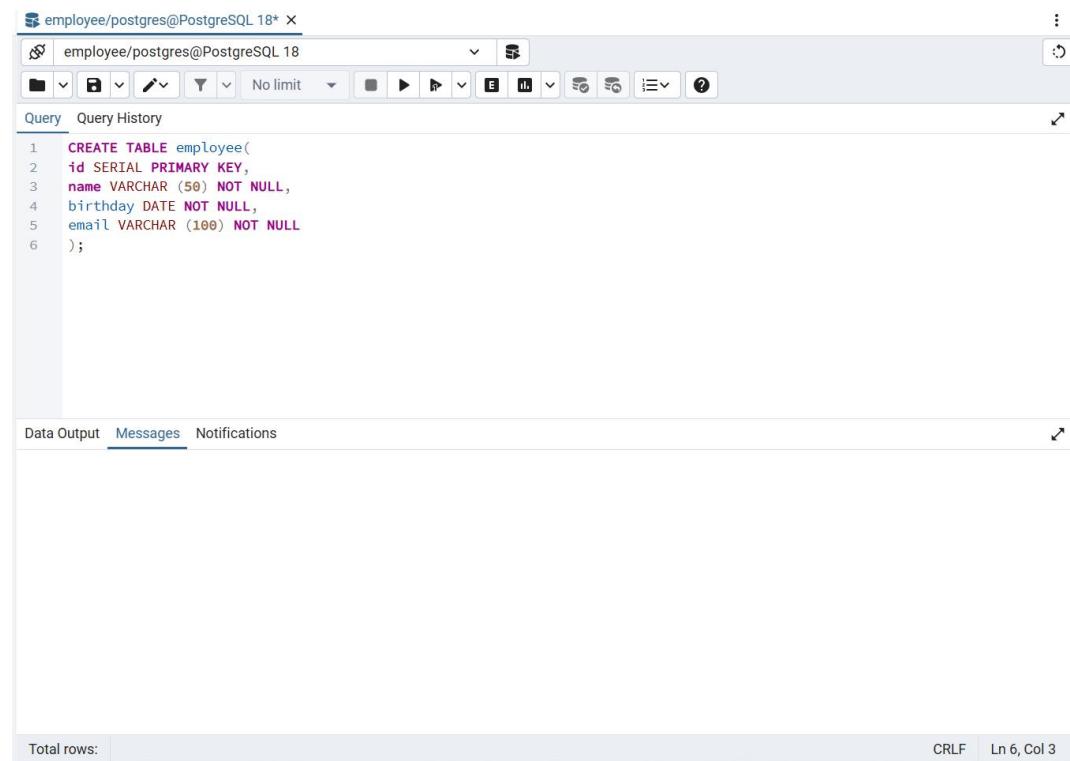


Assignment 8: DVDRental Queries

This assignment focuses on practicing SQL operations on a test database. The main tasks include creating an employee table with basic column information, inserting sample data generated using the Mockaroo service, updating records based on different columns, and deleting specific rows from the table. These exercises help improve understanding of **CREATE, INSERT, UPDATE, DELETE, and working with database records.**

Query 1:

Description: The query creates a table named **employee** with columns for ID, name, birthday, and email. It helps practice how to define a table structure using SQL.



The screenshot shows a PostgreSQL client window with the following details:

- Connection:** employee/postgres@PostgreSQL 18*
- Toolbar:** Includes icons for connection, refresh, search, and various database management functions.
- Query Editor:** Labeled "Query History". It contains the following SQL code:

```
1 CREATE TABLE employee(
2     id SERIAL PRIMARY KEY,
3     name VARCHAR (50) NOT NULL,
4     birthday DATE NOT NULL,
5     email VARCHAR (100) NOT NULL
6 );
```
- Status Bar:** Shows "Data Output" is selected, along with "Messages" and "Notifications".
- Bottom Status:** "Total rows:" (empty), "CRLF", and "Ln 6, Col 3".

