MINISTRY OF EDUCATION AND SCIENCE OF THE REPUBLIC OF KAZAKHSTAN

INTERNATIONAL INFORMATION TECHNOLOGY UNIVERSITY

COMPUTER ENGINEERING AND INFORMATION SECURITY DEPARTMENT

COURSE WORK

Designing Databases. Introduction to SQL

Done by student: Ushurbakiyev Davlatbek

Group: ITSE-1909r

Almaty 2020

1 DESIGNING DATABASE

* 1. Subject area analysis (part 1)

--Font type Times New Roman 12, Interval 1,15

**Subject Area Analysis (job search site database)**

1. The DB is designed to hold information relating to/ about **job search**.
2. 7 entities:

Resume (worker\_id, last\_name,first\_name,phone,city\_of\_living,dateOfBirth,sex,citizen,stageOfWork);

Interviewer (interview\_id,last\_name,first\_name,phone,Company\_of\_entertaining);

Company (comp\_id, comp\_name, stage\_on\_market,activity,comp\_email,comp\_phone);

Proffesions\_Vacancy (prof\_id,prof\_name,prof\_condition,prof\_salary);

Account (acc\_id, , name,phone ,e-mail,socialMedia,district\_of\_search);

Otclik(ot\_id,active,deactive,archieve,all)

Search (search\_id, vacancies\_name,salary,stage\_of\_work,city,region,kind\_of\_company,Workschedule);

1. One Account can has many Resume also can has many Search.

One Interviewer can check many Resume.

One Company can has many Interviewers and many Proffesions\_Vacancies.

One Search can has many Account.

One Proffesion\_Vacancy can has many Search.

One Resume can has many Otclicks.

1. Account`s date of birth must be earlier than 2003.

Company`s activity should be legal.

Proffesions\_Vacancy name,prof\_condition,prof\_salary should not be an empty.

Otclick archieve should not be an empty.

1. The Database is designed for Job Search site workers who has access to personal data of Resumes/Company and admin panel.
2. Find Account by first name.

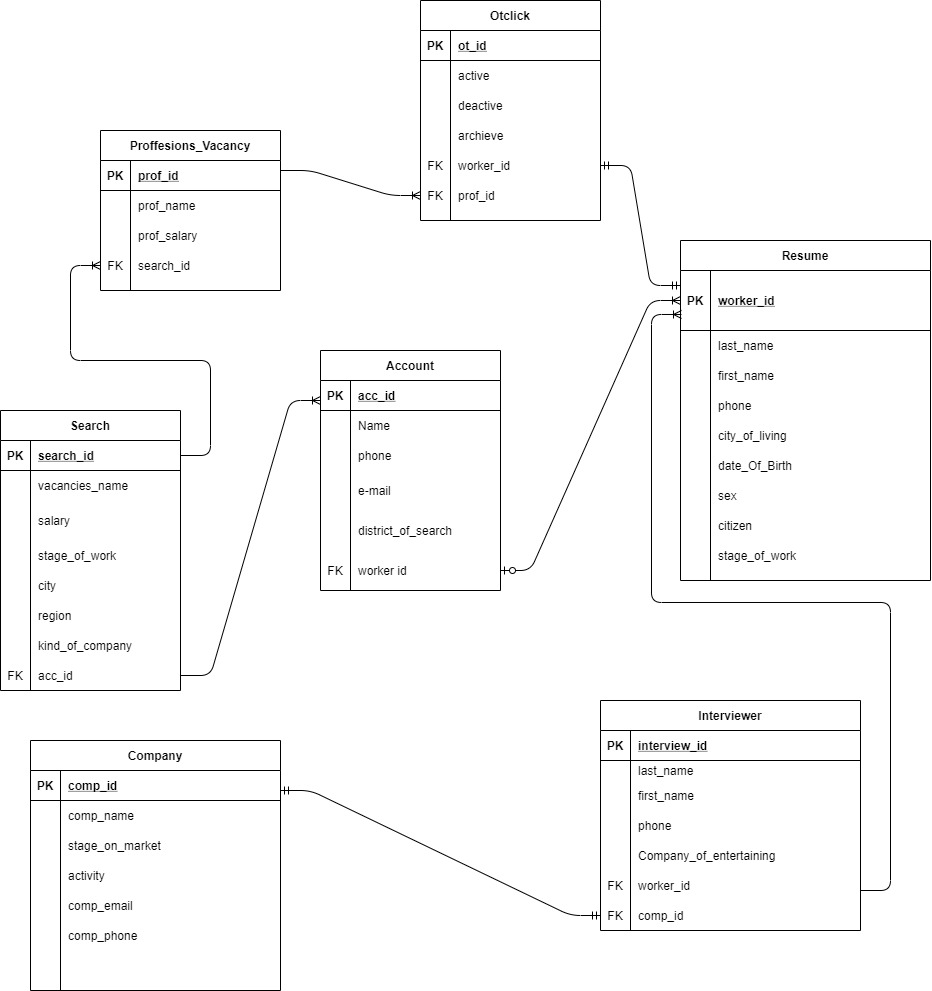
Sort Resumes by stageOfWork.

Find Company by its activity.

Sort Search by region of WorkSchedule.

Find only active Otclick.

* 1. ER diagram (part 2)



* 1. Logical design (part 3)

Dependencies:

One Account has many Resume.

One Search has many Accounts.

One Search has many Proffesion\_Vacancies.

One Proffesion\_Vacancy has many Otclicks.

One Otclick has one Resume

One Company has One Interviewer

One Interviewer has many Resume

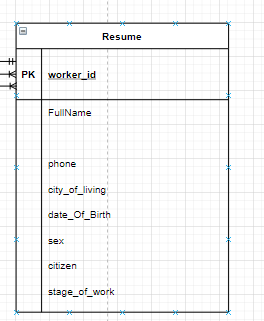
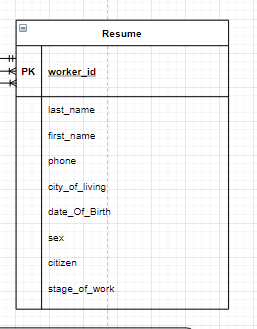
3)

To achieve 3NF comparing with the part 2 I changed relation Entity from table Search to Proffesions\_Vacancy, from “many to many” to “one to one”.

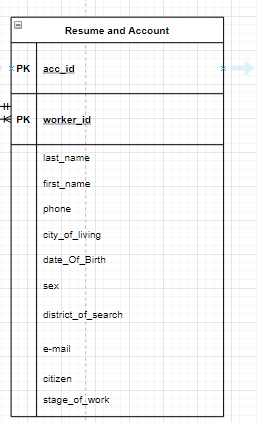
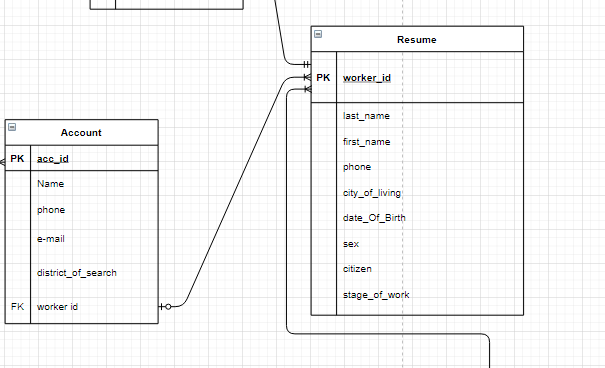
Also From table Search to table Account from many to many to one to one.

Examples :

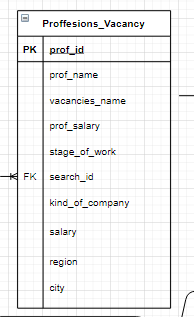
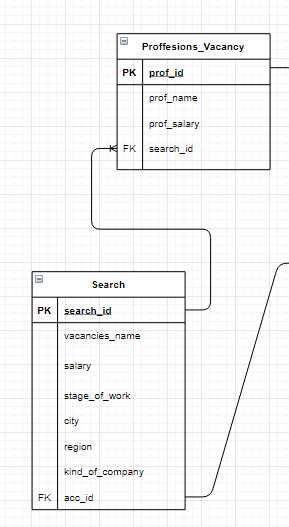
UNF -> 1NF

1NF -> 2NF

2NF -> 3NF

Account entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Field’s content | Field’s name | Type, length | Note(constraints) |
| Account ID | **acc\_id** | N(10) | PK |
| Name | Name | C(10) | NOT NULL |
| phone | phone | N(50) | NOT NULL |
| E-mail | e-mail | C(50) | NOT NULL |
| District of search | district\_of\_search | C(50) | NOT NULL |
| Worker id | worker id | N(10) | FK to Resume |

|  |  |  |  |
| --- | --- | --- | --- |
| Field’s content | Field’s name | Type, length | Note(constraints) |
| Worker id | **worker\_id** | N(10) | PK |
| Last name | last\_name | C(10) | NOT NULL |
| First name | first\_name | N(10) | NOT NULL |
| Phone | phone | N(10) | NOT NULL |
| City | city\_of\_living | C(10) | NOT NULL |
| Birth Date | date\_Of\_Birth | D | NOT NULL |
| Sex | sex | C(10) | NOT NULL |
| Citizen | citizen | B | NOT NULL |
| Stage | stage\_of\_work | N(3) | NOT NULL |

Resume entity:

Otclick entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Field’s content | Field’s name | Type, length | Note(constraints) |
| Otclick id | **ot\_id** | N(10) | PK |
| Active | active | B | NOT NULL |
| Deactive | deactive | B | NOT NULL |
| Acrchieve | archieve | B | NOT NULL |
| Worker id | worker\_id | N(10) | FK to Resume |
| Prof id | prof\_id | N(10) | FK to Proffesions |

Interviewer entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Field’s content | Field’s name | Type, length | Note(constraints) |
| Interviewer id | **interview\_id** | N(10) | PK |
| Last name | last\_name | C(50) | NOT NULL |
| First name | first\_name | C(50) | NOT NULL |
| Phone | phone | N(10) | NOT NULL |
| Company | Company\_of\_entertaining | C(10) | NOT NULL |
| Worker id | worker\_id | N(10) | FK to Resume |
| Comp id | comp\_id | N(10) | FK to Company |

