

Name: Min Thant Hein
ID: 6805020058

Step 1: Relations Creation

Very first step (Creating a new database):
My database name: 'hotel_management'

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer tree, which is expanded to show 'Servers (2)', 'PostgreSQL 13', 'PostgreSQL 17', and 'art_course'. Under 'art_course', there are several sub-folders like 'Casts', 'Catalogs', etc. Under 'Schemas (1)', there is a single entry 'public'. On the right, a 'Create - Database' dialog box is open. It has tabs for 'General', 'Definition', 'Security', 'Parameters', 'Advanced', and 'SQL'. The 'SQL' tab is selected, displaying the following SQL code:

```
CREATE DATABASE hotel_management
WITH
OWNER = postgres
ENCODING = 'UTF8'
CONNECTION LIMIT = -1
IS_TEMPLATE = False;
```

Below the SQL code, there are sections for 'Data Output' and 'Query return'. At the bottom of the dialog are buttons for 'Close', 'Reset', and 'Save'.

#

Creating a new 'Hotel' table:

The screenshot shows the pgAdmin 4 interface for PostgreSQL. On the left, the Object Browser tree shows the database structure under 'hotel_management'. A node for the 'hotel' table is selected, highlighted with a blue background. The main pane contains a SQL query window with the following code:

```
1 CREATE TABLE Hotel (
2     HotelID SERIAL PRIMARY KEY, -- Auto-incrementing primary key
3     HotelName VARCHAR(100) NOT NULL,
4     Address TEXT NOT NULL,
5     City VARCHAR(50) NOT NULL,
6     Country VARCHAR(50) NOT NULL,
7     Rating INTEGER CHECK (Rating >= 1 AND Rating <= 5),
8     Email VARCHAR(100) UNIQUE,
9     PhoneNumber VARCHAR(20),
10    CONSTRAINT chk_phone CHECK (PhoneNumber ~ '^[0-9\-\+\(\)\s]+$')
11 );
12
13
```

Below the query window, the 'Messages' tab is active, showing the message: "CREATE TABLE". Underneath it, it says "Query returned successfully in 160 msec."

#

Creating a new 'Room' table,

The screenshot shows the pgAdmin 4 interface for PostgreSQL 17. The left sidebar is the Object Explorer, showing the database structure under 'hotel_management'. A blue oval highlights the 'Tables (2)' section, which contains entries for 'hotel' and 'room'. The main pane is the Query Editor, displaying the SQL code for creating the 'Room' table:

```
CREATE TABLE Room (
    RoomID SERIAL PRIMARY KEY,
    HotelID INTEGER NOT NULL REFERENCES Hotel(HotelID) ON DELETE CASCADE,
    RoomNo INTEGER NOT NULL CHECK (RoomNo BETWEEN 1 AND 100),
    Type VARCHAR(10) NOT NULL CHECK (Type IN ('Single', 'Double', 'Family')),
    Price NUMERIC(6,2) NOT NULL CHECK (Price BETWEEN 10 AND 100),
    UNIQUE (HotelID, RoomNo) -- Ensures no duplicate room numbers in the same hotel
);
```

The Data Output tab shows the successful execution of the query, returning the message "Query returned successfully in 160 msec."

#

Creating a new ‘Guest’ table:

The screenshot shows the pgAdmin 4 interface for PostgreSQL. On the left, the Object Explorer tree view displays the database schema. A blue box highlights the 'Tables (3)' node under the 'hotel_management' schema. Inside this node, three tables are listed: 'guest', 'hotel', and 'room'. The main pane shows a SQL query window with the following code:

```
CREATE TABLE Guest (
    GuestID SERIAL PRIMARY KEY,
    FullName VARCHAR(100) NOT NULL,
    Email VARCHAR(100) UNIQUE,
    PhoneNumber VARCHAR(20),
    CONSTRAINT chk_guest_phone CHECK (PhoneNumber ~ '^[0-9\-\+\(\)\s]+$')
);
```

The 'Messages' tab in the bottom right shows the result of the query: "Query returned successfully in 46 msec."

