**Project Dentist Booking System**

**(GROUP 23)**

**Database System [Section 51-54 & 61] 2110322(2024/2)**

**สมาชิก**

1. 6733154321 นาย ปุญญพัฒน์ ผิวขำ
2. 5
3. 5
4. 5
5. 5
6. 5

**Functional requirements**

1. The system shall allow a user to register by specifying the name, telephone number, email, and password.
2. After registration, the user becomes a registered user, and the system shall allow the user to log in to use the system by specifying the email and password. The system shall allow a registered user to log out.
3. After login, the system shall allow the registered user to book **only ONE session** by specifying the date and the preferred dentist. The dentist list is also provided to the user. A dentist information includes the dentist’s name, years of experience, and area of expertise.
4. The system shall allow the registered user to view his booking.
5. The system shall allow the registered user to edit his booking.
6. The system shall allow the registered user to delete his booking
7. The system shall allow the admin to view any bookings.
8. The system shall allow the admin to edit any bookings.
9. The system shall allow the admin to delete any bookings.

**Situation**

A dentist shop wants to have a booking system in the form of a web-based application that, when logging in as an administrator, the administrator can add, update, and delete a dentist shop’s overall information, shop location, available services, phone numbers, map, shop’s facilities, e.g., parking, price range, dentist’s information, e.g., name, years of experience, area of expertise, available time slots, etc. An administrator can add dentists as many as needed.

Via the web-based application, without login, a user can browse and search for a dentist and available time slot in the shop with the dentist’s name, different area of expertise, years of experience. However, if a user wants to book a dentist, the user must first register and log in. A user can book an appointment for at most one time slot, and the user can edit, update, and cancel the booking seven days before the booking date.

**ER Diagram (Chen’s notation)**

A diagram of a company

Description automatically generated

**Schema diagram (show Referential Integrity)**

A computer screen shot of a computer program

Description automatically generated

**SQL commands (follow requirements functional)**

**The system shall allow a user to register by specifying the name, telephone number, email, and password.** เราใช้ตัวของ INSERT INTO user\_account ด้วยข้อมูลต่างๆรวมถึง ชื่อ, เบอร์โทร, อีเมล และ รหัสผ่าน

จากนั้นจึงแยกว่า user คนนั้นเป็นผู้มาใช้บริการ(patient) หรือ หมอฟัน(dentist) เป็นการบอกถึง user ที่ registered เข้ามา

ถ้าเป็น patient -> INSERT INTO patient\_account

ถ้าเป็น dentist -> INSERT INTO dentist\_account



ถ้าเป็น dentist จำเป็นต้อง INSERT INTO dentist\_education เป็นข้อมูลเพิ่มเติม

**A screen shot of a computer

Description automatically generated**

**A screenshot of a computer

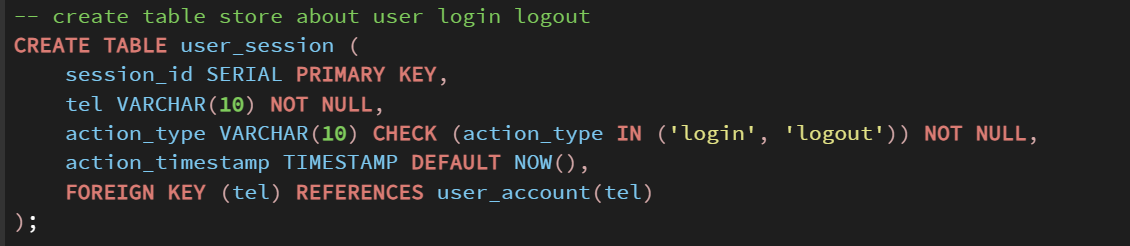
Description automatically generated**

(ตัวอย่าง value ที่ insert เข้าไปใน user\_account Table)

**After registration, the user becomes a registered user, and the system shall allow the user to log in to use the system by specifying the email and password. The system shall allow a registered user to log out.**

user ที่ registered สามารถที่จะ login หรือ logout ได้ โดยใช้ email และ password

สร้าง table user\_session เพื่อเก็บข้อมูลการ login logout ของ user



เขียน stored function 2 ตัว คือ user\_login และ user\_logout

โดย user\_login ทำการค้นหา tel จาก input email ที่เข้ามา เพื่อทำการเช็คใน user\_account ว่ามี tel เป็น null หรือไม่ ถ้าไม่ null ก็จะเช็ค password ของ tel นั้นว่าตรงกับ password ที่ input เข้ามา

