

GROUP PROJECT OUTPUTS

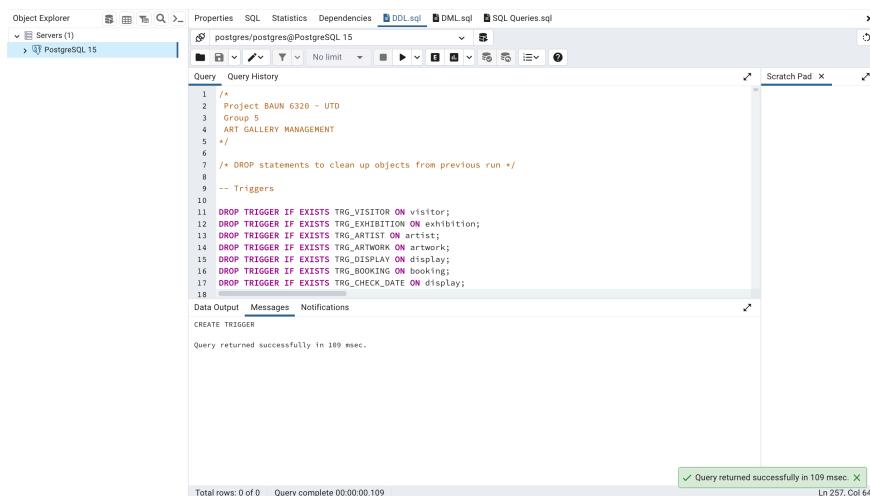
Course: BUAN 6320

Submitted By:

GROUP 5

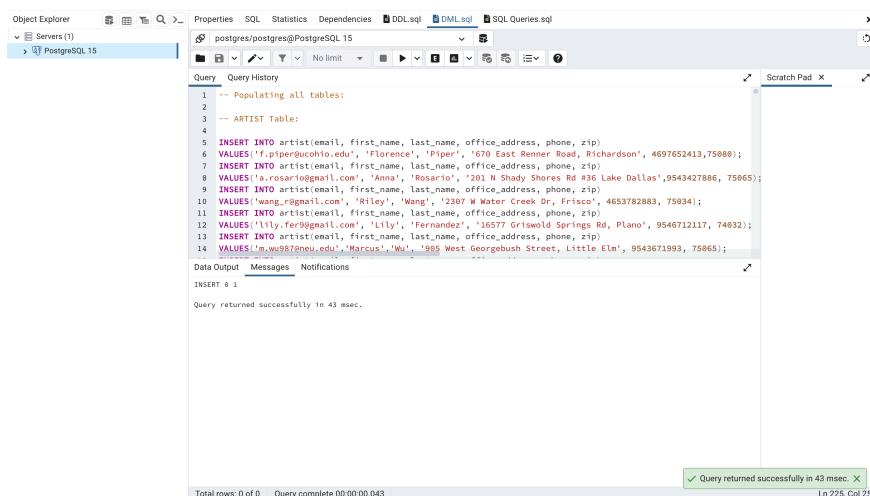
nxs230038, sxf230007, sxa230000, nxk230008

DDL:



```
Object Explorer Properties SQL Statistics Dependencies DDL.sql DML.sql SQL Queries.sql
Servers (1) postgres/postgres@PostgreSQL 15
Query History
1 /*+
2 Project BAUN 6320 - UTD
3 Group 5
4 ART GALLERY MANAGEMENT
5 */
6
7 /* DROP statements to clean up objects from previous run */
8
9 -- Triggers
10 DROP TRIGGER IF EXISTS TRG_VISITOR ON visitor;
11 DROP TRIGGER IF EXISTS TRG_EXHIBITION ON exhibition;
12 DROP TRIGGER IF EXISTS TRG_ARTIST ON artist;
13 DROP TRIGGER IF EXISTS TRG_ARTWORK ON artwork;
14 DROP TRIGGER IF EXISTS TRG_DISPLAY ON display;
15 DROP TRIGGER IF EXISTS TRG_BOOKING ON booking;
16 DROP TRIGGER IF EXISTS TRG_CHECK_DATE ON display;
17 DROP TRIGGER IF EXISTS TRG_CHECK_DATE ON display;
18
CREATE TRIGGER
Query returned successfully in 109 msec.
Ln 257, Col 64
Data Output Messages Notifications
Total rows: 0 of 0 Query complete 00:00:00.109
Query returned successfully in 109 msec. X
Ln 257, Col 64
```

DML:

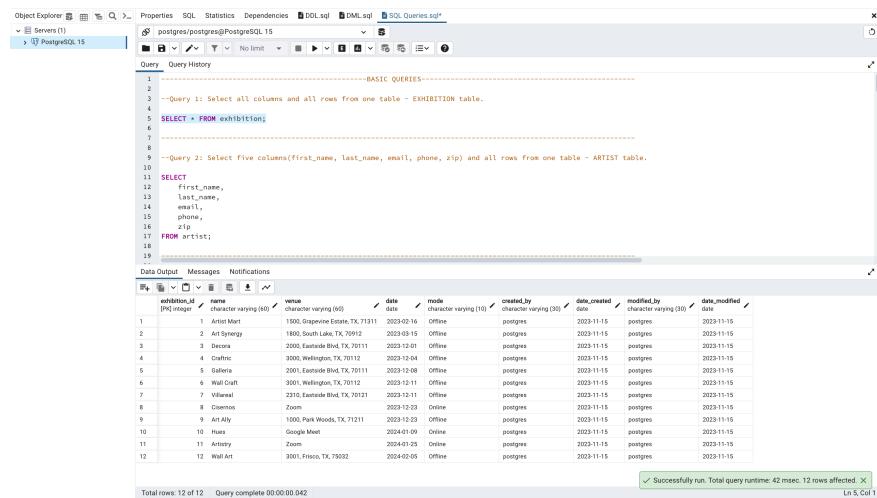


```
Object Explorer Properties SQL Statistics Dependencies DDL.sql DML.sql SQL Queries.sql
Servers (1) postgres/postgres@PostgreSQL 15
Query History
1 -- Populating all tables:
2
3 -- ARTIST Table:
4
5 INSERT INTO artist(email, first_name, last_name, office_address, phone, zip)
6 VALUES('t.piper@covio.edu', 'Florence', 'Piper', '670 East Reeds Road, Richardson', 4697652413, 75080);
7 INSERT INTO artist(email, first_name, last_name, office_address, phone, zip)
8 VALUES('a.rosario@gmail.com', 'Anna', 'Rosario', '201 N Shady Shores Rd #36 Lake Dallas', 9543427886, 75065);
9 INSERT INTO artist(email, first_name, last_name, office_address, phone, zip)
10 VALUES('wang_r@gmail.com', 'Riley', 'Wang', '1307 W Water Creek Dr, Frisco', 4653782883, 75034);
11 INSERT INTO artist(email, first_name, last_name, office_address, phone, zip)
12 VALUES('lily.fern@gmail.com', 'Lily', 'Fernandez', '16577 Grilled Springs Rd, Plano', 9546712117, 75032);
13 INSERT INTO artist(email, first_name, last_name, office_address, phone, zip)
14 VALUES('m.wm987@neu.edu', 'Marcus', 'Wu', '985 West Georgehurst Street, Little Elm', 9543671993, 75065);
Data Output Messages Notifications
INSERT 8 1
Query returned successfully in 43 msec.
Ln 225, Col 25
Total rows: 0 of 0 Query complete 00:00:00.043
Query returned successfully in 43 msec. X
Ln 225, Col 25
```

Basic Queries:

- 1) Select all columns and all rows from one table - EXHIBITION table.

