



Module Code & Module Title

CC5051NA: Databases

Assessment Weightage & Type

50% Individual Coursework

Semester

2019 Autumn

Student Name: Samjhana Ghatani

London Met ID: 18028931

College ID: np05cp4a180055

Assignment Due Date: 30th Dec 2019

Assignment Submission Date: 30th Dec 2019

Academic Supervisor: Mr. Lekhnath Katuwal

Word Count (Where Required):

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

1. Introduction	1
1.1. Introduction of the Hospital.....	1
1.2. Current Business Activities and Operation	1
1.3. Business Rules	1
1.4. Identification of Entities and Attributes	2
1.4.1. Lists of the created objects- Entities and Attributes	2
1.4.2. Identification and representation of Primary keys, Foreign Keys.	3
1.4.3. Entity Relationship Diagram	4
2. Normalization	5
2.1. Normalization from UNF to 3NF	5
2.2. Final Entity Relationship Diagram	7
2.3. Assumptions.....	8
3. Implementation.....	9
3.1. Relations Creation Order	9
3.2. Relations drop order.....	10
3.3. Create Statements for each relations.....	13
3.4. Insert Statements	16
3.5. Select Statements	23
3.6. Information Queries.....	31
3.7. Transaction Queries.....	33
3.8. Creating Dump File	35
4. Critical Evaluation	36
5. Critical Assessment of Coursework.....	37
6.References.....	38

Figure 1 ER Diagram before Normalization.....	4
Figure 2 Final ER diagram after Norma.....	7
Figure 3 Ward dropped	10
Figure 4 Appointment dropped	10
Figure 5 Patient dropped.....	10
Figure 6 Uncertified dropped.....	10
Figure 7 Uncertified dropped.....	10
Figure 8 Staff dropped.....	11
Figure 9 Address_Phone_No dropped	11
Figure 10 Address_Fax dropped	11
Figure 11 Address_Mail dropped	11
Figure 12 Address_Person dropped.....	11
Figure 13 Address dropped.....	12
Figure 14 Person_Cellphoneno dropped.....	12
Figure 15 Person_Email dropped.....	12
Figure 16 Person dropped.....	12
Figure 17 User Creation	13
Figure 18 Person Table generation	13
Figure 19 Person_Email Table generation	13
Figure 20 Person_Cellphoneno Table generation	13
Figure 21 Address Table generation	14
Figure 22 Address_Person Table generation	14
Figure 23 Address_Mail Table generation.....	14
Figure 24 Address_Fax Table generation	14
Figure 25 Address_Phone_no Table generation	14
Figure 26 Staff Table generation	15
Figure 27 Certified Table generation	15
Figure 28 Uncertified Table generation	15
Figure 29 Patient Table generation	15
Figure 30 Appointment Table generation	15
Figure 31 Ward table generation	15
Figure 32 Person Table Insertion	16
Figure 33 Person_Email Insertion	16
Figure 34 Person_Cellphoneno Insertion	17
Figure 35 Address Insertion	17
Figure 36 Address_Person Insertion	18
Figure 37 Address_Mail Insertion.....	18
Figure 38 Address_Fax Insertion	19
Figure 39 Address_Phone_no Insertion	19
Figure 40 Staff Insertion	20
Figure 41 Certified Insertion	20
Figure 42 Uncertified Insertion	21
Figure 43 Patient Insertion	21

Figure 44 Appointment Insertion	22
Figure 45 Ward Insertion	22
Figure 46 Selection Person	23
Figure 47 Selection Person_Email	24
Figure 48 Selection Person_Cellphoneno	24
Figure 49 Seleciont Address	25
Figure 50 Selection Address_Person	25
Figure 51 Selection Address_Mail	26
Figure 52 Selection Address_Fax	26
Figure 53 Selection Address_Phone_No	27
Figure 54 Selection Staff	28
Figure 55 Selection Certified	28
Figure 56 Selection Uncertified	29
Figure 57 Selection Patient	29
Figure 58 Selection Appointment	30
Figure 59 Selection Ward	30
Figure 60 Information Query 1	31
Figure 61 Certified doctor with their Appointment and Salary	32
Figure 62 Uncertified doctors with their Appointment and Salary	33
Figure 63 Certified and Uncertified staff who have conducted an appointment on a given date	34
Figure 64 Dump File Creation	35

1. Introduction

1.1. Introduction of the Hospital

Bir Hospital is the Oldest Hospital of Nepal. It was established in 1947 BS by Bir Shamsheer Jung Bahadur Rana. It is situated in Kathmandu, which is also the Capital of Nepal. When this hospital was established, it was a seven bed hospital with 5 staffs. But today it has a history of 125 years of establishment and also the first hospital of Nepal where there is practice of modern medicine commenced. Currently, this hospital is enrolling around 100 MD, MS and MDS residents each year. The hospital is now known as one of the famous hospital, not only in Nepal, but also it is known as all over the world due to its hard labour and fame. This hospital provides all the services like surgery, operation and many other and has saved the life of many people. The objective of this hospital is to provide higher education in the field of Medicine, to produce skilled and knowledgeable human resources for the treatment of general people. (NAMS, 2014)

1.2. Current Business Activities and Operation

This hospital includes eighties services including gynaecology, paediatrics and psychiatry and so on. This is the first hospital where neurosurgery was started in the times when CTscan was not around. The hospital now have more number of doctors and nurses working together to increase the name and fame of the hospital all over the world. Currently, this hospital has an easy access for the general population. Daily out patient load is about 1500. It has the 460 bed capacity. Around 5000 operations are conducted annually. The hospital has 72% free beds. (NAMS, 2014)

1.3. Business Rules

Once the appointment is booked, it can't be canceled. This the most important business policy of this hospital. If the admitted patient is certified doctor/nurse/assistant, they will get the treatment in free of cost but if the admitted patient is uncertified doctor/nurse/assistant then they will not get any privileges in the treatment and they should pay the same as other patient paid. Hospital charges are not negotiable. For any emergency appointment certain advance amount is to be deposited as per the prevalent rates for different category of rooms at the time of appointment.

1.4. Identification of Entities and Attributes

1.4.1. Lists of the created objects- Entities and Attributes

Person
Person_ID (PK)
Person_Name
Age
Gender

Address
Address_ID(PK)
Country
Province
City
Street
Street_no

Staff
Staff_ID(PK)
Staff_Type
Qualification
Salary

Certified
Certified_ID(PK)
Salary

Uncertified
Uncertified_ID(PK)
Wages

Patient
Patient_ID(PK)
Patient_Type
Blood_Group

Appointment
Appointment_ID(PK)
Treatment_Type
Treatment_Price
Appointment_Date

Ward
Ward_No(PK)
Ward_Name

1.4.2. Identification and representation of Primary keys, Foreign Keys.

Entity Name	Primary Key	Foreign Keys	Reference Tables
Person	Person_ID		
Address	Address_ID		
Staff	Staff_ID	Staff_ID	Person
Certified	Certified_ID	Staff_ID	Staff
Uncertified	Uncertified_ID	Staff_ID	Staff
Patient	Patient_ID	Person_ID, Address_ID	Person,Address
Appointment	Appointment_ID	Patient_ID, Staff_ID	Patient,Staff
Ward	Ward_No	Appointment_ID	Appointment

