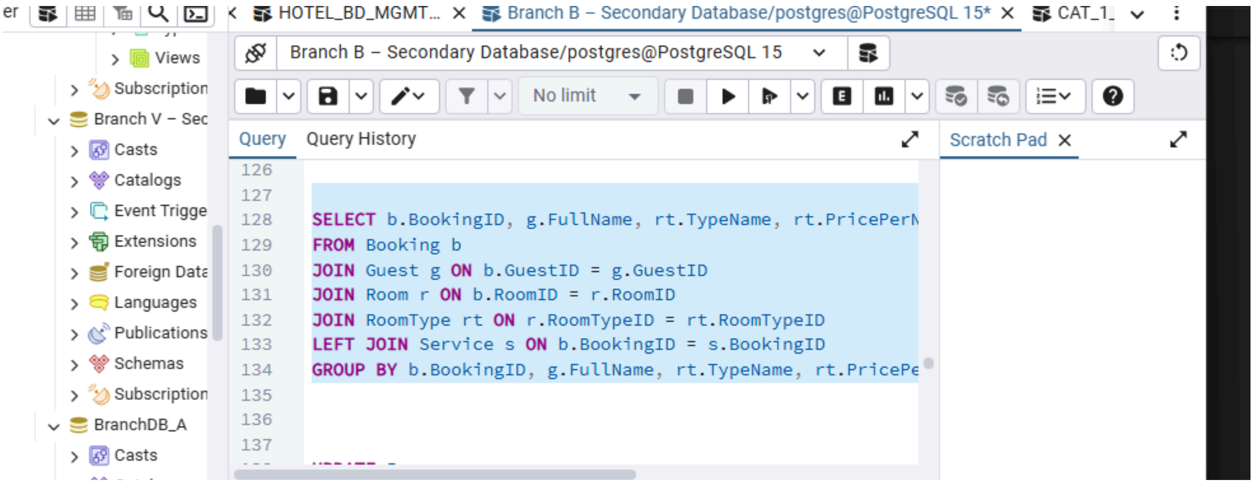
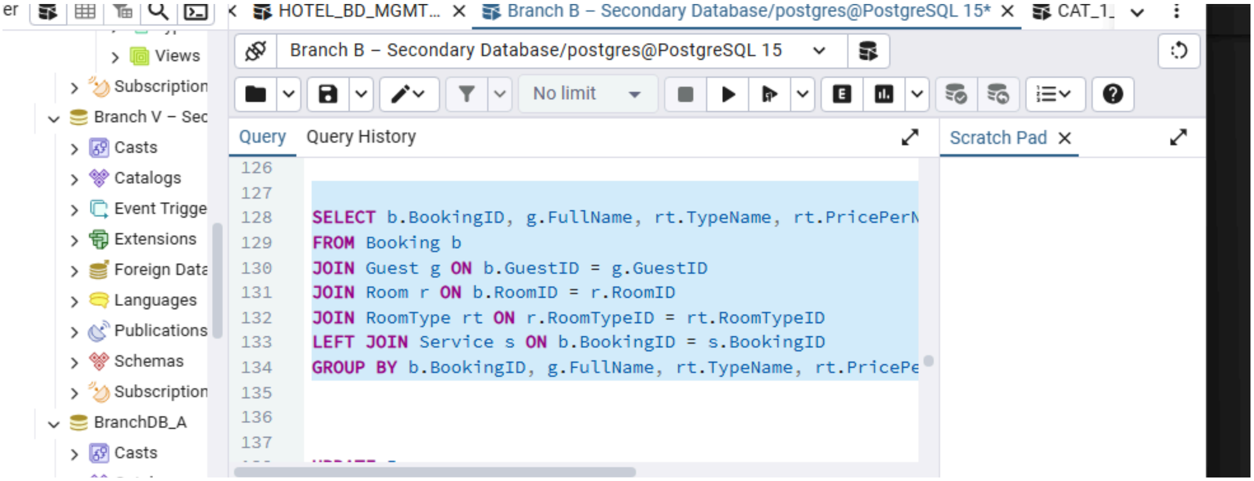
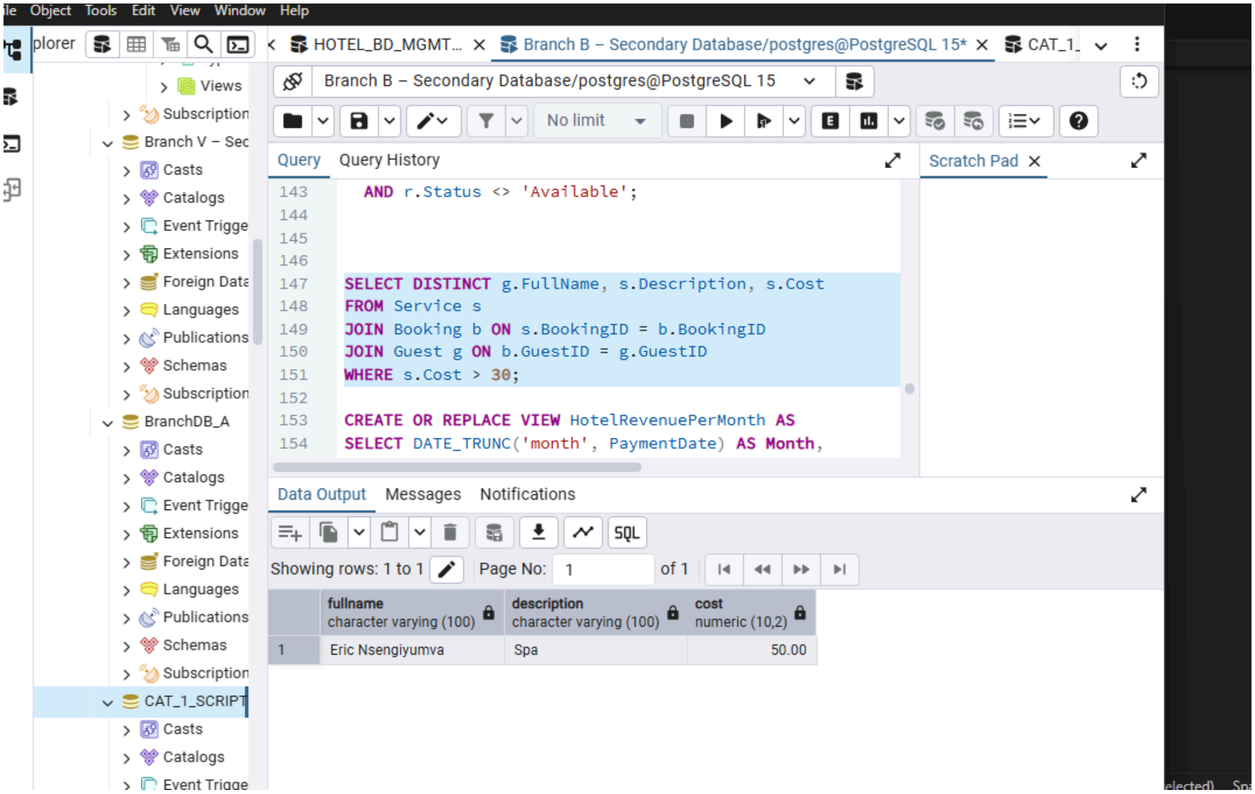


```sql
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```

