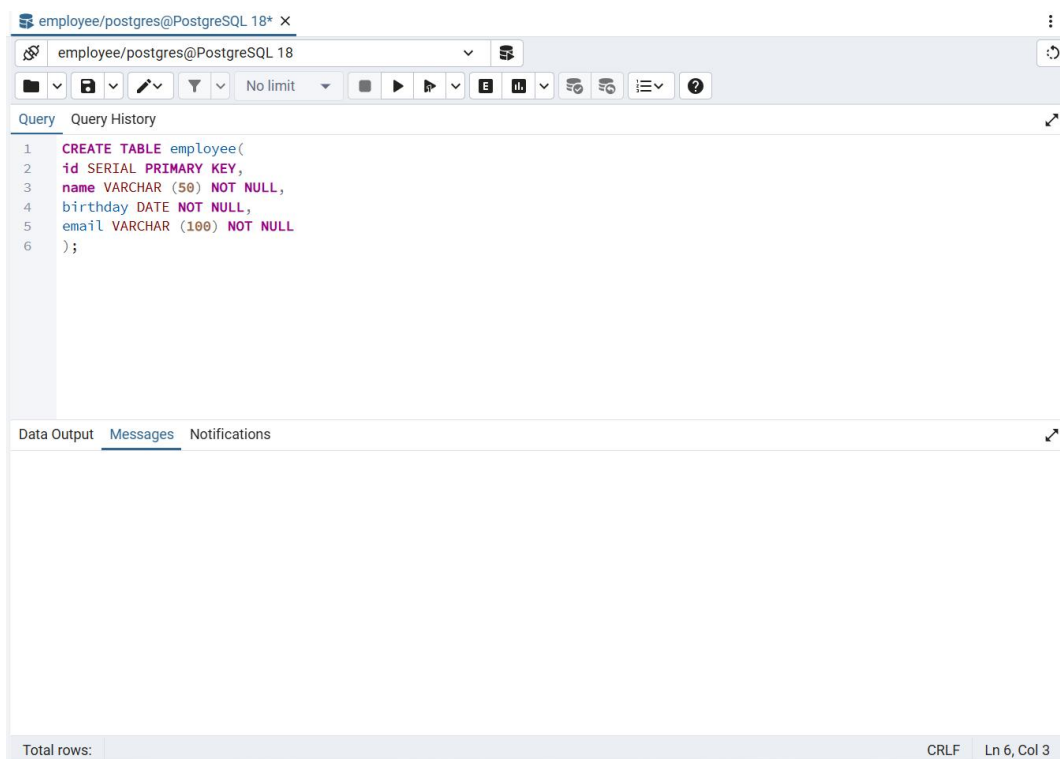


Assignment 8: DVD Rental Queries

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.

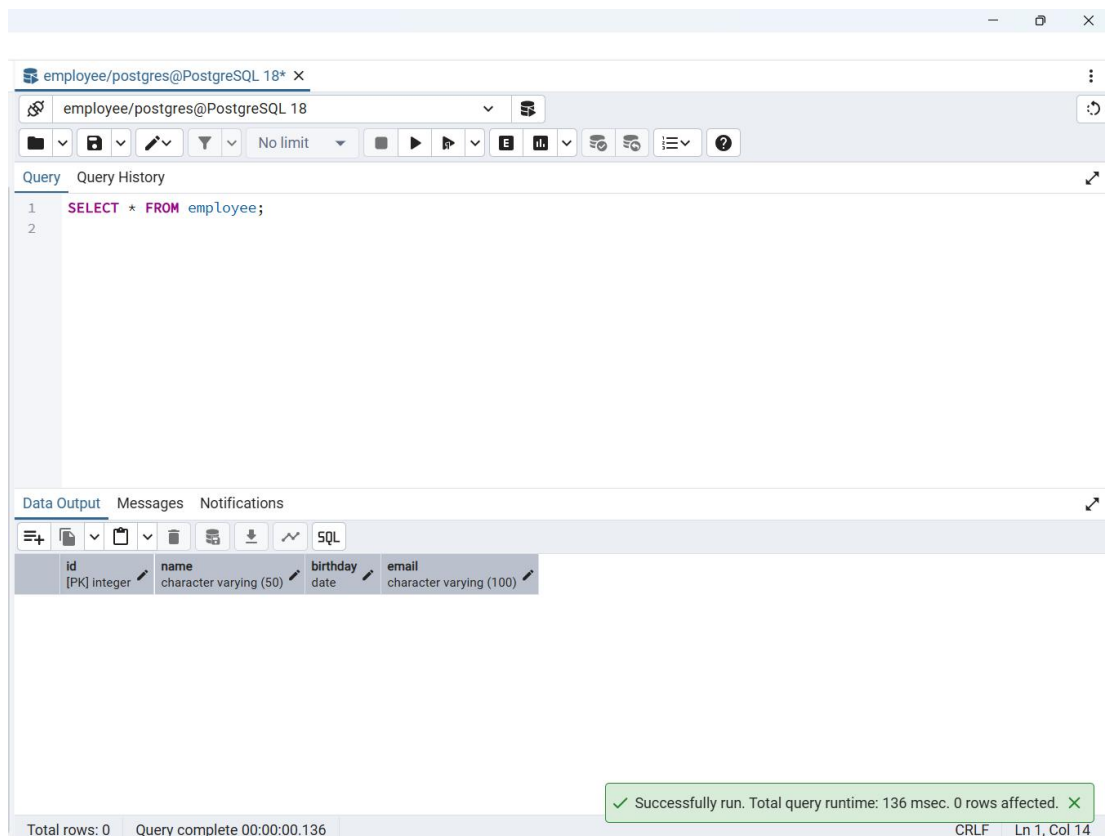


The screenshot shows a PostgreSQL query editor interface. The title bar indicates the connection is to 'employee/postgres@PostgreSQL 18+'. The query editor contains the following SQL code:

```
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 );
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is currently selected. At the bottom of the interface, a status bar shows 'Total rows: 0', 'CRLF', and 'Ln 6, Col 3'.

