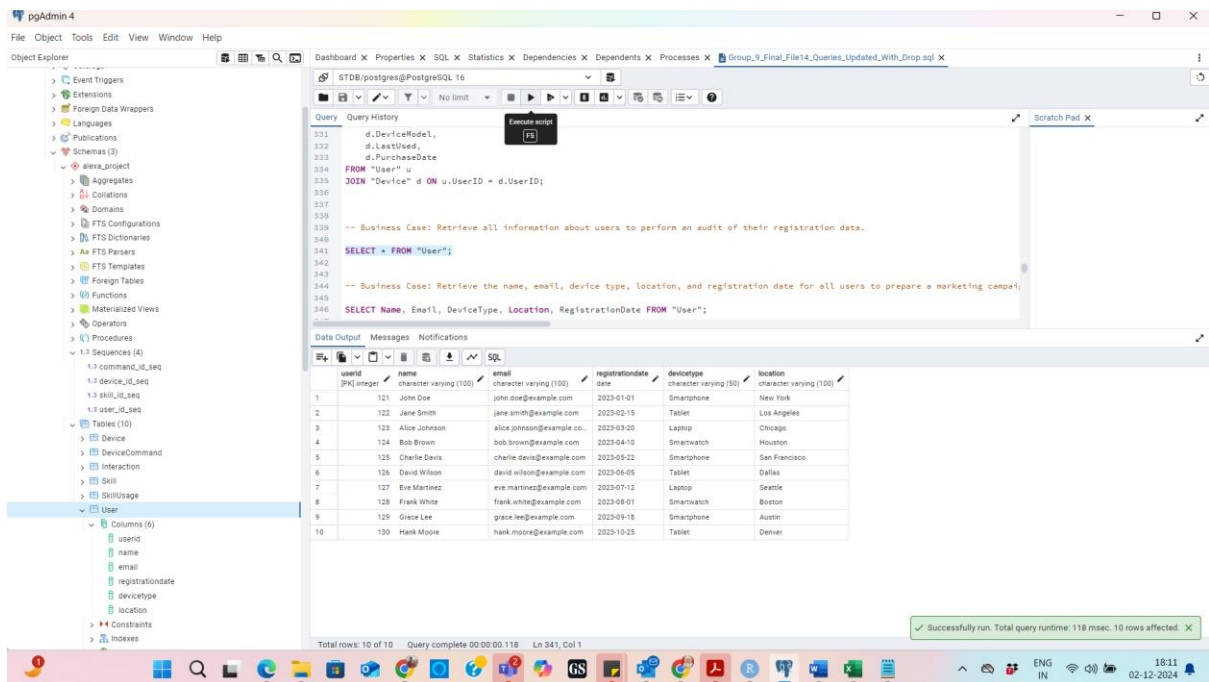


14 SQL Queries (12 Basic, 2 Advanced) Minimum Requirements:

- Query 1: Select all columns and all rows from one table (5 points)

