

Database Management Systems

RAMKHAMHAENG HOSPITAL

Presented by

sec 1 | Vipawan J. 6088044 | Krittamet K. 6088063
Pada K. 6088079 | Wipu K. I 6088095 | Thanirin T. 6088122



Project Phase 2: Business domain study and database design

Ramkhamhaeng Hospital

Team members

| | | |
|-----------|-------------------|---------|
| Vipawan | Jarukitpipat | 6088044 |
| Krittamet | Kiattikulwattanal | 6088063 |
| Pada | Kanchanapinpong | 6088079 |
| Wipu | Kumthong | 6088095 |
| Thanirin | Trironnarith | 6088122 |

Instructor

Asst. Prof. Dr. Songsri Tangsripairoj

ITCS241: Database Management Systems

2nd Semester

Table of Contents

| | |
|---|-----------|
| Company Overview | 1 |
| History | 1 |
| Mission | 2 |
| Services..... | 2 |
| Business functions | 3 |
| 1. Department Arrangement..... | 3 |
| 2. Patient Registration | 4 |
| 3. Visit Registration | 5 |
| 4. Doctor Treatment..... | 6 |
| 5. Medical Order Record and Order Verification..... | 7 |
| 6. Medicine Verification and Arrangement | 7 |
| 7. Finance Queue and Payment..... | 8 |
| Business Rule | 10 |
| Actual Forms..... | 13 |
| Relational Schema..... | 14 |
| Data Dictionary | 15 |
| Normalization..... | |
| CROW foot Diagram | |
| CHEN Diagram..... | |
| Project 2..... | 28 |
| Comments and Errors from Project Phase 1..... | 29 |
| Queries..... | 30 |

Project Assignment

Phase 1: Business domain study and database design



(Photo 1.0: Ramkhamhaeng Hospital Logo)

Business domain: Hospital

Company: Ramkhamhaeng Hospital

Company Overview

History

Ramkhamhaeng Hospital is one of the top hospitals that are trusted by many for its quality, medical specialists, medical personals who place importance in skills and expertise, and the effectiveness of advanced medical equipments, as well as specialized after-hour clinics which get continuous improvement by selecting experienced medical personals who are able to respond to the needs of medical treatment from the patients, moreover for providing services while keeping responsibility in mind.

Ramkhamhaeng Hospital is a private hospital with 485 hospital beds which was founded on the 28th of February, 1988. The hospital is located on Ramkhamhaeng Rd. near Huamark Indoor Stadium and has more than 32,000 square meters which includes 3 medical buildings and a parking able to hold more than 700 vehicles.

The hospital was listed on The Stock Exchange of Thailand and certified an international standard provided by JCI or Joint Commission International Standard Accreditation, USA, which is an international organization that guarantees the quality of the worldwide hospitals with only 200 hospitals accredited. The outstanding aspect of the medical team of Ramkhamhaeng Hospital is that there are medical specialists to provide for medical services in every sub specialty clinic.

Mission

“Ramkhamhaeng Hospital is a forefront hospital that receives confidence and trust from the people of Bangkok with Quality and efficiency of its modern medical equipment, as well as its special clinics, which have been created and developed in a continual manner, which are manned with selective specialized health personnel to provide for services which are full of responsibility awareness and able to meet health care service demand of the people of Bangkok without limitation.” (Ramkhamhaeng Hospital's mission).”

Services

Ramkhamhaeng Hospital's main services are providing medical treatment and medication. The treatment and medication cover health check-up, 4 special medical centers including back pain center, stroke center, heart center, diabetic foot care center, and various clinics namely Obstetrics and Gynecology Clinic, Internal Medicine Clinic, Orthopedic Center, Surgery Clinic, Ear, Nose and Throat Clinic, Brain and Nervous System of Ramkhamhaeng Hospital.

Other services Ramkhamhaeng Hospital provide are Patient Relocation Center 24 hours Emergency Services, Golf Cart Service, Taxi Service, Tabernacle, Muslim Patient Facility and Halal Food, Muslim Prayer Room, Food Center, Cafes and Bakeries, Convenience stores, Banks, and ATM services.

Business functions

Out Patient Department Service consists of several functions

This function is designed for providing the treatment services to outpatients, people with health problems who visit the hospital for diagnosis, but do not require to be admitted for overnight care. This department offer a broad range of treatment services, diagnosis tests and minor surgical procedures.

1. Department Arrangement: The hospital consists of three [1] building. Each building has several [2] departments, and each department has their own [3] room(s). [4] Employees are positioned in each department. Information including [5] name, [6] address and nationality of each employees is recorded.

Responsibility: Building, Departments, and Rooms, Employees

Main Activities:

- Administers of the hospital classify departments according to the scope of work and skills.
- Administers arrange departments in each building.
- Administers assign the room(s) for each department.
- Employees are assigned to particular departments.
- Human Resource Department records personal information of other employees.

Business Requirements:

- The hospital requires the blueprint and record of departments and rooms in each building.
- The hospital requires the lists which contain information of employees in each department.

2. Patient Registration: When the new [7] patients come to the hospital. They have to fill out the form for their personal information and give information about their symptoms in order for the PR to recommend the [8] clinic and doctors that they should visit. The re-visit patients show their [9] doctor appointment. They will get given the visit card for each particular clinic that they have to visit.

Responsibility: Patient and Employees

Main Activities:

- The new patients fill personal information in the form.
- The re-visit patients show their visit appointment to the public relation officers.
- The patients provide information about today visit.
- Public relation officers recommend the clinic and doctors that the patient should visit
- Public relation officers print out the “card” with particular visit number for the patient to contact the nurse counter at particular department.

Business Requirements:

- The hospital requires the system for input and keep the patient's personal information and medical records when they come to register.
- The public relation requires database for searching available doctors for particular departments.
- The hospital requires the system that connected between departments in order to send the patient's information to particular department.

3. Visit Registration: When the patients arrive at the department. They have to contact nurse counter, and the nurse will measure the vital signs including blood pressure, heart rates, temperature. They have to measure the height and weight. The nurse will ask the patient for medical information such as the symptoms that they have. The patient will then wait for the queue for the name to be called to visit the doctor.

Responsibility: Nurse and nurse head.

Main Activities:

- The nurse receives the “card” from the patient and check in the system which doctor has the least queue and assign the patient to that doctor.
- In case the patient request particular doctor, the nurse has to add the patient to that particular doctor waiting list. Some doctors only receive 200 queues a day.
- The nurse measures the vital sign of the patients and ask for more information such as the basic symptoms.

Business Requirements:

- The nurse requires the program for searching doctor and assign the patient to particular doctor waiting list and arranging the queue of the patients.
- The nurse requires program for recording vital signs of the patients and generate the basic report for analyzing this information. Also, this program requires to be able to record the basic symptoms that the patient tell the nurse.

4. Doctor Treatment: The doctor will read the medical history of that particular patient to understand the information and further use this information for the diagnosis. The doctor might ask for more particular information from the patients. The doctor writes [10] diagnosis result for the patients and they might [11] order more test, lab or x-ray. Finally, the doctor can order the medicine and make the next appointment with the patients. These orders are given through hospital network to the nurse counter so that the nurse is able to know where the patients have to go next and to provide the right service to them. If the patients have to do to other department for the lab, they have to come back to this doctor room in order to listen to further diagnosis.

Responsibility: Doctors

Main Activities:

- The doctors have to read the patient's medical information and condition.
- The doctors have to record the diagnosis in the patient's file.
- The doctors can order the test/ lab or x-ray for the patients who need additional care.
- The doctors can order medicine for the patients and make the next appointment.

Business Requirements:

- The doctors require the program to access the patient's information and previous medical record and add the record of the current diagnosis.
- The doctors require the program that can order lab, test, or x-rays and this program is connected to other department in order to automatically send the order and receive the result.
- The doctor requires the program that can access database of available medicine in order to order the medicines and send this order to the nurse counter and pharmacist.
- The doctor requires the program that can make the appointment for the patient.

5. Medical Order Record and Order Verification: After the order has been made by the doctor, the system will record each order as the medical order entry. Each patient visit will have one medical order which consists of several medical order entries. The medical order entry can be either medical equipment, lab order or medicine. Finally, support will review all the medical order entries for the financial process.

Responsibility: Support

Main Activities:

- The supports verify the medical order entries for each patient's medical order.

Business Requirements:

- The hospital needs the system for record each patient's medical order and verify the medical order entries in the order.

6. Medicine Verification and Arrangement: Once the support (pharmacists) receive the order in the computer, they have to verify whether the medicines are conflicted with the patient's' medical information or not. If there is the conflict, the list of medicines is allowed to be changed by pharmacists without informing the doctor. After the medicines are arranged, the queue is submitted to the system and waited to be call after the patients pay the bill.

Responsibility: Support (Pharmacist)

Main Activities:

- The pharmacists have to verify the medical list that received from the nurse and doctor with the patient medical record and check whether there is any conflict.
- The pharmacists have the right to change the medicine in case there is a conflict.
- The pharmacists have to submit the list for arranging the medicine.
- The pharmacists have to arrange the medicine according to the list.

Business Requirements:

- The pharmacist requires the program for verifying the medical lists, assigning the usage of the medicine to particular patient, and changing the medical list in case it has conflict with the patient's medical conditions.
- The pharmacist requires the program for viewing the approved medical list.
- The hospital requires the medical arrangement system.

7. Finance Queue and Payment: After the patients scan their visit card at the kiosk, the patient will get one financial queue, and one medical bill. However, they can separate the bill into several receipts which consist of particular receipt entries according to their payment method.

Responsibility: Support (Financial Officers)

Main Activities:

- The patient scans the “card” at the kiosk and wait for the queue to be called.
- The finance officers receive the queue from the kiosk system.
- The finance officers call the queue to the particular counter and ask the patients for the method that they want to pay.
- The finance officers print out the bill according to the patients’ need.
- The finance officers receive the payment and submit the queue to the pharmacist room.
- The pharmacists submit the queue the system to call the patient.
- The pharmacists provide information about the usage of the medicine to the patients.

Business Requirements:

- The hospitals require the kiosk system that received the patient's number by scanning and sending the queue to the system.
- The hospitals require the kiosk system linked with the patient information database. The system requires the scan of the patient's "card" and submit the queue to the system.
- The finance department requires the system to arrange the queue and print out the bill for each patient.
- The finance and pharmacy department require the queue arranging and calling system.

Business Rule

1. A building has many departments in it, but a department only resides in a building.
2. A department has many rooms in it, but a room can stay in only one department.
3. A department has many employees, but an employee works in only one department.
4. A clinic has many nurses, but a nurse can only work in one clinic.
5. A clinic has many doctors, but a doctor can only work in one clinic.
6. A doctor can have many doctor appointments or none at all, but a doctor appointment must be appointed to only one and at least to one doctor.
7. A doctor can write many diagnosis results or none at all, but a diagnosis result can be written by only and at least one doctor.
8. A doctor can order many to none lab orders, but a lab order must be ordered by and at least one doctor.
9. A doctor issues many to none medical order entries, but a medical order entry must have only and at least one doctor.
10. A Province has at least one to many district lists, and district lists are contained in at least and at most one province list.
11. A Subdistrict list has at least one and only one postal code, and postal code at least one to many subdistrict list.
12. The district list has at least one to many subdistrict list, and subdistrict lists are contained at least and at most one district list
13. An employee has at least and at most one street list, and street list is specified to many to none employee.
14. An employee has at least and at most one subdistrict list, and subdistrict list is specified to many to none employee.
15. An employee has at least and at most one nationality, but a nationality list can have many employees.
16. A lab order contains information about at least and at most one room. Room information can be in many to none lab orders.
17. A lab order contains at least and at most one lab result and a lab result is in at least and at most one lab order.

18. A medical bill can generate many but at least one receipts. A receipt is in at least and at most one medical bill.
19. A medical order has many but at least one medical order entries. Unique medical order entry is in at least and at most one medical order.
20. A medical order entry contains one or none lab order. A lab order is in at least and at most one medical order entry.
21. A medical order entry contains one or none medicine, but medicine is in at least and at most one medical order entry.
22. A nurse can record many to none vital sign records, but a vital sign record is recorded by at least and at most one nurse.
23. A patient can have many to none doctor appointments, a doctor appointment is assigned to at least and at most one patient.
24. A doctor appointment has at least one to none visit, and a visit has at least one and at most one doctor appointment.
25. A patient can have many to none lab order entry, but a lab order entry is assigned to at least and at most one patient.
26. A lab order has at least one to many lab order entries, and lab order entries are contained in one lab order.
27. A lab order entry has at least and at most one fee entry, and a fee entry has at least and at most one lab order entry
28. A lab order entry has
29. A patient has at least and at most one street list, and street list is specified to many to none patient.
30. A patient has at least and at most one subdistrict list, and subdistrict list is specified to many to none patient.
31. A patient has at least and at most one nationality list, and nationality list has many to none patient
32. A receipt has at least one to many receipt entries. A receipt entry is in at least and at most one receipt.

33. A support writes many to none lab results. A lab result is written by at least and at most one support.
34. A support reviews many to none medical order entries, but a medical order entry is reviewed by at least and at most one support.
35. A visit contains at least one to many vital sign records. A visit sign record is in at least and at most one visit.
36. A visit has at least one and only one diagnosis result, but a diagnosis result is in at least and at most one visit.
37. An ICD10 identifies at least and at most one diagnosis result, and a diagnosis result is identified by at least and at most one ICD10
38. A visit has at least and at most one medical order and a medical order is in at least and at most one visit.
39. A Visit has at least and at most financial queues, and a financial queue is assigned to at least and at most one visit.
40. A visit has at least and at most one medical bill, and a medical bill is assigned to at least and at most one visit.
41. Medicine has at least and at most one medicine type, but medicine type contains at least one to many medicines.
42. Medical order entry has at least and at most one dosage list, but dosage list contains at least one to many medical order entries.