ถ้าตรงทุกเงื่อนไข จะทำการ INSERT INTO user\_session table

ส่วนการ logout จะใช้เพียงแค่ email ในการตรวจสอบเพียงอย่างเดียว

A screen shot of a computer

Description automatically generatedA screen shot of a computer program

Description automatically generated

**ตัวอย่างการใช้งาน**

A screen shot of a computer

Description automatically generated

**After login, the system shall allow the registered user to book only ONE session by specifying the date and the preferred dentist. The dentist list is also provided to the user. A dentist information includes the dentist’s name, years of experience, and area of expertise.**

หลังจาก login นั้น user ที่เป็น patient สามารถจองคิวหมอฟัน ได้แค่ 1 booking โดยระบุ date และ dentist ที่ต้องการ โดยสามารถ view dentist list พร้อมกับข้อมูลของหมอฟันคนนั้นๆ

**เขียน stored function ชื่อ**

**get\_available\_dentists( p\_patient\_tel VARCHAR, p\_service\_name VARCHAR, p\_booking\_date DATE )**

ทำหน้าที่ show list ของ dentist ที่สามารถให้บริการ p\_service\_name ได้ โดยระบุวันที่ต้องการจองลงไป

A screenshot of a computer program

Description automatically generated

**ตัวอย่างการใช้งาน**

**A screenshot of a computer

Description automatically generated**

จากนั้นเขียน stored procedure สำหรับ booking รับค่าต่างๆที่ใช้สำหรับการ booking ลงไป

**patient\_booking( p\_patient\_name VARCHAR,**

**p\_patient\_tel VARCHAR,**

**p\_dentist\_name VARCHAR,**

**p\_dentist\_tel VARCHAR,**

**p\_service\_name VARCHAR,**

**p\_booking\_date DATE,**

**p\_clinic\_id INTEGER,**

**p\_status booking\_status DEFAULT 'confirmed' )**

stored procedure นี้จะทำการเช็คว่า patient คนนั้นจองไปหรือยังภายในวันนั้น ถ้าหากวันนั้น patient ทำการจองไปแล้ว จะไม่สามารถจองเพิ่มได้อีก

ถ้าหากเวลาที่จองไปใน slot นั้น dentist มีเวลา start\_time หรือ end\_time เป็น null แปลว่า dentist ไม่ว่างสำหรับขณะนั้น

ถ้าหาการจองราบลื่น ระบบจะทำการเพิ่ม patient คนนั้นไปในระบบ booking

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

***\*หมายเหตุ สำหรับ time\_range ใน booking เรานำ start\_time ของ dentist ที่ dentist\_tel มาใช้ สื่อว่าควรเริ่มมาตาม bookin ณ เวลานี้***

**The system shall allow the registered user to view his booking**

สร้าง view สำหรับการจองของ user คนนั้นๆ

เขียน stored function สำหรับ returns table เป็นตารางที่ user ที่ระบุไปใน parameter จะสามารถ view ดูข้อมูลต่างๆได้

