Null, Unique and Auto increment. Here will be 3 foreign keys: ‘PatientID’, ‘StaffID’, and ‘RoomID’. Three important entities added here: ‘A\_Date’, ‘A\_Time’ and ‘Length’ which will help in the booking procedure. Also a column ‘Reason has been added which can be Null in case no reason will be added. A ‘Timestamp’ and an ‘Updated\_Timestamp’ have been added. [Appendix 6](#_Appendix_6)



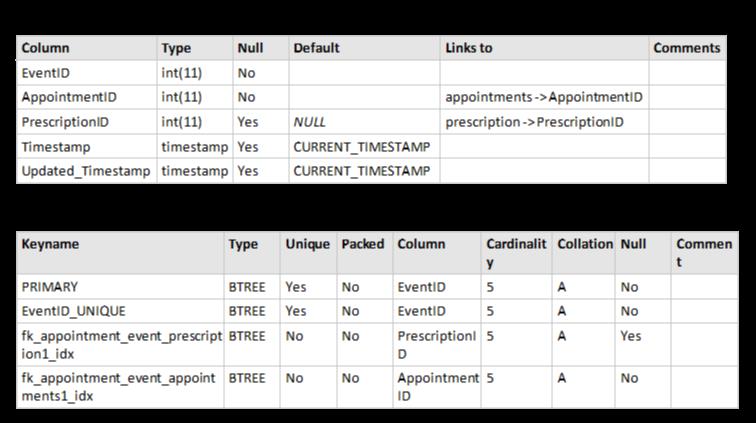
#### ‘Rooms’ table

The 7th table developed it is called ‘Rooms’ and it will have a primary key called ‘RoomID’ and another entity named ‘RoomNR’ which will hold the rooms number hold in the facility. [Appendix 7](#_Appendix_7)



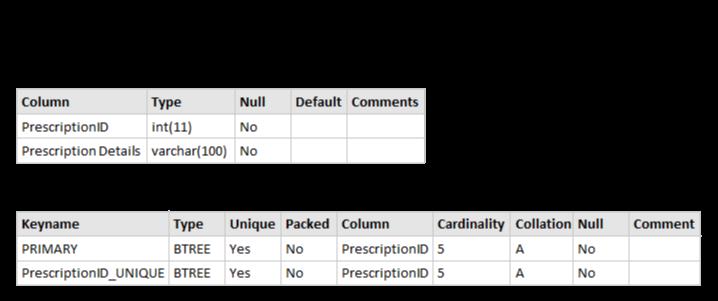
### ‘Appointment\_event’ table

The 8th table created is called ‘Appointment\_event’. This table will have a primary key called ‘EventID’ which will be set as Not Null, Unique and also Auto Increment. Here we can find another two columns set as foreign keys named ‘AppointmentID’ and ‘PrescriptionID’. A trigger used in ‘Appointments’ table will generate automatically an ‘EventID’ in the ‘Appointment\_event’. Again a ‘Timestamp’ and ‘Updated\_timestamp’ have been created for administration purposes. [Appendix 8](#_Appendix_8)



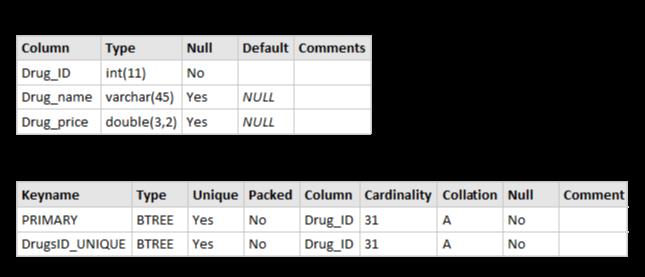
### ‘Prescription’ table

The following table created is called ‘Prescription’. Here, the primary key it will be ‘PrescriptionID’ and it has been set as Not Null, Unique and Auto increment. Another column named ‘Prescription Details’ has been added and it will hold the details about a specific prescription. [Appendix 9](#_Appendix_9)



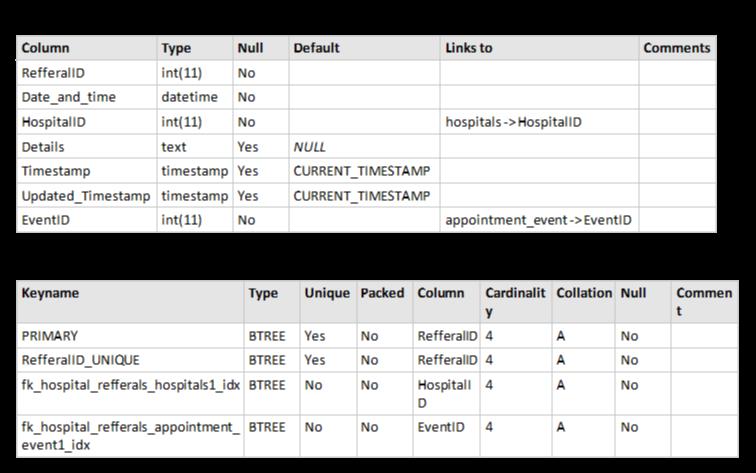
### ‘Drugs’ table

This table will store the name of all medicines and their prices. The primary key here, will be ‘DrugID’, which is set as not null, unique and Autoincrement. [Appendix 10](#_Appendix_10)



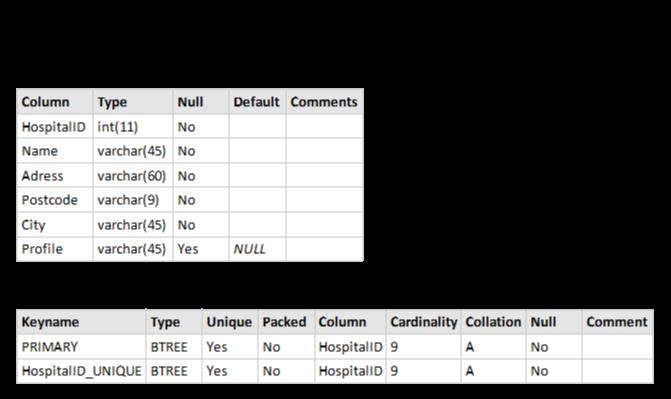
### ‘Hospital\_refferals’ table

‘Hospital\_refferals’ table will store the details for the hospital referrals in case there are any following an appointment event. The primary key here it is ‘RefferalID’ which is set as Not Null, Unique and Auto increment. Foreign here are: ‘EventID’ from ‘Appointment\_event’ table and ‘HospitalID’ from ‘Hospitals’ table. Another entity here are: ‘Date and time’ which will store the date and time of the appointment, without being able to leave it blank, and ‘Details’ in case there are any details to be stored. [Appendix 11](#_Appendix_11)



