Table of Contents

1. Introduction

In this report, we suggest and create a database design model for Care and Cure Hospital. Care and Cure Hospital's management team has employed us to set up a database system which can help them effectively store and manage data regarding departments, facilities, doctors, nurses, patients, other employees etc. This database will be the first phase of a multi-phase process where more information such as supplier and product information will be added in the second phase. Therefore, we will aim to make this database as efficient and logical as possible while making sure it can be further modified in future updates.

The structure of this report will be as follows. First, we will analyse and discuss the various requirements behind the database such as objectives, stakeholders, scope, limitations etc. After this, we will go through the ER diagram and briefly describe it. Next, we will describe our relational model and document the normalization process. A data dictionary of our database which describes all the relations and attributes will also be provided. After that, we have the SQL queries we developed for the database, followed by a discussion of any noteworthy features, possible improvements, and ethical considerations. Our report concludes with the conclusion and is followed by the appendix and bibliography.

2. Requirements Analysis

Stakeholders

The internal stakeholders for Care and Cure Hospital would include employees such as doctors, nurses and other staff, and higher administrative and management positions such as the CEO, head of each department etc. This is because these people are directly impacted by the functionality of the database every, if not, almost every day. The doctors, nurses and other staff will have all their data recorded into the database, so it is of utmost importance that their key data is properly integrated so that there are no inconsistencies between departments. Also, the database is crucial for admin staff to check important information such as employee salaries, patient data, overdue payments etc. While there are internal stakeholders which are directly impacted by the database from within Care and Cure Hospital, there are also external stakeholders which aren't employed within the hospital itself but are still affected by the database's functionality.

The external stakeholders would include accountancy firms and the patients being admitted to the hospital. *(Stakeholdermap.com, 2019)* It is essential that the patients' data are properly recorded into the database so that there are no potentially harmful consequences. This is because the database records information such as treatment advice and diagnosis which are directly used by the doctors and nurses to administer treatments and checkups. Accountancy firms would require financial data from the database such as salaries, revenue, payments etc. The legitimacy and accuracy of this information is vital for key financial processes such as auditing. The internal and external stakeholders also describe the purpose of the database.

There would be more stakeholders such as suppliers, transporters etc. However, these parties would be more affected by the second phase of this database as opposed to the first phase which we are tasked with designing and creating.

Objectives

The main objectives of a database should reflect the requirements for the proper functionality of the hospital. *(Support.office.com, 2019)* Therefore, for Care and Cure Hospital, the database should be as efficient and accurate in its data integration as possible since a lot of key information can be required at a sudden notice. For the database to be efficient, it should reduce redundancy as much as possible, where ideally the same data is not stored several times in the database. This will be done through normalization of the relational model up till its third normal form, where all multivalued attributes have been split, primary keys have been defined and partial and transitive dependencies have been removed. The accuracy aspect of the database depends on how well the users work with the database, so the database should be very consistent in its design so that users can properly navigate through and pull/input the results they need to with little difficulty. With these main requirements, the database will be able to achieve its purpose of providing accurate and complete information as quickly as possible to its users within the hospital.

Scope

The scope of our database mainly pertains to the inner workings of Care and Cure Hospital. This includes information regarding the various different departments, staff such as doctors and nurses, patients, treatments, operations etc. Basically, anything that is of importance to the main functions of the hospital such as operations, treatments, checkups, and all the background information regarding the different entities involved. This is because our database is just the first phase of Care and Cure Hospital's database system, and does not include external entities such as suppliers, transporters etc. and it does not include the products and medicine which the hospital may need or supply.

Limitations

When designing our database, many problems arise which limit our capabilities. Firstly, we are not given any significantly specific information which we're required to put into our database, and as a result cannot test if our database is absolutely what will work for Care and Cure Hospital. Due to this issue, we have created a database with generic information that we would expect any hospital to have but are restrained by the fact that we don't know exactly how Care and Cure Hospital operated compared to other hospitals. For example, what range of treatments are provided, on what basis are staff paid (hourly or salary), how many floors in the hospital etc. These questions are dealt with via our many assumptions which we will mention later.

Another drawback is that we are technically only making half a database. As previously mentioned, our database is only a part of a much bigger database which Care and Cure Hospital plans for. Consequently, we may be increasing redundancy and causing more errors as we normalize our database for when the system is upgraded to include more entities and attributes regarding products and suppliers.

Assumptions

For our database, we had to make many assumptions as there were an increasing number of gaps to fill as we further developed our ER diagram, relation model, and database in general. The main assumptions we used were:

- Care and Cure Hospital is situated in Australia and all patients and staff live in Australia
- There are three floors in the hospital (1st floor for 1-bed wards, 2nd floor for 2-bed wards, 3rd floor for 3-bed wards)
- Fees are paid via cash, EFTPOS or cheque
- Inpatients receive regular treatments and outpatients undergo operations.
- Patients with health cover are allocated to 3-bed (shared) wards, if they pay extra, they can be upgraded to 1-bed (standalone) wards
- Each department has multiple facilities I.e. department facilities is a multi-valued attribute
- For operational treatments, the admission date of hospitalised patients is either before or on the operation date, which is either before or on the discharge date.