1.4.3. Entity Relationship Diagram

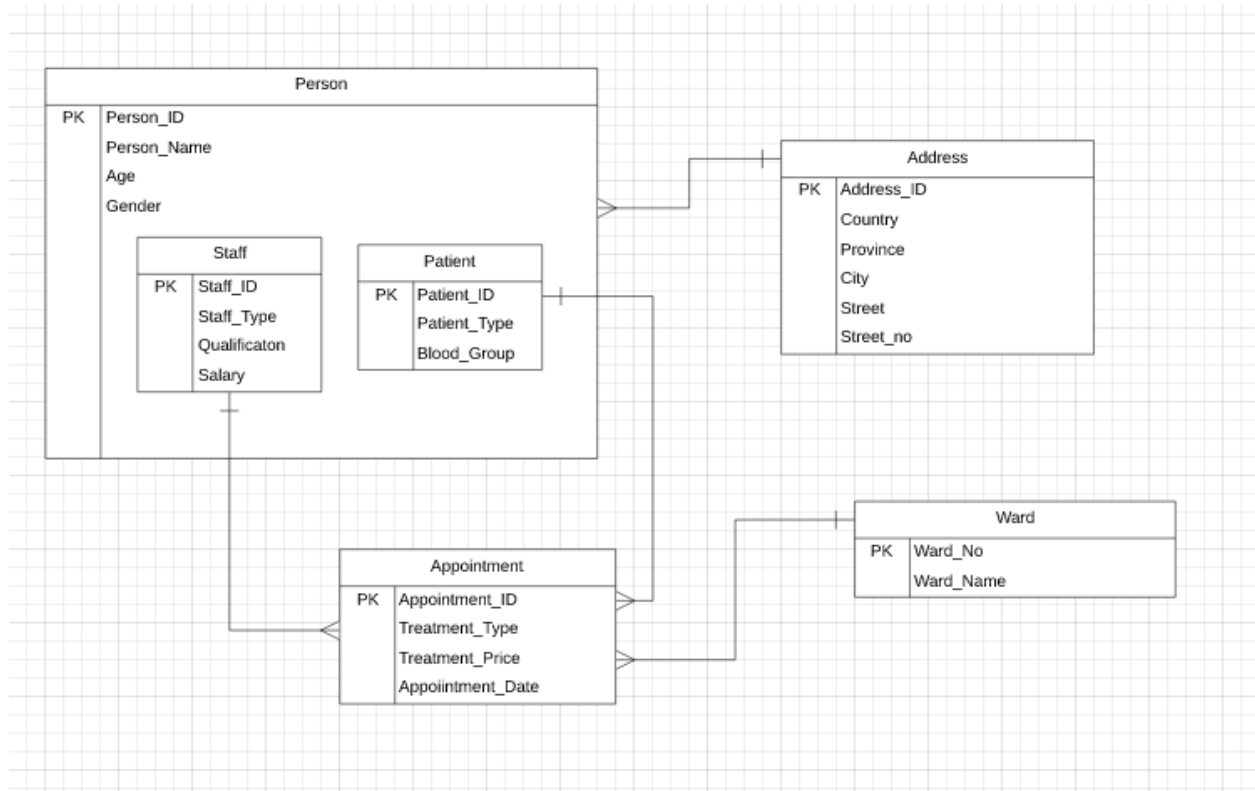


Figure 1 ER Diagram before Normalization

2. Normalization

2.1. Normalization from UNF to 3NF

- Un-Normalized Form (UNF)

- In the Un-Normalized form, all the attributes with repeating groups are included as below:

Person (Person-Id, Person-Name, Age, Gender, {E-mail}, {Contact}, {Address-Id, Country, Province, City, Street, Street-no {Mail}, {Fax-no}, {Phone-no}}, {Patient-type, Blood_Group}, {Staff-type, Qualification, Skill, Salary}), {Appointment-Id, Treatment-type, Treatment-price, Appointment-date, {Ward-no, Ward-name}})

- First Normal Form (1NF)

- In the First Normal Form, the repeating groups are removed and are formed as new entity with their attributes as shown below:

Person (Person-Id, Person-Name, Age, Gender)

Person-Email (E-mail, Person-Id*)

Person-Cellphoneno (Cellphoneno, Person-Id*)

Address (Address-Id, Country, Province, City, Street, Street-no, Person-Id*)

Address-Mail (Mail, Address-Id*)

Address-Phone (Phone-no, Address-Id*)

Address-Fax (Fax-no, Address-Id*)

Patient (Patient-type, Blood_Group, Patient-Id*)

Staff (Staff_type, Qualification, Salary, Certified, Uncertified, Staff_Id*)

Appointment (Appointment-Id, Treatment-price, Appointment-date, Patient_id*, Staff_id*)

Ward (Ward_no, Ward_Name, Appointment_Id*)

- Second Normal Form (2NF)

- In the Second Normal Form, Partial Dependency is removed from Address Entity and is further decomposed into Address_Person Entity as shown below the Address Entity.

Person_ID → Staff_ID, Patient_ID

Appointment_ID → Patient_ID, Staff_ID

Ward_No → Appointment_ID

Street_Name → Street_no

Staff_ID → Certified_ID, Uncertified_ID

Person (Person-Id, Person-Name, Age, Gender)

Person-Email (E-mail, Person-Id*)

Person-CellPhoneno (CellPhoneno, Person-Id*)

Address (Address-Id, Country, Province, City, Street, Street-no)

Address-Person (Address-Id*, Person-Id*)

Address-Mail (Mail, Address-Id*)

Address-Fax (Fax-no, Address-Id*)

Patient (Patient-type, Blood_Group, Patient_Id*)

Staff (Staff_type, Qualification, Salary, Staff_Id*)

Certified (Salary, Staff_Id*)

Uncertified (Wages, Staff_Id*)

Patient (Patient_Type, Blood_Group, Patient_ID*, Address_ID*)

Appointment (Appointment-Id, Treatment_type, Treatment-price, Appointment-date, Person_Id*)

Ward (Ward-no, Ward-name, Appointment_Id*)

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

- There is no Transitive Dependency in 2NF so 2NF is automatically in 3NF. The data are same as of 2NF.

Person (Person-Id, Person-Name, Age, Gender)

Person-Email (E-mail, Person-Id*)

Person-CellPhoneno (CellPhoneno, Person-Id*)

Address (Address-Id, Country, Province, City, Street, Street-no)

Address-Person (Address-Id*, Person-Id*)

Address-Mail (Mail, Address-Id*)

Address-Fax (Fax-no, Address-Id*)

Patient (Patient-type, Blood_Group, Patient_Id*)

Staff (Staff_type, Qualification, Salary, Staff_Id*)

Certified (Salary, Staff_Id*)

Uncertified (Wages, Staff_Id*)

Patient (Patient_Type, Blood_Group, Patient_ID*, Address_ID*)

Appointment (Appointment-Id, Treatment_type, Treatment-price, Appointment-date, Person_Id*)

Ward (Ward-no, Ward-name, Appointment_Id*)

2.2. Final Entity Relationship Diagram

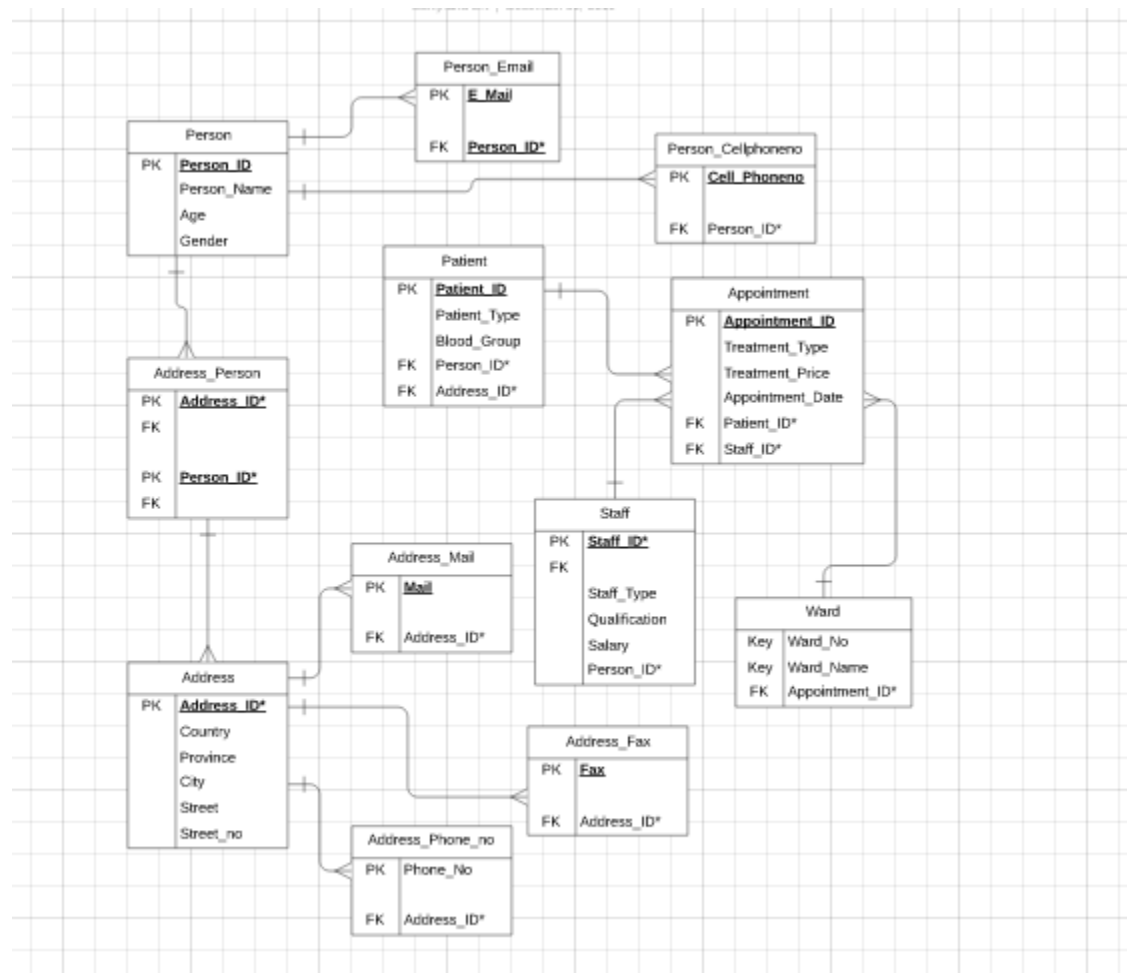


Figure 2 Final ER diagram after Norma

Above figure is the final Entity Relations Diagram after the Normalization. In the above ER diagram, entities and its relationship derived from the normalization with attributes, primary key and foreign key are represented. In the above diagram, Person, Person_Email, Person_Cellphoneno, Address, Address_Person, Address_Mail, Address_Fax, Address_Phone_No, Staff, Patient, Appointment and Ward are all the entities that we need for table creation, insertion and selection in sql. In above figure, I assumed every relation is in the form of One-to-Many. For example, one staff can have many appointment and also one patient can have many appointments. One ward can have many appointments. One person can have many phone number and one person can have many email. All these are my assumption during final ER diagram.