`SELECT * FROM exhibition;`



```

1  -----
2  ----Query 1: Select all columns and all rows from one table - EXHIBITION table.
3
4  SELECT * FROM exhibition;
5
6  -----
7  ----Query 2: Select five columns(first_name, last_name, email, phone, zip) and all rows from one table - ARTIST table.
8
9
10
11
12
13
14
15
16
17
18
19

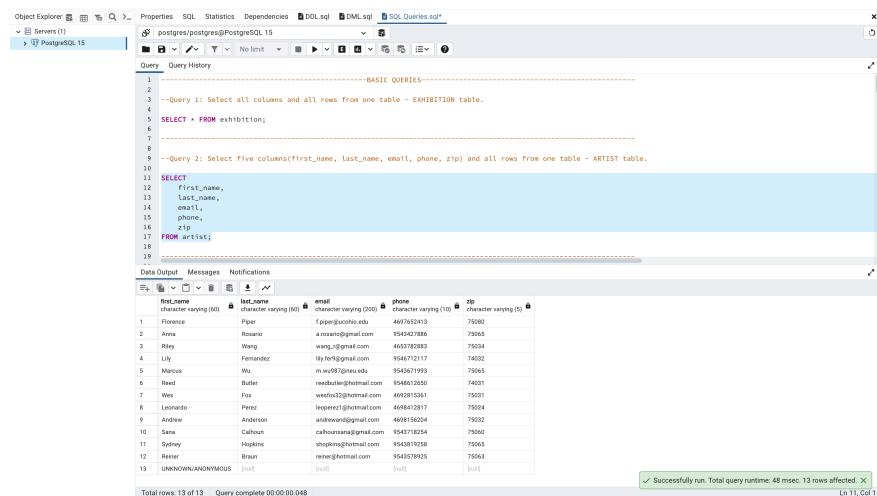
```

exhibition_id	name	venue	date	mode	character varying (10)	created_by	character varying (30)	date_created	modified_by	character varying (30)	date_modified
1	Artist Mart	1600 Grapevine Estate, TX, 77131	2023-02-16	Offline	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
2	Art Synergy	1800 Elmwood Ln, TX, 77131	2023-03-15	Offline	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
3	Drama	2000 Lakewood Blvd, TX, 77131	2023-01-01	Offline	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
4	Crafts	3000 Wellington, TX, 77112	2023-11-04	Offline	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
5	Gallery	2001 Eastside Blvd, TX, 70111	2023-12-08	Offline	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
6	Wall Craft	3001, Wellington, TX, 70112	2023-12-11	Offline	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
7	Villageal	2310 Eastside Blvd, TX, 70121	2023-11-21	Offline	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
8	Cinemas	Zoom	2023-11-23	Online	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
9	Art Ally	1000, Park Woods, TX, 77121	2023-11-23	Offline	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
10	Hues	Google Meet	2024-01-09	Online	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
11	Artistry	Zoom	2024-01-25	Online	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	
12	Wall Art	900, Fresno, TX, 78002	2024-01-05	Offline	postgres	postgres	2023-11-15	postgres	postgres	2023-11-15	

Total rows: 12 of 12 Query complete 00:00:00.042 ✓ Successfully run. Total query runtime: 42 msec. 12 rows affected. Ln 5, Col 1

- 2) Select five columns(first_name, last_name, email, phone, zip) and all rows from one table - ARTIST table.

`SELECT first_name, last_name, email, phone, zip FROM artist;`



```

1  -----
2  ----Query 1: Select all columns and all rows from one table - EXHIBITION table.
3
4  SELECT * FROM exhibition;
5
6  -----
7  ----Query 2: Select five columns(first_name, last_name, email, phone, zip) and all rows from one table - ARTIST table.
8
9
10
11
12
13
14
15
16
17
18
19

```

firstName	lastName	email	phone	zip
Florence	Piper	f.piper@uochio.edu	4697652413	75080
Alex	Hill	a.hill@uochio.edu	4697652409	75080
Nay	Wong	n.wong@uochio.edu	4697652403	75034
UV	Fernandez	l.fernandez@uochio.edu	4694672117	74032
Marcus	Wu	m.wu@uochio.edu	9549371919	75065
Reed	Butler	r.butler@uochio.edu	9548012826	74031
Wen	Fox	w.fox@uochio.edu	4692010361	75031
Leonardo	Perez	leonardo.perez@uochio.edu	4698420121	75024
Andrew	Anderson	andrew.anderson@uochio.edu	4691162624	75022
Sara	Galloway	s.galloway@uochio.edu	4691162624	75065
Sydney	Hostos	s.hostos@uochio.edu	9545197828	75065
Rever	Brian	r.rever@gmail.com	9545078925	75063
UNKNOWN/ANONYMOUS	[null]	[null]	[null]	[null]

Total rows: 13 of 13 Query complete 00:00:00.048 ✓ Successfully run. Total query runtime: 48 msec. 13 rows affected. Ln 11, Col 1

3) Select all columns from all rows from one view - ArtistPortfolio.

`SELECT * FROM ArtistPortfolio;`

```

--Query 3: Select all columns from all rows from one view - ArtistPortfolio
--Query 4: Using a join on 2 tables - ARTIST and ARTWORK tables, select all columns and all rows from the tables without the use of a Cartesian product

```

ARTIST	ARTWORK title
Andrew Anderson	NULL
Anna Rosario	Mechanical Symphony
Anna Rosario	Urban Symphony
Anna Rosario	Whispers of the Breeze
Florence Piper	Lonely Flame
Florence Piper	Summer Mistress
Florence Piper	The Story of the Void
Leonardo Perez	Floating Gardens
Leonardo Perez	Midnight Reverie
Lily Fernandez	Galactic Embrace
Lily Fernandez	Metamorphosis
Lily Fernandez	Whimsical Wonderland
Marcus Wu	Celestial Reverie
Marcus Wu	Crimson Whispers
Marcus Wu	Harmony in Chaos
Marcus Wu	Melancholy Melodies
Reed Butler	Parallel Realms
Reiner Braun	Cerulean Dreams
Reiner Braun	Enigmatic Echoes
Reiner Braun	Harvest Harmony
Reiner Braun	Infinite Echo
Reiner Braun	Metropolis Mirage
Reiner Braun	Sunset Sonata
Riley Wang	Ethereal Dreams
Riley Wang	Silent Reflections
Sana Calhoun	NULL
Sydney Hopkins	Nebulous Odyssey
Sydney Hopkins	Rhythmic Ripples
Wes Fox	Serenity Embrace
UNKNOWN/ANONYMOUS	Luminous Nexus
UNKNOWN/ANONYMOUS	Silent Night
UNKNOWN/ANONYMOUS	The Scream
UNKNOWN/ANONYMOUS	Transcendent Tides
UNKNOWN/ANONYMOUS	Verdant Reverie

ARTIST	ARTWORK title
Andrew Anderson	NULL
Anna Rosario	Mechanical Symphony
Anna Rosario	Urban Symphony
Anna Rosario	Whispers of the Breeze
Florence Piper	Lonely Flame
Florence Piper	Summer Mistress
Florence Piper	The Story of the Void
Leonardo Perez	Floating Gardens
Leonardo Perez	Midnight Reverie
Lily Fernandez	Galactic Embrace
Lily Fernandez	Metamorphosis
Lily Fernandez	Whimsical Wonderland
Marcus Wu	Celestial Reverie
Marcus Wu	Crimson Whispers
Marcus Wu	Harmony in Chaos
Marcus Wu	Melancholy Melodies
Reed Butler	Parallel Realms
Reiner Braun	Cerulean Dreams
Reiner Braun	Enigmatic Echoes
Reiner Braun	Harvest Harmony
Reiner Braun	Infinite Echo
Reiner Braun	Metropolis Mirage
Reiner Braun	Sunset Sonata
Riley Wang	Ethereal Dreams
Riley Wang	Silent Reflections
Sana Calhoun	NULL
Sydney Hopkins	Nebulous Odyssey
Sydney Hopkins	Rhythmic Ripples
Wes Fox	Serenity Embrace
UNKNOWN/ANONYMOUS	Luminous Nexus
UNKNOWN/ANONYMOUS	Silent Night
UNKNOWN/ANONYMOUS	The Scream
UNKNOWN/ANONYMOUS	Transcendent Tides
UNKNOWN/ANONYMOUS	Verdant Reverie