Actual Forms

คันหารายชื่อแพทย์
ระบบคันหารายชื่อแพทย์มีกั้งระบบคันหา
ด้วยความเชี่ยวชาญเฉพาะด้านและค้นหาด้วยแพทย์และช่วงเวลาของแพทย์

คันหาแบบรวดเร็ว

คันหาแบบระบุเงื่อนไข

กรุณเลือกสาขาที่เชี่ยวชาญเพื่อแสดงรายชื่อแพทย์ตามสาขาที่เชี่ยวชาญนั้นๆ หรือในกรณีที่ท่านทราบเชิงแพทย์ ท่านสามารถเลือกแพทย์ได้จากรายชื่อด้านล่าง

| สาขาที่เชี่ยวชาญ | รายชื่อแพทย์ | คันหาแพทย์ | บันทึกใหม่ |
|-----------------------------|-------------------------|------------|------------|
| กฎหมายเลือกสาขาที่เชี่ยวชาญ | กฎหมายเลือกรายชื่อแพทย์ | | |

▶ พญ. กนกวรรณ ยุติธรรม

นักพยาบาล

ชื่อ : กนกวรรณ ยุติธรรม
สาขาแพทย์ : จักษุแพทย์ (ต้อหิน)
ภาษา : Thai, English
การศึกษา : คณะแพทยศาสตร์ศิริราชพยาบาล มหาวิทยาลัยมหิดล (2544)
วุฒิปัจจุบัน : จักษุแพทย์ ราชวิถีอยάลลักษณแพทย์แห่งประเทศไทย (2550)
จักษุแพทย์เฉพาะทางต้อหิน ราชวิถีอยάลลักษณแพทย์แห่งประเทศไทย (2551)
การศึกษาหลังปริญญา : Glaucoma Fellowship, University of California San Francisco (UCSF) ปี 2008-2009
แขนงที่สนใจเป็นพิเศษ : การผ่าตัดต้อหินและต้อหิน การใช้เลเซอร์จักษุโรคต้อหิน[†] โรคต้อหินอยุติธรรม

| วัน | เวลา | สถานที่ |
|--------|---------------|-------------------------|
| จันทร์ | 08:00 - 20:00 | อาคาร 3 ชั้น 2 (แผนกตา) |
| พุธ | 13:00 - 20:00 | อาคาร 3 ชั้น 2 (แผนกตา) |

(Photo 2.0: Form of Doctor)

ข้อมูลผู้ป่วย

ชื่อเมส * :

ชื่อเมส * :

ชื่อ * :

นามสกุล * :

วันที่เกิด * :

เพศ * : ♂ชาย ♀หญิง

ประทศ * : Please Select Country

เคยเป็นผู้ป่วยของโรงพยาบาล : ใช่ ไม่ใช่

เบอร์ติดต่อกันลับ * :

เงื่อนไขการใช้งาน

ข้าพเจ้าขอสงวนสิทธิ์ที่จะไม่ได้รับข้อมูล

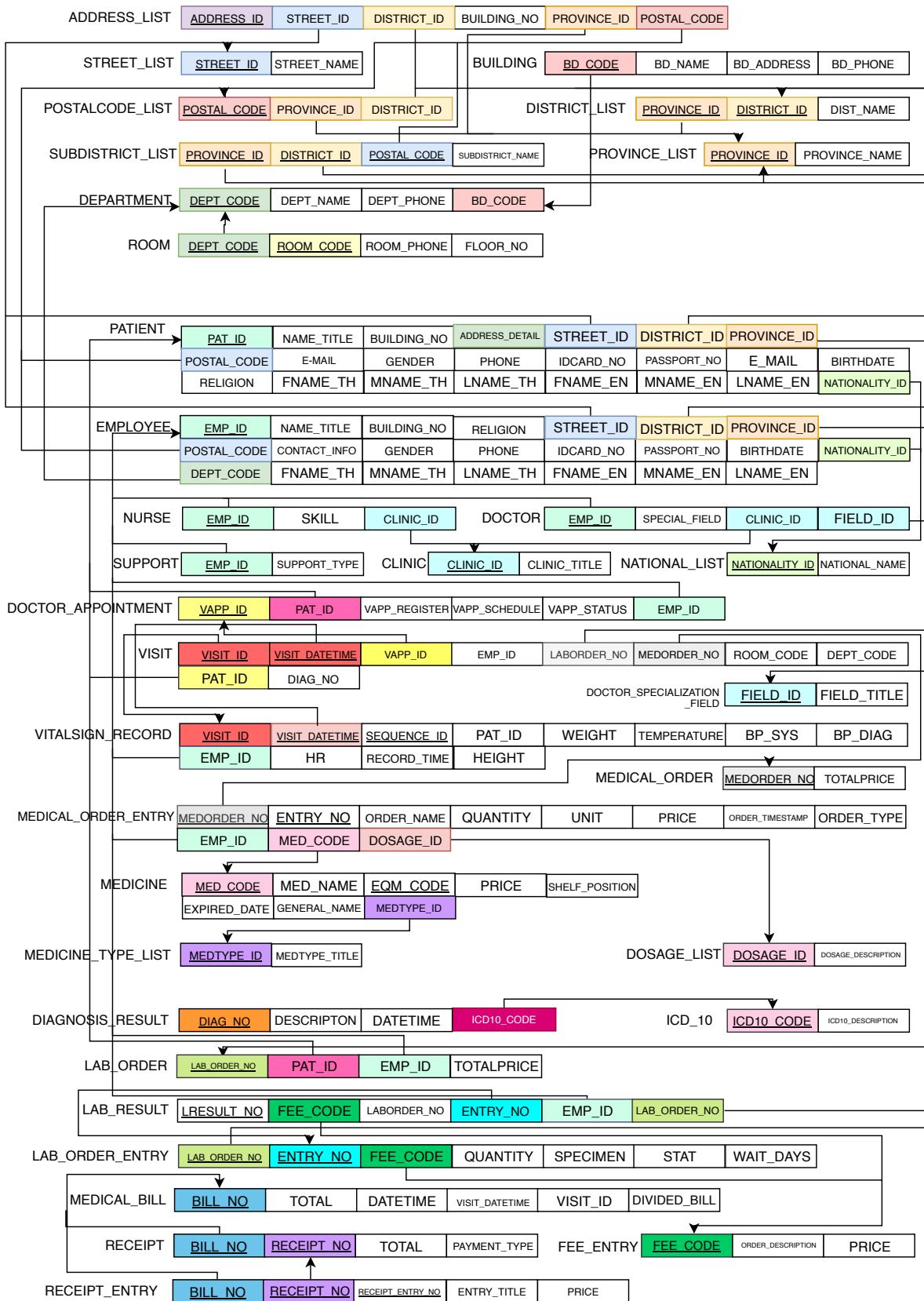
การนัดหมายผ่านทางเว็บไซต์นี้เป็นเพียงการแจ้งความจำนงการนัดหมาย ล่วงหน้าเท่านั้น ยังไม่ใช้การยืนยันวัน เวลา สถานที่ และแพทย์ที่รับนัดทันที เจ้าหน้าที่โรงพยาบาลฯ จะติดต่อกลับเพื่อยืนยันรายละเอียดเพิ่มเติมและทำการยืนยันการนัดหมายอีกครั้ง การนัดหมายและการยืนยันนี้ อาจเปลี่ยนแปลงได้ จนกว่ายันต์หมายหรือผู้ป่วยจะได้ทำการลงทะเบียนคนไข้ อย่างเป็นทางการ ณ โรงพยาบาลฯ

ยอมรับเงื่อนไขและทำรายการนัดหมาย

เวลาในการตอบกลับ: โรงพยาบาลaramคำแนะนำจะตอบรับ การนัดหมายภายใน 72 ชั่วโมง

(Photo 2.1: Form of Patient)

Relational Schema



Data Dictionary

| TABLE | FIELD NAME | DATA TYPE | DATA FORMAT | FIELD SIZE | PK/FK | DESCRIPTION |
|------------|------------|-----------|--------------|------------|-------|---|
| BUILDING | BD_CODE | INT | XXXXXX | 6 | PK | unique identification number for each building |
| | BD_NAME | TEXT | | 20 | | the name of the building |
| | BD_ADDRESS | TEXT | | 20 | FK | the address of the building |
| | BD_PHONE | TEXT | 0XX-XXX-XXXX | 10 | | the hotline phone number of the building |
| DEPARTMENT | DEPT_CODE | INT | XXXXXX | 6 | PK | unique identification number for each department |
| | DEPT_NAME | TEXT | | 20 | | the name of the department |
| | DEPT_PHONE | TEXT | 0XX-XXX-XXXX | 10 | | the hotline phone number of the department |
| | BD_CODE | INT | XXXXXX | 6 | FK | the unique identification number of the building that the department resides in |
| ROOM | ROOM_CODE | INT | | 10 | PK | unique identification number for each room |
| | ROOM_PHONE | TEXT | XXXXX | 5 | PK/FK | the hotline phone number of the room |
| | DEPT_CODE | INT | XXXXXX | 6 | FK | the unique identification number of the department that the room resides in |
| | FLOOR_NO | INT | XX | 2 | | the floor number the room is located on |

| | | | | | | |
|----------|-------------|------|--------------------|----|----|---|
| CLINIC | CLINIC_ID | INT | XXXXXX | 6 | PK | unique identification number for each clinic |
| | TITLE | TEXT | | 20 | | the name of the clinic |
| EMPLOYEE | EMP_ID | INT | XXXXXX | 6 | PK | unique identification number for each employee |
| | NAME_TITLE | INT | | 20 | | the name title of each employee |
| | FNAME_TH | TEXT | | 50 | | the first name in Thai |
| | MNAME_TH | TEXT | | 50 | | the middle name in Thai |
| | LNAME_TH | TEXT | | 50 | | the last name in Thai |
| | FNAME_EN | TEXT | | 50 | | the first name in English |
| | MNAME_EN | TEXT | | 50 | | the middle name in English |
| | LNAME_EN | TEXT | | 50 | | the last name in English |
| | ADDRESS_ID | INT | XXXXXX | 6 | FK | foreign key, the address of the employee |
| | DEPT_CODE | INT | XXXXXX | 6 | FK | foreign key, the department that the employee works in |
| | GENDER | CHAR | M or F | 1 | | the gender of the employee |
| | PHONE | TEXT | 0XX-XXX-XXXX | 10 | | phone number which can be dial to the employee |
| | ID_CARD_NO | TEXT | XXXXXXXXXXXXX X | 13 | | identification card number that identify employee information |
| | PASSPORT_NO | TEXT | XXXXXXX | 8 | | passport number to identify foreign employee |
| | EMP_TYPE | CHAR | | 20 | | type or technical skill of employee |

| | | | | | | |
|-----------|----------------|----------|---------------------|----|----|---|
| | NATIONALITY_ID | INT | XXXXXX | 6 | FK | unique identification number of each nation |
| | BIRTHDATE | DATE | DD/MM/YYYY:HH:MM:SS | 14 | | birthdate of each employee |
| | RELIGION | TEXT | | 20 | | religion of the employee |
| LAB_ORDER | LAB_ORDER_NO | INT | XXXXXX | 6 | PK | unique identification number for each lab order |
| | PAT_ID | INT | XXXXXX | 6 | FK | unique identification number for each patient |
| | EMP_ID | INT | XXXXXX | 6 | FK | unique identification number for each employee |
| | ORDER_DATETIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | datetime which the order is created |
| | ROOM_CODE | INT | XXXXXX | 6 | FK | unique identification number for each room |
| | TOTAL_PRICE | INT | XXXXXX | 6 | | price of lab operations |
| PATIENT | PAT_ID | INT | XXXXXX | 6 | PK | unique identification number for each patient |
| | NAME_TITLE | INT | | 20 | | title of the patient name |
| | FNAME_TH | TEXT | | 50 | | the first name in Thai |
| | MNAME_TH | TEXT | | 50 | | the middle name in Thai |
| | LNAME_TH | TEXT | | 50 | | the last name in Thai |
| | FNAME_EN | TEXT | | 50 | | the first name in English |
| | MNAME_EN | TEXT | | 50 | | the middle name in English |
| | LNAME_EN | TEXT | | 50 | | the last name in English |

| | | | | | | |
|-------------|----------------|------|-------------------------|----|----|--|
| | ADDRESS_ID | INT | | 20 | FK | unique identification number of the address for each patient |
| | GENDER | CHAR | M/F | 1 | | the gender of the patient |
| | E-MAIL | TEXT | | 20 | | electronic mail of the patient |
| | PHONE | TEXT | 0XX-XXX-XXXX | 10 | | phone number which can be dial to the patient |
| | ID_CARD_NO | TEXT | XXXXXXXXXXXXX X | 13 | | identification card number that identify patient information |
| | PASSPORT_NO | TEXT | XXXXXXX | 8 | | passport number to identify foreign patient |
| | NATIONALITY_ID | INT | | 20 | FK | unique identification number of each nation |
| | BIRTHDATE | DATE | DD/MM/YYYY:HH: MM:SS | 14 | | birthdate of each patient |
| | RELIGION | TEXT | | 20 | | religion of the employee |
| ADRESS_LIST | ADDRESS_ID | INT | XXXXXX | 6 | PK | unique identification number of the address |
| | BUILDING_NO | INT | XXXXXX | 6 | | number of building located |
| | STREET_ID | INT | XXXXXX | 6 | FK | street identification number where this address are located |
| | DIST_ID | INT | XXXXXX | 6 | FK | district identification number where this place located |
| | PROVINCE_ID | INT | XXXXXX | 6 | FK | province identification number of the place located |

| | | | | | | |
|-----------------|--------------------|----------|---------------------|----|-------|--|
| | POSTALCODE | INT | XXXXXX | 6 | FK | postacode for sending mail of the place |
| DOCTOR | EMP_ID | INT | XXXXXX | 6 | PK/FK | unique identification number for each employee |
| | SPECIAL_FIELD | TEXT | | 20 | | field of study of doctor |
| NURSE | EMP_ID | INT | XXXXXX | 6 | PK/FK | unique identification number for each employee |
| | SKILL | TEXT | | 20 | | technical skill of nurse |
| SUPPORT | EMP_ID | INT | XXXXXX | 6 | PK/FK | unique identification number for each employee |
| | SUPPORT_TYPE | CHAR | | 20 | | type or technical skill of the support |
| FINANCIAL QUEUE | QUEUE_NO | INT | XXXXXX | 6 | PK/FK | number or identifacation of this queue |
| | QUEUE_STATUS | TEXT | | 20 | | status of the queue when the queue is created |
| | PHARMACY_QUEUEIDX | INT | XXXXXX | 6 | | identification queue that are queueing to pharmacy visit |
| | FINANCIAL_QUEUEIDX | INT | XXXXXX | 6 | | identification queue that are queueing to financial part |
| | CASHIER_NO | INT | XXXXXX | 6 | | number of cashier to visit |
| | REGISTER_TIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | time registering into the queue |
| | FINANCIAL_CALLTIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | timestamp when the patient is called to queue |
| | FINANCIAL_ENDTIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | time when finishing using the financial service |

| | | | | | | |
|---------------|-------------------|----------|---------------------|----|-------|--|
| | PHARMACY_CALLTIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | calling time or starting using of the pharmacy visit |
| | PHARMACY_ENDTIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | time when finish visiting the pharmacy |
| RECEIPT | BILL_NO | INT | XXXXXX | 6 | PK/FK | unique identification number for each bill |
| | RECEIPT_NO | INT | XXXXXX | 6 | PK | unique identification number for the receipt |
| | TOTAL | DOUBLE | XXXXXX | 6 | | total amount of paying in the bill |
| | PAYMENT_TYPE | TEXT | | 20 | | payment method that paid for the service |
| RECEIPT_ENTRY | RECEIPT_ENTRY_NO | INT | XXXXXX | 6 | PK | unique identification number of each receipt entry |
| | BILL_NO | INT | XXXXXX | 6 | PK/FK | unique identification number for each bill |
| | RECEIPT_NO | INT | XXXXXX | 6 | PK/FK | unique identification number for the receipt |
| | RECEIPT_PRICE | DOUBLE | X,XXX,XXX,XXX | 10 | | the total price of the receipt |
| BILL_ENTRY | BILL_NO | INT | XXXXXX | 6 | PK/FK | unique identification number for each bill |
| | TITLE | TEXT | | 20 | | title of the bill |
| | CASHIER_NO | INT | XXXXXX | 6 | | number of the cashier to use |
| | PRICE | DOUBLE | | 20 | | price of the service |

| | | | | | | |
|--------------|----------------|----------|---------------------|----|-------|--|
| | DISCOUNT | DOUBLE | | 20 | | discount of the service |
| MEDICAL_BILL | BILL_NO | INT | XXXXXX | 6 | PK | unique identification number for each bill |
| | TOTAL | DOUBLE | XXXXXX | 6 | | total amount of paying in the bill |
| | DATETIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | datetime of medicalbill created |
| | VISIT_ID | INT | XXXXXX | 6 | FK | unique identification number of visit |
| | VISIT_DATETIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | FK | timestamp when visit |
| | DIVIDED_BILL | INT | XX | 2 | | the number of divided bills |
| VISIT | VAPP_NO | INT | XXXXXX | 6 | PK/FK | unique identification number of visit appointment number |
| | DIAG_NO | INT | XXXXXX | 6 | FK | unique identification number of diagnostic result for each visit |
| | LRESLT_NO | INT | XXXXXX | 6 | | unique identification number for the lab result |
| | VAPP_PRESENT | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | timestamp when the patient gets the visit |
| | VAPP_FINISH | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | timestamp when the patient is done |
| | BILL_NO | INT | XXXXXX | 6 | FK | unique identification number for each bill |