3. ER Model

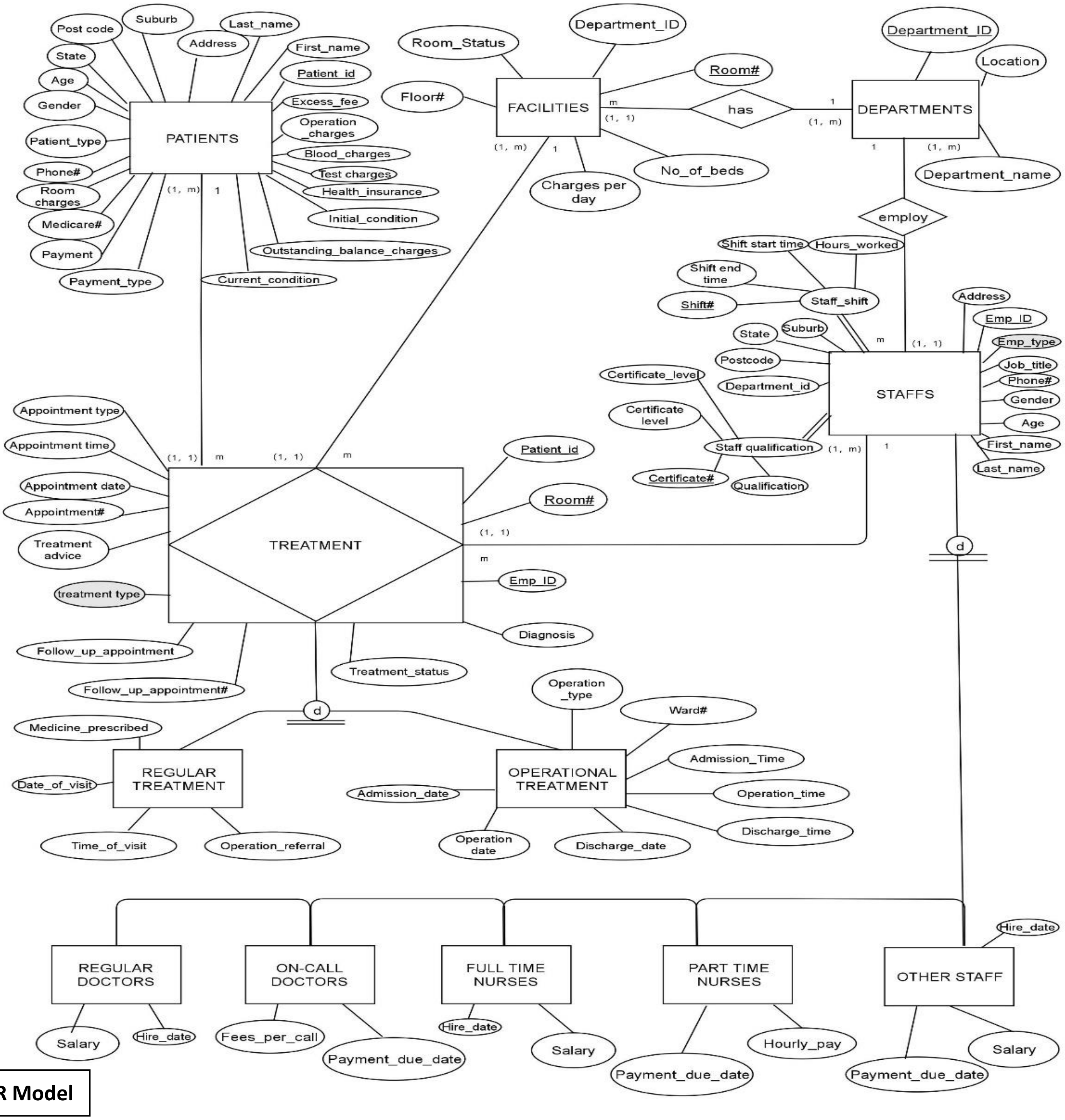
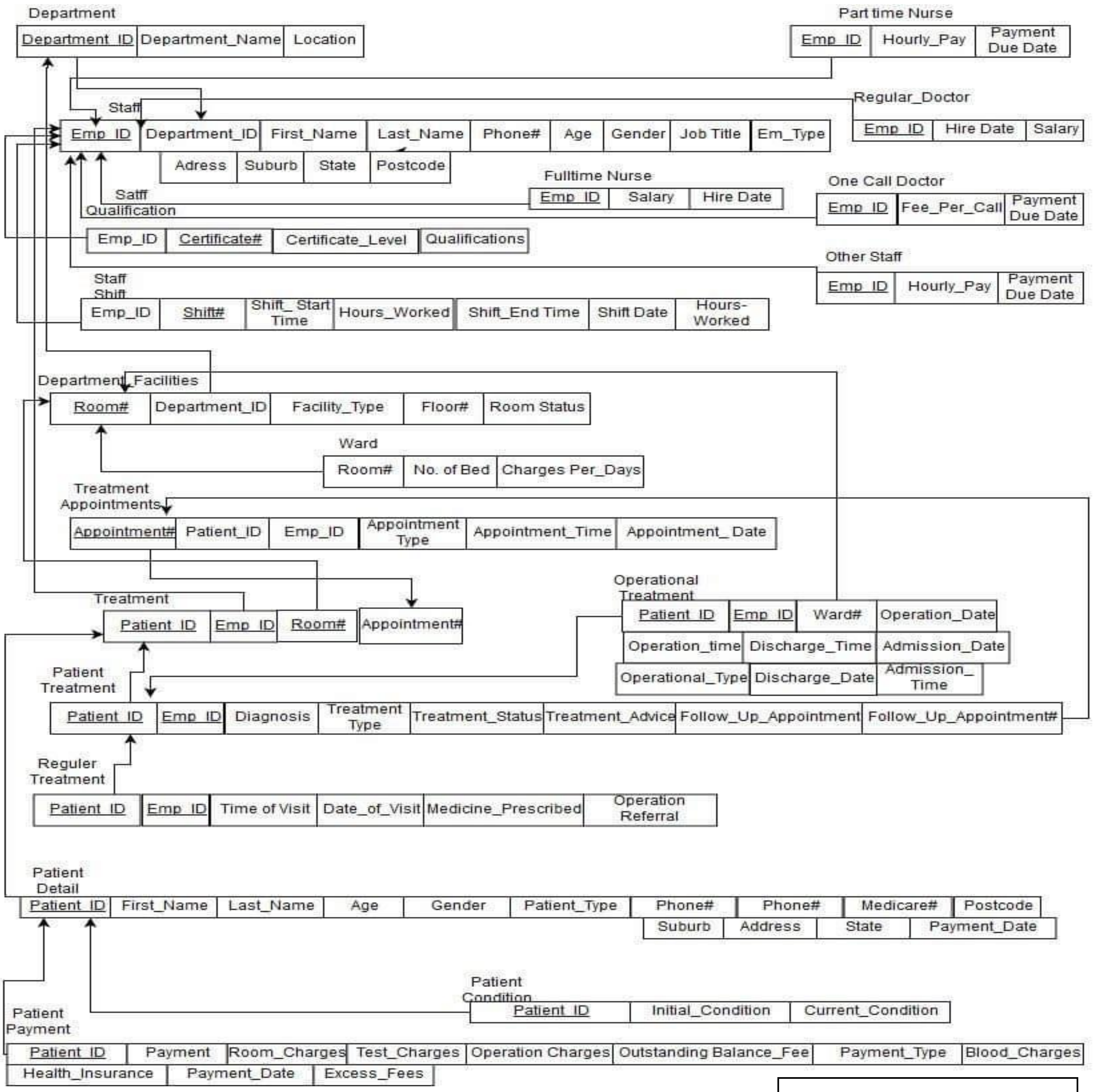


Figure 1: ER Model

Figure 1 shows our ER diagram which conveys many of our design choices. It includes the entities patients, facilities, departments, staff etc. Treatment is a composite entity which is used due to the many-to-many relationship between patients, staff, and facilities and it has two subtypes which represent the different types of treatments performed. Staff is a supertype which has the five different subtypes regular doctors, on-call doctors, full-time nurses, part-time nurses, and other staff. We used this supertype-subtype relationship to show the differences between the job types e.g. whether they receive a set salary or are paid hourly.

