

## ขั้นตอนการรันคำสั่ง

โจทย์ที่ 1: Hotel Booking

กลุ่ม Agado (กลุ่มที่ 6)

### 1. Create Table

ในขั้นตอนนี้มีไฟล์ `create/drop_all.sql` ใช้สำหรับต้องการเริ่มการทำงานใหม่ตั้งแต่ต้น ไม่จำเป็นไม่ต้อง run

#### 1.1) สร้าง user table และ subclass

**คำอธิบาย:** สร้าง table usert (superclass) และ table admin, unregistered\_user, registered\_user (subclass ของ usert) และ

**ส่วนของ SQL command:** `create/create_user.sql`

**คำแนะนำ:** ควร run ทีละคำสั่ง

ผลการรัน:

```
CREATE TABLE unregistered_user (
    user_id INT,
    FOREIGN KEY (user_id) REFERENCES usert(user_id)
);

CREATE TABLE registered_user(
    user_id INT,
    user_name VARCHAR(64) NOT NULL,
    user_password VARCHAR(64) NOT NULL,
    user_email VARCHAR(64) NOT NULL,
    telephone_number VARCHAR(10),
    register_date TIMESTAMP NOT NULL,
    birth_date DATE NOT NULL,
    FOREIGN KEY(user_id) REFERENCES usert(user_id)
);

CREATE TABLE admin(
    user_id INT,
    user_name VARCHAR(64) NOT NULL,
    user_password VARCHAR(64) NOT NULL,
    user_email VARCHAR(64) NOT NULL,
    telephone_number VARCHAR(10),
    register_date TIMESTAMP NOT NULL,
    birth_date DATE NOT NULL,
    FOREIGN KEY(user_id) REFERENCES usert(user_id)
);
```

#### 1.2) สร้าง hotel table และ subclass/relation ที่เกี่ยวข้อง

**คำอธิบาย:** สร้าง table hotel, hotel\_facilities, room, room\_facilities

**ส่วนของ SQL command:** `create/create_hotel.sql`

**คำแนะนำ:** ควร run ทีละคำสั่ง

ผลการรัน:

```

CREATE TABLE room_facilities(
    hotel_id INT,
    room_id INT,
    min_price INT,
    max_price INT,
    status BOOLEAN,
    room_type VARCHAR(64),
    PRIMARY KEY(hotel_id, room_id),
    FOREIGN KEY(hotel_id) REFERENCES hotel(hotel_id)
);

CREATE TABLE room_facilities(
    hotel_id INT,
    room_id INT,
    room_facilities VARCHAR(256), --facilities description
    FOREIGN KEY(hotel_id, room_id) REFERENCES room(hotel_id, room_id)
);

```

Total rows: Query complete 00:00:00.087 CRLF Ln 42, Col 3

### 1.3) สร้าง table ที่เก็บข้อมูลการแก้ไข logs จากการ view/insert/edit/delete ของ user

**คำอธิบาย:** สร้างตาราง booking\_transaction (เก็บคำสั่งการจองห้องพัก), view\_hotel\_info (เก็บข้อมูลการเข้าชม), edit\_overall\_info (เก็บการแก้ไข hotel info ของ admin) และ manages\_booking (เก็บข้อมูลการแก้ไขการจอง)

**ส่วนของ SQL command:** create/create\_relationship.sql

**คำแนะนำ:** ควร run ทีละคำสั่ง

ผลการรัน:

```

CREATE TABLE edit_overall_info(
    hotel_id INT,
    user_id INT,
    edit_timestamp TIMESTAMP,
    FOREIGN KEY(hotel_id) REFERENCES hotel(hotel_id),
    FOREIGN KEY(user_id) REFERENCES user(user_id)
);

--log for admin edit booking details
CREATE TABLE manages_booking(
    user_id INT,
    booking_id INT,
    edit_timestamp TIMESTAMP,
    FOREIGN KEY(booking_id) REFERENCES booking_transaction(booking_id),
    FOREIGN KEY(user_id) REFERENCES user(user_id)
);

```

Total rows: Query complete 00:00:00.070 CRLF Ln 45, Col 3

## 2. Insert ข้อมูลใน Table

### 2.1) ใส่ข้อมูลตัวอย่างสำหรับ user

**คำอธิบาย:** ประกอบด้วย admin 9 คน, registered\_user 63 คน และ unregistered\_user 41 คน  
**ส่วนของ SQL command:**

1. *insert/util\_gen\_user\_id.sql* (สร้างฟังก์ชัน generate id)
2. *insert/user\_insert\_user.sql* (สร้างฟังก์ชัน insert user)
3. *sample\_data/user1.sql* (ข้อมูลทดสอบ)

คำแนะนำ:

ผลการรัน:

#### 2.1.1) ผลที่ปรากฏใน usert

The screenshot shows a SQL query interface with the following details:

- Query History:** Shows the query `select * from usert;`
- Data Output:** Displays the results of the query in a grid format.
- Table Structure:** The table has two columns: `user_id` [PK] integer and `device_type` character varying (32).
- Results:** The grid shows 113 rows of data, with the first few rows being:
  - 1 | 900000001 | iOS
  - 2 | 900000002 | Android
  - 3 | 900000003 | Windows
  - 4 | 500000001 | iOS
  - 5 | 500000002 | Android
  - 6 | 500000003 | Windows
  - 7 | 500000004 | iOS
  - 8 | 500000005 | Android
  - 9 | 500000006 | Windows
  - 10 | 100000001 | iOS
  - 11 | 100000002 | Android
  - 12 | 100000003 | Windows
  - 13 | 900000004 | iOS
  - 14 | 900000005 | Android
  - 15 | 900000006 | Windows
  - 16 | 500000007 | iOS
- Bottom Status:** Total rows: 113, Query complete 00:00:00.068, CRLF, Ln 162, Col 1.

#### 2.1.2) ผลที่ปรากฏใน admin

Agado/admin@DBServer

Query History

```
162 select * from admin;
```

Data Output Messages Notifications

	user_id integer	officer_id integer	admin_name character varying (64)	admin_password character varying (64)	admin_email character varying (64)	telephone_number character varying (10)
1	90000001	113	Sittipong Wongsakul	password123	sittipong.wongsakul@example.com	1090123456
2	90000002	114	Pimchanok Jirawat	password456	pimchanok.jirawat@example.com	1101234567
3	90000003	115	Chanin Sutharak	password789	chanin.sutharak@example.com	1112345678
4	90000004	116	Thanakorn Rattanapong	password123	thanakorn.rattanapong@example.com	1189012345
5	90000005	117	Wassana Pongsuk	password456	wassana.pongsuk@example.com	1190123456
6	90000006	118	Supaporn Rak Sri	password789	supaporn.rak Sri@example.com	1201234567
7	90000007	119	Jirawat Charoenwong	password123	jirawat.charoenwong@example.com	1278901234
8	90000008	120	Piyarat Sittipong	password456	piyarat.sittipong@example.com	1289012345
9	90000009	121	Prapaporn Saengsawang	password789	prapaporn.saengsawang@example.com	1290123456