#

Creating a new ‘Booking’ table:

The screenshot shows the pgAdmin 4 interface. On the left, the Object Browser tree view is open, showing the schema structure. Under the 'Schemas' node, the 'public' schema is expanded, and under 'Tables(4)', the 'booking' table is selected and highlighted with a blue background. The main pane displays a SQL query window with the following code:

```
1 CREATE TABLE Booking (
2     BookingID SERIAL PRIMARY KEY,
3     HotelID INTEGER NOT NULL REFERENCES Hotel(HotelID) ON DELETE CASCADE,
4     GuestID INTEGER NOT NULL REFERENCES Guest(GuestID) ON DELETE CASCADE,
5     RoomNo INTEGER NOT NULL CHECK (RoomNo BETWEEN 1 AND 100),
6     DateFrom DATE NOT NULL CHECK (DateFrom > CURRENT_DATE),
7     DateTo DATE NOT NULL CHECK (DateTo > CURRENT_DATE AND DateTo > DateFrom),
8
9     -- Prevent double-booking of the same room in the same hotel
10    CONSTRAINT no_double_booking UNIQUE (HotelID, RoomNo, DateFrom, DateTo),
11
12    -- Prevent overlapping bookings for the same guest
13    CONSTRAINT no_guest_overlap UNIQUE (GuestID, DateFrom, DateTo)
14 );
```

Below the query window, the 'Messages' tab is active, showing the message: "CREATE TABLE". At the bottom, it says "Query returned successfully in 64 msec."

#

##

Step 2: Insert the data using the following SQL (Show the tables in each data)

#Inserting the data into ‘Hotel’ table:

My Own Note: An error occurred before receiving the following tables. According to the input query lines from the professor, there are only 3 values for 3 columns. However, in the beginning, I created a ‘hotel’ table with 8 columns. Therefore, an error occurred. Here is the texts indicating error:

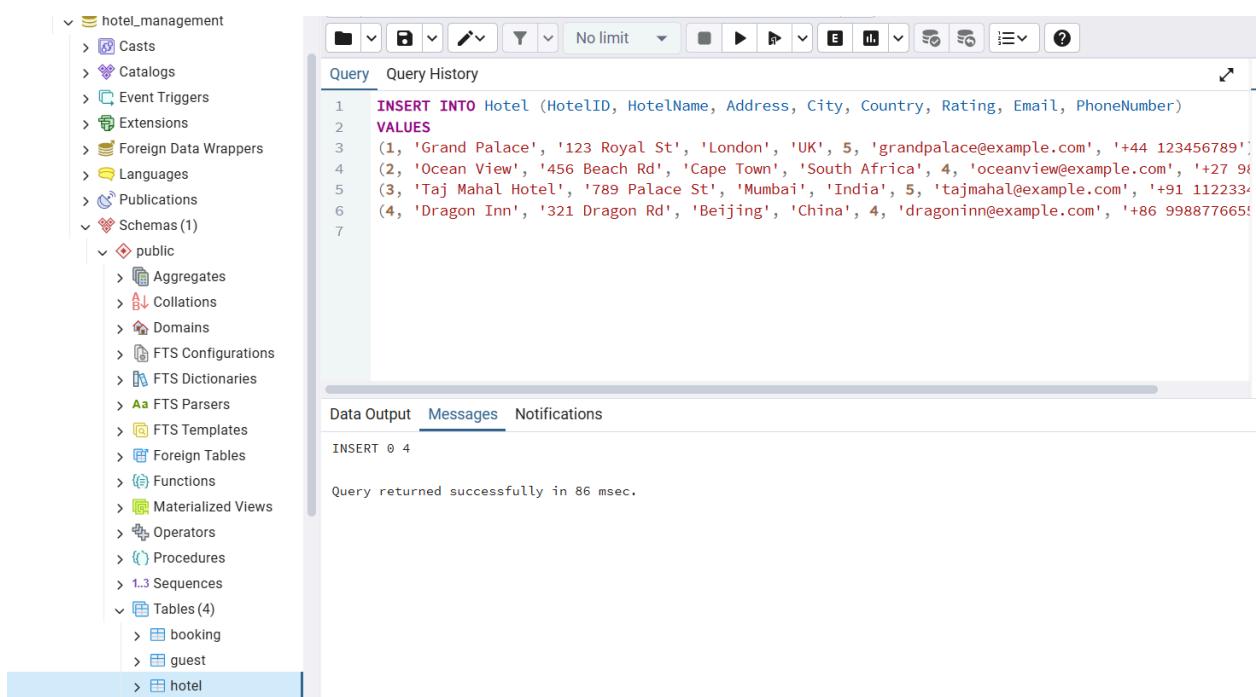
“"ERROR: null value in column "city" of relation "hotel" violates not-null constraint
Failing row contains (1, Grand Palace, London, null, null, null, null, null).