Viewing Table Data

The screenshot shows the pgAdmin 4 interface. At the top, there's a connection bar with 'employee/postgres@PostgreSQL 18*' and a toolbar with various icons. Below that is a 'Query' tab with a history section and a code editor containing the following SQL:

```
1 SELECT * FROM employee;
```

Below the code editor is a results pane with a header row showing column names: 'id [PK] integer', 'name character varying (50)', 'birthday date', and 'email character varying (100)'. The results area is currently empty. At the bottom of the results pane, a green message box says 'Successfully run. Total query runtime: 136 msec. 0 rows affected.' To the left of the message, it says 'Total rows: 0' and 'Query complete 00:00:00.136'. To the right, it says 'CRLF' and 'Ln 1, Col 14'.

Inserting Sample Data with Mockaroo

I used the Mockaroo service to generate 50 rows of sample data for the `employee` table. The fields include realistic names, birthdates, and email addresses. This helped test the table with real looking records and practice data insertion in SQL.

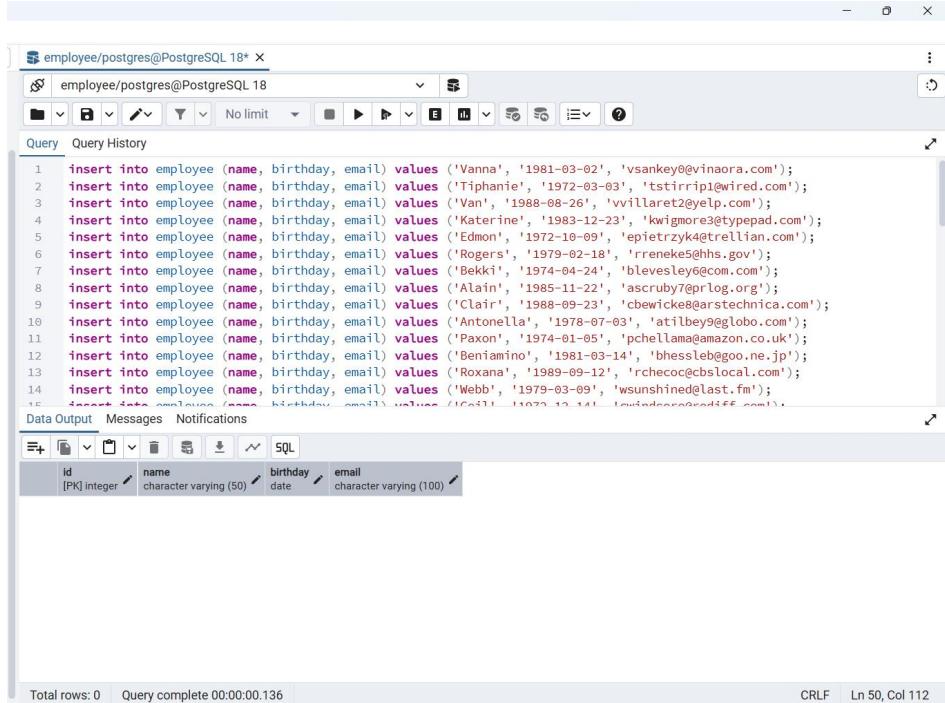
The screenshot shows the Mockaroo web application. At the top, there's a navigation bar with 'SCHEMAS', 'DATASETS', 'APIS', 'PROJECTS', 'SCENARIOS', 'FUNCTIONS', 'DATABASES', and 'DE-IDENTIFY'. Below the navigation is a message: 'Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and Excel formats.' Another message below it says 'Need more data? Plans start at just \$60/year. Mockaroo is also available as a [docker image](#) that you can deploy in your own private cloud.' The main area is a form for generating fields:

Field Name	Type	Options
name	First Name	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>
birthday	Datetime	02/01/1970 <input type="button" value="Calendar"/> to 02/01/1990 <input type="button" value="Calendar"/> format: yyyy-mm-dd <input type="button" value="blank: 0 %"/> <input type="button" value="Σ"/>
email	Email Address	blank: 0 % <input type="button" value="Σ"/> <input type="button" value="X"/>

Buttons at the bottom include '+ ADD ANOTHER FIELD' and 'GENERATE FIELDS USING AI...'. At the very bottom, there are input fields for '# Rows:' (set to 50), 'Format:' (set to SQL), 'Table Name:' (set to 'employee'), and a checked checkbox for 'include CREATE TABLE'.