Total rows: 9 Query complete 00:00:00.108 CRLF Ln 162, Col 21

### 2.1.3) ผลที่ปรากฏใน unregistered\_user

Query History

```
162 select * from unregistered_user;
```

Data Output Messages Notifications

	user_id integer
1	10000001
2	10000002
3	10000003
4	10000004
5	10000005
6	10000006
7	10000007
8	10000008
9	10000009
10	10000010
11	10000011
12	10000012
13	10000013
14	10000014
15	10000015
16	10000016

Total rows: 41 Query complete 00:00:00.079 CRLF Ln 162, Col 34

### 2.1.4) ผลที่ปรากฏใน registered\_user

Query Query History

```
162 select * from registered_user;
```

Data Output Messages Notifications

SQL Showing rows: 1 to 63 Page No: 1 of 1 |◀|◀|▶|▶|

	user_id integer	user_name character varying (64)	user_password character varying (64)	user_email character varying (64)	telephone_number character varying (10)	register_date timestamp without time zone
1	500000001	Sudarat Meesang	regpass123	sudarat.meesang@example.com	1123456789	2025-02-03 16:16:18.6
2	500000002	Sakchai Phongsak	regpass456	sakchai.phongsak@example.com	1134567890	2025-02-03 16:16:18.6
3	500000003	Thanyarat Rattanasak	regpass789	thanyarat.rattanasak@example.com	1145678901	2025-02-03 16:16:18.6
4	500000004	Apichart Tanthanuch	regpass123	apichart.tanthanuch@example.com	1156789012	2025-02-03 16:16:18.6
5	500000005	Kwanjai Chuchai	regpass456	kwanjai.chuchai@example.com	1167890123	2025-02-03 16:16:18.6
6	500000006	Nattanon Manachai	regpass789	nattanon.manachai@example.com	1178901234	2025-02-03 16:16:18.6
7	500000007	Kittima Pongtavee	regpass123	kittima.pongtavee@example.com	1212345678	2025-02-03 16:16:18.6
8	500000008	Phakphum Chailai	regpass456	phakphum.chailai@example.com	1223456789	2025-02-03 16:16:18.6
9	500000009	Nirun Phonjan	regpass789	nirun.phonjan@example.com	1234567890	2025-02-03 16:16:18.6
10	500000010	Rapeeporn Thitiphan	regpass123	rapeeporn.thitiphan@example.com	1245678901	2025-02-03 16:16:18.6
11	500000011	Saranya Kittinan	regpass456	saranya.kittinan@example.com	1256789012	2025-02-03 16:16:18.6
12	500000012	Orawan Khemtong	regpass789	orawan.khemtong@example.com	1267890123	2025-02-03 16:16:18.6
13	500000013	Nattapol Phinyo	regpass123	nattapol.phinyo@example.com	1301234567	2025-02-03 16:16:18.6
14	500000014	Tassana Pongsathaporn	regpass456	tassana.pongsathaporn@example.com	1312345678	2025-02-03 16:16:18.6
15	500000015	Kritsanapong Srisai	regpass789	kritsanapong.srisai@example.com	1323456789	2025-02-03 16:16:18.6

Total rows: 63 Query complete 00:00:00.098 CRLF Ln 162, Col 32

## 2.2) ใส่ข้อมูลตัวอย่างสำหรับ hotel

คำอธิบาย: ประกอบด้วย 10 โรงแรม

ส่วนของ SQL command:

1. `insert/util_gen_hotel_id.sql` (สร้างฟังก์ชัน generate id)
2. `insert/admin_add_room_facilities.sql`
3. `insert/admin_add_room.sql`
4. `insert/admin_insert_hotel_info.sql`
5. `sample_data/add_hotel_overall_info.sql` (ข้อมูลทดสอบ)

คำแนะนำ:

ผลการรัน:

2.2.1) ผลที่ปรากฏใน hotel

Query Query History

```
1 select * from hotel;
```

Data Output Messages Notifications

Showing rows: 1 to 10 | Page No: 1 of 1 |

	hotel_id [PK] integer	map_url character varying	hotel_location character varying (256)	hotel_phonenumber character varying (10)	check_in_time time without time zone	check_out_time time without time zone
1	10000	http://example.com/hotel1	Grand Palace Hotel	0812345678	14:00:00	12:00:00
2	10001	[null]	Cozy Cabin Inn	[null]	[null]	[null]
3	10002	[null]	Beachfront Resort	0888888888	15:30:00	11:00:00
4	10003	http://example.com/hotel4	Mountain View Lodge	0999999999	13:00:00	10:30:00
5	10004	http://example.com/latecheckin	Downtown Plaza Hotel	0777777777	18:00:00	08:00:00
6	10005	http://example.com/max1	Super Skyscraper Hotel	0111111111	12:00:00	10:00:00
7	10006	http://example.com/max2	Luxury Haven Resort	0222222222	13:30:00	11:00:00
8	10007	http://example.com/max3	Ultimate Tower Hotel	0333333333	15:00:00	12:30:00
9	10008	http://example.com/max4	Midnight Express Hotel	0444444444	23:59:00	00:01:00
10	10009	http://example.com/hotel_standard2	Sunset Paradise Resort	0555555555	15:00:00	11:30:00

Total rows: 10 | Query complete 00:00:00.061 | CRLF | Ln 1, Col 21

## 2.2.2) ผลที่ปรากฏใน hotel\_facilities

Query Query History

```
1 select * from hotel_facilities;
```

Data Output Messages Notifications

Showing rows: 1 to 10 | Page No: 1 of 1 |

	hotel_id [PK] integer	wifi character varying (256)	pool character varying (256)	valet_parking character varying (256)
1	10000	Yes	Yes	Yes
2	10001	[default]	[default]	[default]
3	10002	[default]	[default]	[default]
4	10003	Yes	No	No
5	10004	Yes	Yes	Yes
6	10005	Yes	Yes	Yes
7	10006	Yes	Yes	Yes
8	10007	Yes	Yes	Yes
9	10008	Yes	Yes	Yes
10	10009	Yes	Yes	No

Total rows: 10 | Query complete 00:00:00.090 | CRLF | Ln 1, Col 31

## 2.2.3) ผลที่ปรากฏใน edit\_overall\_info เป็นผลมาจากการ insert/edit จะเก็บ timestamp ของการเปลี่ยนแปลงของข้อมูลในตารางด้วย

Query History

```
1 select * from edit_overall_info;
```

Data Output Messages Notifications

	hotel_id	user_id	edit_timestamp
	integer	integer	timestamp without time zone
1	10000	900000007	[null]
2	10001	900000008	[null]
3	10002	900000009	[null]
4	10003	900000009	[null]
5	10004	900000004	[null]
6	10005	900000005	[null]
7	10006	900000005	[null]
8	10007	900000007	[null]
9	10008	900000008	[null]
10	10009	900000002	[null]

