

(M:Manuel olarak, K:SQL komutlariyla, C:Python kodlariyla)

1- 'pycoders' isimli bir server kurun. (M)

The screenshot shows the pgAdmin 4 interface with the title bar "pgAdmin 4". The left sidebar lists "Servers (2)" with entries for "PostgreSQL 14" and "pycoders". The main area is titled "Properties" and contains three expandable sections: "General", "Connection", and "SSL".

General:

- ID: 2
- Name: pycoders
- Server type: PostgreSQL
- Version: PostgreSQL 14.2, compiled by Visual C++ build 1914, 64-bit
- Comments: (empty)

Connection:

- Connected? (switch is on)
- Host name/address: localhost
- Port: 5432
- Maintenance database: postgres
- Username: postgres
- Kerberos authentication? (switch is off)
- GSS authenticated? (switch is off)
- GSS encrypted? (switch is off)
- Role: (empty)
- Service: (empty)

SSL:

- (empty)

2- 'class4' database olusturun (M). Database silin (M). Ayni database yine olusturun (K)

The screenshot shows two instances of the pgAdmin 4 interface. The left instance displays the 'Properties' tab for a database named 'class4'. The right instance shows the 'Query Editor' where a 'CREATE DATABASE class4;' command has been run.

Properties Tab (Left):

- Database:** class4
- OID:** 16394
- Owner:** postgres
- System database?**: Off
- Comment:** (Empty)

Query Editor Tab (Right):

```
1 CREATE DATABASE class4;
```

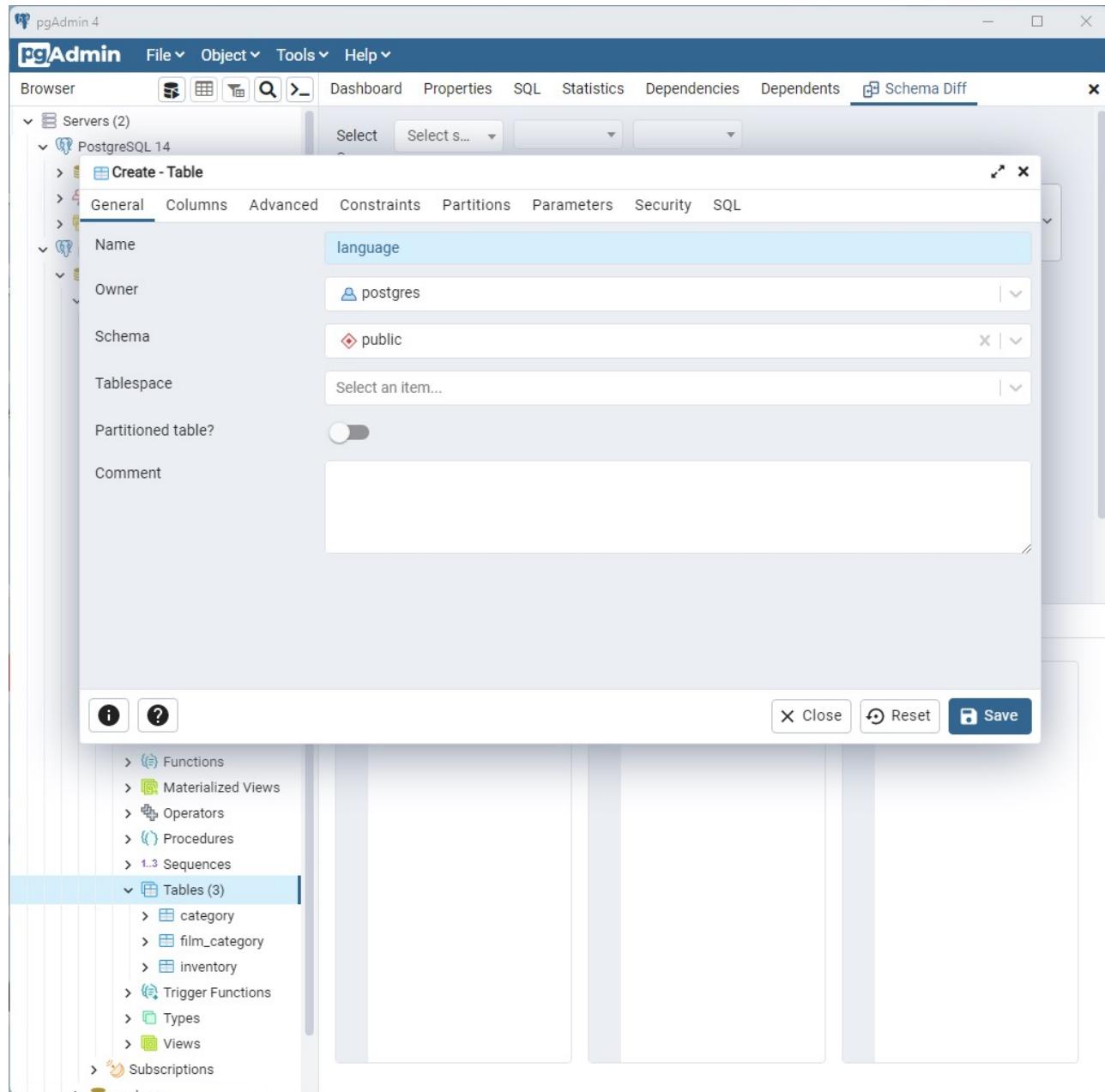
Messages Tab (Bottom Right):

```
CREATE DATABASE

Query returned successfully in 640 msec.
```

