

Davlatbek Ushurbakiyev is presenting : Course work part 7-8.

1 query about how to integrate data from two tables to one

SELECT name,phone,email FROM account

UNION

SELECT last_name,phone,company_of_entertaining FROM interviewer

The screenshot shows the pgAdmin 4 web interface in a browser window. The left sidebar displays the database structure for 'PostgreSQL 13', with 'Schemas (1)' expanded to show 'public'. The main pane is the 'Query Editor' for the 'jobSearch/postgres@PostgreSQL 13' connection. It contains a SQL query that creates a database, drops an existing one, and then performs a UNION of two SELECT statements. The 'Data Output' tab is active, showing a table with 12 rows of data.

```
1 -- Database: jobSearch
2
3 -- DROP DATABASE "jobSearch";
4
5 CREATE DATABASE "jobSearch"
6 WITH
7   OWNER = postgres
8   ENCODING = 'UTF8'
9   LC_COLLATE = 'Kazakh_Kazakhstan.utf8'
10  LC_CTYPE = 'Kazakh_Kazakhstan.utf8'
11  TABLESPACE = pg_default
12  CONNECTION LIMIT = -1;
13
14 SELECT *FROM account
15 SELECT *FROM interviewer
16
17 //First query about how to integrate data from two tables to one
18 SELECT name,phone,email FROM account
19 UNION
20 SELECT last_name,phone,company_of_entertaining FROM interviewer
```

acc_id	name	phone	email	district_of_search	worker_id
1	Davlat	34412	dushurbakiyev@gmail.com	Almaty	2
2	Akim	34412	dushurbakiyev@gmail.com	Almaty	3
3	Baha	34412	dushurbakiyev@gmail.com	Almaty	4
4	Alan	34412	dushurbakiyev@gmail.com	Almaty	5
5	Raf	34412	dushurbakiyev@gmail.com	Almaty	6
6	Aldik	34412	dushurbakiyev@gmail.com	Almaty	7
7	Karin	34412	dushurbakiyev@gmail.com	Almaty	8
8	Adjan	34412	dushurbakiyev@gmail.com	Almaty	9
9	Andish	34412	dushurbakiyev@gmail.com	Almaty	10
10	Arslino	34412	dushurbakiyev@gmail.com	Almaty	11
11	Nariman	34412	dushurbakiyev@gmail.com	Almaty	12
12	Nazirgum	34412	dushurbakiyev@gmail.com	Almaty	13

2 query about how to integrate data to table without data from definite table

SELECT name,phone,email FROM account

EXCEPT

SELECT last_name,phone,company_of_entertaining FROM interviewer

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays the database structure, including a 'jobSearch' database with a 'public' schema containing tables like 'account' and 'interviewer'. The main window shows a SQL query in the 'Query Editor' and its results in the 'Data Output' tab.

```
-- DROP DATABASE "jobSearch";
CREATE DATABASE "jobSearch"
WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'Kazakh_Kazakhstan.utf8'
  LC_CTYPE = 'Kazakh_Kazakhstan.utf8'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1;

SELECT *FROM account
SELECT *FROM interviewer

//First query about how to integrate data from two tables to one
SELECT name,phone,email FROM account
UNION
SELECT last_name,phone,company_of_entertaining FROM interviewer

//2 query about how to integrate data to table without data from definite table
SELECT name,phone,email FROM account
EXCEPT
SELECT last_name,phone,company_of_entertaining FROM interviewer
```

name	phone	email
Alan	34412	dushurbakiev@gmail.com
Raf	34412	dushurbakiev@gmail.com
Aldik	34412	dushurbakiev@gmail.com
Karin	34412	dushurbakiev@gmail.com
Aldun	34412	dushurbakiev@gmail.com
Andun	34412	dushurbakiev@gmail.com
Anduno	34412	dushurbakiev@gmail.com
Nariman	34412	dushurbakiev@gmail.com
Nisigum	34412	dushurbakiev@gmail.com
Zulfra	34412	dushurbakiev@gmail.com
Dilnaz	34412	dushurbakiev@gmail.com
Danik	34412	dushurbakiev@gmail.com