SQL state: 23502

Detail: Failing row contains (1, Grand Palace, London, null, null, null, null, null).””

Therefore, I had to create and add values for the rest of the columns.

Query, ‘Hotel’:



The screenshot shows a database interface with a sidebar containing a tree view of database objects under 'hotel_management'. The 'Tables' node is expanded, showing 'booking', 'guest', and 'hotel'. The 'hotel' table is selected and highlighted with a blue bar at the bottom. The main area is a query editor with the following content:

```
1 INSERT INTO Hotel (HotelID, HotelName, Address, City, Country, Rating, Email, PhoneNumber)
2 VALUES
3 (1, 'Grand Palace', '123 Royal St', 'London', 'UK', 5, 'grandpalace@example.com', '+44 123456789')
4 (2, 'Ocean View', '456 Beach Rd', 'Cape Town', 'South Africa', 4, 'oceanview@example.com', '+27 987654321')
5 (3, 'Taj Mahal Hotel', '789 Palace St', 'Mumbai', 'India', 5, 'tajmahal@example.com', '+91 1123345678')
6 (4, 'Dragon Inn', '321 Dragon Rd', 'Beijing', 'China', 4, 'dragoninn@example.com', '+86 9988776654')
```

The 'Messages' tab at the bottom shows the result of the execution:

INSERT 0 4
Query returned successfully in 86 msec.

Table view, 'Hotel':

The screenshot shows the pgAdmin interface with the 'Query' tab selected. The query window contains the following SQL code:

```
1 SELECT * FROM public.hotel
2 ORDER BY hotelid ASC
```

The results pane displays the data from the 'hotel' table:

	hotelid	hotellname	address	city	country	rating	email	phonenumber
1	1	Grand Palace	123 Royal St	London	UK	5	grandplace@example.com	+44 123456789
2	2	Ocean View	456 Beach Rd	Cape Town	South Africa	4	oceanview@example.com	+27 987654321
3	3	Taj Mahal Hotel	789 Palace St	Mumbai	India	5	tajmahal@example.com	+91 1122334455
4	4	Dragon Inn	321 Dragon Rd	Beijing	China	4	dragoninn@example.com	+86 9988776655

#

#Inserting the data into 'Room' Table:

Another error was raised. It is because of mismatch between columns and values. For me, I created 5 columns in total. However, the input values given by the professor, is only for 4 columns. That's why there is a mismatch and having an error. Therefore, I need to include 'RoomNo' and had to match the column order.

Here is the error:

```
" ERROR: invalid input syntax for type integer: "Single"
LINE 1: INSERT INTO Room VALUES (1, 1, 'Single', 50);
```

^

SQL state: 22P02

Character: 32 "

Query, 'Room':

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer tree, which includes Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), public (with Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences), and Tables (4) containing booking, guest, hotel, and room. The 'room' table is currently selected. The main pane shows a query editor with the following SQL code:

```
1  INSERT INTO Room (RoomID, HotelID, RoomNo, Type, Price)
2  VALUES
3  (1, 1, 1, 'Single', 50),
4  (2, 1, 2, 'Double', 80),
5  (3, 2, 1, 'Family', 90),
6  (4, 3, 2, 'Single', 45),
7  (5, 4, 3, 'Double', 70);
```

Below the query editor, the Data Output tab shows the message "INSERT 0 5" and the message "Query returned successfully in 99 msec."

