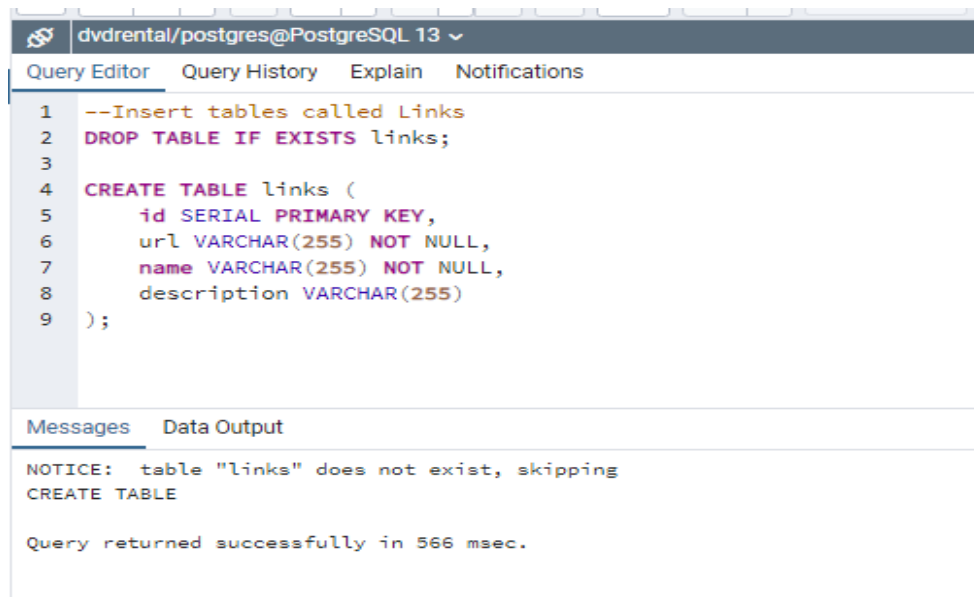


## POSTGRESQL INSERT MULTIPLE ROWS

### Setting up a sample table



The screenshot shows a PostgreSQL Query Editor window with the title bar 'dvdrental/postgres@PostgreSQL 13'. The 'Query Editor' tab is active, displaying a SQL script to create a table named 'links'. The script consists of nine lines: a comment, a DROP statement, and a CREATE TABLE statement with four columns: 'id' (SERIAL PRIMARY KEY), 'url' (VARCHAR(255) NOT NULL), 'name' (VARCHAR(255) NOT NULL), and 'description' (VARCHAR(255)). Below the editor, the 'Messages' tab is active, showing a notice that the table 'links' does not exist and is being skipped, followed by a confirmation that the query returned successfully in 566 msec.

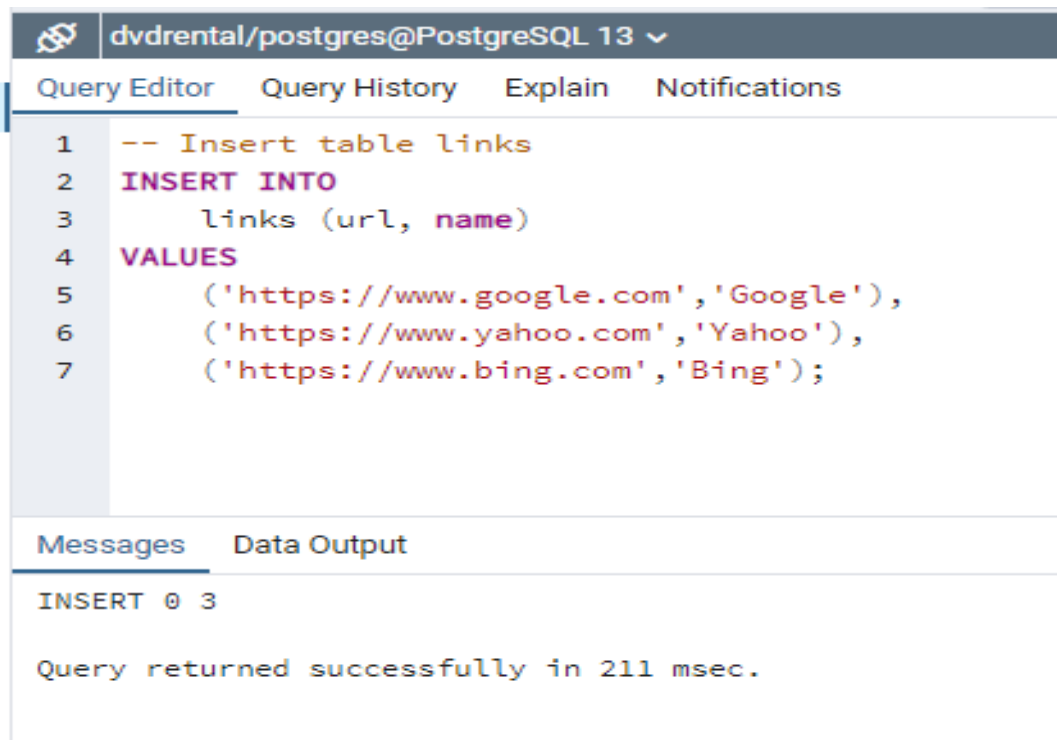
```
1 --Insert tables called Links
2 DROP TABLE IF EXISTS links;
3
4 CREATE TABLE links (
5     id SERIAL PRIMARY KEY,
6     url VARCHAR(255) NOT NULL,
7     name VARCHAR(255) NOT NULL,
8     description VARCHAR(255)
9 );
```