Total rows: 10 Query complete 00:00:00.066 CRLF Ln 1, Col 15

\*edit\_timestamp เป็นค่า [null] เพราะยังไม่ได้ call function ในการใส่ timestamp จากการ insert ลงไป

## 2.2.4) ผลที่ปรากฏใน room

Query History

```
1 select * from room;
```

Data Output Messages Notifications

	hotel_id	room_id	min_price	max_price	status	room_type
	[PK] integer	[PK] integer	integer	integer	boolean	character varying (64)
1	10000	201	800	1500	true	Cheap
2	10000	202	800	1500	true	Cheap
3	10000	203	800	1500	true	Cheap
4	10000	204	800	1500	true	Cheap
5	10000	205	800	1500	true	Cheap
6	10000	206	800	1500	true	Cheap
7	10000	207	800	1500	true	Cheap
8	10000	208	800	1500	true	Cheap
9	10000	209	800	1500	true	Cheap
10	10000	210	800	1500	true	Cheap
11	10000	211	800	1500	true	Cheap
12	10000	212	800	1500	true	Cheap
13	10000	213	800	1500	true	Cheap
14	10000	214	800	1500	true	Cheap
15	10000	215	800	1500	true	Cheap
16	10000	216	800	1500	true	Cheap

Total rows: 15041 Query complete 00:00:00.081 CRLF Ln 1, Col 19

## 2.2.5) ผลที่ปรากฏใน room\_facilities

Query History

```
1 select * from room_facilities;
```

Data Output Messages Notifications

Showing rows: 1 to 1000 | Page No: 1 of 32 | ▶◀◀▶▶

	hotel_id integer	room_id integer	room_facilities character varying (256)
1	10000	201	tv
2	10000	202	tv
3	10000	203	tv
4	10000	204	tv
5	10000	205	tv
6	10000	206	tv
7	10000	207	tv
8	10000	208	tv
9	10000	209	tv
10	10000	210	tv
11	10000	211	tv
12	10000	212	tv
13	10000	213	tv
14	10000	214	tv
15	10000	215	tv
16	10000	216	tv

Total rows: 31394 | Query complete 00:00:00.074 | CRLF | Ln 1, Col 30

### 3. test command insert หรือ update

#### 3.1) ทดสอบคำสั่งจอง

คำอธิบาย: เป็นคำสั่งจอง ซึ่งเป็นการ Insert ข้อมูลลงไปใน booking\_transaction

ส่วนของ SQL command:

1. *insert/util\_gen\_booking\_id.sql*
2. *insert/reguser\_insert\_booking.sql*
3. *sample\_data/reguser\_insert\_booking.sql*

คำแนะนำ:

ผลการรัน:

Query History

```
1 select * from booking_transaction;
```

Data Output Messages Notifications

Showing rows: 1 to 20 Page No: 1 of 1

	booking_id [PK] integer	booking_date date	duration integer	check_in_date date	check_out_date date	user_id integer	hotel_id integer	room_id integer
1	100000001	2025-02-03	2	2025-02-10	2025-02-12	900000001	10000	201
2	100000002	2025-02-03	2	2025-02-10	2025-02-12	900000001	10001	202
3	100000003	2025-02-03	2	2025-02-10	2025-02-12	900000001	10002	203
4	100000004	2025-02-03	2	2025-02-10	2025-02-12	900000001	10003	201
5	100000005	2025-02-03	2	2025-02-10	2025-02-12	900000001	10004	202
6	100000006	2025-02-03	2	2025-02-10	2025-02-12	900000001	10005	203
7	100000007	2025-02-03	2	2025-02-10	2025-02-12	900000001	10006	201
8	100000008	2025-02-03	2	2025-02-10	2025-02-12	900000001	10007	202
9	100000009	2025-02-03	2	2025-02-10	2025-02-12	900000001	10008	203
10	100000010	2025-02-03	2	2025-02-10	2025-02-12	900000001	10009	201
11	100000011	2025-02-03	2	2025-02-10	2025-02-12	500000009	10000	301
12	100000012	2025-02-03	2	2025-02-10	2025-02-12	500000002	10001	302
13	100000013	2025-02-03	2	2025-02-10	2025-02-12	500000003	10002	303
14	100000014	2025-02-03	2	2025-02-10	2025-02-12	500000005	10003	301
15	100000015	2025-02-03	2	2025-02-10	2025-02-12	500000007	10004	302
16	100000016	2025-02-03	2	2025-02-10	2025-02-12	500000008	10005	303
17	100000017	2025-02-03	2	2025-02-10	2025-02-12	500000010	10006	201

Total rows: 20 Query complete 00:00:00.059 CRLF Ln 1, Col 35

Query History

```
1 select * from room where status = false;
```

Data Output Messages Notifications

Showing rows: 1 to 20 Page No: 1 of 1

	hotel_id [PK] integer	room_id [PK] integer	min_price integer	max_price integer	status boolean	room_type character varying (64)
1	10000	201	800	1500	false	Cheap
2	10001	202	200	300	false	Cheap
3	10002	203	700	1400	false	Cheap
4	10003	201	900	1600	false	Cheap
5	10004	202	3000	3800	false	Expensive
6	10005	203	1500	3000	false	Cheap
7	10006	201	25000	30000	false	Expensive
8	10007	202	22000	27000	false	Expensive
9	10008	203	1200	2200	false	Cheap
10	10009	201	700	1400	false	Cheap
11	10000	301	800	1500	false	Cheap
12	10001	302	400	500	false	Medium
13	10002	303	1600	2200	false	Medium
14	10003	301	900	1600	false	Cheap
15	10004	302	750	1300	false	Cheap
16	10005	303	1500	3000	false	Cheap
17	10006	301	20000	20000	false	Expensive

Total rows: 20 Query complete 00:00:00.094 CRLF Ln 1, Col 40

### 3.2) ทดสอบการแก้ไขการจอง

**คำอธิบาย:** เป็นคำสั่งแก้ไข ซึ่งเป็นการที่ regUser/admin จะ Update ข้อมูลลงไปใน booking\_transaction

**ส่วนของ SQL command:**

1. edit/user\_edit\_booking\_hotel.sql
2. edit/user\_edit\_booking\_date.sql
3. sample\_data/Delete Booking Test/user\_edit\_test.sql (ข้อมูลทดสอบ)

#### 4. sample\_data\Delete Booking Test\user\_edit\_hotel\_test.sql(ข้อมูลทดสอบ)

คำแนะนำ: รันทีละบรรทัด

ผลการรัน:

ตาราง booking\_transaction

	booking_id [PK] integer	booking_date date	duration integer	check_in_date date	check_out_date date	user_id integer	hotel_id integer	room_id integer
4	100000001	2025-02-04	2	2025-02-10	2025-02-12	500000009	10000	301
5	1000000012	2025-02-04	2	2025-02-10	2025-02-12	500000002	10001	302
6	1000000013	2025-02-04	2	2025-02-10	2025-02-12	500000003	10002	303
7	1000000014	2025-02-04	2	2025-02-10	2025-02-12	500000005	10003	301
8	1000000015	2025-02-04	2	2025-02-10	2025-02-12	500000007	10004	302
9	1000000016	2025-02-04	2	2025-02-10	2025-02-12	500000008	10005	303
10	1000000017	2025-02-04	2	2025-02-10	2025-02-12	500000010	10006	301
11	1000000018	2025-02-04	2	2025-02-10	2025-02-12	500000011	10007	302
12	1000000019	2025-02-04	2	2025-02-10	2025-02-12	500000012	10008	303
13	1000000020	2025-02-04	2	2025-02-10	2025-02-12	500000004	10009	301
14	100000001	2025-02-04	2	2025-02-10	2025-02-12	900000001	10001	201
15	100000002	2025-02-04	1	2025-02-11	2025-02-12	900000001	10002	201
16	100000003	2025-02-04	3	2025-02-09	2025-02-12	900000001	10003	202
17	100000004	2025-02-04	2	2025-02-12	2025-02-14	900000001	10001	203
18	100000005	2025-02-04	2	2025-02-10	2025-02-12	900000001	10001	204
19	100000006	2025-02-04	3	2025-02-10	2025-02-13	900000001	10001	205
20	100000007	2025-02-04	2	2025-02-10	2025-02-12	900000001	10001	206

Total rows: 20 Query complete 00:00:00.061

CRLF Ln 1, Col 34

ตาราง manages\_booking

	user_id integer	booking_id integer	edit_timestamp timestamp without time zone
1	900000001	100000002	2025-02-04 07:15:51.439798
2	900000004	100000003	2025-02-04 07:15:51.439798
3	900000007	100000004	2025-02-04 07:15:51.439798
4	900000003	100000006	2025-02-04 07:15:51.439798
5	500000001	100000001	2025-02-04 07:16:30.642481
6	500000002	100000002	2025-02-04 07:16:33.472395
7	900000003	100000003	2025-02-04 07:16:36.041906
8	500000004	100000004	2025-02-04 07:16:52.826196
9	900000005	100000005	2025-02-04 07:16:56.06009
10	500000006	100000006	2025-02-04 07:16:59.411359
11	500000009	100000006	2025-02-04 07:17:02.70119
12	900000001	100000007	2025-02-04 07:17:05.474748

Total rows: 12 Query complete 00:00:00.073

CRLF Ln 1, Col 30

3.3) ทดสอบการแก้ไขการข้อมูลโรงเรม

**คำอธิบาย:** เป็นคำสั่งของ ชื่นการที่ regUser/admin จะ Update ข้อมูลลงไปใน hotel และ table ที่เกี่ยวข้อง

#### ส่วนของ SQL command:

1. edit/admin\_update\_hotel\_info
2. sample\_date/update\_hotel\_overall\_info

#### คำแนะนำ:

#### ผลการรัน:

##### 3.3.1) ก่อนแก้ไขข้อมูล

Data Output Messages Notifications

	hotel_id [PK] integer	map_url character varying	hotel_location character varying (256)	hotel_phonenumber character varying (10)	check_in_time time without time zone	check_out_time time without time zone
1	10000	http://example.com/hotel1	Grand Palace Hotel	0812345678	14:00:00	12:00:00
2	10001	[null]	Cozy Cabin Inn	[null]	[null]	[null]
3	10002	[null]	Beachfront Resort	0888888888	15:30:00	11:00:00
4	10003	http://example.com/hotel4	Mountain View Lodge	0999999999	13:00:00	10:30:00
5	10004	http://example.com/latecheckin	Downtown Plaza Hotel	0777777777	18:00:00	08:00:00
6	10005	http://example.com/max1	Super Skyscraper Hotel	0111111111	12:00:00	10:00:00
7	10006	http://example.com/max2	Luxury Haven Resort	0222222222	13:30:00	11:00:00
8	10007	http://example.com/max3	Ultimate Tower Hotel	0333333333	15:00:00	12:30:00
9	10008	http://example.com/max4	Midnight Express Hotel	0444444444	23:59:00	00:01:00
10	10009	http://example.com/hotel_standard2	Sunset Paradise Resort	0555555555	15:00:00	11:30:00

##### 3.3.2) หลังแก้ไขข้อมูล

Query Query History

```
1 select * from hotel;
```

Data Output Messages Notifications

	hotel_id [PK] integer	map_url character varying	hotel_location character varying (256)	hotel_phonenumber character varying (10)	check_in_time time without time zone	check_out_time time without time zone
1	10001	[null]	Cozy Cabin Inn	[null]	[null]	[null]
2	10003	http://example.com/hotel4	Mountain View Lodge	0999999999	13:00:00	10:30:00
3	10005	http://example.com/max1	Super Skyscraper Hotel	0111111111	12:00:00	10:00:00
4	10006	http://example.com/max2	Luxury Haven Resort	0222222222	13:30:00	11:00:00
5	10008	http://example.com/max4	Midnight Express Hotel	0444444444	23:59:00	00:01:00
6	10000	http://newexample.com/hotel1	Grand Palace Hotel	0999999999	14:00:00	12:00:00
7	10002	[null]	Updated Beachfront Resort	0888888888	16:00:00	10:30:00
8	10004	http://example.com/latecheckin	Downtown Plaza Hotel	0777777777	18:00:00	08:00:00
9	10007	http://updated.com/max3	Updated Ultimate Tower	0888888888	14:00:00	11:45:00
10	10009	http://example.com/hotel_standard2	Sunset Paradise Resort	0555555555	15:00:00	12:00:00

#### 4. test command view

##### 4.1) ทดสอบการดู booking

**คำอธิบาย:** พิมพ์ชื่น admin\_view\_booking คือพิมพ์ชื่นที่ใช้ได้ทั้งกับ registered user และ admin โดยเป็นพิมพ์ชื่นที่รับ parameter 3 ตัว ได้แก่ admin\_status BOOLEAN, action\_type VARCHAR และ search\_value VARCHAR โดย

- `admin_status` จะบ่งบอกถึงสถานะผู้ใช้ฟังก์ชัน หากเป็น `true` หมายความว่า ผู้เรียกใช้ฟังก์ชันมีสถานะเป็น `admin` และหากเป็น `false` หมายความว่า ผู้เรียกใช้ฟังก์ชันมีค่าเป็น `registered user`
- `action_type` จะบ่งบอกถึง `data` ที่จะ `filter`
- `search_value` จะบ่งบอกถึงค่าที่ต้องการหา

ส่วนของ SQL command: `view/user_view_booking.sql`

คำแนะนำ: ควรรันเทสเคสใน sample data ให้ครบก่อนจึงจะทดลองเรียกใช้ฟังก์ชันตามแต่ละกรณี ผลการรัน:

#### 4.1.1) view ด้วยสิทธิ์ admin (ดูได้ทั้งหมด)

- กรณีไม่กำหนด filter

Showing rows: 1 to 32								
	booking_id	booking_date	check_in_date	check_out_date	duration	user_id	hotel_id	room_id
1	1010	2025-02-10	2025-04-01	2025-04-05	4	500000005	5	504
2	1009	2025-02-09	2025-03-25	2025-03-27	2	500000004	4	401
3	1008	2025-02-08	2025-03-20	2025-03-21	1	500000003	1	105
4	1007	2025-02-07	2025-03-15	2025-03-18	3	500000002	3	305
5	1006	2025-02-06	2025-03-10	2025-03-12	2	500000001	2	202
6	1005	2025-02-05	2025-03-01	2025-03-05	4	500000005	5	502
7	1004	2025-02-04	2025-02-20	2025-02-25	5	500000004	4	404
8	100000008	2025-02-03	2025-02-10	2025-02-12	2	900000001	10007	202
9	100000009	2025-02-03	2025-02-10	2025-02-12	2	900000001	10008	203
10	100000010	2025-02-03	2025-02-10	2025-02-12	2	900000001	10009	201
11	100000011	2025-02-03	2025-02-10	2025-02-12	2	500000009	10000	301
12	100000012	2025-02-03	2025-02-10	2025-02-12	2	500000002	10001	302
13	100000013	2025-02-03	2025-02-10	2025-02-12	2	500000003	10002	303
14	100000014	2025-02-03	2025-02-10	2025-02-12	2	500000005	10003	301
15	100000015	2025-02-03	2025-02-10	2025-02-12	2	500000007	10004	302
16	100000016	2025-02-03	2025-02-10	2025-02-12	2	500000008	10005	303
17	100000017	2025-02-03	2025-02-10	2025-02-12	2	500000010	10006	301
18	100000018	2025-02-03	2025-02-10	2025-02-12	2	500000011	10007	302
19	100000019	2025-02-03	2025-02-10	2025-02-12	2	500000012	10008	303
20	100000020	2025-02-03	2025-02-10	2025-02-12	2	500000004	10009	301
21	1003	2025-02-03	2025-02-15	2025-02-16	1	500000003	3	301
22	10000001	2025-02-03	2025-02-10	2025-02-12	2	900000001	10000	201
23	100000002	2025-02-03	2025-02-10	2025-02-12	2	900000001	10001	202
24	100000003	2025-02-03	2025-02-10	2025-02-12	2	900000001	10002	203
25	100000004	2025-02-03	2025-02-10	2025-02-12	2	900000001	10003	201

- กรณีต้องการค้นหาตาม booking id

Showing rows: 1 to 1								
	booking_id	booking_date	check_in_date	check_out_date	duration	user_id	hotel_id	room_id
1	1004	2025-02-04	2025-02-20	2025-02-25	5	500000004	4	404

- กรณีต้องการค้นหาตาม duration

Query    Query History

```

3 --Case4 admin view data by filtering duration
4 select * from admin_view_booking(True,'duration','2')

```

Data Output    Messages    Notifications

SQL

	booking_id integer	booking_date date	check_in_date date	check_out_date date	duration integer	user_id integer	hotel_id integer	room_id integer
1	100000001	2025-02-03	2025-02-10	2025-02-12	2	900000001	10000	201
2	100000002	2025-02-03	2025-02-10	2025-02-12	2	900000001	10001	202
3	100000003	2025-02-03	2025-02-10	2025-02-12	2	900000001	10002	203
4	100000004	2025-02-03	2025-02-10	2025-02-12	2	900000001	10003	201
5	100000005	2025-02-03	2025-02-10	2025-02-12	2	900000001	10004	202
6	100000006	2025-02-03	2025-02-10	2025-02-12	2	900000001	10005	203
7	100000007	2025-02-03	2025-02-10	2025-02-12	2	900000001	10006	201
8	100000008	2025-02-03	2025-02-10	2025-02-12	2	900000001	10007	202
9	100000009	2025-02-03	2025-02-10	2025-02-12	2	900000001	10008	203
10	100000010	2025-02-03	2025-02-10	2025-02-12	2	900000001	10009	201
11	100000011	2025-02-03	2025-02-10	2025-02-12	2	500000009	10000	301
12	100000012	2025-02-03	2025-02-10	2025-02-12	2	500000002	10001	302
13	100000013	2025-02-03	2025-02-10	2025-02-12	2	500000003	10002	303
14	100000014	2025-02-03	2025-02-10	2025-02-12	2	500000005	10003	301
15	100000015	2025-02-03	2025-02-10	2025-02-12	2	500000007	10004	302
16	100000016	2025-02-03	2025-02-10	2025-02-12	2	500000008	10005	303
17	100000017	2025-02-03	2025-02-10	2025-02-12	2	500000010	10006	301
18	100000018	2025-02-03	2025-02-10	2025-02-12	2	500000011	10007	302
19	100000019	2025-02-03	2025-02-10	2025-02-12	2	500000012	10008	303
20	100000020	2025-02-03	2025-02-10	2025-02-12	2	500000004	10009	301
21	1001	2025-02-01	2025-02-05	2025-02-07	2	500000001	1	101
22	1006	2025-02-06	2025-03-10	2025-03-12	2	500000001	2	202
23	1009	2025-02-09	2025-03-25	2025-03-27	2	500000004	4	401
24	1011	2025-02-01	2025-02-05	2025-02-07	2	500000001	1	201

- กรณีต้องการค้นหาตาม check in date

```

1 --Case5 admin view data by filtering check in date
2 v select * from admin_view_booking(True,'check_in_date','2025-02-10')

```

Data Output Messages Notifications

	booking_id integer	booking_date date	check_in_date date	check_out_date date	duration integer	user_id integer	hotel_id integer	room_id integer
1	100000001	2025-02-03	2025-02-10	2025-02-12	2	900000001	10000	201
2	100000002	2025-02-03	2025-02-10	2025-02-12	2	900000001	10001	202
3	100000003	2025-02-03	2025-02-10	2025-02-12	2	900000001	10002	203
4	100000004	2025-02-03	2025-02-10	2025-02-12	2	900000001	10003	201
5	100000005	2025-02-03	2025-02-10	2025-02-12	2	900000001	10004	202
6	100000006	2025-02-03	2025-02-10	2025-02-12	2	900000001	10005	203
7	100000007	2025-02-03	2025-02-10	2025-02-12	2	900000001	10006	201
8	100000008	2025-02-03	2025-02-10	2025-02-12	2	900000001	10007	202
9	100000009	2025-02-03	2025-02-10	2025-02-12	2	900000001	10008	203
10	100000010	2025-02-03	2025-02-10	2025-02-12	2	900000001	10009	201
11	100000011	2025-02-03	2025-02-10	2025-02-12	2	500000009	10000	301
12	100000012	2025-02-03	2025-02-10	2025-02-12	2	500000002	10001	302
13	100000013	2025-02-03	2025-02-10	2025-02-12	2	500000003	10002	303
14	100000014	2025-02-03	2025-02-10	2025-02-12	2	500000005	10003	301
15	100000015	2025-02-03	2025-02-10	2025-02-12	2	500000007	10004	302
16	100000016	2025-02-03	2025-02-10	2025-02-12	2	500000008	10005	303
17	100000017	2025-02-03	2025-02-10	2025-02-12	2	500000010	10006	301
18	100000018	2025-02-03	2025-02-10	2025-02-12	2	500000011	10007	302
19	100000019	2025-02-03	2025-02-10	2025-02-12	2	500000012	10008	303
20	100000020	2025-02-03	2025-02-10	2025-02-12	2	500000004	10009	301
21	1002	2025-02-02	2025-02-10	2025-02-13	3	500000002	2	203

