



The screenshot displays a JupyterLab environment with a Jupyter Notebook and a pgAdmin 4 browser window. The notebook contains Python code for connecting to a PostgreSQL database, executing queries, and processing data. The pgAdmin 4 window shows the 'Statistics' tab for a table named 'Hellokate', which displays a single row of statistics.

**Jupyter Notebook Code:**

```

In [9]: def check_if_video_exists(query):
        curr.execute(query)
        #video id as a text
        return curr.fetchall()

In [18]: # def truncate_table():
        # truncate_table()
        # curr.execute(truncate_table())
        #ovo nismo imali

In [19]: def append_from_df_to_db():
        for i, row in df.iterrows():
            insert_into_table(row)
            #ovdje ide i update
            #datum uploada ne

In [20]: #općenito update nove
        #ili sa update row ako
        def update_db(curr, df):
            temp_df = pd.DataFrame()

            for i, row in df.iterrows():

```

**pgAdmin 4 Statistics Table:**

Database	Backends	Xact committed	Xact rolled back	Blocks read
Hellokate	1	561	3	5

Home Page - Select or create a new document

API 2 AWS Kate 2023

localhost:8892/notebooks/API%20AWS%20Kate%202023

jupyter API 2 AWS Kate 2023

File Edit View Insert Cell

```
def check_if_video_exists(video_id):
    # Check if video exists in the database
    cursor.execute("SELECT * FROM public.videos WHERE video_id = %s", (video_id,))
    if cursor.fetchone():
        return True
    else:
        return False

def update_video(video_id, video_title):
    # Update video title in the database
    cursor.execute("UPDATE public.videos SET video_title = %s WHERE video_id = %s", (video_title, video_id))
    conn.commit()

def append_from_df_to_db(df):
    # Append data from DataFrame to the database
    for index, row in df.iterrows():
        video_id = row['video_id']
        video_title = row['video_title']
        update_video(video_id, video_title)

def main():
    # Connect to the database
    host = 'localhost'
    dbname = 'HelloKata'
    port = '5432'
    username = 'postgres'
    password = 'postgres'
    conn = connect_to_db(host, dbname, port, username, password)
    cursor = conn.cursor()

    # Create table
    create_table(cursor)

    # Update video
    update_video(1, 'Queen The Greatest Live: Tie Your Mother Down (Episode 10) out now! #shorts #queen')

    # Append from DataFrame to database
    append_from_df_to_db(df)

    # Execute query
    cursor.execute("SELECT * FROM public.videos ORDER BY video_id ASC")
    results = cursor.fetchall()

    # Print results
    print(results)
```

In [33]:

In [34]: create\_table(cursor)

In [35]: df.update\_db(cursor, df)

In [36]: append\_from\_df\_to\_db(df)

In [37]: cursor.execute("SELECT \* FROM public.videos ORDER BY video\_id ASC")

pgAdmin 4

PostgreSQL 15

Database: HelloKata

Schema: public

Table: videos

Columns: (6)

video\_id

Query:

```
SELECT * FROM public.videos
ORDER BY video_id ASC
```

Query History

Scratch Pad

Data Output

video_id	video_title
1	Queen The Greatest Live: Tie Your Mother Down (Episode 10) out now! #shorts #queen
2	Roger Taylor and Brian May discuss Queen's Live Lighting Secrets #shorts #queen
3	Queen The Greatest Live: Tie Your Mother Down (Episode 10)
4	Queen The Greatest Live: We Are The Champions (Episode 11)
5	Queen + Adam Lambert Return For The Rhapsody Tour Across North America!

Total rows: 5 of 5 Query complete 00:00:00.326 Ln 1, Col 1

Pretraživanje

Automatsko spremanje POSTGRES BAZA SIFRA

Datoteka Polazno Umetanje Crtanje Dizajn Raspored Refer

Ljepljenje

Poništi Međuspremljeno

Navigacija

Pretraživanje u dokumentu

Naslovi Stranice Rezultati

Svorite interaktivnu strukturu dokumenta.

To je sjajan način praćenja gdje se nalazite ili brzog premještanja sadržaja.