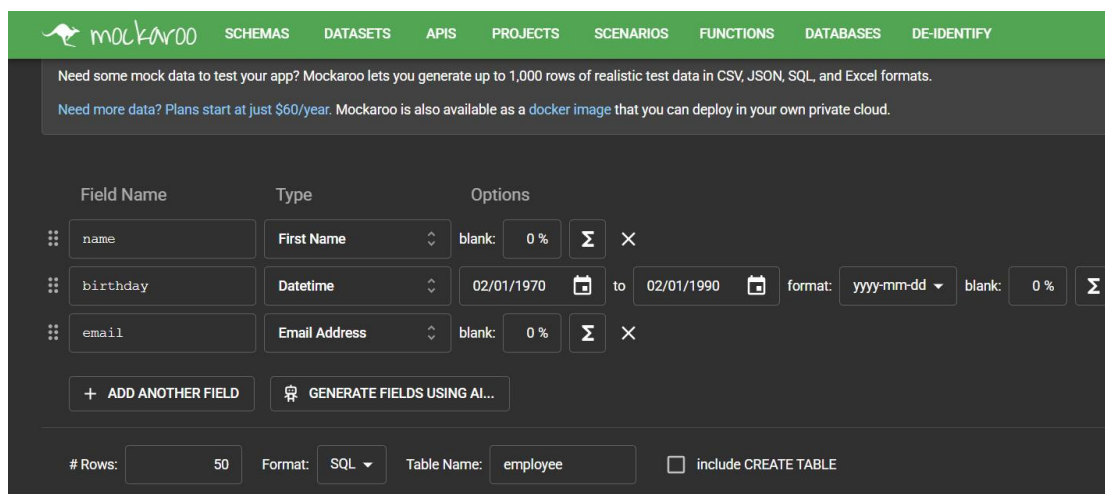
Viewing Table Data



The screenshot shows a PostgreSQL client window titled 'employee/postgres@PostgreSQL 18*'. The query editor contains the SQL statement: `SELECT * FROM employee;`. The 'Data Output' tab is active, displaying the schema of the 'employee' table with four columns: 'id' (integer, primary key), 'name' (character varying (50)), 'birthday' (date), and 'email' (character varying (100)). A green status bar at the bottom indicates: 'Successfully run. Total query runtime: 136 msec. 0 rows affected.' The bottom status bar also shows 'Total rows: 0' and 'Query complete 00:00:00.136'.

