



รายงานเรื่อง Oral Health Record System เว็บไซค์ทันตกรรมคนโภ

จัดทำโดย

นางสาวณุดา แสงอินทร์ 6630202163

นางสาวชนิดาภา อุดมรัชтанนท์ 6630202147

นางสาวณิชา กันหาสินธุ 6630202228

นายปิยะณัฐ ไนมุกข์ 6630202520

นายปิยภัทร คงเจริญวุฒิ 6630202503

เสนอ

ผศ.ดร. สุพาร บรรดาศักดิ์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา ระบบฐานข้อมูลและปฏิบัติการเบื้องต้น
ภาคเรียนที่ 1 ปีการศึกษา 2568

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา

บทนำ

ในยุคดิจิทัลที่เทคโนโลยีสารสนเทศเป็นกลไกขับเคลื่อนสำคัญของการดำเนินงานทุกภาคส่วน, การบริหารจัดการข้อมูลทางการแพทย์ในคลินิกทันตกรรมมีความจำเป็นต้องปรับปรุงให้มีความทันสมัยและมีประสิทธิภาพสูงสุด. การจัดการกับปริมาณข้อมูลที่เพิ่มขึ้นอย่างต่อเนื่อง ไม่ว่าจะเป็นข้อมูลส่วนบุคคลของคนไข้, การบันทึกรายละเอียดการนัดหมายที่แม่นยำ, การจัดการข้อมูลทันตแพทย์และผู้เชี่ยวชาญเฉพาะทาง, ตลอดจนการจัดเก็บรายการหัตถการและราคาและประวัติการรักษาที่ผ่านมา ล้วนเป็นงานหลักที่มีความซับซ้อนและใช้เวลามาก

รายงานฉบับนี้จัดทำขึ้นเพื่อนำเสนอการพัฒนาระบบสนับสนุนงานหลังบ้าน สำหรับทันตแพทย์ โดยเฉพาะ ซึ่งสามารถเข้าถึงและใช้งานผ่านแพลตฟอร์มเว็บไซต์ ระบบนี้ได้รับการออกแบบมาเพื่อทำหน้าที่เป็นศูนย์กลางในการจัดเก็บ จัดการ และเชื่อมโยงข้อมูลสำคัญทางการทันตกรรมทั้งหมดอย่างเป็นระบบระเบียบ

สารบัญ

ขอบเขตการทํางาน.....	1
Data Flow Diagram	2
Data list.....	4
Entity-Relationship Diagram	6
Data Dictionary.....	9
หน้าจอโปรแกรม.....	12
Database Diagram	16
Design Edit Table.....	18
View.....	20
Function ,Store Procedure	22
Trigger	28

ขอบเขตการทำงาน

1. ขอบเขตด้านการทำงาน

ระบบจะครอบคลุมฟังก์ชันหลักที่จำเป็นต่อการบริหารจัดการคลินิกทันตกรรมดังนี้

- 1.1 การเข้าสู่ระบบ: ผู้ใช้สามารถเข้าสู่ระบบเพื่อเข้าถึงฟังก์ชันการจัดการข้อมูลได้
- 1.2 การจัดการข้อมูลคนไข้: เจ้าหน้าที่สามารถดำเนินการเพิ่ม / แก้ไข / ลบ ข้อมูลส่วนตัวของคนไข้ได้
- 1.3 การจัดการข้อมูลทันตแพทย์: เจ้าหน้าที่สามารถดำเนินการจัดการข้อมูลของทันตแพทย์ได้
- 1.4 การจัดการข้อมูลการนัดหมาย: เจ้าหน้าที่สามารถบันทึกและแก้ไข ข้อมูลการนัดหมายได้
- 1.5 การค้นหาการนัดหมาย: เจ้าหน้าที่สามารถค้นหาคิวการนัดหมายตามชื่อคนไข้ได้
- 1.6 การแสดงรายงานสรุป: ระบบสามารถแสดงรายงานสรุปจำนวนคิวการนัดหมาย ในแต่ละวันได้

2. ขอบเขตด้านข้อมูล

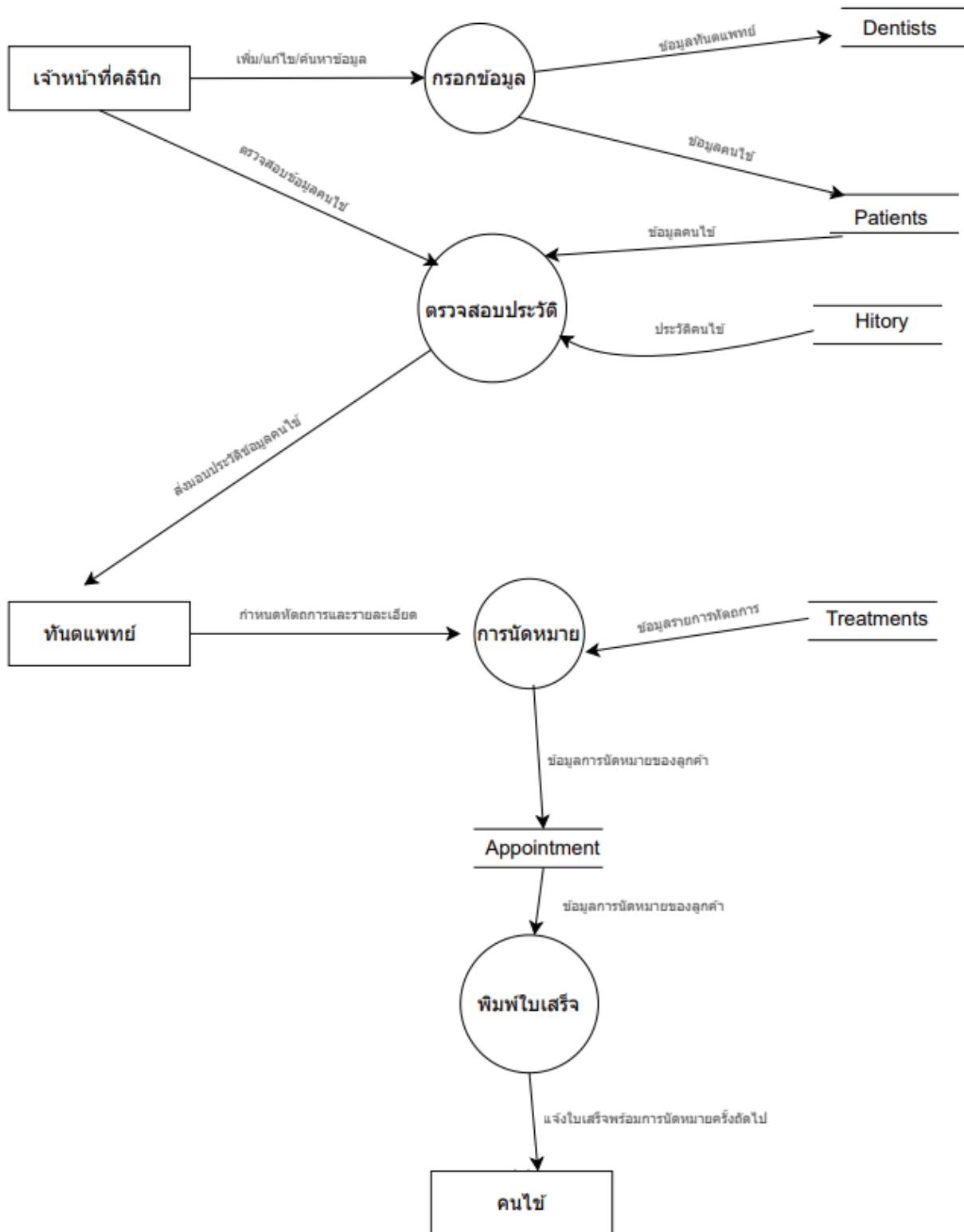
รายงานจะนำเสนอโครงสร้างข้อมูลหลักที่สนับสนุนฟังก์ชันการทำงานข้างต้น โดยแบ่งเป็น 5 หมวดหมู่:

- 2.1 ข้อมูลคนไข้: ประกอบด้วย รหัสคนไข้, ชื่อ, นามสกุล, วันเดือนปีเกิด, และเบอร์มือถือ
- 2.2 ข้อมูลทันตแพทย์: ประกอบด้วย รหัสทันตแพทย์, ชื่อ, นามสกุล, และความเชี่ยวชาญ
- 2.3 ข้อมูลการนัดหมาย: ประกอบด้วย รหัสการนัดหมาย, รหัสคนไข้, รหัสทันตแพทย์, วันที่นัดหมาย, เหตุผล, และสถานะการรักษา
- 2.4 ข้อมูลหัตถการ: ประกอบด้วย รหัสหัตถการ (PK), ชื่อหัตถการ, และราคา
- 2.5 ข้อมูลประวัติการรักษา: ประกอบด้วย รหัสการบันทึก (PK), รหัสคนไข้, รหัสทันตแพทย์, รหัสหัตถการ, และวันเวลาที่ลงบันทึก