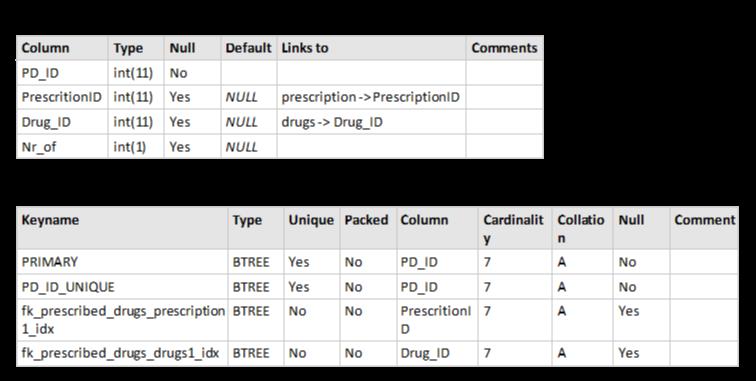
### ‘Hospitals’ table

This table will hold the details for the hospitals like name, address, etc. The primary key in this table is ‘HospitalID’. [Appendix 12](#_Appendix_12)



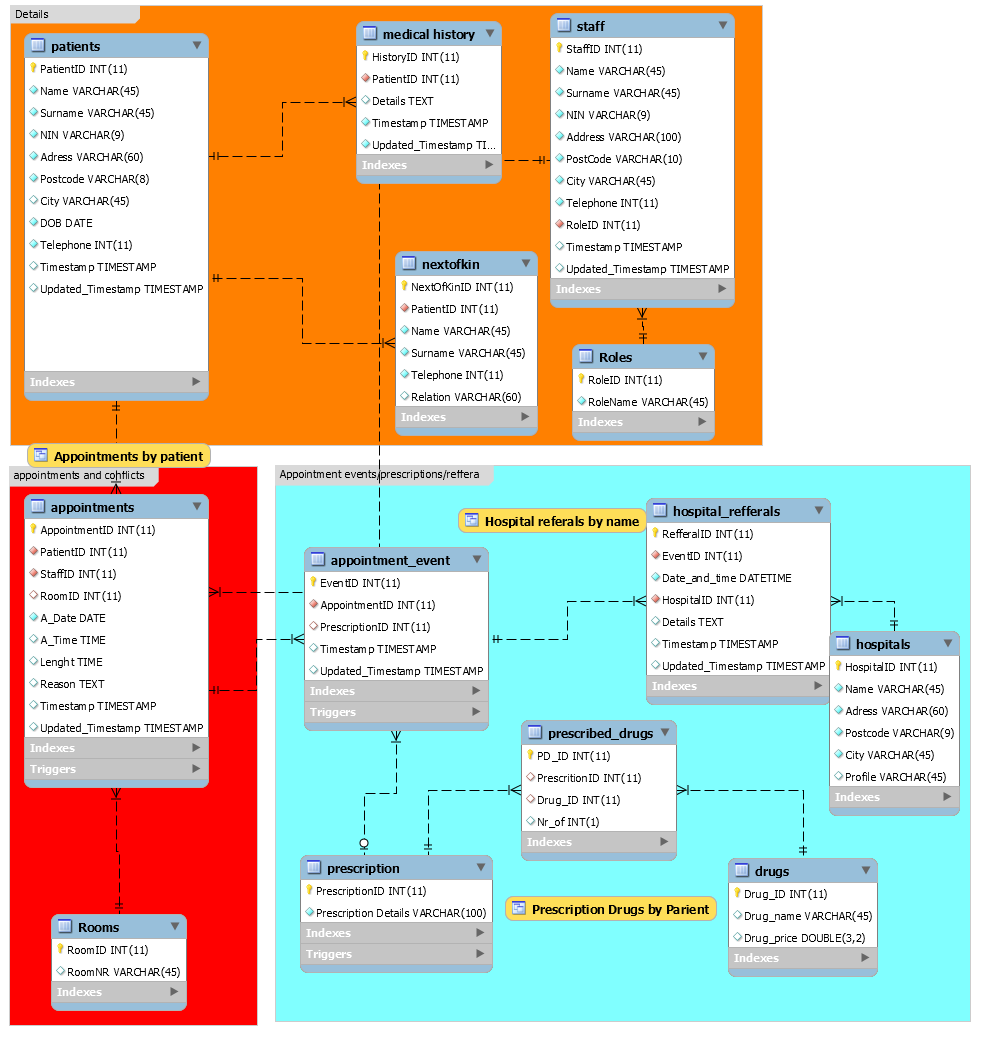
### ‘Prescribed-drugs’ table

In this table it will be stored details about the medicines prescribed after an appointment event. Primary key in here is ‘PD\_ID’ which is set as Not Null, Unique and Auto increment. Foreign keys are ‘PrescriptionID’ and ‘DrugID’. This table will also hold the details about the number of a specific medicine prescribed in a certain prescription. [Appendix 13](#_Appendix_13)



## E-R Diagram

The E-R diagram has been attached in order to show the relationships between the tables and the type of entities each table includes.