Inserting Sample Data with Mockaroo

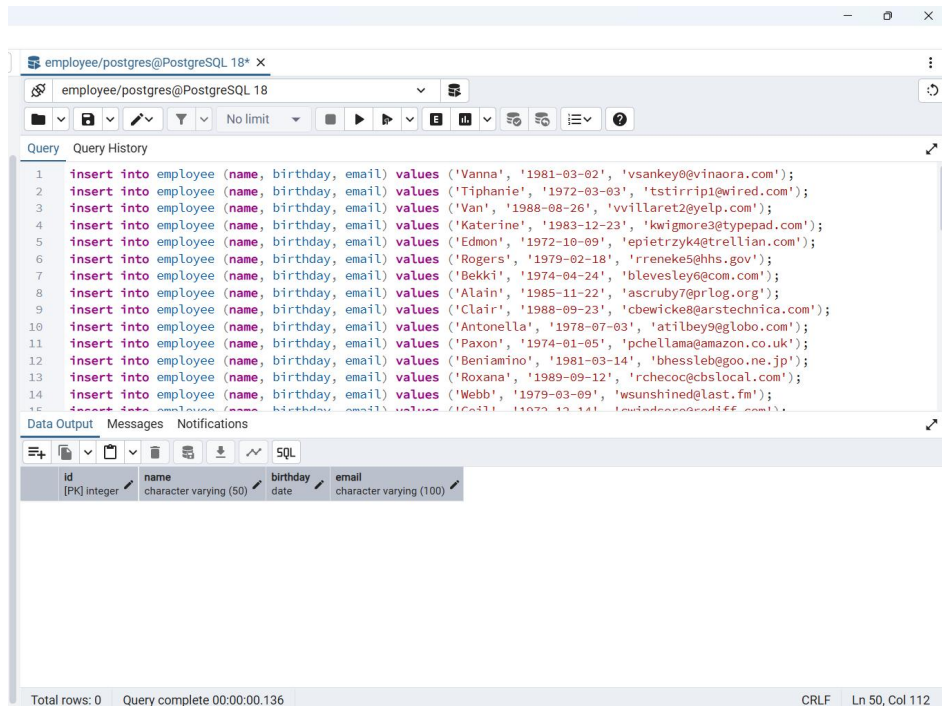
I used the Mockaroo service to generate 50 rows of sample data for the **employee** table. The fields include realistic names, birthdates, and email addresses. This helped test the table with real looking records and practice data insertion in SQL.



The screenshot shows the Mockaroo website interface. The top navigation bar includes links for SCHEMAS, DATASETS, APIS, PROJECTS, SCENARIOS, FUNCTIONS, DATABASES, and DE-IDENTIFY. The main content area has a dark background with a green header. A message states: '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. 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.' Below this, there is a configuration section with three fields: 'name' (First Name), 'birthday' (Datetime), and 'email' (Email Address). Each field has a 'blank' percentage set to 0%. At the bottom, there are buttons for '+ ADD ANOTHER FIELD' and 'GENERATE FIELDS USING AI...'. The bottom status bar shows '# Rows: 50', 'Format: SQL', 'Table Name: employee', and an unchecked checkbox for 'include CREATE TABLE'.