4. Relational Model



Department
Department_ID | Department_Name | Location
Part time Nurse
Emp_ID | Hourly_Pay | Payment Due Date
Regular_Doctor
Emp_ID | Hire Date | Salary
Staff
Emp_ID | Department_ID | First_Name | Last_Name | Phone# | Age | Gender | Job Title | Em_Type
Adress | Suburb | State | Postcode
Fulltime Nurse
Emp_ID | Salary | Hire Date
One Call Doctor
Emp_ID | Fee_Per_Call | Payment Due Date
Satff Qualification
Emp_ID | Certificate# | Certificate_Level | Qualifications
Other Staff
Emp_ID | Hourly_Pay | Payment Due Date
Staff Shift
Emp_ID | Shift# | Shift_ Start Time | Hours_Worked | Shift_End Time | Shift Date | Hours-Worked
Department_Facilities
Room# | Department_ID | Facility_Type | Floor# | Room Status
Ward
Room# | No. of Bed | Charges Per_Days
Treatment Appointments
Appointment# | Patient_ID | Emp_ID | Appointment Type | Appointment_Time | Appointment_ Date
Operational Treatment
Patient_ID | Emp_ID | Ward# | Operation_Date
Operation_time | Discharge_Time | Admission_Date
Operational_Type | Discharge_Date | Admission_ Time
Treatment
Patient_ID | Emp_ID | Room# | Appointment#
Patient Treatment
Patient_ID | Emp_ID | Diagnosis | Treatment Type | Treatment_Status | Treatment_Advice | Follow_Up_Appointment | Follow_Up_Appointment#
Reguler Treatment
Patient_ID | Emp_ID | Time of Visit | Date_of_Visit | Medicine_Prescribed | Operation Referral
Patient Detail
Patient_ID | First_Name | Last_Name | Age | Gender | Patient_Type | Phone# | Phone# | Medicare# | Postcode
Suburb | Address | State | Payment_Date
Patient Condition
Patient_ID | Initial_Condition | Current_Condition
Patient Payment
Patient_ID | Payment | Room_Charges | Test_Charges | Operation Charges | Outstanding Balance_Fee | Payment_Type | Blood_Charges
Health_Insurance | Payment_Date | Excess_Fees

Figure 2: Relational Model

Our normalized relational model, as shown in figure 2, has many changes compared to our initial ER diagram. This is a result of the normalization process where more relations/entities had to be introduced or altered in some way which will be further explained in the next section. Some changes were also due to some key personal design choices which we thought would allows for better functionality. It includes extra entities such as patient payment, patient condition, treatment appointments, staff qualification etc.

5. Normalisation Process

1NF

In order to normalise the database to 1NF, Patient_ID, Emp_ID, Appointment_ID, Department_IDand Room# were introduced as primary keys to uniquely identify rows in the tables. As qualifications and staff shifts were multivalued attributes since each employee may have multiple qualifications and multiple shifts, they have been decomposed into separate tables 'Staff_Qualifications' and 'Staff_Shifts'. Records in these tables can be uniquely identified with the primary keys 'Certificate#' and 'Shift#', with each row recording a different qualification or staff shift.

2NF/3NF

Department Entity

In the 'Department' relation, the facility type, floor# and room status are transitively dependent on the 'room#'. Thus, a new relation 'Department_Facilities' was introduced with 'room#' as the primary key and 'department_id' as the foreign key to the 'Department' table.

Since wards have different number of beds and attract direct charges unlike other department facilities (operating theatres and check-up rooms), a new relation 'Ward' was added with the attributes 'no_of_beds' and 'charges_per_day' which are also transitively dependent on the 'room#'. These attributes apply to wards and but are not attributes of the other types of department facilities.

Patient Entity

Since, patient payment, patient details and patient condition are independent of one another, the patient attributes have been decomposed into separate tables.

In the 'Patient_Details' relation, 'address', 'suburb', 'state' and 'postcode' have transitive dependencies. However, it has not been normalised into 3NF because it would be impractical to add another table which only stores data on patient addresses in which case hospital database users would need to join the two tables together whenever they need to retrieve patient data. Thus, 2NF would improve the efficiency of the database design.

Treatment Entity

Because partial dependencies exist for 'appointment_time', 'appointment_date' and 'appointment_type' which only depend on appointment# but not 'patient_id', 'emp_id' or 'room#', these attributes have been transferred to a new relation 'Treatment_Appointments'. However, it is still necessary to include 'patient_id' and 'emp_id' in both the tables 'Treatment_Appointments' and 'Treatment' even though 'appointment#' can act as a foreign key in the 'Treatment' table. This is because an appointment scheduled for the future will be recorded in 'Treatment_Appointments' but not in 'Treatment' until the actual treatment process commences. Another reason is that in a real hospital setting, appointments do not always run according to schedule and doctors may be rotated to treat patients on short notice thus the doctor requested for a certain appointment may be different to the doctor providing the treatment in some rare cases. In addition, for emergency situations, patients may not even have an 'appointment#' but will need to be treated.

Partial dependencies also exist for attributes like diagnosis, treatment advice and medicine prescribed. This is due to the fact that these attributes are only dependent on 'patient_id' and 'emp_id' but not room#. Consequently, they have been transferred to a new relation 'Patient_Treatment'. A discriminator attribute has also been added named 'treatment_type' which is a supertype to regular treatment and operational treatment. These 2 types of treatments have been decomposed into separate tables as they have different attributes transitively dependent on ‘treatment_type’ e.g. date of visit in ‘Regular_Treatment’ compared to operation date in ‘Operational_Treatment’.

