

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



DATABASE SYSTEMS LABORATORY

Assignment: Database Design

HOSPITAL DATABASE

Instructor:	Phan Trọng Nhân	
Students:	Nguyễn Đình Bảo Phúc	1852068
	Hoàng Nhật Quang	1852691
	Hoàng Cao Quốc Thắng	2050020
	Hoàng Trần Việt Long	1652350
	Trần Lê Nhật Quang	1752443

HO CHI MINH CITY, OCTOBER 2021



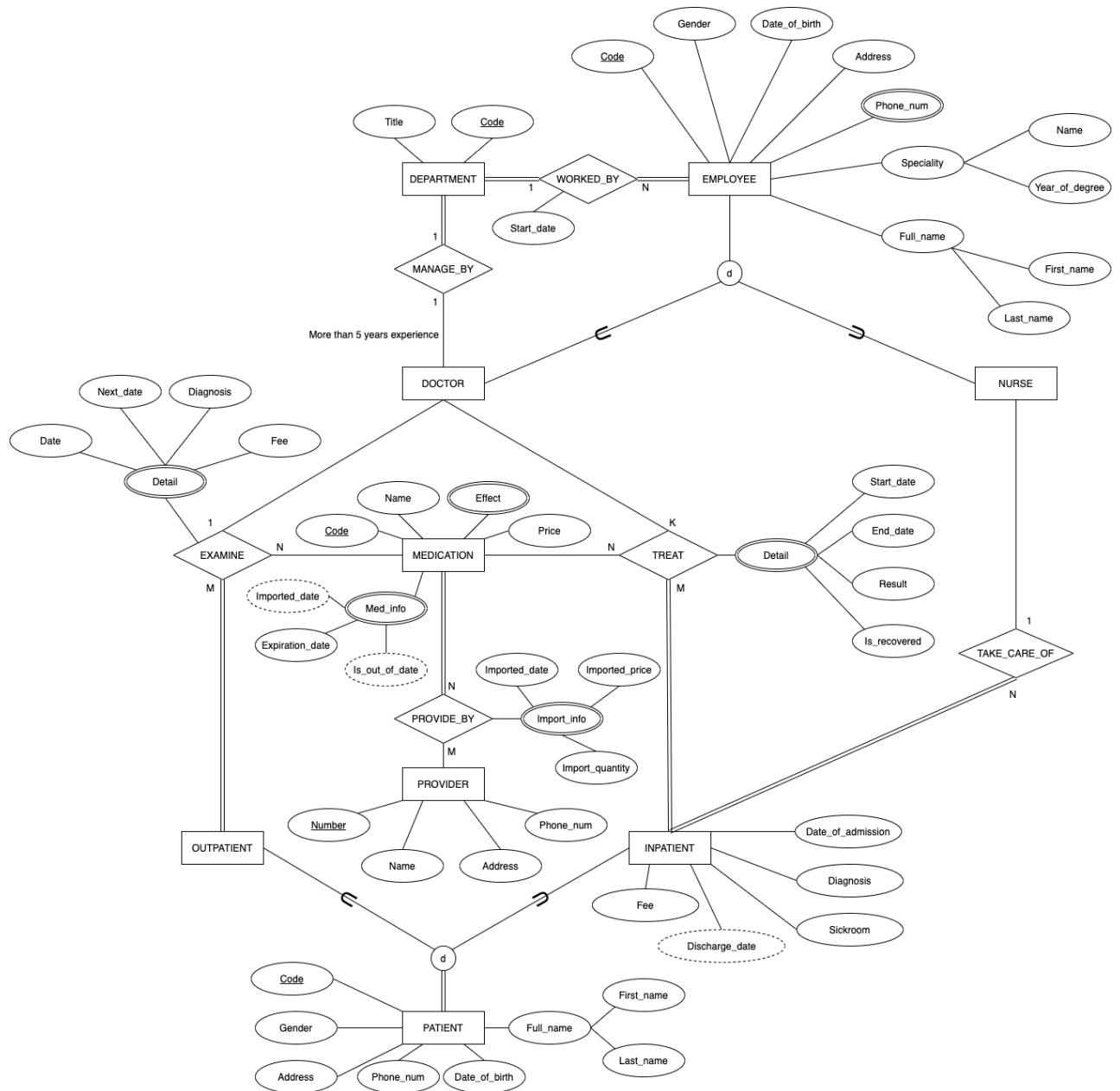
Contents

1	Assignment 1	2
1.1	Entity Relationship diagram	2
1.1.1	Enhance Entity Relationship diagram for Hospital database.	2
1.1.2	Description for Enhance Entity Relationship diagram.	3
1.2	Relational schema	5
1.2.1	Mapping ERD of Hospital database to relational model	5
1.2.2	Description for the relational model above	6
1.3	Unidentified Constraints	8
2	Assignment 2	8
2.1	Part 1: Physical Database Design	8
2.1.1	SQL statement to create database and insert values.	8
2.1.2	Explanation for database	19
2.1.2.a	EMPLOYEE	19
2.1.2.b	EMPLOYEE_PHONE_NUMBER	20
2.1.2.c	DEPARTMENT	21
2.1.2.d	OUTPATIENT TABLE	22
2.1.2.e	EXAMINE TABLE	23
2.1.2.f	EXAMINE_DETAIL TABLE	24
2.1.2.g	INPATIENT TABLE	27
2.1.2.h	TREAT TABLE	28
2.1.2.i	TREATMENT_DETAIL TABLE	29
2.1.2.j	PROVIDER TABLE	31
2.1.2.k	MEDICATION TABLE	31
2.1.2.l	IMPORTED_FROM TABLE	32
2.1.2.m	IMPORTED_INFO TABLE	33
2.1.2.n	MED_EFFECT TABLE	34
2.1.2.o	MED_INFO TABLE	35
2.2	Part 2: Store Procedure / Function / SQL	37
2.2.1	Question a: Increase Inpatient Fee to 10% for all the inpatients who are admitted to hospital from 01/09/2020	37
2.2.2	Question b: Select all the patients (outpatient & inpatient) of the doctor named 'Nguyen Van A'	39
2.2.3	Question c: Write a function to calculate the total medication price a pa- tient has to pay for each treatment or examination	40
2.2.4	Question d: Write a procedure to sort the doctor in increasing number of patients he/she takes care in a period of time	44
2.3	Part 3: Building Applications	45

1 Assignment 1

1.1 Entity Relationship diagram

1.1.1 Enhance Entity Relationship diagram for Hospital database.



1.1.2 Description for Enhance Entity Relationship diagram.

The Enhance Entity Relationship diagram for the Database of a hospital X includes:

- Super-class EMPLOYEE consists of these attributes: a unique code as the key attribute, full name consisting of first name and last name, date of birth, gender, address, start date (first day of work), phone number(s), and speciality with its related name and degree's year.
- Super-class EMPLOYEE is also generalized into 2 sub-classes: DOCTOR and NURSE which are disjoint, each has attributes that inherits from EMPLOYEE.
- The strong entity DEPARTMENT include these attributes: unique code as its key attribute, a title, and a dean who is a doctor. The relationship between DEPARTMENT and EMPLOYEE is WORKED_BY, which has the constraint is that the employees have to belong to a specific department and a department has at least one or many employees.
- Our group specify the relationship between DEPARTMENT and DOCTOR is MANAGED_BY to express the attribute "dean" and the constraint is 1 to 1. There is a special note that the dean must hold a specific speciality and has had more than 5 years of experience, which can not describe by (E)ERD. This will be analyzed in section three of the report.
- The information of a MEDICATION is also describe in (E)ERD. This information consists of a unique code as the key attribute, name of the medication, effects, price, expiration date, and derived attribute is_out_of_date. A medication is provided by one or more providers, and one provider may provide many types of medication. The entity PROVIDER has these attributes: its unique number as the key attribute, name, address, and phone. The relationship between MEDICATION and PROVIDER is PROVIDED_BY (M to N) with additional attributes: Quantity, Import_price and date.
- Super-class PATIENT consists of these attributes: full name (first name and last name), date of birth, gender, address, and phone number. The key attribute of PATIENT is the unique code, which is the combination of their patient type ("OP" or "IP") and the 9 digits number. Our group can not describe this part in (E)ERD, so we use the notation "Code" as key attribute and explain detail in section three.
- Super-class PATIENT is also generalized into 2 sub-classes: OUTPATIENT (OP) and INPATIENT (IP) which are disjoint, each has attributes that inherits from EMPLOYEE. Moreover, the subclass INPATIENT also has their own attributes: Fee, Date_of_admission, Sickroom, diagnosis and derived attribute Discharge_date.
- The relationship EXAMINE is a Ternary (3-ary) Relationship among DOCTOR, OUTPATIENT and MEDICATION with a multi-valued attribute "Detail" which is the combination of examination date, diagnosis, the next examination date, and fee attribute. Each outpatient will be examined by a doctor and a doctor can examine many outpatients or he/she has no outpatient. The outpatients can have many examinations with their examining doctor, so the attribute "Detail" should be multi-valued and a examination can have many medications.
- The relationship TREAT is a Ternary (3-ary) Relationship among DOCTOR, INPATIENT and MEDICATION with a multi-valued attribute "Detail" which is the combination of Start_date, End_date, Result, and Is_recovered. Each inpatient will be treated by at

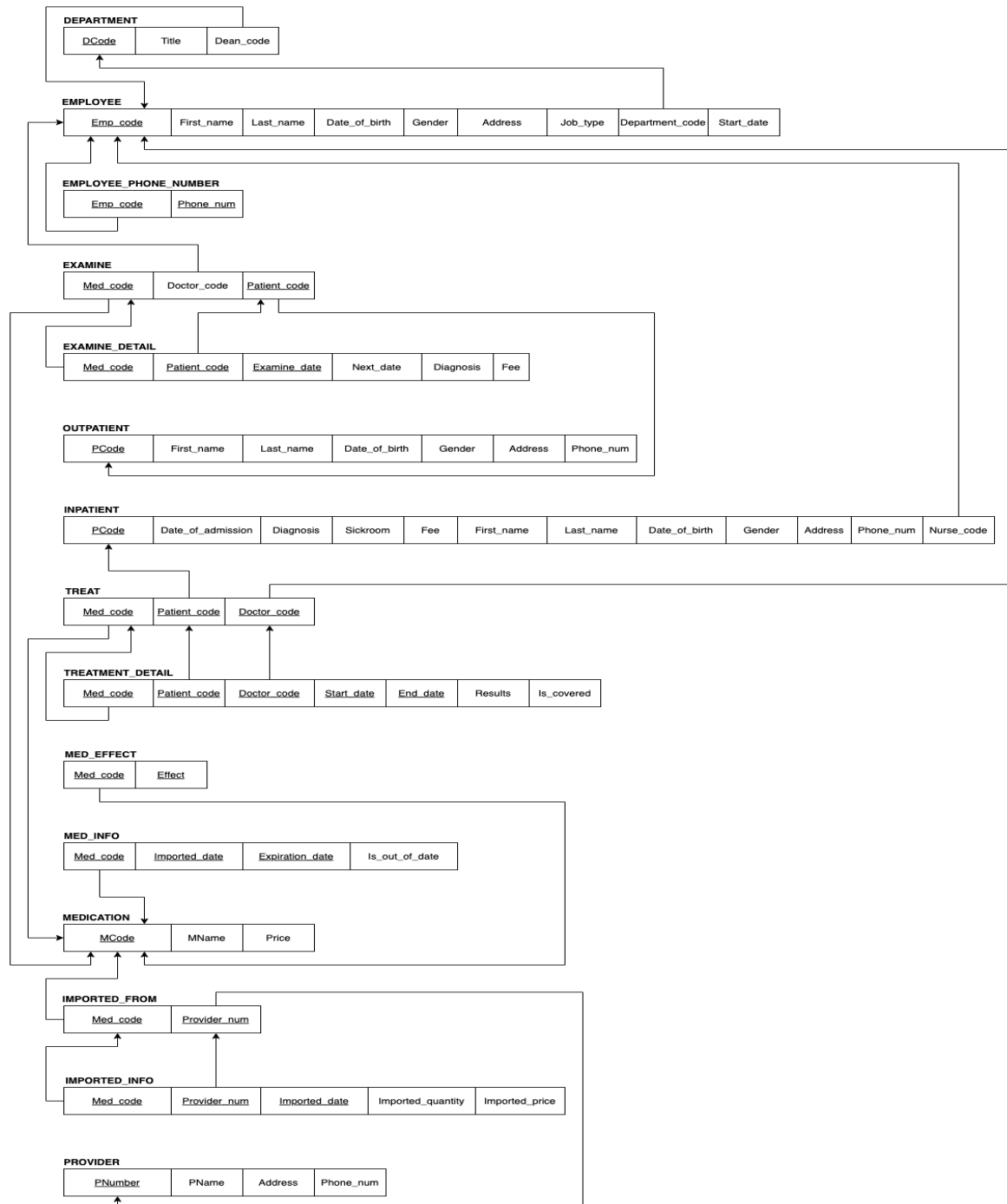


least one doctor and a doctor can treat many inpatients or he/she has no inpatient. The inpatients can have many treatment with their doctor, so the attribute Start_date and End_date should be multi-value and a treatment can have many medications.

- The relationship between NURSE and INPATIENT is TAKE_CARE_OF, which has the constraint is 1 to N because each inpatient is taken care of by a nurse; a nurse can take care of many inpatients at the same time.

1.2 Relational schema

1.2.1 Mapping ERD of Hospital database to relational model



1.2.2 Description for the relational model above

- **Mapping DEPARTMENT, MEDICATION, PROVIDER entity type**

Firstly, we need to mapping 3 entity type DEPARTMENT, MEDICATION, PROVIDER.

We created a new relation name DEPARTMENT which is corresponding to DEPARTMENT entity type. The primary key of DEPARTMENT relation is DCode. DEPARTMENT relation also include other attribute of DEPARTMENT entity type.

We also created a new relation name PROVIDER which is corresponding to PROVIDER entity type. The primary key of PROVIDER relation is PNumber. PROVIDER relation also include other attribute of PROVIDER entity type.

Similar with MEDICATION, we created MEDICATION relation. The primary key of MEDICATION relation is MCode. MEDICATION relation also include other attribute of MEDICATION entity type.

- **Mapping IMPORTED_FROM relationship**

Secondly, we map IMPORTED_FROM relationship type.

Because IMPORTED_FROM is a binary M:N relationship type. Thus, we created a new relation name IMPORTED_FROM which has attributes namely Med_code (a foreign key that refer to MCode in MEDICATION relation), Provider_num (a foreign key that refer to PNumber in PROVIDER relation). The combination of Med_code and Provider_num forms the primary key of IMPORTED_FROM relation.

- **Mapping of multi-valued attributes**

Thirdly, in this step, we map multi-valued attributes namely Effect (an attribute of MEDICATION entity type), Med_info (an attribute of MEDICATION entity type) and Imported_info (an attribute of IMPORTED_FROM relationship type).

We created new relations name EFFECT and MED_INFO which are corresponding to Effect and Med_info, respectively. The EFFECT relation has two attribute are Med_code (a foreign key that refer to MCode in MEDICATION relation), and Effect. The combination of Med_code and Effect forms the primary key of EFFECT relation. In similarity, The MED_INFO relation has two attribute are Med_code (a foreign key that refer to MCode in MEDICATION relation), and other attributes. The combination of Med_code, Imported_date and Expiration_date forms the primary key of MED_INFO relation. Furthermore, we created new relations name IMPORTED_INFO which has 5 attributes namely Med_code (a foreign key that refer to Med_code in IMPORTED_FROM relation), Provider_num (a foreign key that refer to Provider_num in IMPORTED_FROM relation), Imported_date, Imported_quantity, Imported_price. The combination of Med_code, Provider_num and Imported_date form the primary key of the relation.

