Contents

[1.0 Normalization 3](#_Toc483130379)

[1.1 ERD relationsin Third Normal Form 3](#_Toc483130380)

[1.2 FUNCTIONAL DEPENDENCIES and 3NF 5](#_Toc483130381)

[2.0 Relational database implementation 6](#_Toc483130382)

[2.1 Implementation Report 6](#_Toc483130383)

# Normalization

## 1.1 ERD relationsin Third Normal Form

1. **GP** (MedicationProviderNumber, GPName, GPContactNumber)
2. **REFERRAL** (ReferralId, *MedicationProviderNumber*, *AppointmentId*, ReferralDate, Notes)

Foreign Key (*MedicationProviderNumber*) references **GP**

Foreign Key (*AppointmentId*) references **APPOINTMENT**

1. **APPOINTMENT** (AppointmentId*,* *SpecialistId*, *PatientId,* AppoiintmentDate, AppointmentTime, AppointmentType, TotalCharges)

Foreign Key (*SpecialistId*) references **SPECIALIST**

Foreign Key (*PatientId*) references **PATIENT**

1. **DISCOUNT\_APPOINTMENT** (*AppointmentId*, DiscountAmount)

Foreign Key (*AppointmentId*) references **APPOINTMENT**

1. **PAYMENT** (PaymentId, *AppointmentId,*  PaymentDate, Amount)

Foreign Key (*AppointmentId*) references **APPOINTMENT**

1. **SPECIALIST** (SpecialistId, SpecialistName, ContactNumber)
2. **QUALIFICATION** (*SpecialistId,* QualificationName, YearSpecialised)

Foreign Key (*SpecialistId*) references **SPECIALIST**

1. **PRESCRIPTION** (PrescriptionNo, *AppointmentId,*  PrescriptionDate)

Foreign Key (*AppointmentId*) references **APPOINTMENT**

1. **MEDICATION** (*PrescriptionNo, MedicationName*, *PatientId,*  Dosage)

Foreign Key (*PrescriptionNo*) references **PRESCRIPTION**

Foreign Key (*MedicationName*) references **MEDICINE**

Foreign Key (*PatientId*) references **PATIENT**

1. **MEDICINE** (MedicationName, Strength, Instructions)
2. **PATIENT** (PatientId, PatientName, PatientContactNumber, Street, City, State, Postcode, MedicalHistory)
3. **PENSIONER/VETERAN** (*PatientId*, Pensioner/Veteran No)

Foreign Key (*PatientId*) references **PATIENT**

1. **ALLERGY** (*PatientId*, AllergyName)

Foreign Key (*PatientId*) references **PATIENT**

1. **TESTPERFORMANCE** (*AppointmentId, TestId*, TestDetails)

Foreign Key (*AppointmentId*) references **APPOINTMENT**

Foreign Key (*TestId*) references **TEST**

1. **TEST** (TestId, TestName, Charges)

## FUNCTIONAL DEPENDENCIES and 3NF

**GP** and **TEST** relations have been chosen from the above relation set, to list all its functional dependencies and to show how it is in Third Normal Form (3NF)

1. **GP** (MedicationProviderNumber, GPName, GPContactNumber)

**Functional Dependency:** MedicationProviderNumber → GPName, GPContactNumber

**Third Normal Form (3NF):**

The primary key is MedicationProviderNumber which identifies GP. There will be one GPName per MedicationProviderNumber because one GPName is recorded for each GP. The will be only one GPContactNumber for each MedicationProviderNumber. Thus, this relation is in 1NF.

The primary key is MedicationProviderNumber. This is the only candidate key since both GPName and GPContactNumber is not guaranteed to be unique. This means MedicationProviderNumber functionally determines every other attribute in the table. There is a single valued simple candidate key (MedicationProviderNumber) and therefore no partial dependencies are possible. Hence, the relation is in 2NF.

Name cannot be used to functionally determine any other attribute in the table since two different GP may have the same name. Similarly contact number cannot be used to functionally determine any other attribute in the table. Therefore there are no transitive dependencies in the table. Therefore it meets the requirements of first, second and third normal form.

**2 TEST** (TestId, TestName, Charges)

**Functional Dependency:** TestId → TestName, Charges

**Third Normal Form (3NF):**

The primary key is TestId which identifies TEST. There will be one Test Name per Test ID because one Name Test is recorded for each payment. The will be only one Charge for each Test ID. Thus, this relation is in 1NF.

The primary key is TestId. This is the only candidate key since both Test Name and Charges is not guaranteed to be unique. This means TestId functionally determines every other attribute in the table. There is a single valued simple candidate key (TestId) and therefore no partial dependencies are possible. Hence, the relation is in 2NF.

Name cannot be used to functionally determine any other attribute in the table since two different payment may have the same name; likewise Charges cannot be used to functionally determine any other attribute in the table since two different name may have the same charge. Therefore there are no transitive dependencies in the table. Therefore it meets the requirements of first, second and third normal form.

# Relational database implementation

## 2.1 Implementation Report

1. **Interesting procedures learnt:**

With the completion of the assignment I have learnt many interesting things. Some of them are:

**Use of Access Database Application**: I had no previous experience with database applications such as Access. Access was very easy to use. Creating tables and altering values could be easily done. Writing queries to create, delete, update and select necessary information was very interesting to learn. Making relationship among entities could also be easily done. Making forms and reports showed the visual side of access.

**Use of queries for data selection**: Tables could be easily created or deleted from queries but to select necessary information was quite an interesting job to do. The query to SELECT FROM,WHERE, GROUP BY,HAVING,ORDER BY were difficult to understand at first but once you keep on practising it becomes useful and interesting. Queries for selection could be used multiple times to generate different outputs according to the changing values in the table.

1. **Difficulty and solution to complex problem**

I had some difficulty while completing the assignment especially the query related to report section. I had no previous knowledge about writing queries and turning them into reports. It was not clear what we had to show in the report and it how it could be created. The process of altering the values of the report was also unclear.

With the help of our course slides, tutorial solutions provided on week 7, 8 and 9 and with the help of my tutor I was able to come up with the solution to this problem. I was able to write the query to provide all the necessary information for the specialist using the SELECT clause and was able to transform it to a report using report wizard. I was also able to alter between the design view and layout view of the report so it was easy for me to make the report visually appealing.

**ACKNOWLEDGEMENT**

This assignment was completed using COIT20247 course material which includes lecture slides, tutorial solutions and other online materials.