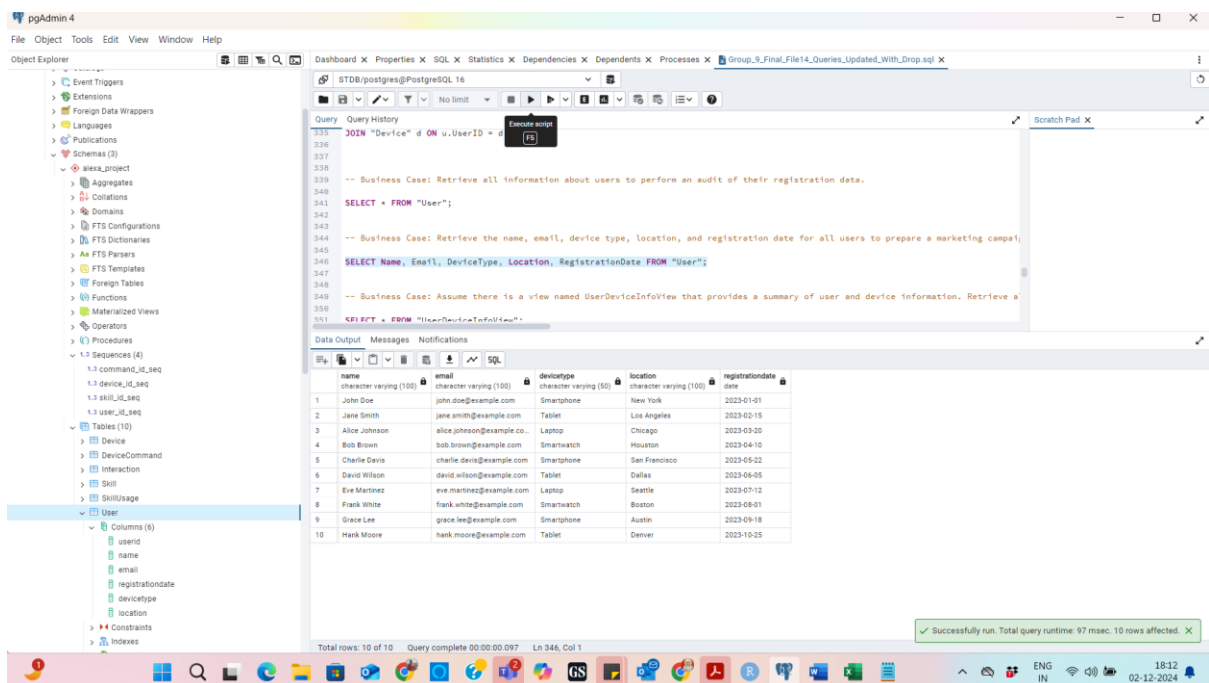


The screenshot shows the pgAdmin 4 interface with the following details:

- Query Editor:** Contains the SQL query: `SELECT * FROM 'User';`
- Data Output:** Displays 10 rows of data from the 'User' table. The columns are: `userid` (integer), `name` (character varying), `email` (character varying), `registrationdate` (date), `devicetype` (character varying), and `location` (character varying).
- Table Structure:** The 'User' table has 6 columns: `userid`, `name`, `email`, `registrationdate`, `devicetype`, and `location`.
- Status:** The query was successfully run, taking 118 msec and affecting 10 rows.

userid	name	email	registrationdate	devicetype	location
1	John Doe	john.doe@example.com	2023-01-01	Smartphone	New York
2	Jane Smith	jane.smith@example.com	2023-02-15	Tablet	Los Angeles
3	Alice Johnson	alice.johnson@example.co.	2023-03-20	Laptop	Chicago
4	Bob Brown	bob.brown@example.com	2023-04-10	Smartwatch	Houston
5	Charlie Davis	charlie.davis@example.com	2023-05-22	Smartphone	San Francisco
6	David Wilson	david.wilson@example.com	2023-06-05	Tablet	Dallas
7	Eve Martinez	eve.martinez@example.com	2023-07-12	Laptop	Seattle
8	Frank White	frank.white@example.com	2023-08-01	Smartwatch	Boston
9	Grace Lee	grace.lee@example.com	2023-09-18	Smartphone	Austin
10	Hank Moore	hank.moore@example.com	2023-10-25	Tablet	Denver

- Query 2: Select five columns and all rows from one table (5 points)



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

- Query Editor:** Contains the SQL query: `SELECT Name, Email, DeviceType, Location, RegistrationDate FROM 'User';`
- Data Output:** Displays 10 rows of data from the 'User' table. The columns are: `name` (character varying), `email` (character varying), `devicetype` (character varying), `location` (character varying), and `registrationdate` (date).
- Table Structure:** The 'User' table has 6 columns: `userid`, `name`, `email`, `registrationdate`, `devicetype`, and `location`.
- Status:** The query was successfully run, taking 97 msec and affecting 10 rows.

name	email	devicetype	location	registrationdate
John Doe	john.doe@example.com	Smartphone	New York	2023-01-01
Jane Smith	jane.smith@example.com	Tablet	Los Angeles	2023-02-15
Alice Johnson	alice.johnson@example.co.	Laptop	Chicago	2023-03-20
Bob Brown	bob.brown@example.com	Smartwatch	Houston	2023-04-10
Charlie Davis	charlie.davis@example.com	Smartphone	San Francisco	2023-05-22
David Wilson	david.wilson@example.com	Tablet	Dallas	2023-06-05
Eve Martinez	eve.martinez@example.com	Laptop	Seattle	2023-07-12
Frank White	frank.white@example.com	Smartwatch	Boston	2023-08-01
Grace Lee	grace.lee@example.com	Smartphone	Austin	2023-09-18
Hank Moore	hank.moore@example.com	Tablet	Denver	2023-10-25

- Query 3: Select all columns from all rows from one view (5 points)

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure, including Schemas, Tables, and Views. The 'User' table is selected. The main pane displays a SQL query in the Query Editor. The query is as follows:

```

SELECT * FROM "User";

-- Business Case: Retrieve the name, email, device type, location, and registration date for all users to prepare a marketing campaign.
SELECT Name, Email, DeviceType, Location, RegistrationDate FROM "User";

-- Business Case: Assume there is a view named UserDeviceInfoView that provides a summary of user and device information. Retrieve a
SELECT * FROM "UserDeviceInfoView";

-- Business Case: Retrieve all user details along with the devices they own.
SELECT *
FROM "User" u
JOIN "Device" d ON u.UserID = d.UserID;