| | | | | | | |
|-------------------------|--------------|----------|---------------------|----|-------|--|
| MEDICAL_ORDER | MED_ORDER_NO | INT | XXXXXX | 6 | PK | unique identification number for the medical order |
| | TOTAL | DOUBLE | XXXXXX | 6 | | total amount of paying in the bill |
| DIAGNOSIS_RESU LT | DIAG_NO | INT | XXXXXX | 6 | PK | unique identification of each diagnosis result |
| | DATETIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | timestamp of diagnosis |
| | DESCRIPTION | CHAR | | 40 | | description of the result |
| | ICD10_CODE | INT | XXXXXX | 6 | FK | unique identification number of icd 10 |
| LAB_RESULT | LRESLT_NO | INT | XXXXXX | 6 | PK | unique identification number for the lab result |
| | LAB_ORDER_NO | INT | XXXXXX | 20 | FK | unique identification number for the order |
| | ENTRY_NO | INT | XXXXXX | 6 | FK | unique identification number for the entry no |
| | FEE_CODE | INT | XXXXXX | 6 | FK | unique identification number of fee entry |
| | EMP_ID | INT | XXXXXX | 6 | FK | unique identification number for each employee |
| MEDICAL_ORDER _ENTRY | MED_ORDER_NO | INT | XXXXXX | 6 | PK/FK | unique identification number for the medical order |
| | ENTRY_NO | INT | XXXXXX | 6 | PK/FK | unique identification number for the entry no |
| | ORDER_NAME | TEXT | | 20 | | the name of the order |

| | | | | | | |
|----------|----------------|----------|---------------------|----|----|--|
| | ORDER_DATETIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | time that the order have been created |
| | PRICE | DOUBLE | XXXXXX | 6 | | the price of the medicine |
| | QUANTITY | INT | XXXXXX | 6 | | quantity of the medicine |
| | UNIT | TEXT | | 5 | | the unit word for each entry, e.g. piece, baht |
| | EMP_ID | INT | XXXXXX | 6 | FK | unique identification number for each employee |
| | USAGE | TEXT | | 20 | | usage of the medicine |
| | MED_CODE | INT | XXXXXX | 6 | FK | unique identification number of medicine prescribed in medical order entry |
| | LAB_ORDER_NO | INT | XXXXXX | 6 | | unique identification number of lab order in each medical order entry |
| MEDICINE | MED_CODE | INT | XXXXXX | 6 | PK | unique identification number of each medicine |
| | NAME | TEXT | | 20 | | name of the medicine |
| | TYPE | TEXT | | 20 | | type of the medicine |
| | PRICE | DOUBLE | XXXXXX | 6 | | price of the medicine |
| | SHELF_POSITION | TEXT | | 20 | | the position and the shelf of the medicine |
| | EXPIRED_DATE | DATE | YYYY-MM-DD | 8 | | expired date of the medicine |

| | | | | | | |
|-----------------------|----------------|----------|---------------------|----|-------|--|
| | EQPM_CODE | INT | XXXXXX | 6 | PK | unique code of each equipment |
| | GENERAL_NAME | TEXT | | 20 | | general name to call the equipment |
| VITAL_SIGN_REC ORD | VAPP_ID | INT | XXXXXX | 6 | PK/FK | unique identification number of visit appointment in vital sign record |
| | SEQUENCE_ID | INT | XXXXXX | 6 | PK | unique identification number of each vital sign record |
| | VISIT_DATETIME | DATETIME | DD/MM/YYYY:HH:MM:SS | 14 | | timestamp when the record is created |
| | HEIGHT | DOUBLE | XXX | 3 | | height of the patient in the record |
| | WEIGHT | DOUBLE | XXX | 3 | | weight of the patient in the record |
| | TEMPERATURE | DOUBLE | XXX | 3 | | temperature of the patient in the record |
| | BP_SYS | DOUBLE | XXX | 3 | | BLOOD PRESSURE WHEN THE HEART CONTRACT |
| | BP_DIAG | DOUBLE | XXX | 3 | | BLOOD PRESSURE WHEN THE HEART RETRACT |
| | HR | DOUBLE | XXX | 3 | | Heart rate |
| | DEPT_CODE | INT | XXXXXX | 6 | | unique identification number for each department |

| | | | | | | |
|------------------|----------------|------|--------|----|-------|--|
| | EMP_ID | INT | XXXXXX | 6 | FK | unique identification number for each employee |
| NATIONALITY_LIST | NATIONALITY_ID | INT | XXXXXX | 6 | PK | unique identification number of each nation |
| | NATION_NAME | TEXT | | 20 | | real nation name of this nationality |
| DISTRICT_LIST | DIST_ID | INT | XXXXXX | 6 | PK | district identification number where the place located |
| | DIST_NAME | TEXT | | 20 | | real name of the district |
| | PROVINCE_ID | INT | XXXXXX | 6 | PK/FK | province identification number of the place located |
| PROVINCE_LIST | PROVINCE_ID | INT | XXXXXX | 6 | PK | province identification number of the place located |
| | PROVINCE_NAME | TEXT | | 20 | | real name of the province |
| POSTALCODE_LIST | POSTALCODE | INT | XXXXXX | 6 | PK | postacode for sending mail of the place |
| | DIST_ID | INT | XXXXXX | 6 | FK | district identification number where the place located |
| | PROVINCE_ID | INT | XXXXXX | 6 | FK | province identification number of the place located |
| STREET_LIST | STREET_ID | INT | XXXXXX | 6 | PK | street identification number where the place are located |
| | STREET_NAME | TEXT | | 20 | | real name of the street |

| | | | | | | |
|--------------------|--------------------|----------|------------------------|----|----|---|
| DOCTOR_APPOINTMENT | VAPP_ID | INT | XXXXXX | 6 | PK | unique identification number of visit appointment for each doctor appointment |
| | PAT_ID | INT | XXXXXX | 6 | FK | unique identification number of each patient assigned to the doctor appointment |
| | VAPP_REGISTER | DATETIME | DD/MM/YYYY HH:MM:SS | 14 | | time of register to a doctor appointment |
| | VAPP_SCHEDULE | DATETIME | YYYY-MM-DD HH:MM:SS | 14 | | Sechdued time to meet a doctor |
| | VAPP_STATUS | CHAR | XXXXXX | 6 | | status if patient is present not showing or cancel |
| | EMP_ID | INT | XXXXXX | 6 | FK | unique identification number for each employee |
| MEDICINE_TYPE_LIST | MEDTYPE_ID | INT | XXXXXX | 6 | PK | unique identification number of medicine type |
| | MEDTYPE_TITLE | TEXT | | 20 | | the title of the each list |
| DOSAGE_LIST | DOSAGE_ID | INT | XXXXXX | 6 | PK | unique identification number of dosage list |
| | DOSAGE_DESCRIPTION | CHAR | | 40 | | description of the list |
| ICD_10 | ICD10_CODE | INT | XXXXXX | 6 | PK | unique identification number of icd 10 |
| | ICD10_DESCRIPTION | CHAR | | 40 | | description of the icd |

| | | | | | | |
|-----------------|-------------------|--------|--------|----|-------|---|
| FEE_ENTRY | FEE_CODE | INT | XXXXXX | 6 | PK | unique identification number of fee entry |
| | PRICE | DOUBLE | XXXXXX | 6 | | price of the fee entry |
| | ORDER_DESCRIPTION | CHAR | | 40 | | description of the fee entry |
| LAB_ORDER_ENTRY | LAB_ORDER_NO | INT | XXXXXX | 6 | PK/FK | unique identification number for each lab order |
| | ENTRY_NO | INT | XXXXXX | 6 | FK | unique identification number for the entry no |
| | FEE_CODE | INT | XXXXXX | 6 | FK | unique identification number of fee entry |
| | QUANTITY | INT | XXXXXX | 6 | | quantity of the lab order entry |
| | SPECIMEN | CHAR | | 20 | | specimen of lab order entry |
| | STAT | INT | XXXXXX | 6 | | statistic of lab order entry |
| | WAIT_DAYS | INT | XXXXXX | 6 | | days waiting of lab order entry |



3NF: STREET_LIST (STREET_ID , STREET_NAME)

PK: STREET_ID

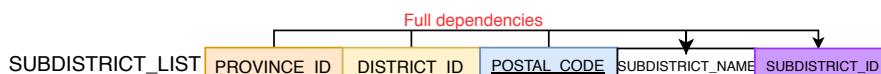
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: DISTRICT_LIST (PROVINCE_ID , DISTRICT_ID , DIST_NAME)

PK: DISTRICT_ID, PROVINCE_ID FK: PROVINCE_ID

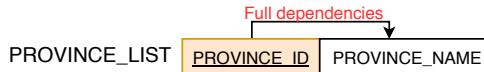
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: SUBDISTRICT_LIST (PROVINCE_ID , DISTRICT_ID , DIST_NAME , POSTAL_CODE , SUBDISTRICT_ID)

PK: POSTAL_CODE, DISTRICT_ID, PROVINCE_ID FK: DISTRICT_ID, PROVINCE_ID

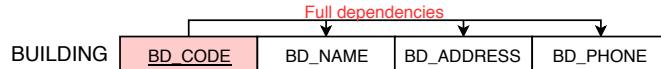
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: PROVINCE_LIST (PROVINCE_ID , PROVINCE_NAME)

PK: PROVINCE_ID

It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: BUILDING (BD_CODE , BD_NAME , BD_ADDRESS , BD_PHONE)

PK: BD_CODE FK: BD_ADDRESS

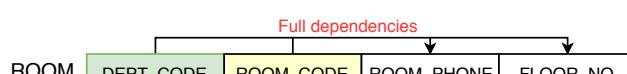
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: DEPARTMENT (DEPT_CODE , DEPT_NAME , BD_ADDRESS , BD_PHONE)

PK: DEPT_CODE FK: BD_CODE

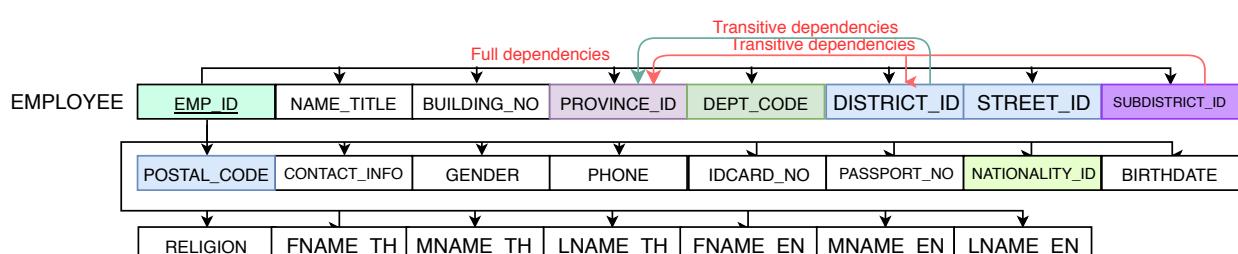
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: ROOM (DEPT_CODE , ROOM_CODE , ROOM_PHONE , FLOOR_NO)

PK: DEPT_CODE, ROOM_CODE FK: DEPT_CODE, ROOM_PHONE

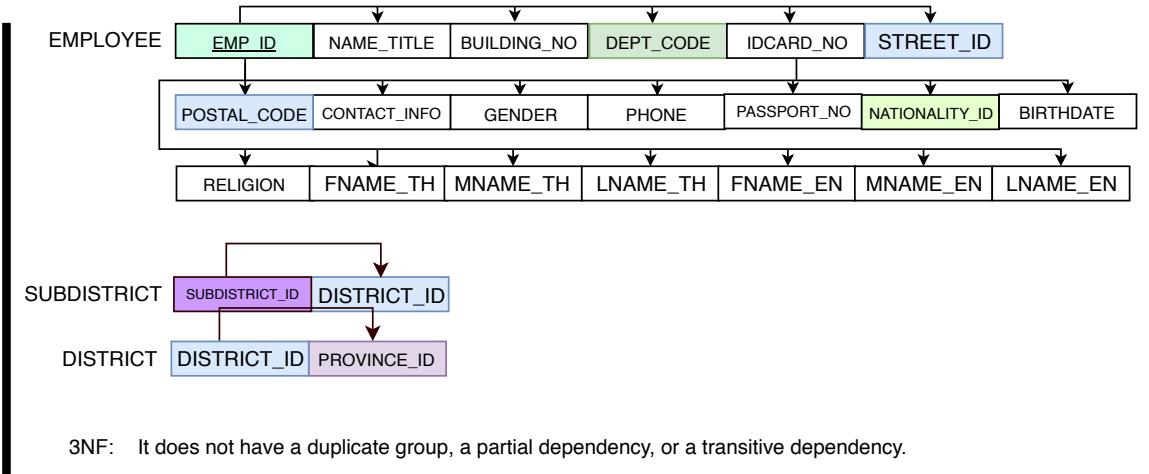
It does not have a duplicate group, a partial dependency, or a transitive dependency.



2NF: EMPLOYEE (EMP_ID , NAME_TITLE , BUILDING_NO , PROVINCE_ID , DEPT_CODE , DISTRICT_ID , STREET_ID ,
SUBDISTRICT_ID , POSTAL_CODE , CONTACT_INFO , GENDER , PHONE , IDCARD_NO , PASSPORT_NO ,
NATIONALITY_ID , BIRTHDATE , RELIGION , FNAME_TH , MNAME_TH , LNAME_TH , FNAME_EN ,
MNAME_EN , LNAME_EN)

PK: EMP_ID FK: ADDRESS_ID, NATIONALITY_ID, DEPT_CODE

It does not have a duplicate group and a partial dependency. But there are transitive dependencies.



3NF: It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: NURSE (EMP_ID , SKILL , CLINIC_ID)

PK: EMP_ID FK: EMP_ID

It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: NATIONAL_LIST (NATIONALITY_ID , NATIONAL_NAME)

PK: NATIONALITY_ID

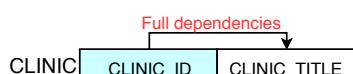
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: DOCTOR (EMP_ID , SPECIAL_FIELD , CLINIC_ID , FIELD_ID)

PK: EMP_ID FK: EMP_ID

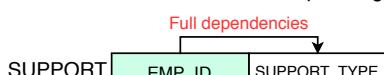
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: CLINIC (CLINIC_ID , TITLE)

PK: CLINIC_ID

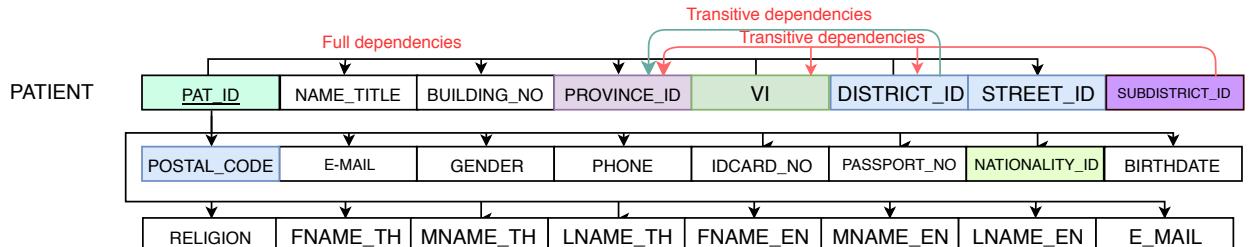
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: SUPPORT (EMP_ID , SUPPORT_TYPE)

PK: EMP_ID FK: EMP_ID

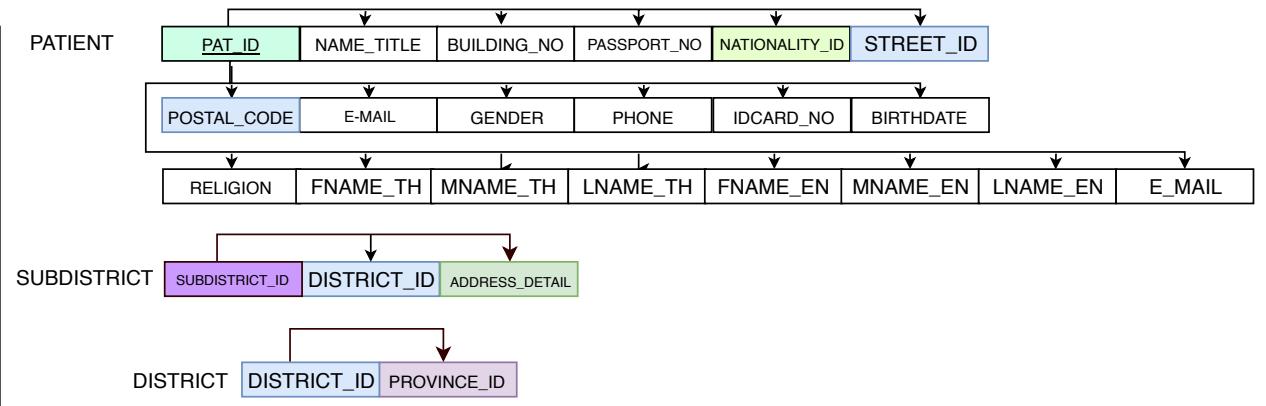
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: PATIENT (EMP_ID , NAME_TITLE , BUILDING_NO , PROVINCE_ID , ADDRESS_DETAIL , DISTRICT_ID , STREET_ID , SUBDISTRICT_ID , POSTAL_CODE , E-MAIL , GENDER , PHONE , IDCARD_NO , PASSPORT_NO , NATIONALITY_ID , BIRTHDATE , RELIGION , FNAME_TH , MNAME_TH , LNAME_TH , FNAME_EN , MNAME_EN , LNAME_EN)

PK: PAT_ID FK: ADDRESS_ID, NATIONALITY_ID

It does not have a duplicate group and a partial dependency. But there are transitive dependencies.