2.3. Assumptions

1. Person entity is the parent class of two sub classes i.e. Staff and Patient.
2. One person can have multiple addresses.
3. Address can have multiple phone number, multiple mail and multiple fax.
4. There are two kinds of Staff i.e. Certified staff and Uncertified staff. The Certified Staff if admitted as patient then they will get the treatment in free of cost whereas Uncertified staff will not get any privileges and should pay same as other patients.
5. One staff can have multiple appointments.
6. One patient can have multiple appointments.
7. One ward can have many appointments.

3. Implementation

3.1. Relations Creation Order

Person
Person_Email
Person_Cellphoneno
Address
Address_Person
Address_Mail
Address_Fax
Address_Phone_No
Staff
Certified
Uncertified
Patient
Appointment
Ward

3.2. Relations drop order

```
SQL> Drop table Ward;  
  
Table dropped.
```

Figure 3 Ward dropped

```
SQL> Drop table Appointment;  
  
Table dropped.
```

Figure 4 Appointment dropped

```
SQL> drop table Patient;  
  
Table dropped.
```

Figure 5 Patient dropped

```
SQL> drop table Uncertified;  
  
Table dropped.
```

Figure 6 Uncertified dropped

```
SQL> drop table Certified;  
  
Table dropped.
```

Figure 7 Uncertified dropped

```
SQL> drop table Staff;  
Table dropped.
```

Figure 8 Staff dropped

```
SQL> drop table Address_Phone_no;  
Table dropped.
```

Figure 9 Address_Phone_No dropped

```
SQL> drop table Address_Fax;  
Table dropped.
```

Figure 10 Address_Fax dropped

```
SQL> drop table Address_Mail;  
Table dropped.
```

Figure 11 Address_Mail dropped

```
SQL> drop table Address_Person;  
Table dropped.
```

Figure 12 Address_Person dropped

```
SQL> drop table Address;  
Table dropped.
```

Figure 13 Address dropped

```
SQL> drop table Person_CellPhoneno;  
Table dropped.
```

Figure 14 Person_Cellphoneno dropped

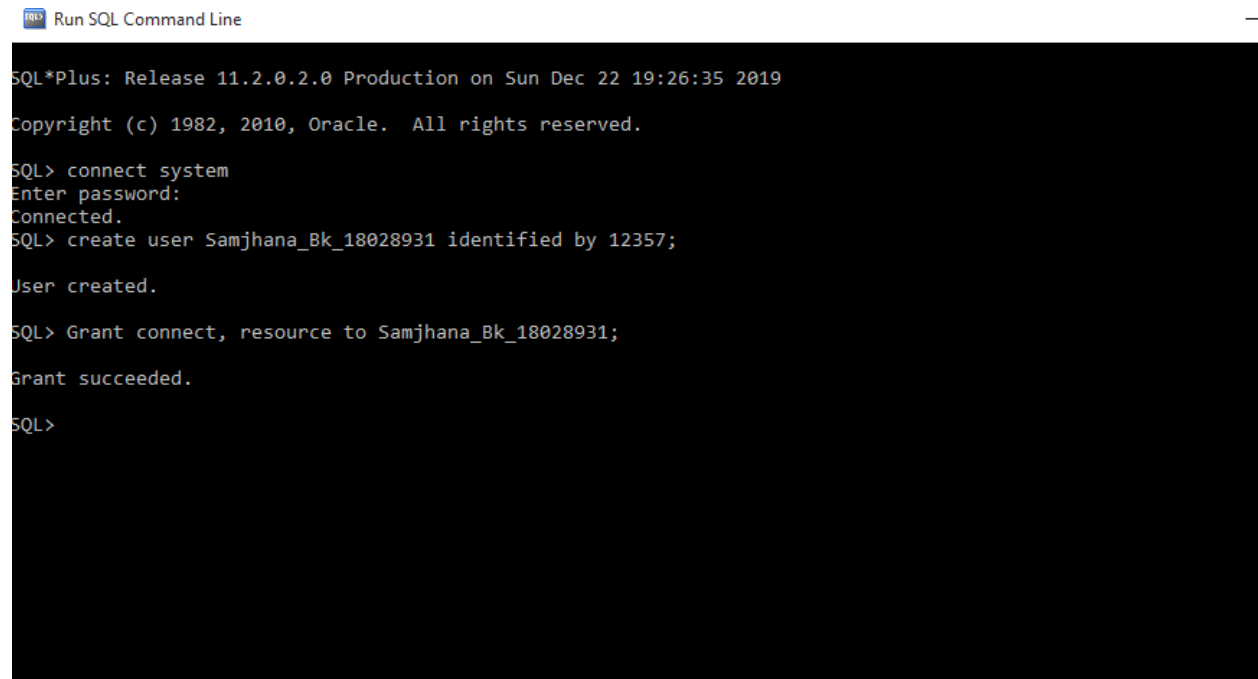
```
SQL> drop table Person_Email;  
Table dropped.
```

Figure 15 Person_Email dropped

```
SQL> drop table Person;  
Table dropped.
```

Figure 16 Person dropped

3.3. Create Statements for each relations



```

Run SQL Command Line

SQL*Plus: Release 11.2.0.2.0 Production on Sun Dec 22 19:26:35 2019

Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> connect system
Enter password:
Connected.
SQL> create user Samjhana_Bk_18028931 identified by 12357;

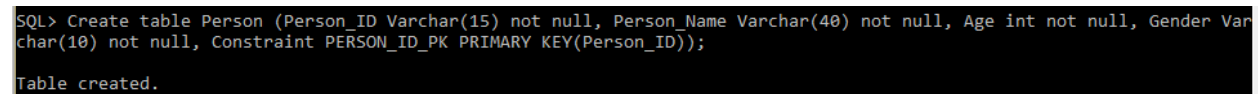
User created.

SQL> Grant connect, resource to Samjhana_Bk_18028931;

Grant succeeded.

SQL>
  
```

Figure 17 User Creation

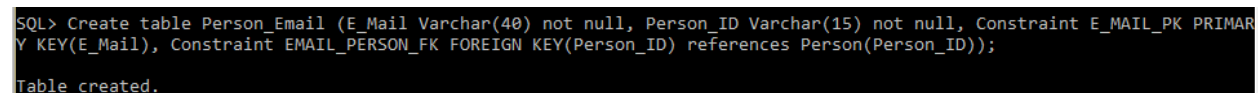


```

SQL> Create table Person (Person_ID Varchar(15) not null, Person_Name Varchar(40) not null, Age int not null, Gender Varchar(10) not null, Constraint PERSON_ID_PK PRIMARY KEY(Person_ID));

Table created.
  
```

Figure 18 Person Table generation

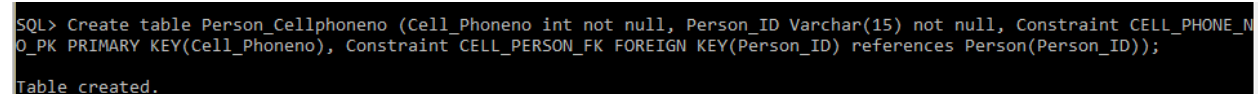


```

SQL> Create table Person_Email (E_Mail Varchar(40) not null, Person_ID Varchar(15) not null, Constraint E_MAIL_PK PRIMARY KEY(E_Mail), Constraint EMAIL_PERSON_FK FOREIGN KEY(Person_ID) references Person(Person_ID));

Table created.
  
```

Figure 19 Person_Email Table generation



```

SQL> Create table Person_Cellphoneno (Cell_Phoneno int not null, Person_ID Varchar(15) not null, Constraint CELL_PHONE_NO_PK PRIMARY KEY(Cell_Phoneno), Constraint CELL_PERSON_FK FOREIGN KEY(Person_ID) references Person(Person_ID));

Table created.
  