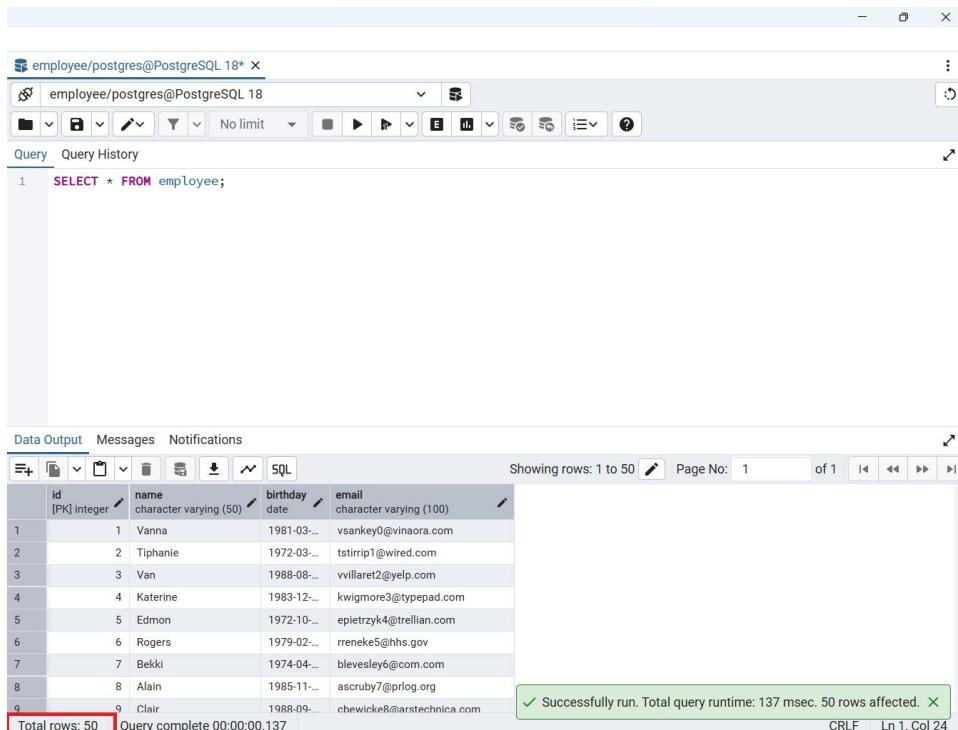
Inserting Records into the Employee Table

I generated 50 sample records using the Mockaroo service and inserted them into the employee table. The dataset includes realistic names, birthdays, and email addresses. This step allowed me to test the table with real looking data instead of adding the values manually.



```
employee/postgres@PostgreSQL 18* 
employee/postgres@PostgreSQL 18
No limit
Query History
Query
Data Output Messages Notifications
Total rows: 0 Query complete 00:00:00.136 CRLF Ln 50, Col 112
SQL
1 insert into employee (name, birthday, email) values ('Vanna', '1981-03-02', 'vsankey0@vinaora.com');
2 insert into employee (name, birthday, email) values ('Tiphanie', '1972-03-03', 'tstirrip1@wired.com');
3 insert into employee (name, birthday, email) values ('Van', '1988-08-26', 'vvillaret2@yelp.com');
4 insert into employee (name, birthday, email) values ('Katerine', '1983-12-23', 'kwigmore3@typepad.com');
5 insert into employee (name, birthday, email) values ('Edmon', '1972-10-09', 'epietrzyk4@trellian.com');
6 insert into employee (name, birthday, email) values ('Rogers', '1979-02-18', 'rreneke5@hhs.gov');
7 insert into employee (name, birthday, email) values ('Bekki', '1974-04-24', 'blevesley6@com.com');
8 insert into employee (name, birthday, email) values ('Alain', '1985-11-22', 'ascrubry7@prlog.org');
9 insert into employee (name, birthday, email) values ('Clair', '1988-09-23', 'cbewicke8@arstechnica.com');
10 insert into employee (name, birthday, email) values ('Antonella', '1978-07-03', 'atilbey9@globo.com');
11 insert into employee (name, birthday, email) values ('Paxon', '1974-01-05', 'pchellama@amazon.co.uk');
12 insert into employee (name, birthday, email) values ('Beniamino', '1981-03-14', 'bhessleb@oo.ne.jp');
13 insert into employee (name, birthday, email) values ('Roxana', '1989-09-12', 'rcheccoc@bslocal.com');
14 insert into employee (name, birthday, email) values ('Webb', '1979-03-09', 'wsunshined@last.fm');
15
Data Output Messages Notifications
```