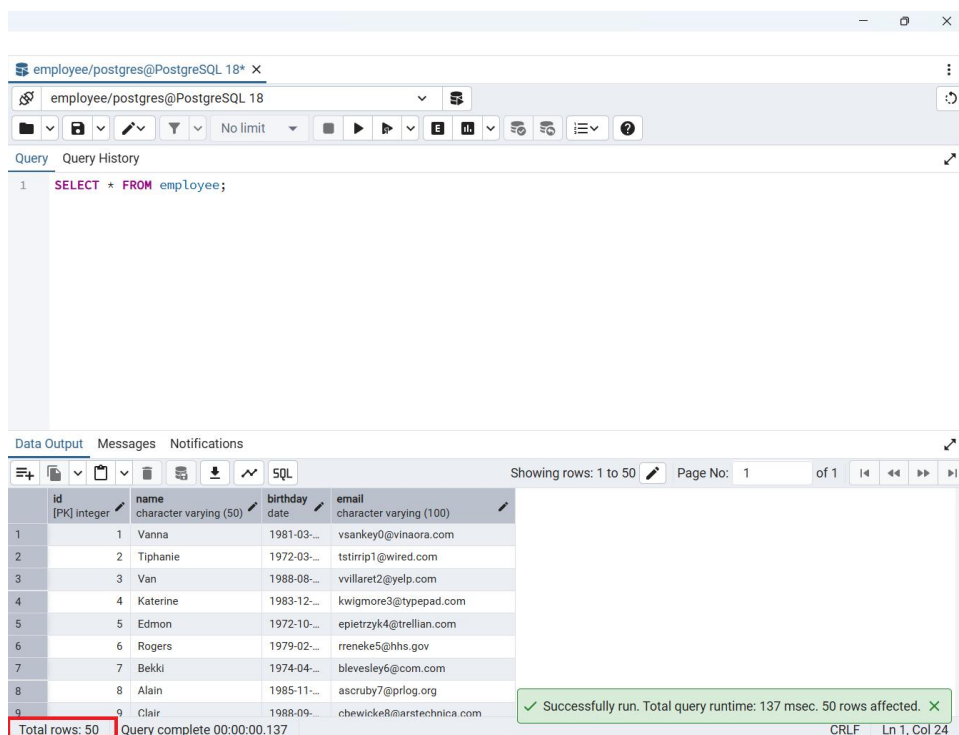
Inserting Records into the Employee Table

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.



The screenshot shows a PostgreSQL client window titled 'employee/postgres@PostgreSQL 18*'. The query editor contains 14 lines of SQL code, each starting with 'insert into employee (name, birthday, email) values' followed by a set of realistic data. Below the query editor, the 'Data Output' tab is active, showing the schema of the 'employee' table: 'id' (PK integer), 'name' (character varying (50)), 'birthday' (date), and 'email' (character varying (100)). The status bar at the bottom indicates 'Total rows: 0' and 'Query complete 00:00:00.136'.

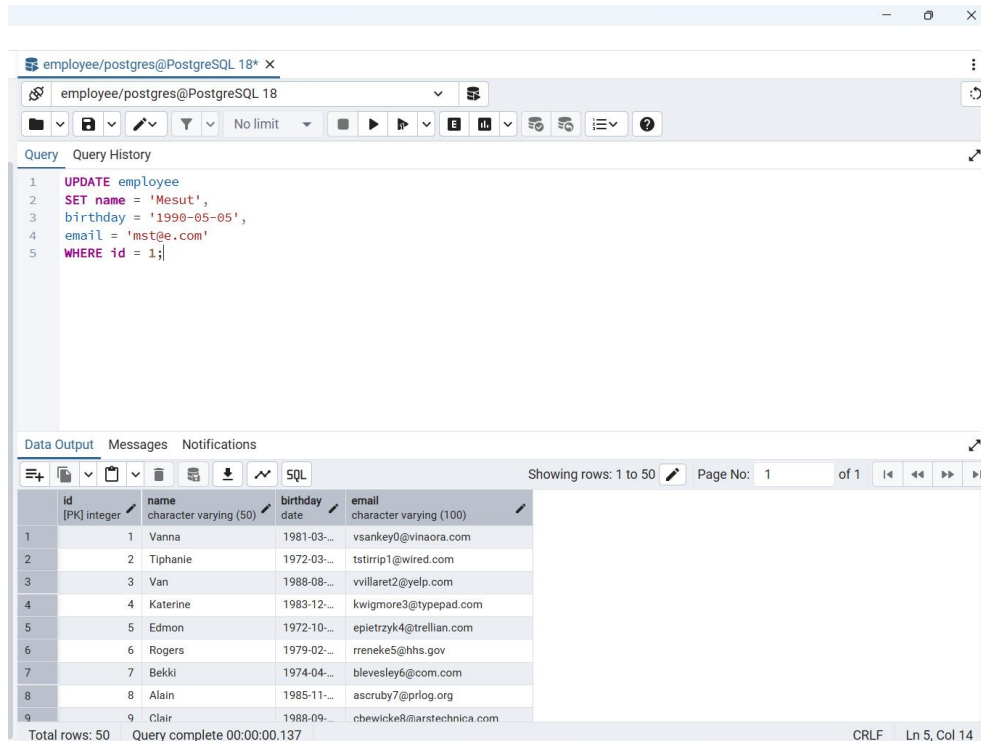
Verifying Inserted Data



The screenshot shows the same PostgreSQL client window. The query editor now contains the query 'SELECT * FROM employee;'. The 'Data Output' tab is active, displaying a table with 50 rows of data. The first row is highlighted, showing '1' in the 'id' column, 'Vanna' in the 'name' column, '1981-03-...' in the 'birthday' column, and 'vsankey0@vinaora.com' in the 'email' column. The status bar at the bottom indicates 'Total rows: 50' and 'Query complete 00:00:00.137'. A green message box at the bottom right states 'Successfully run. Total query runtime: 137 msec. 50 rows affected.'