Proffesions\_vacancy:

|  |  |  |  |
| --- | --- | --- | --- |
| Field’s content | Field’s name | Type, length | Note(constraints) |
| Proffesion id | **prof\_id** | N(10) | PK |
| Proffesion name | prof\_name | D | NOT NULL |
| Proffesion salary | prof\_salary | N(6) | NOT NULL |
| Search id | search\_id | N(10) | NOT NULL |

Search entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Field’s content | Field’s name | Type, length | Note(constraints) |
| Search id | **search\_id** | N(10) | PK |
| Vacancy name | vacancies\_name | N(10) | NOT NULL |
| Salary | salary | N(5) | NOT NULL |
| Stage | stage\_of\_work | N(4) | NOT NULL |
| City | city | C(10) | NOT NULL |
| Region | region | C(10) | NOT NULL |
| Company kind | kind\_of\_company | C(10) | NOT NULL |

Company entity:

|  |  |  |  |
| --- | --- | --- | --- |
| Field’s content | Field’s name | Type, length | Note(constraints) |
| Company id | **comp\_id** | N(10) | PK |
| Company Name | comp\_name | C(50) | NOT NULL |
| Stage | stage\_on\_market | N(10) | NOT NULL |
| Activity | activity | B | NOT NULL |
| Company email | comp\_email | C(60) | NOT NULL |
| Company phone | comp\_phone | N(10) | NOT NULL |

2 CREATING DATABASE

2.1 Creating tables (part 4, 5)

CREATE DATABASE "jobSearch"

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'Kazakh\_Kazakhstan.utf8'

LC\_CTYPE = 'Kazakh\_Kazakhstan.utf8'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1;

CREATE TABLE ACCOUNT(acc\_id int Primary key, Name Varchar(10),

phone int,

email Varchar(50),

district\_of\_search Varchar (50),

worker\_id int)

ALTER TABLE ACCOUNT ADD CONSTRAINT distfk FOREIGN KEY (worker\_id) REFERENCES Resume (worker\_id);

CREATE TABLE Resume (worker\_id int Primary key, last\_name Varchar(10),

first\_name Varchar(10),

phone int,

city\_of\_living Varchar(10),

data\_of\_birth Date,

sex Varchar(10),

citizen Boolean,

stage\_of\_work int)

CREATE TABLE Otclick(ot\_id int Primary key,

active Boolean,

deactive Boolean,

archieve Boolean,

worker\_id int,

prof\_id int)

ALTER TABLE Otclick ADD foreign key (worker\_id) REFERENCES Resume(worker\_id)

ALTER TABLE Otclick ADD foreign key (prof\_id) REFERENCES Proffesions\_vacancy(prof\_id)

CREATE TABLE Interviewer (interview\_id int Primary key,

last\_name Varchar(10),

first\_name Varchar(10),

phone int,

Company\_of\_entertaining Varchar(10),

worker\_id int,

comp\_id int)

ALTER TABLE Interviewer ADD foreign key (worker\_id) REFERENCES Resume(worker\_id)

ALTER TABLE Interviewer ADD foreign key (comp\_id) REFERENCES Company(comp\_id)

CREATE TABLE Proffesions\_vacancy(prof\_id int Primary key,

Name Varchar(10),

prof\_name Varchar(10),

prof\_salary int,

search\_id int)

ALTER TABLE Proffesions\_vacancy ADD foreign key (search\_id) REFERENCES Searchh(search\_id)

CREATE TABLE Searchh (search\_id int Primary key,

vacancies\_name Varchar (10),

salary int,

stage\_of\_work int,

city Varchar(10),

region Varchar (10),

kind\_of\_company Varchar (10))

CREATE TABLE Company (comp\_id int Primary key,

comp\_name Varchar (10),

stage\_on\_market int,

activity Boolean,

comp\_email Varchar (60),

comp\_phone int)

select \* from ACCOUNT

select \* from Resume

select \* from Otclick

select \* from Interviewer

select \* from Proffesions\_vacancy

select \* from Searchh

select \* from Company

insert into ACCOUNT

values(1,'Davlat',34412,'dushurbakiev@gmail.com','Almaty',2),

(2,'Akim',34412,'dushurbakiev@gmail.com','Almaty',3),

(3,'Baha',34412,'dushurbakiev@gmail.com','Almaty',4),

(4,'Alan',34412,'dushurbakiev@gmail.com','Almaty',5),

(5,'Raf',34412,'dushurbakiev@gmail.com','Almaty',6),

(6,'Aldik',34412,'dushurbakiev@gmail.com','Almaty',7),

(7,'Karim',34412,'dushurbakiev@gmail.com','Almaty',8),

(8,'Alduin',34412,'dushurbakiev@gmail.com','Almaty',9),

(9,'Anduin',34412,'dushurbakiev@gmail.com','Almaty',10),

(10,'Arduino',34412,'dushurbakiev@gmail.com','Almaty',11),

(11,'Nariman',34412,'dushurbakiev@gmail.com','Almaty',12),

(12,'Nazugum',34412,'dushurbakiev@gmail.com','Almaty',13),

(13,'Zulfira',34412,'dushurbakiev@gmail.com','Almaty',14),

(14,'Dilnaz',34412,'dushurbakiev@gmail.com','Almaty',15),

(15,'Danik',34412,'dushurbakiev@gmail.com','Almaty',1)

insert into Resume

values(1,'Davlat','adsd',23123,'Almaty','19.03.2000','man',true,2),

(2,'Akim','adsd',23123,'Almaty','19.03.2000','man',true,3),

(3,'Baha','adsd',23123,'Almaty','19.03.2000','man',true,4),

(4,'Alan','adsd',23123,'Almaty','19.03.2000','man',true,5),

(5,'Raf','adsd',23123,'Almaty','19.03.2000','man',true,6),

(6,'Aldik','adsd',23123,'Almaty','19.03.2000','man',true,7),

(7,'Karim','adsd',23123,'Almaty','19.03.2000','man',true,8),

(8,'Alduin','adsd',23123,'Almaty','19.03.2000','man',true,9),

(9,'Anduin','adsd',23123,'Almaty','19.03.2000','man',true,10),

(10,'Arduino','adsd',23123,'Almaty','19.03.2000','man',true,11),

(11,'Nariman','adsd',23123,'Almaty','19.03.2000','man',true,12),

(12,'Nazugum','adsd',23123,'Almaty','19.03.2000','man',true,13),

(13,'Zulfira','adsd',23123,'Almaty','19.03.2000','man',true,14),

(14,'Dilnaz','adsd',23123,'Almaty','19.03.2000','man',true,15),

(15,'Danik','adsd',23123,'Almaty','19.03.2000','man',true,1)

insert into Otclick

values(2,true,false,false,2,2),

(1,true,false,false,1,1),

(3,true,false,false,3,3),

(4,true,false,false,4,4),

(5,true,false,false,5,5),

(6,true,false,false,6,6),

(7,true,false,false,7,7),

(8,true,false,false,8,8),

(9,true,false,false,9,9),

(10,true,false,false,10,10),

(11,true,false,false,11,11),

(12,true,false,false,12,12),

(13,true,false,false,13,13),

(14,true,false,false,14,14),

(15,true,false,false,15,15)

insert into Interviewer

values(1,'Davlat','Davlat',34412,'dushu',2,2),

(2,'Akim','Davlat',34412,'dushu',3,3),

(3,'Baha','Davlat',34412,'dushu',4,4),

(4,'Alan','Davlat',34412,'dushu',5,5),

(5,'Raf','Davlat',34412,'dushu',6,6),

(6,'Aldik','Davlat',34412,'dushu',7,7),

(7,'Karim','Davlat',34412,'dushu',8,8),

(8,'Alduin','Davlat',34412,'dushu',9,9),

(9,'Anduin','Davlat',34412,'dushu',10,10),

(10,'Arduino','Davlat',34412,'dushu',11,11),

(11,'Nariman','Davlat',34412,'dushu',12,12),

(12,'Nazugum','Davlat',34412,'dushu',13,13),

(13,'Zulfira','Davlat',34412,'dushu',14,14),

(14,'Dilnaz','Davlat',34412,'dushu',15,15),

(15,'Danik','Davlat',34412,'dushu',1,1)

insert into Proffesions\_vacancy

values(1,'Davlat','DXD',2,2),

(2,'Akim','DXD',3,3),

(3,'Baha','DXD',4,4),

(4,'Alan','DXD',5,5),

(5,'Raf','DXD',6,6),

(6,'Aldik','DXD',7,7),

(7,'Karim','DXD',8,8),

(8,'Alduin','DXD',9,9),

(9,'Anduin','DXD',10,10),

(10,'Arduino','DXD',11,11),

(11,'Nariman','DXD',12,12),

(12,'Nazugum','DXD',13,13),

(13,'Zulfira','DXD',14,14),

(14,'Dilnaz','DXD',15,15),

(15,'Danik','DXD',1,1)

insert into Searchh

values(1,'CSSE',34412,10,'Almaty','Asia','IT'),

(2,'CSSI',34412,10,'Almaty','Asia','IT'),

(3,'CSS',34412,10,'Almaty','Asia','IT'),

(4,'CS',34412,10,'Almaty','Asia','IT'),

(5,'ML',34412,10,'Almaty','Asia','IT'),

(6,'AM',34412,10,'Almaty','Asia','IT'),

(7,'LA',34412,10,'Almaty','Asia','IT'),

(8,'RR',34412,10,'Almaty','Asia','IT'),

(9,'RP',34412,10,'Almaty','Asia','IT'),

(10,'CSSE',34412,10,'Almaty','Asia','IT'),

(11,'CSSE',34412,10,'Almaty','Asia','IT'),

(12,'CSSE',34412,10,'Almaty','Asia','IT'),

(13,'CSSE',34412,10,'Almaty','Asia','IT'),

(14,'CSSE',34412,10,'Almaty','Asia','IT'),

(15,'CSSE',34412,10,'Almaty','Asia','IT')