Verifying Inserted Data



```
employee/postgres@PostgreSQL 18* 
employee/postgres@PostgreSQL 18
No limit
Query History
Query
Data Output Messages Notifications
Showing rows: 1 to 50 Page No: 1 of 1
Total rows: 50 Query complete 00:00:00.137 CRLF Ln 1, Col 24
SQL
1 SELECT * FROM employee;
```

id	name	birthday	email
1	Vanna	1981-03-02	vsankey0@vinaora.com
2	Tiphanie	1972-03-03	tstirrip1@wired.com
3	Van	1988-08-26	vvillaret2@yelp.com
4	Katerine	1983-12-23	kwigmore3@typepad.com
5	Edmon	1972-10-09	epietrzyk4@trellian.com
6	Rogers	1979-02-18	rreneke5@hhs.gov
7	Bekki	1974-04-24	blevesley6@com.com
8	Alain	1985-11-22	ascrubry7@prlog.org
9	Clair	1988-09-23	cbewicke8@arstechnica.com

✓ Successfully run. Total query runtime: 137 msec. 50 rows affected. X

Query 2:

Description: This query updates the name, birthday, and email of the employee whose ID is 1. It helps practice how to modify specific records in a table using the UPDATE statement and a condition based on the primary key.

The screenshot shows the pgAdmin interface with two tabs: 'employee/postgres@PostgreSQL 18*' and 'employee/postgres@PostgreSQL 18'. The top tab contains the SQL query:

```
1 UPDATE employee
2 SET name = 'Mesut',
3     birthday = '1990-05-05',
4     email = 'mst@e.com'
5 WHERE id = 1;
```

The bottom tab displays the 'Data Output' results table:

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
1	1	Vanna	1981-03-27	vsankey0@vinaora.com
2	2	Tiphanie	1972-03-11	tstirrip1@wired.com
3	3	Van	1988-08-28	vvillaret2@yelp.com
4	4	Katerine	1983-12-10	kwigmore3@typepad.com
5	5	Edmon	1972-10-22	epletzky4@trellian.com
6	6	Rogers	1979-02-19	irneke5@hhs.gov
7	7	Bekki	1974-04-12	blesley6@com.com
8	8	Alain	1985-11-15	ascruby7@prolog.org
9	9	Clair	1988-09-21	cbewicke8@arstechnica.com

Total rows: 50 Query complete 00:00:00.137

Verifying Update Results:

The screenshot shows the pgAdmin interface with two tabs: 'employee/postgres@PostgreSQL 18*' and 'employee/postgres@PostgreSQL 18'. The top tab contains the SQL query:

```
1 SELECT * FROM employee;
```

The bottom tab displays the 'Data Output' results table:

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
42	43	Lizzie	1965-04-26	mlucyke10@qiniu.com
43	44	Kirsten	1970-12-10	kpowderham17@hatena.ne.jp
44	45	Veronique	1977-01-27	vmilan18@tiny.cc
45	46	Kasper	1987-04-10	kcleator19@oaic.gov.au
46	47	Orelle	1988-03-12	odoblin1a@alibaba.com
47	48	Kristopher	1971-01-18	kdensey1b@ft.com
48	49	Torre	1982-11-05	tfell1c@lulu.com
49	50	Bradly	1980-05-22	bmackstead1d@colog-nifty.co...
50	1	Mesut	1990-05-05	mst@e.com

Total rows: 50 Query complete 00:00:00.114

Query 3:

Description: This query changes the name of the employee from "Tiphanie" to "Jack". It helps practice how to update a specific record using a condition based on the name column instead of the primary key.

The screenshot shows the pgAdmin interface with a query editor and a data output viewer. The query editor contains the following SQL code:

```
1 UPDATE employee
2 SET name = 'Jack'
3 WHERE name = 'Tiphanie';
```

The data output viewer shows a table of employees with 50 rows. The row where the name was updated is highlighted in red. The table has columns: id [PK] integer, name character varying (50), birthday date, and email character varying (100). The highlighted row is for employee id 2, name 'Jack', birthday 1972-03-21, and email 'tstirrip1@wired.com'.