- 4) Using a join on 2 tables - ARTIST and ARTWORK tables, select all columns and all rows from the tables without the use of a Cartesian product.

```
SELECT * FROM artist
```

JOIN artwork

USING (artist_id);

```
Object Explorer SQL Statistics Dependencies DDL.sql DML.sql SQL Queries sql*  
postres/postres@PostgreSQL15  
No limit  
Query History  
Query  
--Query 4: Using a Join on 2 tables - ARTIST and ARTWORK tables, select all columns and all rows from the tables without the use of a Cartesian product  
28  
29  
30 SELECT * FROM artist  
31 JOIN artwork  
32 ON (artist.artist_id) =  
33  
34  
Data Output Messages Notifications  
E...  


|    | artist_id | email                 | first_name | last_name | office_address                          | phone        | zip   | created_by | date_created | modified_by |
|----|-----------|-----------------------|------------|-----------|-----------------------------------------|--------------|-------|------------|--------------|-------------|
| 1  | 1         | t.pip@woohoo.edu      | Florence   | Piper     | 670 East Werner Road, Richardson,       | (469)7552413 | 75080 | postgres   | 2023-11-15   | postgres    |
| 2  | 2         | l.florence@woohoo.edu | Florence   | Piper     | 670 East Werner Road, Richardson,       | (469)7552413 | 75080 | postgres   | 2023-11-15   | postgres    |
| 3  | 1         | t.pip@woohoo.edu      | Florence   | Piper     | 670 East Werner Road, Richardson,       | (469)7552413 | 75080 | postgres   | 2023-11-15   | postgres    |
| 4  | 2         | a.mario@woohoo.com    | Anna       | Rosario   | 201 N Shady Shores Rd #510 Lake Dallas, | (954)3427890 | 75045 | postgres   | 2023-11-15   | postgres    |
| 5  | 2         | a.mario@woohoo.com    | Anna       | Rosario   | 201 N Shady Shores Rd #510 Lake Dallas, | (954)3427890 | 75045 | postgres   | 2023-11-15   | postgres    |
| 6  | 2         | a.mario@woohoo.com    | Anna       | Rosario   | 201 N Shady Shores Rd #510 Lake Dallas, | (954)3427890 | 75045 | postgres   | 2023-11-15   | postgres    |
| 7  | 3         | wang@gnat.com         | Riley      | Wang      | 2307 Water Creek Dr. Frisco             | (463)5722883 | 75034 | postgres   | 2023-11-15   | postgres    |
| 8  | 3         | wang@gnat.com         | Riley      | Wang      | 2307 Water Creek Dr. Frisco             | (463)5722883 | 75034 | postgres   | 2023-11-15   | postgres    |
| 9  | 4         | ily.fern@gmail.com    | Lily       | Fernandez | 15577 Gwinwood Springs Rd, Plano        | (954)6072117 | 75022 | postgres   | 2023-11-15   | postgres    |
| 10 | 4         | ily.fern@gmail.com    | Lily       | Fernandez | 15577 Gwinwood Springs Rd, Plano        | (954)6072117 | 75022 | postgres   | 2023-11-15   | postgres    |
| 11 | 5         | m.westfox@juno.com    | Marcus     | Wu        | 905 West Georgetown Street, Dallas, Elm | (954)3811193 | 75061 | postgres   | 2023-11-15   | postgres    |
| 12 | 5         | m.westfox@juno.com    | Marcus     | Wu        | 905 West Georgetown Street, Dallas, Elm | (954)3811193 | 75061 | postgres   | 2023-11-15   | postgres    |
| 13 | 5         | m.westfox@juno.com    | Marcus     | Wu        | 905 West Georgetown Street, Dallas, Elm | (954)3811193 | 75061 | postgres   | 2023-11-15   | postgres    |
| 14 | 7         | westfox@gmail.com     | Wes        | Fox       | 2822 Independence Park, Frisco          | (954)2935361 | 75031 | postgres   | 2023-11-15   | postgres    |
| 15 | 8         | leopre@t@hotmail.com  | Leopoldo   | Perez     | 301 Bushyfork Crest, Denton             | (469)8412817 | 75024 | postgres   | 2023-11-15   | postgres    |
| 16 | 8         | leopre@t@hotmail.com  | Leopoldo   | Perez     | 301 Bushyfork Crest, Denton             | (469)8412817 | 75024 | postgres   | 2023-11-15   | postgres    |
| 17 | 11        | shopkins@hotmail.com  | Sydney     | Hopkins   | 681 Pin Oak Ave., Dallas                | (954)3819258 | 75065 | postgres   | 2023-11-15   | postgres    |
| 18 | 11        | shopkins@hotmail.com  | Sydney     | Hopkins   | 681 Pin Oak Ave., Dallas                | (954)3819258 | 75065 | postgres   | 2023-11-15   | postgres    |
| 19 | 12        | rene@rehm.com         | Renee      | Braun     | 913 Marley Ave, Dallas                  | (954)3539293 | 75063 | postgres   | 2023-11-15   | postgres    |
| 20 | 12        | rene@rehm.com         | Renee      | Braun     | 913 Marley Ave, Dallas                  | (954)3539293 | 75063 | postgres   | 2023-11-15   | postgres    |
| 21 | 12        | rene@rehm.com         | Renee      | Braun     | 913 Marley Ave, Dallas                  | (954)3539293 | 75063 | postgres   | 2023-11-15   | postgres    |
| 22 | 12        | rene@rehm.com         | Renee      | Braun     | 913 Marley Ave, Dallas                  | (954)3539293 | 75063 | postgres   | 2023-11-15   | postgres    |



Total rows: 32. Query complete (00:00:00.035). Lo Gal 201



✓ Successfully run. Total query runtime: 35 msec. 32 rows affected.


```

5) Select and order data - based on date retrieved from one table - EXHIBITION table.

`SELECT * FROM exhibition`

`ORDER BY date ASC;`

The screenshot shows the Object Explorer interface with a connection to 'PostgreSQL15'. A query is running in the 'Query' tab:

```

33
34
35 -->Query 5: Select and order data - based on date retrieved from one table - EXHIBITION table
36
37 SELECT * FROM exhibition
38 ORDER BY date ASC;
39
40

```

The results are displayed in the 'Data Output' tab:

exhibition_id	name	venue	date	mode	created_by	date_created	modified_by	date_modified
1	Artist Mart	1500 Grapevine Estate, TX, 77331	2023-02-16	Offline	postgres	2023-11-15	postgres	2023-11-15
2	Art Synergy	1800 South Lake, TX, 77092	2023-03-15	Offline	postgres	2023-11-15	postgres	2023-11-15
3	Decora	2000 Eastside Blvd, TX, 70111	2023-12-01	Offline	postgres	2023-11-15	postgres	2023-11-15
4	Ouftaff	3000, Wellington, TX, 70112	2023-12-04	Offline	postgres	2023-11-15	postgres	2023-11-15
5	Gallerie	2001, Eastside Blvd, TX, 70111	2023-12-08	Offline	postgres	2023-11-15	postgres	2023-11-15
6	A Creative Art	2116, Wellington, TX, 70112	2023-12-15	Offline	postgres	2023-11-15	postgres	2023-11-15
7	Vision Art	2316, Eastside Blvd, TX, 70123	2023-12-11	Offline	postgres	2023-11-15	postgres	2023-11-15
8	Creams	Zoom	2023-12-23	Offline	postgres	2023-11-15	postgres	2023-11-15
9	Art Ally	1000 Park Woods, TX, 71231	2023-12-23	Offline	postgres	2023-11-15	postgres	2023-11-15
10	Hues	Google Meet	2024-01-09	Online	postgres	2023-11-15	postgres	2023-11-15
11	Artistry	Zoom	2024-01-25	Online	postgres	2023-11-15	postgres	2023-11-15
12	Wall Art	3001, Fresco, TX, 75032	2024-02-05	Offline	postgres	2023-11-15	postgres	2023-11-15