insert into Company

values(1,'GOOGLE',20,true,'google.com',0001),

(2,'facebook',14,true,'facebook.com',0002),

(3,'amazon',25,true,'amazon.com',0003),

(4,'prime',10,true,'prime.com',0004),

(5,'linux',24,true,'linux.com',0005),

(6,'Microsoft',20,true,'Microsoft.com',0006),

(7,'Huawei',21,true,'Huawei.com',0007),

(8,'Apple',26,true,'Apple.com',0008),

(9,'Samsung',24,true,'Samsung.com',0009),

(10,'Helios',10,true,'Helios.com',00010),

(11,'Atlant',5,true,'Atlant.com',00011),

(12,'Sony',28,true,'Sony.com',00012),

(13,'Vitex',24,true,'Vitex.com',00013),

(14,'OMEGA',1,true,'OMEGA.com',00014),

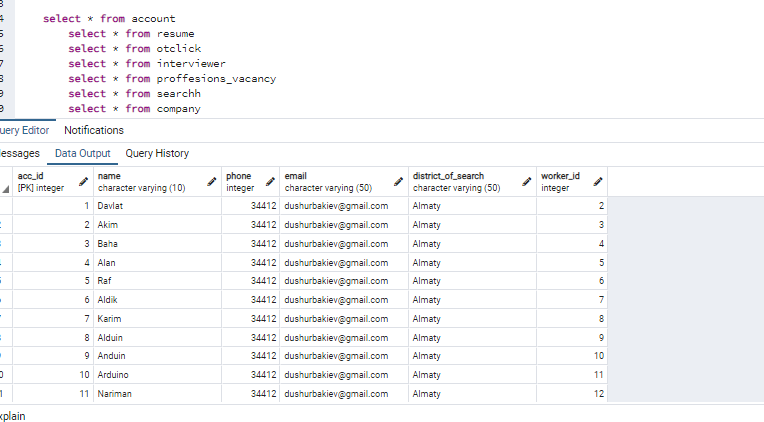
(15,'KtK',2,true,'KtK.com',00015)

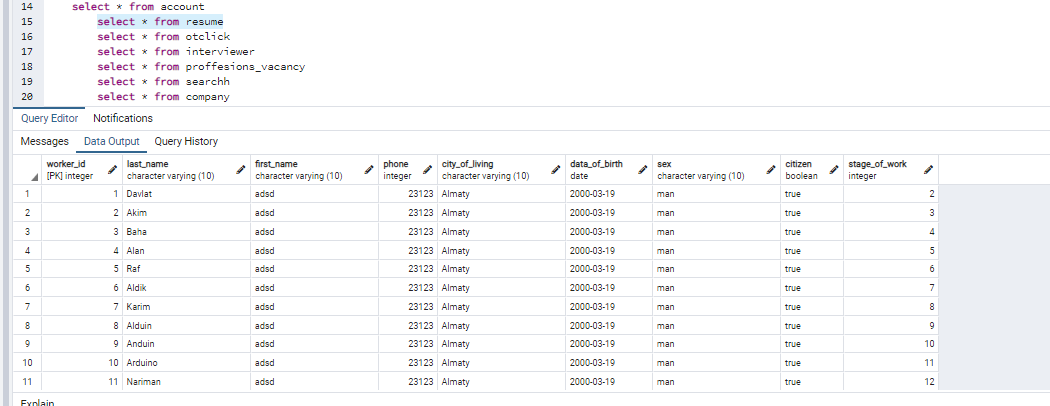
…

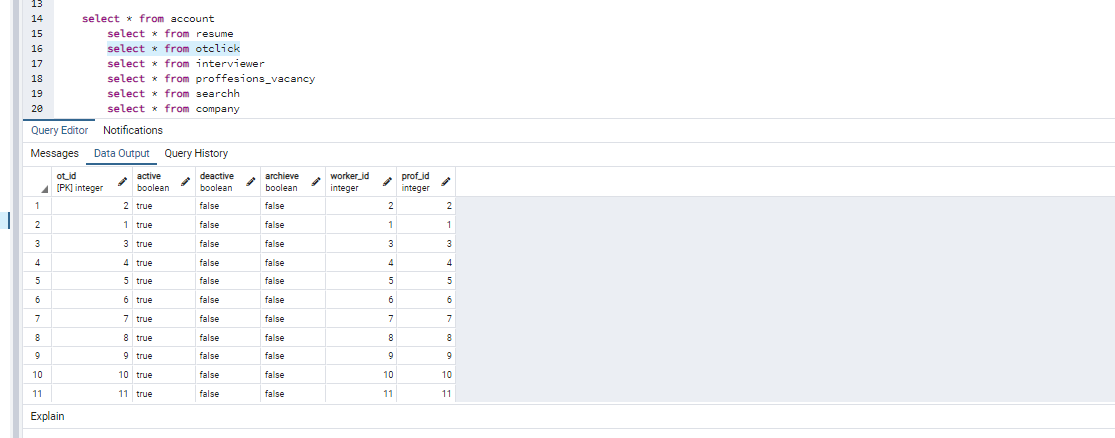
2.2 Populating tables (part 4, 5)

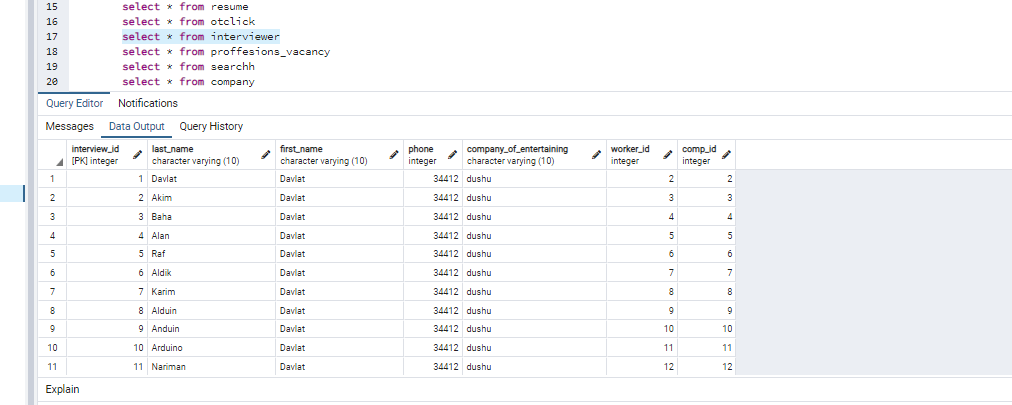
--Insert screenshots of tables with data

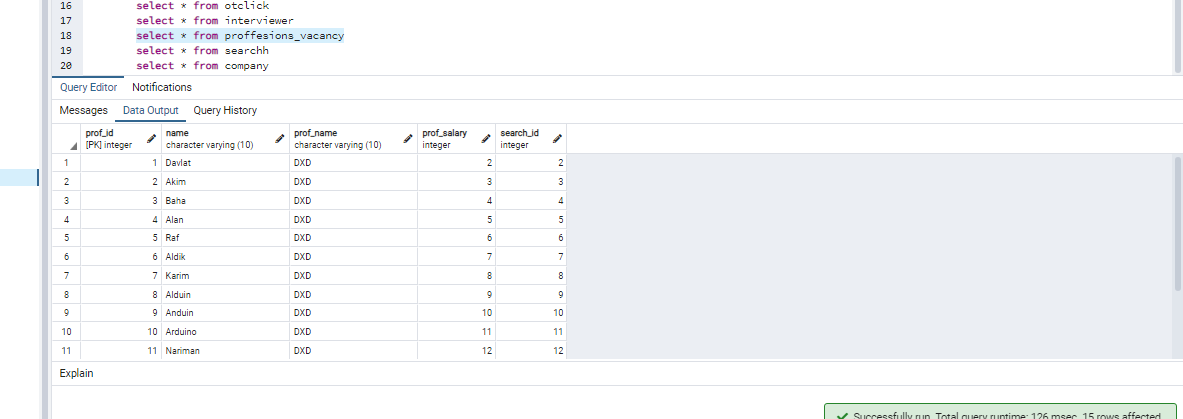
INSERT INTO emp VALUES (…);

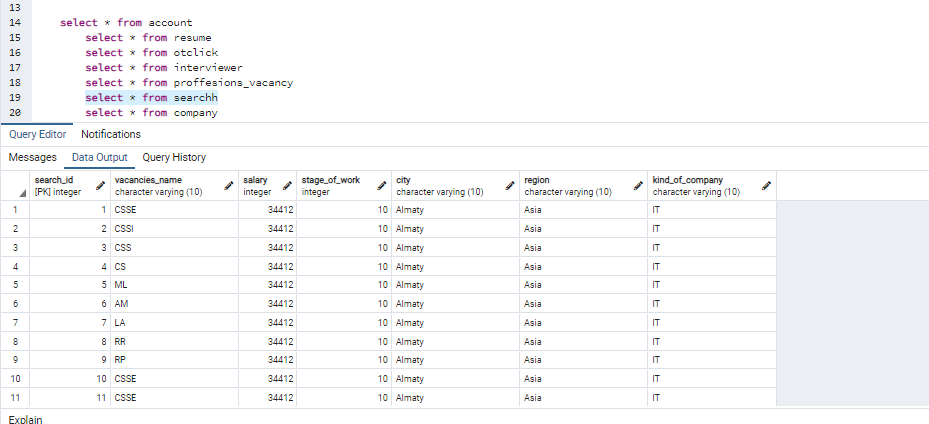


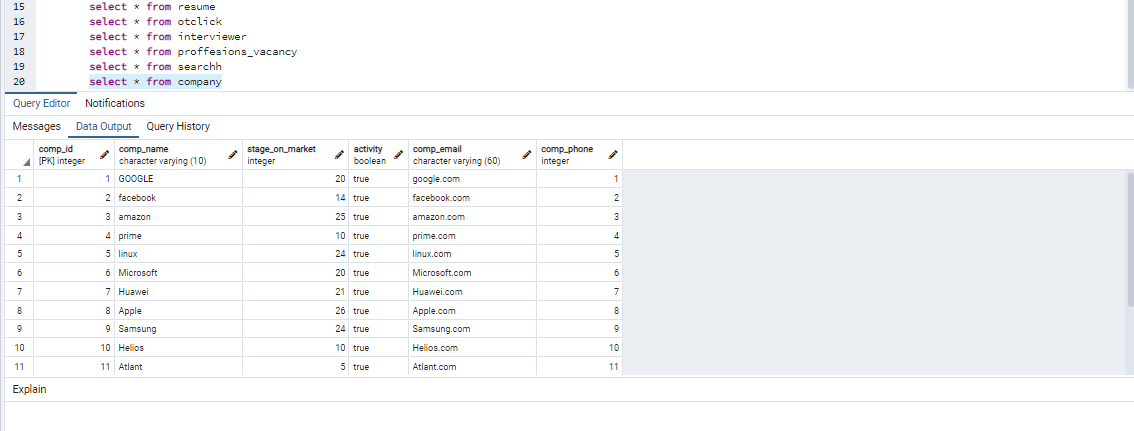












…

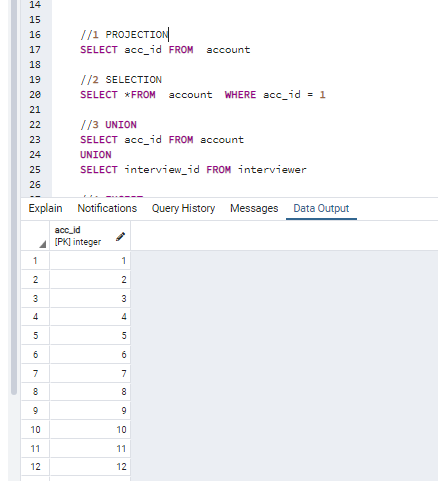
2.3 Relational algebra (part 6)