Da biste započeli, otvorite karticu Polazno, a zatim primijenite stilove naslova na naslove u dokumentu.

pgAdmin 4

PostgreSQL 15

Database: HelloKata

Schema: public

Table: videos

Columns: (6)

video\_id

Query:

```
SELECT * FROM public.videos
ORDER BY video_id ASC
```

Query History

Scratch Pad

Data Output

	upload_date	view_count	like_count	comment_count
	date	integer	integer	integer
1	de 10) out now! #shorts #queen	2023-03-29	17890	2345
2	Lighting Secrets #shorts #queen	2023-03-22	36426	3249
3	de 10)	2023-03-24	48633	3028
4	de 11)	2023-03-31	18168	2103
5	Across North America!	2023-03-24	133884	2930

Total rows: 5 of 5 Query complete 00:00:00.326 Ln 1, Col 1

Stranica 4 od 4 Broj riječi: 130 hrvatski (Hrvatska) Pristupačnost: Informacije

Pretraživanje

23:10 31.3.2023.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays the database structure, with 'public.videos' selected. The 'Query' pane shows the following SQL query:

```
1 SELECT * FROM public.videos
2 ORDER BY video_id ASC
```

The 'Data Output' pane displays the results of the query:

video_id	video_title
6SEFD609S44	Queen The Greatest Live: Tie Your Mother Down (Episode 10) out now! #shorts #queen
8HmtqXLSRsE	Roger Taylor and Brian May discuss Queen&#39;s Live Lighting Secrets 🎸 #shorts #queen
s8w-KpG5vsc	Queen The Greatest Live: Tie Your Mother Down (Episode 10)
sqZU9_4VnE	Queen The Greatest Live: We Are The Champions (Episode 11)
test1	testtitle
zF-5Co1-s	Queen + Adam Lambert Return For The Rhapsody Tour Across North America!

Total rows: 6 of 6 Query complete 00:00:00.341 Ln 1, Col 1

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays the database structure, with 'public.videos' selected. The 'Query' pane shows the following SQL query:

```
1 SELECT * FROM public.videos
2 ORDER BY video_id ASC
```

The 'Data Output' pane displays the results of the query with additional columns:

video_id	upload_date	view_count	like_count	comment_count
de 10) out now! #shorts #queen	2023-03-29	17890	2345	55
Lighting Secrets 🎸 #shorts #queen	2023-03-22	36426	3249	62
de 10)	2023-03-24	48633	3028	168
ode 11)	2023-03-31	18168	2103	156
2023-04-01	10	322	5	
Across North America!	2023-03-24	133884	2930	361

Total rows: 6 of 6 Query complete 00:00:00.341 Ln 1, Col 1

Azuriran viewcount po mom ideu dodatne tablice test bilo je 10 views sad 50

The screenshot displays a JupyterLab environment with a PostgreSQL database connection. The interface is divided into several panes:

- Top Bar:** Shows the connection name 'pgAdmin 4' and various tool icons.
- Left Pane:** A Jupyter notebook titled 'API 2 AWS Kate 2' containing SQL queries. The queries involve selecting data from a table named 'videos' and performing operations like 'curr.execute' and 'curr.fetchall'.
- Middle Pane:** A database browser showing the 'public' schema. The 'videos' table is highlighted, showing its columns: 'video\_id', 'video\_title', 'upload\_date', 'view\_count', 'like\_count', and 'comment\_count'.
- Right Pane:** A 'Query History' tab showing a list of executed queries. The first query is highlighted, showing the SQL: 'SELECT \* FROM public.videos ORDER BY video\_id ASC'. The results of the query are displayed in a table with 6 rows.

The 'Query History' table shows the following results:

video_id	video_title
1	Queen The Greatest Live: Tie Your Mother Down (Episode 10) out now! #shorts #queensthegreatestlive
2	Roger Taylor and Brian May discuss Queen&#39;s Live Lighting Secrets ! #shorts #queensthegreatestlive
3	Queen The Greatest Live: Tie Your Mother Down (Episode 10)
4	Queen The Greatest Live: We Are The Champions (Episode 11)
5	testtitle
6	Queen + Adam Lambert Return For The Rhapsody Tour Across North America!