```

Figure 20 Person_Cellphoneno Table generation

```
SQL> Create table Address (Address_ID Varchar(15) not null, Country Varchar(20) not null, Province Varchar(15) not null, City Varchar(20) not null, Street Varchar(20) not null, Street_no Varchar(15) not null, Constraint ADDRESS_ID_PK PRIMARY KEY(Address_ID));  
Table created.
```

Figure 21 Address Table generation

```
SQL> Create table Address_Person (Address_ID Varchar(15) not null, Person_ID Varchar(15) not null, Constraint ADDRESS_PK PRIMARY KEY(Address_ID, Person_ID), Constraint ADDRESS_FK FOREIGN KEY(Address_ID) references Address(Address_ID), Constraint ADDRESS_PERSON_FK FOREIGN KEY(Person_ID) references Person(Person_ID));  
Table created.
```

Figure 22 Address_Person Table generation

```
SQL> Create table Address_Mail (Mail Varchar(40) not null, Address_ID Varchar(15) not null, Constraint MAIL_PK PRIMARY KEY(Mail), Constraint ADDRESS_MAIL_FK FOREIGN KEY(Address_ID) references Address(Address_ID));  
Table created.
```

Figure 23 Address_Mail Table generation

```
SQL> Create table Address_Fax (Fax int not null, Address_ID Varchar(15) not null, Constraint FAX_NO_PK PRIMARY KEY(Fax), Constraint ADDRESS_ID_FK FOREIGN KEY(Address_ID) references Address(Address_ID));  
Table created.
```

Figure 24 Address_Fax Table generation

```
SQL> Create table Address_Phone_no (Phone_No int not null, Address_ID Varchar(15) not null, Constraint PHONE_NO_PK PRIMARY KEY(Phone_No), Constraint ADDRESS_PHONE_FK FOREIGN KEY(Address_ID) references Address(Address_ID));  
Table created.
```

Figure 25 Address_Phone_no Table generation

```
SQL> Create table Staff (Staff_ID Varchar(15) not null, Staff_Type Varchar(15) not null, Qualification Varchar(20) not null, Salary int not null, Constraint STAFF_ID_PK PRIMARY KEY(Staff_ID), Constraint STAFF_ID_FK FOREIGN KEY(Staff_ID) references Person(Person_ID));  
Table created.
```

Figure 26 Staff Table generation

```
SQL> Create table Certified (Certified_ID Varchar(15) not null, Salary int not null, Constraint CERTIFIED_ID_PK PRIMARY KEY(Certified_ID), Constraint CERTIFIED_ID_FK FOREIGN KEY(Certified_ID) references Staff(Staff_ID));  
Table created.
```

Figure 27 Certified Table generation

```
SQL> Create table Uncertified (Uncertified_ID Varchar(15) not null, Wages int not null, Constraint UNCERTIFIED_ID_PK PRIMARY KEY(Uncertified_ID), Constraint UNCERTIFIED_ID_FK FOREIGN KEY(Uncertified_ID) references Staff(Staff_ID));  
Table created.
```

Figure 28 Uncertified Table generation

```
SQL> Create table Patient (Patient_ID Varchar(15) not null, Patient_Type Varchar(20) not null, Blood_Group Varchar(15) not null, Person_ID Varchar(15) not null, Address_ID Varchar(15) not null, Constraint PATIENT_ID_PK PRIMARY KEY(Patient_ID), Constraint PATIENT_ID_FK FOREIGN KEY(Person_ID) references Person(Person_ID), Constraint PATIENT_ADDRESS_FK FOREIGN KEY(Address_ID) references Address(Address_ID));  
Table created.
```

Figure 29 Patient Table generation

```
SQL> Create table Appointment (Appointment_ID Varchar(15) not null, Treatment_Type Varchar(20) not null, Treatment_Price int not null, Appointment_Date Date not null, Staff_ID Varchar(15) not null, Patient_ID Varchar(15) not null, Constraint APPOINTMENT_ID_PK PRIMARY KEY(Appointment_ID), Constraint PATIENT_ID_FK FOREIGN KEY(Patient_ID) references Patient(Patient_ID), Constraint STAFF_ID_FK FOREIGN KEY(Staff_ID) references Staff(Staff_ID));  
Table created.
```

Figure 30 Appointment Table generation

```
SQL> Create table Ward (Ward_No Varchar(15) not null, Ward_Name Varchar(20) not null, Appointment_ID Varchar(15) not null, Constraint WARD_NO_PK PRIMARY KEY(Ward_No), Constraint APPOINTMENT_ID_FK FOREIGN KEY(Appointment_ID) references Appointment(Appointment_ID));  
Table created.
```

Figure 31 Ward table generation

3.4. Insert Statements

```

SQL> INSERT all
  2 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('115', 'Biju Shrestha', '21', 'Female')
  3 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('953', 'Anish Sapkota', '40', 'Male')
  4 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('527', 'Amisha Baraily', '41', 'Female')
  5 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('172', 'Ankit Shrestha', '25', 'Male')
  6 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('224', 'Sagar Sharma', '46', 'Male')
  7 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('390', 'Deepika Giri', '45', 'Female')
  8 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('705', 'Deep Moktan', '44', 'Male')
  9 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('465', 'Preety Shah', '41', 'Female')
 10 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('895', 'Nisha Ghatani', '50', 'Female')
 11 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('361', 'Yugal Bhujel', '23', 'Male')
 12 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('780', 'Barsha Basnet', '39', 'Female')
 13 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('665', 'Amrita Dhamala', '33', 'Female')
 14 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('126', 'Mandip Shrestha', '23', 'Male')
 15 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('239', 'Bipan Gurung', '28', 'Male')
 16 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('129', 'Alish Rai', '22', 'Male')
 17 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('771', 'Alisha Magar', '21', 'Female')
 18 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('235', 'Rohan Khadka', '20', 'Male')
 19 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('931', 'Malvika Khadka', '27', 'Female')
 20 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('157', 'Nilima Khadka', '50', 'Female')
 21 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('871', 'Nirmal Khadka', '23', 'Male')
 22 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('991', 'Namrata Gurung', '26', 'Female')
 23 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('701', 'Bishwa Khanal', '29', 'Male')
 24 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('105', 'Biswas Rai', '23', 'Male')
 25 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('209', 'Radha Koirala', '22', 'Female')
 26 into Person (Person_ID, Person_Name, Age, Gender) VALUES ('321', 'Raman raut', '33', 'Male')
 27 Select * from dual;

25 rows created.

```

Figure 32 Person Table Insertion

```

SQL> INSERT all
  2 into Person_Email (E_Mail, Person_ID) VALUES ('rautraman123@gmail.com', '321')
  3 into Person_Email (E_Mail, Person_ID) VALUES ('koiralaradha@gmail.com', '209')
  4 into Person_Email (E_Mail, Person_ID) VALUES ('biswasrai120@gmail.com', '105')
  5 into Person_Email (E_Mail, Person_ID) VALUES ('dhamalaamrita001@gmail.com', '665')
  6 into Person_Email (E_Mail, Person_ID) VALUES ('alishrai134@gmail.com', '129')
  7 into Person_Email (E_Mail, Person_ID) VALUES ('khanalbishwa634@gmail.com', '701')
  8 into Person_Email (E_Mail, Person_ID) VALUES ('gurungnamrata631@gmail.com', '991')
  9 into Person_Email (E_Mail, Person_ID) VALUES ('bhujelyugal01@gmail.com', '361')
 10 into Person_Email (E_Mail, Person_ID) VALUES ('girideepika212@gmail.com', '390')
 11 into Person_Email (E_Mail, Person_ID) VALUES ('barshabasnet202@gmail.com', '780')
 12 into Person_Email (E_Mail, Person_ID) VALUES ('ghataninisha292@gmail.com', '895')
 13 into Person_Email (E_Mail, Person_ID) VALUES ('bijushrestha22@gmail.com', '115')
 14 into Person_Email (E_Mail, Person_ID) VALUES ('anishsapkota12@gmail.com', '953')
 15 into Person_Email (E_Mail, Person_ID) VALUES ('shresthaankit112@gmail.com', '172')
 16 Select * from dual;

14 rows created.

```

Figure 33 Person_Email Insertion

```

SQL> INSERT all
2 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9827340197, '321')
3 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9819348281, '871')
4 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9810363572, '701')
5 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9815305130, '931')
6 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9812362542, '157')
7 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9815353527, '115')
8 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9810489977, '527')
9 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9842476936, '105')
10 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9842184081, '390')
11 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9804045666, '172')
12 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9816380984, '953')
13 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9819046670, '780')
14 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9815398926, '361')
15 into Person_Cellphoneno (Cell_Phoneno, Person_ID) VALUES (9800917455, '705')
16 Select * from dual;

14 rows created.

