

ขั้นตอนการรันคำสั่ง

โจทย์ที่ 1: Hotel Booking

กลุ่ม Agado (กลุ่มที่ 6)

1. Create Table

ในขั้นตอนนี้มีไฟล์ *create/drop_all.sql* ใช้สำหรับการเริ่มการทำงานใหม่ตั้งแต่ต้น ไม่จำเป็นต้อง run

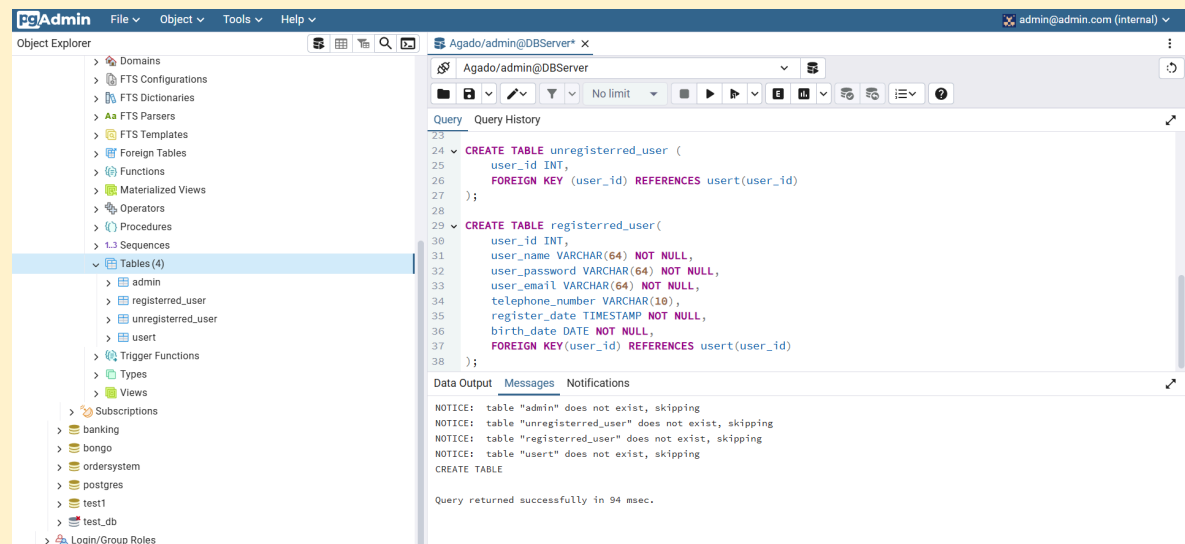
1.1) สร้าง user table และ subclass

คำอธิบาย: สร้าง table user (superclass) และ table admin, unregistered_user, registered_user (subclass ของ user) และ

ส่วนของ SQL command: *create/create_user.sql*

คำแนะนำ: ควร run ทีละคำสั่ง

ผลการรัน:



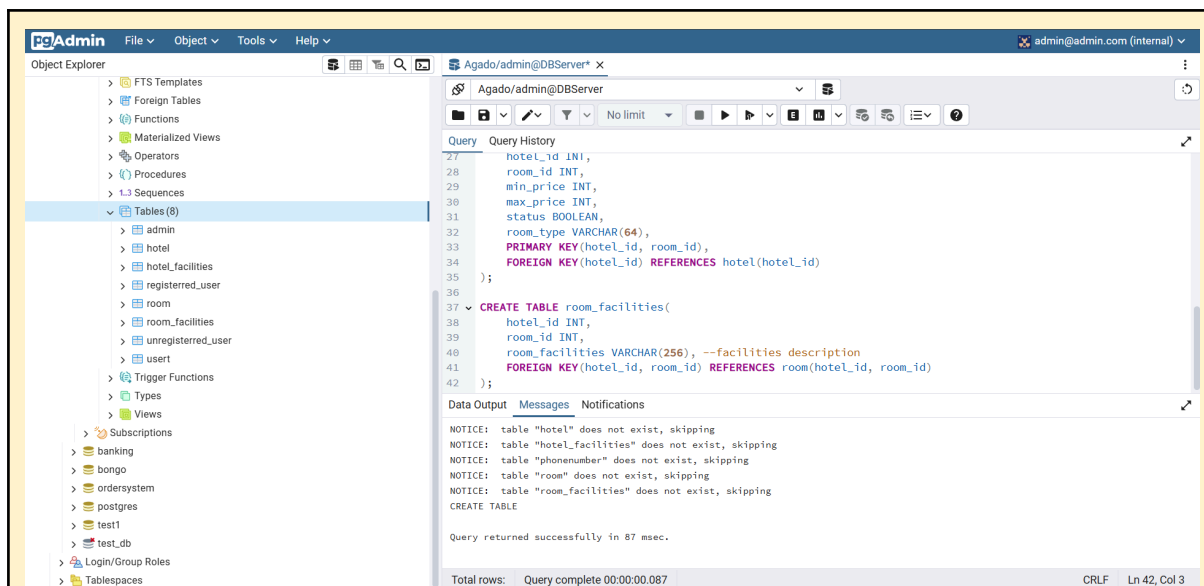
1.2) สร้าง hotel table และ subclass/relation ที่เกี่ยวข้อง