3. ขอบเขตด้านเทคนิคและข้อจำกัด

- 3.1 แพลตฟอร์ม: ระบบนี้ถูกพัฒนาให้สามารถเข้าถึงและจัดการข้อมูลผ่านเว็บไซต์เท่านั้น
- 3.2 ข้อจำกัด: รายงานนี้จะไม่ครอบคลุมถึงระบบการจัดการการเงินที่ซับซ้อน (นอกเหนือจากการบันทึกรายการหัตถการ) และจะไม่ครอบคลุมการเชื่อมต่อกับระบบภายนอกได้ ๆ

Data Flow Diagram



คำอธิบาย Data Flow Diagram

แผนภาพกระแสข้อมูล (Data Flow Diagram: DFD) ฉบับนี้แสดงถึงกระบวนการทำงานภายในระบบจัดการคลินิกทันตกรรม ซึ่งแสดงให้เห็นถึงการไหลของข้อมูลระหว่างบุคลากรที่เกี่ยวข้อง ได้แก่ เจ้าหน้าที่คลินิก ทันตแพทย์ และผู้ป่วย ตลอดจนการเชื่อมโยงของข้อมูลระหว่างกระบวนการต่าง ๆ ภายในระบบ เพื่อให้สามารถจัดเก็บและเรียกใช้งานข้อมูลได้อย่างมีประสิทธิภาพ

กระบวนการเริ่มต้นจาก เจ้าหน้าที่คลินิก ซึ่งเป็นผู้รับผิดชอบในการกรอกข้อมูลพื้นฐานของผู้ป่วยและข้อมูลของทันตแพทย์เข้าสู่ระบบ โดยเจ้าหน้าที่สามารถดำเนินการเพิ่ม แก้ไข หรือค้นหาข้อมูลได้ เพื่อให้ฐานข้อมูลของคลินิกมีความครบถ้วนและเป็นปัจจุบัน ข้อมูลที่จัดเก็บไว้ประกอบด้วยข้อมูลส่วนบุคคลของผู้ป่วย ข้อมูลประวัติการรักษา และข้อมูลการนัดหมาย

ในส่วนของ ทันตแพทย์ มีหน้าที่ตรวจสอบประวัติการรักษาของผู้ป่วยผ่านระบบ เพื่อใช้ประกอบการวินิจฉัยและดำเนินการรักษา ทั้งนี้ ทันตแพทย์จะเป็นผู้บันทึกข้อมูลการรักษา รายการหัตถการ และรายละเอียดที่เกี่ยวข้องลงในระบบ เพื่อให้ข้อมูลเหล่านี้ถูกจัดเก็บไว้ในฐานข้อมูลกลาง ซึ่งสามารถเรียกดูข้อมูลนี้ได้เมื่อต้องการ

ส่วนของ ผู้ป่วย จะมีข้อมูลที่ถูกจัดเก็บในระบบ ได้แก่ ประวัติส่วนบุคคล ข้อมูลการรักษา และข้อมูลการนัดหมาย โดยเมื่อมีการนัดหมายใหม่ ระบบจะบันทึกข้อมูลการนัดหมายไว้ในส่วนของฐานข้อมูลการนัดหมาย (Appointment Data) เพื่อใช้ในการจัดตารางเวลาและการติดตามผลการรักษาในครั้งต่อไป เมื่อกระบวนการรักษาเสร็จสิ้น ระบบจะทำการ พิมพ์ใบเสร็จรับเงิน พร้อมทั้งแจ้ง การนัดหมายครั้งถัดไป ให้แก่ผู้ป่วยโดยอัตโนมัติ ข้อมูลทั้งหมดที่เกี่ยวข้องกับผู้ป่วยและการรักษาจะถูกเชื่อมโยงกันภายในระบบ เพื่อให้เจ้าหน้าที่และทันตแพทย์สามารถเข้าถึงข้อมูลได้อย่างรวดเร็วและถูกต้อง

โดยสรุป แผนภาพนี้สะท้อนให้เห็นถึงการทำงานร่วมกันระหว่างบุคลากรในคลินิกและระบบสารสนเทศ เพื่อสนับสนุนการจัดการข้อมูลผู้ป่วย การดำเนินการรักษา และการบริหารจัดการการนัดหมาย ให้เป็นไปอย่างเป็นระบบ ลดความซ้ำซ้อนของข้อมูล และเพิ่มประสิทธิภาพในการให้บริการของคลินิกทันตกรรมโดยรวม

Data list

ลำดับและชื่อตาราง	รายละเอียดของข้อมูล	ตัวอย่างข้อมูล	ความสัมพันธ์กับตารางอื่น
1. Patients	ให้เก็บข้อมูลของคนไข้ เช่น ชื่อ-นามสกุล วัน/เดือน/ปี - เบอร์โทรศัพท์ เพื่อใช้ในการนัดหมาย	P001, Yda San, 2004-09-12, 08456789	เชื่อมกับ Appointments, History
2. Dentists	ให้เก็บข้อมูลของทันตแพทย์ เช่น ชื่อ นามสกุล ความเชี่ยวชาญ	D001, Dr.ณิชา คนเกี้ย, Orthodontics	เชื่อมกับ Appointments, History
3. Appointments	ให้บันทึกข้อมูลการนัดหมายระหว่างคนไข้ กับทันตแพทย์ เช่น วันที่ เวลา เหตุผล และสถานะการรักษา	A001, P001, D001, 2025/09/18 10:00, "Loose bracket", "Completed"	เชื่อมกับ Patients, Dentists
4. Treatments	ให้เก็บรายการหัตถการทางทันตกรรมที่มีในคลินิก เช่น ชื่อหัตถการและราคา	T014, "Orthodontic Retainer", 1500	เชื่อมกับ History
5. History	ให้บันทึกประวัติการรักษาของคนไข้ โดยเก็บข้อมูลคนไข้ ทันตแพทย์ หัตถการ และวันที่ทำการรักษา	R001, P001, D001, T014, 2025/09/18 11:32	เชื่อมกับ Patients, Dentists, Treatments
6. HistoryDetails	ให้เก็บรายละเอียดของหัตถการที่ทำในแต่ละครั้งของประวัติการรักษา เช่น หัตถการใดบ้าง จำนวนกี่ครั้ง	H001, T014, 1 "ญา ทำ Orthodontic Retainer 1 ครั้ง"	เชื่อมกับ History และ Treatments

คำอธิบายรายละเอียดตาราง

- Patients:** เอนทิตี้ที่เก็บข้อมูลประชากรหลักของผู้ป่วย ซึ่งเป็นเอนทิตี้ตั้งต้นที่สำคัญที่สุดของระบบ ข้อมูลประกอบด้วยรหัสผู้ป่วย ชื่อ-นามสกุล วันเดือนปีเกิด และเบอร์โทรศัพท์ เพื่อใช้ในการอ้างอิงและจัดการการนัดหมาย
ความสัมพันธ์: เชื่อมโยงกับตาราง Appointments และ History
- Dentists:** เอนทิตี้ที่เก็บข้อมูลคุณภาพทางการแพทย์ (ทันตแพทย์) โดยมีข้อมูลสำคัญ เช่น รหัสทันตแพทย์ ชื่อ-นามสกุล และความเชี่ยวชาญ
ความสัมพันธ์: เชื่อมโยงกับตาราง Appointments และ History
- Appointments:** เป็นตารางธุรกรรม (Transactional Table) ที่ใช้บันทึกการนัดหมายระหว่างผู้ป่วยและทันตแพทย์ ข้อมูลที่สำคัญคือ รหัสการนัดหมาย รหัสผู้ป่วย (P001) รหัสทันตแพทย์ (D001) วันที่ เวลา เหตุผล และสถานะของการรักษา (เช่น "Completed")

ความสัมพันธ์: เชื่อมโยงกับตาราง Patients และ Dentists

4. **Treatments:** เป็นตารางข้างอิง (Lookup Table) ที่เก็บรายการหัตถการทางทันตกรรมที่พร้อมให้บริการ ในคลินิก ข้อมูลประกอบด้วยรหัสหัตถการ ชื่อหัตถการ และราคา (เช่น "Orthodontic Retainer", 1500)
- ความสัมพันธ์:** เชื่อมโยงกับตาราง History