--Add a short description before each statement and the screenshot of the result after it

…

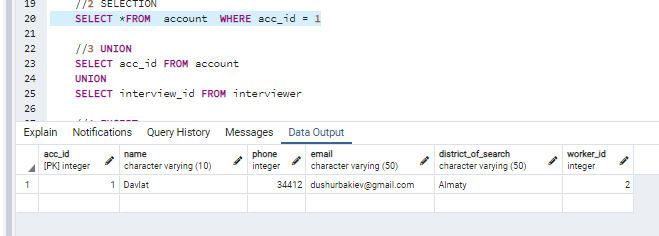
//1 PROJECTION

SELECT acc\_id FROM account



//2 SELECTION

SELECT \*FROM account WHERE acc\_id = 1

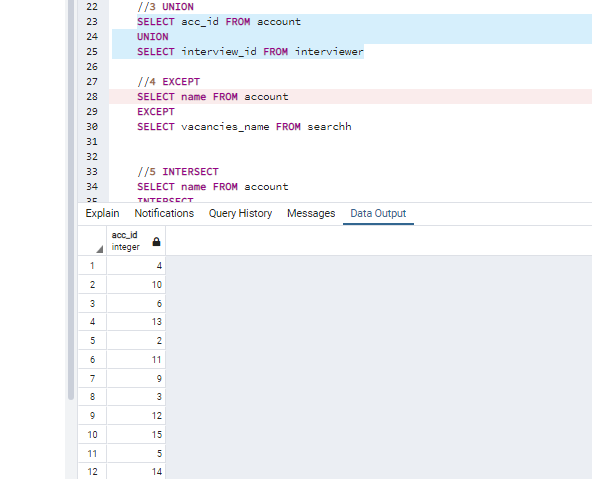


//3 UNION

SELECT acc\_id FROM account

UNION

SELECT interview\_id FROM interviewer

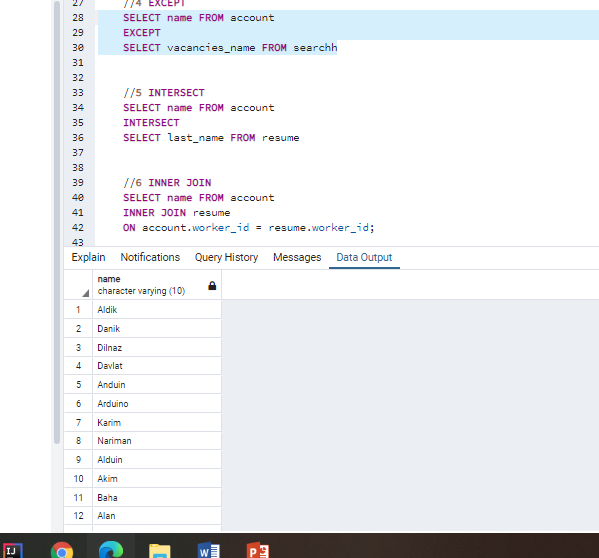


//4 EXCEPT

SELECT name FROM account

EXCEPT

SELECT vacancies\_name FROM searchh

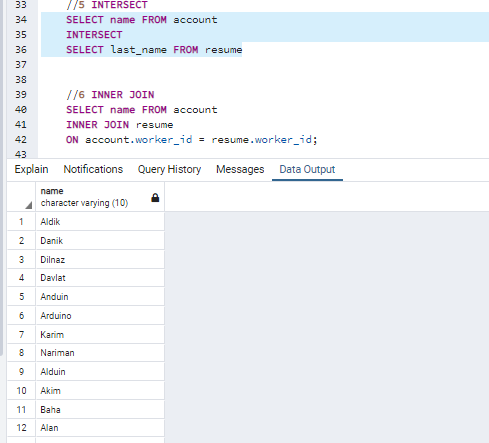


//5 INTERSECT

SELECT name FROM account

INTERSECT

SELECT last\_name FROM resume

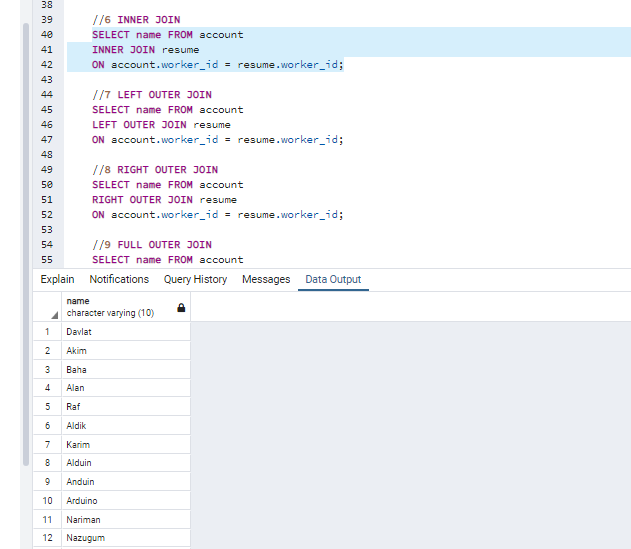


//6 INNER JOIN

SELECT name FROM account

INNER JOIN resume

ON account.worker\_id = resume.worker\_id;

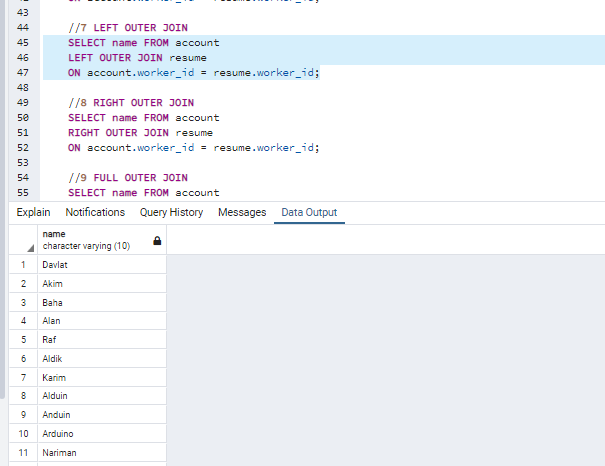


//7 LEFT OUTER JOIN

SELECT name FROM account

LEFT OUTER JOIN resume

ON account.worker\_id = resume.worker\_id;

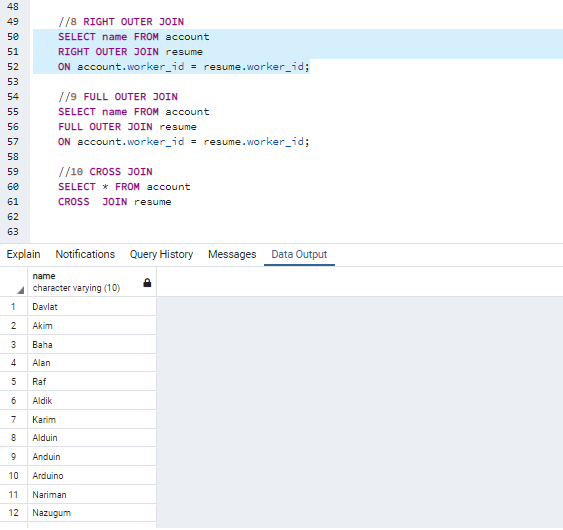


//8 RIGHT OUTER JOIN

SELECT name FROM account

RIGHT OUTER JOIN resume

ON account.worker\_id = resume.worker\_id;

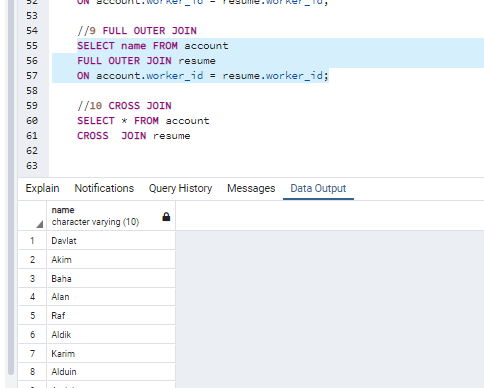


//9 FULL OUTER JOIN

SELECT name FROM account

FULL OUTER JOIN resume

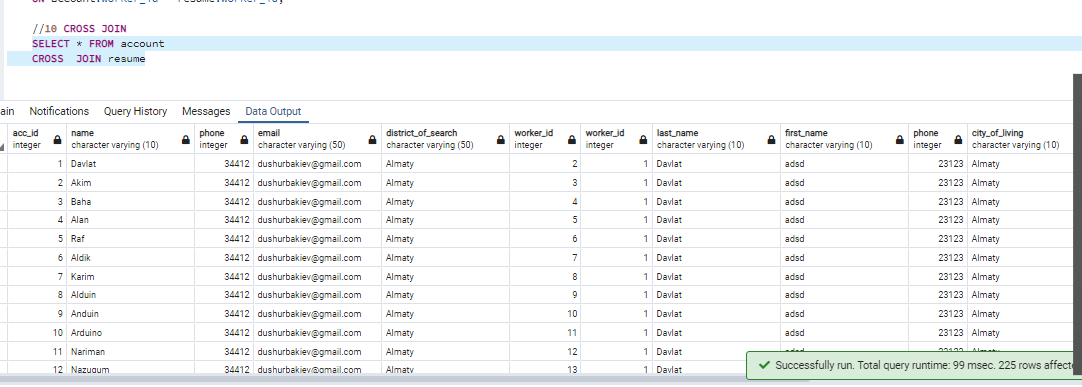
ON account.worker\_id = resume.worker\_id;