- **Mapping of specializations**

Fourthly, we map the two specializations in ERD.

For the Specification of EMPLOYEE with 2 sub-classes, DOCTOR and NURSE, the option Single relation with one type attribute option (8C) is optimal. The first reason for this is that this is a disjoint specification, no EMPLOYEE could be both DOCTOR and NURSE at the same time, one attribute is enough to distinct between 2 types of EMPLOYEE. Therefore, there is no need to use option 8D with multiple attributes which add unnecessary row to the table. Moreover, the two sub-classes DOCTOR and NURSE don't

have any attribute, they just have those attributes inherited from super-class EMPLOYEE. Therefore we don't need to add any more attribute (row) and no NULL is created (waste storage). We won't use option 8A or 8B which will be more complex need more relations.

For the Specification of PATIENT with 2 sub-classes, INPATIENT and OUTPATIENT, a multiple relations—subclass relations only option (8B) is optimal. The option single relation with one or more type attributes shouldn't be use in this case since INPATIENT subclass have a lot of attribute but subclass OUTPATIENT have none, this could create a lot of NULL value when OUTPATIENT is stored in PATIENT relation. Furthermore, this is also a disjoint specification. Therefore, there will be no overlapping data (no PATIENT is both INPATIENT and OUTPATIENT) which won't create no duplicate when use option 8B. Multiple relations—subclass relations only (8B) is chosen over Multiple relations—super-class and sub-classes (8A) since 8B is simpler with 1 less relation and faster because when using option 8A, each time data is retrieved, it have to pass to 2 relations (super-class and subclass).

- **Mapping of MANAGE_BY binary 1:1 relationship type**

Fifthly, there is only one 1:1 relationship, it is MANAGE_BY between DOCTOR and DEPARTMENT. A new attribute named Dean_code is added to DEPARTMENT relation. Dean_code attribute refer to Emp_code of EMPLOYEE. We add a attribute to DEPARTMENT relation because DEPARTMENT is a total participant.

- **Mapping of binary 1:N relationship types**

Sixthly, we map 2 binary 1:N relationship types name WORKED_BY and TAKE_CARE_OF.

The first relationship is WORKED_BY between DEPARTMENT and EMPLOYEE. We add a new attribute Department_code (as a foreign key that refer to DCode in DEPARTMENT relation) to EMPLOYEE relation because the cardinality constraint of EMPLOYEE entity type participating in WORKED_BY is N. Moreover, we also add Start_date (attributes of WORKED_BY relationship) into the WORKED_BY relation.

The second relationship is TAKE_CARE_OF between NURSE and INPATIENT. We add a new attribute Nurse_code (which refers to Emp_code in EMPLOYEE relation) to INPATIENT relation because INPATIENT is on N-side.

- **Mapping of some other multi-valued attribute**

Seventhly, we map Phone_num multi-valued attribute and Detail multi-valued attributes of both EXAMINE and TREAT relationship.

We create a relation EMPLOYEE_PHONE_NUMBER is for the multi-valued attribute Phone_Num from EMPLOYEE entity. It has 2 attributes Emp_code and Phone_num, and the combination of them form the primary key of EMPLOYEE_PHONE_NUMBER relation.

A new relation named EXAMINATION_DETAIL is created for "Detail" multi-valued attribute of EXAMINE relationship. Its primary key is the combination of foreign key Med_code, Patient_code and Examine_date.

A new relation named TREATMENT_DETAIL is created for "Detail" multi-valued attribute of TREAT relationship. Its primary key is the combination of foreign key Med_code, Doctor_code, Patient_code, Start_date, End_date.

- **Mapping of N-ary relationship types**

Finally, we map EXAMINE and TREAT relationship.

A new relation is created for EXAMINE relationship with 3 attributes Med_code, Doctor_code and Patient_code which refer to MCode in MEDICATION, Emp_code in EMPLOYEE and PCode in OUTPATIENT, respectively. Since the cardinality constraints on entity type DOCTOR participating in EXAMINE is 1, then the primary key of EXAMINE relation should not include the foreign key Doctor_code. The combination of Med_code and Patient_code forms the primary key of EXAMINE relation.

In similarity, a new relation is created for TREAT relationship with 3 attributes Med_code, Patient_code and Doctor_code which refer to MCode in MEDICATION, PCode in INPATIENT and Emp_code in EMPLOYEE.

1.3 Unidentified Constraints

- The dean must hold a specific speciality and has had more than 5 years of experience since the date he or she was awarded the speciality degree. We can not use Entity Relationship diagram or Relational Database schema to illustrate this constraint because not all the orders constraints of the business can be performed as the forms of Entity Relationship diagram or Relational Database schema. However, we can note it as labels for later physical and logical designs.
- If one is an outpatient, the unique code for him or her starts with "OP," which is then followed by 9 digits such as "OP000000001." If one is an inpatient, the unique code for him or her starts with "IP," which is then followed by 9 digits such as "IP000000001." The prefix "OP" or "IP" of patient's code can't be illustrated in Entity Relationship diagram.