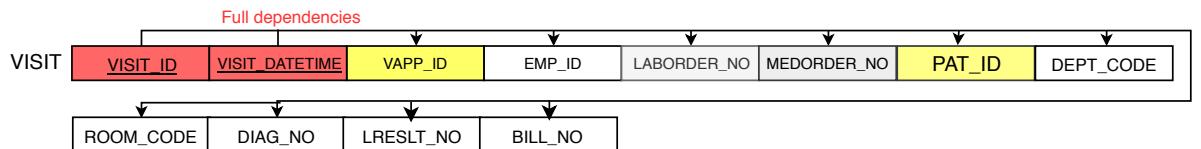
3NF: It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: DOCTOR_APPOINTMENT (VAPP_ID , PAT_ID , VAPP_REGISTER , VAPP_SCHEDULE , VAPP_STATUS , EMP_ID)

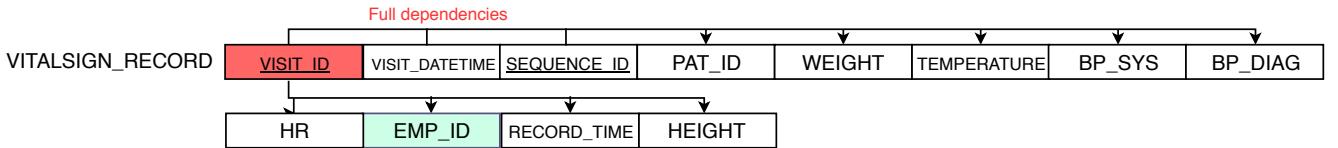
PK: VAPP_ID FK: PAT_ID, EMP_ID

It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: VISIT (VISIT_ID , VISIT_DATETIME , VAPP_ID , EMP_ID , LABORDER_NO , MEDORDER_NO ,
PAT_ID , DEPT_CODE , DIAG_NO , LRESLT_NO , BILL_NO)

It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: VITALSIGN_RECORD (VISIT_ID , VISIT_DATETIME , SEQUENCE_ID , PAT_ID , WEIGHT , TEMPERATURE ,
BP_SYS , BP_DIAG , HR , EMP_ID , RECORD_TIME , HEIGHT)

PK: VISIT_ID, SEQUENCE_ID FK: EMP_ID

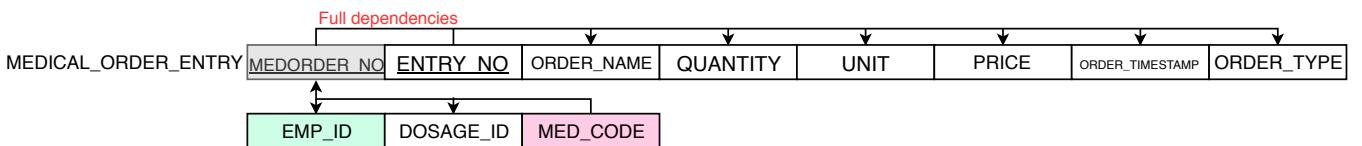
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: MEDICAL_ORDER (MEDORDER_NO , TOTALPRICE)

PK: MEDORDER_NO

It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: MEDICAL_ORDER_ENTRY (MEDORDER_NO , ENTRY_NO , ORDER_NAME , QUANTITY , UNIT , PRICE ,
ORDER_TIMESTAMP , ORDER_TYPE , EMP_ID , DOSAGE_ID , MED_CODE)

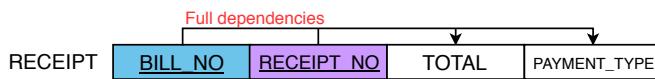
PK: MEDORDER_NO, ENTRY_NO

FK: EMP_ID, MED_CODE

It does not have a duplicate group, a partial dependency, or a transitive dependency.

| | | |
|--|--|--|
| | Full dependencies | |
| MEDICINE | <u>MED_CODE</u> MED_NAME EXPIRED_DATE GENERAL_NAME MEDTYPE_ID PRICE DOSAGE_ID SHELF_POSITION | |
| 3NF: MEDICINE (<u>MED_CODE</u> , MED_NAME , EXPIRED_DATE , GENERAL_NAME , MEDTYPE_ID, PRICE, DOSAGE_ID, SHELF_POSITION) | | |
| PK: EQPM_CODE, MED_CODE | | |
| It does not have a duplicate group, a partial dependency, or a transitive dependency. | | |
| | Full dependencies | |
| MEDICINE_TYPE_LIST | <u>MEDTYPE_ID</u> MEDTYPE_TITLE | |
| 3NF: MEDICINE_TYPE_LIST (<u>MEDTYPE_ID</u> , MEDTYPE_TITLE) | | |
| PK: MEDTYPE_ID | | |
| It does not have a duplicate group, a partial dependency, or a transitive dependency. | | |
| | Full dependencies | |
| DOSAGE_LIST | <u>DOSAGE_ID</u> DOSAGE_DESCRIPTION | |
| 3NF: DOSAGE_LIST (<u>DOSAGE_ID</u> , DOSAGE_DESCRIPTION) | | |
| PK: DOSAGE_ID | | |
| It does not have a duplicate group, a partial dependency, or a transitive dependency. | | |
| | Full dependencies | |
| ICD_10 | <u>ICD10_CODE</u> ICD10_DESCRIPTION | |
| 3NF: ICD_10 (<u>ICD_CODE</u> , ICD10_DESCRIPTION) | | |
| PK: ICD_CODE | | |
| It does not have a duplicate group, a partial dependency, or a transitive dependency. | | |
| | Full dependencies | |
| DIAGNOSIS_RESULT | <u>DIAG_NO</u> DESCRIPTON DATETIME ICD10_CODE | |
| 3NF: DIAGNOSIS_RESULT (<u>DIAG_NO</u> , DESCRIPTON , DATETIME , ICD10_CODE) | | |
| PK: DIAG_NO FK: ICD10_CODE | | |
| It does not have a duplicate group, a partial dependency, or a transitive dependency. | | |
| | Full dependencies | |
| LAB_ORDER | <u>LAB_ORDER_NO</u> PAT_ID EMP_ID TOTALPRICE ORDER_DATETIME ROOM_CODE | |
| 3NF: LAB_ORDER (<u>LAB_ORDER_NO</u> , PAT_ID , EMP_ID , TOTALPRICE , ORDER_DATETIME, ROOM_CODE) | | |
| PK: LAB_ORDER_NO FK: PAT_ID, EMP_ID, ROOM_CODE | | |
| It does not have a duplicate group, a partial dependency, or a transitive dependency. | | |
| | Full dependencies | |
| LAB_RESULT | <u>LRESULT_NO</u> LAB_ORDER_NO ENTRY_NO EMP_ID FEE_CODE | |
| 3NF: LAB_RESULT (<u>LRESULT_NO</u> , FEE_CODE , LABORDER_NO, ENTRY_NO , EMP_ID , LAB_ORDER_NO) | | |
| PK: LAB_ORDER_NO, LRESULT_NO FK: ENTRY_NO, LRESULT_NO, EMP_ID, FEE_CODE | | |
| It does not have a duplicate group, a partial dependency, or a transitive dependency. | | |
| | Full dependencies | |
| LAB_ORDER_ENTRY | <u>LAB_ORDER_NO</u> ENTRY_NO FEE_CODE QUANTITY SPECIMEN STAT WAIT_DAYS | |
| 3NF: LAB_ORDER_ENTRY (<u>LAB_ORDER_NO</u> , ENTRY_NQ , FEE_CODE , QUANTITY , SPECIMEN , STAT , WAIT_DAYS) | | |
| PK: ENTRY_NO, LAB_ORDER_NO FK: ENTRY_NO, LAB_ORDER_NO,, FEE_CODE | | |
| It does not have a duplicate group, a partial dependency, or a transitive dependency. | | |
| | Full dependencies | |
| FEE_ENTRY | <u>FEE_CODE</u> ORDER_DESCRIPTION PRICE | |
| 3NF: FEE_ENTRY (<u>FEE_CODE</u> , ORDER_DESCRIPTION , PRICE) | | |
| PK: FEE_CODE | | |
| It does not have a duplicate group, a partial dependency, or a transitive dependency. | | |
| | Full dependencies | |
| MEDICAL_BILL | <u>BILL_NO</u> TOTAL DATETIME VISIT_DATETIME VISIT_ID DIVIDED_BILL | |
| 3NF: MEDICAL_BILL (<u>BILL_NO</u> , TOTAL , DATETIME , VISIT_DATETIME , VISIT_ID , DIVIDED_BILL) | | |

PK: BILL_NO FK: VISIT_DATETIME, VISIT_ID
It does not have a duplicate group, a partial dependency, or a transitive dependency.



3NF: RECEIPT (BILL_NO , RECEIPT_NO , TOTAL , PAYMENT_TYPE)

PK: BILL_NO, RECEIPT_NO FK: BILL_NO

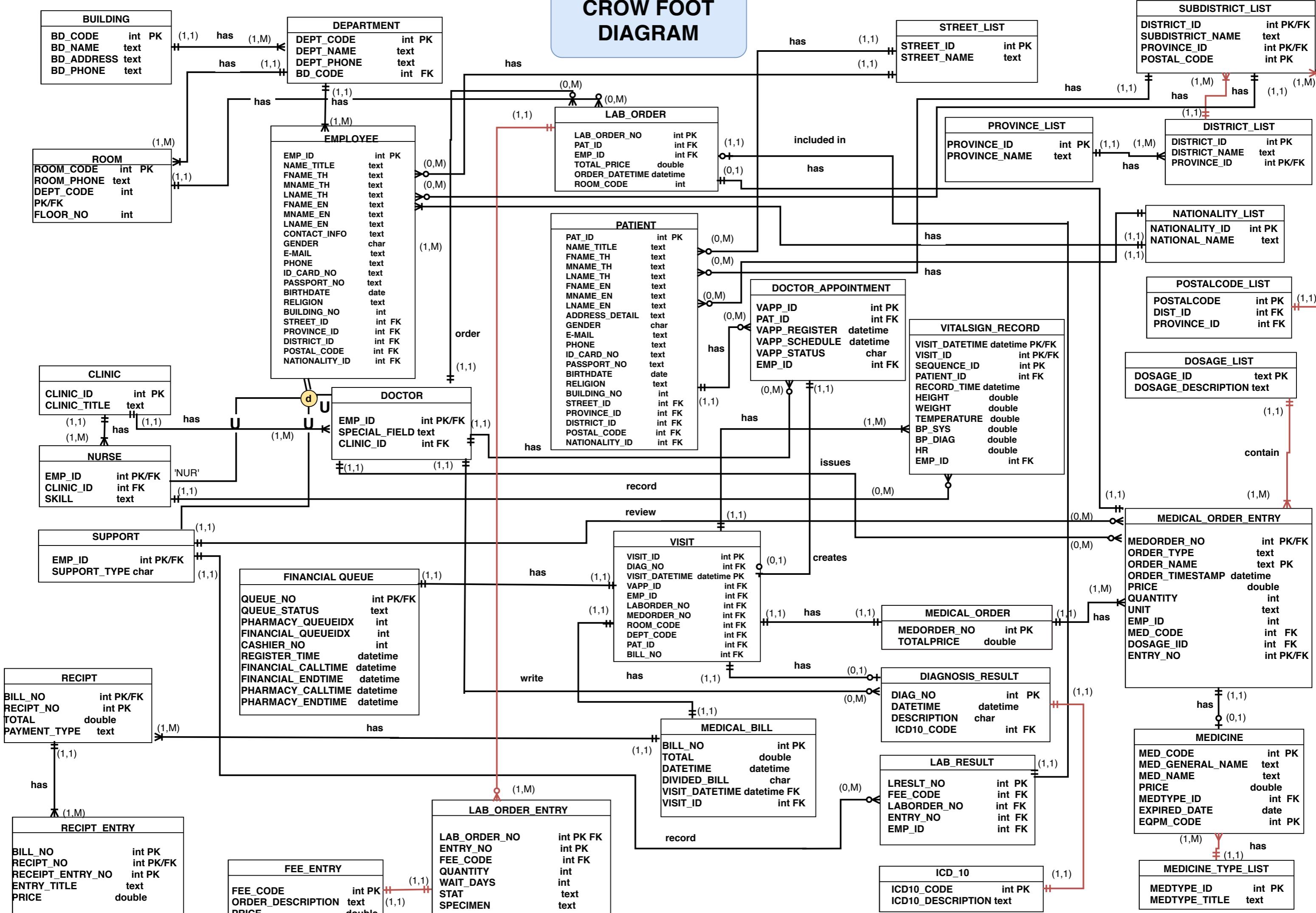
It does not have a duplicate group, a partial dependency, or a transitive dependency.