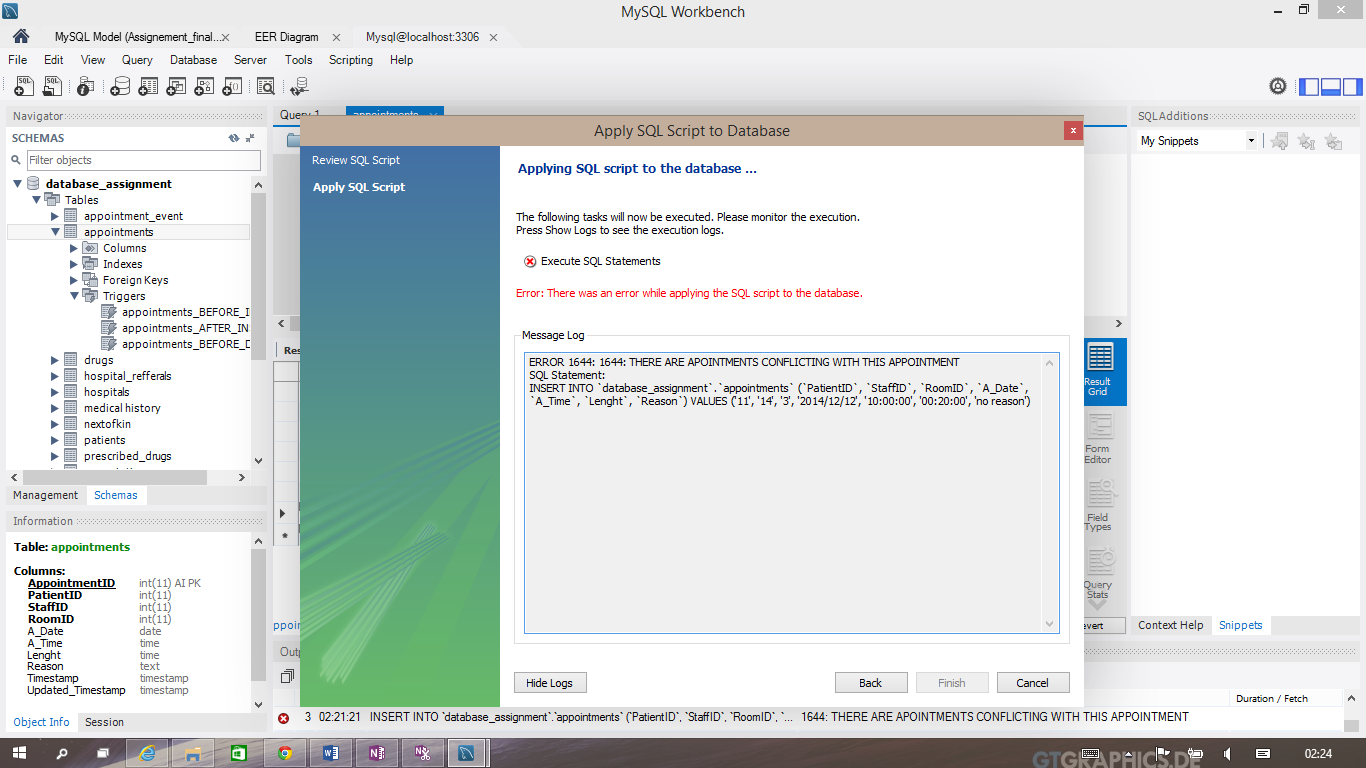


## Triggers

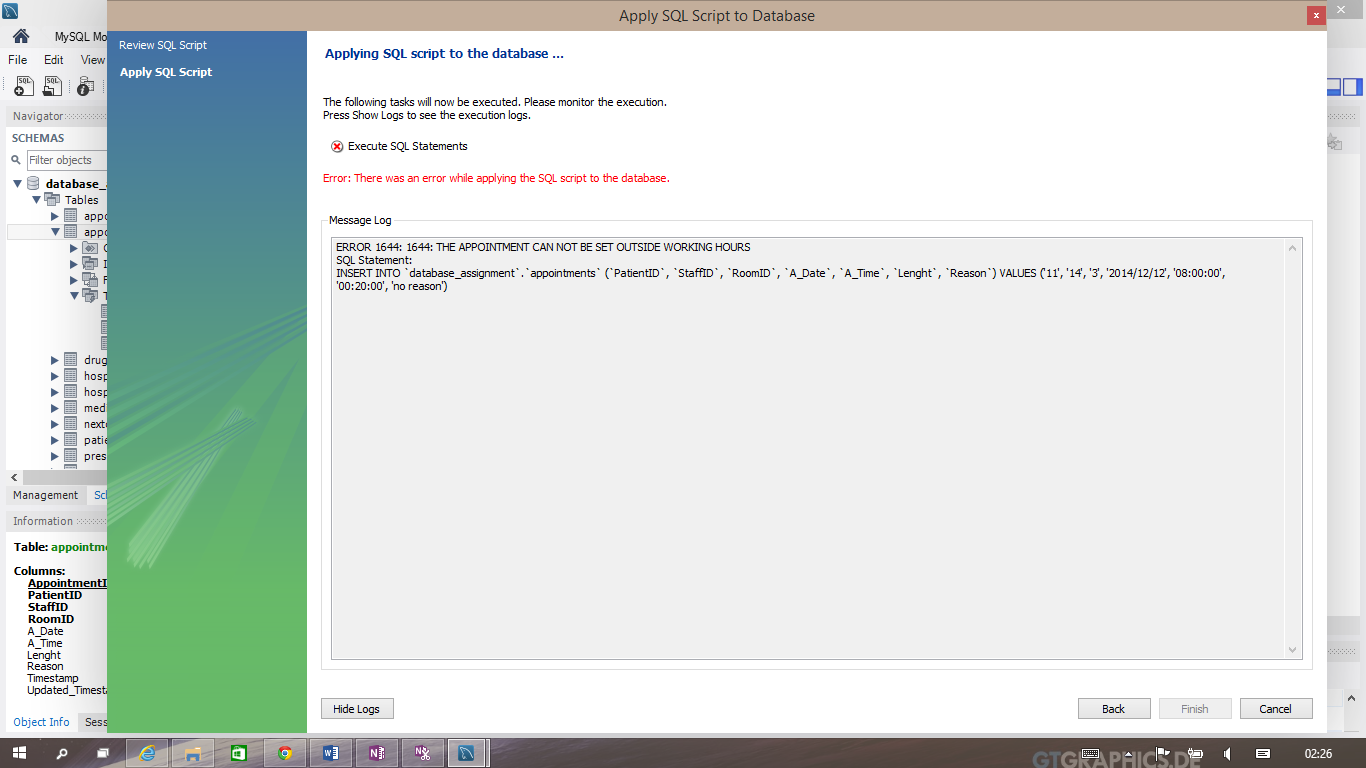
### ‘Before insert’ trigger from ‘Appointments’ table

The first trigger created has been created to add a few rules before booking an appointment. An IF statement has been added in order to check all the requirements specified before inserting the appointment data and time. For example if the receptionist of the GP in case will add at a certain date an appointment, the trigger will help the system to check if the time and length of the appointment it will not overlap with another appointment booked in that time. The trigger will also check the room and the staff booked for the appointment. For example, an appointment can be booked at the same date, time but not in the same room or with the same staff person. If any of those rules are breached then an error message will be displayed: “There are appointments conflicting with this appointment”.

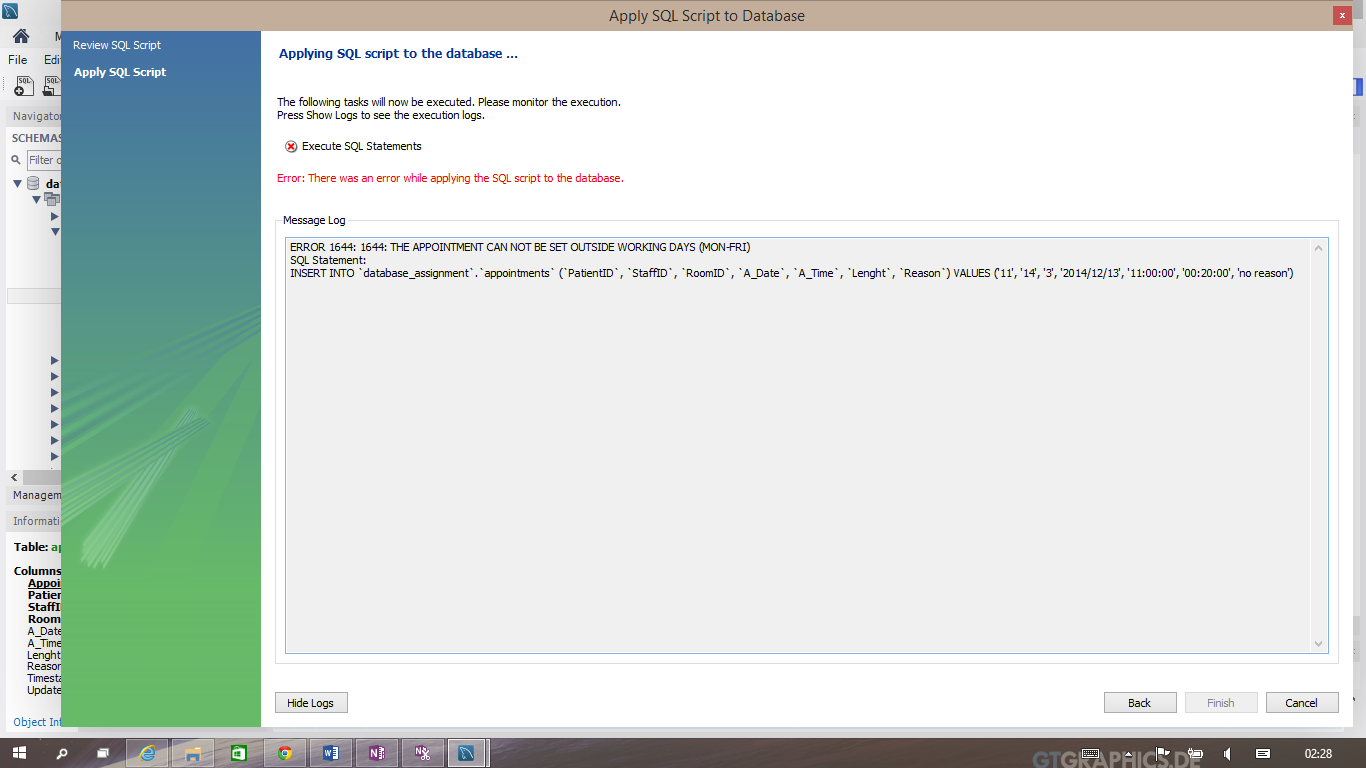
For an example purpose an appointment has been added in 12/12/2014 at 10:00:00 with a length of 00:20:00 in room 3 with staff 15. Another appointment has been tried to be booked on that same date, with a different staff in the same room but at the same time. An error message has been displayed: “There are appointments conflicting with this appointment” [Appendix 14](#_Appendix_14)