Total rows: 12 of 12 Query complete 00:00:00.0962

Successfully run. Total query runtime: 62 msec. 12 rows affected. Ln 32 Col 1

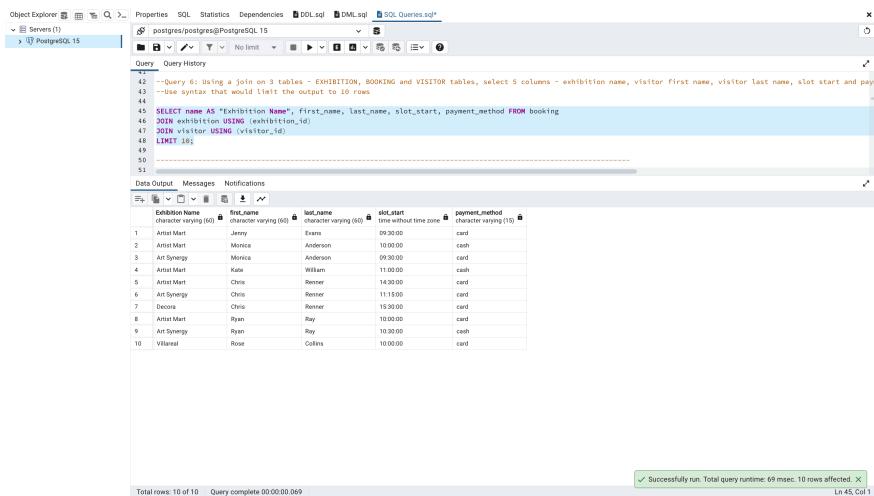
- 6) Using a join on 3 tables - EXHIBITION, BOOKING and VISITOR tables, select 5 columns - exhibition name, visitor first name, visitor last name, slot start and payment method from the 3 tables. Use syntax that would limit the output to 10 rows

```
SELECT name AS "Exhibition Name", first_name, last_name, slot_start, payment_method FROM booking
```

JOIN exhibition USING (exhibition_id)

JOIN visitor USING (visitor_id)

LIMIT 10;



```

41
42 --Query 4: Using a join on 3 tables - EXHIBITION, BOOKING and VISITOR tables, select 5 columns - exhibition name, visitor first name, visitor last name, slot start and pay
43 --use syntax that would limit the output to 10 rows
44
45 SELECT name AS "Exhibition Name", first_name, last_name, slot_start, payment_method FROM booking
46 JOIN exhibition USING (exhibition_id)
47 JOIN visitor USING (visitor_id)
48 LIMIT 10;
49
50
51

```

	Exhibition Name	first_name	last_name	slot_start	payment_method
1	ArtistMart	Jenifer	Evans	09:30:00	card
2	ArtistMart	Monica	Anderson	10:00:00	cash
3	Art Synergy	Monica	Anderson	09:30:00	card
4	ArtistMart	Kate	William	11:00:00	cash
5	ArtistMart	Chris	Renner	14:30:00	card
6	Art Synergy	Chris	Renner	11:15:00	card
7	ArtistMart	Chris	Renner	13:30:00	card
8	ArtistMart	Ryan	Ray	10:00:00	card
9	Art Synergy	Ryan	Ray	10:30:00	cash
10	Visional	Rose	Collins	10:00:00	card

Total rows: 10 of 10 Query complete 00:00:00.069 ✓ Successfully run. Total query runtime: 69 msec. 10 rows affected. Ln 45, Col 1

- 7) Select distinct rows - art_id, title, description, exhibition_id, name using joins on 3 tables - ARTWORK, DISPLAY and EXHIBITION table.

SELECT DISTINCT

A.art_id, A.title, A.description, E.exhibition_id, E.name

FROM display

JOIN artwork A USING (art_id)

JOIN exhibition E USING (exhibition_id);

```

-- Query 7: Select distinct rows - art_id, title, description, exhibition_id, name using joins on 3 tables - ARTWORK, DISPLAY and EXHIBITION tables
SELECT DISTINCT
    art_id,
    title,
    description,
    exhibition_id,
    name
FROM display
JOIN artwork A USING (art_id)
JOIN exhibition E USING (exhibition_id);

```

art_id	title	description	exhibition_id	name
32	Silent Night	A tranquil landscape in the wintery landscapes of The Alps	7	Villareal
2	The Story of the Void	A cosmic narrative, inviting contemplation on the mysteries within the vast emptiness of space, creating an expansive yet intimate visual experience	1	Artist Mart
5	Mechanical Symphony	A futuristic composition of metallic structures and intricate machinery, reflecting the harmonious dance of technology in an industrial landscape	1	Artist Mart
1	Lonely Flame	A solitary, flickering flame against a backdrop of profound darkness, symbolizing isolation and resilience in the face of solitude, evoking a sense of introspection and contemplation within the viewer	2	Art Synergy
2	The Story of the Void	A cosmic narrative, inviting contemplation on the mysteries within the vast emptiness of space, creating an expansive yet intimate visual experience	2	Art Synergy
16	Floating Gardens	Imaginary floating islands adorned with vibrant, fantastical flora, inviting viewers into a world of whimsy and color	4	Craftric
3	Summer Mistress	A capitation of the essence of the season in a vibrant dance of warm hues, depicting nature's beauty as a captivating and ephemeral enchantress	1	Artist Mart
7	Silent Reflections	A tranquil seascape portraying a lone sailboat adrift on calm waters, inviting viewers to contemplate the serenity found in solitary moments	2	Art Synergy
3	Summer Mistress	A capitation of the essence of the season in a vibrant dance of warm hues, depicting nature's beauty as a captivating and ephemeral enchantress	2	Art Synergy
25	Luminous Nexus	An exploration of light and shadow, where intersecting beams create a mesmerizing play of illumination and darkness	6	Wall Craft
4	Whispers of the Breeze	A serene landscape painting depicting a gentle meadow where delicate flowers sway in harmony with a soft breeze, capturing the ephemeral beauty of nature	1	Artist Mart
12	Celestial Reverie	A cosmic artwork that explores the mysteries of the universe, with swirling galaxies and ethereal lights that invite contemplation on the infinite	3	Decorra
15	Midnight Reverie	A nocturnal scene featuring a mysterious moonlit forest, where shadows and silver light create an enchanting atmosphere	3	Decorra
26	Verdant Reverie	A lush and verdant forest scene, where vibrant greenery and dappled sunlight invite viewers into a world of natural splendor	5	Galleria
27	Transcendent Tides	A dynamic representation of ocean waves in various states, symbolizing the relentless ebb and flow of life's experiences	8	Cisernos
1	Lonely Flame	A solitary, flickering flame against a backdrop of profound darkness, symbolizing isolation and resilience in the face of solitude, evoking a sense of introspection and contemplation within the viewer	1	Artist Mart
4	Whispers of the Breeze	A serene landscape painting depicting a gentle meadow where delicate flowers sway in harmony with a soft breeze, capturing the ephemeral beauty of nature	2	Art Synergy
10	Whimsical Wonderland	A whimsical and playful painting depicting a fantastical world filled with imaginative creatures and vibrant, otherworldly landscapes	3	Decorra
5	Mechanical Symphony	A futuristic composition of metallic structures and intricate machinery, reflecting the harmonious dance of technology in an industrial landscape	2	Art Synergy