2 Assignment 2

2.1 Part 1: Physical Database Design

2.1.1 SQL statement to create database and insert values.

```
1 CREATE TABLE EMPLOYEE
2 (
3     Emp_code          VARCHAR(15),
4     First_name        VARCHAR(15)      NOT NULL,
5     Last_name         VARCHAR(15)      NOT NULL,
6     Date_of_birth     DATE,
7     Gender            VARCHAR(6)       NOT NULL,
8     Address           VARCHAR(100),
9     Job_type          VARCHAR(10)      NOT NULL,
10    Department_code    VARCHAR(15),
11    Start_date         DATE,
12
13    PRIMARY KEY (Emp_code)
14 );
15
16 CREATE TABLE EMPLOYEE_PHONE_NUMBER
17 (
```



```
18     Emp_code      VARCHAR(15)      NOT NULL,
19     Phone_num     CHAR(10)         NOT NULL,
20
21     PRIMARY KEY (Emp_code, Phone_num),
22
23     FOREIGN KEY (Emp_code) REFERENCES EMPLOYEE(Emp_code)
24         ON DELETE CASCADE
25 );
26
27 CREATE TABLE DEPARTMENT
28 (
29     Dcode          VARCHAR(15),
30     Title          VARCHAR(15)      NOT NULL,
31     Dean_code      VARCHAR(15)      NOT NULL UNIQUE,
32
33     PRIMARY KEY (Dcode)
34 );
35
36 ALTER TABLE EMPLOYEE
37 ADD CONSTRAINT fk_emp_dept_dcode FOREIGN KEY (Department_code)
38     REFERENCES Department(DCode)
39     ON DELETE CASCADE
40     DEFERRABLE INITIALLY DEFERRED;
41
42 ALTER TABLE DEPARTMENT
43 ADD CONSTRAINT fk_dept_emp_deanCode FOREIGN KEY (Dean_code)
44     REFERENCES EMPLOYEE(Emp_code)
45     ON DELETE SET NULL
46     DEFERRABLE INITIALLY DEFERRED;
47
48 CREATE TABLE OUTPATIENT
49 (
50     PCode          CHAR(11)         CHECK (PCode LIKE 'OP_-----'),
51     First_name      VARCHAR(15)      NOT NULL,
52     Last_name       VARCHAR(15)      NOT NULL,
53     Date_of_birth   DATE              NOT NULL,
54     Gender          VARCHAR(6)        NOT NULL,
55     Address         VARCHAR(100)      NOT NULL,
56     Phone_num       CHAR(10),
57
58     PRIMARY KEY (PCode)
59 );
60
61 CREATE TABLE EXAMINE
62 (
63     Med_code        VARCHAR(15)      NOT NULL,
64     Doctor_code     VARCHAR(15)      NOT NULL,
65     Patient_code    CHAR(11)         NOT NULL,
```

```
66
67     PRIMARY KEY (Med_code, Patient_code),
68
69     CONSTRAINT FK_EXA_MED_MED_CODE FOREIGN KEY (Med_code)
70         REFERENCES MEDICATION(Mcode)
71         ON DELETE CASCADE,
72     CONSTRAINT FK_EXA_MED_DOCTOR_CODE FOREIGN KEY (Doctor_code)
73         REFERENCES EMPLOYEE(Emp_code)
74         ON DELETE SET NULL,
75     CONSTRAINT FK_EXA_OUTPAT_PATIENT_CODE FOREIGN KEY (Patient_code)
76         REFERENCES OUTPATIENT(Pcode)
77         ON DELETE CASCADE
78 );
79
80 CREATE TABLE EXAMINE_DETAIL
81 (
82     Med_code    VARCHAR(15)    NOT NULL,
83     Patient_code CHAR(11)      NOT NULL,
84     Examine_date DATE          NOT NULL,
85     Next_date   DATE,
86     Diagnosis   VARCHAR(50)    NOT NULL,
87     Fee         DECIMAL(8,2)   NOT NULL,
88
89     PRIMARY KEY (Med_code, Patient_code, Examine_date),
90
91     CONSTRAINT FK_EXADE_EXA_MED_CODE_PAT FOREIGN KEY (Med_code,
92         Patient_code)
93         REFERENCES EXAMINE(Med_code, Patient_code)
94         ON DELETE CASCADE
95 );
96
97 CREATE TABLE INPATIENT
98 (
99     Pcode        CHAR(11)      CHECK (PCode LIKE 'IP_-----'),
100     Date_of_admission DATE      NOT NULL,
101     Diagnosis     VARCHAR(30)   NOT NULL,
102     Sickroom      VARCHAR(15)   NOT NULL,
103     Fee           DEC(8,2)      NOT NULL,
104     First_name    VARCHAR(15)   NOT NULL,
105     Last_name     VARCHAR(15)   NOT NULL,
106     Day_of_birth  DATE          NOT NULL,
107     Gender        VARCHAR(6)    NOT NULL,
108     Address       VARCHAR(100)  NOT NULL,
109     Phone_num     CHAR(10),
110     Nurse_code    VARCHAR(15)   NOT NULL,
111
112     PRIMARY KEY(Pcode),
```

```
113 FOREIGN KEY (Nurse_code) REFERENCES EMPLOYEE(Emp_code)
114 ON DELETE SET NULL
115 );
116
117 CREATE TABLE TREAT
118 (
119     Med_code VARCHAR(15),
120     Patient_code CHAR(11),
121     Doctor_code VARCHAR(15),
122
123     PRIMARY KEY (Med_code, Patient_code, Doctor_code),
124
125     FOREIGN KEY (Med_code) REFERENCES MEDICATION(MCode)
126     ON DELETE CASCADE,
127     FOREIGN KEY (Patient_code) REFERENCES INPATIENT(PCode)
128     ON DELETE CASCADE,
129     FOREIGN KEY (Doctor_code) REFERENCES EMPLOYEE(Emp_code)
130     ON DELETE CASCADE
131 );
132
133 CREATE TABLE TREATMENT_DETAIL
134 (
135     Med_code VARCHAR(15) NOT NULL,
136     Patient_code CHAR(11) NOT NULL,
137     Doctor_code VARCHAR(15) NOT NULL,
138     Start_day DATE NOT NULL,
139     End_day DATE,
140     Result VARCHAR(30),
141     Is_covered CHAR(1) CHECK (Is_covered IN ('Y', 'N')),
142
143     PRIMARY KEY (Med_code, Patient_code, Doctor_code, Start_day),
144
145     FOREIGN KEY (Med_code, Patient_code, Doctor_code) REFERENCES
146     TREAT(Med_code, Patient_code, Doctor_code)
147     ON DELETE CASCADE
148 );
149
150 CREATE TABLE PROVIDER
151 (
152     PNumber VARCHAR(15),
153     PName VARCHAR(30) UNIQUE,
154     Address VARCHAR(100),
155     Phone_num CHAR(10),
156
157     PRIMARY KEY (PNumber)
158 );
159
160 CREATE TABLE MEDICATION (
```

```
160     MCode          VARCHAR(15),
161     MName          VARCHAR(30)          UNIQUE,
162     Price          DECIMAL(8,2),
163
164     PRIMARY KEY (MCode)
165 );
166
167 CREATE TABLE IMPORTED_FROM
168 (
169     Med_code        VARCHAR(15),
170     Provider_num    VARCHAR(15),
171
172     CONSTRAINT PK_IMPORTED_FROM PRIMARY KEY (Med_code, Provider_num),
173
174     FOREIGN KEY (Med_code) REFERENCES MEDICATION(MCode)
175         ON DELETE CASCADE,
176     FOREIGN KEY (Provider_num) REFERENCES PROVIDER(PNumber)
177         ON DELETE CASCADE
178 );
179
180 CREATE TABLE IMPORTED_INFO
181 (
182     Med_code        VARCHAR(15),
183     Provider_num    VARCHAR(15),
184     Imported_date    DATE,
185     Imported_price   DECIMAL(8,2)       NOT NULL,
186     Imported_quantity INT              NOT NULL,
187
188     PRIMARY KEY (Med_code, Provider_num, Imported_date),
189
190     FOREIGN KEY (Med_code, Provider_num) REFERENCES IMPORTED_FROM (Med_code,
191         Provider_num)
192         ON DELETE CASCADE
193 );
194
195 CREATE TABLE MED_EFFECT
196 (
197     Med_code        VARCHAR(15),
198     Effect          VARCHAR(30)         NOT NULL,
199
200     PRIMARY KEY (Med_code, Effect),
201
202     FOREIGN KEY (Med_code) REFERENCES MEDICATION(MCode)
203         ON DELETE CASCADE
204 );
205
206 CREATE TABLE MED_INFO
207 (
```

```
207     Med_code          VARCHAR(15),
208     Imported_date      DATE          NOT NULL,
209     Expiration_date    DATE          NOT NULL,
210     Is_out_of_date     CHAR(1)       CHECK (Is_out_of_date IN ('Y',
211                                     'N')),
212
213     PRIMARY KEY (Med_code, Imported_date, Expiration_date),
214
215     FOREIGN KEY (Med_code) REFERENCES MEDICATION(MCode)
216         ON DELETE CASCADE
217 );
218
219 --INSERT
220 INSERT INTO DEPARTMENT VALUES ('101', 'ENT', 'ETN1004556');
221 INSERT INTO DEPARTMENT VALUES ('102', 'Cardiology', 'CDG2004667');
222 INSERT INTO DEPARTMENT VALUES ('103', 'Neurology', 'NEU3006443');
223 INSERT INTO DEPARTMENT VALUES ('202', 'Sexual Health', 'SEH6003251');
224 INSERT INTO DEPARTMENT VALUES ('203', 'Nutritions', 'NUD4009043');
225
226 ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';
227 INSERT INTO EMPLOYEE VALUES ('ETN1004556', 'William', 'Osler',
228                               '12/07/1852', 'MALE', 'Bradford West Gwillimbury, Canada', 'DOCTOR', '101',
229                               '01/04/2015');
230 INSERT INTO EMPLOYEE VALUES ('CDG2004667', 'Andreas', 'Vesalius',
231                               '31/12/1992', 'MALE', 'Belgium', 'DOCTOR', '102', '30/04/2010');
232 INSERT INTO EMPLOYEE VALUES ('NEU3006443', 'Ignaz', 'Semmelweis',
233                               '01/07/1924', 'MALE', 'Tabán, Budapest, Hungary', 'DOCTOR', '103',
234                               '01/06/2020');
235 INSERT INTO EMPLOYEE VALUES ('SEH6003251', 'Joseph', 'Lister', '05/04/1827',
236                               'MALE', 'Upton House, Newham', 'DOCTOR', '202', '02/09/2005');
237 INSERT INTO EMPLOYEE VALUES ('NUD4009043', 'Florence', 'Nightingale',
238                               '12/05/1923', 'FEMALE', 'Florence, Italy', 'NURSE', '203', '01/05/2000');
239 INSERT INTO EMPLOYEE VALUES ('ETN1005776', 'Clarissa', 'Harlowe Barton',
240                               '25/12/1924', 'FEMALE', 'North Oxford, Oxford, Massachusetts, United
241                               States', 'NURSE', '101', '20/11/2001');
242 INSERT INTO EMPLOYEE VALUES ('SEH6003822', 'Mary', 'Eliza Mahoney',
243                               '02/04/1933', 'FEMALE', 'Dorchester, Boston, Massachusetts, United States',
244                               'NURSE', '202', '31/10/2016');
245
246 INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('ETN1004556', '0900423543');
247 INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('ETN1004556', '0902573912');
248 INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('NEU3006443', '0902573451');
249 INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('SEH6003251', '0928371279');
250 INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('NUD4009043', '0932346133');
251
252 INSERT into PROVIDER VALUES ('US4781601046', 'Johnson and Johnson', 'New
253                               Jersey, United States', '0991030201');
```

```
241 INSERT into PROVIDER VALUES('US7711951043', 'Roche', 'Basel, Switzerland',
242 '0902002124');
243 INSERT into PROVIDER VALUES('US7170811035', 'Pfizer', 'New York, United
244 States', '0882001251');
245 INSERT into PROVIDER VALUES('CH0012005267', 'Novartis', 'Basel,
246 Switzerland', '0907012282');
247
248 INSERT INTO MEDICATION VALUES ('N02BE01', 'Paracetamol', 199.99);
249 INSERT INTO MEDICATION VALUES ('N05CF03', 'Zaleplon', 449.89);
250 INSERT INTO MEDICATION VALUES ('J01XA05', 'Oritavancin', 99.89);
251 INSERT INTO MEDICATION VALUES ('L01CA06', 'Vintafolide', 399.99);
252 INSERT INTO MEDICATION VALUES ('A07AB04', 'Succinylsulfathiazole', 219.99);
253 INSERT INTO MEDICATION VALUES ('N02AD02', 'Phenazocine', 119.00);
254 INSERT INTO MEDICATION VALUES ('A12CC03', 'Magnesium gluconate', 79.99);
255
256 INSERT INTO IMPORTED_FROM VALUES ('N02BE01', 'US4781601046');
257 INSERT INTO IMPORTED_FROM VALUES ('N02BE01', 'US7711951043');
258 INSERT INTO IMPORTED_FROM VALUES ('L01CA06', 'US7170811035');
259 INSERT INTO IMPORTED_FROM VALUES ('L01CA06', 'CH0012005267');
260 INSERT INTO IMPORTED_FROM VALUES ('N05CF03', 'US4781601046');
261 INSERT INTO IMPORTED_FROM VALUES ('N05CF03', 'US7711951043');
262 INSERT INTO IMPORTED_FROM VALUES ('J01XA05', 'CH0012005267');
263 INSERT INTO IMPORTED_FROM VALUES ('A07AB04', 'US4781601046');
264 INSERT INTO IMPORTED_FROM VALUES ('N02AD02', 'CH0012005267');
265 INSERT INTO IMPORTED_FROM VALUES ('A12CC03', 'US7170811035');
266
267 INSERT INTO IMPORTED_INFO VALUES ('N02BE01', 'US4781601046', '12/09/2019',
268 99.99, 20000);
269 INSERT INTO IMPORTED_INFO VALUES ('N02BE01', 'US7711951043', '04/12/2019',
270 79.89, 30000);
271 INSERT INTO IMPORTED_INFO VALUES ('L01CA06', 'US7170811035', '22/02/2019',
272 59.99, 45000);
273 INSERT INTO IMPORTED_INFO VALUES ('L01CA06', 'CH0012005267', '16/07/2019',
274 49.00, 15000);
275 INSERT INTO IMPORTED_INFO VALUES ('N05CF03', 'US4781601046', '08/04/2019',
276 199.89, 25000);
277 INSERT INTO IMPORTED_INFO VALUES ('N05CF03', 'US7711951043', '15/04/2019',
278 189.89, 4000);
279 INSERT INTO IMPORTED_INFO VALUES ('J01XA05', 'CH0012005267', '12/10/2019',
280 159.49, 1500);
281 INSERT INTO IMPORTED_INFO VALUES ('A07AB04', 'US4781601046', '11/04/2019',
282 199.49, 1000);
283 INSERT INTO IMPORTED_INFO VALUES ('N02AD02', 'CH0012005267', '04/05/2019',
284 99.99, 2500);
285 INSERT INTO IMPORTED_INFO VALUES ('A12CC03', 'US7170811035', '27/10/2018',
286 49.49, 9000);
287
288 INSERT into MED_INFO VALUES ('N02BE01', '12/09/2019', '30/04/2022', 'N');
```



```
276 INSERT into MED_INFO VALUES ('N02BE01', '04/12/2019', '21/03/2022', 'N');
277 INSERT into MED_INFO VALUES ('L01CA06', '22/02/2019', '15/08/2021', 'Y');
278 INSERT into MED_INFO VALUES ('L01CA06', '16/07/2019', '10/03/2022', 'N');
279 INSERT into MED_INFO VALUES ('N05CF03', '08/04/2019', '26/02/2022', 'N');
280 INSERT into MED_INFO VALUES ('N05CF03', '15/04/2019', '07/08/2021', 'Y');
281 INSERT into MED_INFO VALUES ('J01XA05', '12/10/2019', '26/07/2021', 'Y');
282 INSERT into MED_INFO VALUES ('A07AB04', '11/04/2019', '25/03/2022', 'N');
283 INSERT into MED_INFO VALUES ('N02AD02', '04/05/2019', '27/12/2020', 'Y');
284 INSERT into MED_INFO VALUES ('A12CC03', '27/10/2018', '03/02/2020', 'Y');
285
286 INSERT INTO MED_EFFECT VALUES ('N02BE01', 'analgesic');
287 INSERT INTO MED_EFFECT VALUES ('N02BE01', 'antipyretic');
288 INSERT INTO MED_EFFECT VALUES ('N05CF03', 'sedative');
289 INSERT INTO MED_EFFECT VALUES ('N05CF03', 'soothing');
290 INSERT INTO MED_EFFECT VALUES ('L01CA06', 'immunomodulatory');
291 INSERT INTO MED_EFFECT VALUES ('L01CA06', 'anti-cancer');
292 INSERT INTO MED_EFFECT VALUES ('J01XA05', 'antimicrobial');
293 INSERT INTO MED_EFFECT VALUES ('A07AB04', 'treat diarrhea');
294 INSERT INTO MED_EFFECT VALUES ('A07AB04', 'enteritis');
295 INSERT INTO MED_EFFECT VALUES ('A07AB04', 'anti-infective');
296 INSERT INTO MED_EFFECT VALUES ('N02AD02', 'analgesic');
297 INSERT INTO MED_EFFECT VALUES ('A12CC03', 'mineral supplements');
298
299 INSERT INTO INPATIENT VALUES ('IP205679239', '06/10/2019', 'Catastrophe',
    'A202', 1099.99, 'Claud', 'Riddle', '06/10/1953', 'MALE', '33 Windy Ridge
    Road, Washington', '0909009900', 'NUD4009043');
300 INSERT INTO INPATIENT VALUES ('IP945829231', '08/08/2020', 'Liver Cancer',
    'A109', 2092.89, 'Louis', 'Clark', '08/08/1924', 'MALE', '275 Turner Kyle
    Dr, Savannah, Tennessee', '0909345940', 'ETN1005776');
301 INSERT INTO INPATIENT VALUES ('IP205127452', '26/09/2021', 'Pneumonia',
    'A391', 3942.99, 'Cloe', 'Browning', '26/09/1930', 'MALE', '439 Harlem Ave,
    Forest Park, Illinois', '0909023464', 'ETN1005776');
302 INSERT INTO INPATIENT VALUES ('IP203462341', '03/10/2021', 'Traumatic brain
    injury', 'B029', 1529.99, 'Rosa', 'Lopez', '03/10/1989', 'FEMALE', '20591
    Sycamore Crest Ln, Katy, Texas', '0909056342', 'NUD4009043');
303 INSERT INTO INPATIENT VALUES ('IP104232231', '03/10/2021', 'Appendicitis',
    'B823', 9238.99, 'Kayla', 'Bernard', '10/10/1983', 'FEMALE', '4500 Harbour
    Pointe Blvd, Mukilteo, Washington', '0903457345', 'SEH6003822');
304 INSERT INTO INPATIENT VALUES ('IP923958724', '10/10/2021', 'Cirrhosis',
    'C215', 7792.99, 'Isla', 'Pasteur', '23/06/1965', 'FEMALE', '319 Newberry
    St, Fredericktown, Missouri', '0943476232', 'ETN1005776');
305 INSERT INTO INPATIENT VALUES ('IP383739276', '23/05/2020', 'Pneumonia',
    'B241', 3942.99, 'Brielle', 'Maxwell', '23/01/1945', 'MALE', '3825 Way Crk,
    Chester, Virginia', '0908345624', 'NUD4009043');
306
307 INSERT INTO OUTPATIENT VALUES ('OP000000000', 'Claud', 'Bernard',
    '12/07/1985', 'MALE', '1490 Windy Ridge Road, Washington', '0909009232');
```



```
308 INSERT INTO OUTPATIENT VALUES ('OP000000001', 'Louis', 'Pasteur',  
    '15/12/1996', 'MALE', '275 Turner Kyle Dr, Savannah, Tennessee',  
    '0903847123');  
309 INSERT INTO OUTPATIENT VALUES ('OP000000002', 'Cloe', 'Maxwell',  
    '01/07/1963', 'MALE', '439 Harlem Ave, Forest Park, Illinois',  
    '0903732452');  
310 INSERT INTO OUTPATIENT VALUES ('OP000000003', 'Rosa', 'Mcmillan',  
    '05/04/2012', 'FEMALE', '20591 Sycamore Crest Ln, Katy, Texas', NULL);  
311 INSERT INTO OUTPATIENT VALUES ('OP000000004', 'Kayla', 'Sheppard',  
    '12/06/1935', 'FEMALE', '4500 Harbour Pointe Blvd, Mukilteo, Washington',  
    NULL);  
312 INSERT INTO OUTPATIENT VALUES ('OP000000005', 'Isla', 'Riddle',  
    '20/10/1991', 'FEMALE', '319 Newberry St, Fredericktown, Missouri',  
    '0902748238');  
313 INSERT INTO OUTPATIENT VALUES ('OP000000006', 'Brielle', 'Clark',  
    '07/05/1957', 'MALE', '3825 Way Crk, Chester, Virginia', '0990275396');  
314 INSERT INTO OUTPATIENT VALUES ('OP000000007', 'Molly', 'Browning',  
    '26/06/1978', 'FEMALE', '5550 Owensmouth Ave, Woodland Hills, California',  
    '0905839203');  
315 INSERT INTO OUTPATIENT VALUES ('OP000000008', 'Emily',  
    'Bentley', '20/11/2002', 'FEMALE', '9280 Cottonwood Dr, Bastrop,  
    Louisiana', '0904372973');  
316 INSERT INTO OUTPATIENT VALUES ('OP000000009', 'Christopher', 'Lopez',  
    '29/02/2008', 'MALE', '759 A Euclatubba Rd, Saltillo, Mississippi', NULL);  
317  
318 INSERT INTO EXAMINE VALUES ('J01XA05', 'ETN1004556', 'OP000000000');  
319 INSERT INTO EXAMINE VALUES ('N02BE01', 'ETN1004556', 'OP000000000');  
320 INSERT INTO EXAMINE VALUES ('N05CF03', 'ETN1004556', 'OP000000000');  
321 INSERT INTO EXAMINE VALUES ('N02BE01', 'ETN1004556', 'OP000000001');  
322 INSERT INTO EXAMINE VALUES ('J01XA05', 'ETN1004556', 'OP000000001');  
323 INSERT INTO EXAMINE VALUES ('J01XA05', 'ETN1004556', 'OP000000002');  
324 INSERT INTO EXAMINE VALUES ('N02BE01', 'ETN1004556', 'OP000000002');  
325 INSERT INTO EXAMINE VALUES ('N05CF03', 'ETN1004556', 'OP000000002');  
326 INSERT INTO EXAMINE VALUES ('L01CA06', 'ETN1004556', 'OP000000002');  
327 INSERT INTO EXAMINE VALUES ('J01XA05', 'ETN1004556', 'OP000000003');  
328 INSERT INTO EXAMINE VALUES ('N02BE01', 'ETN1004556', 'OP000000003');  
329 INSERT INTO EXAMINE VALUES ('A12CC03', 'ETN1004556', 'OP000000003');  
330 INSERT INTO EXAMINE VALUES ('N05CF03', 'NEU3006443', 'OP000000004');  
331 INSERT INTO EXAMINE VALUES ('N02AD02', 'SEH6003251', 'OP000000005');  
332 INSERT INTO EXAMINE VALUES ('A12CC03', 'SEH6003251', 'OP000000005');  
333 INSERT INTO EXAMINE VALUES ('L01CA06', 'CDG2004667', 'OP000000006');  
334 INSERT INTO EXAMINE VALUES ('N02AD02', 'CDG2004667', 'OP000000006');  
335 INSERT INTO EXAMINE VALUES ('L01CA06', 'CDG2004667', 'OP000000007');  
336 INSERT INTO EXAMINE VALUES ('A07AB04', 'CDG2004667', 'OP000000007');  
337 INSERT INTO EXAMINE VALUES ('N02AD02', 'SEH6003251', 'OP000000008');  
338 INSERT INTO EXAMINE VALUES ('A12CC03', 'SEH6003251', 'OP000000008');  
339 INSERT INTO EXAMINE VALUES ('A07AB04', 'NUD4009043', 'OP000000009');  
340 INSERT INTO EXAMINE VALUES ('A12CC03', 'NUD4009043', 'OP000000009');
```

```
341
342 INSERT INTO EXAMINE_DETAIL VALUES ('J01XA05', 'OP000000000', '10/07/2021',
    '12/07/2021', 'Fever', 99);
343 INSERT INTO EXAMINE_DETAIL VALUES ('N02BE01', 'OP000000000', '12/07/2021',
    '14/07/2021', 'Fever', 99);
344 INSERT INTO EXAMINE_DETAIL VALUES ('N05CF03', 'OP000000000', '14/07/2021',
    NULL, 'Fever', 199);
345 INSERT INTO EXAMINE_DETAIL VALUES ('N02BE01', 'OP000000001', '02/08/2021',
    '05/08/2021', 'Fever virus', 199);
346 INSERT INTO EXAMINE_DETAIL VALUES ('J01XA05', 'OP000000001', '05/08/2021',
    NULL, 'Fever virus', 199);
347 INSERT INTO EXAMINE_DETAIL VALUES ('J01XA05', 'OP000000002', '14/09/2021',
    '15/09/2021', 'Chronic sinusitis', 99);
348 INSERT INTO EXAMINE_DETAIL VALUES ('N02BE01', 'OP000000002', '15/09/2021',
    '17/09/2021', 'Fever', 99);
349 INSERT INTO EXAMINE_DETAIL VALUES ('N05CF03', 'OP000000002', '17/09/2021',
    '19/09/2021', 'Chronic sinusitis', 199);
350 INSERT INTO EXAMINE_DETAIL VALUES ('L01CA06', 'OP000000002', '19/09/2021',
    NULL, 'Chronic sinusitis', 199);
351 INSERT INTO EXAMINE_DETAIL VALUES ('J01XA05', 'OP000000003', '01/10/2021',
    '03/10/2021', 'Fever', 99);
352 INSERT INTO EXAMINE_DETAIL VALUES ('N02BE01', 'OP000000003', '03/10/2021',
    '05/10/2021', 'Fever', 199);
353 INSERT INTO EXAMINE_DETAIL VALUES ('A12CC03', 'OP000000003', '05/10/2021',
    NULL, 'Fever', 199);
354 INSERT INTO EXAMINE_DETAIL VALUES ('N05CF03', 'OP000000004', '05/09/2021',
    '12/09/2021', 'Parkinson', 499);
355 INSERT INTO EXAMINE_DETAIL VALUES ('N05CF03', 'OP000000004', '12/09/2021',
    '19/09/2021', 'Parkinson', 499);
356 INSERT INTO EXAMINE_DETAIL VALUES ('N05CF03', 'OP000000004', '19/09/2021',
    '26/09/2021', 'Dementia', 499);
357 INSERT INTO EXAMINE_DETAIL VALUES ('N05CF03', 'OP000000004', '26/09/2021',
    NULL, 'Dementia', 499);
358 INSERT INTO EXAMINE_DETAIL VALUES ('N02AD02', 'OP000000005', '11/08/2021',
    '25/08/2021', 'Regular antenatal check-up', 119);
359 INSERT INTO EXAMINE_DETAIL VALUES ('A12CC03', 'OP000000005', '25/08/2021',
    '06/09/2021', 'Regular antenatal check-up', 79);
360 INSERT INTO EXAMINE_DETAIL VALUES ('A12CC03', 'OP000000005', '06/09/2021',
    NULL, 'Regular antenatal check-up', 79);
361 INSERT INTO EXAMINE_DETAIL VALUES ('L01CA06', 'OP000000006', '15/09/2021',
    '22/09/2021', 'Coronary artery disease', 399);
362 INSERT INTO EXAMINE_DETAIL VALUES ('L01CA06', 'OP000000006', '22/09/2021',
    '29/09/2021', 'Coronary artery disease', 399);
363 INSERT INTO EXAMINE_DETAIL VALUES ('N02AD02', 'OP000000006', '29/09/2021',
    NULL, 'Coronary artery disease', 119);
364 INSERT INTO EXAMINE_DETAIL VALUES ('L01CA06', 'OP000000007', '26/09/2021',
    '03/10/2021', 'Breath heavily', 399);
```



```
365 INSERT INTO EXAMINE_DETAIL VALUES ('L01CA06', 'OP000000007', '03/10/2021',  
366 '10/10/2021', 'Swollen feet, legs, ascites', 399);  
367 INSERT INTO EXAMINE_DETAIL VALUES ('A07AB04', 'OP000000007', '10/10/2021',  
368 NULL, 'Heart failure level 2', 219);  
369 INSERT INTO EXAMINE_DETAIL VALUES ('N02AD02', 'OP000000008', '01/10/2021',  
370 '05/10/2021', 'Adolescent pregnancy', 119);  
371 INSERT INTO EXAMINE_DETAIL VALUES ('N02AD02', 'OP000000008', '05/10/2021',  
372 '12/10/2021', 'Adolescent pregnancy', 119);  
373 INSERT INTO EXAMINE_DETAIL VALUES ('A12CC03', 'OP000000008', '12/10/2021',  
374 NULL, 'Adolescent pregnancy', 79);  
375 INSERT INTO EXAMINE_DETAIL VALUES ('A07AB04', 'OP000000009', '25/10/2021',  
376 '02/11/2021', 'Diarrhea', 219);  
377 INSERT INTO EXAMINE_DETAIL VALUES ('A12CC03', 'OP000000009', '02/11/2021',  
378 NULL, 'Prolonged malnutrition', 79);  
379  
380 INSERT INTO TREAT VALUES ('J01XA05', 'IP205679239', 'CDG2004667');  
381 INSERT INTO TREAT VALUES ('N02BE01', 'IP945829231', 'ETN1004556');  
382 INSERT INTO TREAT VALUES ('N02BE01', 'IP205127452', 'NUD4009043');  
383 INSERT INTO TREAT VALUES ('N05CF03', 'IP203462341', 'ETN1004556');  
384 INSERT INTO TREAT VALUES ('N05CF03', 'IP104232231', 'NUD4009043');  
385 INSERT INTO TREAT VALUES ('L01CA06', 'IP923958724', 'ETN1004556');  
386 INSERT INTO TREAT VALUES ('L01CA06', 'IP383739276', 'NUD4009043');  
387 INSERT INTO TREAT VALUES ('L01CA06', 'IP205679239', 'CDG2004667');  
388 INSERT INTO TREAT VALUES ('A07AB04', 'IP945829231', 'ETN1004556');  
389 INSERT INTO TREAT VALUES ('A07AB04', 'IP205127452', 'ETN1004556');  
390 INSERT INTO TREAT VALUES ('A07AB04', 'IP203462341', 'ETN1004556');  
391 INSERT INTO TREAT VALUES ('N02AD02', 'IP104232231', 'ETN1004556');  
392 INSERT INTO TREAT VALUES ('A12CC03', 'IP923958724', 'NEU3006443');  
393 INSERT INTO TREAT VALUES ('A07AB04', 'IP383739276', 'SEH6003251');  
394 INSERT INTO TREAT VALUES ('A07AB04', 'IP205679239', 'NEU3006443');  
395 INSERT INTO TREAT VALUES ('A12CC03', 'IP945829231', 'NEU3006443');  
396 INSERT INTO TREAT VALUES ('A12CC03', 'IP205127452', 'SEH6003251');  
397 INSERT INTO TREAT VALUES ('L01CA06', 'IP203462341', 'SEH6003251');  
398 INSERT INTO TREAT VALUES ('L01CA06', 'IP104232231', 'SEH6003251');  
399 INSERT INTO TREAT VALUES ('J01XA05', 'IP923958724', 'CDG2004667');  
400 INSERT INTO TREAT VALUES ('N02AD02', 'IP383739276', 'CDG2004667');  
401  
402 INSERT INTO TREATMENT_DETAIL VALUES ('J01XA05', 'IP205679239',  
403 'CDG2004667', '06/10/2019', '07/11/2019', 'Catastrophe_initial diag', 'N');  
404 INSERT INTO TREATMENT_DETAIL VALUES ('N02BE01', 'IP945829231',  
405 'ETN1004556', '08/08/2020', '09/09/2020', 'Liver Cancer_initial diag',  
406 'N');  
407 INSERT INTO TREATMENT_DETAIL VALUES ('N02BE01', 'IP205127452',  
408 'NUD4009043', '26/09/2021', '27/10/2021', 'Pneumonia_initial diag', 'N');  
409 INSERT INTO TREATMENT_DETAIL VALUES ('N05CF03', 'IP203462341',  
410 'ETN1004556', '03/10/2021', '04/11/2021', 'Traumatic brain injury diag',  
411 'N');
```

```
399 INSERT INTO TREATMENT_DETAIL VALUES ('N05CF03', 'IP104232231',  
    'NUD4009043', '10/10/2021', '11/11/2021', 'Appendicitis_initial diag',  
    'N');  
400 INSERT INTO TREATMENT_DETAIL VALUES ('L01CA06', 'IP923958724',  
    'ETN1004556', '01/10/2021', '02/12/2021', 'Cirrhosis_initial diag', 'N');  
401 INSERT INTO TREATMENT_DETAIL VALUES ('L01CA06', 'IP383739276',  
    'NUD4009043', '23/05/2020', '24/06/2020', 'Pneumonia_initial diag', 'N');  
402 INSERT INTO TREATMENT_DETAIL VALUES ('L01CA06', 'IP205679239',  
    'CDG2004667', '05/07/2020', '06/12/2020', 'Catastrophe', 'N');  
403 INSERT INTO TREATMENT_DETAIL VALUES ('A07AB04', 'IP945829231',  
    'ETN1004556', '25/10/2021', '26/11/2021', 'Liver Cancer', 'N');  
404 INSERT INTO TREATMENT_DETAIL VALUES ('A07AB04', 'IP205127452',  
    'ETN1004556', '27/11/2021', '28/12/2021', 'Pneumonia', 'N');  
405 INSERT INTO TREATMENT_DETAIL VALUES ('A07AB04', 'IP203462341',  
    'ETN1004556', '22/05/2022', '23/06/2022', 'Traumatic brain injury', 'N');  
406 INSERT INTO TREATMENT_DETAIL VALUES ('N02AD02', 'IP104232231',  
    'ETN1004556', '18/11/2018', '19/01/2019', 'Appendicitis', 'N');  
407 INSERT INTO TREATMENT_DETAIL VALUES ('A12CC03', 'IP923958724',  
    'NEU3006443', '04/12/2021', '05/03/2022', 'Cirrhosis', 'N');  
408 INSERT INTO TREATMENT_DETAIL VALUES ('A07AB04', 'IP383739276',  
    'SEH6003251', '06/10/2019', '07/11/2019', 'Pneumonia', 'N');  
409 INSERT INTO TREATMENT_DETAIL VALUES ('A07AB04', 'IP205679239',  
    'NEU3006443', '08/08/2020', '09/09/2020', 'Post-operative follow-up',  
    'Y');  
410 INSERT INTO TREATMENT_DETAIL VALUES ('A12CC03', 'IP945829231',  
    'NEU3006443', '26/09/2021', '27/09/2021', 'Post-operative follow-up',  
    'Y');  
411 INSERT INTO TREATMENT_DETAIL VALUES ('A12CC03', 'IP205127452',  
    'SEH6003251', '03/10/2021', '04/11/2021', 'Post-operative follow-up',  
    'Y');  
412 INSERT INTO TREATMENT_DETAIL VALUES ('L01CA06', 'IP203462341',  
    'SEH6003251', '10/10/2021', '11/12/2021', 'Post-operative follow-up',  
    'Y');  
413 INSERT INTO TREATMENT_DETAIL VALUES ('L01CA06', 'IP104232231',  
    'SEH6003251', '01/10/2021', '02/11/2021', 'Post-operative follow-up',  
    'Y');  
414 INSERT INTO TREATMENT_DETAIL VALUES ('J01XA05', 'IP923958724',  
    'CDG2004667', '23/05/2020', '24/07/2020', 'Post-operative follow-up',  
    'Y');  
415 INSERT INTO TREATMENT_DETAIL VALUES ('N02AD02', 'IP383739276',  
    'CDG2004667', '05/07/2020', '06/08/2020', 'Post-operative follow-up',  
    'Y');
```