```

The results pane shows the output of the query, displaying a table with 10 rows and 10 columns. The columns are: UserID, Username, Email, DeviceType, Location, RegistrationDate, DeviceID, DeviceName, DeviceModel, and LastUsed. The data is as follows:

UserID	Username	Email	DeviceType	Location	RegistrationDate	DeviceID	DeviceName	DeviceModel	LastUsed
1	John Doe	john.doe@example.com	Smartphone	New York	2023-01-01	345	iPhone 14	Smartphone	2023-12-01 00:00:00
2	Jane Smith	jane.smith@example.com	Tablet	Los Angeles	2023-02-15	346	iPad Pro	Tablet	2023-12-01 00:00:00
3	Alice Johnson	alice.johnson@example.co...	Laptop	Chicago	2023-03-20	347	MacBook Pro	Laptop	2023-12-01 00:00:00
4	Bob Brown	bob.brown@example.com	Smartwatch	Houston	2023-04-10	348	Apple Watch	Smartwatch	2023-12-01 00:00:00
5	Charlie Davis	charlie.davis@example.com	Smartphone	San Francisco	2023-05-22	349	Samsung Galaxy S22	Smartphone	2023-12-01 00:00:00
6	David Wilson	david.wilson@example.com	Tablet	Dallas	2023-06-05	350	Galaxy Tab S8	Tablet	2023-12-01 00:00:00
7	Eve Martinez	eve.martinez@example.com	Laptop	Seattle	2023-07-12	351	Dell XPS 13	Laptop	2023-12-01 00:00:00
8	Frank White	frank.white@example.com	Smartwatch	Boston	2023-08-01	352	Garmin Venu	Smartwatch	2023-12-01 00:00:00
9	Grace Lee	grace.lee@example.com	Smartphone	Austin	2023-09-18	353	Google Pixel 6	Smartphone	2023-12-01 00:00:00
10	Harry Moore	harry.moore@example.com	Tablet	Denver	2023-10-25	354	Samsung Galaxy Tab A8	Tablet	2023-12-01 00:00:00

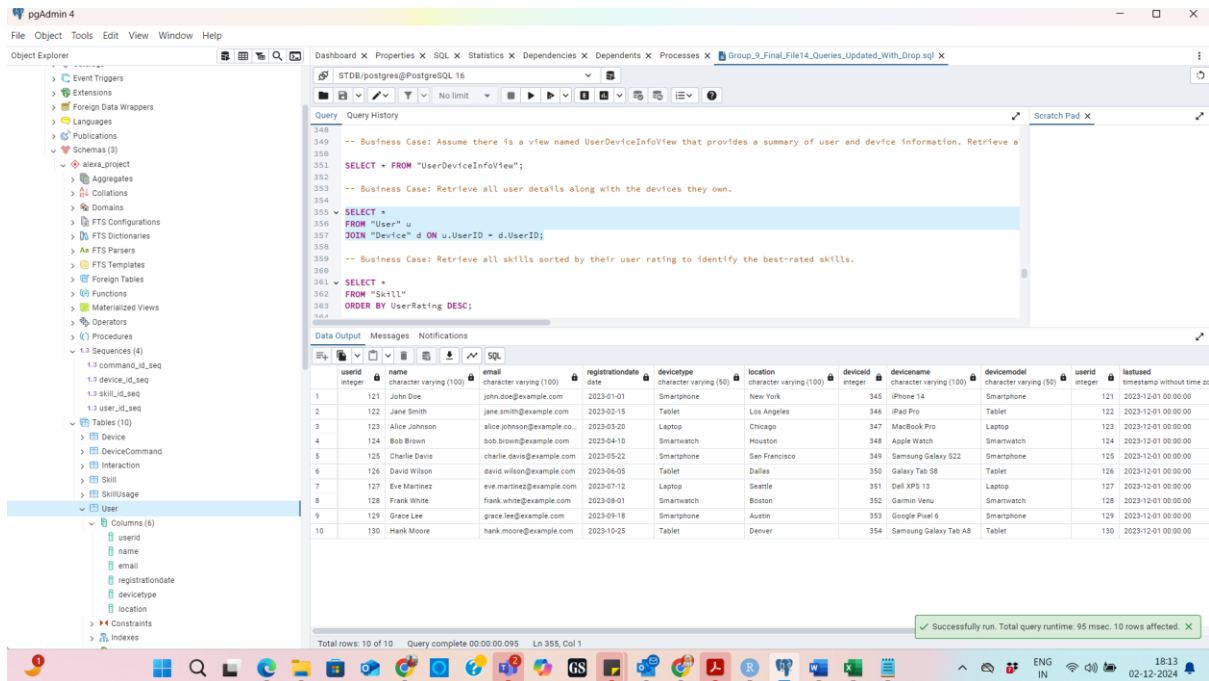
The status bar at the bottom indicates: Total rows: 10 of 10. Query complete 00:00:00.147. Ln 351, Col 1. A green message box at the bottom right says: Successfully run. Total query runtime: 147 msec. 10 rows affected.

```

CREATE VIEW "UserDeviceInfoView" AS
SELECT
    u.UserID,
    u.Name AS UserName,
    u.Email,
    u.DeviceType,
    u.Location,
    u.RegistrationDate,
    d.DeviceID,
    d.DeviceName,
    d.DeviceModel,
    d.LastUsed,
    d.PurchaseDate
FROM "User" u
JOIN "Device" d ON u.UserID = d.UserID;