Messages Data Output

NOTICE: table "links" does not exist, skipping  
CREATE TABLE

Query returned successfully in 566 msec.

### Inserting multiple rows




The screenshot shows a PostgreSQL Query Editor window with the title bar 'dvdrental/postgres@PostgreSQL 13'. The 'Query Editor' tab is active, displaying a SQL script to insert three rows into the 'links' table. The script consists of seven lines: a comment, an INSERT INTO statement with columns 'url' and 'name', and a VALUES clause with three rows of data: ('https://www.google.com', 'Google'), ('https://www.yahoo.com', 'Yahoo'), and ('https://www.bing.com', 'Bing'). Below the editor, the 'Messages' tab is active, showing a confirmation that three rows were inserted, followed by a confirmation that the query returned successfully in 211 msec.

```
1 -- Insert table links
2 INSERT INTO
3     links (url, name)
4 VALUES
5     ('https://www.google.com', 'Google'),
6     ('https://www.yahoo.com', 'Yahoo'),
7     ('https://www.bing.com', 'Bing');
```

Messages Data Output

INSERT 0 3

Query returned successfully in 211 msec.


dvdrental/postgres@PostgreSQL 13 ▾

Query Editor
Query History
Explain
Notifications

```


1  --View the Inserted
2  SELECT * FROM links;

```

Messages
Data Output

	id [PK] integer	url character varying (255)	name character varying (255)	description character varying (255)
1	1	https://www.google.com	Google	[null]
2	2	https://www.yahoo.com	Yahoo	[null]
3	3	https://www.bing.com	Bing	[null]

## Inserting multiple rows and returning inserted row


dvdrental/postgres@PostgreSQL 13 ▾

Query Editor
Query History
Explain
Notifications

```

1  --use the INSERT statement to insert two rows into the links table and returns the inserted rows
2  INSERT INTO
3      links(url,name, description)
4  VALUES
5      ('https://duckduckgo.com/', 'DuckDuckGo', 'Privacy & Simplified Search Engine'),
6      ('https://swisscows.com/', 'Swisscows', 'Privacy safe WEB-search')
7  RETURNING *;

```

Messages
Data Output

	id [PK] integer	url character varying (255)	name character varying (255)	description character varying (255)
1	4	https://duckduckgo.com/	DuckDuckGo	Privacy & Simplified Search Engine
2	5	https://swisscows.com/	Swisscows	Privacy safe WEB-search

```
1 --To return the inserted id list, you can specify the id column in the RETURNING clause like this
2 INSERT INTO
3     links(url,name, description)
4 VALUES
5     ('https://www.searchencrypt.com/', 'SearchEncrypt', 'Search Encrypt'),
6     ('https://www.startpage.com/', 'Startpage', 'The world's most private search engine')
7 RETURNING id;
8
```

	id	
	[PK] integer	
1	6	
2	7	