```sql
SELECT DISTINCT g.FullName, s.Description, s.Cost
FROM Service s
JOIN Booking b ON s.BookingID = b.BookingID
JOIN Guest g ON b.GuestID = g.GuestID
WHERE s.Cost > 30;
```



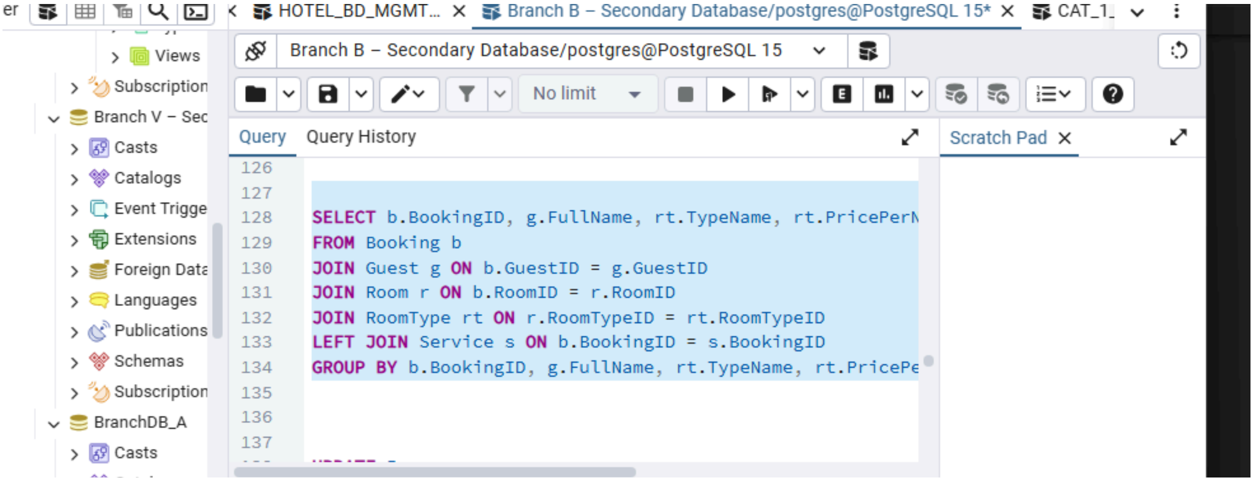
```sql
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```



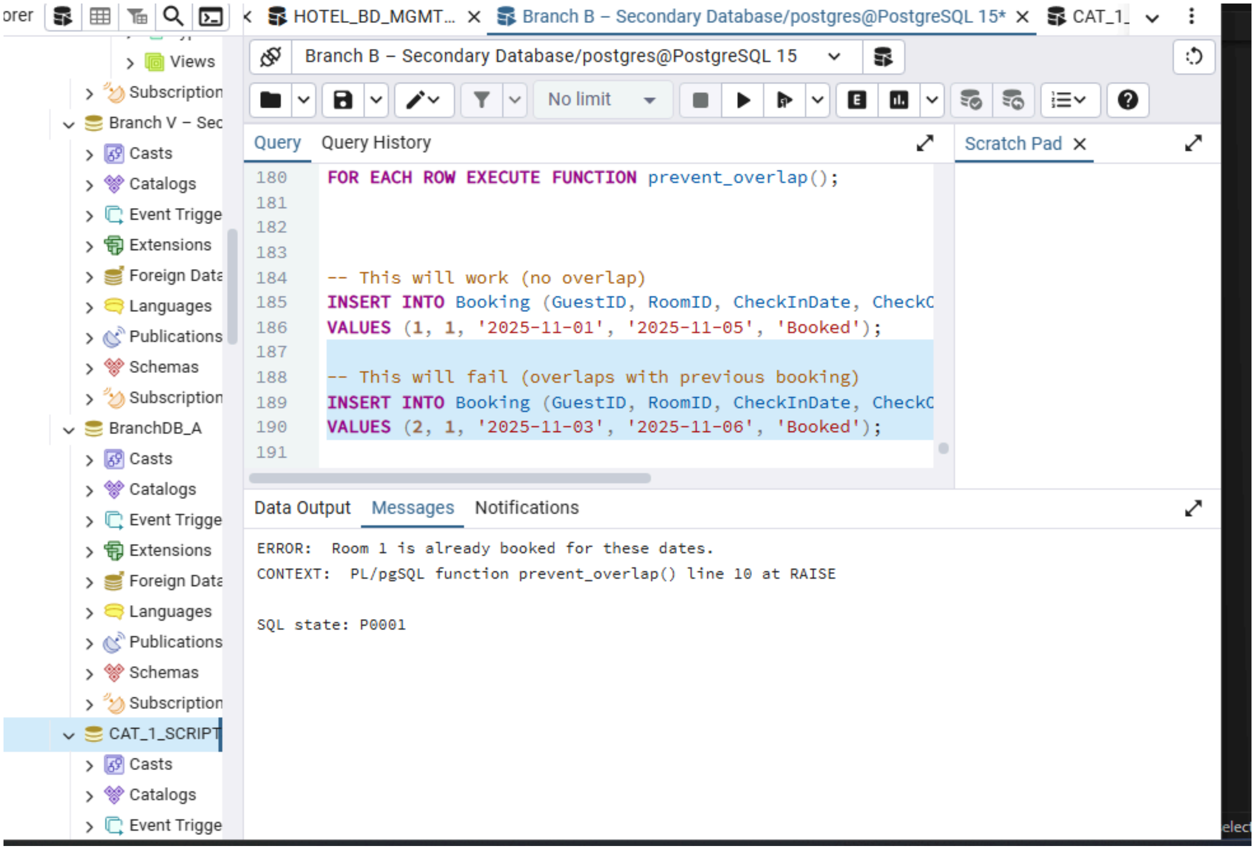
```sql
    AND r.Status <> 'Available';



SELECT DISTINCT g.FullName, s.Description, s.Cost
FROM Service s
JOIN Booking b ON s.BookingID = b.BookingID
JOIN Guest g ON b.GuestID = g.GuestID
WHERE s.Cost > 30;

CREATE OR REPLACE VIEW HotelRevenuePerMonth AS
SELECT DATE_TRUNC('month', PaymentDate) AS Month,
```

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

	fullname character varying (100)	description character varying (100)	cost numeric (10,2)
1	Eric Nsengiyumva	Spa	50.00



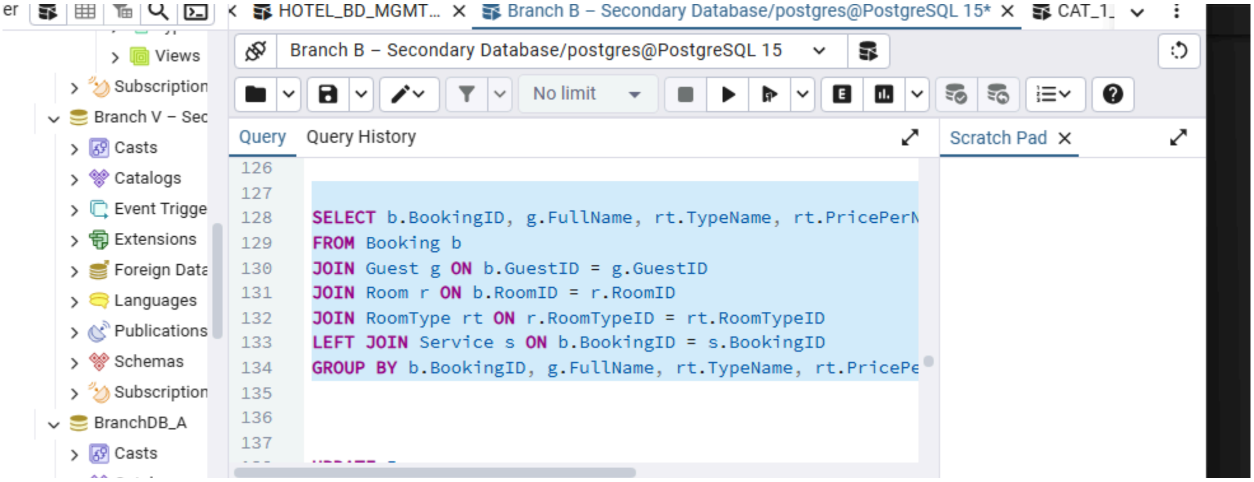
```sql
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```