//10 CROSS JOIN

SELECT \* FROM account

CROSS JOIN resume



2.4 SQL queries (part 7, 8)

--Add a short description before each statement and the screenshot of the result after it

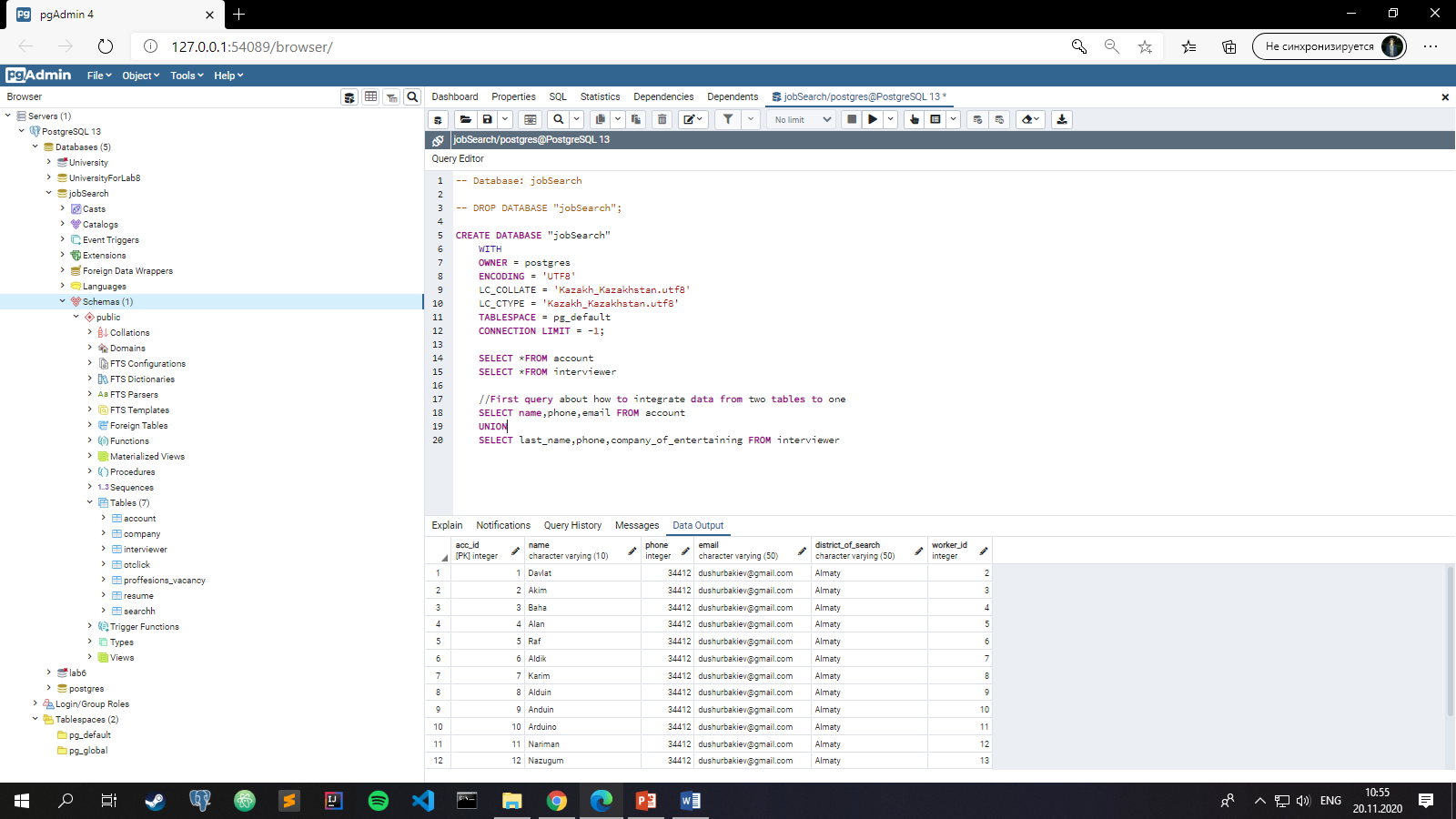
…

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

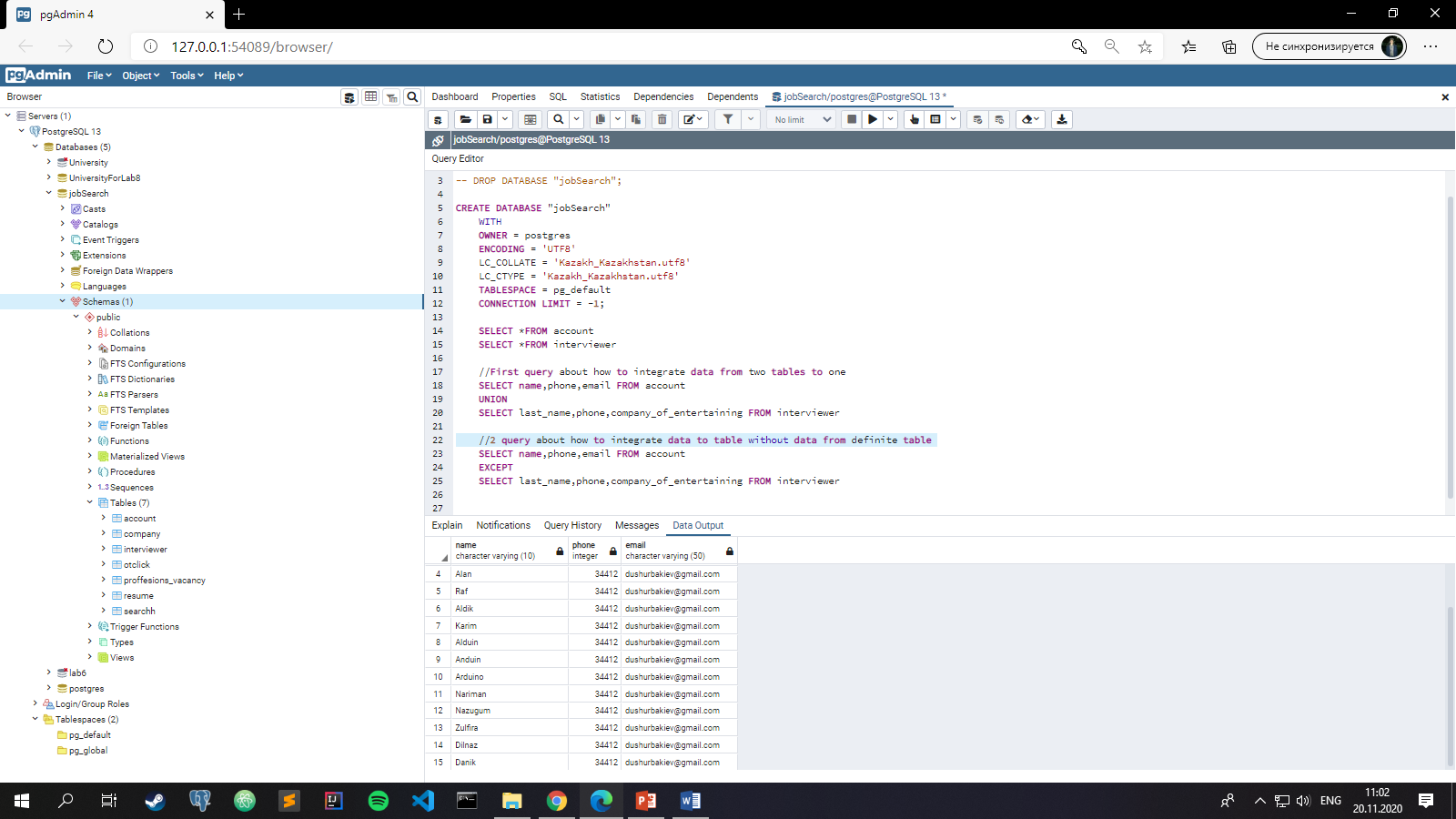


2 query about how to integrate data to table without data from definite table

SELECT name,phone,email FROM account

EXCEPT

SELECT last\_name,phone,company\_of\_entertaining FROM interviewer



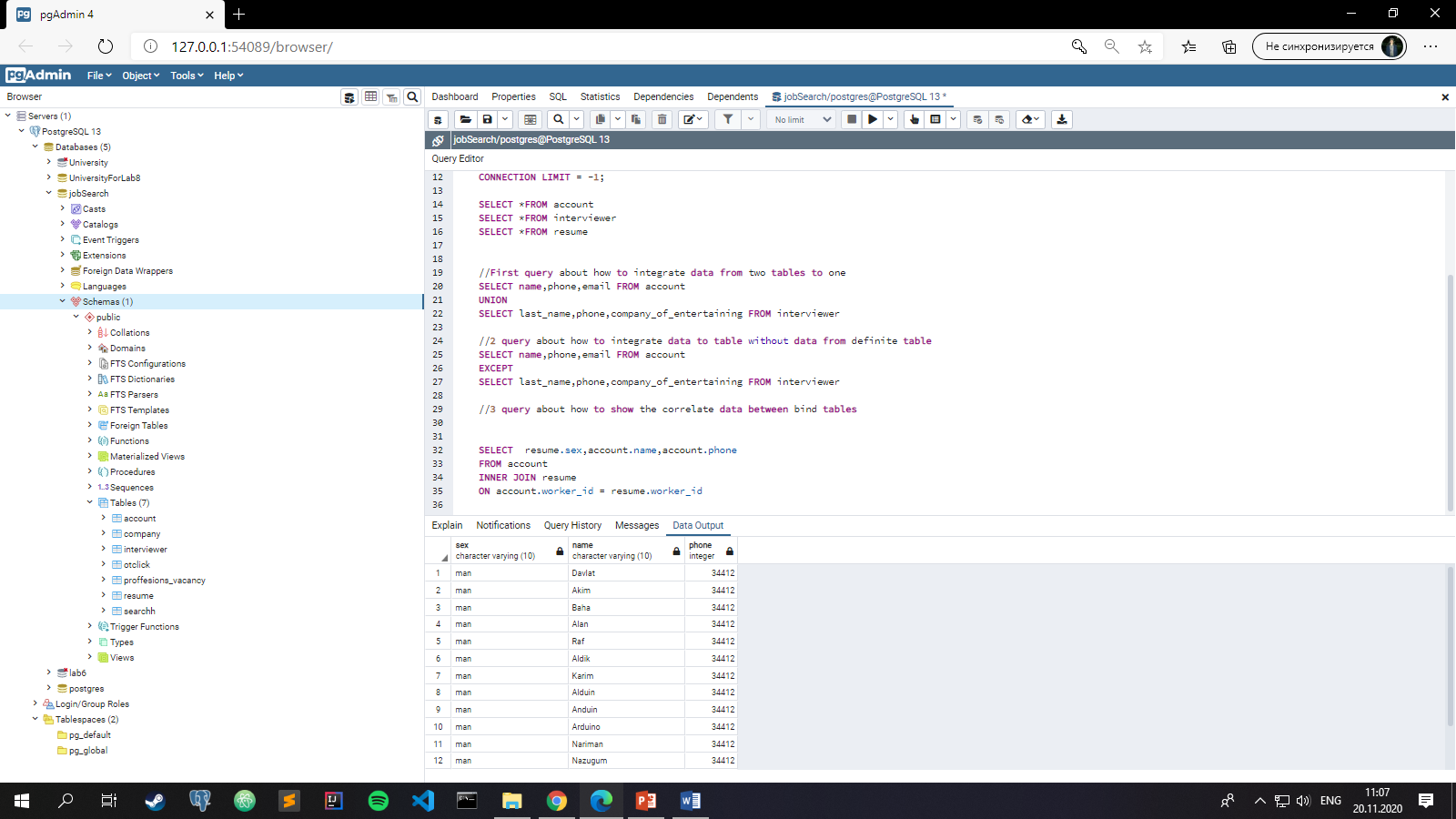
3 query about how to show the correlate data between bind tables