Table view, 'Room':

The screenshot shows the pgAdmin interface with the following details:

- Object Explorer:** On the left, it lists various database objects under the schema 'public'. The 'Tables' node is expanded, showing four tables: booking, guest, hotel, and room.
- Query Editor:** The main area displays a query window with the following SQL code:

```
1 SELECT * FROM public.room
2 ORDER BY roomid ASC
```
- Data Output:** Below the query window, the results are shown in a table format. The columns are: roomid, hotelid, roomno, type, and price. The data is as follows:

roomid	hotelid	roomno	type	price
1	1	1	1 Single	50.00
2	2	1	2 Double	80.00
3	3	2	1 Family	90.00
4	4	3	2 Single	45.00
5	5	4	3 Double	70.00

#

#Inserting the data into 'Guest' Table:

Query, 'Guest':

The screenshot shows the pgAdmin 4 interface connected to a PostgreSQL 17 database. The left sidebar displays the schema browser with the 'Tables' node expanded, showing four tables: booking, guest, hotel, and room. The 'room' table is highlighted with a blue selection bar at the bottom. The main pane contains a query history window with the following content:

```
1 INSERT INTO Guest VALUES (101, 'Alice Johnson', '123 Main St, London, UK');
2 INSERT INTO Guest VALUES (102, 'Ravi Kumar', 'Mumbai, India');
3 INSERT INTO Guest VALUES (103, 'Chen Wei', 'Beijing, China');
4 INSERT INTO Guest VALUES (104, 'Maria Silva', 'Rio de Janeiro, Brazil');
5 INSERT INTO Guest VALUES (105, 'Andile Nkosi', 'Durban, South Africa');
```

Below the query history, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is selected, showing the message 'INSERT 0 1' and the note 'Query returned successfully in 83 msec.'

Table view, ‘Guest’:

The screenshot shows a database interface with the following details:

- Schemas:** hotel_management (selected), public
- Tables (4):** booking, guest, hotel, room
- Query Bar:** SELECT * FROM public.guest ORDER BY guestid ASC
- Data Output:**

	guestid [PK] integer	fullname character varying (100)	email character varying (100)	phonenumber character varying (20)
1	101	Alice Johnson	123 Main St, London, UK	[null]
2	102	Ravi Kumar	Mumbai, India	[null]
3	103	Chen Wei	Beijing, China	[null]
4	104	Maria Silva	Rio de Janeiro, Brazil	[null]
5	105	Andile Nkosi	Durban, South Africa	[null]

#

#Inserting the data into ‘Booking’ Table:

Another error occurred again. That is not having Hotel_ID in my query which created ‘Booking’ TABLE. Therefore, I did have to delete the whole ‘Room’ table and I created another new ‘Room’ Table which include Hotel_ID/ Hotel_No. Column.

Query, 'Booking':

The screenshot shows the pgAdmin 4 interface. On the left, a tree view displays database objects under the 'Schemas' node, specifically the 'public' schema which contains tables like 'booking', 'guest', 'hotel', and 'room'. The 'Tables' node is expanded. On the right, a query editor window titled 'hotel_management/postgres@PostgreSQL 17' shows the following SQL code:

```
1 INSERT INTO Booking (HotelID, GuestID, RoomNo, DateFrom, DateTo)
2 VALUES
3 (1, 101, 1, '2025-10-20', '2025-10-25'),
4 (3, 102, 2, '2025-11-01', '2025-11-05'),
5 (4, 103, 3, '2025-12-10', '2025-12-20');
```

The 'Messages' tab in the query editor shows the output:

```
INSERT 0 3
```

Below the messages, a note states: "Query returned successfully in 91 msec."