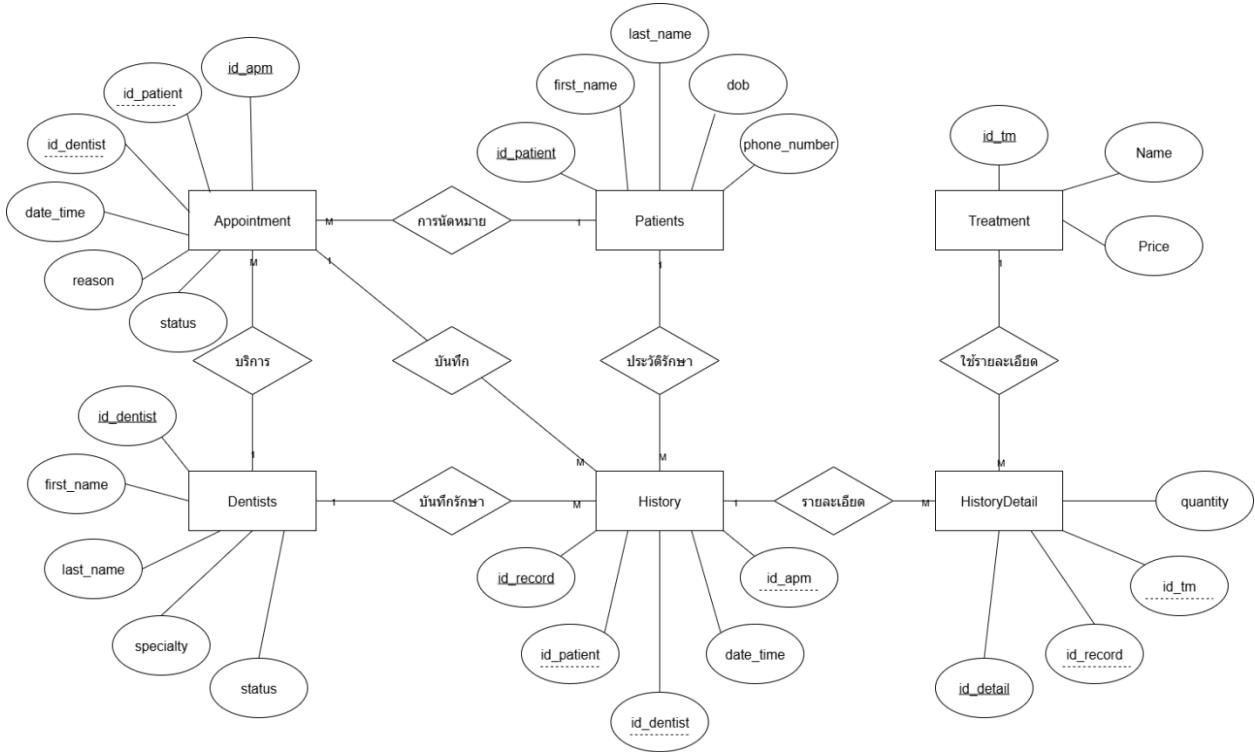
5. **History:** เป็นตารางธุรกรรมหลักที่ใช้บันทึกประวัติการรักษาของผู้ป่วย ข้อมูลจะบันทึกว่าผู้ป่วยรายใด (P001) ได้รับการรักษาจากทันตแพทย์ท่านใด (D001) สำหรับหัตถการใด (T014) และวันเวลาที่ทำการรักษา ซึ่งตารางนี้ทำหน้าที่เป็นศูนย์กลางการเชื่อมโยงข้อมูลระหว่างผู้ป่วย ทันตแพทย์ และหัตถการที่ระบุในตาราง Treatments

ความสัมพันธ์: เชื่อมโยงกับตาราง Patients, Dentists และ Treatments

6. **HistoryDetails:** เป็นตารางรายละเอียดของประวัติการรักษา (History) โดยบันทึกรายละเอียดของหัตถการที่ทำในแต่ละครั้งของประวัติการรักษา (เช่น บันทึกว่ามีการทำหัตถการ T014 จำนวน 1 ครั้ง) ซึ่งให้รายละเอียดปลีกย่อยของรายการที่อยู่ในตาราง History

ความสัมพันธ์: เชื่อมโยงกับตาราง History และ Treatments

Entity-Relationship Diagram



ภาพ ER Diagram

การอธิบายความสัมพันธ์ (Relationships) ใน ER Diagram

ในໂຄໂອແກຣມນີ້ ແຕ່ລະກລ່ວງຄືອ Entity (ຕາຮາງ) ແລະເສັ້ນທີ່ເຊື່ອມຕ່ອກັນຄືອ Relationship (ຄວາມສັນພັນນີ້)

1. Patients (ผู้ป่วย) ↔ Appointments (การนัดหมาย)

- **Relationship:** One-to-Many (1:N)
- **Explanation:** ผู้ป่วย 1 คน สามารถมีໄດ້ຫລາຍການນัดหมาย (Appointments) ແຕ່ການນັດໝາຍ 1 ຄວັງ ຈະມີ
ຜູ້ປ່າຍທີ່ເກີ່ວຂອງໄດ້ເພີ່ງ 1 ດາວກ່າວ

2. Dentists (ทันตแพทย์) ↔ Appointments (การนัดหมาย)

- **Relationship:** One-to-Many (1:N)
- **Explanation:** ທັນຕະພາບ 1 ດາວກ່າວ ສາມາດຮັດໄໝໃບບໍລິການໃນການນັດໝາຍໄດ້ຫລາຍຄັ້ງ ແຕ່ການນັດໝາຍ 1 ຄວັງ
ຈະມີທັນຕະພາບທີ່ຮັບຜົດຂອບໄດ້ເພີ່ງ 1 ດາວກ່າວ

3. Appointments (การนัดหมาย) ↔ History (ประวัติ)

- **Relationship:** One-to-Many (1:N)

- **Explanation:** การนัดหมาย 1 ครั้ง อาจนำไปสู่การบันทึกประวัติการรักษา 1 รายการ (ในตาราง History) แต่การบันทึกประวัติ 1 รายการ จะเชื่อมโยงกลับไปยังการนัดหมายได้เพียง 1 ครั้งเท่านั้น (ดูจากในตาราง History)
 - Note: ในระบบฐานข้อมูลจริง บางทีอาจเป็นการบันทึกประวัติ 1 ครั้งต่อการนัดหมาย 1 ครั้ง คั่นนั้น อาจเป็น Primary Key ในตาราง History ด้วยก็เป็นไปได้ หรือไม่ก็มีประวัติ 1 รายการ เกิดจากการนัดหมาย 1 ครั้ง

4. History (ประวัติ) \leftrightarrow Patients (ผู้ป่วย)

- **Relationship:** One-to-Many (1:N)
- **Explanation:** ผู้ป่วย 1 คน สามารถมีประวัติการรักษาได้หลายรายการ (History records) แต่ประวัติการรักษา 1 รายการ จะเชื่อมโยงกับผู้ป่วยเพียง 1 คนเท่านั้น

5. History (ประวัติ) \leftrightarrow Dentists (ทันตแพทย์)

- **Relationship:** One-to-Many (1:N)
- **Explanation:** ทันตแพทย์ 1 คน สามารถเป็นผู้บันทึกประวัติการรักษาได้หลายรายการ แต่ประวัติการรักษา 1 รายการ ถูกบันทึกโดยทันตแพทย์เพียง 1 คนเท่านั้น

6. History (ประวัติ) \leftrightarrow HistoryDetails (รายละเอียดประวัติ)

Appointments 1 : 1 History ("1:1 นำไปสู่"):

การนัดหมาย 1 ครั้ง จะนำไปสู่การบันทึกประวัติการรักษา 1 รายการ

รายละเอียดการรักษาและค่าใช้จ่าย

History 1 : M HistoryDetails ("1:M มีรายละเอียด"):

ประวัติการรักษา 1 รายการ สามารถมีรายละเอียดหัวใจที่ทำได้หลาย รายการ

Treatments 1 : M HistoryDetails ("1:M ถูกใช้ใน"):

รายการรักษาตามมาตรฐาน 1 รายการ ถูกนำไปใช้ในรายละเอียดประวัติการรักษาได้หลาย รายการ

- **Relationship:** One-to-Many (1:N)
- **Explanation:** ประวัติการรักษา 1 รายการ สามารถประกอบด้วยรายละเอียดการรักษา (HistoryDetails) หลายรายการ (เช่น การทำฟันหลายชิ้น หรือการรักษาหลายขั้นตอน) แต่รายละเอียดการรักษา 1 บรรทัด จะเชื่อมโยงกับประวัติหลักเพียง 1 รายการเท่านั้น

7. HistoryDetails (รายละเอียดประวัติ) \leftrightarrow Treatments (รายการรักษา)

- **Relationship:** One-to-Many (1:N)
- **Explanation:** รายการรักษา 1 รายการ เช่น "อุดฟัน", "ถอนฟัน") สามารถถูกใช้ในรายละเอียดประวัติ (HistoryDetails) ได้หลายครั้ง (โดยผู้ป่วยหลายคน) แต่รายละเอียดประวัติ 1 รายการ จะอ้างอิงถึงรายการรักษาเพียง 1 รายการเท่านั้น

Data Dictionary

ชื่อ Database : Dental

ตารางที่ 1 Patients ข้อมูลคนไข้

PK – id_patient

No.	Column Name	Type	Size	Null	Description	Example
1	id_patient	varchar	50	no	รหัสคนไข้	P001
2	first_name	varchar	50		ชื่อ	Yda
3	last_name	varchar	50		นามสกุล	San
4	dob	DateTime	8	no	วันเดือนปีเกิด	2004/9/12
5	phone_number	varchar	50	no	เบอร์มือถือ	08456789

ตารางที่ 2 Dentists ข้อมูลทันตแพทย์

PK – id_dentist

No.	Column Name	Type	Size	Null	Description	Example
1	id_dentist	varchar	50	no	รหัสทันตแพทย์	D001
2	first_name	varchar	50	no	ชื่อ	Dr.ณิชา
3	last_name	varchar	50		นามสกุล	คงโภค
4	specialty	varchar	50		เฉพาะทาง	Orthodontics
5	status	varchar	50		สถานะ	ว่าง (มีนัดแล้ว)