```

- Query 4: Using a join on 2 tables, select all columns and all rows from the tables without the use of a Cartesian product (5 points)



Query 4: Using a join on 2 tables, select all columns and all rows from the tables without the use of a Cartesian product (5 points)

```

348
349 -- Business Case: Assume there is a view named UserDeviceInfoView that provides a summary of user and device information. Retrieve a
350
351 SELECT * FROM "UserDeviceInfoView";
352
353 -- Business Case: Retrieve all user details along with the devices they own.
354
355 SELECT *
356 FROM "User" u
357 JOIN "Device" d ON u.UserID = d.UserID;
358
359 -- Business Case: Retrieve all skills sorted by their user rating to identify the best-rated skills.
360
361 SELECT *
362 FROM "Skill"
363 ORDER BY UserRating DESC;
364

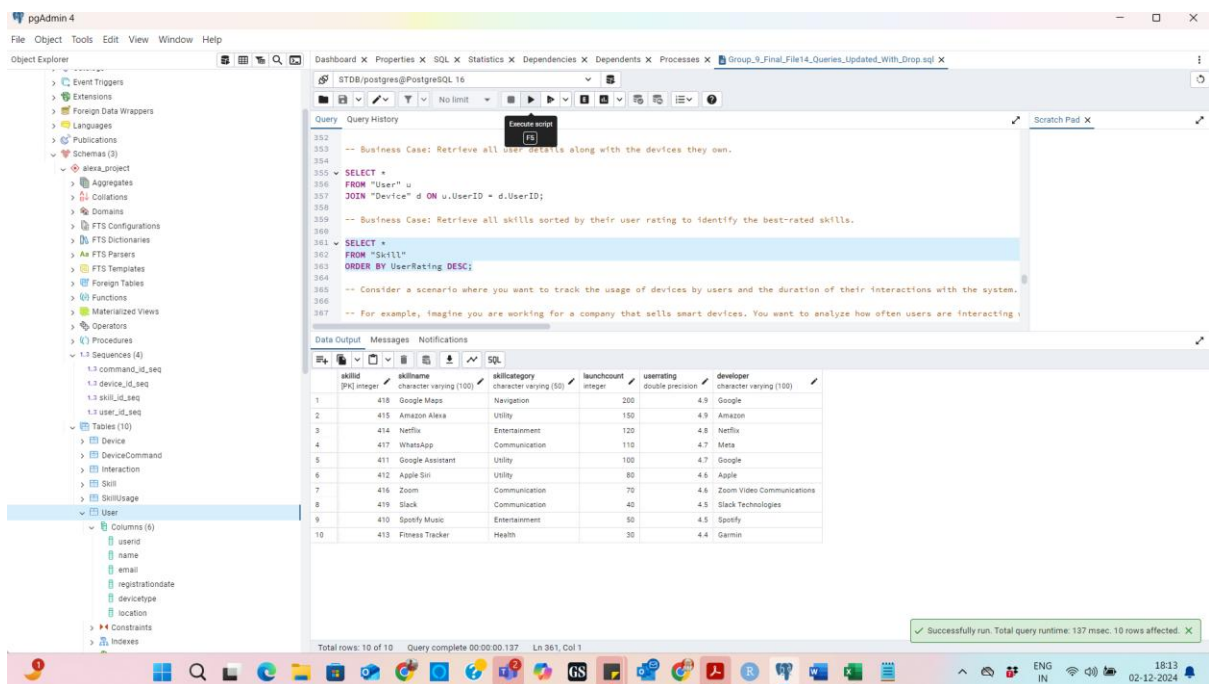
```