Also another appointment has been tried to be created outside the working hours



Or outside the working days



### ‘After insert’ trigger from ‘Appointments’ table

This trigger has been created to automatize the process in the GP. If an appointment will be created the trigger will automatically copy that ‘AppointmentID’ into ‘Appointment\_event’ table which will create a new event ‘EventID’. The conclusion is that an appointment event will be created for every each appointment booked. [Appendix 15](#_Appendix_15)

### ‘Before delete’ trigger from ‘Appointments’ table

In case an appointment will be deleted from different reasons the trigger will also delete the entire row which has been created with the help of precedent trigger. [Appendix 16](#_Appendix_16)

### ‘Before delete’ trigger from ‘Appointment\_event’ table

This trigger will automatically delete a hospital referral in case that referral has been stored into an appointment event, and that event will be deleted. [Appendix 17](#_Appendix_17)

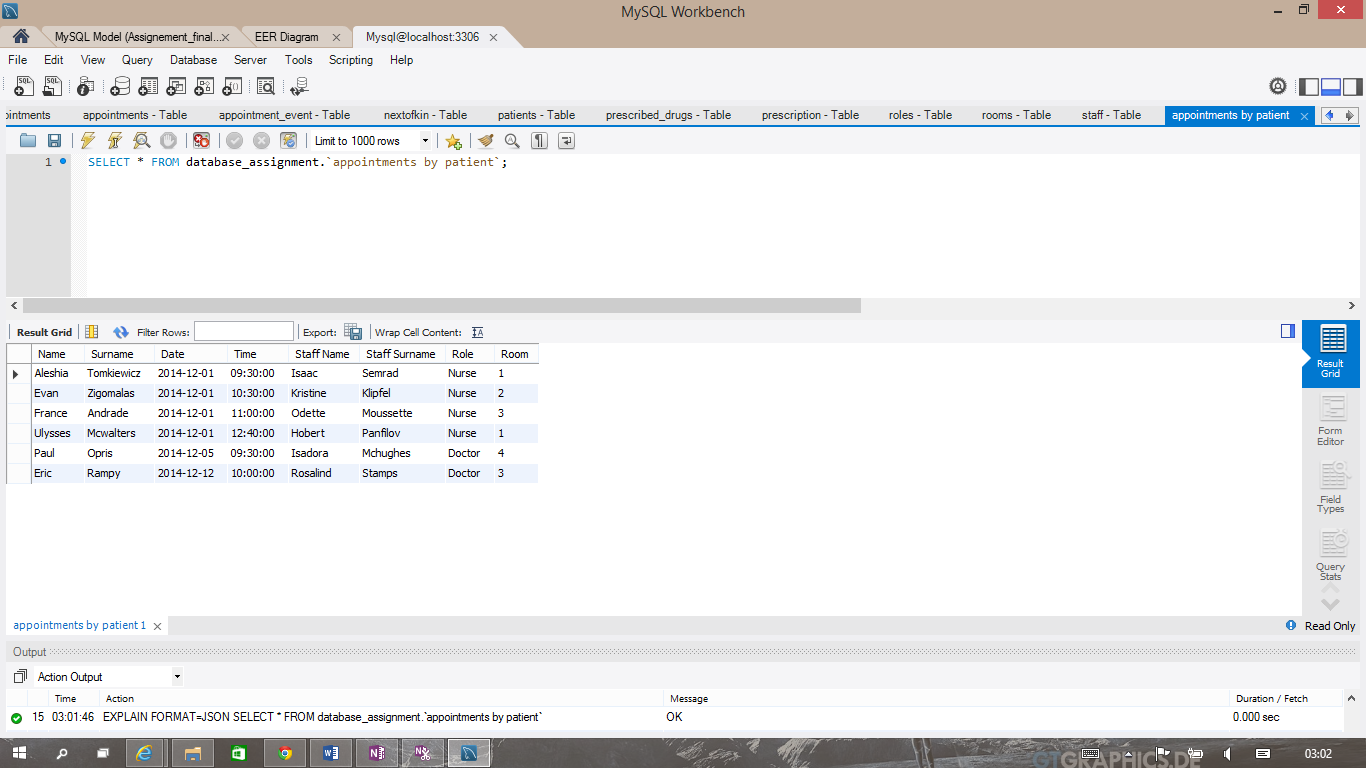
### ‘Before delete’ trigger from ‘Prescription’ table

This trigger will automatically delete the list of the drugs prescribed in a certain prescription, in case that prescription gets deleted from the system. [Appendix 18](#_Appendix_18)