3- <https://www.postgresqltutorial.com/postgresql-sample-database/> adresine gidin ve ER modeli inceleyin. Tablolar arasindaki en az 5 iliskiyi yazın.(Hangi tablolar arasında ne tur bir ilişki var)

4- ER modeldeki tablolardan 3 tanesini M olusturun.



5- ER modeldeki tablolardan 3 tanesini K olusturun.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Servers' tree view shows 'PostgreSQL 14' and 'pycoders' databases. Under 'pycoders', there is a 'class4' database which contains several objects like Casts, Catalogs, Event Triggers, etc. The 'Tables (3)' node under 'class4' is selected, listing 'category', 'film_category', and 'inventory'. In the main pane, a SQL query editor window displays the following CREATE TABLE statements:

```
1 CREATE TABLE category (
2     category_id serial PRIMARY KEY,
3     name VARCHAR ( 50 ) UNIQUE NOT NULL,
4     last_update TIMESTAMP NOT NULL
5 );
6
7 CREATE TABLE inventory (
8     inventory_id serial PRIMARY KEY,
9     film_id serial NOT NULL,
10    store_id VARCHAR ( 50 ) NOT NULL,
11    last_update TIMESTAMP NOT NULL
12 );
13
14 CREATE TABLE film_category (
15     film_id serial PRIMARY KEY,
16     category_id VARCHAR ( 50 ) UNIQUE NOT NULL,
17     last_update TIMESTAMP NOT NULL
18 );
19
```

Below the query editor, the 'Messages' tab is active, showing the message: 'Query returned successfully in 39 msec.'

6- ER modeldeki tablolardan 3 tanesini C olusturun.

```
1 import psycopg2
2
3 try:
4     conn = psycopg2.connect(database = "class4", user = "postgres", password = "123456", host = "localhost", port = "5432")
5 except:
6     print("I am unable to connect to the database")
7
8 cur = conn.cursor()
9 try:
10     cur.execute("CREATE TABLE payment (payment_id serial PRIMARY KEY, amount integer);")
11 except:
12     print("I can't drop our test database!")
13
14 conn.commit() # <-- makes sure the change is shown in the database
15 conn.close()
16 cur.close()
```

(4-5-6. sorulari cozerken toblolar arasindaki iliskileri gozardi edebilirsiniz.
Tum kolonlari girmek zorunda degilsiniz, en az 2 kolon olmasi yeterli.)

7- Olusturdugunuz 3 tabloya M ile 5 veri girisi yapin.

Data Output				Explain	Messages	Notifications
	film_id [PK] integer	category_id character varying (50)	last_update timestamp without time zone			
1	156	2	2013-05-26 14:50:58.951			
2	698	4	2014-05-26 14:50:58.951			
3	7896	8	2015-05-26 14:50:58.951			

8- Olusturdugunuz 3 tabloya K ile 5 veri girisi yapin.

Query Editor Query History

1 `INSERT INTO inventory (inventory_id, film_id, store_id, last_update) VALUES (147, 69, 11, '2006-02-15 10:09:17');`

	inventory_id [PK] integer	film_id integer	store_id character varying (50)	last_update timestamp without time zone
1	145	69	11	2006-02-15 10:09:17
2	147	69	11	2006-02-15 10:09:17

Data Output Explain Messages Notifications

INSERT 0 1

Query returned successfully in 40 msec.

