

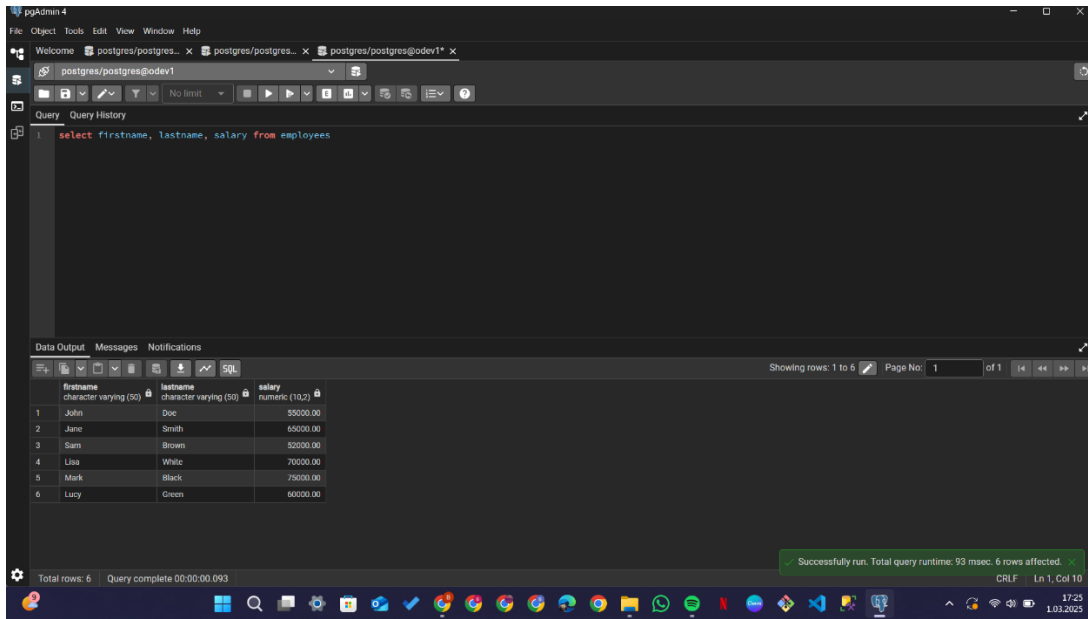
SQL BOOTCAMP ODEV1

BENSU İMAMOĞLU

bensuimamoglu@gmail.com

a. Belirli Kolonları Seçme

Basitçe bir **select** komutu yazdım ve ardından * koymak yerine her şeyi çekmek için, yalnızca çekmek istediğim verilerim kolon adlarını girdim.



The screenshot shows the pgAdmin 4 interface. The query editor contains the following SQL query:

```
select firstname, lastname, salary from employees
```

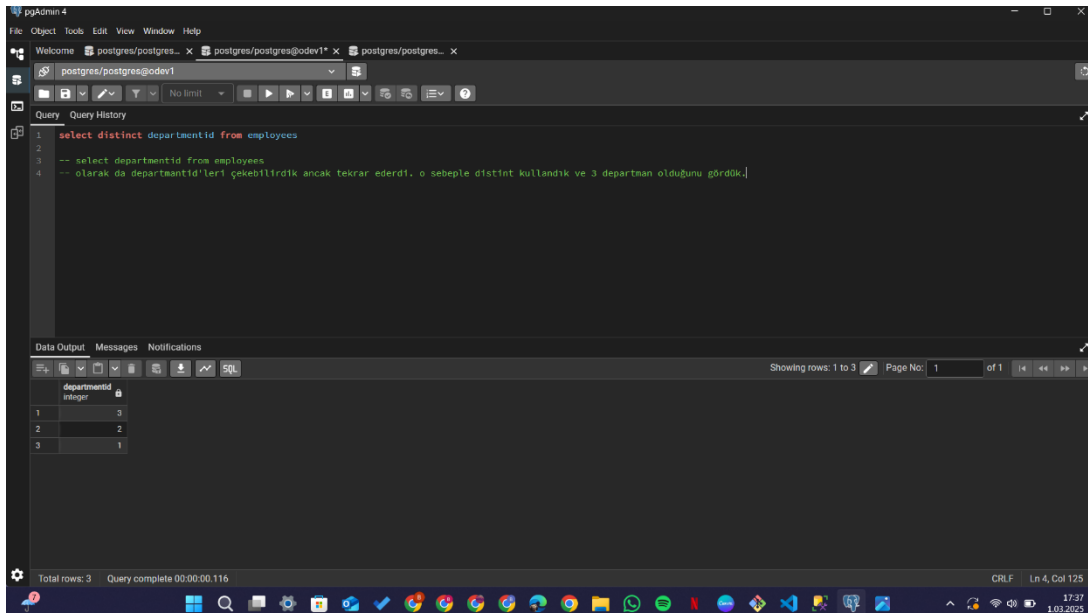
The Data Output tab shows the results of the query:

firstname	lastname	salary
John	Doc	55000.00
Jane	Smith	65000.00
Sam	Brown	52000.00
Lisa	White	70000.00
Mark	Black	75000.00
Lucy	Green	60000.00

The status bar at the bottom indicates: Total rows: 6, Query complete 00:00:00.093, Successfully run. Total query runtime: 93 msec. 6 rows affected.

b. DISTINCT Komutu ile Tekrarları Önleme

departman isimlerini tekrar etmeme için **distinct** komutunu kullandık. Kullanmasaydık eğer tüm çalışanlar kadar departman isimlerini verecekti bize output olarak.



The screenshot shows the pgAdmin 4 interface. The query editor contains the following SQL query:

```
select distinct departmentid from employees
```

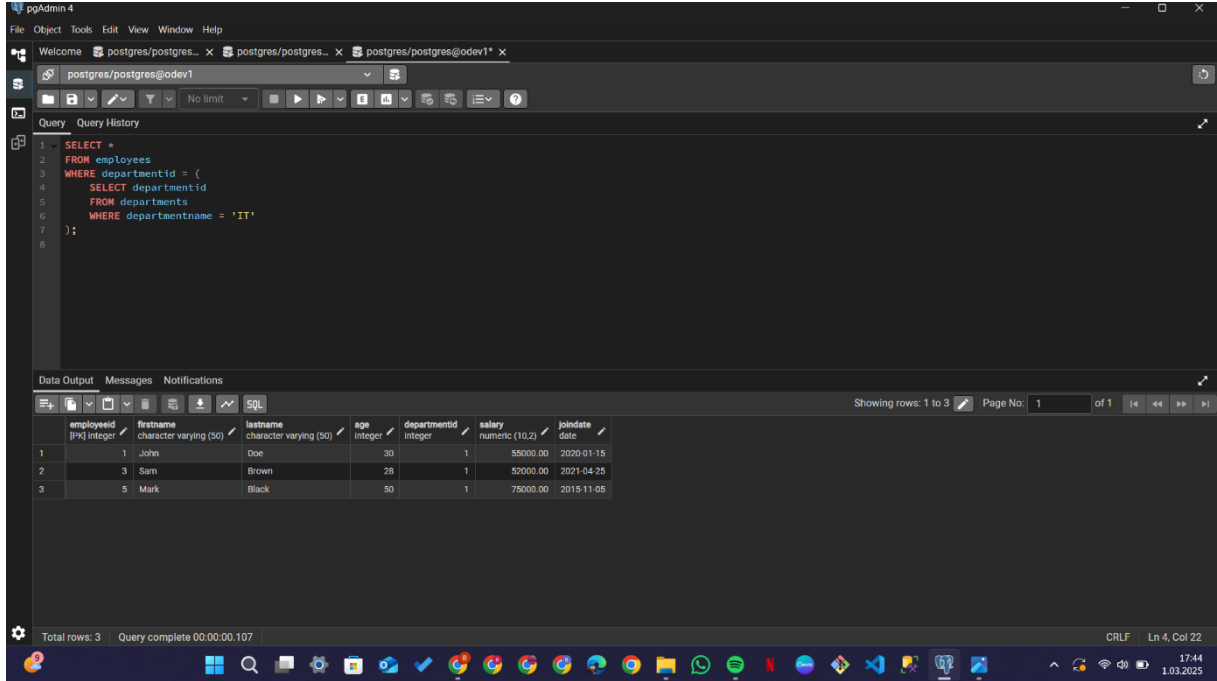
The Data Output tab shows the results of the query:

departmentid
3
2
1

The status bar at the bottom indicates: Total rows: 3, Query complete 00:00:00.116.