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

FROM account

INNER JOIN resume

ON account.worker\_id = resume.worker\_id

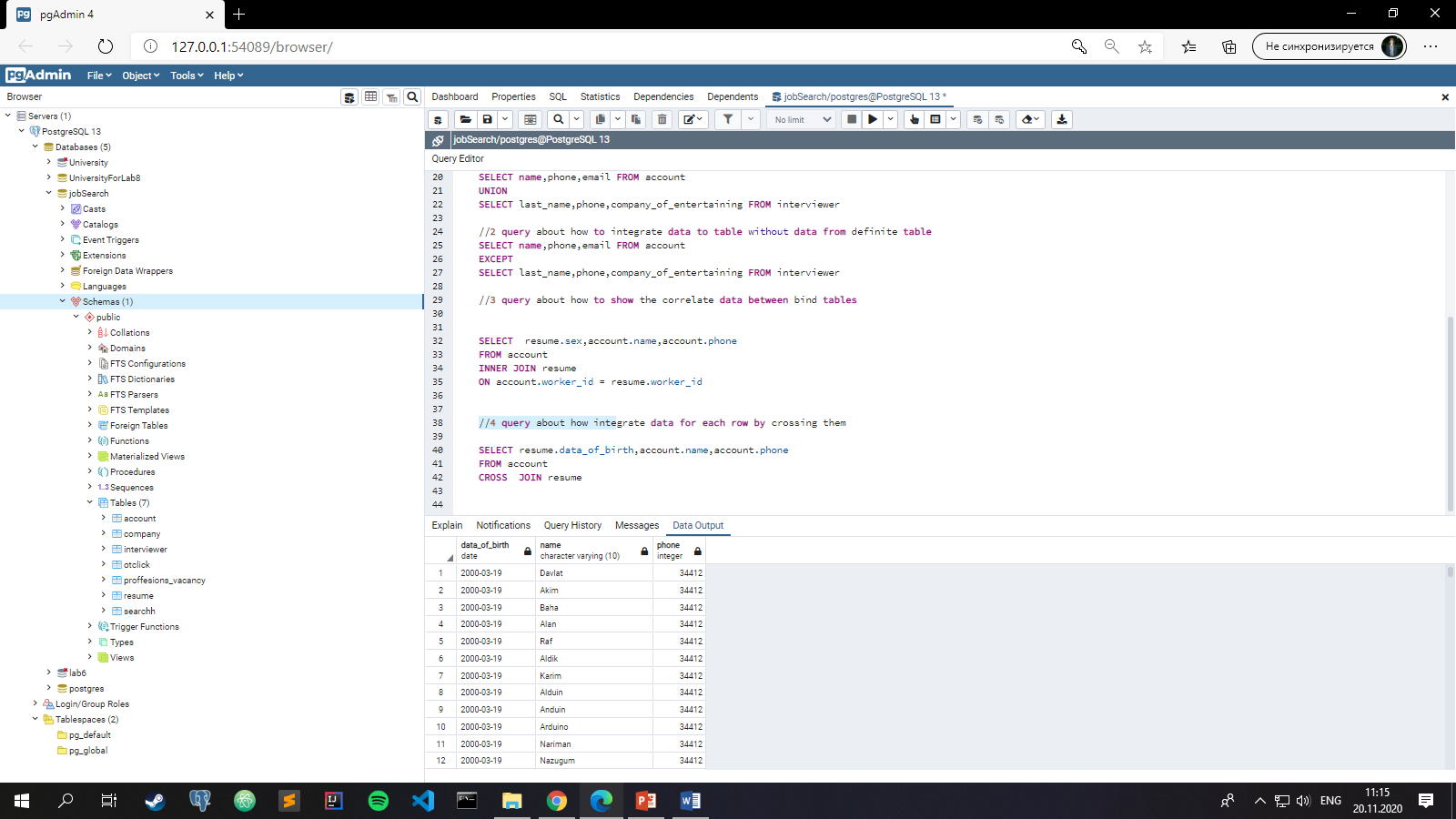


4 query about how integrate data for each row by crossing them

SELECT resume.data\_of\_birth,account.name,account.phone

FROM account

CROSS JOIN resume

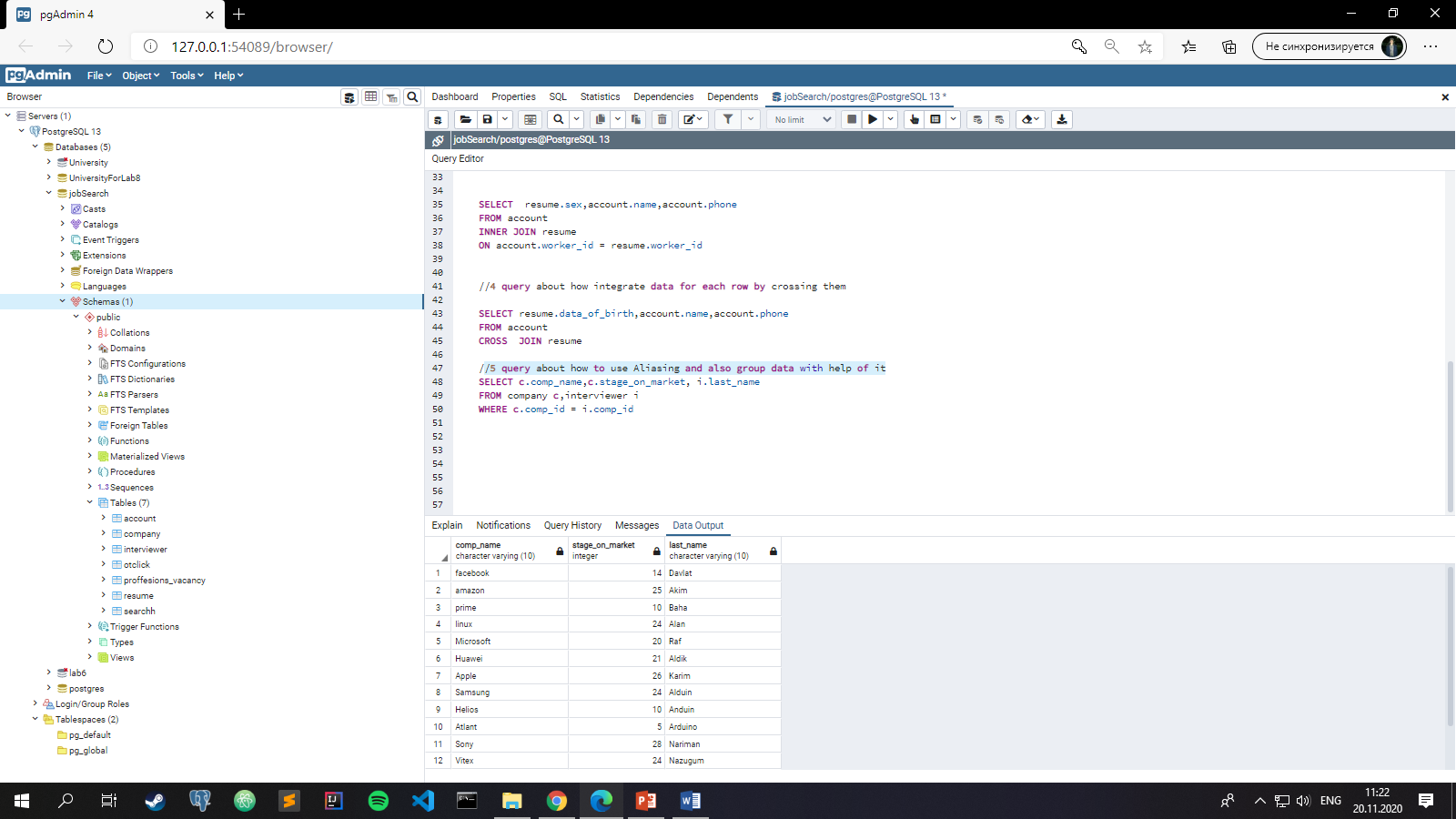


//5 query about how to use Aliasing and also group data with help of it