## Views

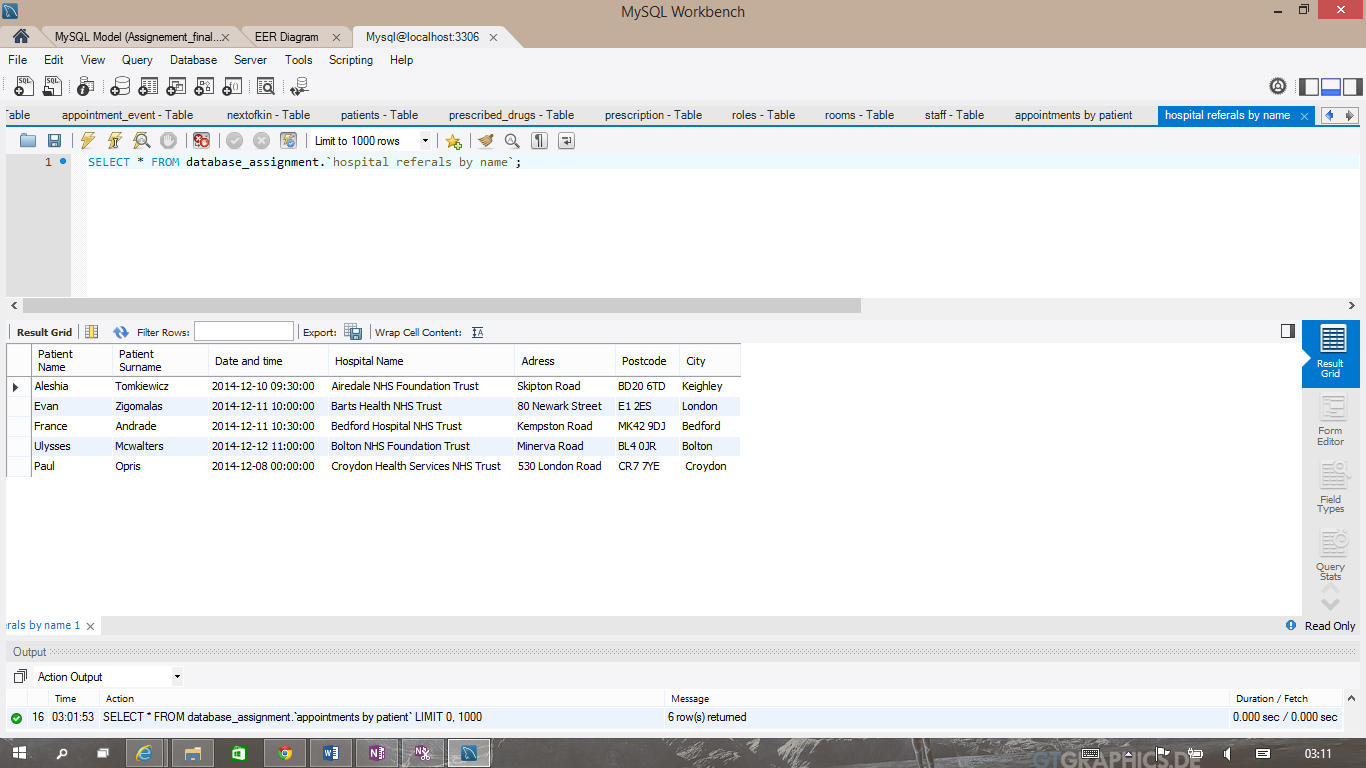
### ‘Appointment by patient’ view

This view will display all the appointments booked for every patient (Name, Surname), in which date and at what time (Date, Time), the staff name and role (Staff Name, Staff Surname, Role) and in which room (Room). This view is using JOIN for four tables in order to obtain all the necessary data. [Appendix 19](#_Appendix_19)



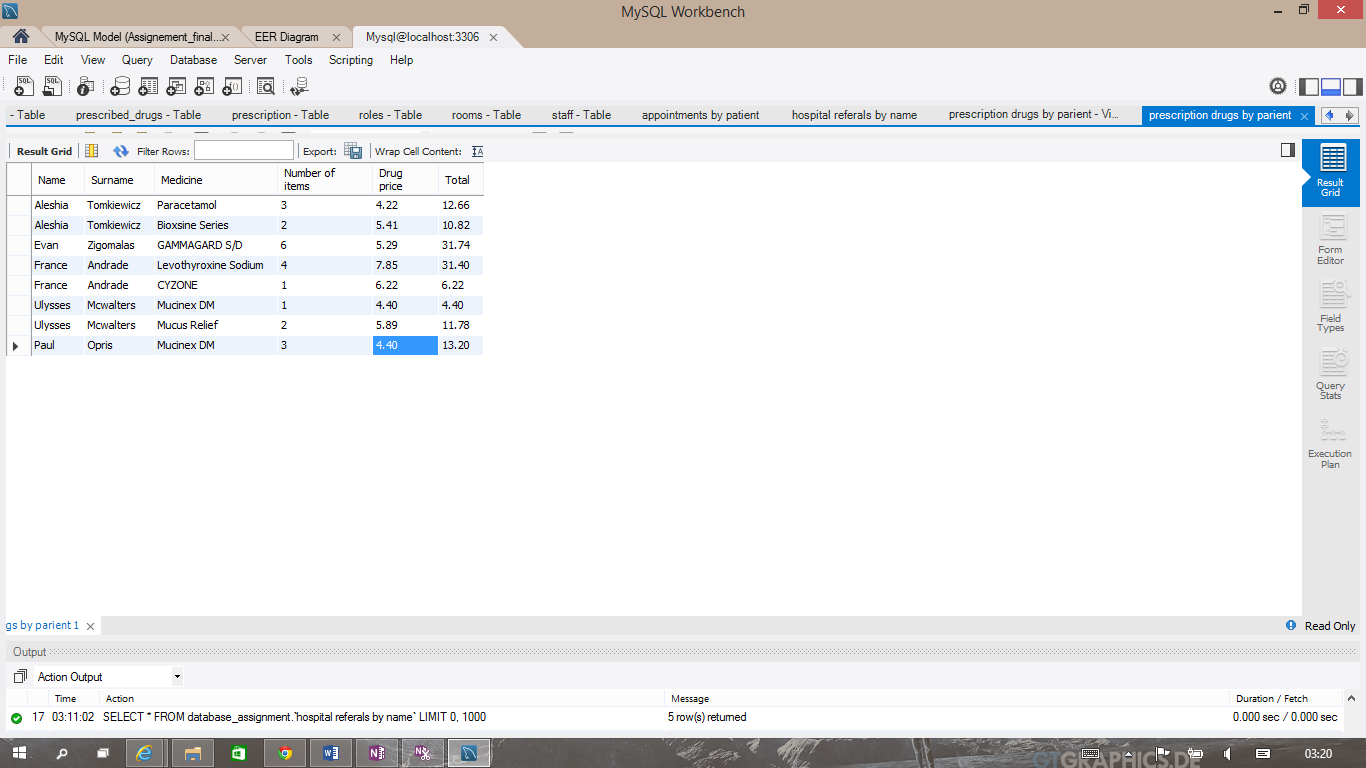
‘Hospital referrals by name’

This view will display all the referrals for all patients including: patient name, surname, date and time of the referral, the hospital name, address, postcode and city. The SQL code used to create this view is using JOIN for four tables to get all the details. [Appendix 20](#_Appendix_20)



### ‘Prescription drugs by patient’

This view will display the medicines prescribed for all patients in the database, how many of each medicine and also their price and a total sum. The SQL code used is including JOIN in four tables and also an aggregate to calculate the total paid for every medicine in a prescription after calculating the price multiplied by the number of items bought. [Appendix 21](#_Appendix_21)



## Security features and advice

“Transactions statements in SQl are used to manage a set of logically related SQL statements. They are used to ensure process and data consistency during database transaction processing. For example, when money is moved from one account to another, the first statement debits the source account while the second credits the destination account. Transactions statements ensure that both statements either fail or succeed to complete the transactions.” (Ebai, 2013)

Although different techniques such as hinting and changing isolation levels to tune looking, the most important consideration is how you manage your transactions. Not managing your transactions effectively can cause significant performance degradation and misunderstanding transaction behaviours will lead to data integrity problems. (Lee & Bieker, 2009)

Even if the hardware and software are reliable, problems can still arise. A very good example will be when multiple people try to use the same database table at the same time. This is called concurrent access and their computers argue about who gets to go first, as contention. Multiple user database systems must be able to handle the ruckus efficiently. (Taylor, 2010)

In order to protect and keep secure the data collected and stored in the database, the access to the computers where is stored should be controlled by the use of log-in ID’s, passwords, and database security restrictions. Authorization to write to the database should be restricted, as should have the ability to retrieve data from the database. It is also important to protect the database from computer hackers. There are different levels of security that can be applied. A system administrator should be responsible for checking for illegal access on a regular basis.