Staff Entity
Similarly to the ‘Patient_Details’ table, the ‘address’ attributes have only been normalised to 2NF for the aforementioned reason.

A discriminator attribute ‘emp_type’ has been introduced since the different staff members are paid differently. Transitive dependencies between ‘emp_type’ and ‘salary’ as well as ‘emp_type’ and ‘hourly_wage’ for instance justified the necessity for extra relations to be introduced ‘Regular_Doctors’, ‘On_Call_Doctors’, etc.

Furthermore, despite regular doctors and full time nurses sharing the same attributes, decomposing data on these employee types into two separate tables ensured consistency in the overall structure of the database so that all employees are categorised by job type and type of employment (full time, part time or casual) except for ‘other_staff’ who are assumed to be employed on a part-time/casual basis. From a practical point of view, database users would be less likely to make data anomalies when data is consistently categorised.

6. Data Dictionary

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERECED TABLE
DEPARTMENT	Department_ID,	Department ID Number	NUMBER(6,0)	999999	100000 - 999999	Y	PK	
	Department_Name	Department Name	VARCHAR2(25)	Xxxxxxxx		Y		
	Location	Location	VARCHAR2(10)	Xxxxxxxx		Y		
Department_Facilities	Department_ID,	Department ID Number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	DEPARTMENT
	Facility_type	Facilities Name	VARCHAR2(25)	Xxxxxxxx		Y		
	Room#	Room Number	NUMBER(4,0)	9999	1000. – 9999	Y	FK	Ward
	Floor#	Floor Number	NUMBER(1,0)	9	1 – 9	Y		
	Room_status	Room Status	VARCHAR2(13)	Xxxxxxxx		Y		
Ward	Room#	Room Number	NUMBER(4,0)	9999	1000 – 9999	Y	PK	
	No._Of_Bedrooms	Number of Bedrooms	NUMBER(1,0)	9	1 – 9	Y		
	Charges_Per_Day	Charges per day	NUMBER(6,2)	999999.99	100000.00 – 999999.99	Y		
Patient_Details	Patient_ID	Patient ID number	VARCHAR2(8)	Xxxxxxxx	10000000 – 99999999	Y	PK	
	First_Name	Patient’s First name	VARCHAR2(50)	Xxxxxxxx		Y		
	Last_Name	Patient’s Last Name	VARCHAR2(50)	Xxxxxxxx		Y		
	Age	Patient’s Age	NUMBER(3,0)	999	100 - 999	Y		
	Gender	Patient’s Gender	CHAR(1)	X	0-9	Y		
	Medicare#	Patient’s Medicare	NUMBER(10,0)	9999999999	100000000 – 9999999999	Y		
	Address	Patient’s Address	VARCHAR2(50)	Xxxxxxxx		Y		

	Suburb	Patient's suburb name	VARCHAR2(50)	Xxxxxxxx		Y		
	State	Patient's state name	VARCHAR2(3)	Xxxxxxxx		Y		
	Postcode	Patient's postcode	NUMBER(4,0)	9999	1000 – 9999	Y		
	Phone#	Patient's phone number	NUMBER(10,0)	9999999999	100000000 – 9999999999	Y		
	Patient_Type	Patient Type (Regular or Hospitilised)	VARCHAR2(15)	Xxxxxxxx		Y		
Patient_Condition	Patient_ID	Patient ID Number	CHAR(8 BYTE)	99999999	10000000 – 99999999	Y	FK	Patient_Details
	Initial_Condition	Patient's initial condition	VARCHAR2(50)	Xxxxxxxx		Y		
	Current_Condition	Patient's Current condition	VARCHAR2(50)	Xxxxxxxx		Y		
Patient_Payment	Patient_ID	Patient ID Number	CHAR(8)	99999999	10000000 – 99999999	Y	FK	Patient_Details
	Payment	Payment	NUMBER(8,2)	99999999.99	10000000.00 – 99999999.99	Y		
	Room_charges	Room charges	NUMBER(6,0)	999999	100000 – 999999	Y		
	Operation_charges	Operation charges	NUMBER(8,0)	99999999	10000000 – 99999999	Y		
	Test_charges	Test charges	NUMBER(7,0)	9999999	1000000 – 9999999	Y		
	Blood_charges	Blood charges	NUMBER(8,0)	99999999	10000000 – 99999999	Y		
	Outstanding_balance	Outstanding balance	NUMBER			Y		
	Payment_type	Payment type	VARCHAR2(15)	Xxxxxxxx		Y		
	Health_insurance	Health insurance type	VARCHAR2(3)	Xxxxxxxx		Y		
	Payment_date	Payment date	DATE	dd-mmm-yyyy		Y		
	Excess_fees	Excess fees	NUMBER(3,0)	999	100- 999	Y		
Treatment	Patient_ID	Patient Id Number	CHAR(8)	99999999	10000000 – 99999999	Y	FK	Patient_details
	Emp_ID,	Employee Id Number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Room#	Room Number	NUMBER(4,0)	9999	1000 – 9999	Y	FK	Ward
	Appointment#	Appointment Number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Treament_Appointments
Patient_Treatment	Patient_ID	Patient Id Number	CHAR(8)	99999999	10000000 – 99999999	Y	FK	Patient_details
	Emp_ID	Employee Id Number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Diagnosis	Diagnosis	VARCHAR2(50)	Xxxxxxxx		Y		
	Treatment_Type	Treatment Type	VARCHAR2(9)	Xxxxxxxx		Y		
	Treatment_Status	Treatment Status		Xxxxxxxx		Y		

	Treatment_Advice
	Follow_Up_Appointment
	Follow_Up_Appointment#
		Treatment Advice
		Follow up Appointment
		Follow Up Appointment Number
			VARCHAR2(15)
			VARCHAR2(50)
			CHAR(1)
			NUMBER(6,0)
				Xxxxxxxx
				9
				999999
					1 – 9
					100000 - 999999
						Y
						Y
						Y		