SELECT c.comp\_name,c.stage\_on\_market, i.last\_name

FROM company c,interviewer i

WHERE c.comp\_id = i.comp\_id

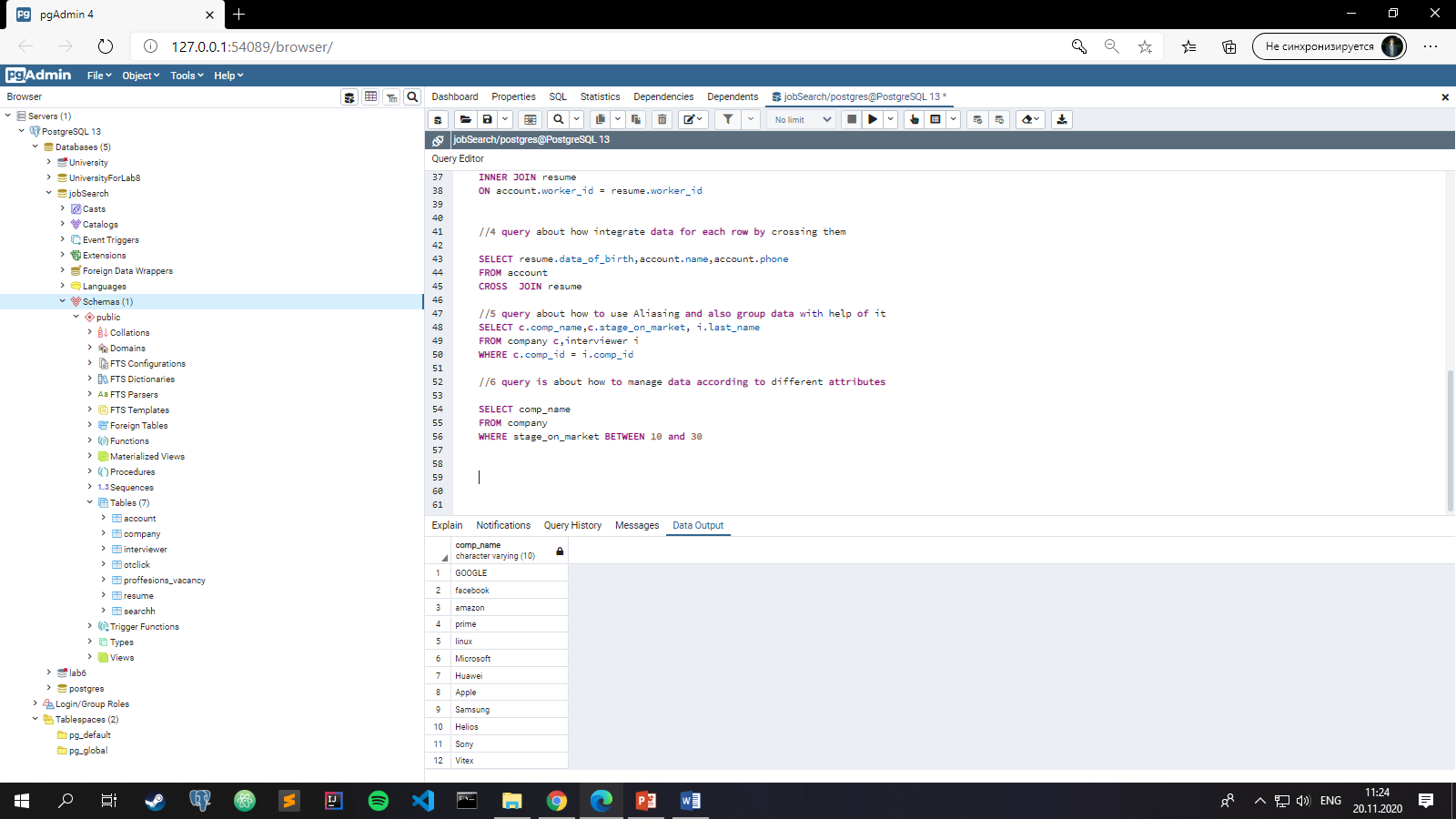


//6 query is about how to manage data according to different attributes

SELECT comp\_name

FROM company

WHERE stage\_on\_market BETWEEN 10 and 30

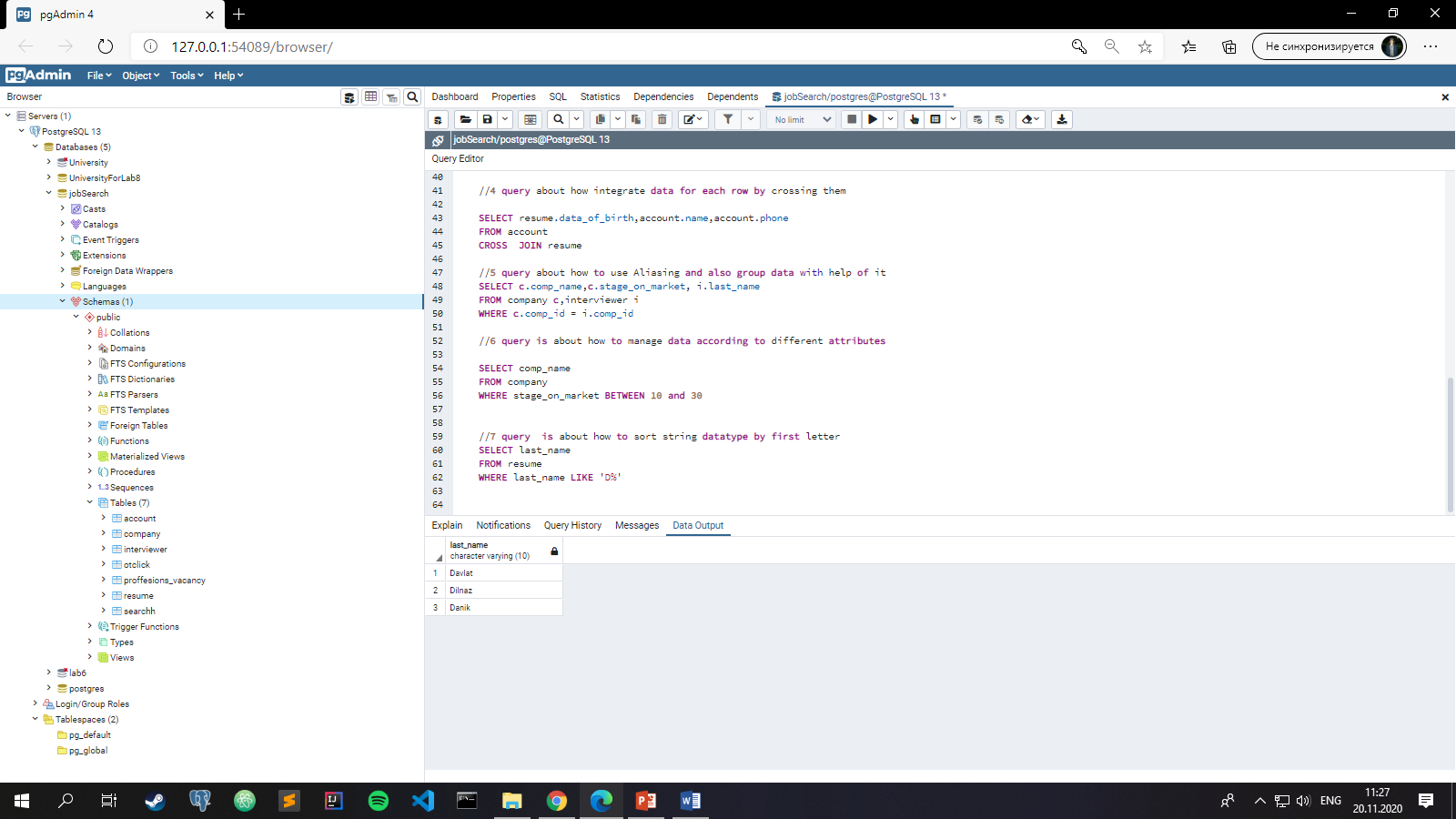


//7 query is about how to sort string datatype by first letter

SELECT last\_name

FROM resume

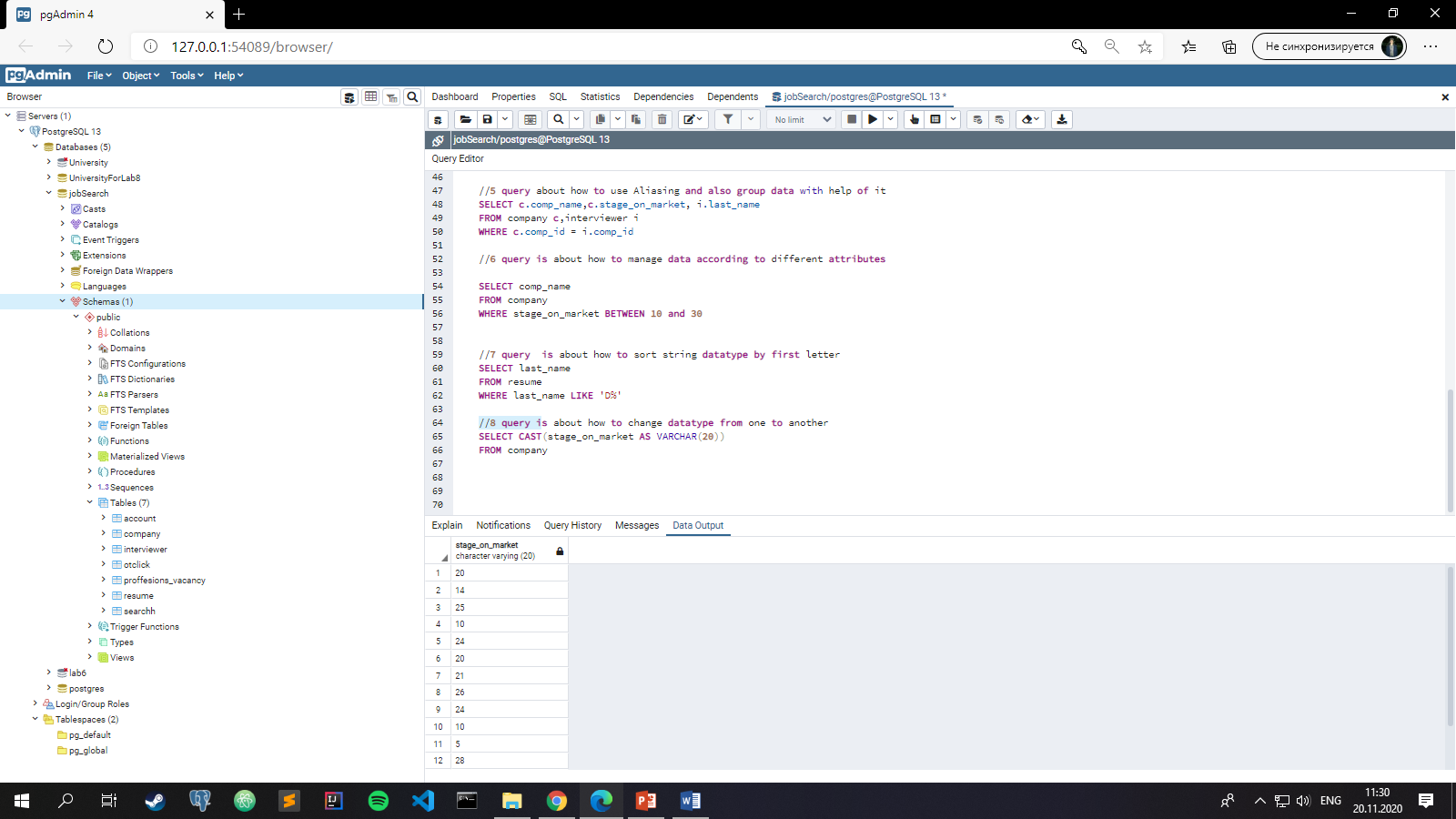