ตารางที่ 3 Appointments การนัดหมาย

PK – id_apm

FK - id_patient, id_dentist

No.	Column Name	Type	Size	Null	Description	Example
1	id_apm	varchar	50	no	รหัสการนัดหมาย	A001
2	id_patient	varchar	50	no	รหัสคนไข้	P001
3	id_dentist	varchar	50	no	รหัสทันตแพทย์	D001
4	date_time	DateTime	DateTime	no	วันที่นัดหมาย	2025/09/18 10:00:00
5	reason	varchar	100		เหตุผล	Loose bracket
6	status	varchar	50	no	สถานะการรักษา	Completed

ตารางที่ 4 Treatments รายการหัตถการ

PK – id_tm

No.	Column Name	Type	Size	Null	Description	Example
1	id_tm	varchar	50	no	รหัสหัตถการ	T014
2	Name	varchar	50	no	ชื่อหัตถการ	Orthodontic Retainer
3	Price	Int	5		ราคา	1500

ตารางที่ 5 History ประวัติการรักษา

PK – id_record

FK - id_patient, id_dentist

No.	Column Name	Type	Size	Null	Description	Example
1	id_record	varchar	50	no	รหัสการบันทึก	A001
2	id_patient	varchar	50		รหัสคนไข้	P001
3	id_dentist	varchar	50		รหัสพนักงานแพทย์	D001
4	date_time	Datetime	8		วันเวลาที่ลงบันทึก	2025/09/18 11:32:00
5	id_apm	varchar	50	no	รหัสการนัดหมาย	A001

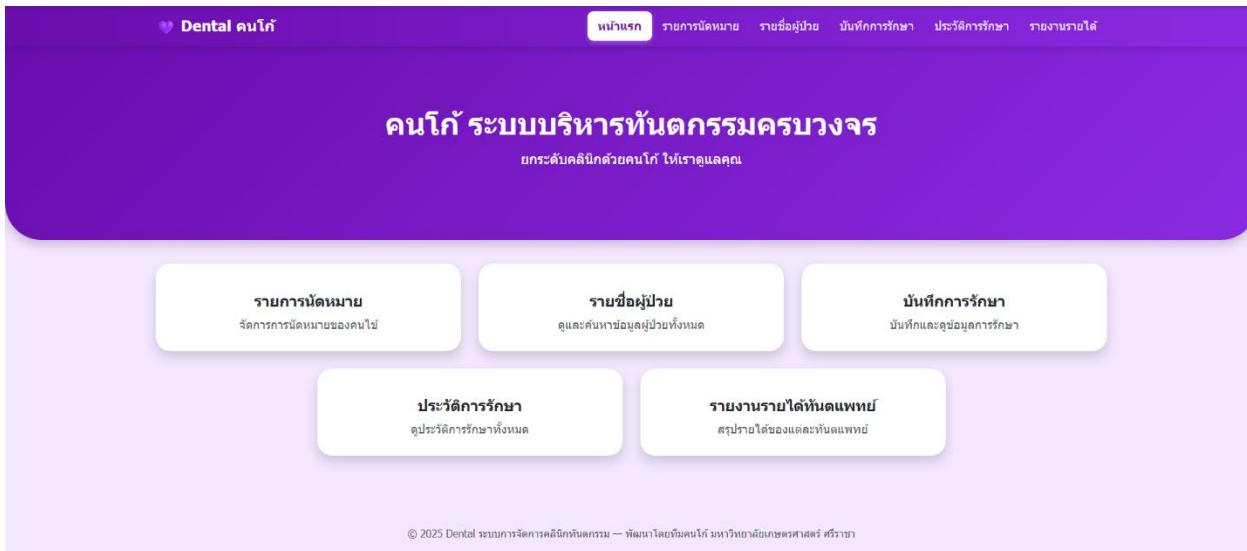
ตารางที่ 6 HistoryDetails รายละเอียดประวัติการรักษา

PK – id_detail

FK - id_record, id_tm

No.	Column Name	Type	Size	Null	Description	Example
1	id_detail	int identity	1,1	no	รหัสรายละเอียด	1
2	id_record	varchar	50	no	รหัสการบันทึก	A001
3	id_tm	varchar	50	no	รหัสหัวตอนการ	T014
4	quantity	Int		no	จำนวนครั้งที่รักษา	1

หน้าจอโปรแกรม



ภาพที่ 1 หน้าจอของโปรแกรม

รายการนัดหมายที่กำลังรอ (3 รายการ)	
+ สร้างบันทึกใหม่	
ID บันทึก	คนไข้
A012	นาย รังสรรค์
A010	นาง ปิยภรณ์
A009	นาย แสงอน
	กับด้วยแพทย์ (ความเชี่ยวชาญ)
	กพญ. อรุณ (Orthodontics)
	กพว.เด็ก (Pediatric Dentistry)
	กพญ.บีบาก (General Dentistry)
	วันที่/เวลา
	2025-11-02 19:33
	2025-11-13 20:41
	2025-11-14 20:39
	สถานะ
	ดำเนินการ
	ยังไม่ได้รับการดำเนินการ
	ยกเว้น
	ยังไม่ได้รับการดำเนินการ
	ยกเว้น
	ยังไม่ได้รับการดำเนินการ
	ยกเว้น

ภาพที่ 2 หน้ารายการนัดหมายในสถานะรอ

สร้างบันทึก ID A013 สำหรับคนไข้ ID P002 สำเร็จแล้ว!

ภาพที่ 3 เมื่อสร้างการนัดหมายสำเร็จ

การสร้างนัดหมายล้มเหลว: ทันตแพทย์ถูกนัดหมายไว้แล้วในช่วงเวลาที่นัด

ภาพที่ 4 เมื่อสร้างการนัดหมายไม่สำเร็จ

Dental คลินิก

หน้าแรก รายการบันทึก รายชื่อผู้ป่วย บันทึกการรักษา ประวัติการรักษา รายงานรายได้

สร้างนัดหมายใหม่

คุณป่วย:

P002 - โน๊ต ใจบุญ ✓

กับแพทย์:

D003 - กว.เด็ก (Pediatric Dentistry) ✓

วันที่:

11/03/2025 เวลา:

10:46 AM

เหตุผลการบันทึกหมาย:

เลี้ยงฟัน

บันทึกบันทึกหมาย

ภาพที่ 5 หน้าการสร้างการนัดหมายใหม่

Dental คลินิก

หน้าแรก รายการบันทึก รายชื่อผู้ป่วย **บันทึกการรักษา** ประวัติการรักษา รายงานรายได้

รายชื่อผู้ป่วย

ID ผู้ป่วย	ชื่อ - นามสกุล	จำนวนครั้งเข้ารักษา	วันที่รักษาล่าสุด	ยอดใช้จ่ายรวม (บาท)
P001	ญาดา ชังอัน	3	02/11/2025	12,000.00
P003	ณเดช ร่างกอง	1	01/11/2025	3,200.00
P005	ໄຕ ໄດเรໂນ	2	02/11/2025	9,800.00
P004	โน๊ต ใจบุญ	4	02/11/2025	7,600.00
P002	โน๊ต ใจบุญ	2	01/11/2025	3,800.00
P006	วิว แสงอุน	1	02/11/2025	1,200.00

ภาพที่ 6 หน้าดูรายชื่อผู้ป่วยที่เข้ารับการรับการรักษา

The screenshot shows a software interface for a dental clinic. At the top, there's a purple header bar with the logo 'Dental คลินิก' and several menu items: หน้าแรก (Home), รายการบันทุกอาการ (List of treatments), รายชื่อผู้ป่วย (Patient list), บันทึกการรักษา (Treatment record), ประวัติการรักษา (Treatment history), and รายงานรายได้ (Report earnings). The main area is titled 'บันทึกประวัติการรักษา' (Treatment record). It displays the following information:

- คณบันทึก: P003 - บันเดช ร่างก่อ (Patient name)
- กันดแพทย์: D002 - พญ.ชุดาดา (General Dentistry) (Doctor name)
- วันที่: A010 - 2025-11-13 20:41 (Date and time)
- รายการรักษาที่ดำเนินการ: T015 - Dental Cleaning (\$800.00) (Treatment performed: Dental Cleaning, cost \$800.00)
- จำนวน: 1 (Quantity)
- ดำเนินการ: เพิ่มรายการรักษา (Add treatment item)
- บันทึกประวัติการรักษา (Save treatment record)

ภาพที่ 7 หน้าบันทึกประวัติการรักษาแบบ walk in

The screenshot shows a software interface for a dental clinic. The main area is titled 'บันทึกประวัติการรักษา' (Treatment record). It displays the following information:

- การบันทึกประวัติการรักษาสำเร็จ ID: H014 (Treatment record completed ID: H014)
- การบันทึกประวัติการรักษาสำเร็จ (Treatment record completed)
- ดูใบเสร็จ (H014) (View receipt (H014))