2.1.2 Explanation for database

2.1.2.a EMPLOYEE

DATA_TYPE:



❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	❖ DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 EMP_CODE	VARCHAR2(15 BYTE)	No	(null)	1 (null)	
2 FIRST_NAME	VARCHAR2(15 BYTE)	No	(null)	2 (null)	
3 LAST_NAME	VARCHAR2(15 BYTE)	No	(null)	3 (null)	
4 DATE_OF_BIRTH	DATE	Yes	(null)	4 (null)	
5 GENDER	VARCHAR2(6 BYTE)	No	(null)	5 (null)	
6 ADDRESS	VARCHAR2(100 B...)	Yes	(null)	6 (null)	
7 JOB_TYPE	VARCHAR2(10 BYTE)	No	(null)	7 (null)	
8 DEPARTMENT_CODE	VARCHAR2(15 BYTE)	Yes	(null)	8 (null)	
9 START_DATE	DATE	Yes	(null)	9 (null)	

CONSTRAINTS:

❖ CONSTRAINT_NAME	❖ CONSTRAINT_TYPE	❖ SEARCH_CONDITION	❖ R_OWNER	❖ R_TABLE_NAME	❖ R_CONSTRAINT_NAME	❖ DELETE_RULE
1 FK_EMP_DEPT_DCODE	Foreign_Key	(null)	SYSTEM	DEPARTMENT	SYS_C007213	CASCADE
2 SYS_C007206	Check	"FIRST_NAME" IS NOT NULL	(null)	(null)	(null)	(null)
3 SYS_C007207	Check	"LAST_NAME" IS NOT NULL	(null)	(null)	(null)	(null)
4 SYS_C007208	Check	"GENDER" IS NOT NULL	(null)	(null)	(null)	(null)
5 SYS_C007209	Check	"JOB_TYPE" IS NOT NULL	(null)	(null)	(null)	(null)
6 SYS_C007210	Primary_Key	(null)	(null)	(null)	(null)	(null)

VALUES:

❖ EMP_CODE	❖ FIRST_NAME	❖ LAST_NAME	❖ DATE_OF_BIRTH	❖ GENDER	❖ ADDRESS	❖ JOB_TYPE	❖ DEPARTMENT_CODE	❖ START_DATE
1 ETH1004556	William	Osler	12-07-1852	MALE	Bradford West Gwillimbury, Canada	DOCTOR	101	01-04-2015
2 CDG2004667	Andreas	Vesalius	31-12-1592	MALE	Belgium	DOCTOR	102	30-04-2010
3 NEU3006443	Ignaz	Semmelweis	01-07-1924	MALE	Tabán, Budapest, Hungary	DOCTOR	103	01-06-2020
4 SER6003251	Joseph	Lister	05-04-1827	MALE	Upton House, Newham	DOCTOR	202	02-09-2005
5 NUD4009043	Florence	Nightingale	12-05-1923	FEMALE	Florence, Italy	DOCTOR	203	01-05-2000
6 ETH1005776	Clarissa	Harlowe Barton	25-12-1924	FEMALE	North Oxford, Oxford, Massachusetts, United States	NURSE	101	20-11-2001
7 SER6003822	Mary	Eliza Mahoney	02-04-1933	FEMALE	Dorchester, Boston, Massachusetts, United States	NURSE	202	31-10-2016

- This table records the information about employees including their unique code, name, birthday, sex, address, job type, start date and department. it is noticeable that almost all the data type are VARCHAR with the length of 15 characters, except Date of birth and Start date which is type DATE and GENDER is 6 characters stand for "MALE" and "FEMALE".
- The Primary key of this table is EMP_code and the foreign key is DEPARTMENT_CODE from DEPARTMENT table.
- Furthermore, we are not allow EMP_CODE, their name, GENDER and JOB_TYPE to be NULL due to the basic requirements of original employee in reality. The column "ADDRESS" has length 100 bytes because some of the address is very long.

2.1.2.b EMPLOYEE_PHONE_NUMBER

DATA_TYPE:



⚡	COLUMN_NAME	DATA_TYPE	⚡	NULLABLE	DATA_DEFAULT	⚡	COLUMN_ID	⚡	COMMENTS
1	EMP_CODE	VARCHAR2(15 BYTE)	No		(null)		1	(null)	
2	PHONE_NUM	CHAR(10 BYTE)	No		(null)		2	(null)	

CONSTRAINTS:

⚡	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	⚡	R_OWNER	R_TABLE_NAME	⚡	R_CONSTRAINT_NAME
1	SYS_C007217	Check	"EMP_CODE" IS NOT NULL	(null)	(null)	(null)	(null)	
2	SYS_C007218	Check	"PHONE_NUM" IS NOT NULL	(null)	(null)	(null)	(null)	
3	SYS_C007219	Primary_Key	(null)	(null)	(null)	(null)	(null)	
4	SYS_C007220	Foreign_Key	(null)	SYSTEM	EMPLOYEE	SYS_C007210		

VALUES:

⚡	EMP_CODE	PHONE_NUM
1	ETN1004556	0900423543
2	ETN1004556	0902573912
3	NEU3006443	0902573451
4	SEH6003251	0928371279
5	NUD4009043	0932346133

- This table records the information about the phone numbers of each employee including: The owner EMP_Code and the phone number.
- In details, We can see that the EMP_code and the phone number is together in order to form a unique primary key. EMP_CODE is a foreign key which referencing that of EMPLOYEE.
- PHONE_NUM is fixed to be 10 bytes (10 characters) plausible with the Vietnamese phone number (as we choose a particular hospital located in Viet Nam).