**สร้าง stored function returns table ที่ query ข้อมูล booking ของ p\_user\_tel**

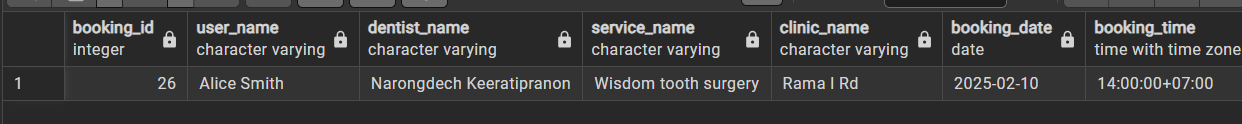
A computer screen shot of a program

Description automatically generated

**สร้าง view ขึ้นมาจาก stored function ที่สร้างไว้**

**A screen shot of a computer

Description automatically generated**



**The system shall allow the registered user to edit his booking.**

User สามารถแก้ไขการ booking ได้

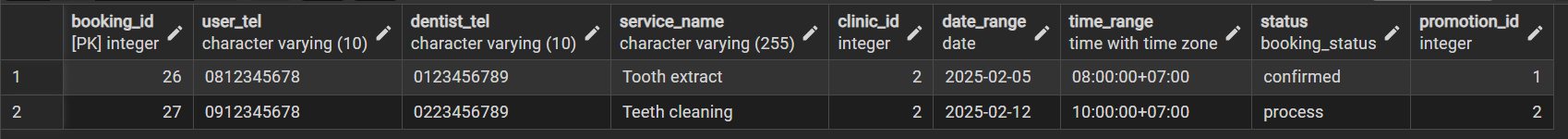
ทำ stored procedure สำหรับ edit โดยเช็คว่า booking มี booking ของ user คนนั้นหรือไม่ และ user สามารถเลือก dentist คนใหม่โดยที่เช็คว่า dentist คนนั้นว่างหรือไม่ด้วย

A screenshot of a computer program

Description automatically generated

**ผลที่ได้จากการใช้งาน**

**A screen shot of a computer program

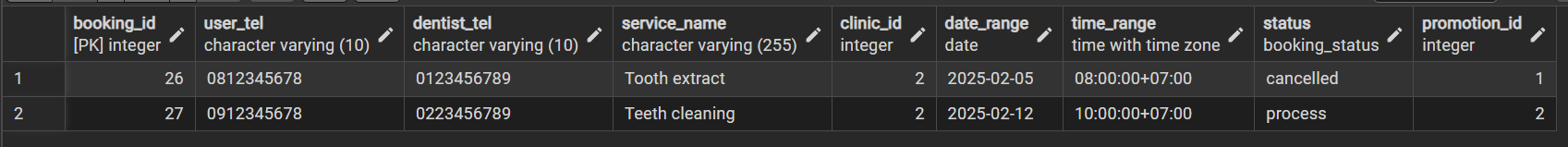
Description automatically generated** ****

**The system shall allow the registered user to delete his booking**

User สามารถที่จะลบ booking ของตนเองได้ แต่เนื่องจากเราอาจเก็บข้อมูลของ user ไว้ เราจึงใช้การปรับ status booking เป็น “cancel” แทน

A screen shot of a computer program

Description automatically generated



* **The system shall allow the admin to view any bookings.**
* **The system shall allow the admin to edit any bookings.**
* **The system shall allow the admin to delete any bookings.**

ในส่วนของ admin จะมีความคล้ายกับ SQL command ที่ผ่านมา **โดยที่ admin สามารถทำการ view , edit และ delete booking ใดๆก็ได้** (ในขณะที่ user ได้แค่ view , edit ,delete booking ของตัวเองเท่านั้น )

ทั้งนี้ขึ้นกับสถานการณ์ต่างๆ สมมติว่า admin เป็นผู้ดูแล database dentist booking system ด้วยตัวเอง ก็สามารถแก้ไขข้อมูลใน database postgresql pgAdmin4 โดยง่ายผ่าน command UPDATE , DELETE etc.

**แต่ถ้าหากมี admin หลายคน เราอาจต้องเขียน SQL command เพิ่มเติมสำหรับการใช้งาน**

สร้าง table สำหรับเก็บ admin account และ insert ข้อมูลเข้าไป (admin ถือว่าเป็น user คนนึง)

A screen shot of a computer code

Description automatically generated

สร้าง admin\_booking\_log table มาเก็บประวัติการแก้ไขข้อมูลของ admin แต่ละคน

(ตาม requirement ของอาจารย์ให้ทำ log เก็บประวัติการแก้ไขของ admin ฉะนั้นจะไม่มี log ของ user ใน projectนี้)

A computer code on a black background

Description automatically generated

สร้าง stored function ที่ track action ของ admin เวลาแก้ไข booking ต่างๆ และนำไป Bind กับ Trigger

โดยมี Trigger สำหรับ Edit , Delete และ Add new booking

A computer screen shot of text

Description automatically generated

A screen shot of a computer program

Description automatically generated

เขียน stored function สำหรับ return table ที่สามารถนำมาทำเป็น view booking ของ admin ได้