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
1	2	Tiphanie	1972-03-21	tstirrip1@wired.com
2	3	Van	1988-08-26	villaret2@yelp.com
3	4	Katerine	1983-12-09	kwigmore3@typepad.com
4	5	Edmon	1972-10-13	epletzky4@trellian.com
5	6	Rogers	1979-02-28	rreneke5@hhs.gov
6	7	Bekki	1974-04-15	blevesley6@com.com
7	8	Alain	1985-11-27	ascruby7@prlog.org
8	9	Clair	1988-09-29	cbewickie8@rstchnica.com
9	10	Antonella	1978-07-14	atilbev9@olobo.com

Verifying Update Results:

The screenshot shows the pgAdmin interface with a query editor and a data output viewer. The query editor contains the following SQL code:

```
1 Select * FROM employee;
```

The data output viewer shows a table of employees with 50 rows. The row where the name was updated is highlighted in red. The table has columns: id [PK] integer, name character varying (50), birthday date, and email character varying (100). The highlighted row is for employee id 2, name 'Jack', birthday 1972-03-21, and email 'tstirrip1@wired.com'.

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
42	44	Kristoff	1980-12-20	kpunderland111@outlook.ne.jp
43	45	Veronique	1977-01-15	vmlan18@tiny.cc
44	46	Kasper	1987-04-12	kcleator19@oaic.gov.au
45	47	Orelle	1988-03-28	odoblin1a@alibaba.com
46	48	Kristopher	1971-01-01	kdensey1b@ft.com
47	49	Torre	1982-11-11	tfell1c@lulu.com
48	50	Bradly	1980-05-20	bmackeastead1d@colog-nifty.co...
49	1	Mesut	1990-05-10	mst@e.com
50	2	Jack	1972-03-21	tstirrip1@wired.com

Query 4:

Description: This query changes the birthday of the employee whose original birthday is '1988-08-26'. It helps practice how to update records using a condition based on the birthday column instead of ID or name.

The screenshot shows the pgAdmin interface with a query editor and a data output viewer. The query editor contains the following SQL code:

```
1 UPDATE employee
2 SET birthday = '1990-02-02'
3 WHERE birthday = '1988-08-26';
```

The data output viewer shows the 'employee' table with 11 rows. The row where the birthday was updated is highlighted in yellow. The table structure is:

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
1	3	Van	1988-08-26	vvillaret2@yelp.com
2	4	Katerine	1983-12-21	kwigmore3@typepad.com
3	5	Edmon	1972-10-10	eplietrzyk4@trellian.com
4	6	Rogers	1979-02-17	rreneke5@hhs.gov
5	7	Bekki	1974-04-15	blevesley6@com.com
6	8	Alain	1985-11-28	ascrubry7@prlog.org
7	9	Clair	1988-09-11	cbewicke8@arstechnica.com
8	10	Antonella	1978-07-29	atilbey9@jibo.com
9	11	Paxton	1974-01-22	ochellama@amazon.co.uk
Total rows: 50	Query complete 00:00:00.092	CRLF	Ln 3, Col 31	

Verifying Update Results:

The screenshot shows the pgAdmin interface with a query editor and a data output viewer. The query editor contains the following SQL code:

```
1 SELECT * FROM employee;
```

The data output viewer shows the 'employee' table with 50 rows. The row where the birthday was updated is highlighted in red. The table structure is:

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
42	43	veronique	1977-01-15	vrillian1@ury.cc
43	46	Kasper	1987-04-10	kcleator19@oaic.gov.au
44	47	Orelle	1988-03-12	odoblin1a@alibaba.com
45	48	Kristopher	1971-01-10	kdensey1b@ft.com
46	49	Torre	1982-11-15	tfell1c@lulu.com
47	50	Brady	1980-05-10	bmackstead1d@cocolog-nifty.co...
48	1	Mesut	1990-05-10	mst@e.com
49	2	Jack	1972-03-10	tstirrip1@wired.com
50	3	Van	1990-02-02	vvillaret2@yelp.com
Total rows: 50	Query complete 00:00:00.116	CRLF	Ln 1, Col 24	

Query 5:

Description: This query replaces the email of an employee whose current address is 'kwigmore3@typepad.com'.