#### - กรณีต้องการค้นหาตาม check out date

Query Query History

```

1 --Case6 admin view data by filtering check out date
2 v select * from admin_view_booking(True,'check_out_date','2025-02-07')

```

Data Output Messages Notifications

	booking_id integer	booking_date date	check_in_date date	check_out_date date	duration integer	user_id integer	hotel_id integer	room_id integer
1	1001	2025-02-01	2025-02-05	2025-02-07	2	500000001	1	101
2	1011	2025-02-01	2025-02-05	2025-02-07	2	500000001	1	201
3	1021	2025-02-01	2025-02-05	2025-02-07	2	500000001	2	201

#### - กรณีต้องการค้นหาตาม booking\_date

```

1 --Case7 admin view data by filtering booking date
2 v select * from admin_view_booking(True,'booking_date','2025-02-01')

```

Data Output Messages Notifications

	booking_id integer	booking_date date	check_in_date date	check_out_date date	duration integer	user_id integer	hotel_id integer	room_id integer
1	1001	2025-02-01	2025-02-05	2025-02-07	2	500000001	1	101
2	1011	2025-02-01	2025-02-05	2025-02-07	2	500000001	1	201
3	1021	2025-02-01	2025-02-05	2025-02-07	2	500000001	2	201

#### - กรณีต้องการค้นหาตาม user\_id

Query Query History

```
1 --Case8 admin view data by filtering user_id
2 select * from admin_view_booking(True,'user_id','500000008')
```

Data Output Messages Notifications

booking\_id booking\_date check\_in\_date check\_out\_date duration user\_id hotel\_id room\_id

	integer	date	date	date	integer	integer	integer	
1	100000016	2025-02-03	2025-02-10	2025-02-12	2	500000008	10005	303

- กรณีต้องการค้นหาตาม hotel id

Query Query History

```
1 --Case9 admin view data by filtering hotel_id
2 select * from admin_view_booking(True,'hotel_id','10006')
```

Data Output Messages Notifications

booking\_id booking\_date check\_in\_date check\_out\_date duration user\_id hotel\_id room\_id

	integer	date	date	date	integer	integer	integer	
1	100000007	2025-02-03	2025-02-10	2025-02-12	2	900000001	10006	201
2	100000017	2025-02-03	2025-02-10	2025-02-12	2	500000010	10006	301

- กรณีต้องการค้นหาตาม room id

Query Query History

```
1 --Case10 admin view data by filtering room_id
2 select * from admin_view_booking(True,'room_id','203')
```

Data Output Messages Notifications

booking\_id booking\_date check\_in\_date check\_out\_date duration user\_id hotel\_id room\_id

	integer	date	date	date	integer	integer	integer	
1	100000003	2025-02-03	2025-02-10	2025-02-12	2	900000001	10002	203
2	100000006	2025-02-03	2025-02-10	2025-02-12	2	900000001	10005	203
3	100000009	2025-02-03	2025-02-10	2025-02-12	2	900000001	10008	203
4	1002	2025-02-02	2025-02-10	2025-02-13	3	500000002		203

#### 4.1.2) view ด้วยสิทธิ์ regUser/unRegUser (จำกัดให้ดูได้บางส่วน)

```
1 --Case1 : reguser view their own booking via booking id
2 select * from admin_view_booking(False,null,'100000001')
3
```

Data Output Messages Notifications

booking\_id booking\_date check\_in\_date check\_out\_date duration user\_id hotel\_id room\_id

	integer	date	date	date	integer	integer	integer	
1	100000001	2025-02-03	2025-02-10	2025-02-12	2	900000001	10000	201

#### 4.2) ทดสอบการดู hotel info

**คำอธิบาย:** เป็นคำสั่งของ ชีํกการที่ regUser/admin จะ Update ข้อมูลลงไปใน hotel และ table ที่เกี่ยวข้อง

### ส่วนของ SQL command:

1.view\view\_hotel\_info.sql

2.sample\_data\view\_hotel\_info\_test.sql(sample test)

**คำแนะนำ:** run sample date ที่ลับรัหด แล้วดูผลลัพธ์ที่ table view\_hotel\_info

ผลการรัน:

Query History										
Data Output										
	hotel_id	map_url	hotel_location	hotel_phonenumber	check_in_time	check_out_time	wifi	pool	valet_parking	available_room
1	10001	[null]	Cozy Cabin Inn	[null]	[null]	[null]	[null]	[null]	[null]	43

  

Query History										
Data Output										
	hotel_id	map_url	hotel_location	hotel_phonenumber	check_in_time	check_out_time	wifi	pool	valet_parking	available_room
1	10002	[null]	Updated Beachfront Resort	0888888888	16:00:00	10:30:00	[null]	[null]	[null]	72

  

Query History										
Data Output										
	hotel_id	map_url	hotel_location	hotel_phonenumber	check_in_time	check_out_time	wifi	pool	valet_parking	available_room
1	10003	http://example.com/hotel4	Mountain View Lodge	0999999999	13:00:00	10:30:00	Yes	No	No	105

  

Query History										
Data Output										
	hotel_id	map_url	hotel_location	hotel_phonenumber	check_in_time	check_out_time	wifi	pool	valet_parking	available_room
1	004	http://example.com/latecheckin	Downtown Plaza Hotel	0777777777	18:00:00	08:00:00	Available	Not Available	Available	196

  

Query History										
Data Output										
	hotel_id	map_url	hotel_location	hotel_phonenumber	check_in_time	check_out_time	wifi	pool	valet_parking	available_room
1	10005	http://example.com/max1	Super Skyscraper Hotel	0111111111	12:00:00	10:00:00	Yes	Yes	Yes	4997

Query History

```
1 select * from view_hotel_info
```

Data Output Messages Notifications

	hotel_id integer	user_id integer	view_timestamp timestamp without time zone
1	10001	100000001	2025-02-04 07:39:43.231625
2	10002	100000002	2025-02-04 07:39:45.301255
3	10003	500000003	2025-02-04 07:39:47.624605
4	10004	500000004	2025-02-04 07:39:50.004812
5	10005	900000005	2025-02-04 07:39:52.254605

## 5. test command delete

### 5.1) ทดสอบการลบคำสั่งจอง

**คำอธิบาย:** พังก์ชัน Delete\_Booking คือพังก์ชันที่ใช้ได้ทั้งกับ admin และ registered user เป็นพังก์ชันที่รับ parameter 2 ตัว ได้แก่ delete\_id INT และ deleter\_id INT สำหรับ admin จะลบได้ทุก booking\_transaction แต่ registered user จะลบได้เฉพาะ booking\_transaction ที่ตนเองเป็นผู้จองได้เท่านั้น โดย