2.1.2.c DEPARTMENT

DATA_TYPE:

⚡	COLUMN_NAME	DATA_TYPE	⚡	NULLABLE	DATA_DEFAULT	⚡	COLUMN_ID	⚡	COMMENTS
1	DCODE	VARCHAR2(15 BYTE)	No		(null)		1	(null)	
2	TITLE	VARCHAR2(15 BYTE)	No		(null)		2	(null)	
3	DEAN_CODE	VARCHAR2(15 BYTE)	No		(null)		3	(null)	



CONSTRAINTS:

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1	FK_DEPT_EMP_DEANCODE	Foreign_Key	(null)	SYSTEM	EMPLOYEE	SYS_C007210
2	SYS_C007211	Check	"TITLE" IS NOT NULL	(null)	(null)	(null)
3	SYS_C007212	Check	"DEAN_CODE" IS NOT NULL	(null)	(null)	(null)
4	SYS_C007213	Primary_Key	(null)	(null)	(null)	(null)
5	SYS_C007214	Unique	(null)	(null)	(null)	(null)

VALUES:

	DCODE	TITLE	DEAN_CODE
1	101	ENT	ETN1004556
2	102	Cardiology	CDG2004667
3	103	Neurology	NEU3006443
4	202	Sexual Health	SEH6003251
5	203	Nutritions	NUD4009043

- This table records the information about hospital's departments including: Department code, its title and a DEAN_CODE of the manager governing that department (Dean).
- The type of them are VARCHAR since it is no restriction had been mentioned in in the BUSINESS DESCRIPTION however 15 characters is suitable to be short and clear to some extent.
- Moreover, we make Dcode and Dean_code to be Primary key and Foreign key, referencing Emp_code of EMPLOYEE, respectively (all of them have to be unique).

2.1.2.d OUTPATIENT TABLE

DATA_TYPE:

	COLUMN_NAME	DATA_TYPE	NULLABLE
1	PCODE	CHAR(11 BYTE)	No
2	FIRST_NAME	VARCHAR2(15 BYTE)	No
3	LAST_NAME	VARCHAR2(15 BYTE)	No
4	DATE_OF_BIRTH	DATE	No
5	GENDER	VARCHAR2(6 BYTE)	No
6	ADDRESS	VARCHAR2(100 BYTE)	No
7	PHONE_NUM	CHAR(10 BYTE)	Yes

CONSTRAINTS:



	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1	SYS_C0011948	Check	"FIRST_NAME" IS NOT NULL
2	SYS_C0011949	Check	"LAST_NAME" IS NOT NULL
3	SYS_C0011950	Check	"DATE_OF_BIRTH" IS NOT NULL
4	SYS_C0011951	Check	"GENDER" IS NOT NULL
5	SYS_C0011952	Check	"ADDRESS" IS NOT NULL
6	SYS_C0011953	Check	PCode LIKE 'OP_____'
7	SYS_C0011954	Primary_Key	(null)

VALUES:

	PCODE	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	GENDER	ADDRESS	PHONE_NUM
1	OP000000000	Claud	Bernard	12-JUL-85	MALE	1490 Windy Ridge Road, Washington	0909009232
2	OP000000001	Louis	Pasteur	15-DEC-96	MALE	275 Turner Kyle Dr, Savannah, Tennessee	0903847123
3	OP000000002	Cloe	Maxwell	01-JUL-63	MALE	439 Harlem Ave, Forest Park, Illinois	0903732452
4	OP000000003	Rosa	Mcmillan	05-APR-12	FEMALE	20591 Sycamore Crest Ln, Katy, Texas	(null)
5	OP000000004	Kayla	Sheppard	12-JUN-35	FEMALE	4500 Harbour Pointe Blvd, Mukilteo, Washington	(null)
6	OP000000005	Isla	Riddle	20-OCT-91	FEMALE	319 Newberry St, Fredericktown, Missouri	0902748238
7	OP000000006	Brielle	Clark	07-MAY-57	MALE	3825 Way Crk, Chester, Virginia	0990275396
8	OP000000007	Molly	Browning	26-JUN-78	FEMALE	5550 Owensmouth Ave, Woodland Hills, California	0905839203
9	OP000000008	Emily	Bentley	20-NOV-02	FEMALE	9280 Cottonwood Dr, Bastrop, Louisiana	0904372973
10	OP000000009	Christopher	Lopez	29-FEB-08	MALE	759 A Euclatubba Rd, Saltillo, Mississippi	(null)

EXPLANATION:

- This table records the information about outpatients including: Pcode (format OP_____), Name, DOB, Gender, Address and phone number (Some patient is too young/old so they maybe do not have phone number).
- 'PCODE' is the primary key with the constraint "PCODE LIKE 'OP_____'" to check the input value must contain OP and 9 character. Ex: OP000000000. So, the data type is exactly CHAR(11).
- The columns 'FIRST_NAME', 'LAST_NAME', 'DATE_OF_BIRTH', 'GENDER', 'ADDRESS' is NOT NULL while 'PHONE_NUM' is nullable. Therefore, the doctor can know the age, status of the patient to give the precisely diagnosis.

2.1.2.e EXAMINE TABLE

DATA_TYPE:

	COLUMN_NAME	DATA_TYPE	NULLABLE
1	MED_CODE	VARCHAR2(15 BYTE)	No
2	DOCTOR_CODE	VARCHAR2(15 BYTE)	No
3	PATIENT_CODE	CHAR(11 BYTE)	No



CONSTRAINTS:

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME
1 FK_EXA_MED_DOCTOR_CODE	Foreign_Key	(null)	SYSTEM	EMPLOYEE
2 FK_EXA_MED_MED_CODE	Foreign_Key	(null)	SYSTEM	MEDICATION
3 FK_EXA_OUTPAT_PATIENT_CODE	Foreign_Key	(null)	SYSTEM	OUTPATIENT
4 SYS_C0011972	Check	"MED_CODE" IS NOT NULL	(null)	(null)
5 SYS_C0011973	Check	"DOCTOR_CODE" IS NOT NULL	(null)	(null)
6 SYS_C0011974	Check	"PATIENT_CODE" IS NOT NULL	(null)	(null)
7 SYS_C0011975	Primary_Key	(null)	(null)	(null)

VALUES:

	MED_CODE	DOCTOR_CODE	PATIENT_CODE
1	J01XA05	ETN1004556	OP000000000
2	N02BE01	ETN1004556	OP000000000
3	N05CF03	ETN1004556	OP000000000
4	N02BE01	ETN1004556	OP000000001
5	J01XA05	ETN1004556	OP000000001
6	J01XA05	ETN1004556	OP000000002
7	N02BE01	ETN1004556	OP000000002
8	N05CF03	ETN1004556	OP000000002
9	L01CA06	ETN1004556	OP000000002
10	J01XA05	ETN1004556	OP000000003
11	N02BE01	ETN1004556	OP000000003
12	A12CC03	ETN1004556	OP000000003
13	N05CF03	NEU3006443	OP000000004
14	N02AD02	SEH6003251	OP000000005
15	A12CC03	SEH6003251	OP000000005
16	L01CA06	CDG2004667	OP000000006
17	N02AD02	CDG2004667	OP000000006
18	L01CA06	CDG2004667	OP000000007
19	A07AB04	CDG2004667	OP000000007
20	N02AD02	SEH6003251	OP000000008
21	A12CC03	SEH6003251	OP000000008
22	A07AB04	NUD4009043	OP000000009
23	A12CC03	NUD4009043	OP000000009

EXPLANATION:

- This table records the overall information about examinations for outpatients including: Pcode (format OP _____) together with their doctor and medication.
- "MED_CODE" and "PATIENT_CODE" are the primary key. The "DOCTOR_CODE" is not chosen to be primary key, was explained in the section Relational Database.
- All the three columns is NOT NULL.

2.1.2.f EXAMINE_DETAIL TABLE

DATA_TYPE:



	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE
1	MED_CODE	VARCHAR2 (15 BYTE)	No
2	PATIENT_CODE	CHAR (11 BYTE)	No
3	EXAMINE_DATE	DATE	No
4	NEXT_DATE	DATE	Yes
5	DIAGNOSIS	VARCHAR2 (30 BYTE)	No
6	FEE	NUMBER (8, 2)	No

CONSTRAINTS:

	↕ CONSTRAINT_NAME	↕ CONSTRAINT_TYPE	SEARCH_CONDITION
1	FK_EXADE_EXA_MED_CODE_PAT	Foreign_Key	(null)
2	SYS_C0011979	Check	"MED_CODE" IS NOT NULL
3	SYS_C0011980	Check	"PATIENT_CODE" IS NOT NULL
4	SYS_C0011981	Check	"EXAMINE_DATE" IS NOT NULL
5	SYS_C0011982	Check	"DIAGNOSIS" IS NOT NULL
6	SYS_C0011983	Check	"FEE" IS NOT NULL
7	SYS_C0011984	Primary_Key	(null)

VALUES:



	MED_CODE	PATIENT_CODE	EXAMINE_DATE	NEXT_DATE	DIAGNOSIS	FEE
1	J01XA05	OP000000000	10-JUL-21	12-JUL-21	Fever	99
2	N02BE01	OP000000000	12-JUL-21	14-JUL-21	Fever	99
3	N05CF03	OP000000000	14-JUL-21	(null)	Fever	199
4	N02BE01	OP000000001	02-AUG-21	05-AUG-21	Fever virus	199
5	J01XA05	OP000000001	05-AUG-21	(null)	Fever virus	199
6	J01XA05	OP000000002	14-SEP-21	15-SEP-21	Chronic sinusitis	99
7	N02BE01	OP000000002	15-SEP-21	17-SEP-21	Fever	99
8	N05CF03	OP000000002	17-SEP-21	19-SEP-21	Chronic sinusitis	199
9	L01CA06	OP000000002	19-SEP-21	(null)	Chronic sinusitis	199
10	J01XA05	OP000000003	01-OCT-21	03-OCT-21	Fever	99
11	N02BE01	OP000000003	03-OCT-21	05-OCT-21	Fever	199
12	A12CC03	OP000000003	05-OCT-21	(null)	Fever	199
13	N05CF03	OP000000004	05-SEP-21	12-SEP-21	Parkinson	499
14	N05CF03	OP000000004	12-SEP-21	19-SEP-21	Parkinson	499
15	N05CF03	OP000000004	19-SEP-21	26-SEP-21	Dementia	499
16	N05CF03	OP000000004	26-SEP-21	(null)	Dementia	499
17	N02AD02	OP000000005	11-AUG-21	25-AUG-21	Regular antenatal check-up	119
18	A12CC03	OP000000005	25-AUG-21	06-SEP-21	Regular antenatal check-up	79
19	A12CC03	OP000000005	06-SEP-21	(null)	Regular antenatal check-up	79
20	L01CA06	OP000000006	15-SEP-21	22-SEP-21	Coronary artery disease	399
21	L01CA06	OP000000006	22-SEP-21	29-SEP-21	Coronary artery disease	399
22	N02AD02	OP000000006	29-SEP-21	(null)	Coronary artery disease	119
23	L01CA06	OP000000007	26-SEP-21	03-OCT-21	Breath heavily	399
24	L01CA06	OP000000007	03-OCT-21	10-OCT-21	Swollen feet, legs, ascites	399
25	A07AB04	OP000000007	10-OCT-21	(null)	Heart failure level 2	219
26	N02AD02	OP000000008	01-OCT-21	05-OCT-21	Adolescent pregnancy	119
27	N02AD02	OP000000008	05-OCT-21	12-OCT-21	Adolescent pregnancy	119
28	A12CC03	OP000000008	12-OCT-21	(null)	Adolescent pregnancy	79
29	A07AB04	OP000000009	25-OCT-21	02-NOV-21	Diarrhea	219
30	A12CC03	OP000000009	02-NOV-21	(null)	Prolonged malnutrition	79

EXPLANATION:

- This table records the detailed information about each examination for each outpatient including: Pcode (format OP_____) together with their medication, Date, Next date, diagnosis and fee.
- "MED_CODE", "PATIENT_CODE", and "EXAMINE_DATE" are the primary key
- All the three columns is NOT NULL except "NEXT_DATE" because the last examination for each patient do not need the next date.
- The column "FEE" has data type NUMBER(8,2) so it can easily calculated the list of fee for each patient in part 2.



2.1.2.g INPATIENT TABLE

DATA_TYPE:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PCODE	CHAR(11 BYTE)	No	(null)	1 (null)	
2	DATE_OF_ADMISSION	DATE	No	(null)	2 (null)	
3	DIAGNOSIS	VARCHAR2(30 BYTE)	No	(null)	3 (null)	
4	SICKROOM	VARCHAR2(15 BYTE)	No	(null)	4 (null)	
5	FEE	NUMBER(8,2)	No	(null)	5 (null)	
6	FIRST_NAME	VARCHAR2(15 BYTE)	No	(null)	6 (null)	
7	LAST_NAME	VARCHAR2(15 BYTE)	No	(null)	7 (null)	
8	DAY_OF_BIRTH	DATE	No	(null)	8 (null)	
9	GENDER	VARCHAR2(6 BYTE)	No	(null)	9 (null)	
10	ADDRESS	VARCHAR2(100 BYTE)	No	(null)	10 (null)	
11	PHONE_NUM	CHAR(10 BYTE)	Yes	(null)	11 (null)	
12	NURSE_CODE	VARCHAR2(15 BYTE)	No	(null)	12 (null)	

CONSTRAINTS:

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE
1	SYS_C0012539	Check	"DATE_OF_ADMISSION" IS NOT NULL	(null)	(null)	(null)	(null)
2	SYS_C0012540	Check	"DIAGNOSIS" IS NOT NULL	(null)	(null)	(null)	(null)
3	SYS_C0012541	Check	"SICKROOM" IS NOT NULL	(null)	(null)	(null)	(null)
4	SYS_C0012542	Check	"FEE" IS NOT NULL	(null)	(null)	(null)	(null)
5	SYS_C0012543	Check	"FIRST_NAME" IS NOT NULL	(null)	(null)	(null)	(null)
6	SYS_C0012544	Check	"LAST_NAME" IS NOT NULL	(null)	(null)	(null)	(null)
7	SYS_C0012545	Check	"DAY_OF_BIRTH" IS NOT NULL	(null)	(null)	(null)	(null)
8	SYS_C0012546	Check	"GENDER" IS NOT NULL	(null)	(null)	(null)	(null)
9	SYS_C0012547	Check	"ADDRESS" IS NOT NULL	(null)	(null)	(null)	(null)
10	SYS_C0012548	Check	"NURSE_CODE" IS NOT NULL	(null)	(null)	(null)	(null)
11	SYS_C0012549	Check	PCode LIKE 'IP_____'	(null)	(null)	(null)	(null)
12	SYS_C0012550	Primary_Key	(null)	(null)	(null)	(null)	(null)

VALUES:

	PCODE	DATE_OF_ADMISSION	DIAGNOSIS	SICKROOM	FEE	FIRST_NAME	LAST_NAME	DAY_OF_BIRTH	GENDER	ADDRESS	PHONE_NUM	NURSE_CODE
1	IP205679239	06-10-2019	Catastrophe	A202	1099,99	Claud	Riddle	06-10-1953	MALE	33 Windy Ridge Road, Washington	0909009900	NUD4009043
2	IP945829231	08-08-2020	Liver Cancer	A109	2092,89	Louis	Clark	08-08-1924	MALE	275 Turner Kyle Dr, Savannah, Tennessee	0909345940	ETN1005776
3	IP205127452	26-09-2021	Pneumonia	A391	3942,99	Cloe	Browning	26-09-1930	MALE	439 Harlem Ave, Forest Park, Illinois	0909023464	ETN1005776
4	IP203462341	03-10-2021	Traumatic brain injury	B029	1529,99	Rosa	Lopez	03-10-1989	FEMALE	20591 Sycamore Crest Ln, Katy, Texas	0909056342	NUD4009043
5	IP104232231	03-10-2021	Appendicitis	B823	9238,99	Kayla	Bernard	10-10-1983	FEMALE	4500 Harbour Pointe Blvd, Mukilteo, Washington	0903457345	SEH6003822
6	IP923958724	10-10-2021	Cirrhosis	C215	7792,99	Isla	Pasteur	23-06-1965	FEMALE	319 Newberry St, Fredericktown, Missouri	0943476232	ETN1005776
7	IP383739276	23-05-2020	Pneumonia	B241	3942,99	Brielle	Maxwell	23-01-1945	MALE	3825 Way Crk, Chester, Virginia	0908345624	NUD4009043

EXPLANATION:

- This table records the overall information about inpatients including: Pcode (format IP_____), DATE_OF_ADMISSION (DATE), DIAGNOSIS is in type of varchar with maximum length of 30 bytes, GENDER (varchar(6)) because it can has value 'MALE' OR 'FEMALE', PHONE_NUM can be NULL (some people doesn't have any phone number), NURSE_CODE is an foreign key that refers to EMP_CODE in table EMPLOYEE.
- PCode should be check if it has the form IP_____ or not.