A screen shot of a computer program

Description automatically generated

ทำการสร้าง ROLE ขึ้นมาเพื่อสามารถ GRANT (มอบสิทธิ์) คำสั่ง INSERT , UPDATE , DELETE บน booking table แก่ ROLE ที่สร้างขึ้นมา , ในกรณีนี้ให้ admin\_role สามารถ INSERT UPDATE DELETE บน booking ได้เท่านั้น

แต่ PostgreSQL ไม่สามารถมอบ role ที่แก่ value ใดๆที่ขึ้นต้นด้วยตัวเลข ,มีอักขระพิเศษ หรือ ช่องว่างได้ (กรณีนี้มี admin\_account ใช้ PK เป็นเบอร์โทร) และในกรณีนี้มี admin 3 คน ดังนั้นมอบ role admin ให้พร้อมกับรหัสประจำตัวของแต่ละคน

ใน project นี้ เราจะสร้าง role admin ใน format admin\_<admin\_tel> เช่น admin\_0123456789 เป็นต้น

เราใช้ EXECUTE FORMAT(‘’) ในการ execute คำสั่งในรูปแบบของ TEXT

เช็คว่าใน pg\_roles system table ของ PostgreSQL ว่ามี role ชื่อ ‘admin\_role’ หรือเปล่า ถ้าไม่มีให้สร้างขึ้นมา

A black screen with text

Description automatically generated

ทำการมอบสิทธิ์ SELECT UPDATE DELETE บน table booking แก่ admin\_role และให้ admin\_role สามารถใช้ USAGE ใน booking table ได้ โดยอิงจากการเรียงลำดับของ booking\_id ที่เป็น auto increment

A black screen with colorful text

Description automatically generated

ทำการ CREATE ROLE ในรูปแบบของ admin\_<admin\_tel> และมอบสิทธิ์ที่ admin\_role มีให้

A screenshot of a computer program

Description automatically generated

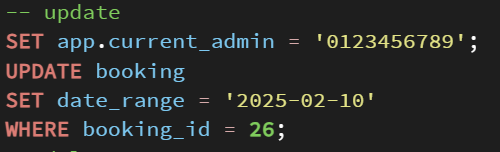
สุดท้ายสร้าง view ของ admin โดยนำมาจาก stored function ที่ query view ไว้

A black screen with text

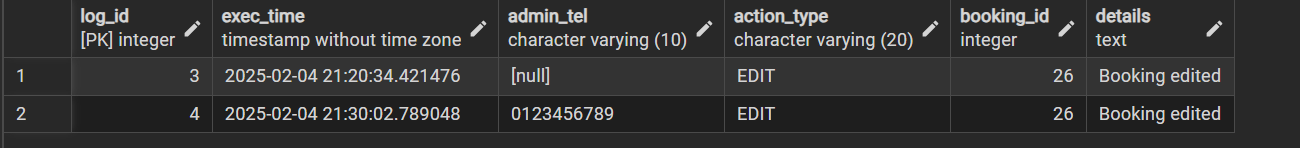
Description automatically generated

**ตัวอย่างคำสั่ง**

**ไม่ได้ระบุใน command ว่า admin คนไหนเป็นคนแก้ไข เพราะเราไมได้ set current admin ไว้ใน Trigger function log\_admin\_booking\_action() ฉะนั้นจึงเรียก SET app.current\_admin ก่อน**