Table view, 'Booking':

The screenshot shows the pgAdmin 4 interface for a PostgreSQL database named 'public.booking/hotel_management/postgres@PostgreSQL 17'. The left sidebar displays the database schema, including Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), and Tables (4). The 'Tables (4)' section is currently selected, showing four tables: booking, guest, hotel, and room. The main pane contains a query editor with the following SQL code:

```
1 SELECT * FROM public.booking
2 ORDER BY bookingid ASC
```

Below the query editor is a 'Data Output' tab, which displays the results of the query as a table:

	bookingid [PK] integer	hotelid integer	guestid integer	roomno integer	datefrom date	datesto date
1	1	1	101	1	2025-10-20	2025-10-25
2	2	3	102	2	2025-11-01	2025-11-05
3	3	4	103	3	2025-12-10	2025-12-20

```
#  
##
```

Step 3: Create Views

Query, ‘Cheapest hotel view’:

The screenshot shows the pgAdmin interface with the following details:

- Object Explorer:** On the left, it shows the database structure under the 'public' schema, including tables like booking, guest, hotel, and room.
- Query Editor:** The main window displays a SQL query to create a view.

```
1 CREATE VIEW CheapestHotels AS
2 SELECT h.HotelID, h.HotelName, h.City, MIN(r.Price) AS CheapestPrice
3 FROM Hotel h
4 JOIN Room r ON h.HotelID = r.HotelID
5 GROUP BY h.HotelID, h.HotelName, h.City
6 HAVING MIN(r.Price) = (
7     SELECT MIN(Price) FROM Room
8 );
9
```
- Data Output:** Below the query editor, it shows the message "CREATE VIEW" and the success message "Query returned successfully in 99 msec."

Table view, ‘Cheapest hotel view’:

The screenshot shows the pgAdmin 4 interface. On the left is the Object Browser tree, which includes nodes for Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, Tables (4), Trigger Functions, Types, and Views (1). The 'cheapesthotels' view is selected under Views (1). The main pane contains a query editor with the following content:

```
1  SELECT * FROM public.cheapesthotels
2
```

Below the query editor is a Data Output tab showing the results of the query:

	hotelid integer	hotelname character varying (100)	city character varying (50)	cheapestprice numeric
1	3	Taj Mahal Hotel	Mumbai	45.00

##

Query, ‘Guests from BRICS countries view’:

The screenshot shows the pgAdmin 4 interface with the following details:

- Left Panel (Object Browser):** Shows the database structure:
 - Public Schema (1 item)
 - Tables (4 items): booking, guest, hotel, room
 - Views (2 items): bricsguests, cheapesthotels
- Center Panel (Query Editor):** A tabbed interface with "Query History" selected.

```
1 CREATE VIEW BRICSGuests AS
2 SELECT GuestID, FullName, Email, PhoneNumber
3 FROM Guest
4 WHERE Email ILIKE ANY (
5     ARRAY['%brazil%', '%russia%', '%india%', '%china%', '%southafrica%']
6 )
7 OR PhoneNumber ILIKE ANY (
8     ARRAY['%+55%', '+7%', '+91%', '+86%', '+27%']
9 );
10
```
- Bottom Panel (Messages):** Shows the message "CREATE VIEW".

Query returned successfully in 46 msec.

Table view, ‘Guests from BRICS countries view’:

The screenshot shows the pgAdmin 4 interface. On the left, a tree view displays the database schema:

- > Publications
- < Schemas (1)
 - public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > 1..3 Sequences
 - < Tables (4)
 - > booking
 - > guest
 - > hotel
 - > room
 - > Trigger Functions
 - > Types
 - < Views (2)
 - > bricsgguests
 - < cheapesthotels

##

Step 4: Create Two User Groups in Pgadmin

#Creating ‘manager’:

The screenshot shows the PgAdmin interface for managing PostgreSQL roles. On the left, a tree view lists database objects: Procedures, Sequences, Tables (4), Trigger Functions, Types, Views (2), Subscriptions, parami_data, and postgres. Below these are Login/Group Roles (11), with 'manager' selected. The main panel displays the 'Login Role - manager' configuration window. The 'Privileges' tab is active, showing the following settings:

Privilege	Status
Can login?	On
Superuser?	Off
Create roles?	Off
Create databases?	On
Inherit rights from the parent roles?	On
Can initiate streaming replication and backups?	Off
Bypass RLS?	Off

At the bottom right of the window are buttons for Close, Reset, and Save.