- พังก์ชันจะตรวจสอบ deleter\_id ที่รับเข้ามา มีสิทธิ์ admin หรือไม่ ถ้ามีพังก์ชันจะลบข้อมูลการจองโรงแรมที่มี booking\_id ตรงกับ delete\_id ที่รับเข้ามา
- ตรวจสอบ deleter\_id ที่รับเข้ามาแล้วไม่มีสิทธิ์ admin แต่เป็นผู้ใช้ที่เป็นเจ้าของ delete\_id ที่รับเข้ามา พังก์ชันจะลบข้อมูลการจองโรงแรมที่มี booking\_id ตรงกับ delete\_id และจะต้องมี user\_id ของผู้จองตรงกับ deleter\_id ที่รับเข้ามา
- ถ้าไม่เป็นไปตามเงื่อนไขก่อนหน้านี้ พังก์ชันจะไม่สามารถลบข้อมูลการจองได้

ส่วนของ SQL command:

1. edit/delete booking.sql
2. sample/test delete booking.sql

คำแนะนำ:

ผลการรัน:

#### 5.1.1) ผลก่อนลบ

Data Output Messages Notifications

Showing rows: 1 to 20 | Page No: 1 of 1 | < << >> > | X

	booking_id [PK] integer	booking_date date	duration integer	check_in_date date	check_out_date date	user_id integer	hotel_id integer	room_id integer
2	100000009	2025-02-04	2	2025-02-10	2025-02-12	900000001	10008	203
3	100000010	2025-02-04	2	2025-02-10	2025-02-12	900000001	10009	201
4	100000011	2025-02-04	2	2025-02-10	2025-02-12	500000009	10000	301
5	100000012	2025-02-04	2	2025-02-10	2025-02-12	500000002	10001	302
6	100000013	2025-02-04	2	2025-02-10	2025-02-12	500000003	10002	303
7	100000014	2025-02-04	2	2025-02-10	2025-02-12	500000005	10003	301
8	100000015	2025-02-04	2	2025-02-10	2025-02-12	500000007	10004	302
9	100000016	2025-02-04	2	2025-02-10	2025-02-12	500000008	10005	303
10	100000017	2025-02-04	2	2025-02-10	2025-02-12	500000010	10006	301
11	100000018	2025-02-04	2	2025-02-10	2025-02-12	500000011	10007	302
12	100000019	2025-02-04	2	2025-02-10	2025-02-12	500000012	10008	303
13	100000020	2025-02-04	2	2025-02-10	2025-02-12	500000004	10009	301
14	100000001	2025-02-04	2	2025-02-10	2025-02-12	900000001	10001	201
15	100000002	2025-02-04	1	2025-02-11	2025-02-12	900000001	10002	201
16	100000003	2025-02-04	3	2025-02-09	2025-02-12	900000001	10003	202
17	100000004	2025-02-04	2	2025-02-12	2025-02-14	900000001	10001	203
18	100000005	2025-02-04	2	2025-02-10	2025-02-12	900000001	10001	204
19	100000006	2025-02-04	3	2025-02-10	2025-02-13	900000001	10001	205
20	100000007	2025-02-04	2	2025-02-10	2025-02-12	900000001	10001	206

Total rows: 20 Query complete 00:00:00.048 CRLF Ln 1, Col 27

### 5.1.2) ผลหลังลบ

#### Query Query History

```
1 select * from booking_transaction
```

#### Data Output Messages Notifications

Showing rows: 1 to 15 | Page No: 1 of 1 | < << >> > | X

	booking_id [PK] integer	booking_date date	duration integer	check_in_date date	check_out_date date	user_id integer	hotel_id integer	room_id integer
1	100000008	2025-02-04	2	2025-02-10	2025-02-12	900000001	10007	202
2	100000009	2025-02-04	2	2025-02-10	2025-02-12	900000001	10008	203
3	100000011	2025-02-04	2	2025-02-10	2025-02-12	500000009	10000	301
4	100000012	2025-02-04	2	2025-02-10	2025-02-12	500000002	10001	302
5	100000013	2025-02-04	2	2025-02-10	2025-02-12	500000003	10002	303
6	100000014	2025-02-04	2	2025-02-10	2025-02-12	500000005	10003	301
7	100000016	2025-02-04	2	2025-02-10	2025-02-12	500000008	10005	303
8	100000017	2025-02-04	2	2025-02-10	2025-02-12	500000010	10006	301
9	100000018	2025-02-04	2	2025-02-10	2025-02-12	500000011	10007	302
10	100000020	2025-02-04	2	2025-02-10	2025-02-12	500000004	10009	301
11	100000002	2025-02-04	1	2025-02-11	2025-02-12	900000001	10002	201
12	100000003	2025-02-04	3	2025-02-09	2025-02-12	900000001	10003	202
13	100000004	2025-02-04	2	2025-02-12	2025-02-14	900000001	10001	203
14	100000006	2025-02-04	3	2025-02-10	2025-02-13	900000001	10001	205
15	100000007	2025-02-04	2	2025-02-10	2025-02-12	900000001	10001	206

### 5.2) ทดสอบการลบโรงแรม

คำอธิบาย: พิมพ์ชื่อ `delete_hotel_overall_info` คือพิมพ์ชื่อที่ใช้ได้เฉพาะ admin เท่านั้น โดยเป็นพิมพ์ชื่อที่รับ parameter 2 ตัว ได้แก่ `admin_id INT` และ `p_hotel_id INT` โดย

- พังก์ชันจะตรวจสอบว่า admin\_id ที่รับเข้ามา มีสิทธิ์ admin หรือไม่ ถ้าไม่มีจะไม่สามารถลบข้อมูลของโรงแรมได้
- หลังจากการตรวจสอบสิทธิ์ admin พังก์ชันจะตรวจสอบว่า p\_hotel\_id มีอยู่จริงหรือไม่ ถ้าไม่มีจะไม่สามารถลบข้อมูลได้
- ถ้า admin\_id และ p\_hotel\_id ถูกต้องตามเงื่อนไขแล้ว พังก์ชันจะลบข้อมูลของโรงแรมที่มี hotel\_id ตรงกับ parameter ที่รับมา

#### ส่วนของ SQL command:

1. edit/delete\_hotel\_overall\_info.sql
2. sample/delete\_hotel\_overall\_info.sql