- All the other columns but PHONE_NUM are NOT NULL.
- PCode is a primary key of the table.

2.1.2.h TREAT TABLE

DATA_TYPE:

	⚡ COLUMN_NAME	⚡ DATA_TYPE	⚡ NULLABLE
1	MED_CODE	VARCHAR2(15 BYTE)	No
2	PATIENT_CODE	CHAR(11 BYTE)	No
3	DOCTOR_CODE	VARCHAR2(15 BYTE)	No

CONSTRAINTS:

	⚡ CONSTRAINT_NAME	⚡ CONSTRAINT_TYPE	SEARCH_CONDITION	⚡ R_OWNER	⚡ R_TABLE_NAME	⚡ R_CONSTRAINT_NAME	⚡ DELETE_RULE
1	SYS_C0011986	Primary_Key	(null)	(null)	(null)	(null)	(null)
2	SYS_C0011987	Foreign_Key	(null)	SYSTEM	MEDICATION	SYS_C0011955	CASCADE
3	SYS_C0011988	Foreign_Key	(null)	SYSTEM	INPATIENT	SYS_C007575	CASCADE
4	SYS_C0011989	Foreign_Key	(null)	SYSTEM	EMPLOYEE	SYS_C0011937	CASCADE

VALUES:

	MED_CODE	PATIENT_CODE	DOCTOR_CODE
1	J01XA05	IP205679239	CDG2004667
2	N02BE01	IP945829231	ETN1004556
3	N02BE01	IP205127452	NUD4009043
4	N05CF03	IP203462341	ETN1004556
5	N05CF03	IP104232231	NUD4009043
6	L01CA06	IP923958724	ETN1004556
7	L01CA06	IP383739276	NUD4009043
8	L01CA06	IP205679239	CDG2004667
9	A07AB04	IP945829231	ETN1004556
10	A07AB04	IP205127452	ETN1004556
11	A07AB04	IP203462341	ETN1004556
12	N02AD02	IP104232231	ETN1004556
13	A12CC03	IP923958724	NEU3006443
14	A07AB04	IP383739276	SEH6003251
15	A07AB04	IP205679239	NEU3006443
16	A12CC03	IP945829231	NEU3006443
17	A12CC03	IP205127452	SEH6003251
18	L01CA06	IP203462341	SEH6003251
19	L01CA06	IP104232231	SEH6003251
20	J01XA05	IP923958724	CDG2004667
21	N02AD02	IP383739276	CDG2004667

EXPLANATION:

- This table records the overall information about treatments for inpatients including: Pcode (format OP_____) together with their doctor and medication.
- "MED_CODE", "PATIENT_CODE" and "DOCTOR_CODE" are the primary key.
- All the three columns is NOT NULL.

2.1.2.i TREATMENT_DETAIL TABLE

DATA_TYPE:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	MED_CODE	VARCHAR2(15 BYTE)	No	(null)	1 (null)	
2	PATIENT_CODE	CHAR(11 BYTE)	No	(null)	2 (null)	
3	DOCTOR_CODE	VARCHAR2(15 BYTE)	No	(null)	3 (null)	
4	START_DAY	DATE	No	(null)	4 (null)	
5	END_DAY	DATE	Yes	(null)	5 (null)	
6	RESULT	VARCHAR2(30 BYTE)	Yes	(null)	6 (null)	
7	IS_COVERED	CHAR(1 BYTE)	Yes	(null)	7 (null)	

CONSTRAINTS:



CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1 SYS_C007271	Check	"MED_CODE" IS NOT NULL
2 SYS_C007272	Check	"PATIENT_CODE" IS NOT NULL
3 SYS_C007273	Check	"DOCTOR_CODE" IS NOT NULL
4 SYS_C007274	Check	"START_DAY" IS NOT NULL
5 SYS_C007275	Check	Is_covered IN ('Y', 'N')
6 SYS_C007276	Primary_Key	(null)
7 SYS_C007277	Foreign_Key	(null)

VALUES:

	MED_CODE	PATIENT_CODE	DOCTOR_CODE	START_DAY	END_DAY	RESULT	IS_COVERED
1	J01XA05	IP205679239	CDG2004667	06-OCT-19	07-NOV-19	Catastrophe_initial diag	N
2	N02BE01	IP945829231	ETN1004556	08-AUG-20	09-SEP-20	Liver Cancer_initial diag	N
3	N02BE01	IP205127452	NUD4009043	26-SEP-21	27-OCT-21	Pneumonia_initial diag	N
4	N05CF03	IP203462341	ETN1004556	03-OCT-21	04-NOV-21	Traumatic brain injury diag	N
5	N05CF03	IP104232231	NUD4009043	10-OCT-21	11-NOV-21	Appendicitis_initial diag	N
6	L01CA06	IP923958724	ETN1004556	01-OCT-21	02-DEC-21	Cirrhosis_initial diag	N
7	L01CA06	IP383739276	NUD4009043	23-MAY-20	24-JUN-20	Pneumonia_initial diag	N
8	L01CA06	IP205679239	CDG2004667	05-JUL-20	06-DEC-20	Catastrophe	N
9	A07AB04	IP945829231	ETN1004556	25-OCT-21	26-NOV-21	Liver Cancer	N
10	A07AB04	IP205127452	ETN1004556	27-NOV-21	28-DEC-21	Pneumonia	N
11	A07AB04	IP203462341	ETN1004556	22-MAY-22	23-JUN-22	Traumatic brain injury	N
12	N02AD02	IP104232231	ETN1004556	18-NOV-18	19-JAN-19	Appendicitis	N
13	A12CC03	IP923958724	NEU3006443	04-DEC-21	05-MAR-22	Cirrhosis	N
14	A07AB04	IP383739276	SEH6003251	06-OCT-19	07-NOV-19	Pneumonia	N
15	A07AB04	IP205679239	NEU3006443	08-AUG-20	09-SEP-20	Post-operative follow-up	Y
16	A12CC03	IP945829231	NEU3006443	26-SEP-21	27-SEP-21	Post-operative follow-up	Y
17	A12CC03	IP205127452	SEH6003251	03-OCT-21	04-NOV-21	Post-operative follow-up	Y
18	L01CA06	IP203462341	SEH6003251	10-OCT-21	11-DEC-21	Post-operative follow-up	Y
19	L01CA06	IP104232231	SEH6003251	01-OCT-21	02-NOV-21	Post-operative follow-up	Y
20	J01XA05	IP923958724	CDG2004667	23-MAY-20	24-JUL-20	Post-operative follow-up	Y
21	N02AD02	IP383739276	CDG2004667	05-JUL-20	06-AUG-20	Post-operative follow-up	Y

EXPLANATION:

- This table records the overall information about the detail of the treatment of Outpatient.
- The primary key consists of 4 fields: "MED_CODE", "PATIENT_CODE", "DOCTOR_CODE" and "Start_day".
- Only End_day and "Result" fields are allowed to be NULL because they are undetermined before the end of the treatment.
- "Is_covered" is a character with length of 1 and only allowing 'Y' or 'N' value.
- This table have 3 foreign keys: "Med_code", "Patient_code", "Doctor_code".
The specifying constraint for "Patient_code" and "Doctor_code" is ON DELETE CASCADE since once the patient or doctor isn't belong to the database, all the information about treatment or examine related to that person no longer useful so the database will delete all of them.

The specifying constraint for "Med_code" is ON DELETE RESTRICT since we don't want any tuple in TREATMENT_DELTAIL table is deleted when a type of medicine is deleted. All the treatment_detail information of a patient will be reserved and the system will give a warning when we try to delete a medicine which have data from treatment_detail table referring to.

2.1.2.j PROVIDER TABLE

DATA_TYPE:

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	PNUMBER	VARCHAR2(15 BYTE)	No	(null)	1 (null)	
2	PNAME	VARCHAR2(30 BYTE)	Yes	(null)	2 (null)	
3	ADDRESS	VARCHAR2(100 BYTE)	Yes	(null)	3 (null)	
4	PHONE_NUM	CHAR(10 BYTE)	Yes	(null)	4 (null)	

CONSTRAINTS:

	❖ CONSTRAINT_NAME	❖ CONSTRAINT_TYPE	SEARCH_CONDITION	❖ R_OWNER	❖ R_TABLE_NAME	❖ R_CONSTRAINT_NAME	❖ DELETE_RULE
1	SYS_C0012473	Primary_Key	(null)	(null)	(null)	(null)	(null)
2	SYS_C0012474	Unique	(null)	(null)	(null)	(null)	(null)

VALUES:

	❖ PNUMBER	❖ PNAME	❖ ADDRESS	❖ PHONE_NUM
1	US7711951043	Roche	Basel, Switzerland	0902002124
2	US7170811035	Pfizer	New York, United States	0882001251
3	CH0012005267	Novartis	Basel, Switzerland	0907012282
4	US4781601046	Johnson and Johnson	New Jersey, United States	0991030201

EXPLANATION:

- This table records the overall information about provider, it includes: Pnumber is an identify number of a provider with datatype VARCHAR and maximum size is 15 bytes because we never know whether Pnumber's length is fixed or not. Similar with Pname (with maximum size is 30 bytes) and Address (with maximum size is 100 bytes).
- There are 2 constraint: Pnumber is a primary key and Pname is unique.

2.1.2.k MEDICATION TABLE

DATA_TYPE:



	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	MCODE	VARCHAR2(15 BYTE)	No	(null)	1 (null)	
2	MNAME	VARCHAR2(30 BYTE)	Yes	(null)	2 (null)	
3	PRICE	NUMBER(8,2)	Yes	(null)	3 (null)	

CONSTRAINTS:

	❖ CONSTRAINT_NAME	❖ CONSTRAINT_TYPE	SEARCH_CONDITION	❖ R_OWNER	❖ R_TABLE_NAME	❖ R_CONSTRAINT_NAME	❖ DELETE_RULE
1	SYS_C0012495	Primary_Key	(null)	(null)	(null)	(null)	(null)
2	SYS_C0012496	Unique	(null)	(null)	(null)	(null)	(null)

VALUES:

	❖ MCODE	❖ MNAME	❖ PRICE
1	N02BE01	Paracetamol	199,99
2	N05CF03	Zaleplon	449,89
3	J01XA05	Oritavancin	99,89
4	L01CA06	Vintafolide	399,99
5	N02AD02	Phenazocine	119
6	A07AB04	Succinylsulfathiazole	219,99
7	A12CC03	Magnesium gluconate	79,99

EXPLANATION:

- This table records the overall information about medications including: Mcode with datatype varchar (maximum size is 15 bytes), Mname with datatype varchar (maximum size is 30 bytes) and its price with datatype number (total number of digit is 8 and number of digit after the decimal point is 2).
- There are 2 constraint: Mcode is a primary key and Mname is unique.

2.1.2.1 IMPORTED_FROM TABLE

DATA_TYPE:

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	MED_CODE	VARCHAR2(15 BYTE)	No	(null)	1 (null)	
2	PROVIDER_NUM	VARCHAR2(15 BYTE)	No	(null)	2 (null)	

CONSTRAINTS:



CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE
1 SYS_C0012505	Check	"MED_CODE" IS NOT NULL	(null)	(null)	(null)	(null)
2 SYS_C0012506	Check	"PROVIDER_NUM" IS NOT NULL	(null)	(null)	(null)	(null)
3 SYS_C0012507	Primary_Key	(null)	(null)	(null)	(null)	(null)
4 SYS_C0012508	Foreign_Key	(null)	HR	MEDICATION	SYS_C0012495	CASCADE
5 SYS_C0012509	Foreign_Key	(null)	HR	PROVIDER	SYS_C0012473	CASCADE

VALUES:

	MED_CODE	PROVIDER_NUM
1	N02BE01	US4781601046
2	N02BE01	US7711951043
3	L01CA06	US7170811035
4	L01CA06	CH0012005267
5	N05CF03	US4781601046
6	N05CF03	US7711951043
7	J01XA05	CH0012005267
8	A07AB04	US4781601046
9	N02AD02	CH0012005267
10	A12CC03	US7170811035

EXPLANATION:

- This table records the overall information about importation of medications including: MED_CODE as above and PROVIDER_NUM.
- "MED_CODE" is a foreign key refers to MCode in MEDICATION table. "PROVIDER_NUM" is a foreign key refers to PNumber in PROVIDER table. And there combination form the primary key of the table.
- All the two columns are NOT NULL.

2.1.2.m IMPORTED_INFO TABLE

DATA_TYPE:



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	MED_CODE	VARCHAR2(15 BYTE)	No	(null)	1 (null)	
2	PROVIDER_NUM	VARCHAR2(15 BYTE)	No	(null)	2 (null)	
3	IMPORTED_DATE	DATE	No	(null)	3 (null)	
4	IMPORTED_PRICE	NUMBER(8,2)	No	(null)	4 (null)	
5	IMPORTED_QUANTITY	NUMBER(38,0)	No	(null)	5 (null)	

CONSTRAINTS:

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE
1	SYS_C0012510	Check	"IMPORTED_PRICE" IS NOT NULL	(null)	(null)	(null)	(null)
2	SYS_C0012511	Check	"IMPORTED_QUANTITY" IS NOT NULL	(null)	(null)	(null)	(null)
3	SYS_C0012512	Primary_Key	(null)	(null)	(null)	(null)	(null)
4	SYS_C0012513	Foreign_Key	(null)	HR	IMPORTED_FROM	SYS_C0012507	CASCADE

VALUES:

	MED_CODE	PROVIDER_NUM	IMPORTED_DATE	IMPORTED_PRICE	IMPORTED_QUANTITY
1	N02BE01	US4781601046	12-09-2019	99,99	20000
2	N02BE01	US7711951043	04-12-2019	79,89	30000
3	L01CA06	US7170811035	22-02-2019	59,99	45000
4	L01CA06	CH0012005267	16-07-2019	49	15000
5	N05CF03	US4781601046	08-04-2019	199,89	25000
6	N05CF03	US7711951043	15-04-2019	189,89	4000
7	J01XA05	CH0012005267	12-10-2019	159,49	1500
8	A07AB04	US4781601046	11-04-2019	199,49	1000
9	N02AD02	CH0012005267	04-05-2019	99,99	2500
10	A12CC03	US7170811035	27-10-2018	49,49	9000

EXPLANATION:

- This table records the detail information about importations including: MED_CODE and PROVIDER_NUM as above, IMPORTED_DATE in type of DATE stores the date that given medication was imported, IMPORTED_PRICE in type of NUMBER with total number of digit is 8 and number of digit after the decimal point is 2. Similar with IMPORTED_QUANTITY in type of INT with maximum number of digit is 38.
- IMPORTED_PRICE and IMPORTED_QUANTITY are NOT NULL.