c. Belirli Bir Departmana Ait Çalışanları Listeleme

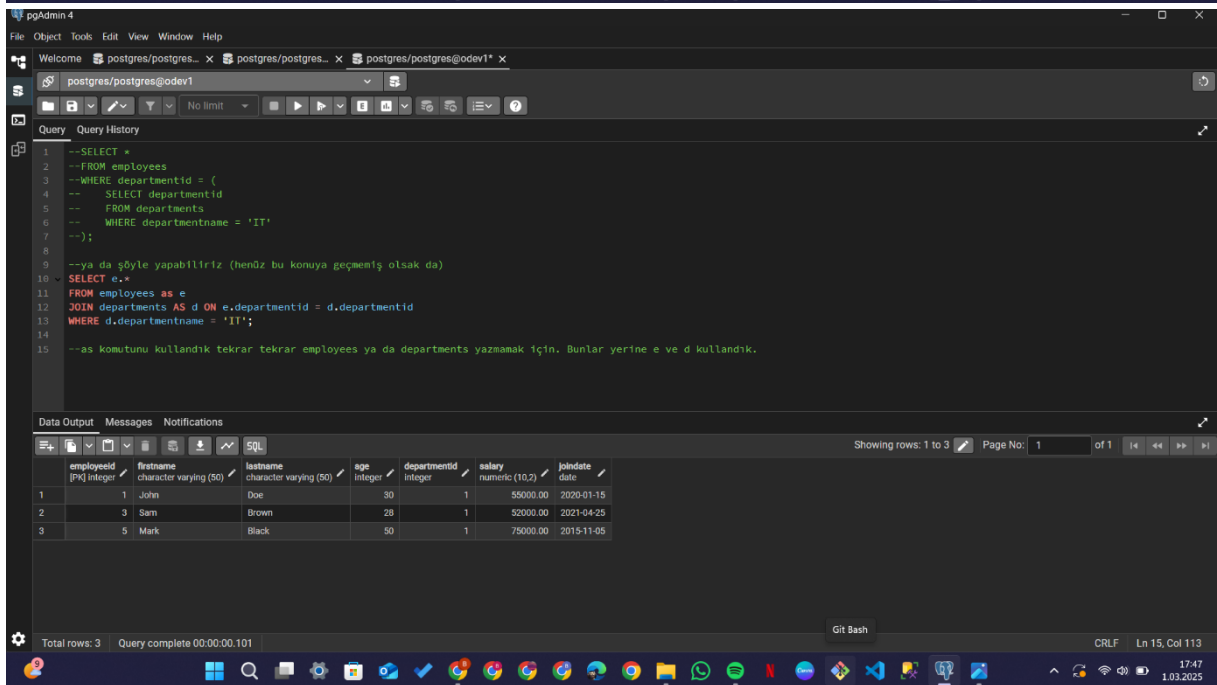
Bunu farklı şekillerde yapabiliriz. **Biri join kullanarak diğeri kullamadan**. Join komutunu henüz görmesek de eklemek istedim. Ek olarak 2. ekran görüntüsünde employees ve departmentsı tekrarlamamak için **as** komutunu kullandım ki derste hocamız da işlemiştii.