คำอธิบาย: สร้าง table hotel, hotel_facilities, room, room_facilities

ส่วนของ SQL command: *create/create_hotel.sql*

คำแนะนำ: ควร run ทีละคำสั่ง

ผลการรัน:



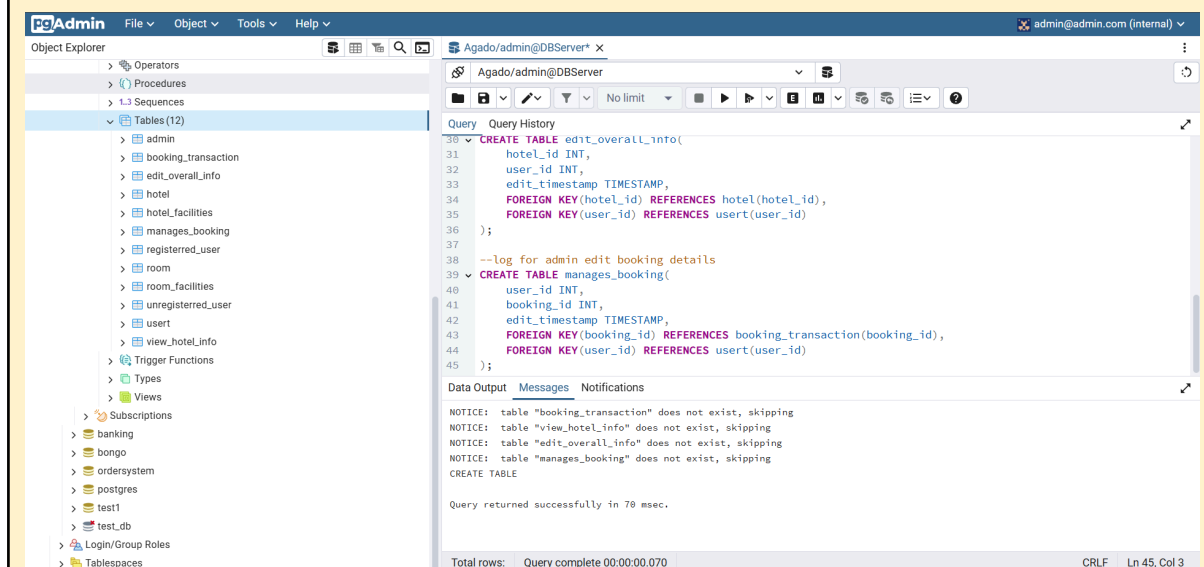
1.3) สร้าง table ที่เกี่ยวข้องการเก็บ logs จากการ view/insert/edit/delete ของ user

คำอธิบาย: สร้างตาราง booking_transaction (เก็บคำสั่งการจองทั้งหมด), 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:



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/uti_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

Query

Query History

162

select * from user;

Data Output

Messages

Notifications

SQL

Showing rows: 1 to 113

Page No: 1

of 1

	user_id [PK] integer	device_type character varying (32)
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