ภาพที่ 8 หน้าบันทึกประวัติการรักษาสำเร็จ

The screenshot shows a software interface for a dental clinic. The main area is titled 'คลินิกทันตกรรม [ชื่อคลินิกของคุณ]' (Dental Clinic [Your clinic name]) and 'ใบเสร็จการรักษา (Treatment Receipt)'. It displays the following information:

รายการรักษา	จำนวน	ราคาต่อหน่วย (B)	ราคารวม (B)
Dental Cleaning	1	800.00	800.00
Orthodontic Retainer	2	1,500.00	3,000.00

รวมทั้งหมด: 3,800.00 ₩

ข้อมูลที่ไว้วางใจคลินิกของเรามากที่สุด

พิมพ์ใบเสร็จ

ภาพที่ 9 ตัวอย่างใบเสร็จ

ບັນທຶກປະວັດກາຮັກຫາ

ກໍາລັງບັນທຶກປະວັດກາຮັກຫາຈາກບັດໜາຍ ID: A012 (ບ້ອນລຸຄນໄປແລະກັນຕແພກຍຸກກອກອັດໂນມັດ)

ວ່າງອັງຈາກບັດໜາຍ: **A012** (ຄນໄປ/ກັນຕແພກຍຸກກໍາເໜດວັດໂນມັດ)

ຄນໄປ: **P003 - ລາຍເຊີງ ຮ່າງກອງ** ກັນຕແພກຍຸກ: **D001 - ກພຍະນຳ (Orthodontics)**

ຮາຍກາຮັກຫາທີ່ດໍາເນີນກາ

-- ເລືອກຮາຍກາຮັກຫາ -- **1** **ບັນທຶກ**

+ ເພີ່ມຮາຍກາຮັກຫາ

ບັນທຶກປະວັດກາຮັກຫາ

ກາພທີ 10 ພໍານັກປະວັດກາຮັກຫາແບບ ນັດລ່ວງໜ້າ

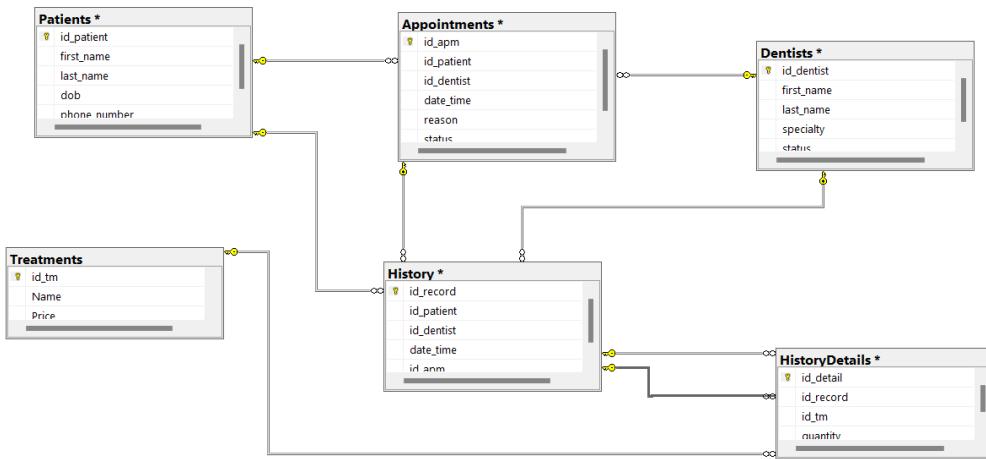
Dental ຄູນໄກ		ໜ້າແພດ	ຮາຍກາຮັກຫາ	ຮາຍເຊື່ອຜູ້ປ້ວຍ	ບັນທຶກກາຮັກຫາ	ປະວັດກາຮັກຫາ	ຮາຍຈານຮາຍໄດ້
ປະວັດກາຮັກຫາທີ່ເສື່ອງສັນ							
ID ປະວັດກາ (Hxxxx)	ID ບັດໜາຍ (Axxxx)	ຄນໄປ	ກັນຕແພກຍຸກ	ຮາຍກາຮັກຫາທີ່ດໍາເນີນ	ຮາຄາວຸນ (ບາດ)	ວັນທີບັນທຶກ	ຈົດກາ
H014	A010	ນາຍເອົງ ຮ່າງກອງ	ກພຍະນຳ	Dental Cleaning (1x)	800.00	2025-11-03 10:48:26	ຖິ່ນສົ່ງ
H013	A011	ຍຸດາ ທັງອັນ	ກພຍະນຳ	Tooth Extraction (2x), Filling (4x)	9,600.00	2025-11-02 17:11:45	ຖິ່ນສົ່ງ
H012	-	ໄຕ ໄກໂກ	ກພຍະນຳ	Filling (5x), Dental Cleaning (1x)	9,800.00	2025-11-02 15:36:53	ຖິ່ນສົ່ງ
H011	-	ໃນ ໄດຣິໄປ	ກພຍະນຳ	Dental Cleaning (1x), Orthodontic Retainer (2x)	3,800.00	2025-11-02 15:35:05	ຖິ່ນສົ່ງ
H010	A008	ວິວ ແລ້ວອັນ	ກພຍະນຳ	Tooth Extraction (1x)	1,200.00	2025-11-02 15:00:40	ຖິ່ນສົ່ງ
H009	-	ໃນ ໄດຣິໄປ	ກພຍະນຳ	Dental Cleaning (1x), Orthodontic Retainer (2x)	3,800.00	2025-11-02 14:59:37	ຖິ່ນສົ່ງ
H007	A003	ຍຸດາ ທັງອັນ	ກພຍະນຳ	Tooth Extraction (2x)	2,400.00	2025-11-01 21:56:48	ຖິ່ນສົ່ງ
H006	A006	ນາຍ ຮ່າງກອງ	ກພຍະນຳ	Dental Cleaning (4x)	3,200.00	2025-11-01 21:47:47	ຖິ່ນສົ່ງ
H005	A005	ໄຕ ໃນຫຼຸກ	ກພຍະນຳ	Dental Cleaning (1x), Orthodontic Retainer (2x)	3,800.00	2025-11-01 21:46:11	ຖິ່ນສົ່ງ

ກາພທີ 11 ພໍານັກປະວັດກາຮັກຫາ

Dental ຄູນໄກ		ໜ້າແພດ	ຮາຍກາຮັກຫາ	ຮາຍເຊື່ອຜູ້ປ້ວຍ	ບັນທຶກກາຮັກຫາ	ປະວັດກາຮັກຫາ	ຮາຍຈານຮາຍໄດ້
ຮາຍຈານສຽງປະວັດກາຮັກຫາ							
ID ມານາດ	ເລືອດ - ດາວໂຫຼວດ	ສະບັບປະຈຸບັນ	ວ່ານວນ Treatments	ຮາຍໄດ້ຮັບ (ບາດ)			
D002	ກພຍະນຳ ດາວໂຫຼວດ ຂອນໄປກີ	ວ່າງ (ມີນັດແສ້ວ)	9	20,500.00			
D001	ກພຍະນຳ ຂອນໄປກີ	ວ່າງ (ມີນັດແສ້ວ)	6	19,000.00			
D003	ກພຍະນຳ ຮ່າງກອງ	ວ່າງ (ມີນັດແສ້ວ)	1	1,200.00			

ກາພທີ 12 ພໍານັກຮາຍຈານສຽງປະວັດກາຮັກຫາໄດ້ຂອງທັນຕະພຶທີ

Database Diagram



คำอธิบาย

1. ตาราง Patients (ผู้ป่วย)

แอ็ตทริบิวต์ (Attributes): id_patient (PK - Primary Key), first_name, last_name, dob (วันเดือนปีเกิด), phone_number

ความสัมพันธ์ (Relationships):

One-to-Many กับ Appointments: ผู้ป่วย 1 คน สามารถมีการนัดหมายได้หลายครั้ง

One-to-Many กับ History: ผู้ป่วย 1 คน สามารถมีประวัติการรักษาที่เกี่ยวข้องได้หลายรายการ

2. ตาราง Dentists (ทันตแพทย์)

แอ็ตทริบิวต์ (Attributes): id_dentist (PK), first_name, last_name, specialty (ความเชี่ยวชาญ), status

ความสัมพันธ์ (Relationships):

One-to-Many กับ Appointments: ทันตแพทย์ 1 คน สามารถมีการนัดหมายได้หลายครั้ง

One-to-Many กับ History: ทันตแพทย์ 1 คน สามารถมีประวัติการรักษาที่เกี่ยวข้องได้หลายรายการ

3. ตาราง Appointments (นัดหมาย)

แอ็ตทริบิวต์ (Attributes): id_apm (PK), id_patient (FK - Foreign Key) (เชื่อม Patients), id_dentist (FK) (เชื่อม Dentists), date_time, reason (เหตุผลที่นัด), status

ความสัมพันธ์ (Relationships):

Many-to-One กับ Patients

Many-to-One กับ Dentists

One-to-One กับ History: การนัดหมาย 1 ครั้ง จะนำไปสู่การบันทึกประวัติการรักษา 1 รายการ

4. ตาราง History (ประวัติการรักษา)