#### คำแนะนำ:

##### ผลการรัน:

###### 5.1.1) ผลก่อนลบ

###### ตาราง hotel

Query Query History

```
1 select * from hotel
```

Data Output Messages Notifications

	hotel_id [PK] integer	map_url character varying	hotel_location character varying (256)	hotel_phonenumber character varying (10)	check_in_time time without time zone	check_out_time time without time zone
1	10000	http://example.com/hotel1	Grand Palace Hotel	0812345678	14:00:00	12:00:00
2	10001	[null]	Cozy Cabin Inn	[null]	[null]	[null]
3	10002	[null]	Beachfront Resort	0888888888	15:30:00	11:00:00
4	10003	http://example.com/hotel4	Mountain View Lodge	0999999999	13:00:00	10:30:00
5	10004	http://example.com/latecheckin	Downtown Plaza Hotel	0777777777	18:00:00	08:00:00
6	10005	http://example.com/max1	Super Skyscraper Hotel	0111111111	12:00:00	10:00:00
7	10006	http://example.com/max2	Luxury Haven Resort	0222222222	13:30:00	11:00:00
8	10007	http://example.com/max3	Ultimate Tower Hotel	0333333333	15:00:00	12:30:00
9	10008	http://example.com/max4	Midnight Express Hotel	0444444444	23:59:00	00:01:00
10	10009	http://example.com/hotel_standard2	Sunset Paradise Resort	0555555555	15:00:00	11:30:00

###### ตาราง room

**Query History**

```
1 select * from room
```

**Data Output** Messages Notifications

	hotel_id [PK] integer	room_id [PK] integer	min_price integer	max_price integer	status boolean	room_type character varying (64)
1	10000	201	800	1500	true	Cheap
2	10000	202	800	1500	true	Cheap
3	10000	203	800	1500	true	Cheap
4	10000	204	800	1500	true	Cheap
5	10000	205	800	1500	true	Cheap
6	10000	206	800	1500	true	Cheap
7	10000	207	800	1500	true	Cheap
8	10000	208	800	1500	true	Cheap
9	10000	209	800	1500	true	Cheap
10	10000	210	800	1500	true	Cheap
11	10000	211	800	1500	true	Cheap
12	10000	212	800	1500	true	Cheap
13	10000	213	800	1500	true	Cheap
14	10000	214	800	1500	true	Cheap
15	10000	215	800	1500	true	Cheap
16	10000	216	800	1500	true	Cheap

Total rows: 15041 Query complete 00:00:00.091 CRLF Ln 1, Col 19

**ตาราง room\_facilities**

**Query History**

```
1 select * from room_facilities
```

**Data Output** Messages Notifications

	hotel_id integer	room_id integer	room_facilities character varying (256)
1	10000	201	tv
2	10000	202	tv
3	10000	203	tv
4	10000	204	tv
5	10000	205	tv
6	10000	206	tv
7	10000	207	tv
8	10000	208	tv
9	10000	209	tv
10	10000	210	tv
11	10000	211	tv
12	10000	212	tv
13	10000	213	tv
14	10000	214	tv
15	10000	215	tv
16	10000	216	tv

Total rows: 31394 Query complete 00:00:00.080 CRLF Ln 1, Col 30

**5.1.2) ผลหลังlab**

**ตาราง hotel**

Query Query History

```
1 select * from hotel
```

Data Output Messages Notifications

	hotel_id [PK] integer	map_url character varying	hotel_location character varying (256)	hotel_phonenumber character varying (10)	check_in_time time without time zone	check_out_time time without time zone
1	10002	[null]	Beachfront Resort	0888888888	15:30:00	11:00:00
2	10006	http://example.com/max2	Luxury Haven Resort	0222222222	13:30:00	11:00:00
3	10007	http://example.com/max3	Ultimate Tower Hotel	0333333333	15:00:00	12:30:00
4	10008	http://example.com/max4	Midnight Express Hotel	0444444444	23:59:00	00:01:00
5	10009	http://example.com/hotel_standard2	Sunset Paradise Resort	0555555555	15:00:00	11:30:00

Total rows: 5 | Query complete 00:00:00.046 | CRLF | Ln 1, Col 20

### ຕາງ່າງ room

Query Query History

```
1 select * from room
```

Data Output Messages Notifications

	hotel_id [PK] integer	room_id [PK] integer	min_price integer	max_price integer	status boolean	room_type character varying (64)
1	10002	201	700	1400	true	Cheap
2	10002	202	700	1400	true	Cheap
3	10002	203	700	1400	true	Cheap
4	10002	204	700	1400	true	Cheap
5	10002	205	700	1400	true	Cheap
6	10002	206	700	1400	true	Cheap
7	10002	207	700	1400	true	Cheap
8	10002	208	700	1400	true	Cheap
9	10002	209	700	1400	true	Cheap
10	10002	210	700	1400	true	Cheap
11	10002	211	700	1400	true	Cheap
12	10002	212	700	1400	true	Cheap
13	10002	213	700	1400	true	Cheap
14	10002	214	700	1400	true	Cheap
15	10002	215	700	1400	true	Cheap
16	10002	301	1600	2200	true	Medium
17	10002	302	1600	2200	true	Medium

Total rows: 9485 | Query complete 00:00:00.077 | CRLF | Ln 1, Col 19

### ຕາງ່າງ room\_facilities

The screenshot shows a MySQL Workbench interface with a query editor and a results grid. The query is:

```
1 select * from room_facilities
```

The results grid has three columns: hotel\_id, room\_id, and room\_facilities. The data shows 20340 rows, all with hotel\_id 10002 and room\_facilities 'tv'. The results are paginated at the bottom.

	hotel_id	room_id	room_facilities
1	10002	201	tv
2	10002	202	tv
3	10002	203	tv
4	10002	204	tv
5	10002	205	tv
6	10002	206	tv
7	10002	207	tv
8	10002	208	tv
9	10002	209	tv
10	10002	210	tv
11	10002	211	tv
12	10002	212	tv
13	10002	213	tv
14	10002	214	tv
15	10002	215	tv
16	10002	301	tv
17	10002	301	tv

Total rows: 20340 Query complete 00:00:00.084 CRLF Ln 1, Col 30

## 6. test complex query

### 6.1) โรงแรมที่มียอดการจองสูงที่สุด

คำสั่ง SQL ที่ใช้:

```
SELECT
    h.hotel_id,
    h.hotel_location,
    COUNT(DISTINCT bt.booking_id) AS total_bookings,
    ROUND(AVG(bt.duration)) AS avg_stay_duration
FROM hotel h
LEFT JOIN booking_transaction bt ON h.hotel_id = bt.hotel_id
GROUP BY h.hotel_id, h.hotel_location
ORDER BY total_bookings DESC
LIMIT 10;
```

คำอธิบาย: เลือก 10 โรงแรมที่มียอดการจองสูงที่สุด

ส่วนของ SQL command:

1. sample\_data/insert\_booking for top10book.sql
2. complex\_query/top\_perform.sql

คำแนะนำ:

ผลการรัน:

Data Output Messages Notifications

Showing rows: 1 to 5 Page No: 1 of 1

	hotel_Id [PK] integer	hotel_Location character varying (256)	total_bookings bigint	avg_stay_duration numeric
1	10002	Beachfront Resort	6	2
2	10007	Ultimate Tower Hotel	5	2
3	10006	Luxury Haven Resort	4	2
4	10009	Sunset Paradise Resort	4	2
5	10008	Midnight Express Hotel	3	2

Total rows: 5 Query complete 00:00:00.057 CRLF Ln 10, Col 10

## 7. Dump data

ดูไฟล์ BackUp2