Total rows: 113

Query complete 00:00:00.068

CRLF

Ln 162, Col 1

2.1.2) ผลที่ปรากฏใน admin

Agado/admin@DBServer

Query Query History

162 `select * from admin;`

Data Output Messages Notifications

Showing rows: 1 to 9 Page No: 1 of 1

	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	900000001	113	Sittipong Wongsakul	password123	sittipong.wongsakul@example.com	1090123456
2	900000002	114	Pimchanok Jirawat	password456	pimchanok.jirawat@example.com	1101234567
3	900000003	115	Chanin Sutharak	password789	chanin.sutharak@example.com	1112345678
4	900000004	116	Thanakorn Rattanapong	password123	thanakorn.rattanapong@example.com	1189012345
5	900000005	117	Wassana Pongsuk	password456	wassana.pongsuk@example.com	1190123456
6	900000006	118	Supaporn Raksri	password789	supaporn.raksri@example.com	1201234567
7	900000007	119	Jirawat Charoenwong	password123	jirawat.charoenwong@example.com	1278901234
8	900000008	120	Piyarat Sittipong	password456	piyarat.sittipong@example.com	1289012345
9	900000009	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 Query History

162 `select * from unregistered_user;`

Data Output Messages Notifications

Showing rows: 1 to 41 Page No: 1 of 1

	user_id integer
1	100000001
2	100000002
3	100000003
4	100000004
5	100000005
6	100000006
7	100000007
8	100000008
9	100000009
10	100000010
11	100000011
12	100000012
13	100000013
14	100000014
15	100000015
16	100000016

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	
		Showing rows: 1 to 63					Page No: 1 of 1	
	user_id	user_name	user_password	user_email	telephone_number	register_date		
	integer	character varying (64)	character varying (64)	character varying (64)	character varying (10)	timestamp without time		
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/uti_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/hotel1.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 Query History

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

Data Output Messages Notifications

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

	hotel_id integer	user_id integer	edit_timestamp 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 Query History

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

Data Output Messages Notifications

Showing rows: 1 to 1000 Page No: 1 of 16

	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 00:00:00.081 CRLF Ln 1, Col 19

2.2.5) ผลที่ปรากฏใน room_facilities

Query

Query History

1

select * from room_facilities;

Data Output

Messages

Notifications

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

Query History

1

select * from booking_transaction;

Data Output

Messages

Notifications

SQL

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	301
18	100000018	2025-02-03	2	2025-02-10	2025-02-12	500000011	10007	302
19	100000019	2025-02-03	2	2025-02-10	2025-02-12	500000012	10008	303
20	100000020	2025-02-03	2	2025-02-10	2025-02-12	500000013	10009	201

Total rows: 20

Query complete 00:00:00.059

CRLF

Ln 1, Col 35

Query Query History

1 select * from room where status = false;

Data Output Messages Notifications

SQL

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	25000	30000	false	Expensive
18	10007	302	22000	27000	false	Expensive
19	10008	303	1200	2200	false	Cheap
20	10009	301	700	1400	false	Cheap

Total rows: 20 Query complete 00:00:00.094

CRLF Ln 1, Col 40

3.2) ทดสอบการแก้ไขการจอง [ยังไม่ได้ทดสอบ]

คำอธิบาย: เป็นคำสั่งแก้ไข ซึ่งเป็นการที่ regUser/admin จะ Update ข้อมูลลงไป 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)

คำแนะนำ:

ผลการรัน:
3.3) ทดสอบการแก้ไขการข้อมูลโรงแรม [ยังไม่ได้ทดสอบ]
<p>คำอธิบาย: เป็นคำสั่งจอง ซึ่งการที่ regUser/admin จะ Update ข้อมูลลงไปใน booking_transaction</p> <p>ส่วนของ SQL command:</p> <p>คำแนะนำ:</p> <p>ผลการรัน:</p>
4. test command view
4.1) ทดสอบการดู [ยังไม่ได้ทดสอบ]
<p>คำอธิบาย:</p> <p>ส่วนของ SQL command:</p> <p>1.</p> <p>คำแนะนำ:</p> <p>ผลการรัน:</p> <p>4.1.1) view ด้วยสิทธิ์ admin (ดูได้ทั้งหมด)</p> <p>4.1.2) view ด้วยสิทธิ์ regUser/unRegUser (จำกัดให้ดูได้บางส่วน)</p>
5. test command delete
5.1) ทดสอบการลบคำสั่งจอง [ยังไม่ได้ทดสอบ]
<p>คำอธิบาย:</p> <p>ส่วนของ SQL command:</p> <p>คำแนะนำ:</p> <p>ผลการรัน:</p> <p>5.1.1) ผลก่อนลบ</p> <p>5.1.2) ผลหลังลบ</p>
5.2) ทดสอบการลบโรงแรม [ยังไม่ได้ทดสอบ]
<p>คำอธิบาย:</p> <p>ส่วนของ SQL command:</p>

คำแนะนำ:

ผลการรัน:

5.1.1) ผลก่อนลบ

5.1.2) ผลหลังลบ

6. test complex query

6.1) โรงแรมที่มียอดการจองสูงที่สุด

คำสั่ง SQL ที่ใช้:

```
SELECT
    h.hotel_id,
    h.hotel_location,
    COUNT(bt.booking_id) AS total_bookings,
    ROUND(AVG(CAST((bt.check_out_date - bt.check_in_date) AS FLOAT)))
AS avg_stay_duration,
    ROUND(SUM(r.max_price - r.min_price) / COUNT(bt.booking_id), 2)
AS avg_booking_amount,
    COUNT(DISTINCT vhi.user_id) AS unique_visitors,
    COUNT(eoi.edit_timestamp) AS admin_edits
FROM hotel h
LEFT JOIN booking_transaction bt ON h.hotel_id = bt.hotel_id
LEFT JOIN room r ON h.hotel_id = r.hotel_id
LEFT JOIN view_hotel_info vhi ON h.hotel_id = vhi.hotel_id
LEFT JOIN edit_overall_info eoi ON h.hotel_id = eoi.hotel_id
GROUP BY h.hotel_id, h.hotel_location
ORDER BY total_bookings DESC
LIMIT 10;
```

คำอธิบาย: เลือก 10 โรงแรมที่มียอดการจองสูงที่สุด

ส่วนของ SQL command: complex_query/top_perform.sql

คำแนะนำ:

ผลการรัน:

QueryQuery History

1 SELECT

2 h.hotel_id,

3 h.hotel_location,

4 COUNT(bt.booking_id) AS total_bookings,

5 ROUND(AVG(CAST((bt.check_out_date - bt.check_in_date) AS FLOAT))) AS avg_stay_duration,

6 ROUND(SUM(r.max_price - r.min_price) / COUNT(bt.booking_id), 2) AS avg_booking_amount,

7 COUNT(DISTINCT vhi.user_id) AS unique_visitors,

8 COUNT(eoi.edit_timestamp) AS admin_edits

Data OutputMessagesNotifications

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

	hotel_id [PK] integer	hotel_location character varying (256)	total_bookings bigint	avg_stay_duration double precision	avg_booking_amount numeric	unique_visitors bigint	admin_edits bigint
1	10005	Super Skyscraper Hotel	10000	2	1500.00	0	0
2	10007	Ultimate Tower Hotel	8100	2	5000.00	0	0
3	10006	Luxury Haven Resort	6400	2	5000.00	0	0
4	10008	Midnight Express Hotel	3600	2	1146.00	0	0
5	10009	Sunset Paradise Resort	720	2	958.00	0	0
6	10000	Grand Palace Hotel	400	2	812.00	0	0
7	10004	Downtown Plaza Hotel	396	2	672.00	0	0
8	10003	Mountain View Lodge	216	2	733.00	0	0
9	10002	Beachfront Resort	150	2	620.00	0	0
10	10001	Cozy Cabin Inn	100	2	100.00	0	0

Total rows: 10Query complete 00:00:00.662CRLF Ln 16, Col 10

7. Dump data