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.



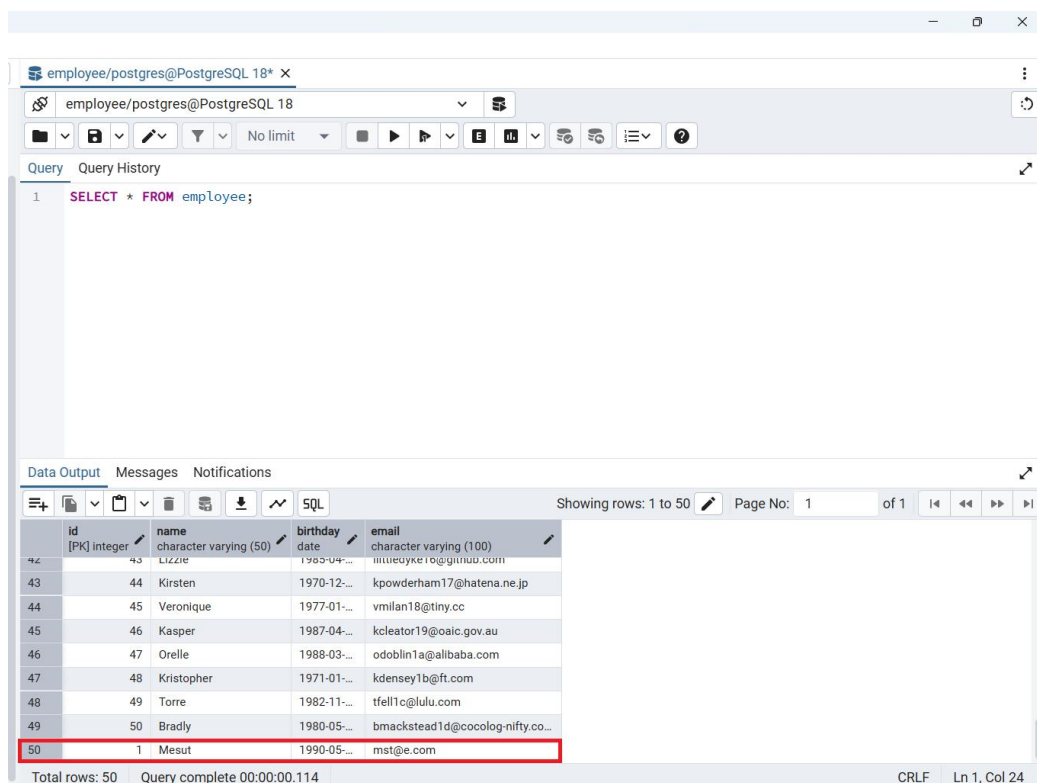
The screenshot shows a PostgreSQL query editor window titled "employee/postgres@PostgreSQL 18* X". The query being executed is:

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

The "Data Output" tab is selected, showing the results of the query. The table has 50 rows, and the query is complete. The status bar indicates "CRLF Ln 5, Col 14".

id [PK] integer	name character varying (50)	birthday date	email character varying (100)
1	Vanna	1981-03-...	vsankey0@vinaora.com
2	Tiphanie	1972-03-...	tstirrip1@wired.com
3	Van	1988-08-...	vvillaret2@yelp.com
4	Katerine	1983-12-...	kwigmore3@typepad.com
5	Edmon	1972-10-...	epletczyk4@trellian.com
6	Rogers	1979-02-...	rreneke5@hhs.gov
7	Bekki	1974-04-...	blevesley6@com.com
8	Alain	1985-11-...	ascruby7@prlog.org
9	Clair	1988-09-...	chewicke8@arstechnica.com

Verifying Update Results:



The screenshot shows the same PostgreSQL query editor window. The query being executed is:

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

The "Data Output" tab is selected, showing the results of the query. The table has 50 rows, and the query is complete. The status bar indicates "CRLF Ln 1, Col 24".