```

Figure 34 Person_Cellphoneno Insertion

```

SQL> INSERT all
2 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_1', 'Nepal', 'Province-1', 'Itahari', 'Aitabare', 05)
3 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_2', 'Nepal', 'Province-1', 'Dharan', 'MilanPath', 15)
4 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_3', 'Nepal', 'Province-2', 'Siraha', 'Lahan', 10)
5 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_4', 'Nepal', 'Province-2', 'Saptari', 'Rajbiraj', 16)
6 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_5', 'Nepal', 'Province-2', 'Janakpur', 'Jaleshwar', 11)
7 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_6', 'Nepal', 'Province-3', 'Bhaktapur', 'Baagishwori', 9)
8 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_7', 'Nepal', 'Province-3', 'Hetauda', 'Lalbandi', 12)
9 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_8', 'Nepal', 'Province-3', 'Kathmandu', 'Chaabil', 13)
10 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_9', 'Nepal', 'Province-3', 'Kathmandu', 'Jorpati', 15)
11 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_10', 'Nepal', 'Province-3', 'Bhaktapur', 'Siddha Pokhari', 16)
12 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_11', 'Nepal', 'Province-3', 'Bhaktapur', 'Siddha Pokhari', 12)
13 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_12', 'Nepal', 'Province-1', 'Dharan', 'Bhanuchowk', 14)
14 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_13', 'Nepal', 'Province-1', 'Dharan', 'Budhasubba Chowk', 13)
15 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_14', 'Nepal', 'Province-1', 'Dharan', 'Annapurna Chowk', 18)
16 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_15', 'Nepal', 'Province-1', 'Dharan', 'Smritipath', 19)
17 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_16', 'Nepal', 'Province-1', 'Biratnagar', 'Netachowk', 11)
18 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_17', 'Nepal', 'Province-1', 'Kantepokhari', 'Harakpur', 01)
19 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_18', 'Nepal', 'Province-1', 'Biratnagar', 'Mahendrachowk', 14)
20 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_19', 'India', 'UttarPradesh', 'Jaipur', 'Rajasthan', 19)
21 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_20', 'India', 'Delhi', 'VijayNagar', 'Shivagange', 04)
22 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_21', 'Nepal', 'Province-1', 'Dharan', 'Bhotepool', 08)
23 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_22', 'Nepal', 'Province-4', 'Tanahun', 'Ringroad', 06)
24 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_23', 'Nepal', 'Province-1', 'Biratnagar', 'Bargaachi', 09)
25 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_24', 'Nepal', 'Province-1', 'Dharan', 'Bargaachi', 13)
26 into Address (Address_ID, Country, Province, City, Street, Street_no) VALUES ('Ad_25', 'Nepal', 'Province-1', 'Dharan', 'Laangali Chowk', 19)
27 Select * from dual;

25 rows created.

```

Figure 35 Address Insertion

```

SQL> INSERT all
  2 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_1', '115')
  3 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_3', '953')
  4 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_2', '780')
  5 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_5', '895')
  6 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_6', '771')
  7 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_7', '157')
  8 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_10', '991')
  9 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_8', '224')
 10 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_9', '172')
 11 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_11', '321')
 12 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_14', '209')
 13 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_19', '701')
 14 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_21', '871')
 15 into Address_Person (Address_ID, Person_ID) VALUES ('Ad_25', '129')
 16 Select * from dual;

14 rows created.

```

Figure 36 Address_Person Insertion

```

SQL> INSERT all
  2 into Address_Mail (Mail, Address_ID) VALUES ('biju.shrestha340@gmail.com', 'Ad_1')
  3 into Address_Mail (Mail, Address_ID) VALUES ('dhamala.amrita31@gmail.com', 'Ad_9')
  4 into Address_Mail (Mail, Address_ID) VALUES ('nirmal.khadka@gmail.com', 'Ad_10')
  5 into Address_Mail (Mail, Address_ID) VALUES ('rai.biswas11@gmail.com', 'Ad_13')
  6 into Address_Mail (Mail, Address_ID) VALUES ('raut.raman66@gmail.com', 'Ad_20')
  7 into Address_Mail (Mail, Address_ID) VALUES ('namrata.gurung16@gmail.com', 'Ad_23')
  8 into Address_Mail (Mail, Address_ID) VALUES ('preety.shah@gmail.com', 'Ad_18')
  9 into Address_Mail (Mail, Address_ID) VALUES ('sagar.sharma109@gmail.com', 'Ad_11')
 10 into Address_Mail (Mail, Address_ID) VALUES ('yugal.bhujel19@gmail.com', 'Ad_2')
 11 into Address_Mail (Mail, Address_ID) VALUES ('khadka.nilima29@gmail.com', 'Ad_4')
 12 into Address_Mail (Mail, Address_ID) VALUES ('khadka.malvika99@gmail.com', 'Ad_8')
 13 into Address_Mail (Mail, Address_ID) VALUES ('moktan.deep90@gmail.com', 'Ad_6')
 14 into Address_Mail (Mail, Address_ID) VALUES ('amisha.baraily40@gmail.com', 'Ad_12')
 15 into Address_Mail (Mail, Address_ID) VALUES ('bipan.gurung10@gmail.com', 'Ad_17')
 16 Select * from dual;

14 rows created.

```

Figure 37 Address_Mail Insertion


```

SQL> INSERT all
  2 into Address_Fax (Fax, Address_ID) VALUES (4567-988, 'Ad_1')
  3 into Address_Fax (Fax, Address_ID) VALUES (4122-908, 'Ad_2')
  4 into Address_Fax (Fax, Address_ID) VALUES (3457-097, 'Ad_3')
  5 into Address_Fax (Fax, Address_ID) VALUES (2467-897, 'Ad_4')
  6 into Address_Fax (Fax, Address_ID) VALUES (2890-707, 'Ad_5')
  7 into Address_Fax (Fax, Address_ID) VALUES (1290-553, 'Ad_6')
  8 into Address_Fax (Fax, Address_ID) VALUES (1288-210, 'Ad_7')
  9 into Address_Fax (Fax, Address_ID) VALUES (1234-177, 'Ad_8')
 10 into Address_Fax (Fax, Address_ID) VALUES (1281-105, 'Ad_9')
 11 into Address_Fax (Fax, Address_ID) VALUES (1780-013, 'Ad_10')
 12 into Address_Fax (Fax, Address_ID) VALUES (1672-075, 'Ad_11')
 13 into Address_Fax (Fax, Address_ID) VALUES (1632-805, 'Ad_12')
 14 into Address_Fax (Fax, Address_ID) VALUES (1122-816, 'Ad_13')
 15 into Address_Fax (Fax, Address_ID) VALUES (1210-111, 'Ad_14')
 16 Select * from dual;

14 rows created.

```

Figure 38 Address_Fax Insertion

```

SQL> INSERT all
  2 into Address_Phone_no (Phone_No, Address_ID) VALUES (9819398380, 'Ad_18')
  3 into Address_Phone_no (Phone_No, Address_ID) VALUES (9803230654, 'Ad_17')
  4 into Address_Phone_no (Phone_No, Address_ID) VALUES (9811022680, 'Ad_13')
  5 into Address_Phone_no (Phone_No, Address_ID) VALUES (9816396728, 'Ad_11')
  6 into Address_Phone_no (Phone_No, Address_ID) VALUES (9804987991, 'Ad_10')
  7 into Address_Phone_no (Phone_No, Address_ID) VALUES (9800948838, 'Ad_1')
  8 into Address_Phone_no (Phone_No, Address_ID) VALUES (9807036605, 'Ad_3')
  9 into Address_Phone_no (Phone_No, Address_ID) VALUES (9817056665, 'Ad_2')
 10 into Address_Phone_no (Phone_No, Address_ID) VALUES (9807044888, 'Ad_20')
 11 into Address_Phone_no (Phone_No, Address_ID) VALUES (9823176436, 'Ad_25')
 12 into Address_Phone_no (Phone_No, Address_ID) VALUES (9842452403, 'Ad_22')
 13 into Address_Phone_no (Phone_No, Address_ID) VALUES (9823100513, 'Ad_12')
 14 into Address_Phone_no (Phone_No, Address_ID) VALUES (9800946949, 'Ad_14')
 15 into Address_Phone_no (Phone_No, Address_ID) VALUES (9817391792, 'Ad_16')
 16 Select * from dual;

14 rows created.

```

Figure 39 Address_Phone_no Insertion

```

SQL> INSERT all
  2 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('115', 'Doctor', 'MBBS', 45000)
  3 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('953', 'Doctor', 'MD', 40000)
  4 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('780', 'Doctor', 'Mha', 35000)
  5 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('895', 'Nurse', 'BSC Nursing', 10000)
  6 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('771', 'Nurse', 'BNS Nursing', 7000)
  7 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('157', 'Nurse', 'BNS Nursing', 6000)
  8 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('991', 'Doctor', 'MD', 49000)
  9 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('224', 'Doctor', 'MD', 48000)
 10 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('172', 'Doctor', 'MBBS', 95000)
 11 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('321', 'Doctor', 'MBBS', 99000)
 12 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('209', 'Assistant', 'Management', 4000)
 13 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('701', 'Assistant', 'Management', 4400)
 14 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('871', 'Assistant', 'Management', 5000)
 15 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('129', 'Doctor', 'MBBS', 70000)
 16 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('527', 'Doctor', 'MBBS', 80000)
 17 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('390', 'Nurse', 'BSC Nursing', 8000)
 18 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('465', 'Nurse', 'BSC Nursing', 9000)
 19 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('361', 'Assistant', 'Management', 5600)
 20 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('126', 'Doctor', 'Mha', 50000)
 21 into Staff (Staff_ID, Staff_Type, Qualification, Salary) VALUES ('105', 'Doctor', 'MBBS', 57000)
 22 Select * from dual;

20 rows created.