****

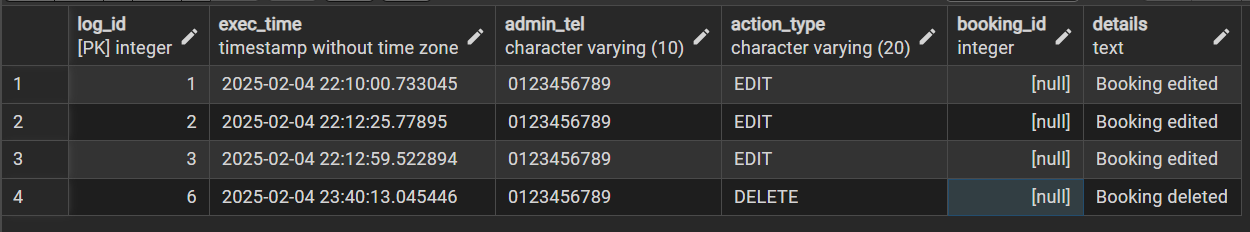
**1.UPDATE**

****

**2.DELETE** เนื่องจาก key ของ admin\_booking\_id reference มาจาก booking , เราไม่สามารถลบได้เพราะอาจตาราง admin\_booking\_id ใช้ค่า booking\_id อยู่ , ดังนั้นเราจึงให้ตารางนี้เมื่อค่าถูกลบออกไป ให้set booking\_id เท่ากับ null ***(ลบทุกๆbooking ที่มี booking\_id = 28)***

**A screenshot of a computer

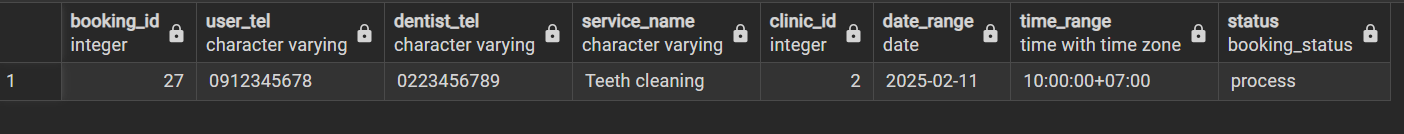
Description automatically generated**

****

**3.View booking (booking\_id = 26 , 28 ถูกลบไปแล้วจึงไม่มี)**

**A close up of a computer screen

Description automatically generated**

****

**SQL complex query**

1.แสดงชื่อ-นามสกุล เงินเดือน ของหมอฟันที่มีเงินเดือนมากกว่าค่าเฉลี่ยของหมอฟันทุกคนในคลินิกสาขาตัวเอง