แอ็ตทริบิวต์ (Attributes): id_record (PK), id_patient (FK) (เชื่อม Patients), id_dentist (FK) (เชื่อม Dentists), date_time, id_apm (FK) (เชื่อม Appointments)

ความสัมพันธ์ (Relationships):

Many-to-One กับ Patients

Many-to-One กับ Dentists

One-to-One กับ Appointments

One-to-Many กับ HistoryDetails: บันทึกประวัติ 1 รายการ สามารถมีรายละเอียดรายการรักษาได้หลายรายการ

5. ตาราง Treatments (รายการรักษา)

แอ็ตทริบิวต์ (Attributes): id_tm (PK), Name (ชื่อการรักษา), Price (ราคา)

ความสัมพันธ์ (Relationships):

One-to-Many กับ HistoryDetails: รายการรักษา 1 ชนิด สามารถปรากฏอยู่ในรายละเอียดประวัติการรักษาได้หลายรายการ

6. ตาราง HistoryDetails (รายละเอียดประวัติการรักษา)

แอ็ตทริบิวต์ (Attributes): id_detail (PK), id_record (FK) (เชื่อม History), id_tm (FK) (เชื่อม Treatments), quantity (จำนวนที่ทำ)

ความสัมพันธ์ (Relationships):

Many-to-One กับ History

Many-to-One กับ Treatments

Design Edit Table

	Column Name	Data Type	Allow Nulls
▶	id_apm	varchar(50)	<input type="checkbox"/>
	id_patient	varchar(50)	<input type="checkbox"/>
	id_dentist	varchar(50)	<input type="checkbox"/>
	date_time	datetime	<input type="checkbox"/>
	reason	varchar(100)	<input checked="" type="checkbox"/>
	status	varchar(50)	<input type="checkbox"/>

ภาพที่ 1 เป็นของตาราง Appointment

	Column Name	Data Type	Allow Nulls
▶	id_dentist	varchar(50)	<input type="checkbox"/>
	first_name	varchar(50)	<input type="checkbox"/>
	last_name	varchar(50)	<input checked="" type="checkbox"/>
	specialty	varchar(50)	<input checked="" type="checkbox"/>
	status	varchar(50)	<input checked="" type="checkbox"/>

ภาพที่ 2 เป็นของตาราง Dentists

	Column Name	Data Type	Allow Nulls
▶	id_record	varchar(50)	<input type="checkbox"/>
	id_patient	varchar(50)	<input checked="" type="checkbox"/>
	id_dentist	varchar(50)	<input type="checkbox"/>
	date_time	datetime	<input type="checkbox"/>
	id_apm	varchar(50)	<input checked="" type="checkbox"/>

ภาพที่ 3 เป็นของตาราง History

	Column Name	Data Type	Allow Nulls
!	id_detail	int	<input type="checkbox"/>
	id_record	varchar(50)	<input type="checkbox"/>
	id_tm	varchar(50)	<input type="checkbox"/>
	quantity	int	<input type="checkbox"/>

ภาพที่ 4 เป็นของตาราง HistoryDetails

	Column Name	Data Type	Allow Nulls
!	id_patient	varchar(50)	<input type="checkbox"/>
	first_name	varchar(50)	<input checked="" type="checkbox"/>
	last_name	varchar(50)	<input checked="" type="checkbox"/>
	dob	datetime	<input type="checkbox"/>
	phone_number	varchar(50)	<input type="checkbox"/>

ภาพที่ 5 เป็นของตาราง Patients

	Column Name	Data Type	Allow Nulls
!	id_tm	varchar(50)	<input type="checkbox"/>
	Name	varchar(50)	<input type="checkbox"/>
	Price	int	<input checked="" type="checkbox"/>

ภาพที่ 6 เป็นของตาราง Treatment

View

Column	Alias	Table	Output	Sort Type	Sort Order	Group By	Filter	Or...	Or...	Or...
status		Dentists	<input checked="" type="checkbox"/>			Group By				
id_record	Treatm...	HistoryDet...	<input checked="" type="checkbox"/>			Count				
dbo.HistoryD...	SUM		<input checked="" type="checkbox"/>			Sum				

```

SELECT dbo.Dentists.id_dentist, dbo.Dentists.first_name, dbo.Dentists.last_name, dbo.Dentists.status, COUNT(dbo.HistoryDetails.id_record) AS TreatmentsCompleted, SUM(dbo.HistoryDetails.quantity * dbo.Treatments.Price) AS TotalTreatmentCost
FROM dbo.Dentists LEFT OUTER JOIN
    dbo.History ON dbo.Dentists.id_dentist = dbo.History.id_dentist INNER JOIN
    dbo.HistoryDetails ON dbo.History.id_record = dbo.HistoryDetails.id_record INNER JOIN
    dbo.Treatments ON dbo.HistoryDetails.id_tm = dbo.Treatments.id_tm
GROUP BY dbo.Dentists.id_dentist, dbo.Dentists.first_name, dbo.Dentists.last_name, dbo.Dentists.status

```

	id_dentist	first_name	last_name	status	Treatments...	SUM
▶	D001	พญ.นิชา	คณิต์	ว่าง (มีนัดแล้ว)	4	9400
	D002	พญ.ชนิดา	ศนฟ์	ว่าง (มีนัดแล้ว)	2	2300

ภาพที่ 1 การวิเคราะห์และการรายงานเชิงลึกของฐานข้อมูล

โดยมีขั้นตอนและหลักการสำคัญดังนี้

1. การเชื่อมโยงตาราง (JOIN Operations)

การวิเคราะห์เริ่มต้นจากตารางหลัก Dentists และใช้การเชื่อมโยง (JOIN) กับตารางอื่น ๆ ที่เกี่ยวข้อง ตามลำดับความสัมพันธ์: History (ประวัติการรักษา), HistoryDetails (รายละเอียดประวัติการรักษา), และ Treatments (รายการหัตถการ)

มีการใช้ LEFT OUTER JOIN จากตาราง Dentists ไปยัง History เพื่อให้มั่นใจว่าข้อมูลของทันตแพทย์ทุกคนจะถูกดึงออกมาระบบ แสดงผลทั้งหมด แม้ว่าทันตแพทย์บางท่านจะยังไม่มีประวัติการรักษาในระบบก็ตาม

การคำนวณตัวชี้วัดหลัก (Key Metric Calculation)

คำสั่ง SQL ได้กำหนดการคำนวณตัวชี้วัดหลัก 2 ตัว ดังนี้

1. TreatmentsCompleted (จำนวนการรักษาที่สำเร็จ): คำนวณจากการนับจำนวนรายการบันทึกประวัติการรักษา (COUNT(HistoryDetails.id_record)) เพื่อแสดงปริมาณงานที่สำเร็จของทันตแพทย์แต่ละท่าน
2. SUM (ยอดรวมรายได้): คำนวณยอดรวมรายได้จากการรักษาทั้งหมด โดยใช้สูตร: (dbo.HistoryDetails.quantity * dbo.Treatments.Price) ซึ่งเป็นการนำจำนวนหัตถการที่ทำจริงคูณด้วย ราคาต่อหัตถการที่กำหนดไว้

การจัดกลุ่มข้อมูล (Grouping)

ผลลัพธ์ทั้งหมดถูก จัดกลุ่ม (GROUP BY) ตามรหัสพัฒนาแพทย์ และข้อมูลส่วนตัวอื่น ๆ เพื่อร่วมผลการนับ และการคำนวณยอดรวมเข้ากับทันตแพทย์แต่ละคนอย่างถูกต้อง

Column	Alias	Table	Output	Sort Type	Sort Order	Group By	Filter	Or...	Or...	Or...
<code>id_patient</code>		Patients	☒			Group By				
<code>first_name</code>		Patients	☒			Group By				
<code>last_name</code>		Patients	☒			Group By				
<code>id_record</code>	TotalVis...	History	☒			Count				
<code>date_time</code>	LastVis...	History	☒			Max				
<code>dbo.HistoryD...</code>	TotalSp...	History	☒			Sum				

```

SELECT dbo.Patients.id_patient, dbo.Patients.first_name, dbo.Patients.last_name, COUNT(dbo.History.id_record) AS TotalVisits, MAX(dbo.History.date_time) AS LastVisitDate, SUM(dbo.HistoryDetails.quantity * dbo.Treatments.Price)
AS TotalSpending
FROM dbo.History INNER JOIN
dbo.HistoryDetails ON dbo.History.id_record = dbo.HistoryDetails.id_record INNER JOIN
dbo.Treatments ON dbo.HistoryDetails.id_tm = dbo.Treatments.id_tm INNER JOIN
dbo.Patients ON dbo.History.id_patient = dbo.Patients.id_patient
GROUP BY dbo.Patients.id_patient, dbo.Patients.first_name, dbo.Patients.last_name

```

ภาพที่ 2 การวิเคราะห์ใช้ลีกของข้อมูลผู้ป่วย

นี่คือการวิเคราะห์ที่มุ่งเน้นการประเมินความถี่ในการเข้าใช้บริการและยอดค่าใช้จ่ายรวมของผู้ป่วยแต่ละราย

1. ส่วนบน: แผนภาพความสัมพันธ์ของตาราง (Schema Relationship)