2.1.2.n MED_EFFECT TABLE

DATA_TYPE:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	MED_CODE	VARCHAR2(15 BYTE)	No	(null)	1 (null)	
2	EFFECT	VARCHAR2(30 BYTE)	No	(null)	2 (null)	



CONSTRAINTS:

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE
1 SYS_C0012497	Check	"EFFECT" IS NOT NULL	(null)	(null)	(null)	(null)
2 SYS_C0012498	Primary_Key	(null)	(null)	(null)	(null)	(null)
3 SYS_C0012499	Foreign_Key	(null)	HR	MEDICATION	SYS_C0012495	CASCADE

VALUES:

	MED_CODE	EFFECT
1	N02BE01	analgesic
2	N02BE01	antipyretic
3	N05CF03	sedative
4	N05CF03	soothing
5	L01CA06	immunomodulatory
6	L01CA06	anti-cancer
7	J01XA05	antimicrobial
8	A07AB04	treat diarrhea
9	A07AB04	enteritis
10	A07AB04	anti-infective
11	N02AD02	analgesic
12	A12CC03	mineral supplements

EXPLANATION:

- This table records the overall information about effects of medications including: MED_CODE is identifying number of medication with type varchar as above and EFFECT in type varchar with maximum size is 30 bytes.
- "MED_CODE" is a foreign key refers to MCode in MEDICATION table. "MED_CODE" and "EFFECT" form the primary key.
- EFFECT is NOT NULL.

2.1.2.o MED_INFO TABLE

DATA_TYPE:



	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	MED_CODE	VARCHAR2(15 BYTE)	No	(null)	1 (null)	
2	IMPORTED_DATE	DATE	No	(null)	2 (null)	
3	EXPIRATION_DATE	DATE	No	(null)	3 (null)	
4	IS_OUT_OF_DATE	CHAR(1 BYTE)	Yes	(null)	4 (null)	

CONSTRAINTS:

	❖ CONSTRAINT_NAME	❖ CONSTRAINT_TYPE	SEARCH_CONDITION	❖ R_OWNER	❖ R_TABLE_NAME	❖ R_CONSTRAINT_NAME	❖ DELETE_RULE
1	SYS_C0012500	Check	"IMPORTED_DATE" IS NOT NULL	(null)	(null)	(null)	(null)
2	SYS_C0012501	Check	"EXPIRATION_DATE" IS NOT NULL	(null)	(null)	(null)	(null)
3	SYS_C0012502	Check	Is_out_of_date IN ('Y', 'N')	(null)	(null)	(null)	(null)
4	SYS_C0012503	Primary_Key	(null)	(null)	(null)	(null)	(null)
5	SYS_C0012504	Foreign_Key	(null)	HR	MEDICATION	SYS_C0012495	CASCADE

VALUES:

	❖ MED_CODE	❖ IMPORTED_DATE	❖ EXPIRATION_DATE	❖ IS_OUT_OF_DATE
1	N02BE01	12-09-2019	30-04-2022	N
2	N02BE01	04-12-2019	21-03-2022	N
3	L01CA06	22-02-2019	15-08-2021	Y
4	L01CA06	16-07-2019	10-03-2022	N
5	N05CF03	08-04-2019	26-02-2022	N
6	N05CF03	15-04-2019	07-08-2021	Y
7	J01XA05	12-10-2019	26-07-2021	Y
8	A07AB04	11-04-2019	25-03-2022	N
9	N02AD02	04-05-2019	27-12-2020	Y
10	A12CC03	27-10-2018	03-02-2020	Y

EXPLANATION:

- This table records the more information about medications including: MED_CODE as above, IMPORTED_DATE in type DATE to stores the date that the medications was imported, similar with EXPIRATION_DATE (datatype DATE) is the date that the medications will be expired and IS_OUT_OF_DATE in type CHAR with 1 byte fix length to check if the given medication is out of date or not.
- "MED_CODE" is a foreign key refers to MCode in MEDICATION table. MED_CODE, IMPORTED_DATE and EXPIRATION_DATE form the primary key of the table.
- IMPORTED_DATE AND EXPIRATION_DATE is NOT NULL. There are also a CHECK constraint type that check IS_OUT_OF_DATE has 'Y' or 'N' value, not the other word.

2.2 Part 2: Store Procedure / Function / SQL

2.2.1 Question a: Increase Inpatient Fee to 10% for all the inpatients who are admitted to hospital from 01/09/2020

First of all, we will update the fee of current data in IMPATIENT TABLE by using the procedure **Update_Fee**

```

1  --Date format 'dd-mm-yy'
2  create or replace PROCEDURE Update_Fee
3  AS
4  BEGIN
5      UPDATE INPATIENT
6      SET Fee = Fee * 1.1
7      WHERE DATE_OF_ADMISSION >= '01-09-2020';
8  END;
```

The value of **Fee** in INPATIENT TABLE initially:

	PCODE	DATE_OF_ADMISSION	DIAGNOSIS	SICKROOM	FEE
1	IP205679239	06-OCT-19	Catastrophe	A202	1099.99
2	IP945829231	08-AUG-20	Liver Cancer	A109	2092.89
3	IP205127452	26-SEP-21	Pneumonia	A391	3942.99
4	IP383739276	23-MAY-20	Pneumonia	B241	3942.99
5	IP203462341	03-OCT-21	Traumatic brain injury	B029	1529.99
6	IP104232231	03-OCT-21	Appendicitis	B823	9238.99
7	IP923958724	10-OCT-21	Cirrhosis	C215	7792.99

Running the procedure using statements:

```

1  --Date format 'dd-mm-yy'
2  ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';
3  exec update_fee;
```

The value of **Fee** in INPATIENT TABLE after update:

	PCODE	DATE_OF_ADMISSION	DIAGNOSIS	SICKROOM	FEE
1	IP205679239	06-10-2019	Catastrophe	A202	1099.99
2	IP945829231	08-08-2020	Liver Cancer	A109	2092.89
3	IP205127452	26-09-2021	Pneumonia	A391	4337.29
4	IP383739276	23-05-2020	Pneumonia	B241	3942.99
5	IP203462341	03-10-2021	Traumatic brain injury	B029	1682.99
6	IP104232231	03-10-2021	Appendicitis	B823	10162.89
7	IP923958724	10-10-2021	Cirrhosis	C215	8572.29

Base on **Date_of_admission** of each inpatient, the inpatient number 3, 5, 6, and 7 are the those who are admitted to hospital from or later than 01/09/2020 have their **Fee** increased by 10%, from \$3492.9 to \$4337.29, from \$1529.99 to \$1682.99, from \$9238.99 to \$10162.89 and from \$7792.99 to \$8572.29, respectively.

Finally, we create a trigger to update the **Fee** automatically whenever we insert the new values into INPATIENT. This trigger will compare the **Date_of_admission** and increase the **Fee** if the condition is satisfied.

```
1  --Date format 'dd-mm-yy'
2  create or replace TRIGGER Auto_Increase_Fee
3  BEFORE INSERT ON INPATIENT
4  FOR EACH ROW
5  BEGIN
6      if (:NEW.DATE_OF_ADMISSION >= '01-09-2020') then
7          :NEW.FEE := :NEW.FEE * 1.1;
8      end if;
9  END;
```

To test the trigger working correctly, we will insert two new inpatients (id are 'IP000000000' and 'IP000000001', respectively) into INPATIENT table, the first has **Date_of_admission** is 31/08/2020 before 01/09/2020 while the second one has satisfied **Date_of_admission** which is 02/09/2020. Both new inpatients has the initially **Fee** is \$2000.

```
1  --Date format 'dd-mm-yy'
2  ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';
3  INSERT INTO INPATIENT VALUES ('IP000000000', '31-08-2020', 'Catastrophe',
4  'A282', 2000, 'ABC', 'XYZ', '10-06-2001', 'M', '33 Windy Re hjoad,
Washington', '0909009800', 'NUD6969696');
5  INSERT INTO INPATIENT VALUES ('IP000000001', '02-09-2020', 'Catastrophe',
6  'A282', 2000, 'CDF', 'IJK', '10-06-2001', 'F', '33 Windy Re hjoad,
Washington', '0909009811', 'NUD6969696');
```

Result in INPATIENT table:

	PCODE	DATE_OF_ADMISSION	DIAGNOSIS	SICKROOM	FEE
1	IP205679239	06-10-2019	Catastrophe	A202	1099.99
2	IP945829231	08-08-2020	Liver Cancer	A109	2092.89
3	IP205127452	26-09-2021	Pneumonia	A391	4337.29
4	IP383739276	23-05-2020	Pneumonia	B241	3942.99
5	IP203462341	03-10-2021	Traumatic brain injury	B029	1682.99
6	IP104232231	03-10-2021	Appendicitis	B823	10162.89
7	IP923958724	10-10-2021	Cirrhosis	C215	8572.29
8	IP000000000	31-08-2020	Catastrophe	A282	2000
9	IP000000001	02-09-2020	Catastrophe	A282	2200

The **Fee** of satisfied inpatient (id is 'IP000000001') increases to \$2200. So the trigger was working as expected.

2.2.2 Question b: Select all the patients (outpatient & inpatient) of the doctor named 'Nguyen Van A'

We created a procedure **patient_of** with parameter **doctor_name** in type VARCHAR. This procedure will print out list of patients of the given doctor.

```
1 CREATE OR REPLACE PROCEDURE patient_of
2 (
3     doctor_name IN VARCHAR2
4 )
5 AS
6 BEGIN
7     DBMS_OUTPUT.PUT_LINE('Patient Code | Fullname');
8     DBMS_OUTPUT.PUT_LINE('-----|-----');
9     FOR rec IN (
10        SELECT
11            pcode AS "PATIENT CODE",
12            CONCAT(CONCAT(first_name, ' '), last_name) AS "FULL NAME"
13        FROM inpatient JOIN treat
14            ON inpatient.pcode = treat.patient_code
15        WHERE
16            treat.doctor_code = (
17                SELECT emp_code
18                FROM employee
19                WHERE CONCAT(CONCAT(employee.first_name, ' '),
20                    employee.last_name) = doctor_name
21            )
22        UNION
23        SELECT
24            pcode AS "PATIENT CODE",
25            CONCAT(CONCAT(first_name, ' '), last_name) AS "FULL NAME"
26        FROM outpatient JOIN examine
27            ON outpatient.pcode = examine.patient_code
28        WHERE
29            examine.doctor_code = (
30                SELECT emp_code
31                FROM employee
32                WHERE CONCAT(CONCAT(employee.first_name, ' '),
33                    employee.last_name) = doctor_name
34            )
35        )
36    LOOP
37        DBMS_OUTPUT.PUT_LINE(rec."PATIENT CODE" || ' | ' || rec."FULL
38            NAME");
39    END LOOP;
40 END;
```

RESULT: To run the procedure, we run the following statement:


```
1 EXEC patient_of('William Osler');
```

This statement will print out list of patients of doctor whose name "William Osler":

Patient Code	Fullname
IP104232231	Kayla Bernard
IP203462341	Rosa Lopez
IP205127452	Cloe Browning
IP923958724	Isla Pasteur
IP945829231	Louis Clark
OP000000000	Claud Bernard
OP000000001	Louis Pasteur
OP000000002	Cloe Maxwell
OP000000003	Rosa Mcmillan

2.2.3 Question c: Write a function to calculate the total medication price a patient has to pay for each treatment or examination

We output the required information about **medicine price** and **fee** for an input patient by using this function:

```
1 create or replace FUNCTION total_fee
2 (
3     input_patient_code IN CHAR
4 )
5 RETURN NUMBER
6 AS
7     --patient_code inpatient.pcode%TYPE;
8     --medication examine_detail.med_code%TYPE;
9     total_fee NUMBER(8,2);
10    total_medication_price NUMBER(8,2);
11    fee NUMBER(8,2);
12    --start_date DATE;
13    --end_date DATE;
14    --diagnosis VARCHAR2(50);
15 BEGIN
16    DBMS_OUTPUT.PUT_LINE ('PAYMENT REPORT');
17    DBMS_OUTPUT.PUT_LINE ('Patient ID: ' || input_patient_code);
18
19    IF input_patient_code LIKE 'OP_-----' THEN
20        BEGIN
```

```
21      SELECT SUM( medication.price) INTO total_medication_price
22      FROM examine_detail INNER JOIN medication
23      ON examine_detail.med_code = medication.mcode
24      WHERE examine_detail.patient_code = input_patient_code;
25      DBMS_OUTPUT.PUT_LINE ('Total Medication price: ' ||
26      total_medication_price);
27
28      SELECT SUM( examine_detail.fee) INTO fee
29      FROM examine_detail
30      WHERE examine_detail.patient_code = input_patient_code;
31      DBMS_OUTPUT.PUT_LINE ('Treatment Fee: ' || fee);
32
33      SELECT SUM(examine_detail.fee + medication.price) INTO
34      total_fee
35      FROM examine_detail INNER JOIN medication
36      ON examine_detail.med_code = medication.mcode
37      WHERE examine_detail.patient_code = input_patient_code;
38      DBMS_OUTPUT.PUT_LINE ('Total Fee: ' || total_fee);
39
40
41      DBMS_OUTPUT.PUT_LINE ('List of examinations:');
42      DBMS_OUTPUT.PUT_LINE ('Examine Date' || '|Medication Code Used'
43      || '|Examination Fee' || '|Medication Price' );
44      FOR rec IN (
45          SELECT
46              examine_detail.patient_code,
47              examine_detail.fee AS examine_fee,
48              examine_detail.examine_date,
49              examine_detail.med_code,
50              medication.price,
51              medication.mname
52
53          FROM examine_detail INNER JOIN medication
54              ON examine_detail.med_code = medication.mcode
55          WHERE examine_detail.patient_code = input_patient_code
56      )
57      LOOP
58          DBMS_OUTPUT.PUT_LINE (' ' || rec.examine_date || ' | '
59          || rec.mname || ' | ' || rec.examine_fee || ' | ' ||
60          rec.price || ' | ');
61      END LOOP;
62
63      END;
END IF;
```

```
64 IF input_patient_code LIKE 'IP_-----' THEN
65     BEGIN
66
67         SELECT fee
68         INTO fee
69         FROM inpatient
70         WHERE pcode = input_patient_code;
71
72         DBMS_OUTPUT.PUT_LINE ('Treatment Fee: ' || fee);
73
74         SELECT SUM(medication.price)
75         INTO total_medication_price
76         FROM treat JOIN medication
77             ON treat.med_code = medication.mcode
78         WHERE
79             patient_code = input_patient_code;
80
81         DBMS_OUTPUT.PUT_LINE ('Total Medication price: ' ||
82             total_medication_price);
83
84         total_fee := fee + total_medication_price;
85         DBMS_OUTPUT.PUT_LINE ('Total Fee: ' || total_fee);
86
87         DBMS_OUTPUT.PUT_LINE ('List of medications:');
88         DBMS_OUTPUT.PUT_LINE (' Medication Code' || ' | ' || '
89             Medication Name and Price ');
90         FOR rec IN (
91             SELECT
92                 treat.med_code,
93                 medication.mname,
94                 medication.price
95             FROM treat JOIN medication
96                 ON treat.med_code = medication.mcode
97             WHERE
98                 patient_code = input_patient_code
99         )
100         LOOP
101             DBMS_OUTPUT.PUT_LINE ('          ' || rec.med_code || '          |
102                 ' || rec.mname || '          Price: ' || rec.price);
103         END LOOP;
104
105     END;
106 END IF;
107 RETURN total_fee;
108 END;
```

To run the function, we run the following statement:



```
1 DECLARE
2   a number;
3 BEGIN
4   a := total_fee('IP203462341');
5 END;
```

RESULT:

The output table for an Inpatient with ID of "IP203462341":

```
PAYMENT REPORT
Patient ID: IP203462341
Treatment Fee: 1529.99
Total Medication price: 1069.87
Total Fee: 2599.86
List of medications:
Medication Code | Medication Name and Price
N05CF03         | Zaleplon      Price: 449.89
L01CA06         | Vintafolide   Price: 399.99
A07AB04         | Succinylsulfathiazole Price: 219.99
```

Our data about medication and fee:

	⚡ MCODE	⚡ MNAME	⚡ PRICE
1	N02BE01	Paracetamol	199.99
2	N05CF03	Zaleplon	449.89
3	J01XA05	Oritavancin	99.89
4	L01CA06	Vintafolide	399.99
5	A07AB04	Succinylsulfathiazole	219.99
6	N02AD02	Phenazocine	119
7	A12CC03	Magnesium gluconate	79.99

The Fee the patient has to pay indeed is 1529.99.