```sql
FOR EACH ROW EXECUTE FUNCTION prevent_overlap();



-- This will work (no overlap)
INSERT INTO Booking (GuestID, RoomID, CheckInDate, CheckC
VALUES (1, 1, '2025-11-01', '2025-11-05', 'Booked');

-- This will fail (overlaps with previous booking)
INSERT INTO Booking (GuestID, RoomID, CheckInDate, CheckC
VALUES (2, 1, '2025-11-03', '2025-11-06', 'Booked');
```

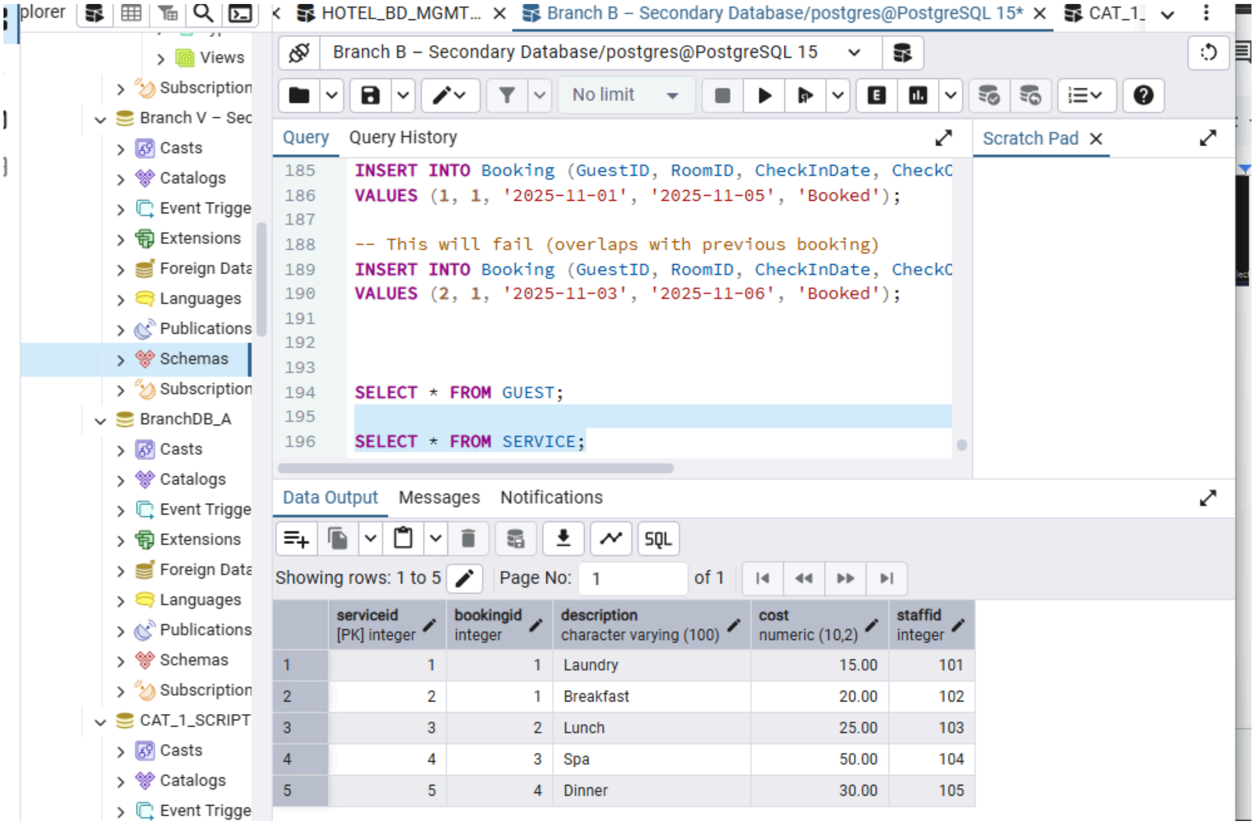
```
ERROR:  Room 1 is already booked for these dates.
CONTEXT:  PL/pgSQL function prevent_overlap() line 10 at RAISE

SQL state: P0001
```

SELECT * FROM GUEST;



```sql
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```

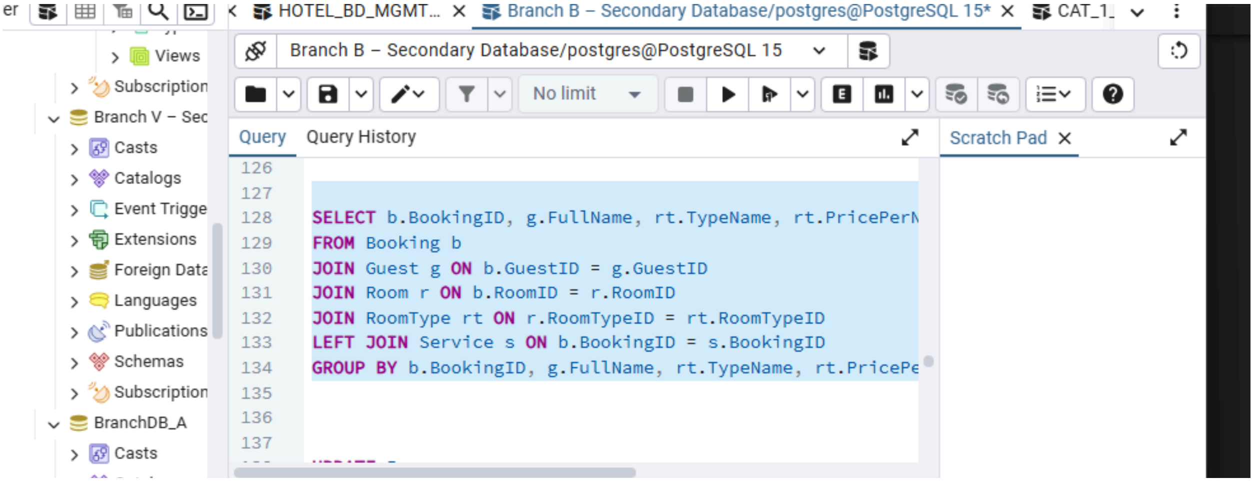
SELECT * FROM SERVICE;



```sql
INSERT INTO Booking (GuestID, RoomID, CheckInDate, CheckO
VALUES (1, 1, '2025-11-01', '2025-11-05', 'Booked');

-- This will fail (overlaps with previous booking)
INSERT INTO Booking (GuestID, RoomID, CheckInDate, CheckO
VALUES (2, 1, '2025-11-03', '2025-11-06', 'Booked');



SELECT * FROM GUEST;

SELECT * FROM SERVICE;
```