art_id	title	description	exhibition_id	name
32	Silent Night	A tranquil landscape in the wintery landscapes of The Alps	7	Villareal
2	The Story of the Void	A cosmic narrative, inviting contemplation on the mysteries within the vast emptiness of space, creating an expansive yet intimate visual experience	1	Artist Mart
5	Mechanical Symphony	A futuristic composition of metallic structures and intricate machinery, reflecting the harmonious dance of technology in an industrial landscape	1	Artist Mart
1	Lonely Flame	A solitary, flickering flame against a backdrop of profound darkness, symbolizing isolation and resilience in the face of solitude, evoking a sense of introspection and contemplation within the viewer	2	Art Synergy
2	The Story of the Void	A cosmic narrative, inviting contemplation on the mysteries within the vast emptiness of space, creating an expansive yet intimate visual experience	2	Art Synergy
16	Floating Gardens	Imaginary floating islands adorned with vibrant, fantastical flora, inviting viewers into a world of whimsy and color	4	Craftric
3	Summer Mistress	A capitation of the essence of the season in a vibrant dance of warm hues, depicting nature's beauty as a captivating and ephemeral enchantress	1	Artist Mart
7	Silent Reflections	A tranquil seascape portraying a lone sailboat adrift on calm waters, inviting viewers to contemplate the serenity found in solitary moments	2	Art Synergy
3	Summer Mistress	A capitation of the essence of the season in a vibrant dance of warm hues, depicting nature's beauty as a captivating and ephemeral enchantress	2	Art Synergy
25	Luminous Nexus	An exploration of light and shadow, where intersecting beams create a mesmerizing play of illumination and darkness	6	Wall Craft
4	Whispers of the Breeze	A serene landscape painting depicting a gentle meadow where delicate flowers sway in harmony with a soft breeze, capturing the ephemeral beauty of nature	1	Artist Mart
12	Celestial Reverie	A cosmic artwork that explores the mysteries of the universe, with swirling galaxies and ethereal lights that invite contemplation on the infinite	3	Decorra
15	Midnight Reverie	A nocturnal scene featuring a mysterious moonlit forest, where shadows and silver light create an enchanting atmosphere	3	Decorra
26	Verdant Reverie	A lush and verdant forest scene, where vibrant greenery and dappled sunlight invite viewers into a world of natural splendor	5	Galleria
27	Transcendent Tides	A dynamic representation of ocean waves in various states, symbolizing the relentless ebb and flow of life's experiences	8	Cisernos
1	Lonely Flame	A solitary, flickering flame against a backdrop of profound darkness, symbolizing isolation and resilience in the face of solitude, evoking a sense of introspection and contemplation within the viewer	1	Artist Mart
4	Whispers of the Breeze	A serene landscape painting depicting a gentle meadow where delicate flowers sway in harmony with a soft breeze, capturing the ephemeral beauty of nature	2	Art Synergy
10	Whimsical Wonderland	A whimsical and playful painting depicting a fantastical world filled with imaginative creatures and vibrant, otherworldly landscapes	3	Decorra
5	Mechanical Symphony	A futuristic composition of metallic structures and intricate machinery, reflecting the harmonious dance of technology in an industrial landscape	2	Art Synergy

- 8) Use GROUP BY and HAVING in a select statement using one or more tables - ARTWORK table.

```
SELECT artist_id, COUNT(artist_id)
```

```
FROM artwork
```

```
GROUP BY artist_id
```

```
HAVING COUNT(artist_id)>2;
```

The screenshot shows the PostgreSQL Management Studio interface. The query window contains the following SQL code:

```
--Query 8: Use GROUP BY and HAVING in a select statement using one or more tables - ARTWORK table
SELECT artist_id, COUNT(artist_id)
FROM artwork
GROUP BY artist_id
HAVING COUNT(artist_id)>2;
```

The results pane displays a table with three columns: artist_id, count, and type. The data is as follows:

artist_id	count	type
1	4	3
2	13	5
3	2	3
4	12	6
5	1	3
6	5	4

At the bottom of the interface, a message indicates "Successfully run. Total query runtime: 43 msec. 6 rows affected." and "Ln 64, Col 1".

- 9) Use IN clause to select data from one or more tables - ARTWORK table.

```
SELECT title, description, medium FROM artwork
```

```
WHERE medium IN ('Oil', 'Acrylic', 'Watercolor');
```

The screenshot shows the PostgreSQL Management Studio interface. The query window contains the following SQL code:

```
--Query 9: Use IN clause to select data from one or more tables - ARTWORK table
SELECT title, description, medium
FROM artwork
WHERE medium IN ('Oil', 'Acrylic', 'Watercolor');
```

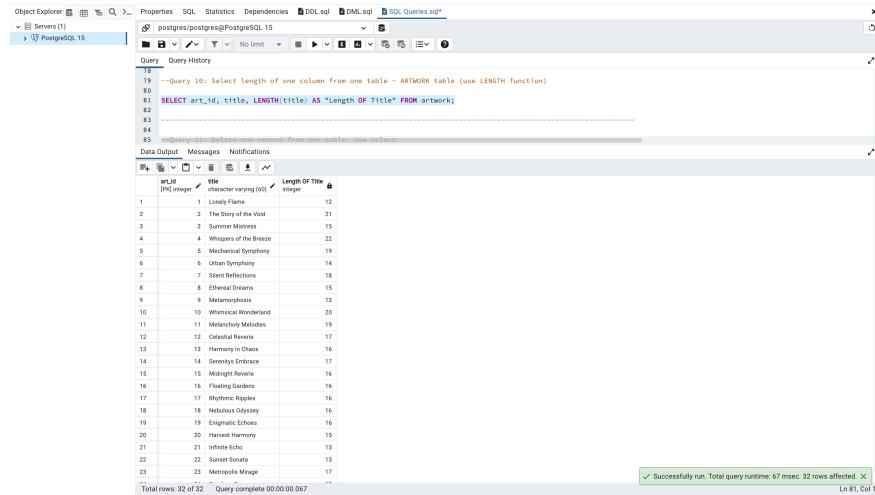
The results pane displays a table with four columns: title, description, medium, and character varying (100). The data is as follows:

title	description	medium	character varying (100)
Landscape Flame	A fiery, turbulent flame against a backdrop of profound darkness, symbolizing isolation and resilience in the face of solitude, evoking a sense of introspection and contemplation within the viewer.	Oil	
The Story of the Void	A cosmic narrative, inviting contemplation on the mystique within the vast emptiness of space, creating an expansive yet intimate visual experience.	Oil	
Summer Mirthes	A capture of the essence of the season in a vibrant burst of warm hues, depicting nature's beauty as a captivation and ephemeral enchantment.	Oil	
Whispers of the Breeze	A serene landscape painting depicting a gentle meadow where delicate flowers sway in harmony with a soft breeze, capturing the ephemeral beauty of nature.	Oil	
Mechanical Symmetry	A futuristic composition of metallic structures and intricate machinery, reflecting the harmonious dance of technology in an industrial landscape.	Oil	
Urban Symmetry	An abstract composition pulsing with vibrant colors and dynamic lines, illustrating the rhythmic energy of urban life in a bustling metropolis.	Oil	
Silent Reflections	A tranquil seascape portraying a lone sailboat adrift on calm waters, inviting viewers to contemplate the serenity found in solitary moments.	Watercolor	
Morphogenesis	An abstract expressionist piece featuring a tapestry of chaotic, translucent shapes and vibrant colors, symbolizing the organic beauty that can emerge from disorder.	Acrylic	
Harmony in Chaos	Imaginative florals interwoven with vibrant, fractal-like forms, inviting viewers into a world of symmetry and order amidst chaos.	Oil	
Flowing Gardens	An abstract expressionist piece featuring a tapestry of chaotic, translucent shapes and vibrant colors, symbolizing the organic beauty that can emerge from disorder.	Watercolor	
Infinite Echo	An exploration of symmetry and reflection, creating an illusion of endless corridors and repeating patterns.	Oil	
Sunset Sonata	A warm and vibrant portrayal of a sun setting over a serene landscape, capturing the poetic beauty of the evening sky.	Oil	
Metropolis Merge	An abstract chrysalis with distorted perspectives and vivid colors, reflecting the dynamic energy of an ever-evolving urban environment.	Acrylic	
Luminous Nexus	An exploration of light and shadow, where intersecting beams create a mesmerizing play of illumination and darkness.	Oil	
Verdant Reverie	A lush and verdant forest scene, where vibrant greenery and dappled sunlight invite viewers into a world of natural splendor.	Acrylic	
Crimson Whispers	An abstract expressionist piece dominated by deep red tones, evoking passion and intensity in a visual symphony of emotions.	Oil	
The Scream	An expressive composition based on a scream piercing through nature while on a walk, after two companions leave someone.	Acrylic	
Silent Night	A tranquil landscape in the wintry landscapes of The Alps.	Watercolor	

At the bottom of the interface, a message indicates "Successfully run. Total query runtime: 46 msec. 18 rows affected." and "Ln 73, Col 1".

10) Select length of one column from one table - ARTWORK table (use LENGTH function).

```
SELECT art_id, title, LENGTH(title) AS "Length OF Title" FROM artwork;
```



The screenshot shows the SSMS interface with a query window containing the following code:

```
--Query 10: Select length of one column from one table - ARTWORK table (use LENGTH function)
SELECT art_id, title, LENGTH(title) AS "Length OF Title" FROM artwork;
```

The results grid displays the following data:

art_id	title	Length OF Title
1	Lonely Flame	12
2	The Story of the Void	21
3	Summer Mistress	15
4	Whispers of the Breeze	22
5	Mechanical Symphony	19
6	Urban Symphony	14
7	Silent Reflections	18
8	Ethereal Dreams	15
9	Metamorphosis	13
10	Whimsical Wonderland	20
11	Melancholy Melodies	19
12	Celestial Reverie	17
13	Harmony in Chaos	16
14	Serenity Embrace	17
15	Midnight Reverie	16
16	Floating Gardens	16
17	Rhythmic Ripples	16
18	Nebulous Odyssey	16
19	Enigmatic Echoes	16
20	Harvest Harmony	15
21	Infinite Echo	13
22	Sunset Sonata	13
23	Metropolis Mirage	17
24	Cerulean Dreams	15
25	Luminous Nexus	14
26	Verdant Reverie	15
27	Transcendent Tides	18
28	Crimson Whispers	16
29	Galactic Embrace	16
30	Parallel Realms	15
31	The Scream	10
32	Silent Night	12

Total rows: 32 of 32 Query complete 00:00:00.067

Successfully run. Total query runtime: 67 msec. 32 rows affected.

art_id	title	Length OF Title
1	Lonely Flame	12
2	The Story of the Void	21
3	Summer Mistress	15
4	Whispers of the Breeze	22
5	Mechanical Symphony	19
6	Urban Symphony	14
7	Silent Reflections	18
8	Ethereal Dreams	15
9	Metamorphosis	13
10	Whimsical Wonderland	20
11	Melancholy Melodies	19
12	Celestial Reverie	17
13	Harmony in Chaos	16
14	Serenity Embrace	17
15	Midnight Reverie	16
16	Floating Gardens	16
17	Rhythmic Ripples	16
18	Nebulous Odyssey	16
19	Enigmatic Echoes	16
20	Harvest Harmony	15
21	Infinite Echo	13
22	Sunset Sonata	13
23	Metropolis Mirage	17
24	Cerulean Dreams	15
25	Luminous Nexus	14
26	Verdant Reverie	15
27	Transcendent Tides	18
28	Crimson Whispers	16
29	Galactic Embrace	16
30	Parallel Realms	15
31	The Scream	10
32	Silent Night	12

11) Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement..

```
SELECT * FROM booking;
```

```
BEGIN;
```

```
DELETE FROM booking
```

```
WHERE booking_id = 1;
```

```
SELECT * FROM booking;
```

```
ROLLBACK;
```

```
SELECT * FROM booking;
```

The screenshot shows the Object Explorer and Properties tabs for a database named 'PostgreSQL15'. The main window displays the 'booking' table with 19 rows of data. The columns are: booking_id, visitor_id, exhibition_id, time, slot_start, slot_end, payment_method, created_by, date_created, modified_by, and date_modified. The data includes various booking entries with visitor IDs ranging from 1 to 19 and exhibition IDs from 2 to 5.

booking_id	visitor_id	exhibition_id	time	slot_start	slot_end	payment_method	created_by	date_created	modified_by	date_modified
1	1	3	13:00:00	09:30:00	12:00:00	cash	postgres	2023-11-15	postgres	2023-11-15
2	2	2	14:00:00	10:00:00	13:00:00	cash	postgres	2023-11-15	postgres	2023-11-15
3	3	3	15:00:00	11:00:00	13:00:00	cash	postgres	2023-11-15	postgres	2023-11-15
4	4	4	15:45:00	14:30:00	17:00:00	card	postgres	2023-11-15	postgres	2023-11-15
5	5	5	16:00:00	10:00:00	18:00:00	card	postgres	2023-11-15	postgres	2023-11-15
6	6	2	17:00:00	09:30:00	17:00:00	card	postgres	2023-11-15	postgres	2023-11-15
7	7	3	18:00:00	10:00:00	18:00:00	cash	postgres	2023-11-15	postgres	2023-11-15
8	8	10	18:00:00	11:00:00	13:00:00	cash	postgres	2023-11-15	postgres	2023-11-15
9	9	19	21:45:00	10:20:00	11:30:00	cash	postgres	2023-11-15	postgres	2023-11-15
10	10	4	21:00:00	11:15:00	13:00:00	card	postgres	2023-11-15	postgres	2023-11-15
11	11	21	22:00:00	18:00:00	20:00:00	card	postgres	2023-11-15	postgres	2023-11-15
12	12	15	17:45:00	16:00:00	17:45:00	cash	postgres	2023-11-15	postgres	2023-11-15
13	13	21	18:00:00	14:00:00	14:00:00	card	postgres	2023-11-15	postgres	2023-11-15
14	14	7	19:00:00	10:00:00	12:00:00	cash	postgres	2023-11-15	postgres	2023-11-15
15	15	24	19:00:00	10:00:00	12:00:00	cash	postgres	2023-11-15	postgres	2023-11-15
16	16	4	19:00:00	15:30:00	18:00:00	card	postgres	2023-11-15	postgres	2023-11-15
17	17	7	20:00:00	17:00:00	20:00:00	card	postgres	2023-11-15	postgres	2023-11-15
18	18	15	21:25:00	19:30:00	21:00:00	card	postgres	2023-11-15	postgres	2023-11-15
19	19	9	13:30:00	10:30:00	13:20:00	card	postgres	2023-11-15	postgres	2023-11-15