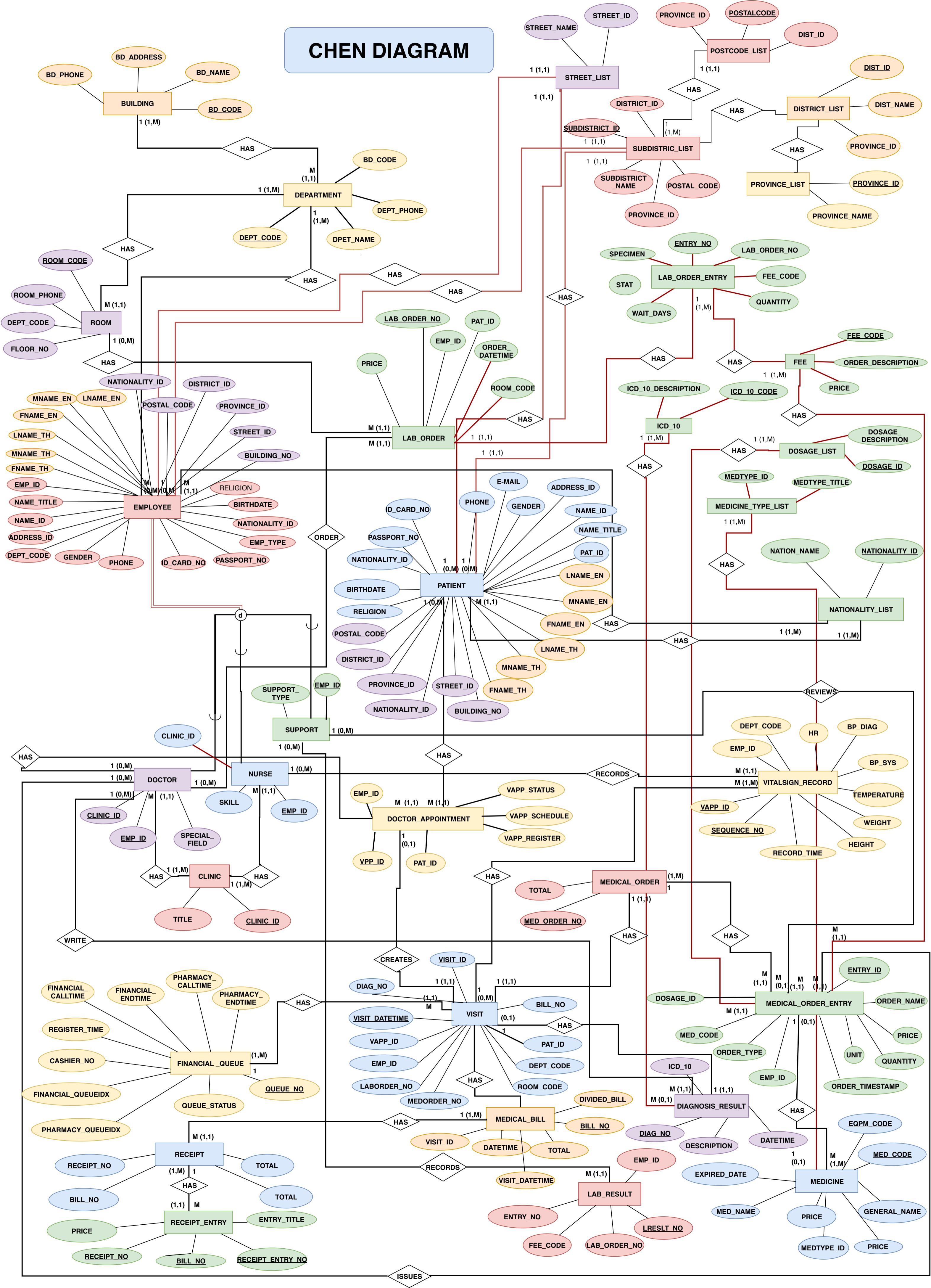
3NF: RECEIPT_ENTRY (BILL_NO , RECEIPT_NO , RECEIPT_ENTRY_NO , ENTRY_TITLE , PRICE)
PK: BILL_NO, RECEIPT_NO, RECEIPT_ENTRY_NO FK: BILL_NO, RECEIPT_NO

It does not have a duplicate group, a partial dependency, or a transitive dependency.

CROW FOOT DIAGRAM



CHEN DIAGRAM



Project Phase #2

Comments and Errors from Project Phase 1

Crow Diagram

- Do NOT use diamond shape for relationship --> only line
- Doctor Appointment and Visit should NOT be EER
- Review the business rule and diagram
- Patient cannot have medical bill --> Visit has medical bill and patient has (0,M) visit
- Write (min, max)
- Write PK and FK
- Combine medicine and medical equipment
- Edited attribute according to all others

Relational Schema

- Identifying Primary Key and Drawing the relationship between Schema.
- Added relational lines indicating relations
- Combine medicine and medical equipment
- Add relationship in medical order and medical order entry

Chen's Diagram

- Added some entities and attributes
- Removed some unrelated from entities and attributes
- Corrected some relationships
- Corrected attribute according to other diagram
- Combine medicine and medical equipment

Data dictionary

- Edited according to the diagrams
- Adding entity and attribute according to diagrams

Queries Description

1. DDL: Data Definition Language [12]

Description

- 1) Create Clinic Table

SQL

```
CREATE TABLE CLINIC (
    CLINIC_ID int PRIMARY KEY,
    CLINIC_TITLE varchar(50) NOT NULL
);
```

Result

| | CLINIC_ID | CLINIC_TITLE |
|--|-----------|--------------|
| | | |

Description

- 2) Create Doctor Table

SQL

```
CREATE TABLE DOCTOR (
    EMP_ID int PRIMARY KEY,
    CLINIC_ID int NOT NULL,
    FOREIGN KEY (EMP_ID) REFERENCES EMPLOYEE(EMP_ID),
    FOREIGN KEY (CLINIC_ID) REFERENCES CLINIC(CLINIC_ID)
);
```

Result

| | EMP_ID | CLINIC_ID |
|--|--------|-----------|
| | | |

Description

- 3) Create a Patient Table

SQL

```
CREATE TABLE PATIENT (
    PAT_ID int PRIMARY KEY,
    NAME_TITLE varchar(8),
    FNAME_TH varchar(30),
    MNAME_TH varchar(30),
    LNAME_TH varchar(30),
    FNAME_EN varchar(30),
    MNAME_EN varchar(30),
    LNAME_EN varchar(30),
    GENDER varchar(8) NOT NULL,
    PHONE_NO varchar(10),
    BIRTHDATE date,
    RELIGION varchar(20),
    BUILDING_NO varchar(10),
    IDCARD_NO char(13),
    PASSPORT_NO char(9),
    E_MAIL varchar(30),
);
```

Result

| PAT_ID | NAME_TITLE | FNAME_TH | MNAME_TH | LNAME_TH | FNAME_EN | MNAME_EN | LNAME_EN | GENDER | PHONE_NO | BIRTHDATE | RELIGION | BUILDING_NO | IDCARD_NO | PASSPORT_NO | E_MAIL |
|--------|------------|----------|----------|----------|----------|----------|----------|--------|----------|-----------|----------|-------------|-----------|-------------|--------|
|--------|------------|----------|----------|----------|----------|----------|----------|--------|----------|-----------|----------|-------------|-----------|-------------|--------|

Description

- 4) Create Visit Table

SQL

```
CREATE TABLE VISIT (
    VISIT_ID int NOT NULL,
    VISIT_DATETIME datetime NOT NULL,
    VAPP_ID int,
    EMP_ID int NOT NULL,
    PAT_ID int NOT NULL,
    MEDORDER_NO int NOT NULL,
    DIAG_NO int NOT NULL,
    BILL_NO int,
    CONSTRAINT PK_VISIT PRIMARY KEY (VISIT_ID, VISIT_DATETIME),
    FOREIGN KEY (VAPP_ID) REFERENCES DOCTOR_APPOINTMENT(VAPP_ID),
    FOREIGN KEY (PAT_ID) REFERENCES PATIENT(PAT_ID),
    FOREIGN KEY (EMP_ID) REFERENCES DOCTOR(EMP_ID),
    FOREIGN KEY (MEDORDER_NO) REFERENCE MEDICAL_ORDER(MEDORDER_NO),
    FOREIGN KEY (DIAG_NO) REFERENCES DIAGNOSIS_RESULT(DIAG_NO),
);
```

Result

| VISIT_ID | VISIT_DATETIME | VAPP_ID | EMP_ID | PAT_ID | MEDORDER_NO | DIAG_NO | BILL_NO |
|----------|----------------|---------|--------|--------|-------------|---------|---------|
|----------|----------------|---------|--------|--------|-------------|---------|---------|

Description

- 5) Create Diagnostic Result Table

SQL

```
CREATE TABLE DIAGNOSIS_RESULT (
    DIAG_NO int PRIMARY KEY,
    DIAG_DESCRIPTION varchar(100) NOT NULL,
    DIAG_DATETIME datetime NOT NULL,
    ICD10_CODE char(7) NOT NULL,
    FOREIGN KEY (ICD10_CODE) REFERENCES ICD_10(ICD10_CODE)
);
```

Result

| | DIAG_NO | DIAG_DESCRIPTION | DIAG_DATETIME | ICD10_CODE |
|--|---------|------------------|---------------|------------|
|--|---------|------------------|---------------|------------|

Description

- 6) Create Medicine Table

SQL

```
CREATE TABLE MEDICINE (
    MED_CODE int PRIMARY KEY,
    MED_NAME varchar(50) NOT NULL,
    MED_GENERAL_NAME varchar(50),
    PRICE decimal(10,2) NOT NULL,
    EXPIRED_DATE date NOT NULL,
);
```

Result

| | MED_CODE | MED_NAME | MED_GENERAL_NAME | PRICE | EXPIRED_DATE |
|--|----------|----------|------------------|-------|--------------|
|--|----------|----------|------------------|-------|--------------|

Description

7) Create Medical Order Entry Table

SQL

```
CREATE TABLE MEDICAL_ORDER_ENTRY (
    MEDORDER_NO int NOT NULL,
    ENTRY_NO int NOT NULL,
    ORDER_NAME varchar(40) NOT NULL,
    QUANTITY int NOT NULL,
    UNIT varchar(30),
    PRICE decimal(10,2) NOT NULL,
    ORDER_TIMESTAMP datetime NOT NULL,
    ORDER_TYPE varchar(30),
    EMP_ID int,
    MED_CODE int
    CONSTRAINT PK_MEDICAL_ORDER_ENTRY PRIMARY KEY
    (MEDORDER_NO, ENTRY_NO),
    FOREIGN KEY (EMP_ID) REFERENCES DOCTOR(EMP_ID),
    FOREIGN KEY (MEDORDER_NO) REFERENCES
    MEDICAL_ORDER(MEDORDER_NO),
    FOREIGN KEY (MED_CODE) REFERENCES MEDICINE(MED_CODE)
);
```

Result

| | MEDORDER_NO | ENTRY_NO | ORDER_NAME | QUANTITY | UNIT | PRICE | ORDER_TIMESTAMP | ORDER_TYPE | EMP_ID | MED_CODE |
|--|-------------|----------|------------|----------|------|-------|-----------------|------------|--------|----------|
|--|-------------|----------|------------|----------|------|-------|-----------------|------------|--------|----------|

Description

8) Create Medical Order Table

SQL

```
CREATE TABLE MEDICAL_ORDER (
    MEDORDER_NO int PRIMARY KEY,
    TOTAL_PRICE decimal(10,2) NOT NULL
);
```

Result

| | MEDORDER_NO | TOTAL_PRICE |
|--|-------------|-------------|
|--|-------------|-------------|

Description

9) Create Receipt Entry Table

SQL

```
CREATE TABLE RECEIPT_ENTRY (
    BILL_NO int NOT NULL,
    RECEIPT_NO int NOT NULL,
    RECEIPT_ENTRY_NO int NOT NULL,
    ENTRY_TITLE varchar(50) NOT NULL,
    PRICE decimal(10,2) NOT NULL,
    CONSTRAINT PK_RECEIPT_ENTRY PRIMARY KEY
    (BILL_NO, RECEIPT_NO, RECEIPT_ENTRY_NO),
    FOREIGN KEY (BILL_NO, RECEIPT_NO) REFERENCES
    RECEIPT(BILL_NO, RECEIPT_NO)
);
```

Result

| | BILL_NO | RECEIPT_NO | RECEIPT_ENTRY_NO | ENTRY_TITLE | PRICE |
|--|---------|------------|------------------|-------------|-------|
|--|---------|------------|------------------|-------------|-------|

Description

10) Create Receipt Table

SQL

```
CREATE TABLE RECEIPT (
    BILL_NO int NOT NULL,
    RECEIPT_NO int NOT NULL,
    TOTAL decimal(10,2) NOT NULL,
    PAYMENT_TYPE varchar(30) NOT NULL,
    CONSTRAINT PK_RECEIPT PRIMARY KEY (BILL_NO, RECEIPT_NO),
    FOREIGN KEY (BILL_NO) REFERENCES MEDICAL_BILL(BILL_NO)
)
```

Result

| | BILL_NO | RECEIPT_NO | TOTAL | PAYMENT_TYPE |
|--|---------|------------|-------|--------------|
|--|---------|------------|-------|--------------|

Description

11) Alter the Doctor table and add med_record as a new attribute

SQL

```
ALTER TABLE DOCTOR  
ADD med_record varchar(10);
```

Result

| Columns | EMP_ID | CLINIC_ID | med_record |
|--------------------------------|--------|-----------|------------|
| EMP_ID (PK, FK, int, not null) | 1 | 30001143 | NULL |
| CLINIC_ID (FK, int, not null) | 2 | 30099998 | NULL |
| med_record (varchar(10), null) | | | |

Description

12) Drop column med_record from Doctor Table

SQL

```
ALTER TABLE DOCTOR  
DROP COLUMN med_record;
```

Result

| Columns | EMP_ID | CLINIC_ID |
|--------------------------------|--------|-----------|
| EMP_ID (PK, FK, int, not null) | 1 | 30001143 |
| CLINIC_ID (FK, int, not null) | 2 | 30099998 |

2. Insert, Update, Delete [10]

Description

- 1) Insert Clinic Information to clinic table

SQL

```
INSERT INTO CLINIC (CLINIC_ID,CLINIC_TITLE)
VALUES (110010,'Cardiac'),
(110012,'Health visiting'),
(110013,'Nursing Informatics'),
(110014,'Neurosurgical'),
(110015,'Neonatal'),
(110018,'Genetics'),
(110041,'Dental'),
(110033,'Diabetes'),
(110045,'Neurosurgical'),
(110067,'Orthopaedic');
```

Result

| | CLINIC_ID | CLINIC_TITLE |
|----|-----------|---------------------|
| 1 | 110010 | Cardiac |
| 2 | 110012 | Health visiting |
| 3 | 110013 | Nursing Informatics |
| 4 | 110014 | Neurosurgical |
| 5 | 110015 | Neonatal |
| 6 | 110018 | Genetics |
| 7 | 110033 | Diabetes |
| 8 | 110041 | Dental |
| 9 | 110045 | Neurosurgical |
| 10 | 110067 | Orthopaedic |

Description

- 2) Insert Doctors Employee ID information to clinic table

SQL

```
INSERT INTO DOCTOR(EMP_ID, CLINIC_ID)
VALUES
(30333345, 110010),
(30993355, 110012),
(30001143, 110013),
(30113141, 110010),
(30448342, 110013),
(30099998, 110014),
(30932131, 110018),
(30332446, 110041),
(30948440, 110033),
(30912400, 110010);
```

Result

| | EMP_ID | CLINIC_ID |
|----|----------|-----------|
| 1 | 30001143 | 110013 |
| 2 | 30099998 | 110014 |
| 3 | 30113141 | 110010 |
| 4 | 30332446 | 110041 |
| 5 | 30333345 | 110010 |
| 6 | 30448342 | 110013 |
| 7 | 30912400 | 110010 |
| 8 | 30932131 | 110018 |
| 9 | 30948440 | 110033 |
| 10 | 30993355 | 110012 |

Description

- 3) Insert information to patient table

SQL

```
INSERT INTO PATIENT
(PAT_ID, NAME_TITLE, FNAME_TH, MNAME_TH, LNAME_TH, FNAME_EN, MNAME_EN
, LNAME_EN, GENDER, PHONE_NO, BIRTHDATE, RELIGION, BUILDING_NO, IDCARD
_NO, PASSPORT_NO, E_MAIL)
```

VALUES

(58000001, 'นางสาว', 'วิชญา', NULL, 'ชื่อสั้นๆ',
 'VIDCHAYADA', NULL, 'CHAISANGKHA', 'F', '0911115893',
 '12-Feb-1998', 'Buddhism', 1, '1710100456834',
 'AA1231256', 'vch.za@gmail.com'),