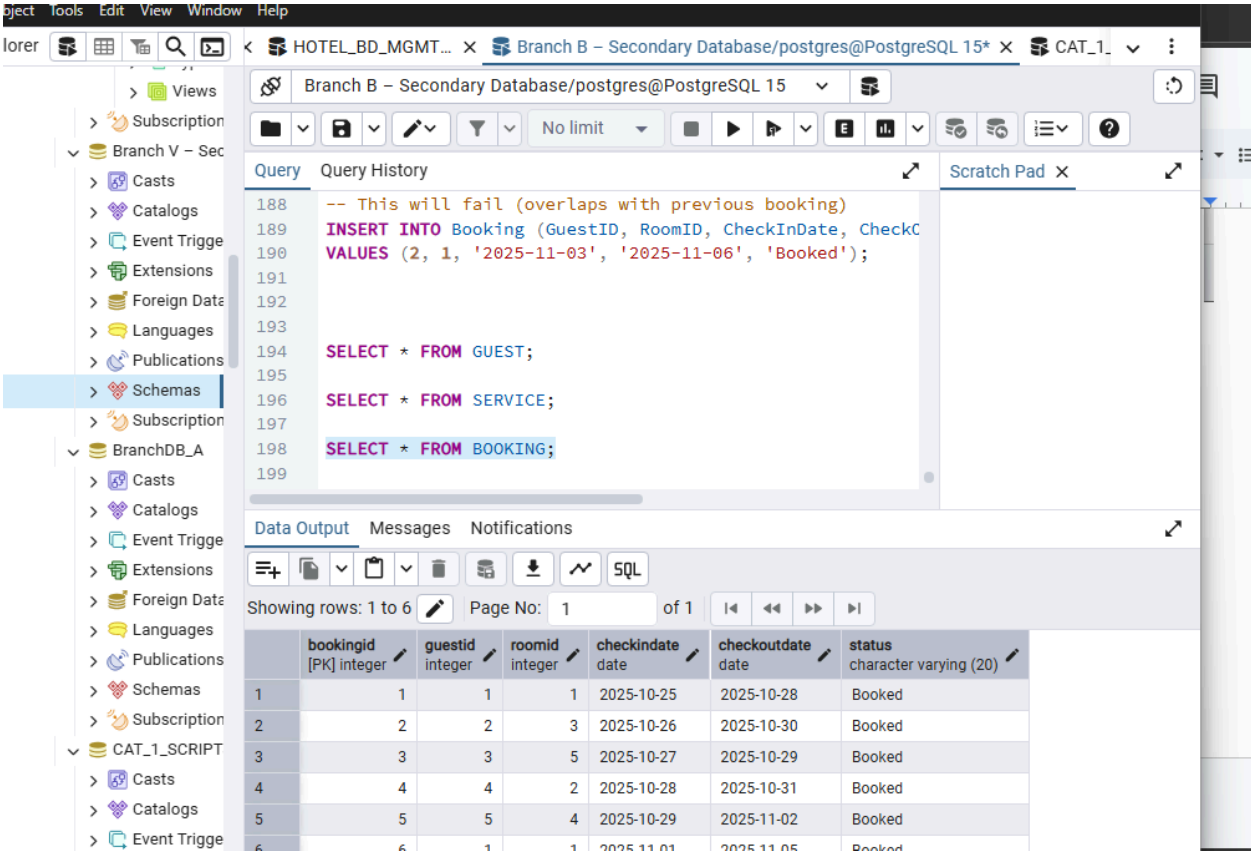
Showing rows: 1 to 5 Page No: 1 of 1

	serviceid [PK] integer	bookingid integer	description character varying (100)	cost numeric (10,2)	staffid integer
1	1	1	Laundry	15.00	101
2	2	1	Breakfast	20.00	102
3	3	2	Lunch	25.00	103
4	4	3	Spa	50.00	104
5	5	4	Dinner	30.00	105



```sql
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```

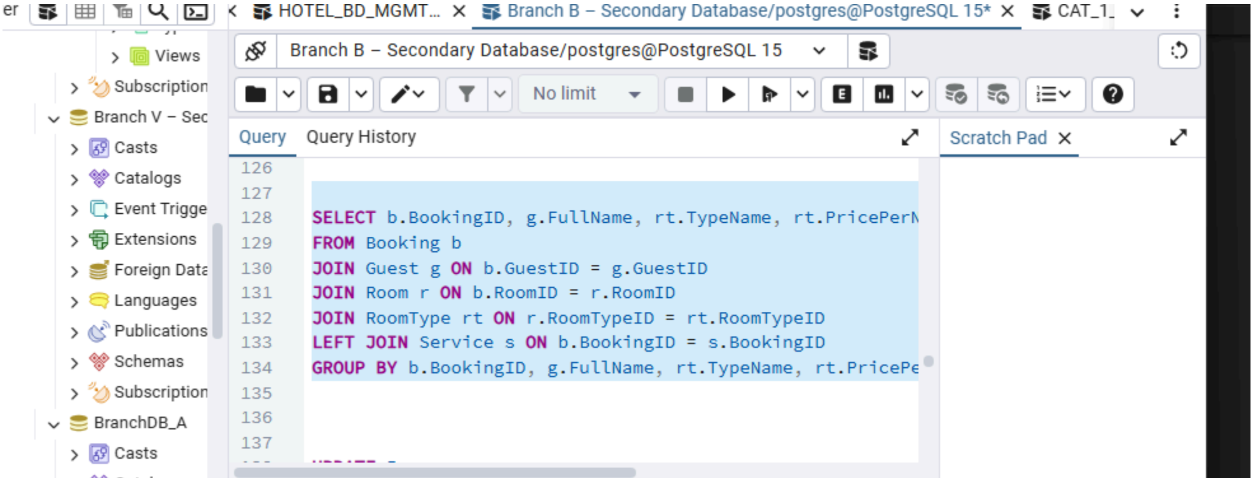
SELECT * FROM BOOKING;



```sql
-- This will fail (overlaps with previous booking)
INSERT INTO Booking (GuestID, RoomID, CheckInDate, CheckO
VALUES (2, 1, '2025-11-03', '2025-11-06', 'Booked');



SELECT * FROM GUEST;

SELECT * FROM SERVICE;