The screenshot shows the pgAdmin 4 interface with a query editor and a data output viewer. The query editor contains the following SQL code:

```
1 UPDATE employee
2 SET email = 'kate@irina.com'
3 WHERE email = 'kwigmore3@typepad.com';
```

The data output viewer shows a table with 12 rows of employee data. The row where the update was performed (id 4, name Katerine) has its 'email' column value ('kwigmore3@typepad.com') highlighted in blue, indicating it is selected or the result of the previous query.

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
1	4	Katerine	1983-12-...	kwigmore3@typepad.com
2	5	Edmon	1972-10-...	epletrzyk4@trellian.com
3	6	Rogers	1979-02-...	rreneke5@hhs.gov
4	7	Bekki	1974-04-...	blevesley6@com.com
5	8	Alain	1985-11-...	ascraby7@prlog.org
6	9	Clair	1988-09-...	cbewicke8@arstechnica.com
7	10	Antonella	1978-07-...	atilbey9@globo.com
8	11	Paxton	1974-01-...	pchellama@amazon.co.uk
9	12	Reniamino	1981-03-...	bhessleb@noon.ne.in
Total rows: 50 Query complete 00:00:00.116				

Verifying Update Results:

The screenshot shows the pgAdmin 4 interface with a query editor and a data output viewer. The query editor contains the following SQL code:

```
1 SELECT * FROM employee;
```

The data output viewer shows the same table of employee data. The row where the update was performed (id 4, name Katerine) now has its 'email' column value ('kate@irina.com') highlighted in red, indicating it is selected or the result of the current query.

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
42	40	Kasper	1988-04-...	kasperator19@oarc.gov.au
43	47	Orelle	1988-03-...	odeblin1a@alibaba.com
44	48	Kristopher	1971-01-...	kdensey1b@ft.com
45	49	Torre	1982-11-...	tfell1c@lulu.com
46	50	Bradly	1980-05-...	bmackstead1d@cocolog-nifty.co...
47	1	Mesut	1990-05-...	mst@e.com
48	2	Jack	1972-03-...	tstirrip1@wired.com
49	3	Van	1990-02-...	vvillaret2@yelp.com
50	4	Katerine	1983-12-...	kate@irina.com
Total rows: 50 Query complete 00:00:00.086				

Query 6:

Description: This query updates both the name and birthday of the employee with ID 11

The screenshot shows the pgAdmin 4 interface. The top bar displays the connection information: employee/postgres@PostgreSQL 18*. Below the toolbar, the Query tab is selected, showing the following SQL code:

```
1 UPDATE employee
2 SET name = 'Alan',
3     birthday = '2000-06-08'
4 WHERE id = '11';
```

Under the Data Output tab, the results of the query are displayed:

```
UPDATE 1
Query returned successfully in 56 msec.
```

At the bottom, status information includes "Total rows: 1" and "Query complete 00:00:00.056".

Query 7: Deleting a Record by ID

Description: This query removes the employee whose ID is 11 from the table. It's a simple way to test how deletion works using a condition based on the primary key.

The screenshot shows the pgAdmin 4 interface. The top bar displays the connection information: employee/postgres@PostgreSQL 18*. Below the toolbar, the Query tab is selected, showing the following SQL code:

```
1 DELETE FROM employee
2 WHERE id = '11';
3
```

Under the Data Output tab, the results of the query are displayed:

```
Total rows: 1
```

Below the results, the employee table is shown in a grid format. The row where ID 11 was located is highlighted with a red box. The table has columns: id, name, birthday, and email. The original data for ID 11 was: id 11, name Alan, birthday 2000-06-08, and email pchellama@amazon.co.uk.

Query 8:

Description: This query deletes the employee named 'Katerine' from the table. It shows how to remove a specific row using a condition based on the name column instead of ID.