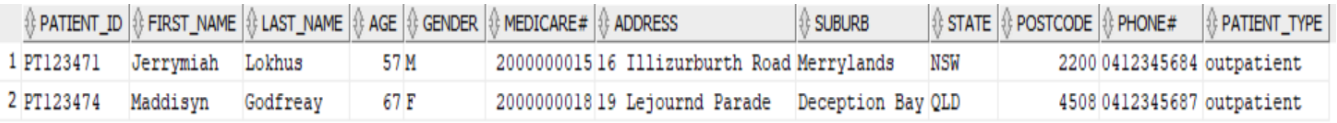
Treatment_Appointments	Appointment#	Appointment Number	NUMBER(6,0)	999999	100000 – 999999	Y	PK	
	Patient_ID	Patient Id Number	CHAR(8)	99999999	10000000 – 99999999	Y	FK	Patient_details
	Emp_ID	Employee Id Number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Appointment_Type	Appointment type	VARCHAR2(90	Xxxxxxxx		Y		
	Appointment_Time	Appointment time	CHAR(5 BYTE)	99999	10000 - 99999	Y		
	Appointment_Date	Appointment date	DATE	dd-mmm-yyyy		Y		
Regular_Treatment	Patient_ID	Patient Id Number	CHAR(8)	99999999	10000000 – 99999999	Y	Fk	Patient_details
	Emp_ID	Employee Id Number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Time_Of_Visit	Time of visit	CHAR(5)	99999	10000 – 99999	Y		
	Date_Of_Visit	Date of visit	DATE	dd-mmm-yyyy		Y		
	Medicine_Prescribed	Medicine prescribed	VARCHAR2(50)	Xxxxxxxx		Y		
	Operation_Referral	Operational referral	VARCHAR2(3)	Xxxxxxxx		Y		
Operational_Treatment	Patient_ID	Patient Id Number	CHAR(8)	99999999	10000000 – 99999999	Y	FK	Patient_details
	Emp_ID	Employee Id Number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Operation_Type	Operation type	VARCHAR2(20)	Xxxxxxxx		Y		
	Admission_Time	Admission time	CHAR(5)	99999	10000 – 99999	Y		
	Admission_Date	Admission date	DATE	dd-mmm-yyyy		Y		
	Operation_Time	Operation time	CHAR(5)	99999	10000 – 99999	Y		
	Operation_Date	Operation date	DATE	dd-mmm-yyyy		Y		
	Discharge_Time	Discharge time	CHAR(5)	99999	10000 – 99999	Y		
	Discharge_Date,	Discharge date	DATE	dd-mmm-yyyy		Y		
	Ward#	Ward number	NUMBER(4,0)	9999	1000 - 9999	Y		
Staff_Details	Emp_ID	Employee id number	NUMBER(6,0)	999999	100000 – 999999	Y	PK	
	Department_ID	Depratment id	NUMBER(6,0)	999999	100000 – 999999	Y	FK	DEPARTMENT
	First_Name	First name	VARCHAR2(50)	Xxxxxxxx		Y		
	Last_Name	Last name	VARCHAR2(50)	Xxxxxxxx		Y		
	Age	Age	NUMBER(3)	999	100 – 999	Y		
	Gender	Gender	CHAR(1)	X	M/F	Y		
	Address,	Address	VARCHAR2(50)	Xxxxxxxx		Y		
	Suburb	Suburb	VARCHAR2(50)	Xxxxxxxx		Y		

	State	State	VARCHAR2(3)	Xxx		Y		
	Postcode,	Postcode	NUMBER(4)	9999	1000 - 9999	Y		
	Phone#	Phone number	VARCHAR2(50)	Xxxxxxxx		Y		
	Emp_Type,	Employee type	VARCHAR2(15)	Xxxxxxxx		Y		
	Job_Title	Job title	VARCHAR2(20)	Xxxxxxxx		Y		
Staff_Qualifications	Emp_ID	Employee id number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Qualifications	Qualifications	VARCHAR2(25)	Xxxxxxxx		Y		
	Certificate#	Certificate number	VARCHAR2(10)	Xxxxxxxx		Y		
	Certificate_Level	Certificate Level	NUMBER(1,0)	9	1 - 9	Y		
Staff_Shifts	Emp_ID	Employee id number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Shift_Start_Time	Shift start time	CHAR(5)			Y		
	Shift_End_Time	Shift end time	CHAR(5)	99:99		Y		
	Shift_Date	Shift date	DATE	dd-mmm-yyyy		Y		
	Hours_Worked	Hours worked	NUMBER(1,0)	9	9	Y		
Regular_Doctors	Emp_ID	Employee id number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Salary	Salary	NUMBER(9,2)	999999999.99	100000000.000 - 999999999.99	Y		
	Hire_Date	Hire date	DATE	dd-mmm-yyyy		Y		
On_Call_Doctors	Emp_ID	Employee id numbe	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Fees_Per_Call	Fees per day	NUMBER(7,2)	9999999.99	1000000.00 - 9999999.99	Y		
	Payment_Due_Date	Payment due date	DATE	dd-mmm-yyyy		y		
Full_Time_Nurses	Emp_ID	Employee id number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Salary	Salary	NUMBER(9,2)	999999999.99	100000000.000 - 999999999.99	Y		
	Hire_Date	Hire date	DATE	dd-mmm-yyyy		y		
Part_Time_Nurses	Emp_ID	Employee id number	NUMBER(6,0)	999999	100000 – 999999	Y	FK	Staff_details
	Hourly_Pay	Hourly pay	NUMBER(7,2)	9999999.99	1000000.00 - 9999999.99	Y		
	Payment_Due_Date	Payment due date	DATE	dd-mmm-yyyy		y		
Other_Staff	Emp_ID	Employee id number	NUMBER(6,0)	999999	100000 – 999999	Y	Fk	Staff_details
	Hourly_Pay	Hourly pay	NUMBER(7,2)	9999999.99	1000000.00 - 9999999.99	Y		
	Payment_Due_Date	Payment due date	DATE	dd-mmm-yyyy		y		

7. SQL Queries

Query 1

```
SELECT *
FROM patient_details
WHERE patient_type = 'outpatient' AND age > 55;
```