SELECT * FROM BOOKING;
```

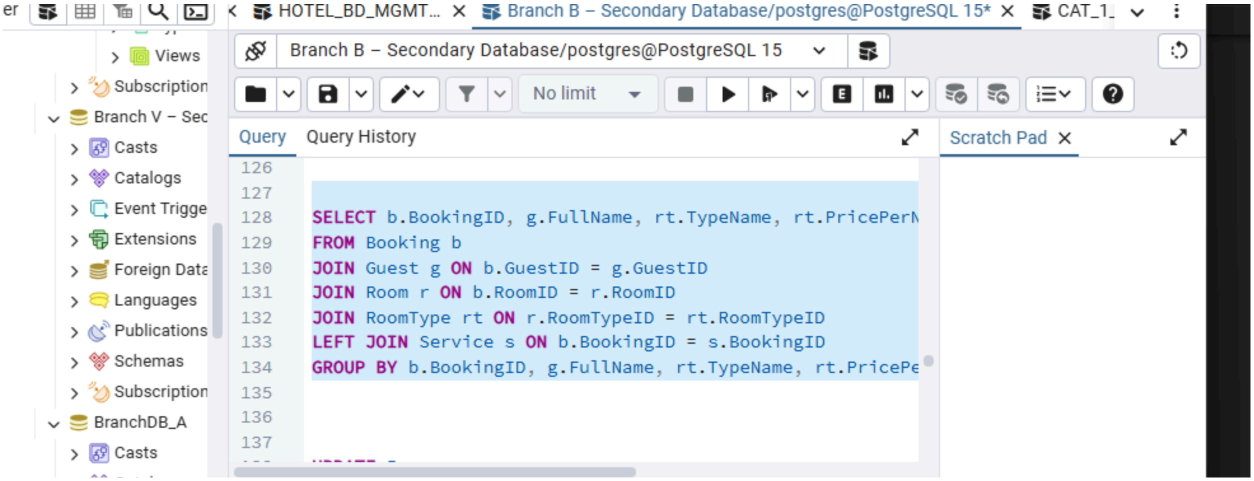
Showing rows: 1 to 6

	bookingid [PK] integer	guestid integer	roomid integer	checkindate date	checkoutdate date	status character varying (20)
1	1	1	1	2025-10-25	2025-10-28	Booked
2	2	2	3	2025-10-26	2025-10-30	Booked
3	3	3	5	2025-10-27	2025-10-29	Booked
4	4	4	2	2025-10-28	2025-10-31	Booked
5	5	5	4	2025-10-29	2025-11-02	Booked
6	6	1	1	2025-11-01	2025-11-05	Booked

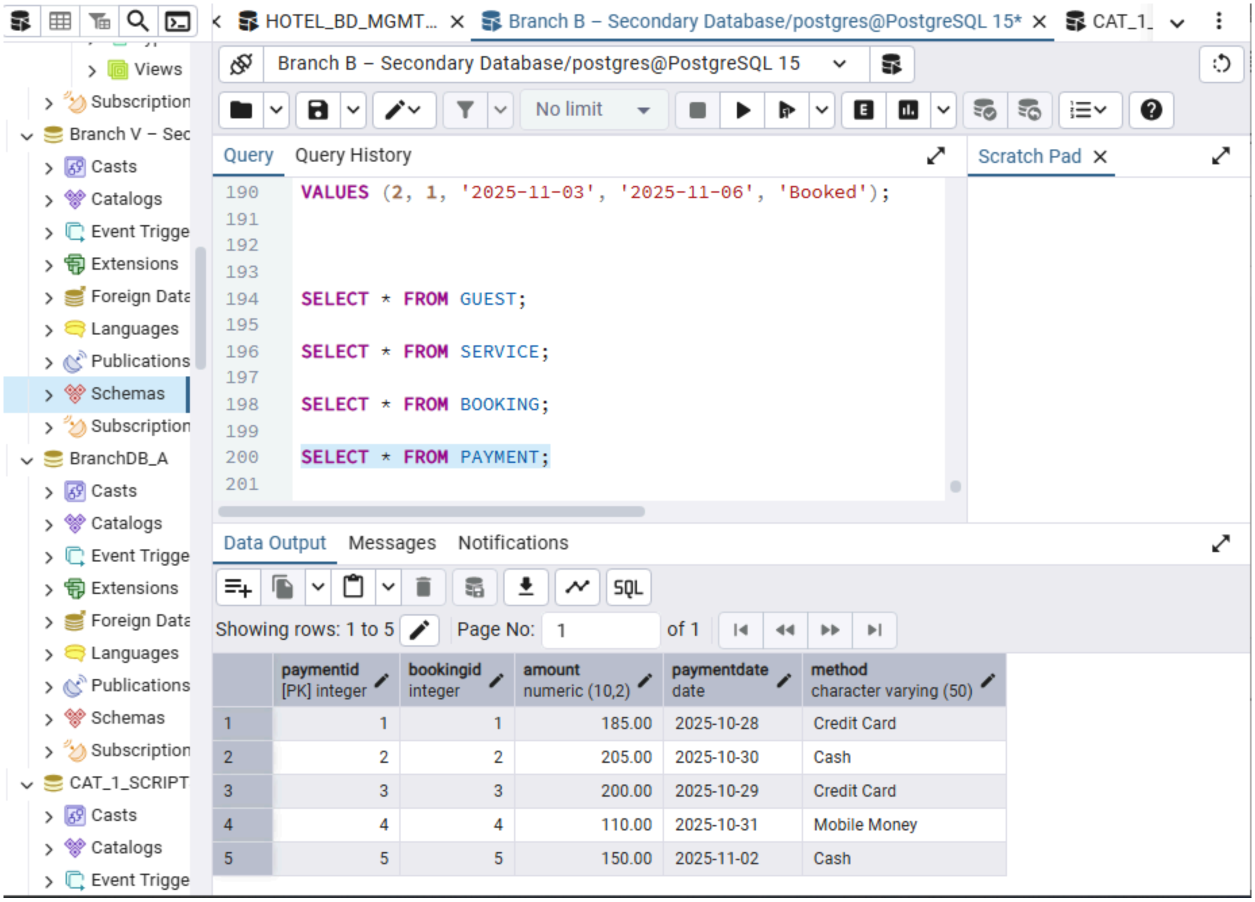


```
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```

SELECT * FROM PAYMENT;



```
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```

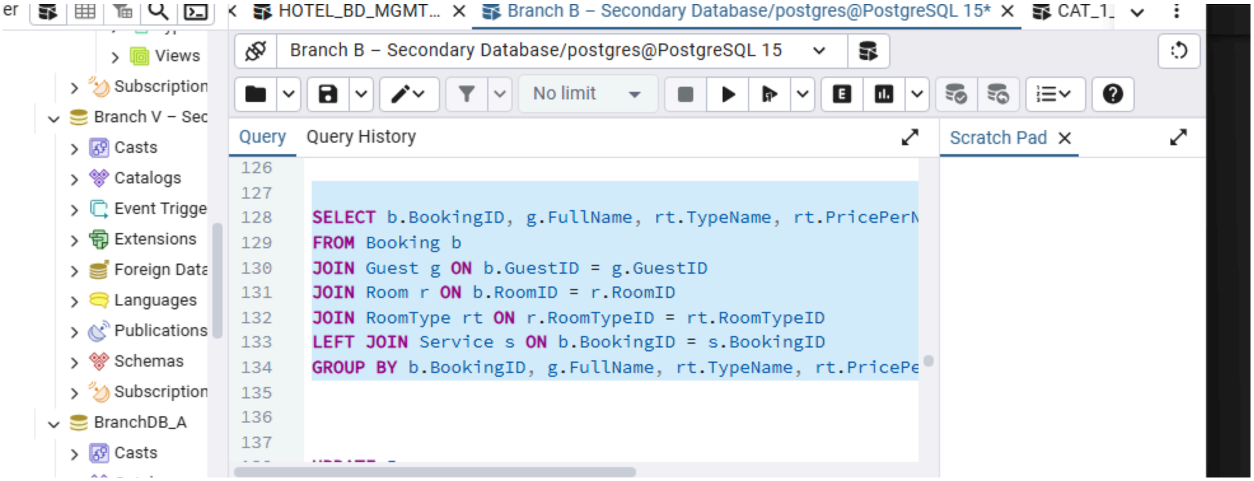


```
VALUES (2, 1, '2025-11-03', '2025-11-06', 'Booked');



SELECT * FROM GUEST;

SELECT * FROM SERVICE;

SELECT * FROM BOOKING;