3 query about how to show the correlate data between bind tables

SELECT resume.sex,account.name,account.phone

FROM account

INNER JOIN resume

ON account.worker_id = resume.worker_id

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Servers' tree is expanded to show the 'public' schema. The 'Query Editor' window displays a SQL query with three parts: a connection limit, a UNION query, and an INNER JOIN query. The 'Data Output' tab shows the results of the INNER JOIN query, which correlates data between the 'account' and 'resume' tables.

```
12 CONNECTION LIMIT = -1;
13
14 SELECT *FROM account
15 SELECT *FROM interviewer
16 SELECT *FROM resume
17
18 //first query about how to integrate data from two tables to one
19 SELECT name,phone,email FROM account
20
21 UNION
22 SELECT last_name,phone,company_of_entertaining FROM interviewer
23
24 //2 query about how to integrate data to table without data from definite table
25 SELECT name,phone,email FROM account
26 EXCEPT
27 SELECT last_name,phone,company_of_entertaining FROM interviewer
28
29 //3 query about how to show the correlate data between bind tables
30
31
32 SELECT resume.sex,account.name,account.phone
33 FROM account
34 INNER JOIN resume
35 ON account.worker_id = resume.worker_id
36
```

sex	name	phone
man	Devlat	34412
man	Akim	34412
man	Eara	34412
man	Alan	34412
man	Raf	34412
man	Aldik	34412
man	Karim	34412
man	Alduin	34412
man	Anduin	34412
man	Arduno	34412
man	Nariman	34412
man	Nazugum	34412

4 query about how integrate data for each row by crossing them

SELECT resume.data_of_birth,account.name,account.phone

FROM account

CROSS JOIN resume

The screenshot shows the pgAdmin 4 web interface in a browser window. The left sidebar displays the database structure for 'PostgreSQL 13', including databases like 'UniversityForLab8' and 'jobSearch', and various system catalogs. The main pane is the 'Query Editor' for the 'jobSearch/postgres@PostgreSQL 13' connection. It contains a SQL script with four queries. The first three queries are commented out or partially shown. The fourth query, starting at line 40, is executed and its results are displayed in the 'Data Output' tab. The results table has three columns: 'data_of_birth', 'name', and 'phone'. It contains 12 rows of data, each representing a person's birth date, name, and phone number.

```
20 SELECT name,phone,email FROM account
21 UNION
22 SELECT last_name,phone,company_of_entertaining FROM interviewer
23
24 //2 query about how to integrate data to table without data from definite table
25 SELECT name,phone,email FROM account
26 EXCEPT
27 SELECT last_name,phone,company_of_entertaining FROM interviewer
28
29 //3 query about how to show the correlate data between bind tables
30
31
32 SELECT resume.sex,account.name,account.phone
33 FROM account
34 INNER JOIN resume
35 ON account.worker_id = resume.worker_id
36
37
38 //4 query about how integrate data for each row by crossing then
39
40 SELECT resume.data_of_birth,account.name,account.phone
41 FROM account
42 CROSS JOIN resume
43
44
```

	data_of_birth	name	phone
	date	character varying (10)	integer
1	2000-03-19	Devlet	34412
2	2000-03-19	Akim	34412
3	2000-03-19	Bapa	34412
4	2000-03-19	Alm	34412
5	2000-03-19	Baf	34412
6	2000-03-19	Aldik	34412
7	2000-03-19	Ketim	34412
8	2000-03-19	Aldun	34412
9	2000-03-19	Andun	34412
10	2000-03-19	Andun	34412
11	2000-03-19	Nariman	34412
12	2000-03-19	Nazugum	34412

//5 query about how to use Aliasing and also group data with help of it

```
SELECT c.comp_name,c.stage_on_market, i.last_name
```

```
FROM company c,interviewer i
```

```
WHERE c.comp_id = i.comp_id
```