userid	name	email	registrationdate	devicetype	location	deviceid	devicename	devicemodel	userlat	lastused
1	John Doe	john.doe@example.com	2023-01-01	Smartphone	New York	345	iPhone 14	Smartphone	121	2023-12-01 00:00:00
2	Jane Smith	jane.smith@example.com	2023-02-15	Tablet	Los Angeles	346	iPad Pro	Tablet	122	2023-12-01 00:00:00
3	Alice Johnson	alice.johnson@example.co.	2023-03-20	Laptop	Chicago	347	MacBook Pro	Laptop	123	2023-12-01 00:00:00
4	Bob Brown	bob.brown@example.com	2023-04-10	Smartwatch	Houston	348	Apple Watch	Smartwatch	124	2023-12-01 00:00:00
5	Charlie Davis	charlie.davis@example.com	2023-05-22	Smartphone	San Francisco	349	Samsung Galaxy S22	Smartphone	125	2023-12-01 00:00:00
6	David Wilson	david.wilson@example.com	2023-06-05	Tablet	Dallas	350	Galaxy Tab S8	Tablet	126	2023-12-01 00:00:00
7	Eve Martinez	eve.martinez@example.com	2023-07-12	Laptop	Seattle	351	Dell XPS 13	Laptop	127	2023-12-01 00:00:00
8	Frank White	frank.white@example.com	2023-08-01	Smartwatch	Boston	352	Garmin Venu	Smartwatch	128	2023-12-01 00:00:00
9	Grace Lee	grace.lee@example.com	2023-09-18	Smartphone	Austin	353	Google Pixel 6	Smartphone	129	2023-12-01 00:00:00
10	Hank Moore	hank.moore@example.com	2023-10-25	Tablet	Denver	354	Samsung Galaxy Tab A8	Tablet	130	2023-12-01 00:00:00

Total rows: 10 of 10 Query complete 00:00:00.095 Ln 355, Col 1

Successfully run. Total query runtime: 95 msec. 10 rows affected.

- Query 5: Select and order data retrieved from one table (5 points)



Query 5: Select and order data retrieved from one table (5 points)

```

362
363 -- Business Case: Retrieve all user details along with the devices they own.
364
365 SELECT *
366 FROM "User" u
367 JOIN "Device" d ON u.UserID = d.UserID;
368
369 -- Business Case: Retrieve all skills sorted by their user rating to identify the best-rated skills.
370
371 SELECT *
372 FROM "Skill"
373 ORDER BY UserRating DESC;
374
375 -- Consider a scenario where you want to track the usage of devices by users and the duration of their interactions with the system.
376
377 -- For example, imagine you are working for a company that sells smart devices. You want to analyze how often users are interacting

```

skillid	skillname	skillcategory	launchcount	userrating	developer
1	Google Maps	Navigation	200	4.9	Google
2	Amazon Alexa	Utility	150	4.9	Amazon
3	Netflix	Entertainment	120	4.8	Netflix
4	WhatsApp	Communication	110	4.7	Meta
5	Google Assistant	Utility	100	4.7	Google
6	Apple Siri	Utility	80	4.6	Apple
7	Zoom	Communication	70	4.6	Zoom Video Communications
8	Slack	Communication	60	4.5	Slack Technologies
9	Spotify Music	Entertainment	50	4.5	Spotify
10	Fitness Tracker	Health	30	4.4	Garmin

Total rows: 10 of 10 Query complete 00:00:00.137 Ln 361, Col 1

Successfully run. Total query runtime: 137 msec. 10 rows affected.

- Query 6: Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 3 rows (5 points)

The screenshot shows the pgAdmin 4 interface with a SQL query executed. The query joins three tables: User, Device, and Interaction. It selects 5 columns: user ID, user name, device name, last used timestamp, and duration. The results are limited to 3 rows.

```

-- run on all
-- ORDER BY UserRating DESC;
-- Consider a scenario where you want to track the usage of devices by users and the duration of their interactions with the system.
-- For example, imagine you are working for a company that sells smart devices. You want to analyze how often users are interacting
SELECT
  u.UserID,
  u.Name AS UserName,
  d.DeviceName,
  d.LastUsed,
  i.Duration
FROM "User" u
JOIN "Device" d ON u.UserID = d.UserID
JOIN "Interaction" i ON u.UserID = i.UserID
LIMIT 3;

```

userid	username	deviceName	lastused	duration
1	John Doe	iPhone 14	2023-12-01 00:00:00	5
2	Jane Smith	iPad Pro	2023-12-01 00:00:00	4
3	Alice Johnson	MacBook Pro	2023-12-01 00:00:00	6

Successfully run. Total query runtime: 77 msec. 3 rows affected.