SELECT * FROM PAYMENT;

```

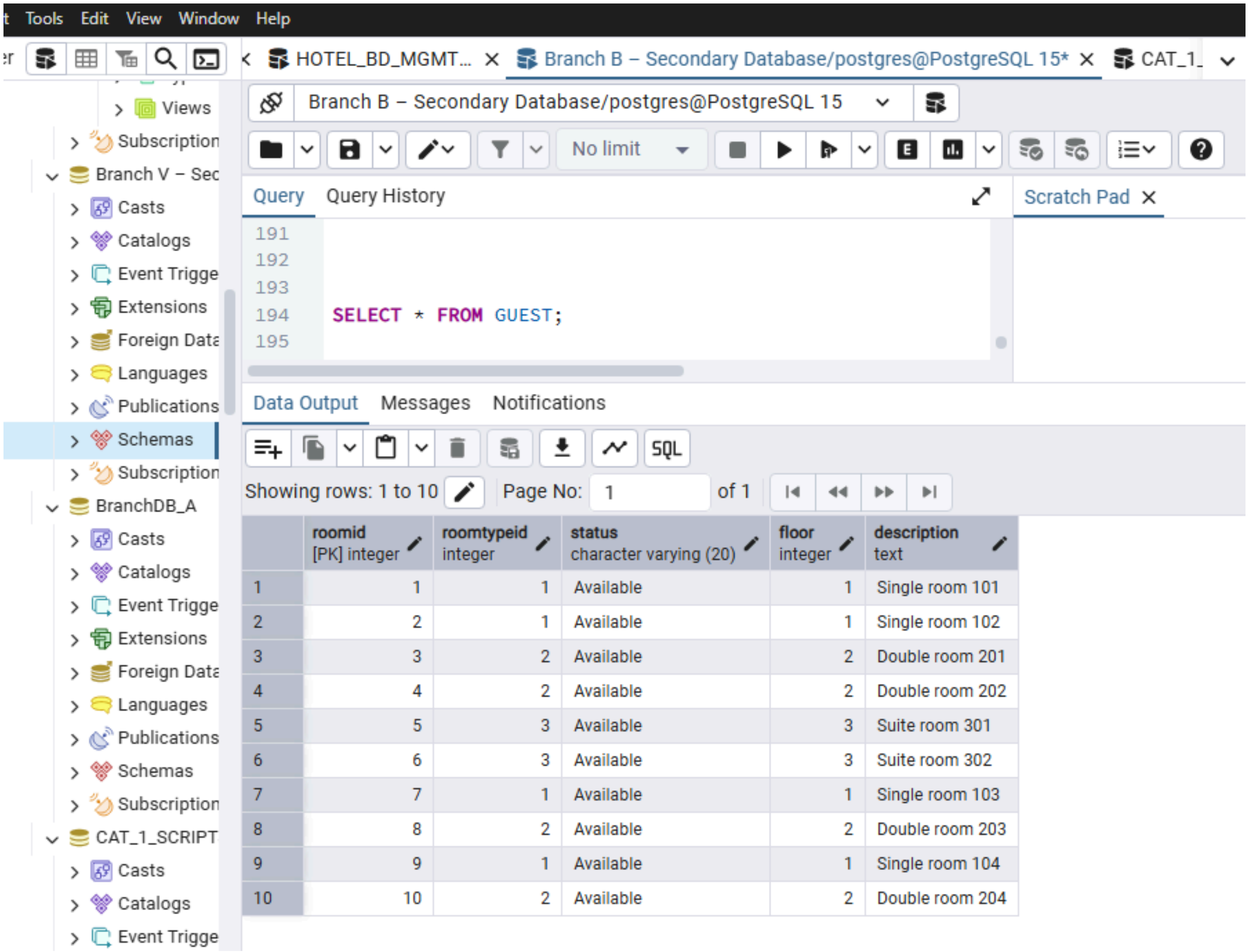
Showing rows: 1 to 5 Page No: 1 of 1

	paymentid [PK] integer	bookingid integer	amount numeric (10,2)	paymentdate date	method character varying (50)
1	1	1	185.00	2025-10-28	Credit Card
2	2	2	205.00	2025-10-30	Cash
3	3	3	200.00	2025-10-29	Credit Card
4	4	4	110.00	2025-10-31	Mobile Money
5	5	5	150.00	2025-11-02	Cash

SELECT * FROM ROOM;



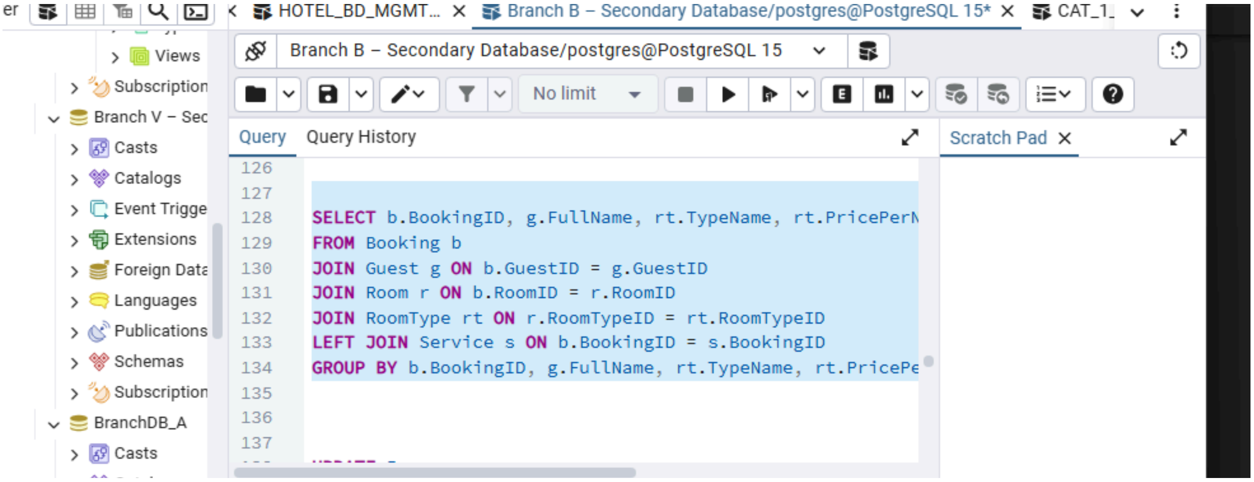
```
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```



```
SELECT * FROM GUEST;
```