9- Olusturdugunuz 3 tabloya C ile 5 veri girisi yapin.

10- 3 tablodaki birer veriyi M ile degistirin.

11- 3 tablodaki birer veriyi K ile degistirin.

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

- Connection:** class4/postgres@pycoders
- Tab:** Query Editor (selected)
- Query:**

```
1 UPDATE language
2 SET last_update = '2020-08-01'
3 WHERE language_id = 3;
4
```
- Execution Results:**
 - Messages tab is selected.
 - Output:

```
UPDATE 1
```
 - Message:

```
Query returned successfully in 31 msec.
```

12- 3 tablodaki birer veriyi C ile degistirin.

13- 3 tablonun son satirini M ile silin.

14- 3 tablonun son satirini K ile silin.

A screenshot of a PostgreSQL query editor interface. The title bar says "class4/postgres@pycoders". The tabs at the top are "Query Editor" (selected) and "Query History". The main area contains the following SQL code:

```
1 DELETE FROM language
2 WHERE language_id = 6;
3
```

Below the code, there are four tabs: "Data Output", "Explain", "Messages" (selected), and "Notifications". The "Messages" tab shows the output of the query:

```
DELETE 1
Query returned successfully in 187 msec.
```

15- 3 tablonun son satirini C ile silin.

16- 1 tabloyu M ile silin.

17- 1 tabloyu K ile silin.

A screenshot of a PostgreSQL query editor interface. The title bar says "class4/postgres@pycoders". The tabs at the top are "Query Editor" (selected) and "Query History". The main area contains the following SQL code:

```
1 DROP TABLE language;
2
```

Below the code, there are four tabs: "Data Output", "Explain", "Messages" (selected), and "Notifications". The "Messages" tab shows the output of the query:

```
DROP TABLE
Query returned successfully in 62 msec.
```

18- 1 tabloyu C ile silin.

19- Kalan tablolardan 1 tanesinin 2 veya 3 sutununu K ile baska bir tablo olarak olusturun.

20- Kalan tablolardan 1 tanesinin 2 veya 3 sutununu C ile baska bir tablo olarak olusturun.

21- Tablolardan 1 tanesini M ile truncate edin.

The screenshot shows the pgAdmin 4 interface. On the left is a tree view of database objects. In the center, a context menu is open over a schema named 'public'. The menu items are:

- Count Rows
- Create >
- Delete/Drop
- Refresh...
- Restore...
- Backup...
- Drop Cascade
- Import/Export Data...
- Reset Statistics
- Maintenance...
- Scripts >
- Truncate > **Truncate** (highlighted)
- View/Edit Data
- Search Objects...
- PSQL Tool
- Query Tool
- Properties...

On the right, there are two panes: 'Query Editor' containing a partial SQL query and 'Data Output' showing a table with two rows.

```
1 SELECT
2 ORDER
```

staff_id	[PK] integ
1	
2	

22- Tablolardan 1 tanesini K ile truncate edin.

The screenshot shows the pgAdmin 4 interface. On the left, the Server Explorer tree view shows two servers: PostgreSQL 14 and pycoders. Under pycoders, there are two databases: class4 and public. The class4 database is expanded, showing various schema objects like Casts, Catalogs, Event Triggers, etc. The public database is also expanded, showing tables such as film_category, inventory, language, payment, Trigger Functions, Types, and Views. The central pane is the Query Editor, which contains the following SQL command:

```
1 TRUNCATE TABLE payment;
```

Below the query, the Messages tab is selected, showing the result of the execution:

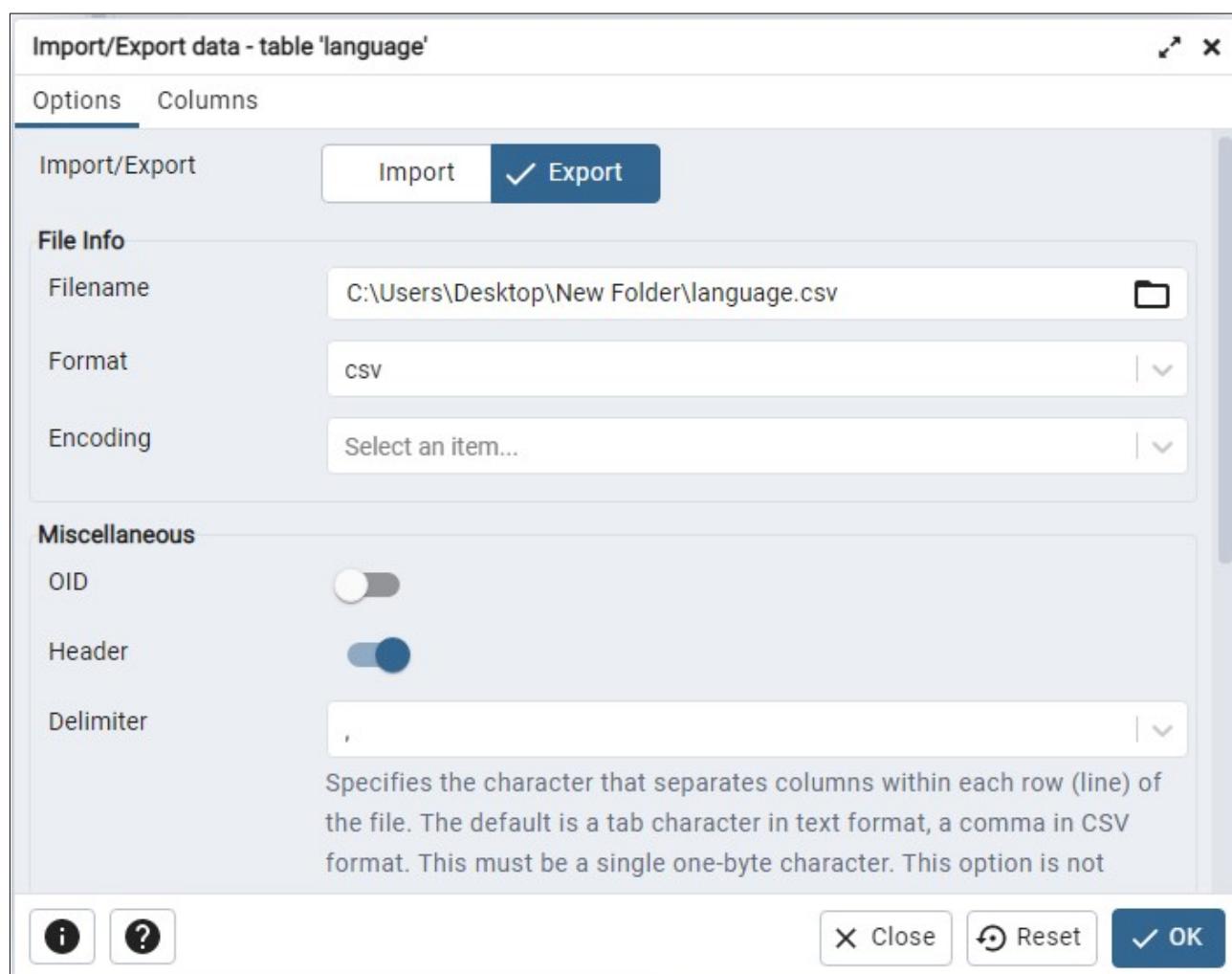
```
TRUNCATE TABLE
```

Query returned successfully in 33 msec.

23- Tablolardan 1 tanesini C ile truncate edin.

```
1 import psycopg2
2
3 try:
4     conn = psycopg2.connect(database = "class4", user = "postgres", password = "123456",
5 except:
6     print("I am unable to connect to the database")
7
8 cur = conn.cursor()
9 try:
10    cur.execute("TRUNCATE TABLE inventory ;" )
11 except:
12    print("I can't drop our test database!")
13
14 conn.commit() # <--- makes sure the change is shown in the database
15 conn.close()
16 cur.close()
```

- 24- Truncate edilmiş tablolari M ile silin.
- 25- 2 tabloyu K ile silin.
- 26- 2 tabloyu C ile silin.
- 27- Elimizde veri olan 1 tablo kalmış olması lazım. Bu tabloyu csv olarak bilgisayarınıza yükleyin.



- 28- Postgresql arayuzundeki son tabloyu da K ile silin.
- 29- Bilgisayarınızdaki csv yi arayuze import edin.
- 30- Import ettiğiniz bu tabloyu C ile silin.