```

Figure 40 Staff Insertion

```

SQL> INSERT all
  2 into Certified (Certified_ID, Salary) VALUES ('115', 45000)
  3 into Certified (Certified_ID, Salary) VALUES ('953', 40000)
  4 into Certified (Certified_ID, Salary) VALUES ('780', 42000)
  5 into Certified (Certified_ID, Salary) VALUES ('895', 4000)
  6 into Certified (Certified_ID, Salary) VALUES ('771', 7000)
  7 into Certified (Certified_ID, Salary) VALUES ('361', 5000)
  8 into Certified (Certified_ID, Salary) VALUES ('129', 55000)
  9 into Certified (Certified_ID, Salary) VALUES ('390', 5500)
 10 into Certified (Certified_ID, Salary) VALUES ('105', 5900)
 11 into Certified (Certified_ID, Salary) VALUES ('465', 8000)
 12 Select * from dual;

10 rows created.

```

Figure 41 Certified Insertion


```

SQL> INSERT all
  2 into Uncertified (Uncertified_ID, Wages) VALUES ('157', 4000)
  3 into Uncertified (Uncertified_ID, Wages) VALUES ('991', 25000)
  4 into Uncertified (Uncertified_ID, Wages) VALUES ('224', 30000)
  5 into Uncertified (Uncertified_ID, Wages) VALUES ('172', 35000)
  6 into Uncertified (Uncertified_ID, Wages) VALUES ('321', 39000)
  7 into Uncertified (Uncertified_ID, Wages) VALUES ('209', 3000)
  8 into Uncertified (Uncertified_ID, Wages) VALUES ('701', 2500)
  9 into Uncertified (Uncertified_ID, Wages) VALUES ('871', 2100)
 10 into Uncertified (Uncertified_ID, Wages) VALUES ('527', 45000)
 11 into Uncertified (Uncertified_ID, Wages) VALUES ('126', 41000)
 12 Select * from dual;

10 rows created.

```

Figure 42 Uncertified Insertion

```

SQL> INSERT all
  2 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA01', 'Regular', 'AB+', '115', 'Ad_1')
  3 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA02', 'New', 'AB+', '953', 'Ad_2')
  4 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA03', 'New', 'AB-', '780', 'Ad_3')
  5 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA04', 'Regular', 'AB-', '895', 'Ad_4')
  6 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA05', 'Regular', 'B-', '771', 'Ad_5')
  7 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA06', 'New', 'B-', '361', 'Ad_21')
  8 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA07', 'New', 'B+', '129', 'Ad_22')
  9 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA08', 'Regular', 'B+', '390', 'Ad_12')
 10 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA09', 'Regular', 'AB+', '105', 'Ad_13')
 11 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA10', 'Regular', 'B+', '465', 'Ad_11')
 12 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA11', 'Regular', 'O-', '465', 'Ad_7')
 13 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA12', 'New', 'O-', '157', 'Ad_8')
 14 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA13', 'New', 'A+', '991', 'Ad_9')
 15 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA14', 'Regular', 'AB-', '224', 'Ad_10')
 16 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA15', 'New', 'AB-', '172', 'Ad_19')
 17 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA16', 'Regular', 'A+', '321', 'Ad_14')
 18 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA17', 'New', 'A+', '209', 'Ad_16')
 19 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA18', 'New', 'A-', '701', 'Ad_20')
 20 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA19', 'Regular', 'B+', '871', 'Ad_23')
 21 into Patient (Patient_ID, Patient_Type, Blood_Group, Person_ID, Address_ID) VALUES ('PA20', 'Regular', 'B+', '527', 'Ad_24')
 22 Select * from dual;

20 rows created.

```

Figure 43 Patient Insertion

```

SQL> INSERT all
2 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_11', 'Video X-Ray', 3100, to_date('2017-03-23', 'YYYY-MM-DD'), 'PA01', '953')
3 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_12', 'X-Ray', 1000, to_date('2018-03-23', 'YYYY-MM-DD'), 'PA02', '780')
4 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_13', 'Health Checkup', 500, to_date('2015-05-13', 'YYYY-MM-DD'), 'PA03', '895')
5 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_14', 'Kidney Checkup', 5000, to_date('2015-05-23', 'YYYY-MM-DD'), 'PA04', '771')
6 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_15', 'Sugar Checkup', 700, to_date('2016-06-23', 'YYYY-MM-DD'), 'PA05', '157')
7 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_16', 'Blood Checkup', 200, to_date('2017-07-20', 'YYYY-MM-DD'), 'PA06', '871')
8 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_17', 'High Fever', 900, to_date('2018-09-20', 'YYYY-MM-DD'), 'PA07', '465')
9 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_18', 'Headache', 100, to_date('2019-09-12', 'YYYY-MM-DD'), 'PA08', '209')
10 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_19', 'BP Checkup', 70, to_date('2017-12-12', 'YYYY-MM-DD'), 'PA09', '321')
11 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_20', 'Body Checkup', 1000, to_date('2017-12-13', 'YYYY-MM-DD'), 'PA10', '126')
12 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_21', 'Heart Checkup', 1000, to_date('2017-10-13', 'YYYY-MM-DD'), 'PA11', '527')
13 into Appointment (Appointment_ID, Treatment_Type, Treatment_Price, Appointment_Date, Patient_ID, Staff_ID) VALUES ('APP_22', 'Endoscopy', 20000, to_date('2017-11-11', 'YYYY-MM-DD'), 'PA12', '390')
14 Select * from dual;

12 rows created.

```

Figure 44 Appointment Insertion

```

SQL> INSERT all
2 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_01', 'Emergency', 'APP_11')
3 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_02', 'General', 'APP_12')
4 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_03', 'Gyno', 'APP_12')
5 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_04', 'Cardiology', 'APP_13')
6 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_05', 'General', 'APP_11')
7 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_06', 'Emergency', 'APP_16')
8 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_07', 'Emergency', 'APP_18')
9 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_08', 'General', 'APP_19')
10 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_09', 'Gyno', 'APP_19')
11 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_10', 'Gyno', 'APP_20')
12 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_11', 'Emergency', 'APP_20')
13 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_12', 'Emergency', 'APP_21')
14 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_13', 'Cardiology', 'APP_21')
15 into Ward (Ward_No, Ward_Name, Appointment_ID) VALUES ('W_14', 'Emergency', 'APP_22')
16 Select * from dual;

14 rows created.

```

Figure 45 Ward Insertion

3.5. Select Statements

```
SQL> Select * from Person;
```

PERSON_ID	PERSON_NAME	AGE	GENDER
115	Biju Shrestha	21	Female
953	Anish Sapkota	40	Male
527	Amisha Baraily	41	Female
172	Ankit Shrestha	25	Male
224	Sagar Sharma	46	Male
390	Deepika Giri	45	Female
705	Deep Moktan	44	Male
465	Preety Shah	41	Female
895	Nisha Ghatani	50	Female
361	Yugal Bhujel	23	Male
780	Barsha Basnet	39	Female
665	Amrita Dhamala	33	Female
126	Mandip Shrestha	23	Male
239	Bipan Gurung	28	Male
129	Alish Rai	22	Male
771	Alisha Magar	21	Female
235	Rohan Khadka	20	Male
931	Malvika Khadka	27	Female
157	Nilima Khadka	50	Female
871	Nirmal Khadka	23	Male
991	Namrata Gurung	26	Female
701	Bishwa Khanal	29	Male
105	Biswas Rai	23	Male
209	Radha Koirala	22	Female
321	Raman raut	33	Male

25 rows selected.

Figure 46 Selection Person

```
SQL> Select * from Person_Email;
```

E_MAIL	PERSON_ID
rautraman123@gmail.com	321
koiralaradha@gmail.com	209
biswasrai120@gmail.com	105
dhamalaamrita001@gmail.com	665
alishrai134@gmail.com	129
khanalbishwa634@gmail.com	701
gurungnamrata631@gmail.com	991
bhujelyugal01@gmail.com	361
girideepika212@gmail.com	390
barshabasnet202@gmail.com	780
ghataninisha292@gmail.com	895
bijushrestha22@gmail.com	115
anishsapkota12@gmail.com	953
shresthaankit112@gmail.com	172

14 rows selected.

Figure 47 Selection Person_Email

```
SQL> Select * from Person_Cellphoneno;
```

CELL_PHONENO	PERSON_ID
9827340197	321
9819348281	871
9810363572	701
9815305130	931
9812362542	157
9815353527	115
9810489977	527
9842476936	105
9842184081	390
9804045666	172
9816380984	953
9819046670	780
9815398926	361
9800917455	705

14 rows selected.