Total rows: 19 of 19 Query complete 00:00:00.041 Successfully run. Total query runtime: 41 msec. 19 rows affected. Un 87, Col 1

The screenshot shows the Object Explorer and Properties tabs for a database named 'PostgreSQL15'. The main window displays the 'booking' table with 18 rows of data. The columns are: booking_id, visitor_id, exhibition_id, time, slot_start, slot_end, payment_method, created_by, date_created, modified_by, and date_modified. The data includes various booking entries with visitor IDs ranging from 1 to 19 and exhibition IDs from 2 to 5, excluding the row where booking_id = 1.

booking_id	visitor_id	exhibition_id	time	slot_start	slot_end	payment_method	created_by	date_created	modified_by	date_modified
1	2	2	14:20:00	10:00:00	13:30:00	cash	postgres	2023-11-15	postgres	2023-11-15
2	3	3	15:00:00	10:00:00	13:30:00	cash	postgres	2023-11-15	postgres	2023-11-15
3	4	4	15:45:00	10:00:00	13:30:00	card	postgres	2023-11-15	postgres	2023-11-15
4	5	5	16:00:00	10:00:00	14:00:00	card	postgres	2023-11-15	postgres	2023-11-15
5	6	2	17:00:00	09:30:00	17:00:00	card	postgres	2023-11-15	postgres	2023-11-15
6	7	5	17:45:00	10:30:00	18:00:00	cash	postgres	2023-11-15	postgres	2023-11-15
7	8	10	18:00:00	11:30:00	12:00:00	card	postgres	2023-11-15	postgres	2023-11-15
8	9	19	21:45:00	10:20:00	11:30:00	cash	postgres	2023-11-15	postgres	2023-11-15
9	10	4	21:00:00	11:15:00	13:00:00	card	postgres	2023-11-15	postgres	2023-11-15
10	11	21	22:00:00	11:00:00	20:00:00	card	postgres	2023-11-15	postgres	2023-11-15
11	12	21	22:00:00	11:00:00	17:00:00	cash	postgres	2023-11-15	postgres	2023-11-15
12	13	21	23:45:00	11:00:00	14:00:00	card	postgres	2023-11-15	postgres	2023-11-15
13	14	7	19:00:00	10:00:00	12:00:00	card	postgres	2023-11-15	postgres	2023-11-15
14	15	26	19:00:00	10:00:00	12:00:00	cash	postgres	2023-11-15	postgres	2023-11-15
15	16	4	19:00:00	15:30:00	18:00:00	card	postgres	2023-11-15	postgres	2023-11-15
16	17	7	20:00:00	17:00:00	20:00:00	card	postgres	2023-11-15	postgres	2023-11-15
17	18	15	21:50:00	19:30:00	21:00:00	card	postgres	2023-11-15	postgres	2023-11-15
18	19	9	13:30:00	10:30:00	13:20:00	card	postgres	2023-11-15	postgres	2023-11-15

Total rows: 18 of 18 Query complete 00:00:00.100 Successfully run. Total query runtime: 100 msec. 18 rows affected. Un 87, Col 1

12) Update one record from one table. Use select statements to demonstrate the table contents before and after the UPDATE statement.

```
SELECT * FROM display;
```

BEGIN;

UPDATE display SET location = 'N/A'

WHERE display_id = 1;

```
SELECT * FROM display;
```

ROLLBACK;

SELECT * FI

Object Explorer >
▼ Servers (1)

```

Query History
100
101
102 --Query 12: Update one record from one table. Use select statements to demonstrate the table contents before and after the UPDATE statement.
103
104 SELECT * FROM display;
105
106 BEGIN;
107
108 UPDATE display
109 SET location = 'N/A';
110
Data Output Messages Notifications



|    | display_id | art_id | ref_id   | start_time | end_time | location | artist_presence | created_by | date_created | modified_by | date_modified |
|----|------------|--------|----------|------------|----------|----------|-----------------|------------|--------------|-------------|---------------|
| 1  | 1          | 1      |          | 11:00:00   | 18:00:00 | A21      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 2  | 2          | 2      |          | 12:00:00   | 17:00:00 | B34      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 3  | 3          | 3      |          | 10:00:00   | 19:00:00 | C42      | false           | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 4  | 4          | 4      |          | 09:00:00   | 17:00:00 | A18      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 5  | 5          | 5      |          | 10:00:00   | 19:00:00 | D53      | false           | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 6  | 6          | 1      | 2        | 11:00:00   | 19:00:00 | A09      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 7  | 7          | 2      | 2        | 12:00:00   | 19:00:00 | C18      | false           | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 8  | 8          | 3      | 2        | 09:00:00   | 14:00:00 | B18      | false           | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 9  | 9          | 4      | 2        | 14:00:00   | 20:00:00 | A31      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 10 | 10         | 5      | 2        | 10:00:00   | 14:00:00 | C42      | false           | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 11 | 11         | 7      | 2        | 09:00:00   | 15:00:00 | B37      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 12 | 12         | 15     | 3        | 15:00:00   | 21:00:00 | C29      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 13 | 13         | 10     | 3        | 16:00:00   | 21:00:00 | B12      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 14 | 14         | 11     | 3        | 17:00:00   | 22:00:00 | C01      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 15 | 15         | 14     | 4        | 12:00:00   | 22:00:00 | C04      | false           | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 16 | 16         | 5      | 10:00:00 | 19:00:00   | D09      | false    | postgres        | 2023-11-15 | postgres     | 2023-11-15  |               |
| 17 | 17         | 25     | 13:00:00 | 22:00:00   | D16      | false    | postgres        | 2023-11-15 | postgres     | 2023-11-15  |               |
| 18 | 18         | 32     | 7        | 19:00:00   | 22:00:00 | B19      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |
| 19 | 19         | 27     | 8        | 13:00:00   | 20:00:00 | C30      | true            | postgres   | 2023-11-15   | postgres    | 2023-11-15    |


```

Object Explorer Properties SQL Statistics Dependencies DDL.sql DML.sql SQL Queries.sql*

Postgres/postgres@PostgreSQL 15

No limit

Query Query History

```

105
106 BEGIN;
107
108 UPDATE display
109 SET location = 'N/A'
110 WHERE display_id = 1;
111
112 SELECT * FROM display;
113
114 ROLLBACK;

```

Data Output Messages Notifications

display_id	art_id	exhibition_id	start_time	end_time	location	artist_presence	created_by	date_created	modified_by	date_modified
1	2	2	12:00:00	17:00:00	B34	true	postgres	2023-11-15	postgres	2023-11-15
2	3	3	12:00:00	17:00:00	C40	false	postgres	2023-11-15	postgres	2023-11-15
3	4	4	09:00:00	17:00:00	A18	true	postgres	2023-11-15	postgres	2023-11-15
4	5	5	10:00:00	19:00:00	D13	true	postgres	2023-11-15	postgres	2023-11-15
5	6	1	21:00:00	19:00:00	A05	true	postgres	2023-11-15	postgres	2023-11-15
6	7	2	2:00:00	19:00:00	C18	false	postgres	2023-11-15	postgres	2023-11-15
7	8	3	2:00:00	14:00:00	B18	false	postgres	2023-11-15	postgres	2023-11-15
8	9	4	2:14:00	20:00:00	A31	true	postgres	2023-11-15	postgres	2023-11-15
9	10	5	2:10:00	14:00:00	C42	false	postgres	2023-11-15	postgres	2023-11-15
10	11	7	2:15:00	15:00:00	B37	true	postgres	2023-11-15	postgres	2023-11-15
11	12	15	3:15:00	21:00:00	C29	true	postgres	2023-11-15	postgres	2023-11-15
12	13	10	3:14:00	21:00:00	B10	true	postgres	2023-11-15	postgres	2023-11-15
13	14	12	3:05:00	14:00:00	D21	true	postgres	2023-11-15	postgres	2023-11-15
14	15	16	4:12:00	22:00:00	C04	false	postgres	2023-11-15	postgres	2023-11-15
15	16	26	5:10:00	19:00:00	D09	false	postgres	2023-11-15	postgres	2023-11-15
16	17	25	6:13:00	22:00:00	D16	false	postgres	2023-11-15	postgres	2023-11-15
17	18	32	7:19:00	22:00:00	B19	true	postgres	2023-11-15	postgres	2023-11-15
18	19	27	8:13:00	20:00:00	C05	true	postgres	2023-11-15	postgres	2023-11-15
19	1	1	11:00:00	18:00:00	N/A	true	postgres	2023-11-15	postgres	2023-11-15