# Conclusion

In this report have been presented all the steps involved in the developing of the GP’s database and also security advices have been shown.

# Bibliography

Ebai, V., 2013. *Whats is SQL?* Bllomington: AuthorHouse.

Lee, M. & Bieker, G., 2009. *Mastering SQL Server 2008*. Indianapolis: Wiley Publishing Inc.

Taylor, A.G., 2010. *SQL for Dummies*. Indianapolis: Wiley Publishing Inc.

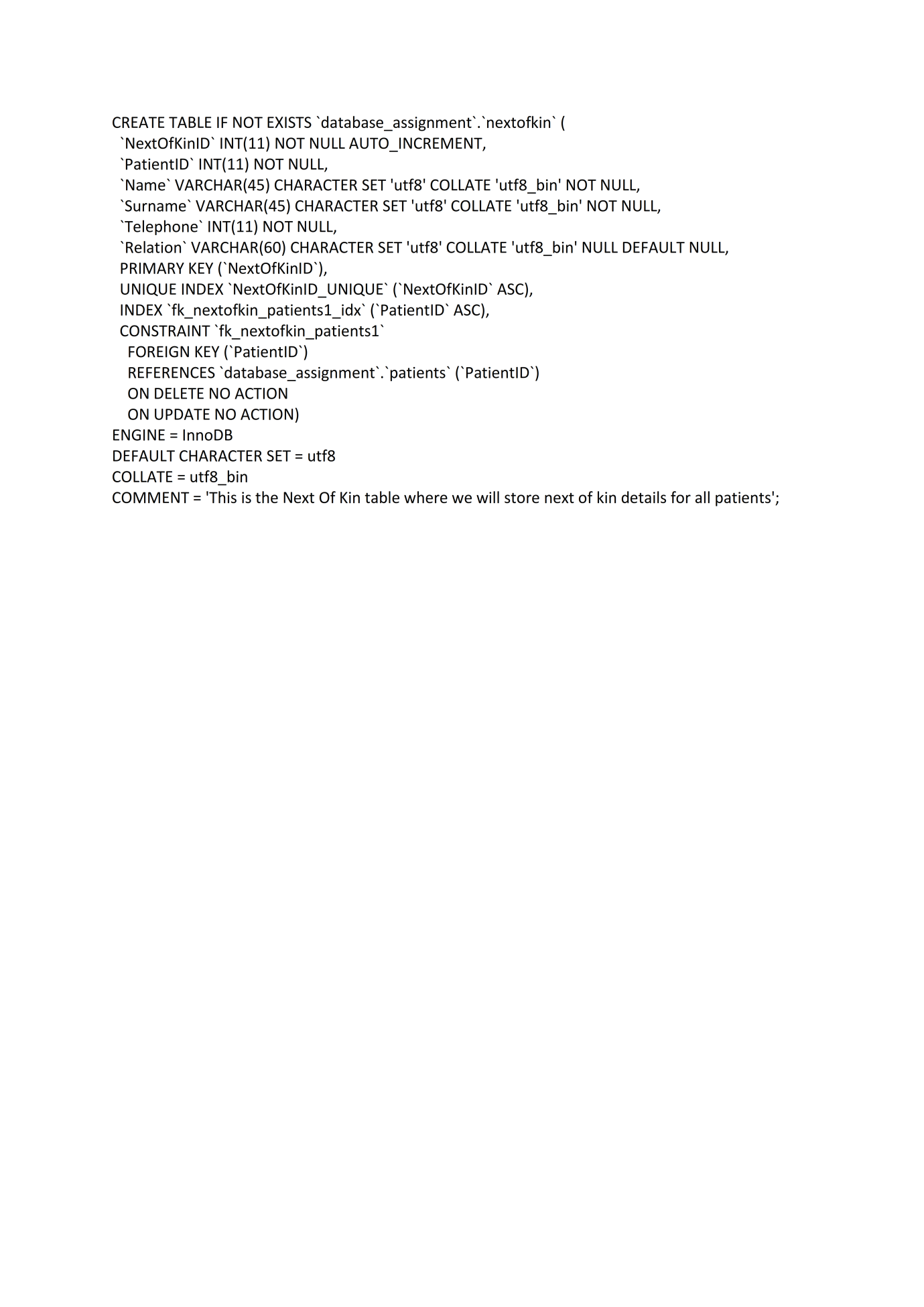
# 

# Appendix

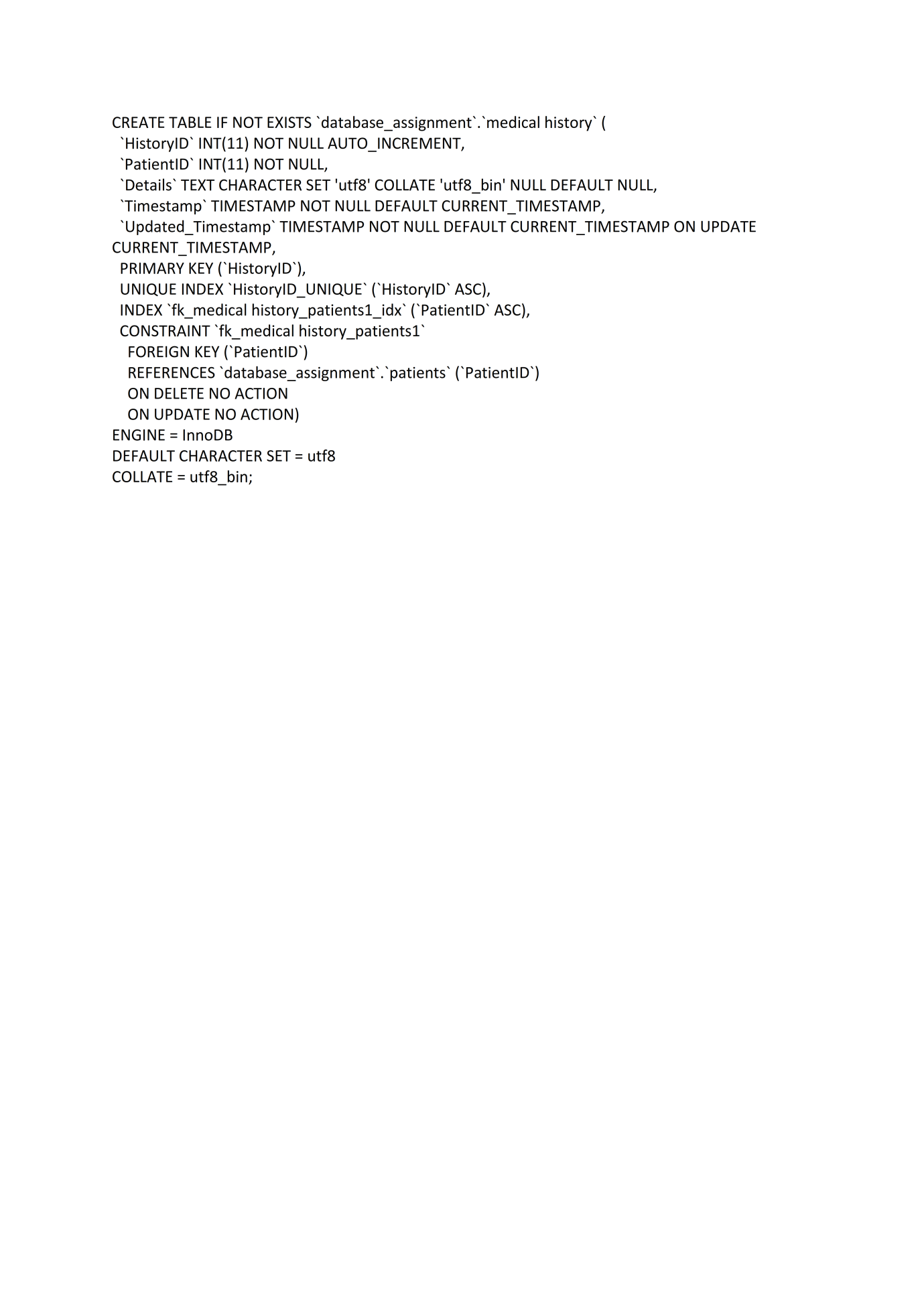
## Appendix 1

## 

## Appendix 2



## Appendix 3



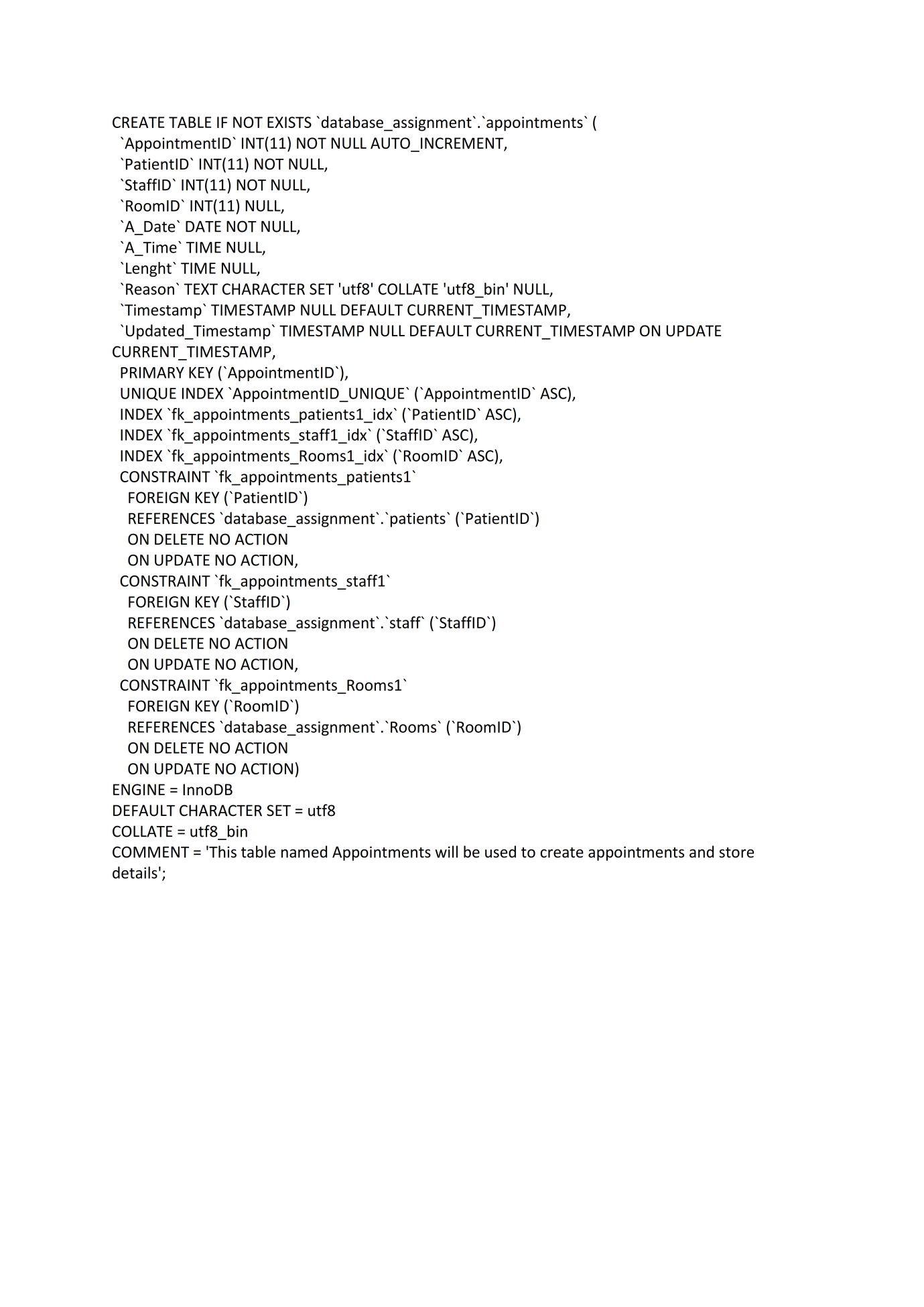
## Appendix 4

## 

## Appendix 5

## 

## Appendix 6



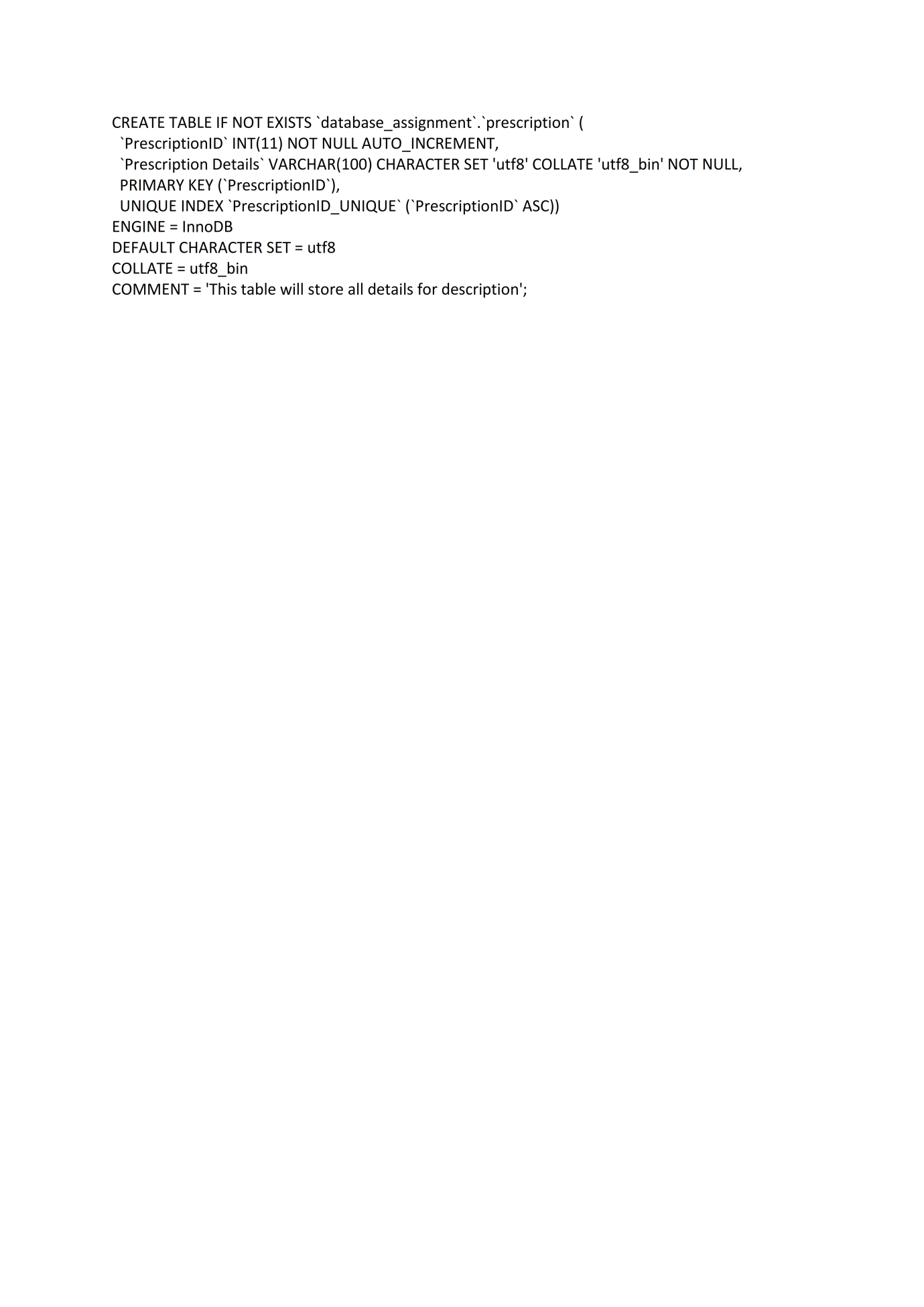