#

#Giving ‘manager’ view access to both ‘bricsgguest’ and ‘cheapesthotel’

The screenshot shows the pgAdmin 4 interface. On the left, a tree view displays the database schema:

- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (4): booking, guest, hotel, room
- Trigger Functions
- Types
- Views (2): bricsgguests, cheapesthotels
 - Columns
 - Rules
 - Triggers
- Subscriptions
- parami_data
- postgres
- Login/Group Roles (11): manager

The 'manager' role is currently selected, indicated by a blue background.

In the center, the query editor window has the following details:

- Connection: hotel_management/postgres@PostgreSQL 17
- Toolbar buttons: folder, dropdown, search, edit, filter, no limit, refresh, play, stop.
- Tab: Query (selected)
- Query text:

```
1 GRANT SELECT ON CheapestHotels TO manager;
2 GRANT SELECT ON BRICSGuests TO manager;
```
- Buttons below the query: Data Output, Messages (selected), Notifications.
- Messages tab content: GRANT
- Notifications tab content: Query returned successfully in 47 msec.

#

#Creating ‘staff’:

The screenshot shows the pgAdmin interface for managing PostgreSQL roles. On the left, a tree view lists database objects: Procedures, Sequences, Tables (4), Trigger Functions, Types, Views (2), Subscriptions, parami_data, postres, Login/Group Roles (11), and two specific roles: manager and staff. The staff role is highlighted with a blue selection bar at the bottom.

The main window displays the configuration for the 'staff' role under the 'Login Role - staff' tab. The 'Privileges' tab is selected. The configuration includes:

- Can login? (Enabled)
- Superuser? (Disabled)
- Create roles? (Disabled)
- Create databases? (Enabled)
- Inherit rights from the parent roles? (Enabled)
- Can initiate streaming replication and backups? (Disabled)
- Bypass RLS? (Disabled)

Below the configuration, a 'Data Output' section shows a table with one column, 'guestid', containing integer values 10, 10, and 10 for rows 1, 2, and 3 respectively. There are also 'i' and '?' icons next to the table.

At the bottom right of the configuration window are buttons for Close, Reset, and Save.

#

#Giving ‘staff’ view access only to ‘Bricsgguess’:

The screenshot shows the pgAdmin 4 interface with a database schema tree on the left and a query editor on the right.

Database Schema Tree:

- > 1..3 Sequences
- < Tables (4)
 - > booking
 - > guest
 - > hotel
 - > room
- > Trigger Functions
- > Types
- < Views (2)
 - > bricsgguests
 - < cheapesthotels
 - > Columns
 - > Rules
 - > Triggers
- > Subscriptions
- > parami_data
- > postgres
- < Login/Group Roles (11)
 - < manager
 - < pg_execute_server_program
 - < pg_monitor
 - < pg_read_all_settings
 - < pg_read_all_stats
 - < pg_read_server_files
 - < pg_signal_backend
 - < pg_stat_scan_tables
 - < pg_write_server_files
 - < postgres
 - < staff

#View permission screenshots, for both, ‘bricsgguest’ and ‘cheapesthotel’ views:

#Both ‘manager’ and ‘staff’ having access to ‘bricsgguest’ view:

The screenshot shows the pgAdmin interface for managing PostgreSQL objects. On the left, the object browser tree shows the schema structure under 'hotel_management' including FTS configurations, Dictionaries, Parsers, Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, Tables (booking, guest, hotel, room), Trigger Functions, Types, and Views (bricsgguests, cheapesthotels). The 'bricsgguests' view is selected and highlighted with a blue background.

The main pane displays the 'bricsgguests' view configuration. The 'Security' tab is active, showing the 'Privileges' section. The table lists three grants:

Grantee	Privileges	Grantor
manager	r	postgres
postgres	Dadrtwx	postgres
staff	r	postgres

Below the privileges, there is a 'Security labels' section with a single row:

Provider	Security label

At the bottom of the pane, the status message 'Query returned successfully' is displayed. The bottom right corner of the main window contains buttons for 'Close', 'Reset', and 'Save'.