id [PK] integer	name character varying (50)	birthday date	email character varying (100)
43	Lizzie	1963-04-...	lilueuyke16@github.com
44	Kirsten	1970-12-...	kpowderham17@hatena.ne.jp
45	Veronique	1977-01-...	vmilan18@tiny.cc
46	Kasper	1987-04-...	kcleator19@oaic.gov.au
47	Orelle	1988-03-...	odoblin1a@alibaba.com
48	Kristopher	1971-01-...	kdenseny1b@ft.com
49	Torre	1982-11-...	tfell1c@lulu.com
50	Bradly	1980-05-...	bmackstead1d@cocolog-nifty.co...
50	1 Mesut	1990-05-...	mst@e.com

Query 3:

Description: This query changes the name of the employee from **"Tiphannie"** 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 a PostgreSQL query editor window titled "employee/postgres@PostgreSQL 18* X". The query is:

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

The "Data Output" tab shows the results of the query. The table has 10 rows. The first row is highlighted, showing the update of the employee named "Tiphannie" to "Jack".

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
1	2	Tiphannie	1972-03-...	tstirrip1@wired.com
2	3	Van	1988-08-...	vvillaret2@yelp.com
3	4	Katerine	1983-12-...	kwigmore3@typepad.com
4	5	Edmon	1972-10-...	epletzkyk4@trellian.com
5	6	Rogers	1979-02-...	rreneke5@hhs.gov
6	7	Bekki	1974-04-...	blevesley6@com.com
7	8	Alain	1985-11-...	ascruby7@prlog.org
8	9	Clair	1988-09-...	cbewicke8@arstechnica.com
9	10	Antonella	1978-07-...	atilbev9@nolobo.com

Total rows: 50 Query complete 00:00:00.114 CRLF Ln 3, Col 25

Verifying Update Results:

The screenshot shows a PostgreSQL query editor window titled "employee/postgres@PostgreSQL 18* X". The query is:

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

The "Data Output" tab shows the results of the query. The table has 50 rows. The row for the employee named "Jack" is highlighted, showing the successful update.

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
42	44	Kristen	1970-12-...	kpowuerrnain17@riatena.ile.jp
43	45	Veronique	1977-01-...	vmilan18@tiny.cc
44	46	Kasper	1987-04-...	kcleator19@oalc.gov.au
45	47	Orelle	1988-03-...	odoblin1a@alibaba.com
46	48	Kristopher	1971-01-...	kdensey1b@ft.com
47	49	Torre	1982-11-...	tfell1c@lulu.com
48	50	Bradly	1980-05-...	bmackstead1d@cocolog-nifty.co...
49	1	Mesut	1990-05-...	mst@e.com
50	2	Jack	1972-03-...	tstirrip1@wired.com

Total rows: 50 Query complete 00:00:00.092 CRLF Ln 1, Col 24

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 a PostgreSQL query editor window titled "employee/postgres@PostgreSQL 18". The query is:

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

The "Data Output" tab shows the results of the query. The table has 50 rows. The row with ID 3, name "Van", and birthday "1988-08-26" is highlighted in blue, indicating it was updated. The status bar shows "Total rows: 50" and "Query complete 00:00:00.092".

id	name	birthday	email
1	Van	1988-08-26	vvillaret2@yelp.com
2	Katerine	1983-12-...	kwigmore3@typepad.com
3	Edmon	1972-10-...	epletczyk4@trellian.com
4	Rogers	1979-02-...	rreneke5@hhs.gov
5	Bekki	1974-04-...	blevesley6@com.com
6	Alain	1985-11-...	ascruby7@prlog.org
7	Clair	1988-09-...	cbewicke8@arstechnica.com
8	Antonella	1978-07-...	atlibey9@globo.com
9	Paxon	1974-01-...	nchellama@amazon.co.uk

Verifying Update Results:

The screenshot shows a PostgreSQL query editor window titled "employee/postgres@PostgreSQL 18". The query is:

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

The "Data Output" tab shows the results of the query. The table has 50 rows. The row with ID 3, name "Van", and birthday "1990-02-02" is highlighted in blue, indicating it was updated. The status bar shows "Total rows: 50" and "Query complete 00:00:00.116".