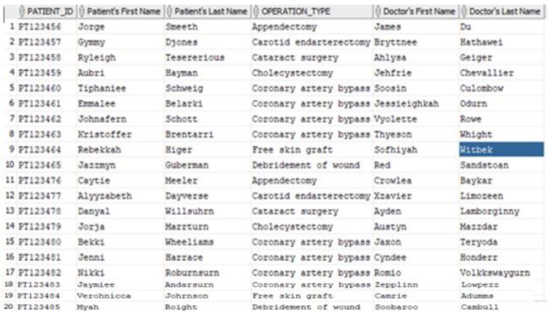


	PATIENT_ID	FIRST_NAME	LAST_NAME	AGE	GENDER	MEDICARE#	ADDRESS	SUBURB	STATE	POSTCODE	PHONE#	PATIENT_TYPE
1	PT123471	Jerrymiah	Lokhus	57	M	2000000015	16 Illizurburth Road	Merrylands	NSW	2200	0412345684	outpatient
2	PT123474	Maddisyn	Godfreay	67	F	2000000018	19 Lejournd Parade	Deception Bay	QLD	4508	0412345687	outpatient

Figure 3: Query 1 Results

Query 2

```
SELECT p.patient_id, p.first_name "Patient's First Name", p.last_name "Patient's Last Name",
o.operation_type, s.first_name "Doctor's First Name", s.last_name "Doctor's Last Name"
FROM patient_details p
JOIN operational_treatment o
   ON p.patient_id = o.patient_id
JOIN staff_details s
   ON o.emp_id = s.emp_id
WHERE o.operation_date >= '01-JAN-18';
```

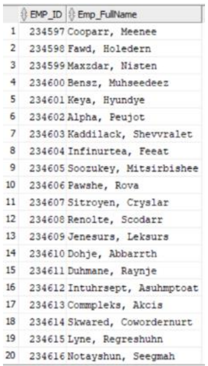


	PATIENT_ID	Patient's First Name	Patient's Last Name	OPERATION_TYPE	Doctor's First Name	Doctor's Last Name
1	PT123456	Jorge	Smeeth	Appendectomy	James	Du
2	PT123457	Gymmy	Djones	Carotid endarterectomy	Bryttnee	Hathawei
3	PT123458	Ryleigh	Tesererious	Cataract surgery	Ahlysa	Geiger
4	PT123459	Aubri	Hayman	Cholecystectomy	Jehfrie	Chevallier
5	PT123460	Tiphaniee	Schweig	Coronary artery bypass	Soosin	Culombow
6	PT123461	Emmalee	Belarki	Coronary artery bypass	Jessieighkah	Odurn
7	PT123462	Johnafern	Schott	Coronary artery bypass	Vyolette	Rowe
8	PT123463	Kristoffer	Brentarri	Coronary artery bypass	Thyeson	Whight
9	PT123464	Rebekkah	Higer	Free skin graft	Sofhiyah	Witbek
10	PT123465	Jazzmyn	Guberman	Debridement of wound	Red	Sandstoan
11	PT123476	Caytie	Meeler	Appendectomy	Crowlea	Baykar
12	PT123477	Alyyzabeth	Dayverse	Carotid endarterectomy	Xzavier	Limozeen
13	PT123478	Danyal	Willsuhrn	Cataract surgery	Ayden	Lamborginny
14	PT123479	Jorja	Marrturn	Cholecystectomy	Austyn	Mazzdar
15	PT123480	Bekki	Wheeliams	Coronary artery bypass	Jaxon	Teryoda
16	PT123481	Jenni	Harrace	Coronary artery bypass	Cyndee	Honderr
17	PT123482	Nikki	Roburnsurn	Coronary artery bypass	Romio	Volkkswaygurn
18	PT123483	Jaymiee	Andarsurn	Coronary artery bypass	Zepplinn	Lowpezz
19	PT123484	Verohnicca	Johrnson	Free skin graft	Camrie	Adumms
20	PT123485	Myah	Roight	Debridement of wound	Soobaroo	Cambull

Figure 4: Query 2 Results

Query 3

```
SELECT emp_id, last_name || ', ' || first_name "Emp_FullName"
FROM staff_details
WHERE emp_type = 'Full-Time Nurse' OR emp_type = 'Part-Time Nurse';
```

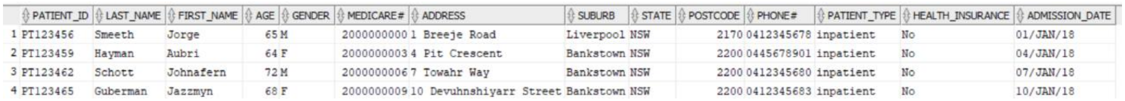


	EMP_ID	Emp_FullName
1	234597	Cooparr, Meenee
2	234598	Fawd, Holedern
3	234599	Maxzdar, Nisten
4	234600	Bensz, Muhseedeez
5	234601	Keya, Hyundye
6	234602	Alpha, Peujot
7	234603	Kaddilack, Shevvralet
8	234604	Infinurtea, Feeat
9	234605	Soozukey, Mitsirbishee
10	234606	Pawshe, Rova
11	234607	Sitroyen, Cryslar
12	234608	Renolte, Scodarr
13	234609	Jenesurs, Leksurs
14	234610	Dohje, Abbarrth
15	234611	Duhmane, Raynje
16	234612	Intuhrsept, Asuhmptoat
17	234613	Commpleks, Akcis
18	234614	Skwared, Cowordernurt
19	234615	Lyne, Regreshuhn
20	234616	Notayshun, Seegmah

Figure 5: Query 3 Results

Query 4

```
SELECT pd.patient_id, pd.last_name, pd.first_name, pd.age, pd.gender, pd.medicare#,
   pd.address, pd.suburb, pd.state, pd.postcode, pd.phone#, pd.patient_type, pp.health_insurance,
o.admission_date
FROM patient_details pd
JOIN patient_payment pp
   ON pd.patient_id = pp.patient_id
JOIN operational_treatment o
   ON pp.patient_id = o.patient_id
WHERE pp.health_insurance = 'No' AND o.admission_date BETWEEN '01/JAN/17' AND '31/DEC/18';
```