## Appendix 7

## 

## Appendix 8

## 

## Appendix 9



## Appendix 10

## Appendix 11

## 

## Appendix 12

## 

## 

## Appendix 13

## 

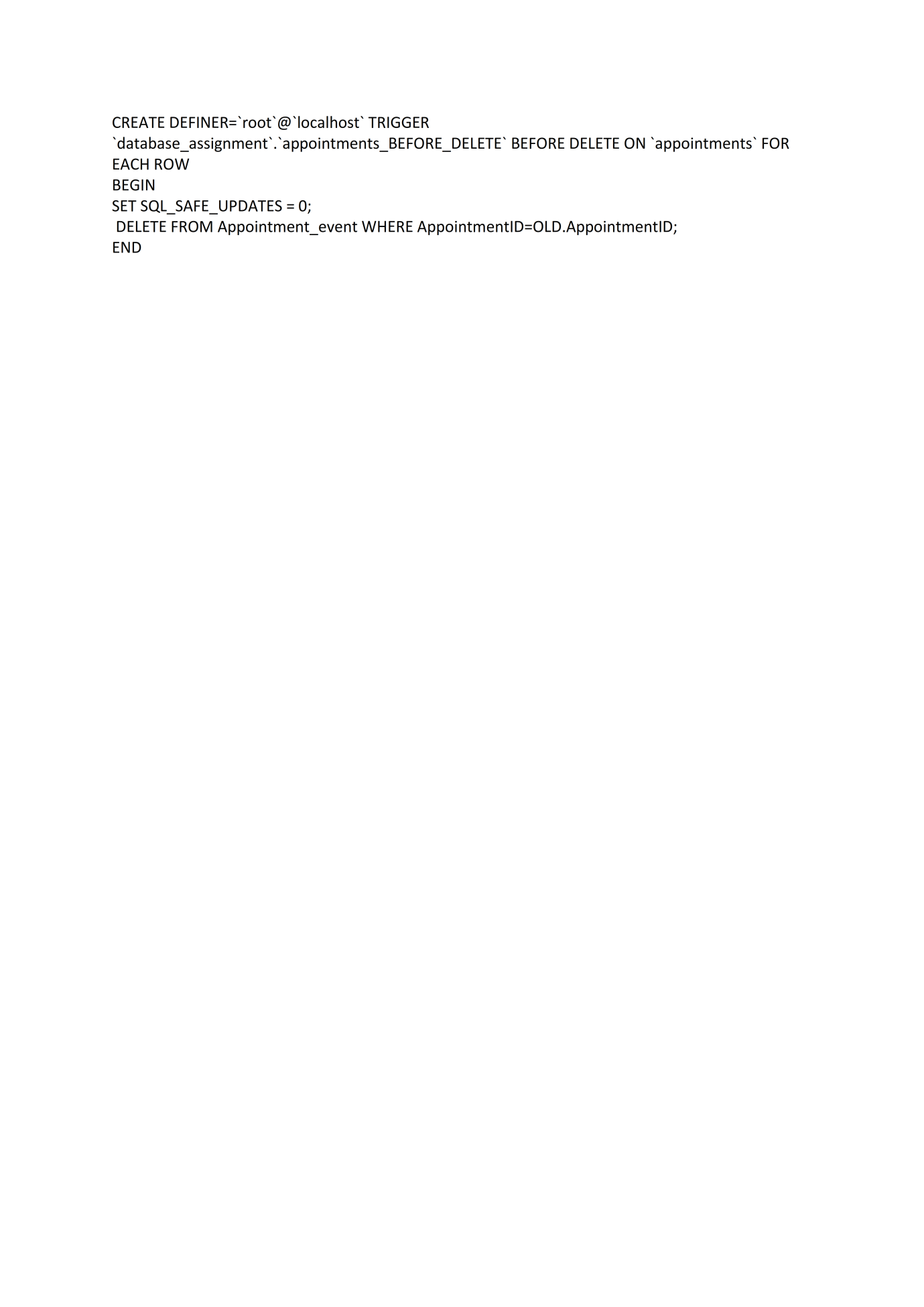
## Appendix 14

## 

## Appendix 15

## 

## Appendix 16



## Appendix 17

## 

## Appendix 18

## 

## Appendix 19

## 

## Appendix 20

## 

## Appendix 21