(58000232, 'นาง', 'ร่มita', NULL, 'ช่องแขม',
 'RAMITA', NULL, 'CHONGCHAM', 'F', '0981737748',
 '20-Aug-1960', 'Buddhism', 2, '5580300238755',
 'AA9444856', 'ramita23@gmail.com'),

(59330043, 'นางสาว', 'อ้อร์วี', NULL, 'ตอนนัมป์รี', 'ORRAWI', NULL,
 'DONNOMPRI', 'F', '0818656654', '8-Nov-1993'
 'Buddhism', 1, '2324499000812', NULL
 ,'Aoraoraor@hotmail.com'),

(59330555, 'นาย', 'คณิสร', NULL, 'อิศรัณกุล', 'KANISORN', NULL,
 'ISSARANKURA', 'M', '0942245562', '7-Jul-1977'
 ,'Buddhism', 1, '1210911013878', 'AA9889064',
 'Kaniteach@gmail.com'),

(59420005, 'นาย', 'จัสติน', 'ดริว', 'เบอร์', 'JUSTIN'
 ,'DREW', 'BIEBER', 'M', '0332224562', '9-Jan-1940',
 'Christianity', 3, NULL, 'PA0048355'
 ,'justinloveseveryone@gmail.com'),

(59331022, 'นาง', '瓦丽莎拉', NULL, 'จันมาทนาวัฒน์',
 'VARISARA', NULL, 'JANMATAKULWAT', 'F',
 '0645568424', '10-Jan-2000', 'Buddhism', 2,
 '5589200345876', 'AB2342222', 'varis@hotmail.com'),

(59300007, 'นาย', 'จิรากร', NULL, 'จิราพลปลายเส้า',
 'JIRAKORN', NULL, 'JIRAPONPAISAL', 'M'
 ,'0552352255', '11-Apr-1985', 'Buddhism', 2,
 '2229487009231', NULL, 'Jira12345@gmail.com'),

(60000008, 'นาย', 'วรวิตร', NULL, 'แก้วประเสริฐ', 'WORAWIT', NULL,
 'KAEWPRASERT', 'M', '0813315245', '30-Aug-1993', 'Islam',
 1, '5568399458009', 'AA4456254', 'Nattaaa@gmail.com'),

(60330009, 'นางสาว', 'ณัฐพร', NULL, 'นาราษัน', 'NATTHAPORN'
 ,NULL, 'NARAYAN', 'F', '0891100033', '14-Mar-1994',
 'Buddhism', 2, '2341233987098', NULL,
 'Natandnon@gmail.com'),

(60000230, 'นาย', 'ภูมิรักษ์', NULL, 'ภูมิเมียนแซบ', 'PHUMRAPEE',
 NULL, 'LIMPIANCHOP', 'M', '0821146790', '14-Aug-1998'
 ,'Islam', 1, '6545666789076', 'AC2220593'
 ,'Phum.Vipu@gmail.com');

Result

| | PAT_ID | NAME_TIT... | FNAME_TH | MNAME_... | LNAME_TH | FNAME_EN | MNAME_EN | LNAME_EN | GENDER | PHONE_NO | BIRTHDATE | RELIGION | BUILDING_... | IDCARD_NO | PASSPORT_... | E_MAIL |
|----|----------|-------------|----------|-----------|----------------|------------|----------|---------------|--------|------------|------------|--------------|--------------|---------------|--------------|--------------------------------|
| 1 | 58000001 | นางสาว | วิชญา | NULL | ชัยสัจยา | VIDCHAYADA | NULL | CHAISSANGKHA | F | 0911115893 | 1998-02-12 | Buddhism | 1 | 1710100456834 | AA1231256 | vch.za@gmail.com |
| 2 | 58000232 | นาง | ร่มลดา | NULL | ช่องเย็น | RAMITA | NULL | CHONGCHAM | F | 0981737748 | 1960-08-20 | Buddhism | 2 | 5580300238755 | AA9444856 | ramita23@gmail.com |
| 3 | 59300007 | นาย | ธีร์กานา | NULL | จิราพูลภานุเดช | JIRAKORN | NULL | JIRAPONPAISAL | M | 0552392256 | 1985-04-11 | Buddhism | 2 | 2229487009231 | NULL | Jira12345@gmail.com |
| 4 | 59330043 | นางสาว | ออยร์ | NULL | ศอนนท์ปัช | ORRAWI | NULL | DONNOMPRI | F | 0818656654 | 1993-11-08 | Buddhism | 1 | 2324499000812 | NULL | Aoraoraor@hotmail.com |
| 5 | 59330555 | นาย | พนิษฐ์ | NULL | อิศรัตน์กุล | KANISORN | NULL | ISSARANKURA | M | 0942245562 | 1977-07-07 | Buddhism | 1 | 1210911013878 | AA9889064 | Kaniteach@gmail.com |
| 6 | 59331022 | นาง | วีร์ภา | NULL | จันมาขุวัฒน์ | VARISARA | NULL | JANMATAKULWAT | F | 0645568424 | 2000-01-10 | Buddhism | 2 | 5589200345876 | AB2342222 | varis@hotmail.com |
| 7 | 59420005 | นาง | ฉัลลิน | พิศา | บีนอร์ | JUSTIN | DREW | BIEBER | M | 0332224562 | 1940-01-09 | Christianity | 3 | NULL | PA0048355 | justinloveseveryone@gmail.c... |
| 8 | 60000008 | นาย | วรรติค | NULL | แพรกประดิษฐ์ | WORAWIT | NULL | KAEPRASSERT | M | 0813315245 | 1993-08-30 | Islam | 1 | 5568399458009 | AA4456254 | Nattaaa@gmail.com |
| 9 | 60000230 | นาย | ภูมิเก | NULL | พิมเปียห์สอน | PHUMRAPEE | NULL | LIMPANCHOP | M | 0821146790 | 1998-08-14 | Islam | 1 | 6545666789076 | AC2220593 | Phum.Vipu@gmail.com |
| 10 | 60330009 | นางสาว | ณัฐพ | NULL | นารายณ์ | NATTAPORN | NULL | NARAYAN | F | 0891100033 | 1994-03-14 | Buddhism | 2 | 2341233987098 | NULL | Natandnon@gmail.com |

Description

4) Insert information to visit Table

SQL

```
INSERT INTO VISIT (VISIT_ID, VISIT_DATETIME, VAPP_ID, EMP_ID,
PAT_ID, MEDORDER_NO, DIAG_NO, BILL_NO)
VALUES
(620015, '20-Apr-2019 15:00:00', 6001122, 30333345,
58000001, 610033, 107000, 111022),
(620016, '20-Apr-2019 15:12:01', 6001123, 30993355,
58000232, 610034, 107001, 111023),
(620017, '20-Apr-2019 15:24:02', 6001124, 30001143,
59330043, 610035, 107002, 111024),
(620018, '20-Apr-2019 15:30:03', 6001125, 30113141,
59330555, 610036, 107003, 111025),
(620019, '20-Apr-2019 15:39:04', 6001126, 30448342,
59420005, 610037, 107004, 111026),
(620020, '20-Apr-2019 15:30:04', 6001127, 30099998,
59331022, 610038, 107005, 111027),
(620021, '20-Apr-2019 15:39:05', 6001128, 30932131,
59300007, 610039, 107006, 111028),
(620022, '20-Apr-2019 15:30:05', 6001129, 30332446,
60000008, 610040, 107007, 111029),
(620023, '20-Apr-2019 15:39:06', 6001130, 30948440,
60330009, 610041, 107008, 111030),
```

```
(620024,      '20-Apr-2019 15:30:06',      6001131,      30912400,
60000230,    610042,      107009,      111031)
```

Result

| | VISIT_ID | VISIT_DATETIME | VAPP_ID | EMP_ID | PAT_ID | MEDORDER_NO | DIAG_NO | BILL_NO |
|----|----------|-------------------------|---------|----------|----------|-------------|---------|---------|
| 1 | 620015 | 2019-04-20 15:00:00.000 | 6001122 | 30333345 | 58000001 | 610033 | 107000 | 111022 |
| 2 | 620016 | 2019-04-20 15:12:01.000 | 6001123 | 30993355 | 58000232 | 610034 | 107001 | 111023 |
| 3 | 620017 | 2019-04-20 15:24:02.000 | 6001124 | 30001143 | 59330043 | 610035 | 107002 | 111024 |
| 4 | 620018 | 2019-04-20 15:30:03.000 | 6001125 | 30113141 | 59330555 | 610036 | 107003 | 111025 |
| 5 | 620019 | 2019-04-20 15:39:04.000 | 6001126 | 30448342 | 59420005 | 610037 | 107004 | 111026 |
| 6 | 620020 | 2019-04-20 15:30:04.000 | 6001127 | 30099998 | 59331022 | 610038 | 107005 | 111027 |
| 7 | 620021 | 2019-04-20 15:39:05.000 | 6001128 | 30932131 | 59300007 | 610039 | 107006 | 111028 |
| 8 | 620022 | 2019-04-20 15:30:05.000 | 6001129 | 30332446 | 60000008 | 610040 | 107007 | 111029 |
| 9 | 620023 | 2019-04-20 15:39:06.000 | 6001130 | 30948440 | 60330009 | 610041 | 107008 | 111030 |
| 10 | 620024 | 2019-04-20 15:30:06.000 | 6001131 | 30912400 | 60000230 | 610042 | 107009 | 111031 |

Description

- 5) Insert information to Diagnostic Result Table

SQL

```
INSERT INTO DIAGNOSIS_RESULT(DIAG_NO,      DIAG_DESCRIPTION,
DIAG_DATETIME,      ICD10_CODE)
VALUES
(107000,      'Risk of poverty',      '13-Mar-2019 15:00:00',
'E00-E90'),
(107001,      'Fever',      '13-Mar-2019 15:12:01',      'E00-E90'),
(107002,      'Hypotension',      '13-Mar-2019 15:24:02',      'I00-I99'),
(107003,      'Allergic to nuts',      '13-Mar-2019 15:30:03',
'C00-D48'),
(107004,      'Cardiogenic Shock',      '13-Mar-2019 15:39:04',
'J00-J99'),
(107005,      'Fever',      '13-Mar-2019 16:00:05',      'E00-E90'),
(107006,      'Low Blood Pressure',      '13-Mar-2019 16:20:06',
'G00-G99'),
```

```

(107007,      'Fever',      '13-Mar-2019 16:32:07',      'C00-D48'),
(107008,      'Fever',      '13-Mar-2019 16:40:08',      'A00-B99'),
(107009,      'Risk of poverty',      '14-Mar-2019 09:00:09',
'K00-K93');

```

Result

| | DIAG_NO | DIAG_DESCRIPTION | DIAG_DATETIME | ICD10_CODE |
|----|---------|--------------------|-------------------------|------------|
| 1 | 107000 | Risk of poverty | 2019-03-13 15:00:00.000 | E00-E90 |
| 2 | 107001 | Fever | 2019-03-13 15:12:01.000 | E00-E90 |
| 3 | 107002 | Hypotension | 2019-03-13 15:24:02.000 | I00-I99 |
| 4 | 107003 | Allergic to nuts | 2019-03-13 15:30:03.000 | C00-D48 |
| 5 | 107004 | Cardiogenic Shock | 2019-03-13 15:39:04.000 | J00-J99 |
| 6 | 107005 | Fever | 2019-03-13 16:00:05.000 | E00-E90 |
| 7 | 107006 | Low Blood Pressure | 2019-03-13 16:20:06.000 | G00-G99 |
| 8 | 107007 | Fever | 2019-03-13 16:32:07.000 | C00-D48 |
| 9 | 107008 | Fever | 2019-03-13 16:40:08.000 | A00-B99 |
| 10 | 107009 | Risk of poverty | 2019-03-14 09:00:09.000 | K00-K93 |

Description

- 6) Insert Information to Medicine Table

SQL

```

INSERT INTO MEDICINE(MED_CODE,      MED_NAME,      MED_GENERAL_NAME,
PRICE,      EXPIRED_DATE)
VALUES
(110001,      'Aceon',      'FLOMAX',      1200,      '17-Apr-2020'),
(110002,      'Acetaminophen',      'NAQUA',      100,      '20-Jul-2021'),
(110003,      'Acetic Acid',      'NORPRAMIN',      230,      '20-Jan-2020'),
(110101,      'Baclofen',      'TREXAN',      260,      '01-Jan-2022'),
(110102,      'Benadryl',      'VIVELLE',      120,      '01-Apr-2020'),
(110103,      'Bentyl',      'BENADRYL',      80,      '09-Sep-2022'),
(110104,      'Benzagel',      'FEPANIL',      190,      '30-Apr-2020'),
(110105,      'Bepridil',      'GLUCOPHASE',      2300,      '30-Jun-2021'),
(110106,      'Besivance',      'ROSCILLIN',      245,      '17-Aug-2022'),
(110201,      'Capastat Sulfate',      'SULBASIN',      3492,      '20-Apr-2020');

```

Result

| | MED_CODE | MED_NAME | MED_GENERAL_NAME | PRICE | EXPIRED_DATE |
|----|----------|------------------|------------------|---------|--------------|
| 1 | 110001 | Aceon | FLOMAX | 1200.00 | 2020-04-17 |
| 2 | 110002 | Acetaminophen | NAQUA | 100.00 | 2021-07-20 |
| 3 | 110003 | Acetic Acid | NORPRAMIN | 230.00 | 2020-01-20 |
| 4 | 110101 | Baclofen | TREXAN | 260.00 | 2022-01-01 |
| 5 | 110102 | Benadryl | VIVELLE | 120.00 | 2020-04-01 |
| 6 | 110103 | Bentyl | BENADRYL | 80.00 | 2022-09-09 |
| 7 | 110104 | Benzagel | FEPANIL | 190.00 | 2020-04-30 |
| 8 | 110105 | Bepridil | GLUCOPHASE | 2300.00 | 2021-06-30 |
| 9 | 110106 | Besivance | ROSCILLIN | 245.00 | 2022-08-17 |
| 10 | 110201 | Capastat Sulf... | SULBASIN | 3492.00 | 2020-04-20 |

Description

- 7) Insert Information to Medical Order Table

SQL

```
INSERT INTO MEDICAL_ORDER (MEDORDER_NO, TOTAL_PRICE)
VALUES
(610033,      2500),
(610034,      1200),
(610035,      540),
(610036,      655),
(610037,      295),
(610038,     1090),
(610039,      290),
(610040,      225),
(610041,      550),
(610042,     1290);
```

Result

| | MEDORDER_NO | TOTAL_PRICE |
|----|-------------|-------------|
| 1 | 610033 | 2500.00 |
| 2 | 610034 | 1200.00 |
| 3 | 610035 | 540.00 |
| 4 | 610036 | 655.00 |
| 5 | 610037 | 295.00 |
| 6 | 610038 | 1090.00 |
| 7 | 610039 | 290.00 |
| 8 | 610040 | 225.00 |
| 9 | 610041 | 550.00 |
| 10 | 610042 | 1290.00 |