31- <https://www.postgresqltutorial.com/postgresql-sample-database/> linkindeki örnek DB yi bilgisayariniza indirin ve arayuze yukleyin.

32- DB nizde 15 adet tablo olmasi lazim. Her tabloyu teker teker goruntuleyin ve kolon isimlerine bakarak, 5 tabloda hangi kolonun PK ve FK oldugunu yazin.

Table Name	PK	FK
1.actor	actor_id	---
2.address	address_id	city_id
3.category	category_id	---
4.city	city_id	country_id
5.country	country_id	---

Sorgular? (Asagidaki sorularin cevaplarini ve bu cevabi bulurken kullandiginiz kodlari yazin)

33- Action filmlerinin ortalama suresi ne kadar?

The screenshot shows a PostgreSQL query editor interface. The top bar displays the connection information: **dvdrental/postgres@pycoders**. Below the bar, there are two tabs: **Query Editor** (which is selected) and **Query History**. The main area contains the following SQL code:

```
1 select avg(length) from film
2 where film_id in (select film_id from film_category
3 where category_id = 1 )
4
5
```

Below the code, there are four tabs: **Data Output** (selected), **Explain**, **Messages**, and **Notifications**. The **Data Output** tab displays the result of the query:

	avg	
	numeric	lock
1	111.609375000000000000	

34- En çok staff olan store hangisidir?

35- 'Gene Willis' adli actorun oynadigi filmlerin ratingi nedir?

The screenshot shows the pgAdmin 4 interface. At the top, there's a toolbar with various icons for database management. Below the toolbar, the connection status is shown as "dvdrental/postgres@pycoders". The main area has tabs for "Query Editor" (which is selected) and "Query History".

The Query Editor contains the following SQL code:

```
1 select film_id,rating from film
2
3 where film_id in (
4 select film_id from film_actor
5 where actor_id=(select actor_id from actor
6 where first_name = 'Gene' and last_name='Willis'));
7
8 --order by first_name
9
```

Below the code, there are tabs for "Data Output", "Explain", "Messages", and "Notifications". The "Data Output" tab is selected and displays a table with two columns: "film_id" and "rating". The table contains 23 rows of data.

	film_id	rating
1	8	R
2	36	PG-13
3	40	R
4	54	R
5	58	G
6	66	NC-17
7	134	PG
8	209	NC-17
9	244	NC-17
10	320	PG-13
11	430	G
12	452	R
13	486	R
14	572	PG-13
15	590	PG
16	661	PG-13
17	778	G
18	832	PG
19	846	NC-17
20	874	PG
21	945	R
22	968	G
23	987	PG

36- Aktif customer sayisi nedir?

Query Editor Query History

```
1 select count(active) from customer
2 where active = 1
3
4
```

Data Output Explain Messages Notifications

	count	lock
1	584	

37- 'C' harfiyle baslayan filmler hangileridir?

The screenshot shows a PostgreSQL query editor interface. The top navigation bar includes 'Dashboard', 'Properties', 'SQL', 'Statistics', 'Dependencies', and 'Dependents'. Below the navigation is a toolbar with various icons. The connection information 'dvdrental/postgres@pycoders' is displayed. The 'Query Editor' tab is selected, showing the following SQL code:

```
1 select title from film
2
3 where title LIKE 'C%' OR title LIKE 'c%'
```

The 'Data Output' tab is selected, displaying the results of the query:

	title
1	Chamber Italian
2	Casualties Encino
3	Cabin Flash
4	Caddyshack Jedi
5	Calendar Gunfight
6	California Birds
7	Camelot Vacation
8	Campus Remember
9	Candidate Perdition
10	Candles Grapes
11	Canyon Stock
12	Caper Motions
13	Caribbean Liberty
14	Carol Texas
15	Carrie Bunch
16	Casablanca Super
17	Casper Dragonfly
18	Cassidy Wyoming
19	Cat Coneheads
20	Catch Amistad
21	Cause Date
22	Celebrity Horn
23	Center Dinosaur
24	Chainsaw Uptown
25	Champion Flatliners
26	Chance Resurrection
27	Cloudy Foothills

38- 4\$ den az odeme yapan musterilerin e-mail adresleri nedir?