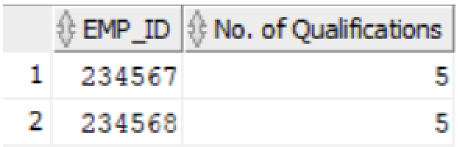
	PATIENT_ID	LAST_NAME	FIRST_NAME	AGE	GENDER	MEDICARE#	ADDRESS	SUBURB	STATE	POSTCODE	PHONE#	PATIENT_TYPE	HEALTH_INSURANCE	ADMISSION_DATE
1	PT123456	Smeeth	Jorge	65	M	2000000000	1 Breeje Road	Liverpool	NSW	2170	0412345678	inpatient	No	01/JAN/18
2	PT123459	Hayman	Aubri	64	F	2000000003	4 Pit Crescent	Bankstown	NSW	2200	0445678901	inpatient	No	04/JAN/18
3	PT123462	Schott	Johnafern	72	M	2000000006	7 Towahr Way	Bankstown	NSW	2200	0412345680	inpatient	No	07/JAN/18
4	PT123465	Guberman	Jazzmyn	68	F	2000000009	10 Devuhnshiyarr Street	Bankstown	NSW	2200	0412345683	inpatient	No	10/JAN/18

Figure 6: Query 4 Results

Query 5

```
SELECT sq.emp_id, COUNT(sq.emp_id) "No. of Qualifications"
FROM staff_details s
JOIN staff_qualifications sq
```

```
   ON s.emp_id = sq.emp_id
WHERE s.emp_type = 'Regular Doctor' OR s.emp_type = 'On-Call Doctor'
GROUP BY sq.emp_id
HAVING COUNT(sq.emp_id) > 1;
```

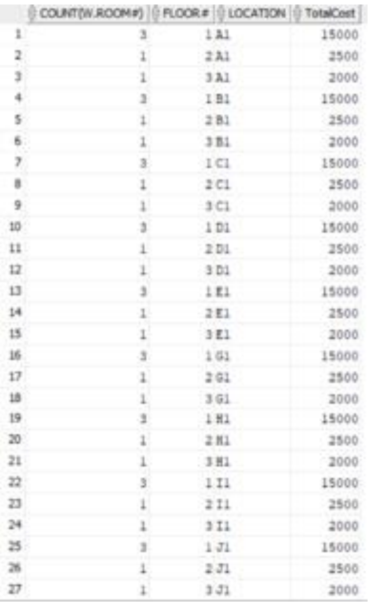


	EMP_ID	No. of Qualifications
1	234567	5
2	234568	5

Figure 7: Query 5 Results

Query 6

```
SELECT COUNT(w.room#), df.floor#, d.location, w.charges_per_day * COUNT(w.room#) "TotalCost"
FROM department d
JOIN department_facilities df
   ON d.department_id = df.department_id
JOIN ward w
   ON df.room# = w.room#
WHERE NOT EXISTS(SELECT ot.admission_date, ot.discharge_date
   FROM operational_treatment ot
   JOIN ward w
      ON ot.ward# = w.room#
   WHERE ot.admission_date <= '01-Jan-19' AND ot.discharge_date > '01-Jan-19' AND ot.ward# =
w.room#)
GROUP BY df.floor#, d.location, w.charges_per_day
ORDER BY d.location, df.floor#;
```

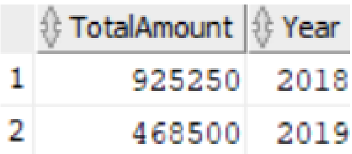


	COUNT(W.ROOM#)	FLOOR#	LOCATION	TotalCost
1	3	1	A1	15000
2	1	2	A1	2500
3	1	3	A1	2000
4	3	1	B1	15000
5	1	2	B1	2500
6	1	3	B1	2000
7	3	1	C1	15000
8	1	2	C1	2500
9	1	3	C1	2000
10	3	1	D1	15000
11	1	2	D1	2500
12	1	3	D1	2000
13	3	1	E1	15000
14	1	2	E1	2500
15	1	3	E1	2000
16	3	1	G1	15000
17	1	2	G1	2500
18	1	3	G1	2000
19	3	1	H1	15000
20	1	2	H1	2500
21	1	3	H1	2000
22	3	1	I1	15000
23	1	2	I1	2500
24	1	3	I1	2000
25	3	1	J1	15000
26	1	2	J1	2500
27	1	3	J1	2000

Figure 8: Query 6 Results

Query 7

```
SELECT SUM(room_charges) + SUM(operation_charges) + SUM(test_charges) + SUM(blood_charges)
+ SUM(excess_fees) "TotalAmount",
   extract(year from payment_date) "Year"
FROM patient_payment
WHERE payment_date IS NOT NULL
GROUP BY extract(year from payment_date)
ORDER BY "TotalAmount" DESC;
```



	TotalAmount	Year
1	925250	2018
2	468500	2019

Figure 9: Query 7 Results

8. Noteworthy Features

Our database included some features which were beyond the requirements given by Care and Cure Hospital. We not only added the ability to search up which operating theatre each patient is being treated in, but also the ward they get transferred to after the operation before they leave the hospital. This allows the hospital to keep track of where patients who aren't admitted to the hospital stay when they aren't being operated on.

Another notable aspect our database has is where we separated appointment information and treatment information via the tables 'Treatment_Appointments' and 'Treatment' respectively instead of having a single 'Treatment' table. This improves the flexibility of the database in the case that a patient does not turn up to an appointment so that only data in the appointment table is changed but not the treatment table.

We also added many check constraints wherever possible. This reduces the likelihood of users in Care and Cure Hospital accidentally entering wrong data which has the potential to have very drastic consequences.

9. Possible Improvements

After completing our database, we realised there were a few issues which if accounted for, would have been a definite improvement to our current set-up. One such issue was that it was difficult to retrieve data on all the wards which were vacant on a past date. This problem could have been fixed through the addition of a new table where past room status was recorded which would include the dates for which each ward was occupied.

Another possible improvement is to add languages other than just 'English' for the database. Australia contains many different cultures, a database addressing a diverse range of user ethnicities will indefinitely make it more accessible to its various stakeholders. Providing unique experience via translating entity and attribute names to their preferred language will help accommodate improved user interaction.

Since the database has been normalized to 2NF/3NF, the users (especially doctors) need to enter data in multiple tables when treating patients which can reduce the efficiency of hospital operations. In addition, thousands of patients could visit the hospital everyday thus over several years there could possibly be several million records. Since the character lengths of Patient_ID and Appointment# have been restricted to only 6 numerical figures which can only hold up to 1 million unique numbers, there may be insufficient unique numbers to accommodate all of these records. For this reason, 3NF may no longer be viable in the future and a BigData system would have to be implemented to improve the database.