Description

- 8) Insert Information to Receipt Table

SQL

```
INSERT INTO RECEIPT(BILL_NO, RECEIPT_NO, TOTAL,           PAYMENT_TYPE)
VALUES
(111022,      702201,      3560, 'Cash'),
(111022,      702202,      1560, 'Cash'),
(111022,      702203,      2205, 'Cash'),
(111022,      702204,      2045, 'Cash'),
(111023,      702205,      1000, 'Cash'),
(111024,      702206,      1900, 'Cash'),
(111024,      702207,      1255, 'Cash'),
(111024,      702208,      1900, 'Cash'),
(111024,      702209,      1490, 'Cash'),
(111025,      702210,      1800, 'Cash'),
(111025,      702211,      1500, 'Cash'),
(111025,      702212,      1500, 'Cash'),
(111026,      702213,      3095, 'Credit Card'),
(111026,      702214,      2500, 'Credit Card'),
(111026,      702215,       755, 'Credit Card'),
(111027,      702216,      1630, 'Cash'),
(111027,      702217,       930, 'Cash'),
(111027,      702218,       650, 'Cash'),
```

```

(111028,    702219,    3560, 'Credit Card'),
(111028,    702220,    3505, 'Credit Card'),
(111028,    702221,    1560, 'Credit Card'),
(111028,    702222,    3560, 'Credit Card'),
(111029,    702223,    3220, 'Credit Card'),
(111029,    702224,    1900, 'Credit Card'),
(111030,    702225,    3220, 'Cash'),
(111030,    702226,    2205, 'Cash'),
(111030,    702227,    1045, 'Cash'),
(111031,    702228,    2205, 'Cash'),
(111031,    702229,    3550, 'Cash'),
(111031,    702230,    1900, 'Cash');

```

Result

| | BILL_NO | RECEIPT_NO | TOTAL | PAYMENT_TYPE |
|----|---------|------------|---------|--------------|
| 1 | 111022 | 702201 | 3560.00 | Cash |
| 2 | 111022 | 702202 | 1560.00 | Cash |
| 3 | 111022 | 702203 | 2205.00 | Cash |
| 4 | 111022 | 702204 | 2045.00 | Cash |
| 5 | 111023 | 702205 | 1000.00 | Cash |
| 6 | 111024 | 702206 | 1900.00 | Cash |
| 7 | 111024 | 702207 | 1255.00 | Cash |
| 8 | 111024 | 702208 | 1900.00 | Cash |
| 9 | 111024 | 702209 | 1490.00 | Cash |
| 10 | 111025 | 702210 | 1800.00 | Cash |
| 11 | 111025 | 702211 | 1500.00 | Cash |

Description

- 9) Update the Clinic Title on Clinic Table

SQL

```

UPDATE CLINIC
SET CLINIC_TITLE = 'Phychiatry' WHERE CLINIC_ID = '110014';

```

Result

| | CLINIC_ID | CLINIC_TITLE |
|----|-----------|---------------------|
| 1 | 110010 | Cardiac |
| 2 | 110012 | Health visiting |
| 3 | 110013 | Nursing Informatics |
| 4 | 110014 | Phychiatry |
| 5 | 110015 | Neonatal |
| 6 | 110018 | Genetics |
| 7 | 110033 | Diabetes |
| 8 | 110041 | Dental |
| 9 | 110045 | Neurosurgical |
| 10 | 110067 | Orthopaedic |

Description

- 10) Delete extra clinic from the system

SQL

```
INSERT INTO CLINIC (CLINIC_ID,CLINIC_TITLE)
VALUES (110016,'Nepprology');

DELETE FROM CLINIC WHERE CLINIC_ID = '110016';
```

Result

| | CLINIC_ID | CLINIC_TITLE |
|----|-----------|---------------------|
| 1 | 110010 | Cardiac |
| 2 | 110012 | Health visiting |
| 3 | 110013 | Nursing Informatics |
| 4 | 110014 | Phychiatry |
| 5 | 110015 | Neonatal |
| 6 | 110018 | Genetics |
| 7 | 110033 | Diabetes |
| 8 | 110041 | Dental |
| 9 | 110045 | Neurosurgical |
| 10 | 110067 | Orthopaedic |

2. Basic Queries [18]

Description

- 1) Find the maximum of ICD10 code in the diagnosis_result table [MAX]

SQL

```
WITH Z (ICD10_CODE, MAX_ICD10) AS (SELECT ICD10_CODE,
COUNT(ICD10_CODE) FROM DIAGNOSIS_RESULT GROUP BY ICD10_CODE)
SELECT ICD10_CODE, MAX_ICD10
FROM Z
WHERE MAX_ICD10 = (SELECT MAX(MAX_ICD10) FROM Z)
```

Result

| | ICD10_CODE | MAX_ICD10 |
|---|------------|-----------|
| 1 | E00-E90 | 3 |

Description

- 2) Find the minimum of IDC10 code in the diagnosis_result table

SQL

```
WITH Z (ICD10_CODE, MAX_ICD10) AS (SELECT ICD10_CODE,
COUNT(ICD10_CODE) FROM DIAGNOSIS_RESULT GROUP BY ICD10_CODE)
SELECT ICD10_CODE, MAX_ICD10
FROM Z
WHERE MAX_ICD10 = (SELECT MAX(MAX_ICD10) FROM Z)
```

Result

| | ICD10_CODE | MIN_ICD10 |
|---|------------|-----------|
| 1 | G00-G99 | 1 |
| 2 | I00-I99 | 1 |
| 3 | J00-J99 | 1 |
| 4 | K00-K93 | 1 |
| 5 | A00-B99 | 1 |

Description

- 3) Find the average spending of the patients [AVG]

SQL

```
SELECT AVG(TOTAL) AS AVG_SPENDING FROM MEDICAL_BILL
```

Result

| | AVG_SPENDING |
|---|--------------|
| 1 | 6270.500000 |

Description

- 4) Get the age of every patient in the patient table [Select]

SQL

```
SELECT PAT_ID, NAME_TITLE, FNAME_EN, LNAME_EN,  
FLOOR(DATEDIFF(DAY, BIRTHDATE, Getdate()) / 365.25) AS AGE  
FROM PATIENT
```

Result

| | PAT_ID | NAME_TITLE | FNAME_EN | LNAME_EN | AGE |
|----|----------|------------|------------|----------------|-----|
| 1 | 58000001 | นางสาว | VIDCHAYADA | CH AISANGKHA | 21 |
| 2 | 58000232 | นาง | RAMITA | CHONGCHAM | 58 |
| 3 | 59300007 | นาย | JIRAKORN | JIRAPONPAISAL | 34 |
| 4 | 59330043 | นางสาว | ORRAWI | DONNOMPRI | 25 |
| 5 | 59330555 | นาย | KANISORN | ISSARANKURA | 41 |
| 6 | 59331022 | นาง | VARISARA | JANMATAKULW... | 19 |
| 7 | 59420005 | นาย | JUSTIN | BIEBER | 79 |
| 8 | 60000008 | นาย | WORAWIT | KA EWPRASERT | 25 |
| 9 | 60000230 | นาย | PHUMRAPEE | LIMPIANCHOP | 20 |
| 10 | 60330009 | นางสาว | NATTAPORN | NARAYAN | 25 |

Description

- 5) Get the medicine code and count each quantity from the medical_order_entry table

SQL

```
SELECT COUNT(MED_CODE) as COUNT, MED_CODE  
FROM MEDICAL_ORDER_ENTRY  
GROUP BY MED_CODE
```

Result

| | COUNT | MED_CODE |
|----|-------|----------|
| 1 | 3 | 110001 |
| 2 | 3 | 110002 |
| 3 | 3 | 110003 |
| 4 | 3 | 110101 |
| 5 | 3 | 110102 |
| 6 | 3 | 110103 |
| 7 | 3 | 110104 |
| 8 | 3 | 110105 |
| 9 | 3 | 110106 |
| 10 | 3 | 110201 |

Description

- 6) Get the date time and other details of the visit table

SQL

```
SELECT VISIT_ID, PAT_ID, EMP_ID, VISIT_DATETIME  
FROM VISIT
```

Result

| | VISIT_ID | PAT_ID | EMP_ID | VISIT_DATETIME |
|----|----------|----------|----------|-------------------|
| 1 | 620015 | 58000001 | 30333345 | 2019-04-20 15:... |
| 2 | 620016 | 58000232 | 30993355 | 2019-04-20 15:... |
| 3 | 620017 | 59330043 | 30001143 | 2019-04-20 15:... |
| 4 | 620018 | 59330555 | 30113141 | 2019-04-20 15:... |
| 5 | 620019 | 59420005 | 30448342 | 2019-04-20 15:... |
| 6 | 620020 | 59331022 | 30099998 | 2019-04-20 15:... |
| 7 | 620021 | 59300007 | 30932131 | 2019-04-20 15:... |
| 8 | 620022 | 60000008 | 30332446 | 2019-04-20 15:... |
| 9 | 620023 | 60330009 | 30948440 | 2019-04-20 15:... |
| 10 | 620024 | 60000230 | 30912400 | 2019-04-20 15:... |

Description

- 7) Find the accumulated price classified as cash in the payment_type attribute of receipt table

SQL

```
SELECT SUM(TOTAL) as TOTAL_PRICE  
FROM RECEIPT  
WHERE PAYMENT_TYPE LIKE '%Cash%'
```

Result

| | TOTAL_PRICE |
|---|-------------|
| 1 | 39050.00 |

Description

- 8) Find the accumulated price classified as a credit card in the payment_type attribute of receipt table [Sum]

SQL

```
SELECT SUM(TOTAL) as TOTAL_PRICE  
FROM RECEIPT  
WHERE PAYMENT_TYPE LIKE '%Credit card%'
```

Result

| TOTAL_PRICE | |
|-------------|----------|
| 1 | 23655.00 |

Description

- 9) Find the payment type that patients use most when pay the bills

SQL

```
WITH Z (PAYMENT_TYPE, TOTAL_SUM) AS (SELECT PAYMENT_TYPE,  
SUM(TOTAL) FROM RECEIPT GROUP BY PAYMENT_TYPE)  
SELECT TOTAL_SUM, PAYMENT_TYPE  
FROM Z  
WHERE TOTAL_SUM = (SELECT MAX(TOTAL_SUM) FROM Z)
```

Result

| | TOTAL_SUM | PAYMENT_TYPE |
|---|-----------|--------------|
| 1 | 39050.00 | Cash |

Description

- 10) Find the maximum amount of medicines used in medical_order_entry table [Count]

SQL

```
WITH Z (MED_CODE, MED_COUNT) AS (SELECT MED_CODE,  
COUNT(MED_CODE) FROM MEDICAL_ORDER_ENTRY GROUP BY MED_CODE)  
SELECT MED_CODE, MED_COUNT  
FROM Z  
WHERE MED_COUNT = (SELECT MAX(MED_COUNT) FROM Z)
```

Result

| | MED_CODE | MED_COUNT |
|----|----------|-----------|
| 1 | 110001 | 3 |
| 2 | 110002 | 3 |
| 3 | 110003 | 3 |
| 4 | 110101 | 3 |
| 5 | 110102 | 3 |
| 6 | 110103 | 3 |
| 7 | 110104 | 3 |
| 8 | 110105 | 3 |
| 9 | 110106 | 3 |
| 10 | 110201 | 3 |

Description

- 11) Find the minimum amount of medicines used in medical_order_entry table

SQL

```
WITH Z (MED_CODE, MED_COUNT) AS (SELECT MED_CODE,
COUNT(MED_CODE) FROM MEDICAL_ORDER_ENTRY GROUP BY MED_CODE)
SELECT MED_CODE, MED_COUNT
FROM Z
WHERE MED_COUNT = (SELECT MIN(MED_COUNT) FROM Z)
```

Result

| | MED_CODE | MED_COUNT |
|----|----------|-----------|
| 1 | 110001 | 3 |
| 2 | 110002 | 3 |
| 3 | 110003 | 3 |
| 4 | 110101 | 3 |
| 5 | 110102 | 3 |
| 6 | 110103 | 3 |
| 7 | 110104 | 3 |
| 8 | 110105 | 3 |
| 9 | 110106 | 3 |
| 10 | 110201 | 3 |

Description

12) Create a view that classifies children (age below 13) from the patient table

SQL

```
CREATE VIEW Children AS
SELECT PAT_ID, NAME_TITLE, FNAME_EN, LNAME_EN,
FLOOR(DATEDIFF(DAY, BIRTHDATE, Getdate()) / 365.25) AS AGE
FROM PATIENT
WHERE FLOOR(DATEDIFF(DAY, BIRTHDATE, Getdate()) / 365.25) < 13

SELECT * FROM Children
```

Result

| PAT_ID | NAME_TITLE | FNAME_EN | LNAME_EN | AGE |
|--------|------------|----------|----------|-----|
| | | | | |

(Since there is no patient of age less than 13, the table is empty.)

Description

13) Create a view that classifies teenagers (age between 13 to 19) from the patient table

SQL

```
CREATE VIEW Teen AS
SELECT PAT_ID, NAME_TITLE, FNAME_EN, LNAME_EN,
FLOOR(DATEDIFF(DAY, BIRTHDATE, Getdate()) / 365.25) AS AGE
FROM PATIENT
WHERE FLOOR(DATEDIFF(DAY, BIRTHDATE, Getdate()) / 365.25)
BETWEEN 13 AND 19

SELECT * FROM Teen
```

Result

| | PAT_ID | NAME_TITLE | FNAME_EN | LNAME_EN | AGE |
|---|----------|------------|----------|----------------|-----|
| 1 | 59331022 | นาง | VARISARA | JANMATAKULW... | 19 |

Description

14) Create a view that classifies adults (age between 20 to 59) from the patient table

SQL

```
CREATE VIEW Adult AS
SELECT PAT_ID, NAME_TITLE, FNAME_EN, LNAME_EN,
FLOOR(DATEDIFF(DAY, BIRTHDATE, Getdate()) / 365.25) AS AGE
FROM PATIENT
WHERE FLOOR(DATEDIFF(DAY, BIRTHDATE, Getdate()) / 365.25)
BETWEEN 20 AND 59

SELECT * FROM Adult
```

Result

| | PAT_ID | NAME_TITLE | FNAME_EN | LNAME_EN | AGE |
|---|----------|------------|------------|---------------|-----|
| 1 | 58000001 | นางสาว | VIDCHAYADA | CHAISANGKHA | 21 |
| 2 | 58000232 | นาง | RAMITA | CHONGCHAM | 58 |
| 3 | 59300007 | นาย | JIRAKORN | JIRAPONPAISAL | 34 |
| 4 | 59330043 | นางสาว | ORRAWI | DONNOMPRI | 25 |
| 5 | 59330555 | นาย | KANISORN | ISSARANKURA | 41 |
| 6 | 60000008 | นาย | WORAWIT | KAEWPRASERT | 25 |
| 7 | 60000230 | นาย | PHUMRAPEE | LIMPIANCHOP | 20 |
| 8 | 60330009 | นางสาว | NATTHAPORN | NARAYAN | 25 |

Description

- 15) Create a view that classifies elders (age more than 59) from the patient table

SQL

```
CREATE VIEW Elder AS
SELECT PAT_ID, NAME_TITLE, FNAME_EN, LNAME_EN,
FLOOR(DATEDIFF(DAY, BIRTHDATE, Getdate()) / 365.25) AS AGE
FROM PATIENT
WHERE FLOOR(DATEDIFF(DAY, BIRTHDATE, Getdate()) / 365.25) > 59
```

```
SELECT * FROM Elder
```

Result

| | PAT_ID | NAME_TITLE | FNAME_EN | LNAME_EN | AGE |
|---|----------|------------|----------|----------|-----|
| 1 | 59420005 | นาย | JUSTIN | BIEBER | 79 |

Description

- 16) Calculate which clinic has the most number of doctor

SQL

```
WITH Z (CLINIC_ID, NUM_CLINIC) AS (SELECT CLINIC_ID,  
COUNT(CLINIC_ID) FROM DOCTOR GROUP BY CLINIC_ID)  
SELECT CLINIC_ID, NUM_CLINIC  
FROM Z  
WHERE NUM_CLINIC = (SELECT MAX(NUM_CLINIC) FROM Z)
```