Only 'manager' having access to 'cheapesthotels' view:

The screenshot shows the pgAdmin interface for managing a database object named 'cheapesthotels'. The left sidebar lists various database objects like FTS Dictionaries, Foreign Tables, Functions, etc. The 'Views (2)' section is expanded, showing 'bricsgquests' and 'cheapesthotels'. The 'cheapesthotels' view is selected and highlighted with a blue background.

The main window has tabs for General, Definition, Code, Security, and SQL. The Security tab is active, showing the 'Privileges' section. It contains two entries:

Grantee	Privileges	Grantor
manager	r	postgres
postgres	Dadrtwx	postgres

Below the privileges is a 'Security labels' section with a single entry:

Provider	Security label

At the bottom of the main window, there are tabs for Data Output, Messages, and Notes. The 'Messages' tab is active, showing the message: 'GRANT Query returned successfully'. There are also buttons for Close, Reset, and Save at the bottom right.

##

Step 5: Test in PSql

#Opening Psql and connecting as ‘manager’

```
Server [localhost]:  
Database [postgres]:  
Port [5432]:  
Username [postgres]:  
Password for user postgres:  
  
psql (13.22)  
WARNING: Console code page (437) differs from Windows code page (1252)  
8-bit characters might not work correctly. See psql reference  
page "Notes for Windows users" for details.  
Type "help" for help.  
  
postgress=# \c hotel_management manager  
Password for user manager:  
  
You are now connected to database "hotel_management" as user "manager".  
hotel_management=> |
```

#Seeing CheapestHotels View as 'manager' and seeing BRICSGuests View as 'manager'

```
Username [postgres]:  
Password for user postgres:  
  
psql (13.22)  
WARNING: Console code page (437) differs from Windows code page (1252)  
8-bit characters might not work correctly. See psql reference  
page "Notes for Windows users" for details.  
Type "help" for help.  
  
postgres=# \c hotel_management manager  
Password for user manager:  
  
You are now connected to database "hotel_management" as user "manager".  
hotel_management=> SELECT * FROM CheapestHotels;  
 hotelid |   hotelname   |   city    |   cheapestprice  
-----+-----+-----+-----  
      3 | Taj Mahal Hotel | Mumbai |        45.00  
(1 row)  
  
hotel_management=> SELECT * FROM BRICSGuests;  
 guestid |   fullname   |           email          |   phononenumber  
-----+-----+-----+-----  
    102 | Ravi Kumar | Mumbai, India |  
    103 | Chen Wei   | Beijing, China |  
    104 | Maria Silva | Rio de Janeiro, Brazil |  
(3 rows)  
  
hotel_management=> |
```

#Switching to 'staff'

```
WARNING: Console code page (437) differs from Windows code page (1252)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \c hotel_management manager
Password for user manager:

You are now connected to database "hotel_management" as user "manager".
hotel_management=> SELECT * FROM CheapestHotels;
 hotelid |   hotelname    |   city    |   cheapestprice
-----+-----+-----+-----+
      3 | Taj Mahal Hotel | Mumbai |        45.00
(1 row)

hotel_management=> SELECT * FROM BRICSGuests;
 guestid |   fullname    |   email    |   phononenumber
-----+-----+-----+-----+
    102 | Ravi Kumar | Mumbai, India | 
    103 | Chen Wei   | Beijing, China | 
    104 | Maria Silva | Rio de Janeiro, Brazil | 
(3 rows)

hotel_management=> \c hotel_management staff
Password for user staff:

You are now connected to database "hotel_management" as user "staff".
hotel_management=> |
```

#Seeing CheapestHotels View as 'staff' and seeing BRICSGuests View as 'staff'

```
hotel_management=> \c hotel_management staff
Password for user staff:

You are now connected to database "hotel_management" as user "staff".
hotel_management=> SELECT * FROM CheapestHotels;
ERROR: permission denied for view cheapesthotels
hotel_management=> SELECT * FROM BRICSGuests;
 guestid | fullname      |           email           | phononenumber
-----+-----+-----+-----+
    102 | Ravi Kumar   | Mumbai, India          |
    103 | Chen Wei     | Beijing, China         |
    104 | Maria Silva  | Rio de Janeiro, Brazil |
(3 rows)

hotel_management=> |
```

##