Figure 48 Selection Person_Cellphoneno

```
SQL> Select * from Address;
```

ADDRESS_ID	COUNTRY	PROVINCE	CITY	STREET	STREET_NO
Ad_1	Nepal	Province-1	Itahari	Aitabare	5
Ad_2	Nepal	Province-1	Dharan	MilanPath	15
Ad_3	Nepal	Province-2	Siraha	Lahan	10
Ad_4	Nepal	Province-2	Saptari	Rajbiraj	16
Ad_5	Nepal	Province-2	Janakpur	Jaleswar	11
Ad_6	Nepal	Province-3	Bhaktapur	Baagishwori	9
Ad_7	Nepal	Province-3	Hetauda	Lalbandi	12
Ad_8	Nepal	Province-3	Kathmandu	Chaabil	13
Ad_9	Nepal	Province-3	Kathmandu	Jorpati	15
Ad_10	Nepal	Province-3	Bhaktapur	Siddha Pokhari	16
Ad_11	Nepal	Province-3	Bhaktapur	Siddha Pokhari	12
Ad_12	Nepal	Province-1	Dharan	Bhanuchowk	14
Ad_13	Nepal	Province-1	Dharan	Budhasubba Chowk	13
Ad_14	Nepal	Province-1	Dharan	Annapurna Chowk	18
Ad_15	Nepal	Province-1	Dharan	Smritipath	19
Ad_16	Nepal	Province-1	Biratchowk	Netachowk	11
Ad_17	Nepal	Province-1	Kanepokhari	Harakpur	1
Ad_18	Nepal	Province-1	Biratnagar	Mahendrachowk	14
Ad_19	India	UttarPradesh	Jaipur	Rajasthan	19
Ad_20	India	Delhi	VijayNagar	Shivagange	4
Ad_21	Nepal	Province-1	Dharan	Bhotepool	8
Ad_22	Nepal	Province-4	Tanahun	Ringroad	6
Ad_23	Nepal	Province-1	Biratnagar	Bargaachi	9
Ad_24	Nepal	Province-1	Dharan	Bargaachi	13
Ad_25	Nepal	Province-1	Dharan	Laangali Chowk	19

25 rows selected.

Figure 49 Seleciont Address

ADDRESS_ID	PERSON_ID
Ad_1	115
Ad_10	991
Ad_11	321
Ad_14	209
Ad_19	701
Ad_2	780
Ad_21	871
Ad_25	129
Ad_3	953
Ad_5	895
Ad_6	771
Ad_7	157
Ad_8	224
Ad_9	172

14 rows selected.

Figure 50 Selection Address_Person

MAIL	ADDRESS_ID
biju.shrestha34@gmail.com	Ad_1
dhamala.amrita31@gmail.com	Ad_9
nirmal.khadka@gmail.com	Ad_10
rai.biswas11@gmail.com	Ad_13
raut.raman66@gmail.com	Ad_20
namrata.gurung16@gmail.com	Ad_23
preety.shah@gmail.com	Ad_18
sagar.sharma109@gmail.com	Ad_11
yugal.bhujel19@gmail.com	Ad_2
khadka.nilima29@gmail.com	Ad_4
khadka.malvika99@gmail.com	Ad_8
moktan.deep90@gmail.com	Ad_6
amisha.baraily40@gmail.com	Ad_12
bipan.gurung10@gmail.com	Ad_17

14 rows selected.

Figure 51 Selection Address_Mail

```
SQL> Select * from Address_Fax;
```

FAX	ADDRESS_ID
3579	Ad_1
3214	Ad_2
3360	Ad_3
1570	Ad_4
2183	Ad_5
737	Ad_6
1078	Ad_7
1057	Ad_8
1176	Ad_9
1767	Ad_10
1597	Ad_11
827	Ad_12
306	Ad_13
1099	Ad_14

14 rows selected.

Figure 52 Selection Address_Fax

```
SQL> Select * from Address_Phone_No;

  PHONE_NO  ADDRESS_ID
-----
9819398380 Ad_18
9803230654 Ad_17
9811022680 Ad_13
9816396728 Ad_11
9804987991 Ad_10
9800948838 Ad_1
9807036605 Ad_3
9817056665 Ad_2
9807044888 Ad_20
9823176436 Ad_25
9842452403 Ad_22
9823100513 Ad_12
9800946949 Ad_14
9817391792 Ad_16

14 rows selected.
```

Figure 53 Selection Address_Phone_No

```
SQL> Select * from Staff;
```

STAFF_ID	STAFF_TYPE	QUALIFICATION	SALARY
115	Doctor	MBBS	45000
953	Doctor	MD	40000
780	Doctor	Mha	35000
895	Nurse	BSC Nursing	10000
771	Nurse	BNS Nursing	7000
157	Nurse	BNS Nursing	6000
991	Doctor	MD	49000
224	Doctor	MD	48000
172	Doctor	MBBS	95000
321	Doctor	MBBS	99000
209	Assistant	Management	4000
701	Assistant	Management	4400
871	Assistant	Management	5000
129	Doctor	MBBS	70000
527	Doctor	MBBS	80000
390	Nurse	BSC Nursing	8000
465	Nurse	BSC Nursing	9000
361	Assistant	Management	5600
126	Doctor	Mha	50000
105	Doctor	MBBS	57000

```
20 rows selected.
```

Figure 54 Selection Staff

```
SQL> Select * from Certified;
```

CERTIFIED_ID	SALARY
115	45000
953	40000
780	42000
895	4000
771	7000
361	5000
129	55000
390	5500
105	5900
465	8000

```
10 rows selected.
```

Figure 55 Selection Certified


```
SQL> Select * from Uncertified;
```

UNCERTIFIED_ID	WAGES
157	4000
991	25000
224	30000
172	35000
321	39000
209	3000
701	2500
871	2100
527	45000
126	41000

```
10 rows selected.
```

Figure 56 Selection Uncertified

```
SQL> Select * from Patient;
```

PATIENT_ID	PATIENT_TYPE	BLOOD_GROUP	PERSON_ID	ADDRESS_ID
PA01	Regular	AB+	115	Ad_1
PA02	New	AB+	953	Ad_2
PA03	New	AB-	780	Ad_3
PA04	Regular	AB-	895	Ad_4
PA05	Regular	B-	771	Ad_25
PA06	New	B-	361	Ad_21
PA07	New	B+	129	Ad_22
PA08	Regular	B+	390	Ad_12
PA09	Regular	AB+	105	Ad_13
PA10	Regular	B+	465	Ad_11
PA11	Regular	O-	465	Ad_7
PA12	New	O-	157	Ad_8
PA13	New	A+	991	Ad_9
PA14	Regular	AB-	224	Ad_10
PA15	New	AB-	172	Ad_19
PA16	Regular	A+	321	Ad_14
PA17	New	A+	209	Ad_16
PA18	New	A-	701	Ad_20
PA19	Regular	B+	871	Ad_23
PA20	Regular	B+	527	Ad_24

```
20 rows selected.
```

Figure 57 Selection Patient

```
SQL> Select * from Appointment;
```

APPOINTMENT_ID	TREATMENT_TYPE	TREATMENT_PRICE	APPOINTME	STAFF_ID	PATIENT_ID
APP_11	Video X-Ray	3100	23-MAR-17	953	PA01
APP_12	X-Ray	1000	23-MAR-18	780	PA02
APP_13	Health Checkup	500	13-MAY-15	895	PA03
APP_14	Kidney Checkup	5000	23-MAY-15	771	PA04
APP_15	Sugar Checkup	700	23-JUN-16	157	PA05
APP_16	Blood Checkup	200	20-JUL-17	871	PA06
APP_17	High Fever	900	20-SEP-18	465	PA07
APP_18	Headache	100	12-SEP-19	209	PA08
APP_19	BP Checkup	70	12-DEC-17	321	PA09
APP_20	Body Checkup	1000	13-DEC-17	126	PA10
APP_21	Heart Checkup	1000	13-OCT-17	527	PA11
APP_22	Endoscopy	20000	11-NOV-17	390	PA12

12 rows selected.

Figure 58 Selection Appointment

```
SQL> Select * from Ward;
```

WARD_NO	WARD_NAME	APPOINTMENT_ID
W_01	Emergency	APP_11
W_02	General	APP_12
W_03	Gyno	APP_12
W_04	Cardiology	APP_13
W_05	General	APP_11
W_06	Emergency	APP_16
W_07	Emergency	APP_18
W_08	General	APP_19
W_09	Gyno	APP_19
W_10	Gyno	APP_20
W_11	Emergency	APP_20
W_12	Emergency	APP_21
W_13	Cardiology	APP_21
W_14	Emergency	APP_22

14 rows selected.