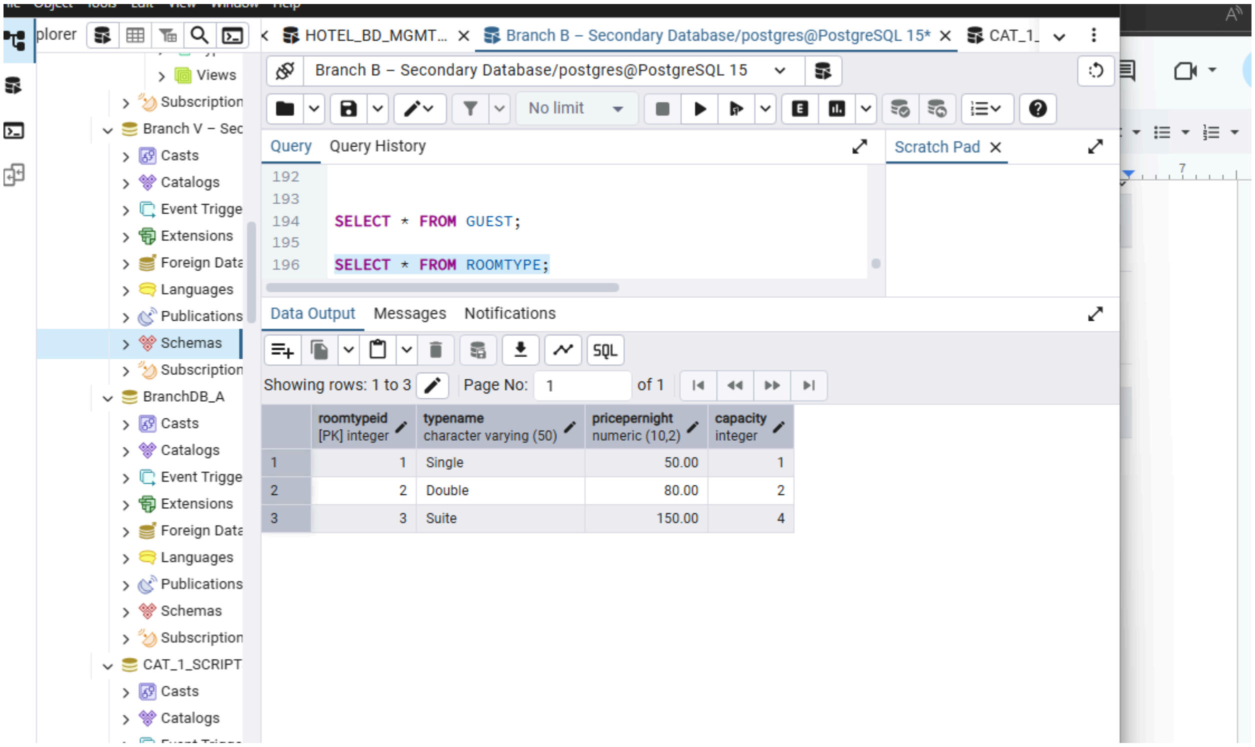
Showing rows: 1 to 10 Page No: 1 of 1

	roomid [PK] integer	roomtypeid integer	status character varying (20)	floor integer	description text
1	1	1	Available	1	Single room 101
2	2	1	Available	1	Single room 102
3	3	2	Available	2	Double room 201
4	4	2	Available	2	Double room 202
5	5	3	Available	3	Suite room 301
6	6	3	Available	3	Suite room 302
7	7	1	Available	1	Single room 103
8	8	2	Available	2	Double room 203
9	9	1	Available	1	Single room 104
10	10	2	Available	2	Double room 204



```sql
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```

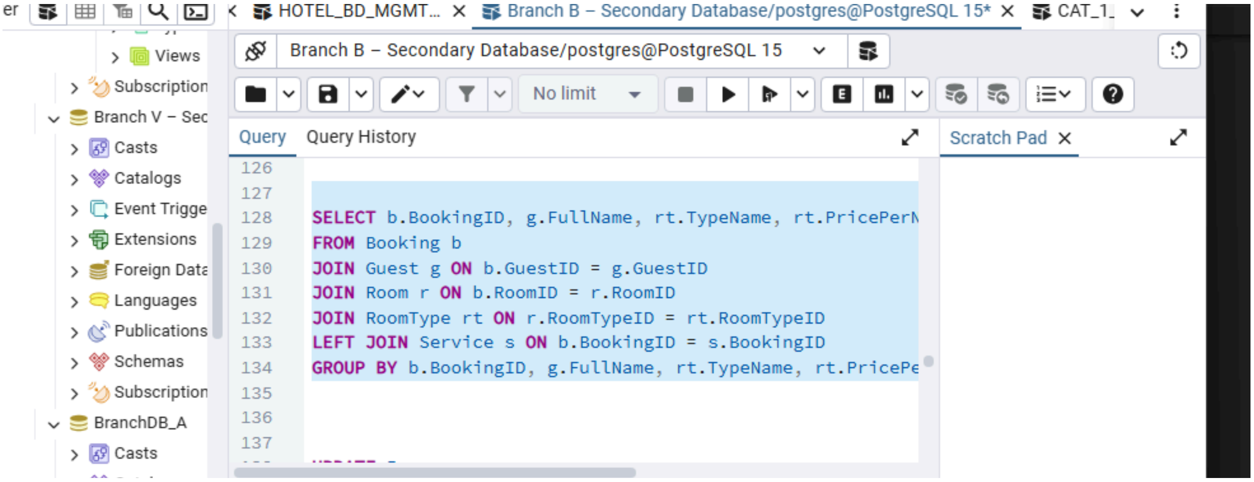
SELECT * FROM ROOMTYPE;



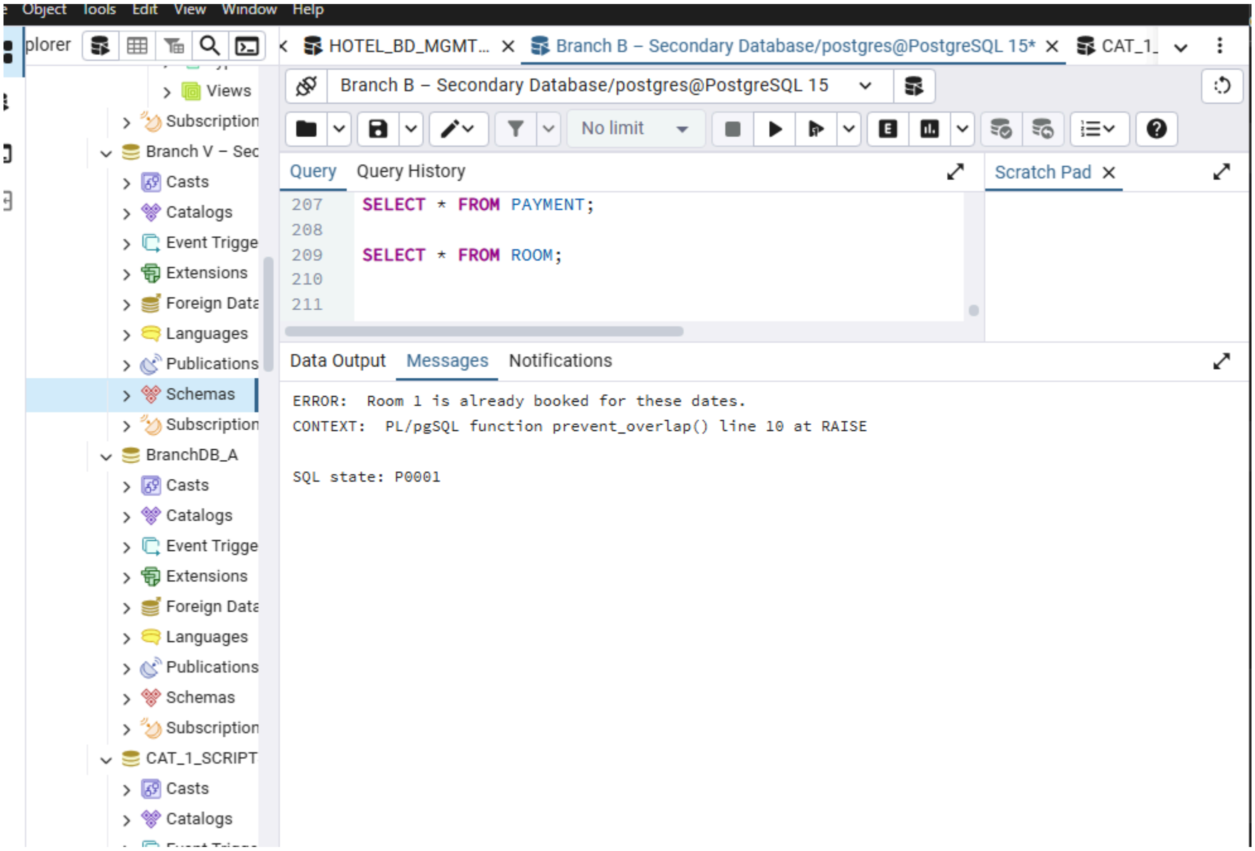
```sql
SELECT * FROM GUEST;

SELECT * FROM ROOMTYPE;
```

Showing rows: 1 to 3   Page No: 1 of 1

	roomtypeid [PK] integer	typename character varying (50)	pricepernight numeric (10,2)	capacity integer
1	1	Single	50.00	1
2	2	Double	80.00	2
3	3	Suite	150.00	4



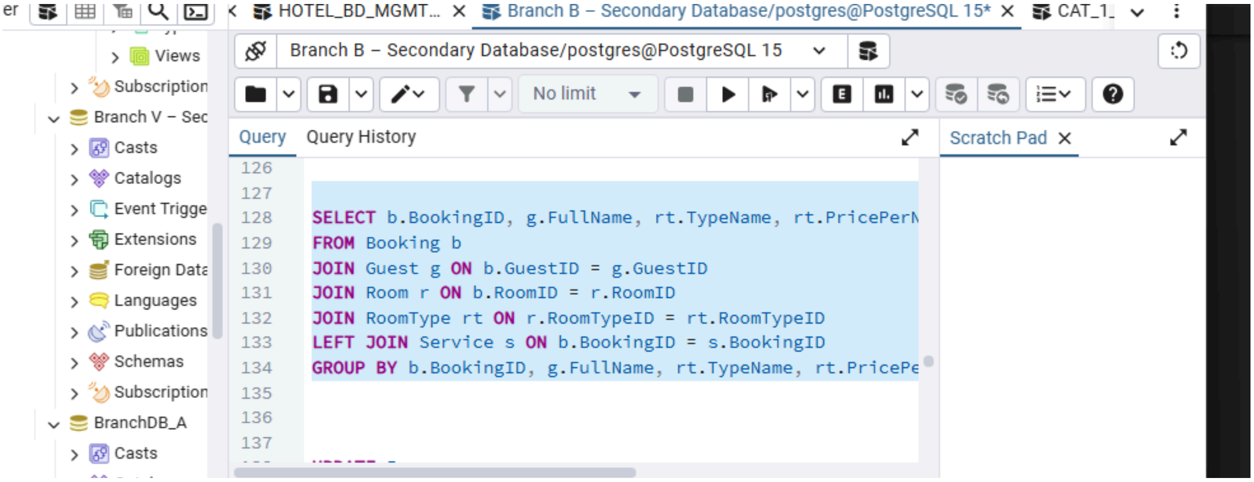
```sql
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```



```sql
SELECT * FROM PAYMENT;

SELECT * FROM ROOM;
```

```
ERROR:  Room 1 is already booked for these dates.
CONTEXT:  PL/pgSQL function prevent_overlap() line 10 at RAISE

SQL state: P0001
```