10. Ethics

The main ethical issue relating to the use of a database system pertains to the confidentiality of the patient information and the staff information recorded in the information system. Particularly, sensitive information such as residential addresses and historical illness records could be leaked to external parties which breaches the privacy of patients and employees in the case of a data breach incident or an act of fraud by malicious insiders. Moreover, ethical implications could result from data anomalies, as patient data may be entered or retrieved incorrectly.

Without adequately stringent security policies, confidential information could be accessed by hackers who infiltrate the hospital's servers known as a data breach. If CCH fails to adopt the appropriate data security standards such as the use of packet-filtering firewalls to block incoming server traffic from potentially harmful sources, the hospital would be unable to detect suspicious activity on its network servers and mitigate a cyber-attack. Thus, it could risk leaking data to third parties who may distribute or sell the information illegally to data companies. As a result, the confidential information of patients and staff members could circulate around the Internet with a large number of web users gaining unauthorised access to the data.

Fraudulent practices by employees is another ethical implication, especially the improper use of patient data by staff members with malicious intentions. Poor relations between staff members and department heads may provoke employees to behave in an unethical manner where they may retaliate by disclosing patient and staff data to other hospitals and medical clinics, or even steal the data for themselves if they decide to resign from the hospital and start their own medical clinic. This is another risk of storing patient and staff data in a database where their privacy is compromised.

Ethical concerns extending beyond data privacy include issues arising from data anomalies primarily the input or retrieval of the wrong data. For instance, the inaccurate input of data into the database may result in prescriptions of the wrong medicine during follow-up appointments, since doctors rely on the data entries on patients' diagnoses from the first appointment to determine the type of medicine and the amount of medicine to prescribe depending on patients' current health conditions. Data anomalies may also lead to surgeons operating on the incorrect patient if the room number of the operating theatre is entered incorrectly into the database or result in surgeons performing the incorrect operation if the operation type is entered incorrectly. These scenarios pose very serious ethical concerns as they could have drastic consequences on patients' health and may even cause fatalities in some extreme situations due to the side-effects of potent medicine as well as the debilitating impacts of operational treatment. Furthermore, if patients with health insurance are recorded as having no health insurance in the database, they will be charged higher costs for their treatment which is unfair and unjust towards the affected patients despite the unintentional motives of the employees responsible for the errors.

11. Conclusion

As we progressed through our report, we gained a great understanding of what makes an efficient database. Although it was a long and strenuous process, we eventually made a working database which accomplished all of Care and Cure Hospital's needs.

Appendix

Critical Reflection

Our group consists of five group members. The communication tool for the group is "messenger" and planned regular meetings are on Monday in the morning. We used a shared working document which is on outlook and can be accessed and edited by any and all members at the same time. The project has been broken down to different small parts and assigned to different group members. Each member has been assigned to at least one writing and one designing job. The ER diagram was designed by every group member in the first stage although a large part of ER diagram has been modified and changed in the later stage. The finalization of the ER diagram, data dictionary, relational model, design tables and data, and queries are assigned to a different member. Throughout our progress of working on the project, issues and difficulties were exposed and we understood that there are some improvements we can do in any future group work.

The first issue is that the different thoughts and idea from group members may conflict. For instance, there are 2 possible assumptions that we made during the ER diagram designing stage regarding the entity "staff" and subtypes of staff. The first assumption is making only on-call doctors, nurses, and other staffs as subtypes of staff since regular doctors may not have any unique or defining attributes to separate them from “other staff”. In comparison, the other assumption aimed to make all kinds of doctors and nurses as subtypes of staff in order to assure consistency for both data and tables. A significant amount of negotiation and discussion occurred through our group communication tool “messenger” aimed to address the issue. Technically both assumptions worked, but the second assumption was selected in the final discussion due to a defining attribute being found for all types of doctors and nurses which separate them from other staff. And, tables derived by second assumption may be suitable for other hospitals to use as well.

The other issue outside of the content of the project is the time conflicting between team members. The start time of the project was week 4. As a result, a meeting was held on the Monday of week 4 in the library and draft of the ER diagram was made. However, in the next several weeks, no meeting has been done since group members have different time schedule for the class and many mid-term exam and test are placed within week 5-6. Therefore, finding a time that fits everyone’s schedule in the group to meet up is one of the hardest issues we faced. In this case, nearly no work was done from week 5 to 6 and we are behind many other groups. Fortunately, no severe issue is caused by this problem since many ideas and thoughts are shared via messenger and in week 7 two meetings were held on Monday and Thursday in order to follow up the progress. Therefore, much of the work was done during the week 7.

Since trimester has been applied, it leads to the difficulty of time management. For instance, the working intensity has been largely increased. All the team members must conduct multiple assignments or projects besides this group project at the same time. Based on the above situation, one of the most important aspects of time management might be finding a balance between time spent on each assignment or projects, while allowing enough time to complete each assignment. For our group, at first, the leader gathered each team member’s timetable to arrange the meeting time. However, since this project is not the only assignment that group members focus on, it is hard to figure out a meeting time that suits everyone. Thus, after discussion, our group decided to use face to face plus social media meetings rather than only using traditional face to face meetings, which allows the team member who cannot attend the meeting to join the discussion. During our work progress, for better time management the project was split into several stages, and every stage contains several tasks which were allocated to each member. After assigning tasks to each team member, the team leader also marked due dates for each stage’s tasks to ensure everyone is keeping up with our schedule. The reason for doing this is to leave small pressure and offer an opportunity for team members to arrange their time. Also, it prevents team members from leaving all their work till the very end, which would negatively affect the quality of work.

From this project, all team members have learned that when doing a complex group project, it is more efficient for group members to split and allocate tasks. Moreover, it is better for group members to have a clear aim of when to finish each stage’s task. Also, since we have multiple projects or assignments to deal with, our time needs to be reasonably allocated to different tasks according to workload. Another useful lesson we learned from this project is to not leave work until the last minute. In future group projects, we will make a clear structure about the project and efficiently allocate tasks to team members.

References

- Stakeholdermap.com. (2019). *Stakeholders in Hospitals | list stakeholders*. [online] Available at: https://www.stakeholdermap.com/stakeholders-in-hospitals.html [Accessed 15 Jul. 2019].
- Support.office.com. (2019). *Database design basics*. [online] Available at: https://support.office.com/en-ie/article/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5 [Accessed 14 Jul. 2019].