Figure 59 Selection Ward

3.6. Information Queries

1. List all Patients, Regular and New

SQL> Select Patient_ID, Patient_Type from Patient Where Patient_Type = 'New' or Patient.Patient_Type = 'Regular';

```
SQL> Select Patient_ID, Patient_Type from Patient Where Patient_Type = 'New' or Patient.Patient_Type = 'Regular';
```

PATIENT_ID	PATIENT_TYPE
PA01	Regular
PA02	New
PA03	New
PA04	Regular
PA05	Regular
PA06	New
PA07	New
PA08	Regular
PA09	Regular
PA10	Regular
PA11	Regular
PA12	New
PA13	New
PA14	Regular
PA15	New
PA16	Regular
PA17	New
PA18	New
PA19	Regular
PA20	Regular

20 rows selected.

Figure 60 Information Query 1

2. List all Patient with all their Address

**SQL>Select Address.Address_ID, Address.Country, Patient.Patient_ID
from Patient inner join Address_Person on Patient.Patient_ID =
Address_Person.Person_ID inner join Address on
Address_Person.Address_ID = Address.Address_ID;**

3. For a given Certified doctor, find all the appointments he/she have conducted and the amount hr/she got for conducting the appointment.

SQL> select Person.Person_ID, Person.Person_Name, Staff.Staff_Type, Appointment.Treatment_Price, Appointment.Appointment_ID from Person inner join Certified on Person.Person_ID = Certified.Certified_ID inner join Staff on Certified.Certified_ID = Staff.Staff_ID inner join Appointment on Staff.Staff_ID = Appointment.Staff_ID where Staff_Type = 'Doctor';

```
SQL> select Person.Person_ID, Person.Person_Name, Staff.Staff_Type, Appointment.Treatment_Price, Appointment.Appointment_ID from Person inner join Certified on
Person.Person_ID = Certified.Certified_ID inner join Staff on Certified.Certified_ID = Staff.Staff_ID inner join Appointment on Staff.Staff_ID = Appointment.Sta
ff_ID where Staff_Type = 'Doctor';
```

PERSON_ID	PERSON_NAME	STAFF_TYPE	TREATMENT_PRICE	APPOINTMENT_ID
953	Anish Sapkota	Doctor	3100	APP_11
780	Barsha Basnet	Doctor	1800	APP_12

Figure 61 Certified doctor with their Appointment and Salary

4. List all staffs that are also a patient.

SQL> Select Person.Person_ID, Person.Person_Name, Staff.Profession, Patient.Patient_Type from Person inner join Staff on Person.Person_ID = Staff.Staff_ID inner join Patient on Staff.Staff_ID = Patient.Patient_ID;

3.7. Transaction Queries

1. List all uncertified doctors who have been attended an appointment for a treatment and the amount he/she have paid.

SQL> Select Person.Person_ID, Person.Person_Name, Staff.Staff_Type, Appointment.Treatment_Price, Appointment.Patient_ID from Person inner join Uncertified on Person.Person_ID = Uncertified.Uncertified_ID inner join Staff on Uncertified.Uncertified_ID = Staff.Staff_ID inner join Appointment on Staff.Staff_ID = Appointment.Staff_ID where Staff.Staff_Type = 'Doctor';

```
SQL> select Person_ID, Person_Name, Staff.Staff_ID, Appointment.Treatment_Price, Appointment.Appointment_ID from Person inner join Uncertified on Person.Person_ID = Uncertified.Uncertified_ID inner join Staff on Uncertified.Uncertified_ID = Staff.Staff_ID inner join Appointment on Staff.Staff_ID = Appointment.Staff_ID where Staff.Staff_Type = 'Doctor';
```

PERSON_ID	PERSON_NAME	STAFF_ID	TREATMENT_PRICE	APPOINTMENT_ID
527	Amisha Barailly	527	1000	APP_21
126	Mandip Shrestha	126	1000	APP_20
321	Raman raut	321	70	APP_19

SQL>

Figure 62 Uncertified doctors with their Appointment and Salary

2. List the appointments that have been conducted in an emergency ward.

SQL> Select Ward.Ward_No, Ward.Ward_Name, Appointment.Appointment_ID, Appointment.Treatment_Type from Ward inner join Appointment on Ward.Ward_NO = Appointment.ward_No where Ward.Ward_Name = 'Emergency';

- List all staffs (certified and uncertified) who have conducted or will conduct an appointment on a given date.

**SQL> Select Staff.Staff_ID, Person.Person_Name,
Appointment.Appointment_ID, Appointment.Appointment_Date from
Person inner join Staff on Person.Person_ID = Staff.Staff_ID inner join
Appointment on Staff.Staff_ID = Appointment.Staff_ID where
Appointment.Appointment_Date = '13-MAY-15';**

```
SQL> Select Staff.Staff_ID,Person.Person Name,Appointment.Appointment_ID,Appointment.Appointment Date from Person inner join Staff on Person.Person_ID=Staff.Sta
ff_ID inner join Appointment on Staff.Staff_ID=Appointment.Staff_ID where Appointment.Appointment_Date = '13-MAY-15';
```

STAFF_ID	PERSON_NAME	APPOINTMENT_ID	APPOINTME
895	Nisha Ghatani	APP_13	13-MAY-15

Figure 63 Certified and Uncertified staff who have conducted an appointment on a given date

- List all patients booked for an appointment on a given date.

**SQL> Select Patient.Patient_ID, Person.Person_Name,
Appointment.Appointment_ID, Appointment.Appointment_Date from
Person inner join Patient on Person.Person_ID = Patient.Patient_ID inner
join Appointment on Patient.Patient_ID = Appointment.Patient_ID where
Appointment_Date = '20-SEP-18';**

3.8. Creating Dump File

```

Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\dell\Desktop\Database CW>exp Samjhana_Bk_18028931/12357 file = "Samjhana dmp"

Export: Release 11.2.0.2.0 - Production on Mon Dec 30 11:47:30 2019

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Express Edition Release 11.2.0.2.0 - Production
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
server uses AL32UTF8 character set (possible charset conversion)
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user SAMJHANA_BK_18028931
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user SAMJHANA_BK_18028931
About to export SAMJHANA_BK_18028931's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export SAMJHANA_BK_18028931's tables via Conventional Path ...
. . exporting table ADDRESS 25 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table ADDRESS_FAX 14 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table ADDRESS_MAIL 14 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table ADDRESS_PERSON 14 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table ADDRESS_PHONE_NO 14 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table APPOINTMENT 12 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table CERTIFIED 10 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table PATIENT 20 rows exported
EXP-00091: Exporting questionable statistics.

```

Figure 64 Dump File Creation

4. Critical Evaluation

The Coursework was about developing the database for Patient Recording System for Hospital. Before starting the System, I was in dilemma that whether I could finish all those tasks on time or not. I had faced a lot of obstacles during the assignment. Because I haven't used the SQL commands till now.

Before starting the coursework, I had collected some research about the Hospital which could be helpful for me in the assignment and it did too. I got to know about the past and present status of the hospital. I figure out some entities and attributes. Person, Address, Staff, Patient, Certified, Uncertified, Appointment, Ward are its entities whereas attributes are like Person_ID, Patient_ID, Person_Name, Appointment_ID, Ward_name and so on. The relations between the entities were determined before the Normalization. After the normalization, the tables creation and insertion seems very difficult to do. It took me about 5-6 days to finalize the final tables name and the values to be inserted in the tables. I also did research on the queries commands in sql.

While Normalizing into 1 NF, all the repeated groups were separated. Normalizing on to 2 NF, partial dependencies were separated and finally transitive dependencies were removed. The normalized entities were entered in database as tables which were there and they are populated with respective values. Entity Relations diagrams were now developed for better understanding of the database model. After the sql queries completed database was tested with the queries which was given in the coursework-questions.

Lastly, I have completed the task on time. My experience in doing this assignment was very fruitful and I got to know many important things needed in solving the queries in sql. From this assignment, I learn that errors are not actually the problems but are the staircase to success. Many problems emerged during the assignment but I tried my best to tackle with all those obstacles.

5. Critical Assessment of Coursework

This System describes about the Patient Recording System for Hospital named Bir Hospital, which is the oldest hospital of Nepal and now one of the popular hospital all over the world. Before starting the coursework, I research about this hospital. I learnt about the past and present conditions of this hospital. I am not perfect in database and I am just learning it. But I felt quite satisfaction that I completed the coursework on given deadline by learning about the implementation of tools and as well referencing websites online. I got the knowledge to manage the entity and attributes during designing this system which are required in the coursework. While I was doing the coursework, I got to collect lots of information about the SQL commands and also my improvement in using SQL server increases. I enjoyed a lot with doing the coursework.

The database module is very much related with our software engineering module. The common parts between the database module and SE module are ER diagrams and Entity development. Due to this, it was easier for me in creation of entity and attributes. The information which I gather while doing the coursework will help me in my further career developing as database designer.

References

- NAMS, 2014. *NAMS*. [Online]
Available at: <https://nams.org.np/about.php>
[Accessed 26 12 2019].
- NAMS, 2014. *NAMS*. [Online]
Available at: <https://nams.org.np/about.php>
[Accessed 27 12 2019].