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays the database structure, with the 'company' table selected under the 'public' schema. The main pane shows the 'Query Editor' with the following SQL code:

```
33
34
35 SELECT resume.sex,account.name,account.phone
36 FROM account
37 INNER JOIN resume
38 ON account.worker_id = resume.worker_id
39
40
41 //4 query about how integrate data for each row by crossing then
42
43 SELECT resume.data_of_birth,account.name,account.phone
44 FROM account
45 CROSS JOIN resume
46
47 //5 query about how to use Aliasing and also group data with help of it
48 SELECT c.comp_name,c.stage_on_market, i.last_name
49 FROM company c,interviewer i
50 WHERE c.comp_id = i.comp_id
51
52
53
54
55
56
57
```

Below the query editor, the 'Data Output' tab is active, displaying the results of the query in a table:

comp_name	stage_on_market	last_name
facebook	14	Devlet
amazon	25	Akim
prime	10	Baha
linux	24	Alan
Microsoft	20	Raf
Huawei	21	Aldik
Apple	26	Karim
Samsung	24	Aldun
Helios	10	Andun
Adiant	5	Arbuno
Sony	29	Nanman
Vtex	24	Nazugun

//6 query is about how to manage data according to different attributes

```
SELECT comp_name
```

```
FROM company
```

```
WHERE stage_on_market BETWEEN 10 and 30
```

The screenshot shows the pgAdmin 4 web interface in a browser window. The left sidebar displays the database structure, with the 'company' table under the 'public' schema selected. The main pane shows a SQL query in the 'Query Editor' tab. The query is as follows:

```
37 INNER JOIN resume
38 ON account.worker_id = resume.worker_id
39
40
41 //4 query about how integrate data for each row by crossing them
42
43 SELECT resume.data_of_birth,account.name,account.phone
44 FROM account
45 CROSS JOIN resume
46
47
48 //5 query about how to use Aliasing and also group data with help of it
49 SELECT c.comp_name,c.stage_on_market, t.last_name
50 FROM company c,interviewer i
51 WHERE c.comp_id = i.comp_id
52
53 //6 query is about how to manage data according to different attributes
54
55 SELECT comp_name
56 FROM company
57 WHERE stage_on_market BETWEEN 10 and 30
58
59
60
61
```

Below the query editor, the 'Data Output' tab is active, displaying a table with 12 rows of results:

comp_name
character_varying(10)
0000LE
facebook
amazon
prime
linux
Microsoft
Huawei
Apple
Samsung
Hellon
Sony
Vtex

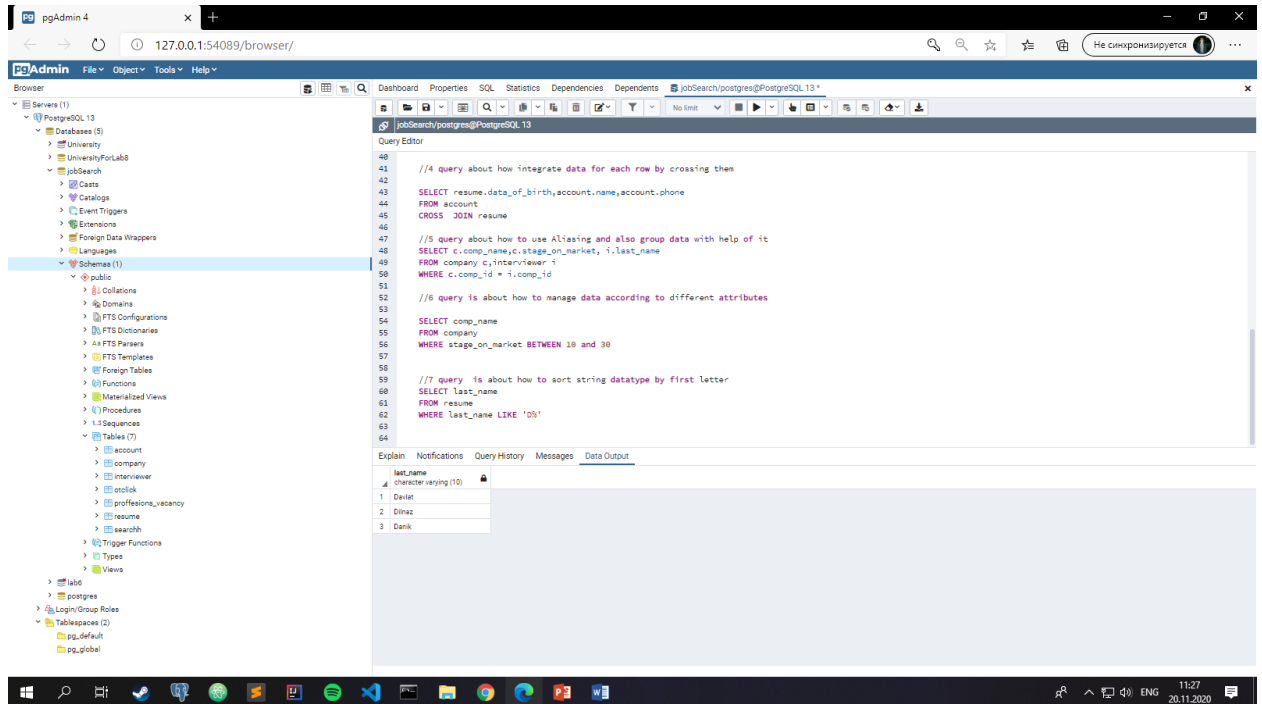
The Windows taskbar at the bottom shows the system clock as 11:24 on 20.11.2020.