-- This will work (no overlap)

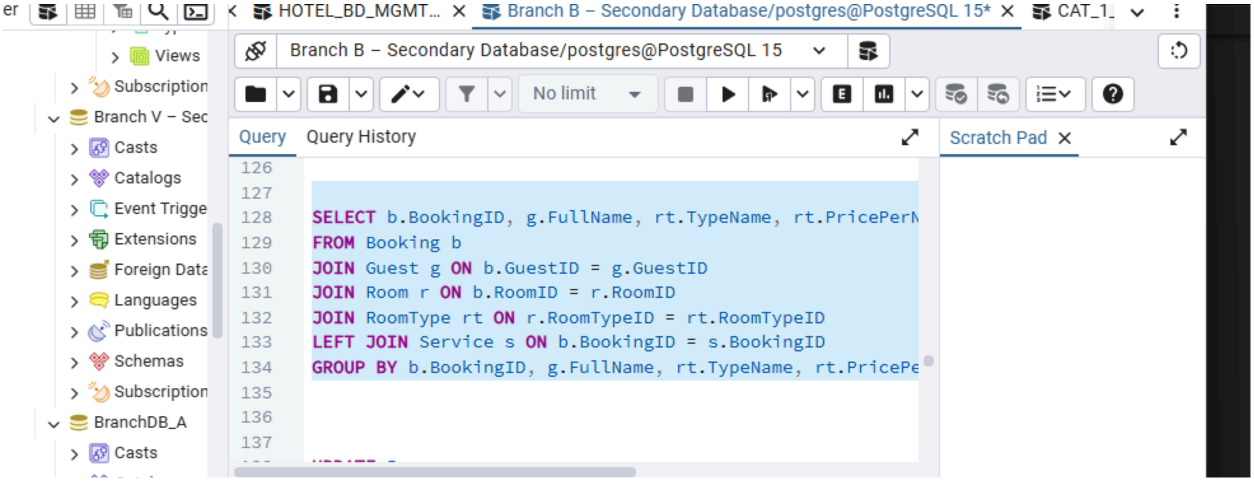


```
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```

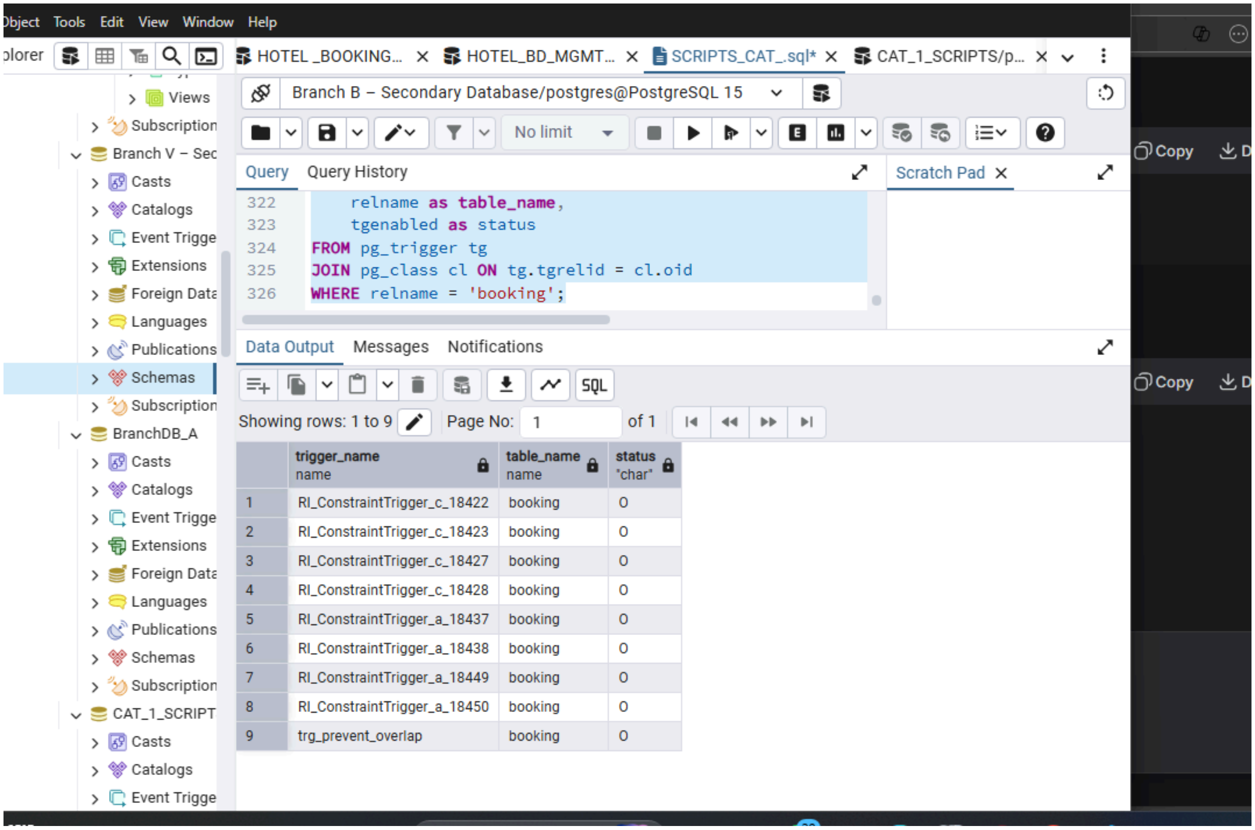
INSERT INTO Booking (GuestID, RoomID, CheckInDate, CheckOutDate, Status)
VALUES (1, 1, '2025-11-01', '2025-11-05', 'Booked');

-- This will fail (overlaps with previous booking)

INSERT INTO Booking (GuestID, RoomID, CheckInDate, CheckOutDate, Status)
VALUES (2, 1, '2025-11-03', '2025-11-06', 'Booked');



```sql
SELECT b.BookingID, g.FullName, rt.TypeName, rt.PricePerN
FROM Booking b
JOIN Guest g ON b.GuestID = g.GuestID
JOIN Room r ON b.RoomID = r.RoomID
JOIN RoomType rt ON r.RoomTypeID = rt.RoomTypeID
LEFT JOIN Service s ON b.BookingID = s.BookingID
GROUP BY b.BookingID, g.FullName, rt.TypeName, rt.PricePe
```



```sql
    relname as table_name,
    tgenabled as status
FROM pg_trigger tg
JOIN pg_class cl ON tg.tgrelid = cl.oid
WHERE relname = 'booking';
```

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

	trigger_name name	table_name name	status "char"
1	RI_ConstraintTrigger_c_18422	booking	O
2	RI_ConstraintTrigger_c_18423	booking	O
3	RI_ConstraintTrigger_c_18427	booking	O
4	RI_ConstraintTrigger_c_18428	booking	O
5	RI_ConstraintTrigger_a_18437	booking	O
6	RI_ConstraintTrigger_a_18438	booking	O
7	RI_ConstraintTrigger_a_18449	booking	O
8	RI_ConstraintTrigger_a_18450	booking	O
9	trg_prevent_overlap	booking	O