The screenshot shows a PostgreSQL query editor interface. At the top, there are tabs for Dashboard, Properties, SQL, Statistics, Dependencies, and a connection dropdown set to dvdrental/postgres@pycoders. Below the connection dropdown are several icons for file operations like Open, Save, and Delete. The main area has two tabs: Query Editor (selected) and Query History. The Query Editor tab contains the following SQL code:

```
1 select email from customer
2 where customer_id in (
3 select customer_id from payment
4 where amount < 4
5 order by amount)
6
7
8
9
10
```

Below the code, there are tabs for Data Output, Explain, Messages, and Notifications. The Data Output tab is selected and displays a table with one column labeled 'email'. The table has 20 rows, each containing an email address starting with a number from 1 to 20 followed by a unique email suffix. The first few rows are:

	email
1	jared.ely@sakilacustomer.org
2	mary.smith@sakilacustomer.org
3	patricia.johnson@sakilacustomer.org
4	linda.williams@sakilacustomer.org
5	barbara.jones@sakilacustomer.org
6	elizabeth.brown@sakilacustomer.org
7	jennifer.davis@sakilacustomer.org
8	maria.miller@sakilacustomer.org
9	susan.wilson@sakilacustomer.org
10	margaret.moore@sakilacustomer.org
11	dorothy.taylor@sakilacustomer.org
12	lisa.anderson@sakilacustomer.org
13	nancy.thomas@sakilacustomer.org
14	karen.jackson@sakilacustomer.org
15	betty.white@sakilacustomer.org
16	helen.harris@sakilacustomer.org
17	sandra.martin@sakilacustomer.org
18	donna.thompson@sakilacustomer.org
19	carol.garcia@sakilacustomer.org
20	edith.martinez@sakilacustomer.org

39- Moscow'da ikamet eden staff ve customer tablosu? (sadece isim/soyisim sutunu olsun)

The screenshot shows a database interface with a toolbar at the top, followed by two tabs: "Query Editor" (selected) and "Query History". Below these are two code snippets. The first snippet is a complex query involving three tables: customer, address, and city. The second snippet is a simple SELECT statement. The "Data Output" tab is selected, displaying a table with two columns: "first_name" and "last_name". There is one row with values "Catherine" and "Campbell".

```
1 select first_name, last_name from customer
2 where address_id in (
3 select address_id from address
4 where city_id = (
5 select city_id from city
6 where city = 'Moscow'))
```

```
1
2
3
4
5
6
7
8
9
```

	first_name	last_name
1	Catherine	Campbell

40- En az kiralanan 5 film hangisidir?

The screenshot shows the MySQL Workbench interface. The top menu bar includes Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, and a toolbar with various icons. Below the menu is a sub-menu with icons for file operations like Open, Save, Print, and a search bar. The main area has tabs for Query Editor (which is selected) and Query History. The Query Editor contains the following SQL code:

```
1 select title from film
2 where film_id in (
3 select film_id from inventory
4 where inventory_id in (select inventory_id from rental)
5 group by film_id
6 order by sum(inventory_id)
7 limit 5)
```

Below the code, there are tabs for Data Output, Explain, Messages, and Notifications. The Data Output tab displays a table with one column, 'title', containing five rows of film titles:

title
Ace Goldfinger
Academy Dinosaur
Adaptation Holes
African Egg
Affair Prejudice

41- 2006 yılında yayınlanan İngilizce filmler hangileridir?

The screenshot shows a PostgreSQL query editor interface. At the top, there's a toolbar with various icons for database management. Below the toolbar, tabs for "Query Editor" and "Query History" are visible, with "Query Editor" being active. The main area contains a numbered SQL query:

```
1 select title from film
2 where release_year = '2006' and language_id=(select language_id from language
3                               where name='English' )
4
```

Below the query results, there are tabs for "Data Output", "Explain", "Messages", and "Notifications". The "Data Output" tab is active, displaying a table with 19 rows of film titles. The table has a header row with a lock icon and a title column.

	title
1	Chamber Italian
2	Grosse Wonderful
3	Airport Pollock
4	Bright Encounters
5	Academy Dinosaur
6	Ace Goldfinger
7	Adaptation Holes
8	Affair Prejudice
9	African Egg
10	Agent Truman
11	Airplane Sierra
12	Alabama Devil
13	Aladdin Calendar
14	Alamo Videotape
15	Alaska Phantom
16	Date Speed
17	Ali Forever
18	Alice Fantasia
19	Alien Center