Result

| | CLINIC_ID | NUM_CLINIC |
|---|-----------|------------|
| 1 | 110010 | 3 |

Description

- 17) Calculate the clinic that has the least number of doctor

SQL

```
WITH Z (CLINIC_ID, NUM_CLINIC) AS (SELECT CLINIC_ID,  
COUNT(CLINIC_ID) FROM DOCTOR GROUP BY CLINIC_ID)  
SELECT CLINIC_ID, NUM_CLINIC  
FROM Z  
WHERE NUM_CLINIC = (SELECT MIN(NUM_CLINIC) FROM Z)
```

Result

| | CLINIC_ID | NUM_CLINIC |
|---|-----------|------------|
| 1 | 110014 | 1 |
| 2 | 110018 | 1 |
| 3 | 110033 | 1 |
| 4 | 110041 | 1 |
| 5 | 110012 | 1 |

Description

18) Show how many doctors each clinic has

SQL

```
WITH Z (CLINIC_ID, NUM_CLINIC) AS (SELECT CLINIC_ID,  
COUNT(CLINIC_ID) FROM DOCTOR GROUP BY CLINIC_ID)  
SELECT CLINIC_ID, NUM_CLINIC  
FROM Z  
ORDER BY NUM_CLINIC ASC
```

Result

| | CLINIC_ID | NUM_CLINIC |
|---|-----------|------------|
| 1 | 110014 | 1 |
| 2 | 110018 | 1 |
| 3 | 110033 | 1 |
| 4 | 110041 | 1 |
| 5 | 110012 | 1 |
| 6 | 110013 | 2 |
| 7 | 110010 | 3 |

3. Advanced Queries [10]

Description

- 1) Show doctors information in each clinic

SQL

```
SELECT e.EMP_ID, NAME_TITLE, FNAME_TH, LNAME_TH, FNAME_EN, LNAME_EN,
c.CLINIC_ID, CLINIC_TITLE
FROM EMPLOYEE e
INNER JOIN DOCTOR d
ON e.EMP_ID = d.EMP_ID
INNER JOIN CLINIC c
ON d.CLINIC_ID = c.CLINIC_ID
```

Result

| | EMP_ID | NAME_TITLE | FNAME_TH | LNAME_TH | FNAME_EN | LNAME_EN | CLINIC_ID | CLINIC_TITLE |
|----|----------|------------|------------|-----------|------------|--------------|-----------|---------------------|
| 1 | 30001143 | นาง | ณัฐริดา | ณปคำส์ | NUTTIDA | NAPAKAS | 110013 | Nursing Informatics |
| 2 | 30099998 | นาย | สรัญพจน์ | สุกันเกตุ | SARANPOJ | SUKANNHAKETU | 110014 | Neurosurgical |
| 3 | 30113141 | นางสาว | ชนาธร | แก้วรา | THANATORN | KEWLA | 110010 | Cardiac |
| 4 | 30332446 | นาง | วิภาดา | แก้วทอง | WIPADA | KEWTHONG | 110041 | Dental |
| 5 | 30333345 | นาย | ภูมิพัฒน์ | ไทยเจ็ย | PUMIPAT | THAIJIA | 110010 | Cardiac |
| 6 | 30448342 | นาง | สรัญญา | ทองต่อสัก | SARANNUCH | TRONGTOSAK | 110013 | Nursing Informatics |
| 7 | 30912400 | นาย | พัฒน์ธรงค์ | จอมขามศรี | PATARAPONG | JOMKHAMSRI | 110010 | Cardiac |
| 8 | 30932131 | นาย | วิศรุท | มากrun | VISARUT | MARGUN | 110018 | Genetics |
| 9 | 30948440 | นาง | รังสิมัน | เพ็งพล | RUNGSIAMUN | PERMPOOL | 110033 | Diabetes |
| 10 | 30993355 | นาย | ณภัทร | วงศ์หาญ | NAPHAT | HONGHAN | 110012 | Health visiting |

Description

- 2) Show disease description of ICD_10 from Diagnostic result

SQL

```
SELECT i.ICD10_CODE, ICD10_DESCRIPTION, DIAG_DESCRIPTION
FROM ICD_10 i
INNER JOIN DIAGNOSIS_RESULT d
ON i.ICD10_CODE = d.ICD10_CODE
```

Result

| | ICD10_CODE | ICD10_DESCRIPTION | DIAG_DESCRIPTION |
|----|------------|--------------------------------------|--------------------|
| 1 | E00–E90 | Endocrine, nutritional and meta... | Risk of poverty |
| 2 | E00–E90 | Endocrine, nutritional and meta... | Fever |
| 3 | I00–I99 | Diseases of the circulatory syst... | Hypotension |
| 4 | C00–D48 | Neoplasms | Allergic to nuts |
| 5 | J00–J99 | Diseases of the respiratory syst... | Cardiogenic Shock |
| 6 | E00–E90 | Endocrine, nutritional and meta... | Fever |
| 7 | G00–G99 | Diseases of the nervous system | Low Blood Pressure |
| 8 | C00–D48 | Neoplasms | Fever |
| 9 | A00–B99 | Certain infectious and parasitic ... | Fever |
| 10 | K00–K93 | Diseases of the digestive system | Risk of poverty |

Description

- 3) Show patient information who do not come according to the visit

SQL

Inner join

```
SELECT
p.PAT_ID, NAME_TITLE, FNAME_TH, LNAME_TH, FNAME_EN, LNAME_EN, VISIT_I
D, VISIT_DATETIME
FROM PATIENT p
LEFT JOIN VISIT
ON PATIENT.PAT_ID = VISIT.PAT_ID
WHERE PAT_ID IS NULL;
```

Subqueries

```
SELECT p.PAT_ID, NAME_TITLE, FNAME_TH, LNAME_TH, FNAME_EN,
LNAME_EN
FROM PATIENT p
WHERE PAT_ID NOT IN (SELECT PAT_ID FROM VISIT)
```

Result

| | PAT_ID | NAME_TITLE | FNAME_TH | LNAME_TH | FNAME_EN | LNAME_EN | VISIT_ID | VISIT_DATETIME |
|--|--------|------------|----------|----------|----------|----------|----------|----------------|
| | PAT_ID | NAME_TITLE | FNAME_TH | LNAME_TH | FNAME_EN | LNAME_EN | | |

Description

- 4) List of patients who have the highest medical bill

SQL

```
SELECT p.PAT_ID, FNAME_EN, LNAME_EN, TOTAL
FROM PATIENT p
INNER JOIN MEDICAL_BILL m
ON m.PAT_ID = p.PAT_ID
ORDER BY TOTAL DESC
```

Result

| | PAT_ID | FNAME_EN | LNAME_EN | TOTAL |
|----|----------|------------|---------------|----------|
| 1 | 59300007 | JIRAKORN | JIRAPONPAISAL | 12185.00 |
| 2 | 58000001 | VIDCHAYADA | CH AISANGKHA | 9370.00 |
| 3 | 60000230 | PHUMRAPEE | LIMPIANCHOP | 7655.00 |
| 4 | 59330043 | ORRAWI | DONNOMPRI | 6545.00 |
| 5 | 60330009 | NATTHAPORN | NARAYAN | 6470.00 |
| 6 | 59420005 | JUSTIN | BIEBER | 6350.00 |
| 7 | 60000008 | WORAWIT | KAEWPRASERT | 5120.00 |
| 8 | 59330555 | KANISORN | ISSARANKURA | 4800.00 |
| 9 | 59331022 | VARISARA | JANMATAKULWAT | 3210.00 |
| 10 | 58000232 | RAMITA | CHONGCHAM | 1000.00 |

Description

- 5) List of the aggregated sum of money each clinic makes

SQL

```
WITH Z (EMP_ID, SUM_MONEY) AS (
    SELECT EMP_ID, SUM(PRICE)
    FROM MEDICAL_ORDER_ENTRY
    GROUP BY EMP_ID
),
Y (CLINIC_ID, CLINIC_TITLE, SUM_MONEY) AS (
    SELECT C.CLINIC_ID, C.CLINIC_TITLE, SUM(Z.SUM_MONEY)
    FROM Z
    JOIN DOCTOR D ON Z.EMP_ID = D.EMP_ID
    JOIN CLINIC C ON C.CLINIC_ID = D.CLINIC_ID
    GROUP BY C.CLINIC_ID, C.CLINIC_TITLE
)
SELECT *
FROM Y
ORDER BY SUM_MONEY DESC
```

Result

| | CLINIC_ID | CLINIC_TITLE | SUM_MONEY |
|---|-----------|-------------------|-----------|
| 1 | 110010 | Cardiac | 8400.00 |
| 2 | 110013 | Nursing Inform... | 7475.00 |
| 3 | 110041 | Dental | 3810.00 |
| 4 | 110014 | Neurosurgical | 3490.00 |
| 5 | 110033 | Diabetes | 1790.00 |
| 6 | 110012 | Health visiting | 1720.00 |
| 7 | 110018 | Genetics | 1370.00 |

Description

- 6) List of patients who have the most visits

SQL

```
WITH Z (PAT_ID, COUNT_VISIT) AS (
```

```

        SELECT PAT_ID, COUNT(VISIT_ID)
        FROM VISIT
        GROUP BY PAT_ID
    )
SELECT Z.COUNT_VISIT, Z.PAT_ID, P.NAME_TITLE, P.FNAME_EN,
P.LNAME_EN
FROM Z
INNER JOIN PATIENT P
ON Z.PAT_ID = P.PAT_ID
WHERE Z.COUNT_VISIT = (SELECT MAX(COUNT_VISIT) FROM Z)

```

Result

| | COUNT_VISIT | PAT_ID | NAME_TITLE | FNAME_EN | LNAME_EN |
|----|-------------|----------|------------|------------|----------------|
| 1 | 1 | 58000001 | นางสาว | VIDCHAYADA | CHAISANGKHA |
| 2 | 1 | 58000232 | นาง | RAMITA | CHONGCHAM |
| 3 | 1 | 59300007 | นาย | JIRAKORN | JIRAPONPAISAL |
| 4 | 1 | 59330043 | นางสาว | ORRAWI | DONNOMPRI |
| 5 | 1 | 59330555 | นาย | KANISORN | ISSARANKURA |
| 6 | 1 | 59331022 | นาง | VARISARA | JANMATAKULW... |
| 7 | 1 | 59420005 | นาย | JUSTIN | BIEBER |
| 8 | 1 | 60000008 | นาย | WORAWIT | KAEWPRASERT |
| 9 | 1 | 60000230 | นาย | PHUMRAPEE | LIMPIANCHOP |
| 10 | 1 | 60330009 | นางสาว | NATTHAPORN | NARAYAN |

Description

- 7) The disease that most patients have

SQL

```

WITH Z (ICD10_CODE, MAX_ICD10) AS (
    SELECT ICD10_CODE, COUNT(ICD10_CODE)
    FROM DIAGNOSIS_RESULT
    GROUP BY ICD10_CODE
)
SELECT z.ICD10_CODE, z.MAX_ICD10, I.ICD10_DESCRIPTION
FROM Z
INNER JOIN ICD_10
ON I.ICD10_CODE = Z.ICD10_CODE
WHERE MAX_ICD10 = (SELECT MAX(MAX_ICD10) FROM Z)

```

Result

| | ICD10_CODE | MAX_ICD10 | ICD10_DESCRIP... |
|---|------------|-----------|--------------------|
| 1 | E00-E90 | 3 | Endocrine, nutr... |

Description

- 8) The number of patients that each doctor have treated

SQL

```
WITH Z (EMP_ID, COUNT_PAT) AS (
    SELECT EMP_ID, COUNT(PAT_ID)
    FROM VISIT
    GROUP BY EMP_ID
)
SELECT Z.EMP_ID, e.NAME_TITLE, e.FNAME_EN, e.LNAME_EN,
Z.COUNT_PAT
FROM Z
INNER JOIN EMPLOYEE e
ON e.EMP_ID = Z.EMP_ID
```

Result

| | EMP_ID | NAME_TITLE | FNAME_EN | LNAME_EN | COUNT_PAT |
|----|----------|------------|------------|--------------|-----------|
| 1 | 30001143 | นาง | NUTTIDA | NAPAKAS | 1 |
| 2 | 30099998 | นาย | SARANPOJ | SUKANNHAKETU | 1 |
| 3 | 30113141 | นางสาว | THANATORN | KEWLA | 1 |
| 4 | 30332446 | นาง | WIPADA | KAETHONG | 1 |
| 5 | 30333345 | นาย | PUMIPAT | THAIJIA | 1 |
| 6 | 30448342 | นาง | SARANNUCH | TRONGTOSAK | 1 |
| 7 | 30912400 | นาย | PATARAPONG | JOMKHAMSRI | 1 |
| 8 | 30932131 | นาย | VISARUT | MARGUN | 1 |
| 9 | 30948440 | นาง | RUNGSIYMUN | PERMPOOL | 1 |
| 10 | 30993355 | นาย | NAPHAT | HONGHAN | 1 |

Description

- 9) Medicines that make the most money

SQL

```
WITH Z (MED_CODE, SUM_PRICE) AS (
    SELECT MED_CODE, SUM(PRICE)
    FROM MEDICAL_ORDER_ENTRY
    GROUP BY MED_CODE
)
SELECT Z.MED_CODE, M.MED_NAME, M.MED_GENERAL_NAME, Z.SUM_PRICE
FROM Z
INNER JOIN MEDICINE M
ON M.MED_CODE = Z.MED_CODE
WHERE Z.SUM_PRICE = (SELECT MAX(SUM_PRICE) FROM Z)
```

Result

| | MED_CODE | MED_NAME | MED_GENERAL... | SUM_PRICE |
|---|----------|----------|----------------|-----------|
| 1 | 110102 | Benadryl | VIVELLE | 5295.00 |

Description

- 10) List of doctors who have treated the most number of patients

SQL

```
WITH Z (EMP_ID, COUNT_PAT) AS (
    SELECT EMP_ID, COUNT(PAT_ID)
    FROM VISIT
    GROUP BY EMP_ID
)
SELECT Z.COUNT_PAT, Z.EMP_ID, e.NAME_TITLE, e.FNAME_EN,
e.LNAME_EN
FROM Z
INNER JOIN EMPLOYEE e
ON e.EMP_ID = Z.EMP_ID
WHERE Z.COUNT_PAT = (SELECT MAX(COUNT_PAT) FROM Z)
```

Result

| | COUNT_PAT | EMP_ID | NAME_TITLE | FNAME_EN | LNAME_EN |
|----|-----------|----------|------------|------------|--------------|
| 1 | 1 | 30001143 | นาง | NUTTIDA | NAPAKAS |
| 2 | 1 | 30099998 | นาย | SARANPOJ | SUKANNHAKETU |
| 3 | 1 | 30113141 | นางสาว | THANATORN | KAEWLA |
| 4 | 1 | 30332446 | นาง | WIPADA | KAEWTHONG |
| 5 | 1 | 30333345 | นาย | PUMIPAT | THAIJIA |
| 6 | 1 | 30448342 | นาง | SARANNUCH | TRONGTOSAK |
| 7 | 1 | 30912400 | นาย | PATARAPONG | JOMKHAMSRI |
| 8 | 1 | 30932131 | นาย | VISARUT | MARGUN |
| 9 | 1 | 30948440 | นาง | RUNGSIMUN | PERMPOOL |
| 10 | 1 | 30993355 | นาย | NAPHAT | HONGHAN |

Bibliography

- Ramkhamhaeng Hospital. History, Vision. from <https://www.ram-hosp.co.th/aboutus>
- Ramkhamhaeng Hospital. Doctor Appointment. from https://www.ramhosp.co.th/doctorsetdate_step1
- Ramkhamhaeng Hospital. Medical Checkup Programs. from <https://www.ramhosp.co.th/readpackage/10>
- Ramkhamhaeng Hospital. Form of Doctor. from <https://www.ram-hosp.co.th/searchdoctor1>