id	name	birthday	email
42	veronique	1977-01-...	vimian1@univ.cc
43	Kasper	1987-04-...	kcleator19@oaic.gov.au
44	Orelle	1988-03-...	odoblin1a@alibaba.com
45	Kristopher	1971-01-...	kdensey1b@ft.com
46	Torre	1982-11-...	tfell1c@lulu.com
47	Bradly	1980-05-...	bmackstead1d@cocolog-nifty.co...
48	Mesut	1990-05-...	mst@e.com
49	Jack	1972-03-...	tstirrip1@wired.com
50	Van	1990-02-02	vvillaret2@yelp.com

Query 5:

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

The screenshot shows a PostgreSQL query editor window titled "employee/postgres@PostgreSQL 18* X". The query is as follows:

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

The "Data Output" tab is selected, showing the results of the query. The table has 5 columns: id, name, birthday, and email. The results are as follows:

id	name	birthday	email
4	Katerine	1983-12-...	kwigmore3@typepad.com
5	Edmon	1972-10-...	epletzyk4@trellian.com
6	Rogers	1979-02-...	rreneke5@hhs.gov
7	Bekki	1974-04-...	blevesley6@com.com
8	Alain	1985-11-...	ascrub7@prlog.org
9	Clair	1988-09-...	cbewicke8@arstechnica.com
10	Antonella	1978-07-...	atilbey9@globo.com
11	Paxon	1974-01-...	pchellama@amazon.co.uk
12	Reniamino	1981-03-...	bhessle@nnon ne in

The status bar at the bottom indicates "Total rows: 50" and "Query complete 00:00:00.116".

Verifying Update Results:

The screenshot shows a PostgreSQL query editor window titled "employee/postgres@PostgreSQL 18* X". The query is as follows:

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

The "Data Output" tab is selected, showing the results of the query. The table has 5 columns: id, name, birthday, and email. The results are as follows:

id	name	birthday	email
40	Kasper	1987-04-...	kcreator19@oiaic.gov.au
47	Orelle	1988-03-...	odoblin1a@alibaba.com
48	Kristopher	1971-01-...	kdensey1b@ft.com
49	Torre	1982-11-...	tfell1c@lulu.com
50	Bradly	1980-05-...	bmackstead1d@cocolog-nifty.co...
1	Mesut	1990-05-...	mst@e.com
2	Jack	1972-03-...	tstirrip1@wired.com
3	Van	1990-02-...	vvillaret2@yelp.com
4	Katerine	1983-12-...	kate@rina.com

The status bar at the bottom indicates "Total rows: 50" and "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 a PostgreSQL query editor window titled "employee/postgres@PostgreSQL 18* X". The query is as follows:

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

The "Data Output" tab shows the result: "UPDATE 1" and "Query returned successfully in 56 msec." The status bar at the bottom indicates "Total rows: Query complete 00:00:00.056" and "CRLF Ln 3, Col 6".

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 a PostgreSQL query editor window titled "employee/postgres@PostgreSQL 18* X". The query is as follows:

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

The "Data Output" tab shows a table with 50 rows. The table has columns: id (PK integer), name (character varying (50)), birthday (date), and email (character varying (100)). The row with id 11 (Alan) is highlighted with a red box. The status bar at the bottom indicates "Total rows: 50" and "Query complete 00:00:00.136".