//7 query is about how to sort string datatype by first letter

```
SELECT last_name
```

```
FROM resume
```

```
WHERE last_name LIKE 'D%'
```



//8 query is about how to change datatype from one to another

SELECT CAST(stage_on_market AS VARCHAR(20))

FROM company

The screenshot shows the pgAdmin 4 web interface in a browser window. The left sidebar displays the database structure for 'PostgreSQL 13', with the 'company' table under the 'public' schema selected. The main pane shows a SQL query in the 'Query Editor':

```
45
46
47 //5 query about how to use Aliasing and also group data with help of it
48 SELECT c.comp_name,c.stage_on_market, t.last_name
49 FROM company c,interviewer t
50 WHERE c.comp_id = t.comp_id
51
52 //6 query is about how to manage data according to different attributes
53 SELECT comp_name
54 FROM company
55 WHERE stage_on_market BETWEEN 10 and 30
56
57
58
59 //7 query is about how to sort string datatype by first letter
60 SELECT last_name
61 FROM resume
62 WHERE last_name LIKE 'DS'
63
64 //8 query is about how to change datatype from one to another
65 SELECT CAST(stage_on_market AS VARCHAR(20))
66 FROM company
67
68
69
70
```

Below the query editor, the 'Data Output' tab shows the results of the selected query (line 65). The results are displayed in a table with two columns: 'stage_on_market' and 'character varying (20)'. The data is as follows:

stage_on_market	character varying (20)
1	20
2	14
3	25
4	10
5	24
6	20
7	21
8	26
9	24
10	10
11	5
12	28

//9 query is about how to union attributes to one with saving data to it

SELECT active, deactive || ' ' || archive AS inactive

FROM otclick

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays the database structure, including Schemas (1), Tables (7), and Views. The main pane shows a SQL query in the Query Editor:

```
50 WHERE c.comp_id = i.comp_id
51
52 //8 query is about how to manage data according to different attributes
53
54 SELECT comp_name
55 FROM company
56 WHERE stage_on_market BETWEEN 10 and 30
57
58
59 //7 query is about how to sort string datatype by first letter
60 SELECT last_name
61 FROM resume
62 WHERE last_name LIKE 'DS'
63
64 //6 query is about how to change datatype from one to another
65 SELECT CAST(stage_on_market AS VARCHAR(20))
66 FROM company
67
68 //9 query is about how to union attributes to one with saving data to it
69
70 SELECT active, deactive || ' ' || archive AS inactive
71 FROM otclick
72
73
74
```

Below the query editor, the 'Data Output' tab is active, showing a table with 12 rows and 2 columns: 'active' and 'inactive'. The 'inactive' column contains the concatenated string of 'active' and 'deactive' values from the 'otclick' table.