⚡ PCODE	⚡ DATE_OF_ADMISSION	⚡ DIAGNOSIS	⚡ SICKROOM	⚡ FEE	⚡ FIRST_NAME	⚡ LAST_NAME	
1	IP205679239	06-OCT-19	Catastrophe	A202	1099.99	Claud	Riddle
2	IP945829231	08-AUG-20	Liver Cancer	A109	2092.89	Louis	Clark
3	IP205127452	26-SEP-21	Pneumonia	A391	3942.99	Cloe	Browning
4	IP203462341	03-OCT-21	Traumatic brain injury	B029	1529.99	Rosa	Lopez
5	IP104232231	03-OCT-21	Appendicitis	B823	9238.99	Kayla	Bernard

The output table for an Outpatient with ID of "OP000000003":

PAYMENT REPORT

Patient ID: OP000000003

Total Medication price: 379.87

Treatment Fee: 497

Total Fee: 876.87

List of examinations:

Examine Date	Medication Code Used	Examination Fee	Medication Price
03-OCT-21	Paracetamol	199	199.99
01-OCT-21	Oritavancin	99	99.89
05-OCT-21	Magnesium gluconate	199	79.99

2.2.4 Question d: Write a procedure to sort the doctor in increasing number of patients he/she takes care in a period of time

- In order to make a list of doctors which sorted in increasing number of patients he/she takes care in a period of time.
- **Firstly**, We have to join (by "INNER JOIN") EMPLOYEE, EXAMINE, EXAMINE_DETAIL and TREATMENT_DETAIL together, which EXAMINE, EXAMINE_DETAIL belong to outpatients and TREATMENT_DETAIL belong to inpatients. **Secondly**, limiting the START_DATE of TREATMENT_DETAIL and EXAMINE_DATE of EXAMINE_DETAIL in a period of time that satisfied the input Start date and End date by "WHERE".
- **The third step** is grouping them by the doctors code ("EMP_CODE") and sort by ascending order by two syntax "GROUP BY" (ASC by default) and "ORDER BY" respectively. **Lastly**, is to save the the result table including Doctor code, total number of outpatients and total number of inpatients in "rec" and display the list by a loop and "DBMS_OUTPUT.PUT_LINE()";

```
1 CREATE OR REPLACE PROCEDURE Doc_Sort1
2 (
3     start_date IN DATE,
4     end_date IN DATE
5 )
6 AS
7     EMP_ID EMPLOYEE.EMP_CODE%TYPE;
8     Total_OP INTEGER;
9     Total_IP INTEGER;
10 BEGIN
11     FOR rec IN (SELECT EMPLOYEE.EMP_CODE, COUNT(DISTINCT
12 EXAMINE_DETAIL.PATIENT_CODE) AS "ToTal Outpatients", COUNT(DISTINCT
13 TREATMENT_DETAIL.PATIENT_CODE) AS "ToTal Inpatients"
14 INTO EMP_ID, Total_OP, Total_IP
15 FROM EMPLOYEE
16 INNER JOIN EXAMINE ON EMPLOYEE.EMP_CODE = EXAMINE.DOCTOR_CODE
```



```
15  INNER JOIN EXAMINE_DETAIL ON EXAMINE.PATIENT_CODE =  
    EXAMINE_DETAIL.PATIENT_CODE  
16  INNER JOIN TREATMENT_DETAIL ON EMPLOYEE.EMP_CODE =  
    TREATMENT_DETAIL.DOCTOR_CODE  
17  WHERE EXAMINE_DETAIL.EXAMINE_DATE >= start_date AND  
    EXAMINE_DETAIL.EXAMINE_DATE <= end_date AND TREATMENT_DETAIL.START_DAY  
    >= start_date AND TREATMENT_DETAIL.START_DAY <= end_date  
18  GROUP BY EMPLOYEE.EMP_CODE  
19  ORDER BY "ToTal Outpatients", "ToTal Inpatients")  
20  LOOP  
21  DBMS_OUTPUT.PUT_LINE ('Doctor ID: ' || rec.EMP_CODE || '; ToTal  
    Outpatients: ' || rec."ToTal Outpatients" || '; ToTal Inpatients: ' ||  
    rec."ToTal Inpatients");  
22  END LOOP;  
23  END;
```

RESULT: To run the procedure, we run the following statement:

```
1  EXEC Doc_Sort1('14-07-2021', '25-12-2021');
```

This statement will print out the list of doctors who treat patients in a period between 14/07/2021 and 25/12/2021 with increasing order in the DBMS output:

```
Doctor ID: NEU3006443; ToTal Outpatients: 1; ToTal Inpatients: 3  
Doctor ID: NUD4009043; ToTal Outpatients: 1; ToTal Inpatients: 3  
Doctor ID: CDG2004667; ToTal Outpatients: 2; ToTal Inpatients: 3  
Doctor ID: SEH6003251; ToTal Outpatients: 2; ToTal Inpatients: 4  
Doctor ID: ETN1004556; ToTal Outpatients: 4; ToTal Inpatients: 5
```

2.3 Part 3: Building Applications

Build an application with the following requirements:

- Programming environment: Android App
- Programming language: Kotlin.
- The application connects to the database created in Part 1 and Part 2.
- Display the data on the form and perform the requirements below.
- Students need to prepare data and scripts for demonstration at the reporting session.

I. Create user (0.5 mark)

Log in to the database with DBA privileges such as SYS / SYSTEM, create a user named "Manager" and assign all access rights to this user.

Creating user named "Manager" with password "123456789":

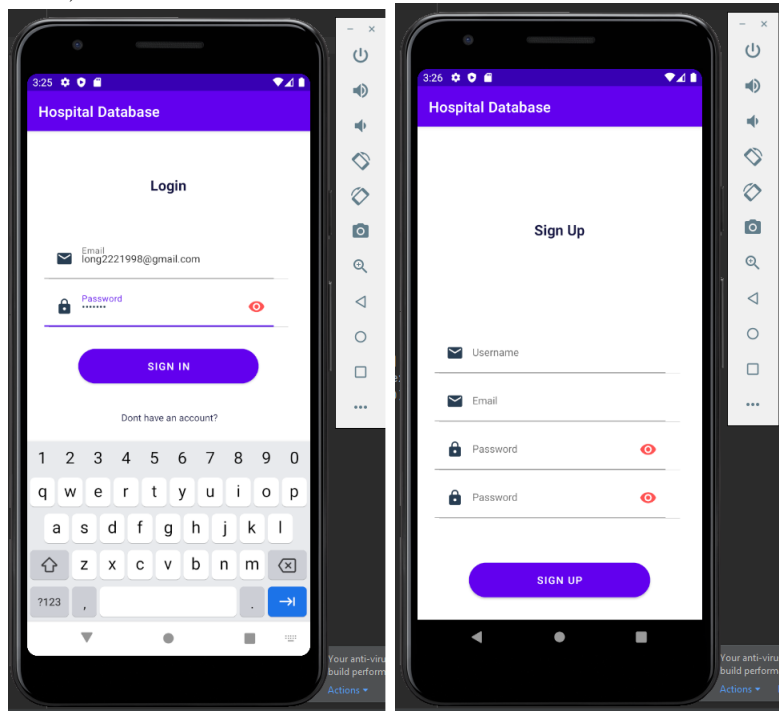
```
1  CREATE USER Manager IDENTIFIED BY 123456789;
```

Grant all access rights to this user:

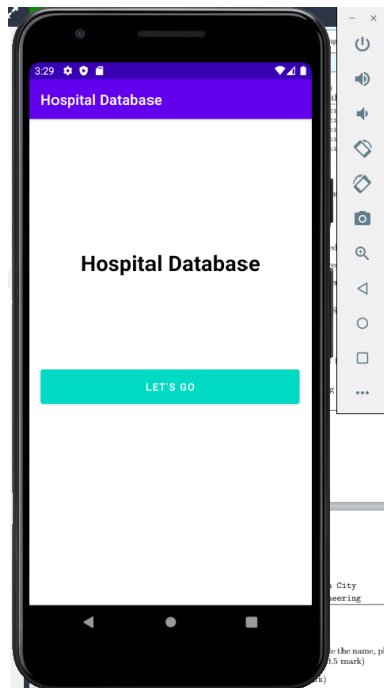
1 GRANT ALL PRIVILEGES TO Manager;

II. Requirement function (2.5 marks)

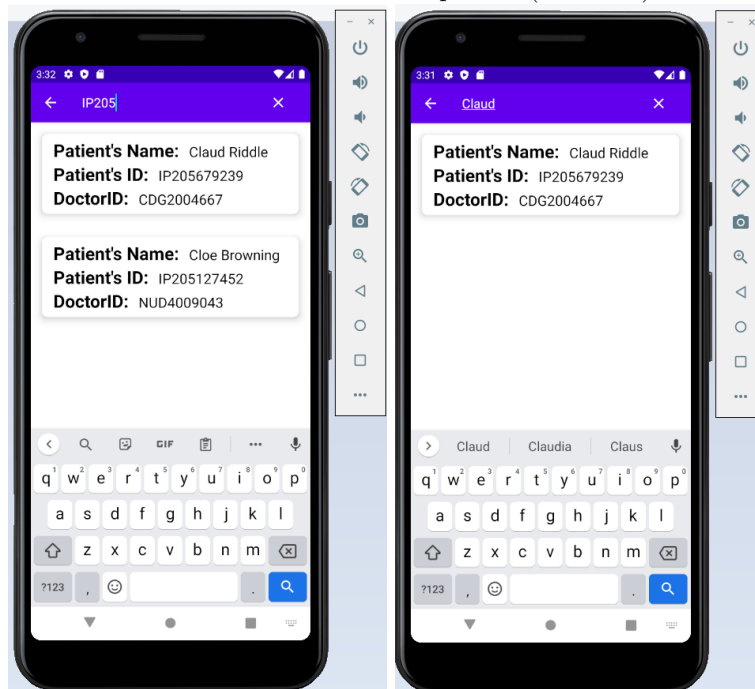
- Log in, log out (enter the user name/password for Manager account to log in/out). (0.5 mark)



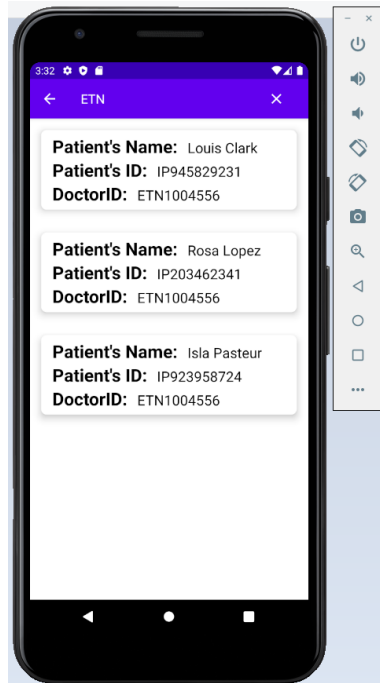
After authentication, the screen immediately move to the opening area that is ready for accessing the Database as below:



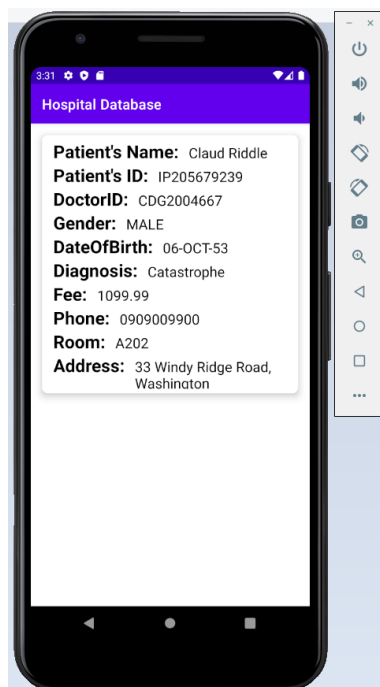
- Log in to the user manager and do the following (2 marks):
 - o Hospital Database
 1. Search patient information: Search results include the name, phone number and information about the treatment and visit of the patient.(0.5 mark)



2. Add information for a new patient. (0.5 mark)
3. List details of all patients which are treated by a doctor. (Doctor's ID instead) (0.5 mark)



4. Make a report that provides full information about the payment for each treatment or examination of a patient.





The application was implemented on Android Studio platform.

```
1 package com.example.ass2
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.os.Bundle
7 import android.text.TextWatcher
8 import android.view.Menu
9 import android.widget.EditText
10 import androidx.appcompat.widget.AppCompatEditText
11 import androidx.appcompat.widget.SearchView
12 import androidx.recyclerview.widget.LinearLayoutManager
13 import androidx.recyclerview.widget.RecyclerView
14 import com.google.android.gms.common.util.ArrayUtils.contains
15 import com.google.firebase.database.*
16 import kotlin.android.synthetic.main.activity_userlist.*
17 import java.util.*
18 import kotlin.collections.ArrayList
19
20 class UserListActivity : AppCompatActivity() {
21     private lateinit var dbRef : DatabaseReference
22     private lateinit var userRecyclerView: RecyclerView
23     private lateinit var userAdapter: MyAdapter
24     private lateinit var userArrayList : ArrayList<User>
25     private lateinit var userRecyclerView : RecyclerView
26
27     override fun onCreate(savedInstanceState: Bundle?) {
28         super.onCreate(savedInstanceState)
29         setContentView(R.layout.activity_userlist)
30
31         userArrayList = arrayListOf<User>()
32         userRecyclerView = arrayListOf<User>()
33
34         userRecyclerView = findViewById(R.id.userlist)
35         userRecyclerView.layoutManager = LinearLayoutManager(this)
36         userRecyclerView.setHasFixedSize(true)
37
38         userAdapter = MyAdapter( arrayListOf<User>(){user->showUserDetails(user)})
39         userRecyclerView.adapter=userAdapter
```