- Query 7: Select distinct rows using joins on 3 tables (5 points)

The screenshot shows the pgAdmin 4 interface with a SQL query executed. The query joins three tables: User, Device, and VoiceCommand. It selects distinct combinations of user name, device name, and command text. The results are limited to 3 rows.

```

-- In this case, you want to eliminate duplicate combinations of users, devices, and commands to identify unique user-device-com
SELECT DISTINCT
  u.Name AS UserName,
  d.DeviceName,
  vc.CommandText
FROM "User" u
JOIN "Device" d ON u.UserID = d.UserID
JOIN "VoiceCommand" vc ON u.UserID = vc.UserID
LIMIT 3;
-- The company wants to analyze the usage patterns of voice commands across different devices to determine which voice commands are
SELECT
  d.DeviceName,
  vc.CommandText
FROM "Device" d
JOIN "VoiceCommand" vc ON d.DeviceName = vc.DeviceName
LIMIT 3;

```

username	deviceName	commandText
Alice Johnson	MacBook Pro	Open browser
Bob Brown	Apple Watch	Start workout
Charlie Davis	Samsung Galaxy S22	Send message

Successfully run. Total query runtime: 79 msec. 3 rows affected.

- Query 8: Use GROUP BY and HAVING in a select statement using one or more tables (5 points)

Query 8: Use GROUP BY and HAVING in a select statement using one or more tables (5 points)

```

397 FROM "User" u
398 JOIN "Device" d ON u.UserID = d.UserID
399 JOIN "VoiceCommand" vc ON u.UserID = vc.UserID
400 LIMIT 3;
401
-- The company wants to analyze the usage patterns of voice commands across different devices to determine which voice commands are
402
403 SELECT
404     d.DeviceName,
405     vc.CommandText
406 FROM "Device" d
407 JOIN "VoiceCommand" vc ON d.UserID = vc.UserID
408 ORDER BY d.DeviceName, vc.CommandText;
409
-- The company wants to retrieve details of devices that have been used by users who are located in either "New York" or "San Francisco"
410

```

deviceName	commandText
Apple Watch	Start workout
Dell XPS 13	Turn on light
Galaxy Tab S8	Read email
Garmin Venu	Set timer
Google Pixel 6	Make a call
iPad Pro	Set alarm
iPhone 14	Play music
MacBook Pro	Open browser
Samsung Galaxy S22	Send message
Samsung Galaxy Tab A8	Search news

Total rows: 10 of 10 Query complete 00:00:00.097 Ln 394, Col 1

- Query 9: Use IN clause to select data from one or more tables (5 points)

Query 9: Use IN clause to select data from one or more tables (5 points)

```

401
402 -- The company wants to retrieve details of devices that have been used by users who are located in either "New York" or "San Francisco"
403
404 SELECT
405     d.DeviceName,
406     d.DeviceModel,
407     u.Name,
408     u.Location
409 FROM "Device" d
410 JOIN "User" u ON d.UserID = u.UserID
411 WHERE u.Location IN ('New York', 'San Francisco');
412
-- The company wants to analyze the length of device model names to understand how concise or detailed the device naming conventions
413
414 SELECT
415     DeviceName,
416     (LENGTH(DeviceModel) AS DeviceModelLength)
417 FROM "Device" d

```

deviceName	deviceModel	name	location
iPhone 14	Smartphone	John Doe	New York
Samsung Galaxy S22	Smartphone	Charlie Davis	San Francisco

Total rows: 2 of 2 Query complete 00:00:00.071 Ln 404, Col 1

- Query 10: Select length of one column from one table (use LENGTH function) (5 points)

pgAdmin 4

Object Explorer

- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- Schemas (3)
 - alex_project
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences (4)
 - 1.3 command_id_seq
 - 1.3 device_id_seq
 - 1.3 skill_id_seq
 - 1.3 user_id_seq
 - Tables (10)
 - Device
 - DeviceCommand
 - Interaction
 - Skill
 - SkillUsage
 - User
 - Columns (6)
 - userid
 - name
 - email
 - registrationdate
 - devicetype
 - location
 - Constraints
 - Indexes

Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | Processes | Group_9_Final_File14_Queries_Updated_With_Drop.sql