ส่วนนี้แสดงตารางที่เกี่ยวข้องในการวิเคราะห์ซึ่งเนื่องกับภาพก่อนหน้า แต่เน้นการเขียนโดยจากตาราง

Patients เป็นหลัก:

- Patients (ผู้ป่วย) เชื่อมไปยัง History (ประวัติการรักษา)
- History เชื่อมไปยัง HistoryDetails (รายละเอียดประวัติ)
- HistoryDetails เชื่อมไปยัง Treatments (รายการหัตถการ)

สถิติความถี่และยอดค่าใช้จ่ายรวมของผู้ป่วย

เราใช้คำสั่ง SQL ได้ดำเนินการคำนวณของผู้ป่วยแต่ละราย ได้แก่ จำนวนครั้งที่เข้าใช้บริการรวม (TotalVisits) วัน/เวลาที่เข้าใช้บริการล่าสุด (LastVisitDate) และ ยอดค่าใช้จ่ายรวม (TotalSpending) ข้อมูลดังกล่าวมีความสำคัญอย่างยิ่งต่อฝ่ายการตลาดและการบริการลูกค้าเพื่อใช้ในการวางแผนส่งเสริมการขายหรือการติดตามผู้ป่วยที่ไม่กลับมา nhậnบริการตามกำหนด

Function ,Store Procedure

```

CREATE PROCEDURE [dbo].[CreateAppointmentID]
    @NewAppointmentID VARCHAR(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @MaxID VARCHAR(50);
    DECLARE @NewIDNum INT;

    -- ค้นหา ID สูงสุดที่ขึ้นต้นด้วย 'A'
    SELECT @MaxID = MAX(id_apm)
    FROM Appointments
    WHERE id_apm LIKE 'A%';

    -- คำนวณรหัสตัวเลขใหม่ (ดึงตัวเลข 3 หลัก)
    IF @MaxID IS NULL OR @MaxID NOT LIKE 'A[0-9]%'
        SET @NewIDNum = 1;
    ELSE
        -- ดึงเฉพาะตัวเลข 3 หลักอุ่กมา แล้วบวกเพิ่มไป 1
        SET @NewIDNum = CAST(SUBSTRING(@MaxID, 2, 3) AS INT) + 1;

    -- *** แก้ไขตรงนี้: เปลี่ยน '0000' เป็น '000' และ VARCHAR(4) เป็น VARCHAR(3) ***
    SET @NewAppointmentID = 'A' + RIGHT('000' + CAST(@NewIDNum AS VARCHAR(3)), 3);
END;
GO

```

ภาพที่ 1 SP CreateAppointmentID สร้างรหัสการนัดหมายใหม่

วัตถุประสงค์

การสร้างรหัสการนัดหมายใหม่ที่มีรูปแบบการเรียงลำดับที่แน่นอนให้กับรายการการนัดหมายใหม่ในตาราง Appointments รหัสที่สร้างขึ้นจะถูกส่งออกไปใช้งานภายนอกผ่านพารามิเตอร์ข้อออก (OUTPUT)

การทำงาน

1. โค้ดจะค้นหา รหัส id_apm ที่มีค่าสูงสุด ที่ขึ้นต้นด้วยตัวอักษร 'A' ในตาราง Appointments
2. คำนวณลำดับถัดไป
 - ไม่พบรหัสเดิม ระบบจะเริ่มต้นรหัสใหม่ที่หมายเลข 1 (A001)
 - พบรหัสเดิม โค้ดจะทำการ แยกตัวเลข 3 หลักท้าย ออกมานะ แปลงเป็นจำนวนเต็ม และ เพิ่มค่าเข้าไป 1 เพื่อให้ได้ลำดับถัดไป
3. นำตัวอักษร 'A' มาต่อ กับ ตัวเลขลำดับใหม่ เพื่อสร้างรหัสใหม่ที่สมบูรณ์ จากนั้นรหัสนี้จะถูกส่งออกไปเพื่อใช้ในการแทรกข้อมูลการนัดหมายใหม่ต่อไป

```

USE [Dental]
GO

***** Object: StoredProcedure [dbo].[CreateNewPatient] Script Date: 3/11/2568 21:24:48 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[CreateNewPatient]
    @first_name VARCHAR(50),
    @last_name VARCHAR(50),
    @dob DATETIME,
    @phone_number VARCHAR(50),
    @NewPatientID VARCHAR(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @MaxID VARCHAR(50);
    DECLARE @NewIDNum INT;

    -- 1. สร้าง ID คนไข้ใหม่ (Pxxx)
    SELECT @MaxID = MAX(id_patient)
    FROM Patients
    WHERE id_patient LIKE 'P%';

    IF @MaxID IS NULL OR @MaxID NOT LIKE 'P[0-9]%'
        SET @NewIDNum = 1;
    ELSE
        -- ตีแผ่เฉพาะตัวเลข 3 หลักของมา แล้วบวกเพิ่มไป 1
        SET @NewIDNum = CAST(SUBSTRING(@MaxID, 2, 3) AS INT) + 1;

    -- *** แก้ไขตรงนี้: เปลี่ยน '0000' เป็น '000' และ VARCHAR(4) เป็น VARCHAR(3) ***
    SET @NewPatientID = 'P' + RIGHT('000' + CAST(@NewIDNum AS VARCHAR(3)), 3);

    -- 2. บันทึกข้อมูลคนไข้ใหม่ลงตาราง Patients
    INSERT INTO Patients (id_patient, first_name, last_name, dob, phone_number)
    VALUES (@NewPatientID, @first_name, @last_name, @dob, @phone_number);

    -- 3. คืนค่า ID ใหม่
    SELECT @NewPatientID AS NewPatientID;
END;
GO

```

ภาพที่ 2 CreateNewPatient

วัตถุประสงค์

CreateNewPatient มีหน้าที่ในการสร้างรหัสผู้ป่วยใหม่แบบเรียงลำดับ โดยอัตโนมัติ และบันทึกข้อมูลผู้ป่วยใหม่ลงในตาราง Patients โดยรูปแบบรหัสผู้ป่วยที่สร้างจะขึ้นต้นด้วยตัวอักษร 'P' และตามด้วยตัวเลข 3 หลัก ที่มีการเติมศูนย์นำหน้า (เช่น P001, P015, P123)

ขั้นตอนการทำงานโดยสรุป

1. หา ID ล่าสุดดึงค่ารหัสผู้ป่วยที่มากที่สุด ที่เริ่มต้นด้วย 'P' จากตาราง Patients
2. คำนวณเลขลำดับ โดยถ้าไม่พบรหัสเดิม หรือรหัสเดิมพิศรุปแบบ จะกำหนดให้เลขลำดับเริ่มต้นที่ 1 ถ้าพบรหัสเดิมที่ถูกต้อง จะตัดเอาเลขลำดับ 3 หลัก จากรหัสเดิมมา บวก 1 เพื่อใช้เป็นเลขลำดับใหม่
3. สร้าง ID ใหม่ นำเลขลำดับใหม่มาแปลงเป็นข้อความ 3 หลัก แล้วนำมาต่อกับตัวอักษร 'P' เพื่อสร้างรหัสใหม่ที่สมบูรณ์
4. บันทึกข้อมูลโดยนำรหัสผู้ป่วยใหม่ที่สร้างได้ พร้อมกับข้อมูลที่รับมาบันทึกลงในตาราง Patients
5. คืนค่า ID ใหม่แล้วส่งค่ารหัสผู้ป่วยใหม่ที่สร้างขึ้นออกไปที่ตัวแปร @NewPatientID

```

USE [Dental]
GO

***** Object: StoredProcedure [dbo].[RecordTreatment] Script Date: 3/11/2568 21:25:38 *****
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[RecordTreatment] (
    @id_patient VARCHAR(50),
    @id_dentist VARCHAR(50),
    @id_apm VARCHAR(50) = NULL,
    @NewRecordID VARCHAR(50) OUTPUT
)
WITH RECOMPILE -- << คำสั่งไม่ใช้กับ MySQL
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @MaxID VARCHAR(50);
    DECLARE @NewIDNum INT;

    -- 1. หานั้น ID ปัจจุบัน (Hxxx)
    SELECT @MaxID = MAX(id_record)
    FROM History
    WHERE id_record LIKE 'H[0-9]%';

    IF @MaxID IS NULL OR @MaxID NOT LIKE 'H[0-9]%'
        SET @NewIDNum = 1;
    ELSE
        SET @NewIDNum = CAST(SUBSTRING(@MaxID, 2, 3) AS INT) + 1;

    SET @NewRecordID = 'H' + RIGHT('000' + CAST(@NewIDNum AS VARCHAR(3)), 3);

    -- 2. บันทึกข้อมูลการรักษาหน้า (History)
    -- *** บันทึกข้อมูล id_treatments ในรายการของลักษณะและที่ตั้ง INSERT ***
    INSERT INTO History (
        id_record,
        id_patient,
        id_dentist,
        date_time,
        id_apm
    )
    VALUES (
        @NewRecordID,
        @id_patient,
        @id_dentist,
        GETDATE(),
        @id_apm
    );

    -- 3. อัปเดตสถานะในตาราง Appointments
    IF @id_apm IS NOT NULL AND @id_apm <> ''
    BEGIN
        UPDATE Appointments
        SET status = 'สำเร็จ'
        WHERE id_apm = @id_apm;
    END
END;
GO