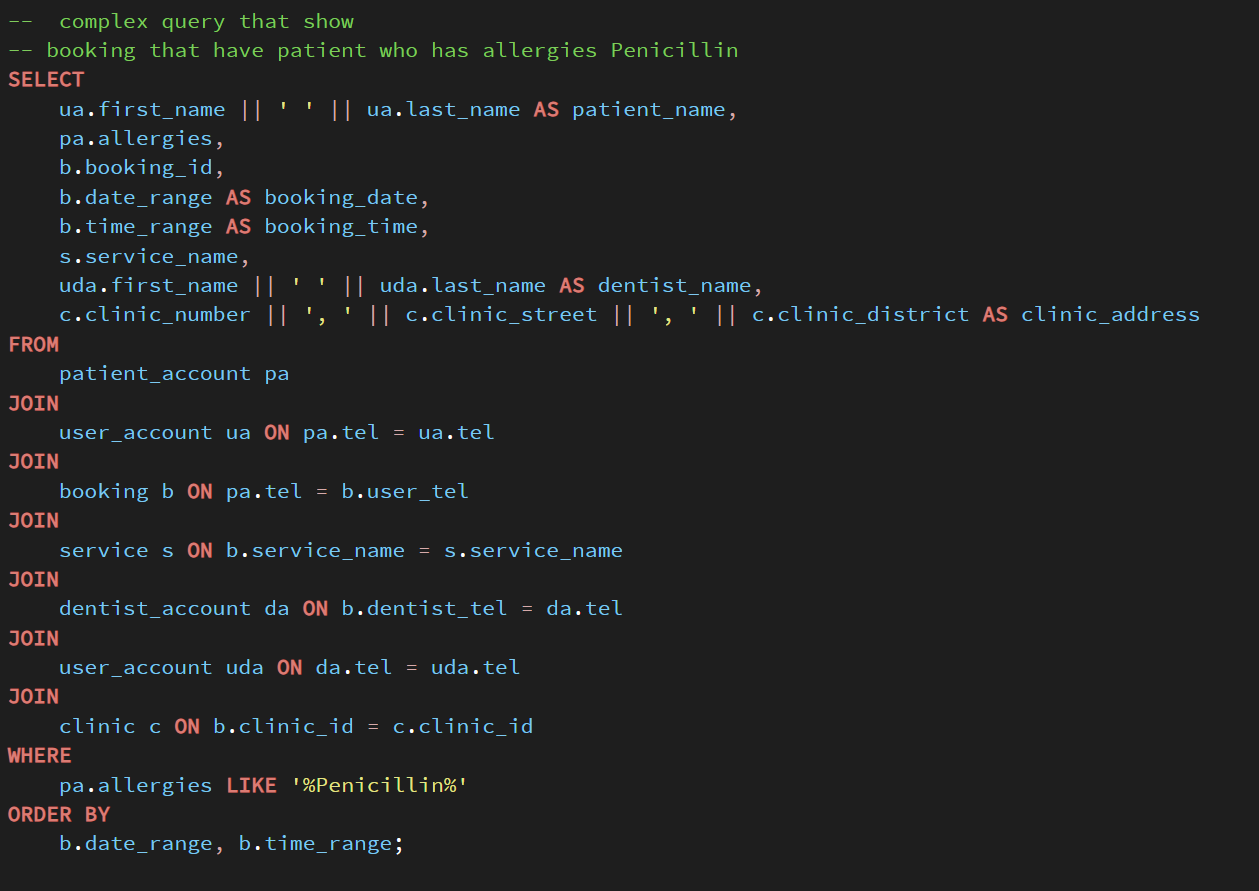
A computer screen with text

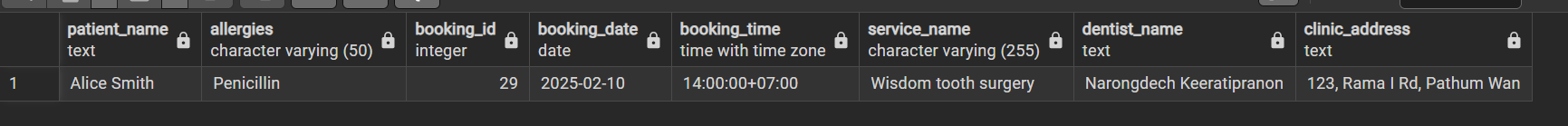
Description automatically generated

A screenshot of a computer

Description automatically generated

2.แสดงรายชื่อของ patient ที่ booking และมีอาการแพ้ของยา Penicillin





**Document-based design schema 1 collection**

สำหรับ document-based design schema เราจะยกตัวอย่าง collection clinit ที่ relation กับ dentist\_account ดังนี้

* Clinic กับ dentist มีความสัมพันธ์แบบ 1 : M
* Dentist จำเป็นต้องอยู่ใน clinic ในขณะที่ clinic อาจมี dentist บางคนหรือไม่มีก็ได้
* Clinic ประกอบด้วย dentist จึงมี field ชื่อ dentist\_in\_clinic เก็บเป็น array ของ dentist แต่ละคนที่ทำงานใน clinic นี้

A screenshot of a computer

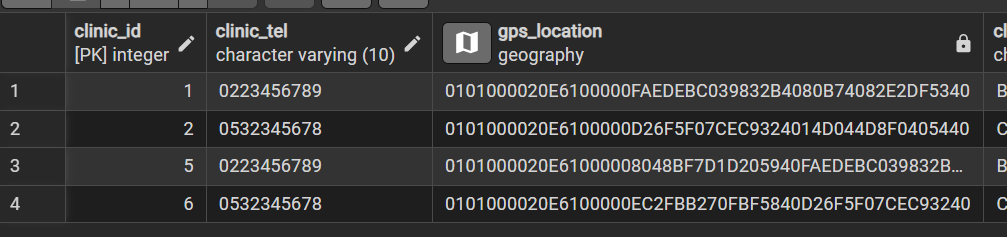
Description automatically generatedA screen shot of a computer program

Description automatically generated

**เพิ่มเติม**

Database นี้มีการใช้ postgis Extension สำหรับหา Coordinate บนโลก สำหรับพิกัด Clinic ที่มีอยู่

**สามารถเข้าไปดูไฟล์ได้ที่ https://github.com/Pulaphew/Dentist\_system\_postgresql.git**



A screenshot of a computer

Description automatically generated

**ER-Diagram From Pgadmin4**

A screenshot of a computer

Description automatically generated