Query Editor: STDB/postgres@PostgreSQL 16

```

408 u.location
409 FROM "Device" d
410 JOIN "User" u ON d.UserID = u.UserID
411 WHERE u.location IN ('New York', 'San Francisco');
412
413 -- The company wants to analyze the length of device model names to understand how concise or detailed the device naming conventions
414
415 SELECT
416     DeviceName,
417     LENGTH(DeviceModel) AS DeviceModelLength
418 FROM "Device";
419
420 -- The company wants to remove an old device from the database. In this case, we will delete a record for a device that has been dis-
421
422 SELECT * FROM "Device" WHERE DeviceName = 'iPhone 14';
423 DELETE FROM "Device" WHERE DeviceName = 'iPhone 14';
424
425 -- The company needs to update the LastUsed date for a device after it is used by the user in a new session. The goal is to keep tra-
426
427 SELECT * FROM "Device" WHERE DeviceName = 'iPad Pro';
428
429 UPDATE "Device"
  
```

Data Output: Messages | Notifications

deviceid	devicename	devicemodel	userid	lastused	purchase date
1	iPhone 14				
2	iPad Pro				
3	MacBook Pro				
4	Apple Watch				
5	Samsung Galaxy S22				
6	Galaxy Tab S8				
7	Dell XPS 13				
8	Garmin Venu				
9	Google Pixel 6				
10	Samsung Galaxy Tab A8				

Successfully run. Total query runtime: 117 msec. 10 rows affected.

- Query 11: Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed (5 points)

pgAdmin 4

Object Explorer

- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- Schemas (3)
 - alex_project
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences (4)
 - 1.3 command_id_seq
 - 1.3 device_id_seq
 - 1.3 skill_id_seq
 - 1.3 user_id_seq
 - Tables (10)
 - Device
 - DeviceCommand
 - Interaction
 - Skill
 - SkillUsage
 - User
 - Columns (6)
 - userid
 - name
 - email
 - registrationdate
 - devicetype
 - location
 - Constraints
 - Indexes

Dashboard | Properties | SQL | Statistics | Dependencies | Dependents | Processes | Group_9_Final_File14_Queries_Updated_With_Drop.sql

Query Editor: STDB/postgres@PostgreSQL 16

```

415 SELECT
416     DeviceName,
417     LENGTH(DeviceModel) AS DeviceModelLength
418 FROM "Device";
419
420 -- The company wants to remove an old device from the database. In this case, we will delete a record for a device that has been dis-
421
422 SELECT * FROM "Device" WHERE DeviceName = 'iPhone 14';
423 DELETE FROM "Device" WHERE DeviceName = 'iPhone 14';
424
425 -- The company needs to update the LastUsed date for a device after it is used by the user in a new session. The goal is to keep tra-
426
427 SELECT * FROM "Device" WHERE DeviceName = 'iPad Pro';
428
429 UPDATE "Device"
  
```

Data Output: Messages | Notifications

deviceid	devicename	devicemodel	userid	lastused	purchase date
----------	------------	-------------	--------	----------	---------------

Successfully run. Total query runtime: 79 msec. 0 rows affected.

- Query 12: Update one record from one table. Use select statements to demonstrate the table contents before and after the UPDATE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed (5 points)

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Dashboard Properties SQL Statistics Dependencies Dependents Processes Group_9_Final_File14_Queries_Updated_WiTh_Drop.sql

Query Query History

```

421
422 SELECT * FROM "Device" WHERE DeviceName = 'iPhone 14';
423 DELETE FROM "Device" WHERE DeviceName = 'iPhone 14';
424 SELECT * FROM "Device" WHERE DeviceName = 'iPhone 14';
425
426 -- The company needs to update the LastUsed date for a device after it is used by the user in a new session. The goal is to keep tra
427
428 SELECT * FROM "Device" WHERE DeviceName = 'iPad Pro';
429
430 UPDATE "Device"
431 SET LastUsed = '2023-12-02'
432 WHERE DeviceName = 'iPad Pro';
433
434 SELECT * FROM "Device" WHERE DeviceName = 'iPad Pro';
435
436 -- The company wants to find out which device models are the most popular in each city. This will help them tailor marketing and inv
437

```

Data Output Messages Notifications

deviceid	deviceName	deviceModel	userid	lastUsed	purchasedate
346	iPad Pro	Tablet	122	2023-12-01 00:00:00	2023-02-15

Total rows: 1 of 1 Query complete 00:00:00.118 Ln 426, Col 1

Successfully run. Total query runtime: 118 msec. 1 rows affected.