```

ภาพที่ 3 RecordTreatment

วัตถุประสงค์

RecordTreatment มีหน้าที่ในการบันทึกประวัติการรักษาใหม่ลงในตาราง History และ ปรับปรุงสถานะของการนัดหมายที่เกี่ยวข้องในตาราง Appointments

โดยรูปแบบรหัสประวัติการรักษาที่สร้างจะขึ้นต้นด้วยตัวอักษร 'H' และตามด้วยตัวเลข 3 หลัก ที่มีการเติมศูนย์นำหน้า

ขั้นตอนการทำงาน

1. หา ID ล่าสุด ดึงค่ารหัสประวัติการรักษาที่มากที่สุดที่เริ่มต้นด้วย 'H' จากตาราง History
2. คำนวนเลขลำดับถ้าไม่พบรหัสเดิม หรือรหัสเดิมผิดรูปแบบ จะกำหนดให้เลขลำดับเริ่มต้นที่ 1 ถ้าพบรหัสเดิมที่ถูกต้อง จะตัดเอาเลขลำดับ 3 หลัก จากรหัสเดิมมา บวก 1 เพื่อใช้เป็นเลขลำดับใหม่
3. สร้าง ID ใหม่โดยนำเลขลำดับใหม่มาแปลงเป็นข้อความ 3 หลักแล้วนำมารอต่อ กับตัวอักษร 'H' เพื่อสร้างรหัสใหม่ที่สมบูรณ์ แล้วส่งค่าออกไปที่ตัวแปร @NewRecordID
4. บันทึกข้อมูลโดยนำรหัสประวัติใหม่ที่สร้างได้ พร้อมกับ @id_patient, @id_dentist, GETDATE() และ @id_apm บันทึกลงในตาราง History
5. ปรับปรุงสถานะเมื่อกำหนดหมายมากด้วย ทำการอัปเดตสถานะของการนัดหมายนั้นในตาราง Appointments ให้เป็น 'สำเร็จ'

```

USE [Dental]
GO

***** Object: StoredProcedure [dbo].[RecordTreatmentDetails] Script Date: 3/11/2568 21:26:34 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[RecordTreatmentDetails]
    @id_record VARCHAR(50),          --ID ประวัติเพิ่งสร้างจาก RecordTreatment
    @id_treatment VARCHAR(50),        --รหัสการรักษา
    @quantity INT                      --จำนวน
AS
BEGIN
    SET NOCOUNT ON;
    --บันทึกรายการรักษาลงใน HistoryDetails
    INSERT INTO HistoryDetails (
        id_record,
        id_tm,
        quantity
    )
    VALUES (
        @id_record,
        @id_treatment,
        @quantity
    );
END;
GO

```

ภาพที่ 4 RecordTreatmentDetails

วัตถุประสงค์

RecordTreatmentDetails มีหน้าที่หลักคือ การบันทึกรายละเอียดการรักษาแต่ละรายการที่เกิดขึ้นในการรักษาหนึ่งๆ ซึ่งถูกบันทึกไว้แล้วใน History ลงในตาราง HistoryDetails

ขั้นตอนการทำงาน

1. รับข้อมูล: รับข้อมูล 3 ส่วนเป็น Input Parameters
2. @id_record ID ประวัติการรักษาหลัก ที่เพิ่งสร้างจาก Procedure ก่อนหน้า RecordTreatment
3. @id_treatment รหัสรายการรักษา หรือบริการที่ทำ
4. @quantity จำนวน ของรายการรักษาหรือบริการนั้นๆ
5. บันทึกรายละเอียด: นำค่า Input Parameters ทั้งสามรายการมาทำการบันทึกลงในตาราง HistoryDetails

Trigger

```
--Trigger ตรวจสอบไม่ให้ทันตแพทย์ถูกนัดซ้ำช้อนในเวลาเดียวกัน
CREATE TRIGGER CheckDentist
ON Appointments
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;
    --ตรวจสอบว่ามีรายการนัดหมายใหม่ (inserted) ที่ช่วงเวลาเดียวกับรายการที่ถูกยุบรวมหรือไม่
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN Appointments a
            ON i.id_dentist = a.id_dentist      --ทันตแพทย์คนเดียวกัน
            AND i.date_time = a.date_time       --เวลาเดียวกัน
        WHERE a.id_apm <> i.id_apm          --ต้องไม่ใช่นัดหมายเดียวกัน
    )
    BEGIN
        --ถ้าพบการนัดกันให้ยกเลิกการท่ารายการและแจ้งข้อผิดพลาด
        RAISERROR('Error: The selected Dentist is already scheduled at this time.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END
END
GO
```

ภาพที่ 1 Trigger CheckDentist การควบคุมความซ้ำช้อนของการนัดหมาย

วัตถุประสงค์

การป้องกันไม่ให้ทันตแพทย์ถูกนัดหมายซ้ำช้อนในช่วงวันและเวลาเดียวกัน

การทำงาน

- Trigger จะทำงานโดยอัตโนมัติ หลังจากการแทรกรายการนัดหมายใหม่ เข้าสู่ตาราง Appointments
- โค้ดจะทำการตรวจสอบว่ารายการนัดหมายใหม่ที่ถูกแทรกเข้ามานั้น มีรายการอื่นใดในตารางที่ตรงกันทั้ง รหัสทันตแพทย์ และ วัน-เวลา หรือไม่
- หากพบความซ้ำช้อน ระบบจะทำการ ROLLBACK TRANSACTION พร้อมแสดงข้อความข้อผิดพลาด เพื่อให้ข้อมูลที่ขัดแย้งกับความเป็นจริงจะไม่ถูกบันทึกในฐานข้อมูล

```

--Triggerสถานะทันตแพทย์
CREATE TRIGGER UpdateDentistStatus
ON Appointments
AFTER INSERT, DELETE
AS
BEGIN
    SET NOCOUNT ON;
    --สร้างตารางชั่วคราวเพื่อรวม ID ของทันตแพทย์ที่เกี่ยวข้อง
    SELECT id_dentist INTO #AffectedDentists FROM inserted
    UNION
    SELECT id_dentist FROM deleted;

    --อุปกรณ์อัปเดตสถานะของทันตแพทย์ที่ถูกกรอบพื้นที่
    UPDATE D
    SET
        D.status =
        CASE
            WHEN (
                SELECT COUNT(*)
                FROM Appointments A
                WHERE A.id_dentist = D.id_dentist
                AND A.status = 'กำลังรอด'
            ) >= 5 THEN 'เต็ม (คิวแน่น)' --ตั้งค่า Threshold ตามที่ต้องการ
            WHEN (
                SELECT COUNT(*)
                FROM Appointments A
                WHERE A.id_dentist = D.id_dentist
                AND A.status = 'กำลังรอด'
            ) >= 1 THEN 'ว่าง (มีนัด)'
            ELSE 'ว่าง (ไม่มีคิว)'
        END
    FROM Dentists D
    INNER JOIN #AffectedDentists AD ON D.id_dentist = AD.id_dentist;
    DROP TABLE #AffectedDentists;
END
GO

```

ภาพที่ 2 Trigger UpdateDentistStatus อัปเดตสถานะการพร้อมให้บริการ

วัตถุประสงค์

อัปเดตสถานะการพร้อมให้บริการของทันตแพทย์ในตาราง Dentists โดยอัตโนมัติ เพื่อจะท่อนถึงภาระงานนัดหมายที่กำลังดำเนินการอยู่ (สถานะเป็น 'กำลังรอด')

การทำงาน

1. Trigger ถูกกำหนดให้ทำงาน หลังการทำรายการแทรก (INSERT), ลบ (DELETE), หรือแก้ไข (UPDATE) บนตาราง Appointments
2. ระบบจะทำการสร้างตารางชั่วคราว (#AffectedDentists) เพื่อรับรวมรหัสทันตแพทย์ที่ได้รับผลกระทบจากการทำรายการล่าสุด
3. ให้ UPDATE สถานะของทันตแพทย์ที่อยู่ในกลุ่มเป้าหมาย โดยใช้คำสั่ง CASE WHEN เพื่อ คำนวณจำนวนรายการนัดหมายที่มีสถานะ 'กำลังรอ' ของทันตแพทย์แต่ละคน ในขณะนี้
4. กำหนดสถานะใหม่:
 - ถ้าจำนวนนัดหมายสถานะ 'กำลังรอ' ≥ 5 รายการ: สถานะจะถูกตั้งเป็น 'เต็ม (มีคิว)'
 - ถ้าจำนวนนัดหมายสถานะ 'กำลังรอ' ≥ 1 รายการ: สถานะจะถูกตั้งเป็น 'ว่าง (มีคิว)'
 - ถ้า 'ไม่มี' นัดหมายสถานะ 'กำลังรอ': สถานะจะถูกตั้งเป็น 'ว่าง (ไม่มีคิว)'