id (PK integer)	name (character varying (50))	birthday (date)	email (character varying (100))
42	Urene	1988-03-...	0000111a@airbus.com
43	Kristopher	1971-01-...	kdensey1b@ft.com
44	Torre	1982-11-...	tfell1c@lulu.com
45	Bradly	1980-05-...	bmackstead1d@cocolog-nifty.co...
46	Mesut	1990-05-...	mst@e.com
47	Jack	1972-03-...	tstirip1@wired.com
48	Van	1990-02-...	vvillaret2@yelp.com
49	Katerine	1983-12-...	kate@rina.com
50	Alan	2000-06-...	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 a PostgreSQL query editor window titled 'employee/postgres@PostgreSQL 18* X'. The query is as follows:

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

The 'Data Output' tab is selected, showing a table with 49 rows. The table has columns: id [PK] integer, name character varying (50), birthday date, and email character varying (100). The row for 'Katerine' (id 4) is highlighted with a red box. The status bar at the bottom indicates 'Total rows: 49' and 'Query complete 00:00:00.097'.

id	name	birthday	email
40	Kasper	1987-04-...	kcreator19@oalc.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@cocolog-nifty.co...
46	Mesut	1990-05-...	mst@e.com
47	Jack	1972-03-...	tstirrip1@wired.com
48	Van	1990-02-...	vvillaret2@yelp.com
49	Katerine	1983-12-...	kate@rina.com

Query 9:

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

The screenshot shows a PostgreSQL query editor window titled 'employee/postgres@PostgreSQL 18* X'. The query is as follows:

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

The 'Data Output' tab is selected, showing a table with 48 rows. The row for 'Van' (id 3) with birthday '1990-02-02' is highlighted with a red box. The status bar at the bottom indicates 'Total rows: 48' and 'Query complete 00:00:00.098'.

id	name	birthday	email
40	Veronique	1977-01-...	vmilian1a@lulu.co
41	Kasper	1987-04-...	kcreator19@oalc.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@cocolog-nifty.co...
46	Mesut	1990-05-...	mst@e.com
47	Jack	1972-03-...	tstirrip1@wired.com
48	Van	1990-02-...	vvillaret2@yelp.com

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 a PostgreSQL query editor window titled 'employee/postgres@PostgreSQL 18*'. The query is:

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

The 'Data Output' tab shows the results of the query. The table has 47 rows. The row with ID 2 and name 'Jack' is highlighted in red. The status bar at the bottom indicates 'Total rows: 47' and 'Query complete 00:00:00.118'.

id	name	birthday	email
44	Kirsten	1970-12-...	kpowderham17@hatena.ne.jp
40	Veronique	1977-01-...	vmilan18@tiny.cc
41	Kasper	1987-04-...	kcleator19@oalc.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@cocolog-nifty.co...
46	Mesut	1990-05-...	mst@e.com
47	Jack	1972-03-...	tstirrip1@wired.com

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 a PostgreSQL query editor window titled 'employee/postgres@PostgreSQL 18*'. The query is:

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

The 'Data Output' tab shows the results of the query. The table has 46 rows. The row with email 'mst@e.com' is highlighted in red. The status bar at the bottom indicates 'Total rows: 46' and 'Query complete 00:00:00.073'.

id	name	birthday	email
43	Lizzie	1963-04-...	litueuyke1o@ginuuo.com
39	Kirsten	1970-12-...	kpowderham17@hatena.ne.jp
40	Veronique	1977-01-...	vmilan18@tiny.cc
41	Kasper	1987-04-...	kcleator19@oalc.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@cocolog-nifty.co...
46	Mesut	1990-05-...	mst@e.com

Verifying Update Results:

employee/postgres@PostgreSQL 18*

employee/postgres@PostgreSQL 18

Query Query History

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

Data Output Messages Notifications

Showing rows: 1 to 45 Page No: 1 of 1

	id [PK] integer	name character varying (50)	birthday date	email character varying (100)
37	42	Halli	1986-05-...	ncanet15@e-recntz4.de
38	43	Lizzie	1985-04-...	llittledyke16@github.com
39	44	Kirsten	1970-12-...	kpowderham17@hatena.ne.jp
40	45	Veronique	1977-01-...	vmilan18@tiny.cc
41	46	Kasper	1987-04-...	kcleator19@oalc.gov.au
42	47	Orelle	1988-03-...	odoblin1a@alibaba.com
43	48	Kristopher	1971-01-...	kdensey1b@ft.com
44	49	Torre	1982-11-...	tfell1c@lulu.com
45	50	Bradly	1980-05-...	bmackstead1d@cocolog-nifty.co...

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