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Dashboard Properties SQL Statistics Dependencies Dependents Processes Group_9_Final_File14_Queries_Updated_WiTh_Drop.sql

Query Query History

```

421
422 SELECT * FROM "Device" WHERE DeviceName = 'iPhone 14';
423 DELETE FROM "Device" WHERE DeviceName = 'iPhone 14';
424 SELECT * FROM "Device" WHERE DeviceName = 'iPhone 14';
425
426 -- The company needs to update the LastUsed date for a device after it is used by the user in a new session. The goal is to keep tra
427
428 SELECT * FROM "Device" WHERE DeviceName = 'iPad Pro';
429
430 UPDATE "Device"
431 SET LastUsed = '2026-12-02'
432 WHERE DeviceName = 'iPad Pro';
433
434 SELECT * FROM "Device" WHERE DeviceName = 'iPad Pro';
435
436 -- The company wants to find out which device models are the most popular in each city. This will help them tailor marketing and inv
437

```

Data Output Messages Notifications

deviceid	deviceName	deviceModel	userid	lastUsed	purchasedate
346	iPad Pro	Tablet	122	2026-12-02 00:00:00	2023-02-15

Total rows: 1 of 1 Query complete 00:00:00.127 Ln 434, Col 1

Successfully run. Total query runtime: 127 msec. 1 rows affected.

Perform 2 Additional Advanced Queries (40 points)

1)

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- Schemas (3)
 - alex_project
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences (4)
 - command_id_seq
 - device_id_seq
 - skill_id_seq
 - user_id_seq
 - Tables (10)
 - Device
 - DeviceCommand
 - Interaction
 - Skill
 - SkillUsage
 - User
 - Columns (6)
 - userid
 - name
 - email
 - registrationdate
 - location
 - devicetype
 - Constraints
 - Indexes

Dashboard Properties SQL Statistics Dependencies Dependents Processes Group_9_Final_File14_Queries_Updated_With_Drop.sql

STDB/postgres@PostgreSQL 16

Query Query History

```

433
434 SELECT * FROM "Device" WHERE DeviceName = 'iPad Pro';
435
436 -- The company wants to find out which device models are the most popular in each city. This will help them tailor marketing and inv
437
438 SELECT
439   u.Location,
440   d.DeviceModel,
441   COUNT(d.DeviceID) AS DeviceCount
442 FROM "Device" d
443 JOIN "User" u ON d.UserID = u.UserID
444 GROUP BY u.Location, d.DeviceModel
445 ORDER BY u.Location, DeviceCount DESC;
446
447 -- Advanced Query 2: List all devices used by users from a specific location (e.g., 'New York') who have used a voice command
448
  
```

Data Output Messages Notifications

location	deviceModel	devicecount
1 Austin	Smartphone	1
2 Boston	Smartwatch	1
3 Chicago	Laptop	1
4 Dallas	Tablet	1
5 Denver	Tablet	1
6 Houston	Smartwatch	1
7 Los Angeles	Tablet	1
8 San Francisco	Smartphone	1
9 Seattle	Laptop	1

Total rows: 9 of 9 Query complete 00:00:00.077 Ln 438, Col 1

Successfully run. Total query runtime: 77 msec. 9 rows affected.

2)

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- Schemas (3)
 - alex_project
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences (4)
 - command_id_seq
 - device_id_seq
 - skill_id_seq
 - user_id_seq
 - Tables (10)
 - Device
 - DeviceCommand
 - Interaction
 - Skill
 - SkillUsage
 - User
 - Columns (6)
 - userid
 - name
 - email
 - registrationdate
 - location
 - devicetype
 - Constraints
 - Indexes

Dashboard Properties SQL Statistics Dependencies Dependents Processes Group_9_Final_File14_Queries_Updated_With_Drop.sql

STDB/postgres@PostgreSQL 16

Query Query History

```

440 d.DeviceModel,
441 COUNT(d.DeviceID) AS DeviceCount
442 FROM "Device" d
443 JOIN "User" u ON d.UserID = u.UserID
444 GROUP BY u.Location, d.DeviceModel
445 ORDER BY u.Location, DeviceCount DESC;
446
447 -- Advanced Query 2: List all users who have used a specific voice command ("Send message")
448
449 SELECT DISTINCT u.Name, u.Email
450 FROM "User" u
451 JOIN "VoiceCommand" vc ON u.UserID = vc.UserID
452 WHERE vc.CommandText = 'Send message';
453
454
455
  
```

Data Output Messages Notifications

name	email
1 Charlie Davis	charlie.davis@example.com

Total rows: 1 of 1 Query complete 00:00:00.082 Ln 448, Col 1

Successfully run. Total query runtime: 82 msec. 1 rows affected.