active	inactive
1	true false false
2	true false false
3	true false false
4	true false false
5	true false false
6	true false false
7	true false false
8	true false false
9	true false false
10	true false false
11	true false false
12	true false false

//10 query is about how to show grouped data by different attributes

```
SELECT a.name, a.phone, a.email, a.district_of_search
```

```
FROM account a
```

```
GROUP BY a.name,a.phone,a.email,a.district_of_search
```

The screenshot shows the pgAdmin 4 web interface in a browser. The left sidebar displays the database structure for 'PostgreSQL 13', including 'Databases (5)', 'Schemas (1)', and 'Tables (7)'. The 'account' table is selected under the 'public' schema. The main pane shows the 'Query Editor' with a SQL query that has been executed. The query is as follows:

```
--  
--  
60 //7 query is about how to sort string datatype by first letter  
61 SELECT last_name  
62 FROM resume  
63 WHERE last_name LIKE 'D'  
64  
65 //8 query is about how to change datatype from one to another  
66 SELECT CAST(stage_on_market AS VARCHAR(20))  
67 FROM company  
68  
69 //9 query is about how to union attributes to one with saving data to it  
70 SELECT active, deactive || ' ' || archive AS inactive  
71 FROM otclick  
72  
73  
74  
75 //10 query is about how to show grouped data by different attributes  
76 SELECT a.name, a.phone, a.email, a.district_of_search  
77 FROM account a  
78 GROUP BY a.name,a.phone,a.email,a.district_of_search
```

Below the query editor, the 'Data Output' tab shows the results of the query. The results are displayed in a table with 4 columns: 'name', 'phone', 'email', and 'district_of_search'. There are 11 rows of data.

	name	phone	email	district_of_search
1	Arduino	34412	dushurbakiev@gmail.com	Almaty
2	Naiman	34412	dushurbakiev@gmail.com	Almaty
3	Andun	34412	dushurbakiev@gmail.com	Almaty
4	Dilnaz	34412	dushurbakiev@gmail.com	Almaty
5	Dariat	34412	dushurbakiev@gmail.com	Almaty
6	Alim	34412	dushurbakiev@gmail.com	Almaty
7	Zulfira	34412	dushurbakiev@gmail.com	Almaty
8	Natigum	34412	dushurbakiev@gmail.com	Almaty
9	Dank	34412	dushurbakiev@gmail.com	Almaty
10	Karim	34412	dushurbakiev@gmail.com	Almaty
11	Alan	34412	dushurbakiev@gmail.com	Almaty

At the bottom right, a green status bar indicates: 'Successfully run. Total query runtime: 71 msec. 15 rows affected.'

//11 query is about how to show ordered data in Descending order

SELECT a.name

FROM account a

ORDER BY a.name DESC

The screenshot shows the pgAdmin 4 web interface in a browser window. The left sidebar displays the database structure, with 'Schemas (1)' expanded under 'PostgreSQL 13'. The main pane is divided into two sections: a 'Query Editor' and a 'Data Output' section.

Query Editor: The SQL query is as follows:

```
60 SELECT last_name
61 FROM resume
62 WHERE last_name LIKE 'D%'
63
64 //8 query is about how to change datatype from one to another
65 SELECT CAST(stage_on_market AS VARCHAR(20))
66 FROM company
67
68 //9 query is about how to union attributes to one with saving data to it
69
70 SELECT active, deactive || ' ' || archive AS inactive
71 FROM otclick
72
73
74 //10 query is about how to show sorted data by different attributes
75 SELECT a.name, a.phone, a.email, a.district_of_search
76 FROM account a
77 GROUP BY a.name, a.phone, a.email, a.district_of_search
78
79 //11 query is about how to show ordered data in Descending order
80 SELECT a.name
81 FROM account a
82 ORDER BY a.name DESC
83
84
```

Data Output: The results of the query are displayed in a table with the following data:

name
Character varying (10)
1 Zulfira
2 Raf
3 Nazigum
4 Naziman
5 Katim
6 Dilnaz
7 Darlat
8 Dant
9 Baha
10 Ardano
11 Ardun
12 Aldun