The screenshot shows the pgAdmin interface with a query editor and a data output viewer. The query editor contains the following SQL code:

```
1 DELETE FROM employee
2 WHERE name = 'Katerine';
```

The data output viewer shows a table with columns: id, name, birthday, and email. The row for 'Katerine' is highlighted with a red border. The table data is as follows:

	id	name	birthday	email
40	40	Kasper	1987-04-15	kcleator19@oaic.gov.au
41	41	Orelle	1988-03-15	odoblin1a@alibaba.com
42	42	Kristopher	1971-01-15	kdensey1b@ft.com
43	43	Torre	1982-11-15	tfell1c@lulu.com
44	44	Bradly	1980-05-15	bmackstead1d@cocolog-nifty.co...
45	45	Mesut	1990-05-15	mst@e.com
46	46	Jack	1972-03-15	tstirip1@wired.com
47	47	Van	1990-02-15	vvillaret2@yelp.com
48	48	Katerine	1983-12-15	kate@rina.com
49	49			

Total rows: 49 Query complete 00:00:00.097 CRLF Ln 2, Col 24

Query 9:

Description: This query removes the employee whose birthday is '1990-02-02'.

The screenshot shows the pgAdmin interface with a query editor and a data output viewer. The query editor contains the following SQL code:

```
1 DELETE FROM employee
2 WHERE birthday = '1990-02-02';
```

The data output viewer shows a table with columns: id, name, birthday, and email. The row for the employee with birthday '1990-02-02' is highlighted with a red border. The table data is as follows:

	id	name	birthday	email
40	40	Veronique	1977-01-15	vhilari18@ury.cc
41	41	Kasper	1987-04-15	kcleator19@oaic.gov.au
42	42	Orelle	1988-03-15	odoblin1a@alibaba.com
43	43	Kristopher	1971-01-15	kdensey1b@ft.com
44	44	Torre	1982-11-15	tfell1c@lulu.com
45	45	Bradly	1980-05-15	bmackstead1d@cocolog-nifty.co...
46	46	Mesut	1990-05-15	mst@e.com
47	47	Jack	1972-03-15	tstirip1@wired.com
48	48	Van	1990-02-15	vvillaret2@yelp.com
49	49			

Total rows: 48 Query complete 00:00:00.098 CRLF Ln 2, Col 32

Query 10:

Description: This query removes the employee whose ID is 2 and whose name is 'Jack'. I was using both conditions together.

The screenshot shows the pgAdmin interface with a query editor and a data viewer. The query editor contains the following SQL code:

```
1 DELETE FROM employee
2 WHERE id = '2' AND NAME = 'Jack';
```

The data viewer shows a table with 47 rows. The row for employee ID 2, name Jack, is highlighted with a red border. The table has columns: id, name, birthday, and email. The data is as follows:

id	name	birthday	email
38	Kirsten	1970-12-...	kpowderham17@hatena.ne.jp
40	Veronique	1977-01-...	vmlan18@tiny.cc
41	Kasper	1987-04-...	kcleator19@oaic.gov.au
42	Orelle	1988-03-...	odoblin1a@alibaba.com
43	Kristopher	1971-01-...	kdensey1b@ft.com
44	Torre	1982-11-...	tfell1c@lulu.com
45	Bradly	1980-05-...	bmackstead1d@colog-nifty.co...
46	Mesut	1990-05-...	mst@e.com
47	Jack	1972-03-...	tstirip1@wired.com

Total rows: 47 Query complete 00:00:00.118 CRLF Ln 2, Col 34

Query 11: Deleting a Record by Email

Description: This query deletes the employee whose email is 'mst@e.com'. If you want to remove a specific user based on their contact information rather than ID or name.

The screenshot shows the pgAdmin interface with a query editor and a data viewer. The query editor contains the following SQL code:

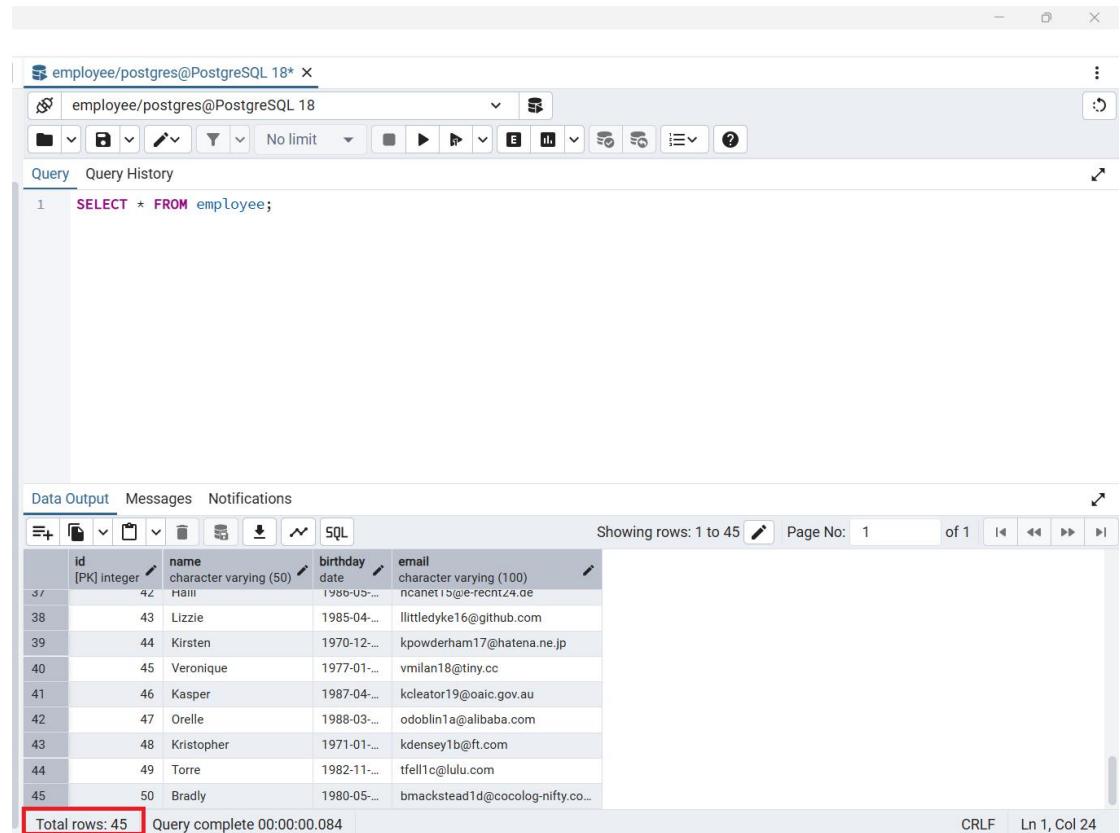
```
1 DELETE FROM employee
2 WHERE email = 'mst@e.com';
```

The data viewer shows a table with 46 rows. The row for employee email mst@e.com is highlighted with a red border. The table has columns: id, name, birthday, and email. The data is as follows:

id	name	birthday	email
38	Lizzie	1965-04-...	mlueuyke1o@jimdo.com
39	Kirsten	1970-12-...	kpowderham17@hatena.ne.jp
40	Veronique	1977-01-...	vmlan18@tiny.cc
41	Kasper	1987-04-...	kcleator19@oaic.gov.au
42	Orelle	1988-03-...	odoblin1a@alibaba.com
43	Kristopher	1971-01-...	kdensey1b@ft.com
44	Torre	1982-11-...	tfell1c@lulu.com
45	Bradly	1980-05-...	bmackstead1d@colog-nifty.co...
46	Mesut	1990-05-...	mst@e.com

Total rows: 46 Query complete 00:00:00.073 CRLF Ln 2, Col 27

Verifying Update Results:



The screenshot shows the pgAdmin 4 interface with a query editor and a results grid.

Query Editor (Top):

```
employee/postgres@PostgreSQL 18* 
employee/postgres@PostgreSQL 18
Query History
SELECT * FROM employee;
```

Results Grid (Bottom):

	id	name	birthday	email
37	42	Hallie	1986-05-10	ncaneti1b@e-recht24.de
38	43	Lizzie	1985-04-15	littledyke16@github.com
39	44	Kirsten	1970-12-10	kpowderham17@hatena.ne.jp
40	45	Veronique	1977-01-05	vmilan18@tiny.cc
41	46	Kasper	1987-04-20	kcleator19@oaic.gov.au
42	47	Orelle	1988-03-25	odoblin1a@alibaba.com
43	48	Kristopher	1971-01-15	kdensey1b@ft.com
44	49	Torre	1982-11-05	tfell1c@lulu.com
45	50	Bradly	1980-05-10	bmackstead1d@colog-nifty.co...

Total rows: 45 Query complete 00:00:00.084 CRLF Ln 1, Col 24