The screenshot shows the pgAdmin 4 interface. The Query History tab displays a SQL query that filters employees by department name 'IT'. The Data Output tab shows the results of the query, which are three rows of employee data.

```
1 SELECT *
2 FROM employees
3 WHERE departmentid = (
4     SELECT departmentid
5     FROM departments
6     WHERE departmentname = 'IT'
7 );
8
```

employeeid	firstname	lastname	age	departmentid	salary	hiredate
1	John	Doe	30	1	55000.00	2020-01-15
2	Sara	Brown	28	1	52000.00	2021-04-25
3	Mark	Black	50	1	75000.00	2015-11-05



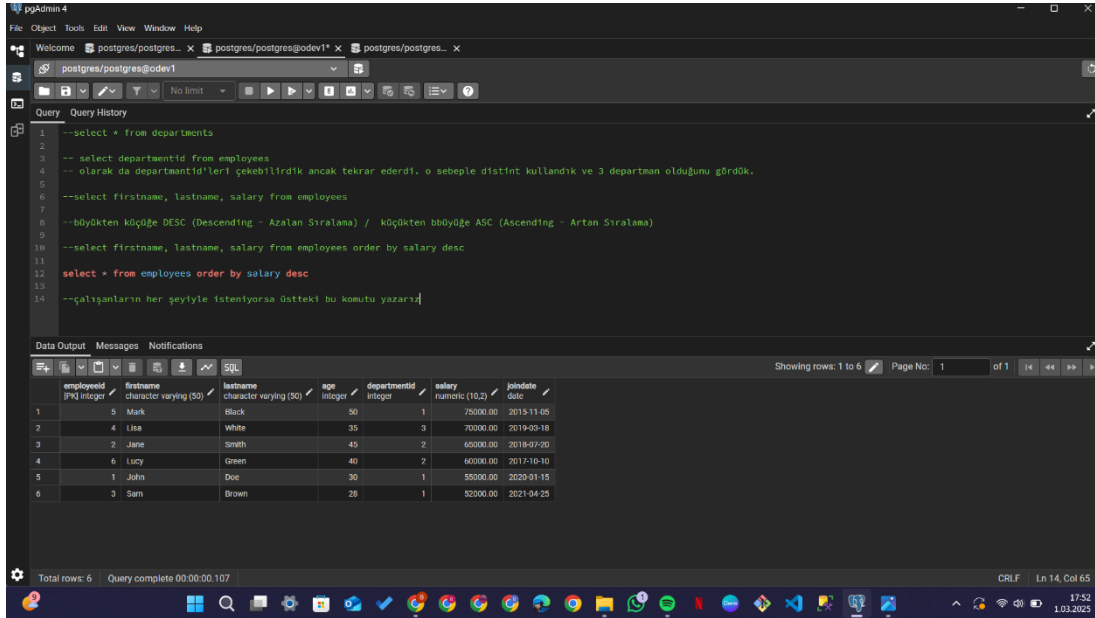
The screenshot shows the pgAdmin 4 interface. The Query History tab displays a SQL query that uses a join to filter employees by department name 'IT'. The Data Output tab shows the results of the query, which are three rows of employee data.

```
1 --SELECT *
2 --FROM employees
3 --WHERE departmentid = (
4 --    SELECT departmentid
5 --    FROM departments
6 --    WHERE departmentname = 'IT'
7 --);
8
9 --ya da şöyle yapabiliriz (henüz bu konuya geçmemiş olsak da)
10 SELECT e.*
11 FROM employees AS e
12 JOIN departments AS d ON e.departmentid = d.departmentid
13 WHERE d.departmentname = 'IT';
14
15 --as komutunu kullandık tekrar tekrar employees ya da departments yazmamak için. Bunlar yerine e ve d kullandık.
```

employeeid	firstname	lastname	age	departmentid	salary	hiredate
1	John	Doe	30	1	55000.00	2020-01-15
2	Sara	Brown	28	1	52000.00	2021-04-25
3	Mark	Black	50	1	75000.00	2015-11-05

d. Maaşa Göre Sıralama

Çalışanların maaşlarını büyükten küçüğe doğru sıralamak için **desc (descending - azalan)** komutu kullanırız. Tam tersini isteseydi asc (ascending - artan) kullanacaktım.



e. Kolonları Birleştirme (Concatenation)

İlk yazmış olduğumda yalnızca full_name kolonunu getirdi. 2. Yazmış olduğum komutta diğer kolonların da isimlerini eklediğim için çalışanların bilgilerini de getirdi.