✓ Successfully run. Total query runtime: 47 msec. 19 rows affected. X

Total rows: 19 of 19 Query complete 00:00:00.047

Ln 106, Col 1

Object Explorer Properties SQL Statistics Dependencies DDL.sql DML.sql SQL Queries.sql*

Postgres/postgres@PostgreSQL 15

No limit

Query Query History

```

109 SET location = 'N/A'
110 WHERE display_id = 1;
111
112 SELECT * FROM display;
113
114 ROLLBACK;
115
116 SELECT * FROM display;
117
118 -----

```

Data Output Messages Notifications

display_id	art_id	exhibition_id	start_time	end_time	location	artist_presence	created_by	date_created	modified_by	date_modified
1	1	1	11:00:00	18:00:00	A21	true	postgres	2023-11-15	postgres	2023-11-15
2	2	2	12:00:00	17:00:00	B34	false	postgres	2023-11-15	postgres	2023-11-15
3	3	3	10:00:00	19:00:00	C42	false	postgres	2023-11-15	postgres	2023-11-15
4	4	4	09:00:00	17:00:00	A18	true	postgres	2023-11-15	postgres	2023-11-15
5	5	5	10:00:00	19:00:00	D13	true	postgres	2023-11-15	postgres	2023-11-15
6	6	1	21:00:00	19:00:00	A05	true	postgres	2023-11-15	postgres	2023-11-15
7	7	2	12:00:00	18:00:00	C18	false	postgres	2023-11-15	postgres	2023-11-15
8	8	3	2:00:00	14:00:00	B18	false	postgres	2023-11-15	postgres	2023-11-15
9	9	4	2:14:00	20:00:00	A31	true	postgres	2023-11-15	postgres	2023-11-15
10	10	5	2:10:00	14:00:00	C42	false	postgres	2023-11-15	postgres	2023-11-15
11	11	7	2:15:00	15:00:00	B37	true	postgres	2023-11-15	postgres	2023-11-15
12	12	15	3:15:00	21:00:00	C29	true	postgres	2023-11-15	postgres	2023-11-15
13	13	10	3:14:00	21:00:00	B10	true	postgres	2023-11-15	postgres	2023-11-15
14	14	12	3:05:00	14:00:00	D21	true	postgres	2023-11-15	postgres	2023-11-15
15	15	16	4:12:00	22:00:00	C04	false	postgres	2023-11-15	postgres	2023-11-15
16	16	26	5:10:00	19:00:00	D09	false	postgres	2023-11-15	postgres	2023-11-15
17	17	25	6:13:00	22:00:00	D16	false	postgres	2023-11-15	postgres	2023-11-15
18	18	32	7:19:00	22:00:00	B19	true	postgres	2023-11-15	postgres	2023-11-15
19	19	27	8:13:00	20:00:00	C05	true	postgres	2023-11-15	postgres	2023-11-15

✓ Successfully run. Total query runtime: 86 msec. 19 rows affected. X

Total rows: 19 of 19 Query complete 00:00:00.086

Ln 114, Col 1

Advanced Queries:

- 13) Find out the visitor who can visit the maximum number of artworks across all exhibitions.
Display visitor_id, visitor name, Number of artworks they could visit and the number of exhibitions that facilitate this.

SELECT

v.first_name || ' ' || v.last_name AS "Visitor Name",

COUNT(DISTINCT(b.exhibition_id)) AS "Number of Exhibitions",

COUNT(art_id) AS "Number of Artworks"

FROM booking b

JOIN display d USING (exhibition_id)

JOIN visitor v USING (visitor_id)

WHERE

slot_start < end_time AND slot_end > start_time

GROUP BY

v.visitor_id, v.first_name || ' ' || v.last_name

ORDER BY COUNT(art_id) DESC

LIMIT 1;

The screenshot shows a SQL query window in SSMS. The query is:

```
120
121
122 --Query 13: Find out the visitor who can visit the maximum number of artworks across all exhibitions. Display visitor_id, visitor name, Number of artworks they could visit
123
124
125     v.first_name || ' ' || v.last_name AS "Visitor Name",
126     COUNT(DISTINCT b.exhibition_id) AS "Number of Exhibitions",
127     COUNT(art_id) AS "Number of Artworks"
128 FROM booking b
129 JOIN display d USING (exhibition_id)
130 JOIN visitor v USING (visitor_id)
131 WHERE
132     slot_start < end_time AND slot_end > start_time
133 GROUP BY v.visitor_id, v.first_name || ' ' || v.last_name
134 ORDER BY COUNT(art_id) DESC
135
136 LIMIT 1;
137
138
```

The results grid shows one row:

Visitor Name	Number of Exhibitions	Number of Artworks
Chris Renner	3	12

Message bar at the bottom: Successfully run. Total query runtime: 43 msec; 1 rows affected.

- 14) Display the Exhibition name, number of visitors for each exhibition and the number of artists that were present to display their work in the exhibitions. Order the exhibitions by total people who were present at some point in the exhibition.

SELECT

name AS "Exhibition Name",

COUNT(DISTINCT(visitor_id)) AS "Visitor Count",

COUNT(DISTINCT(artist_id, exhibition_id)) AS "Present Artist Count",

COUNT(DISTINCT(visitor_id)) + COUNT(DISTINCT(artist_id, exhibition_id)) AS "Total People Count"

```

FROM booking

JOIN exhibition e USING (exhibition_id)

JOIN display USING (exhibition_id)

JOIN artwork USING (art_id)

WHERE artist_presence = TRUE

GROUP BY e.name

ORDER BY COUNT(DISTINCT(visitor_id)) + COUNT(DISTINCT(artist_id, exhibition_id)) DESC;

```

The screenshot shows a PostgreSQL database interface with a query editor and a results table.

Query Editor:

```

--Query 14: Display the Exhibition name, number of visitors for each exhibition and the number of artists that were present to display their work in the exhibitions. Order by total count descending.
136   LIMIT 1;
137
138
139
140
141
142   name AS "Exhibition Name",
143   COUNT(DISTINCT(visitor_id)) AS "Visitor Count",
144   COUNT(DISTINCT(artist_id, exhibition_id)) AS "Present Artist Count",
145   COUNT(DISTINCT(visitor_id)) + COUNT(DISTINCT(artist_id, exhibition_id)) AS "Total People Count"
146
147   FROM booking
148   JOIN exhibition e USING (exhibition_id)
149   JOIN display d USING (exhibition_id)
150   JOIN artwork USING (art_id)
151
152 WHERE artist_presence = TRUE
153 GROUP BY e.name
154 ORDER BY COUNT(DISTINCT(visitor_id)) + COUNT(DISTINCT(artist_id, exhibition_id)) DESC;
155

```

Results Table:

Exhibition Name	Visitor Count	Present Artist Count	Total People Count
An Synergy	6	3	9
ArtistMart	6	2	8
Decora	2	3	5
Vibrant	1	1	2