WHERE last\_name LIKE 'D%'



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

SELECT CAST(stage\_on\_market AS VARCHAR(20))

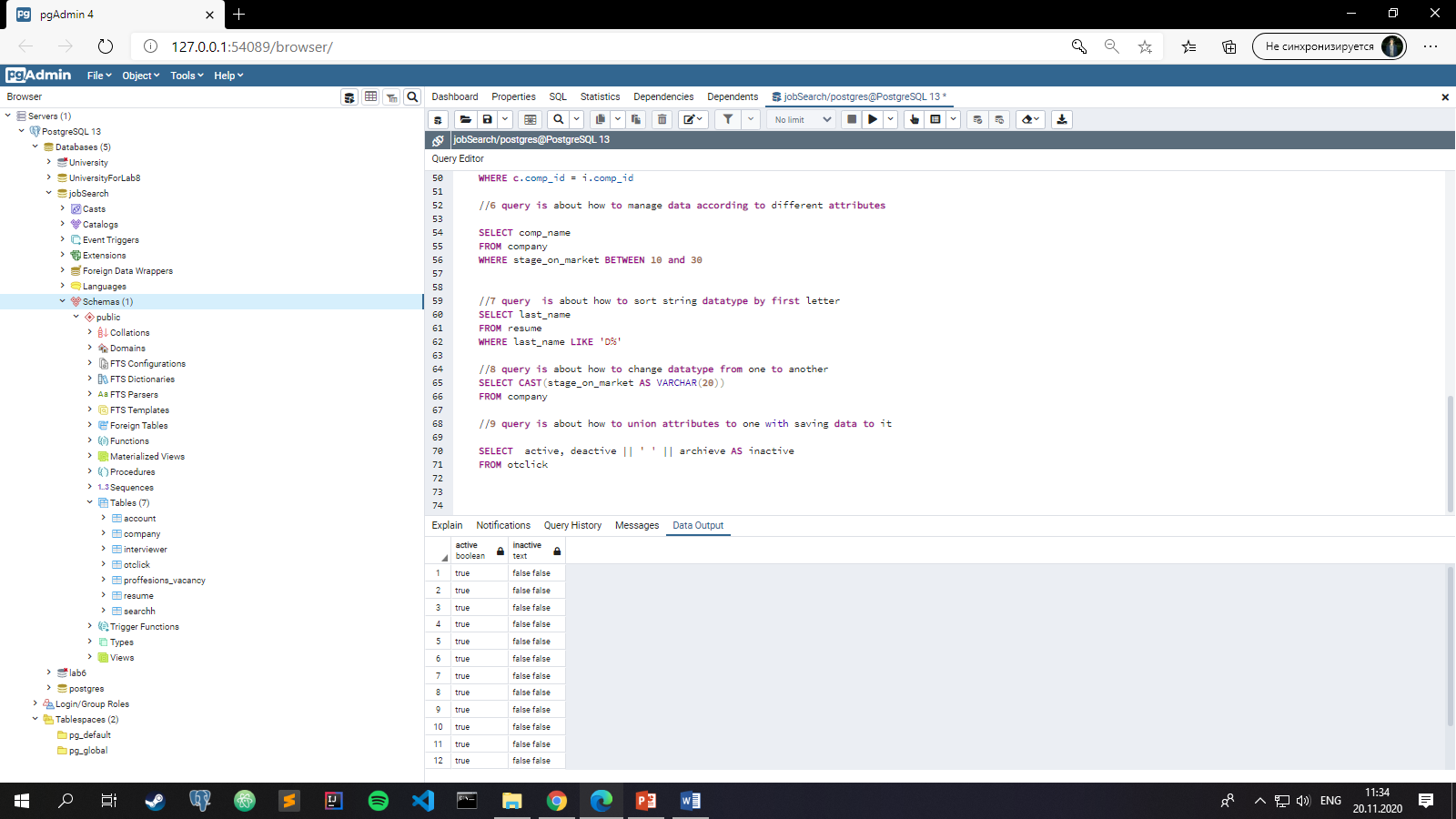
FROM company



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

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

FROM otclick

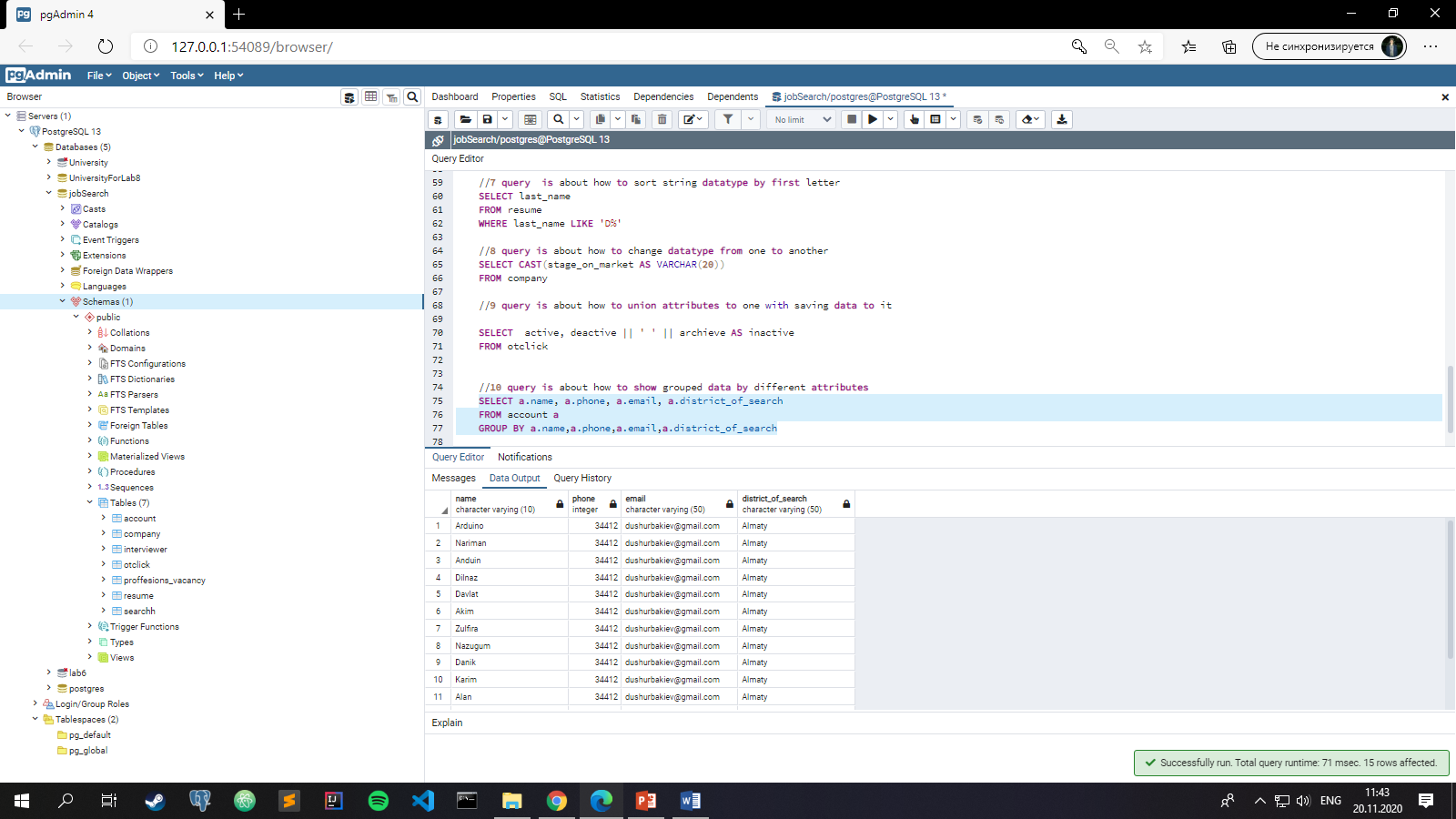


//10 query is about how to show grouped data by different attributes

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

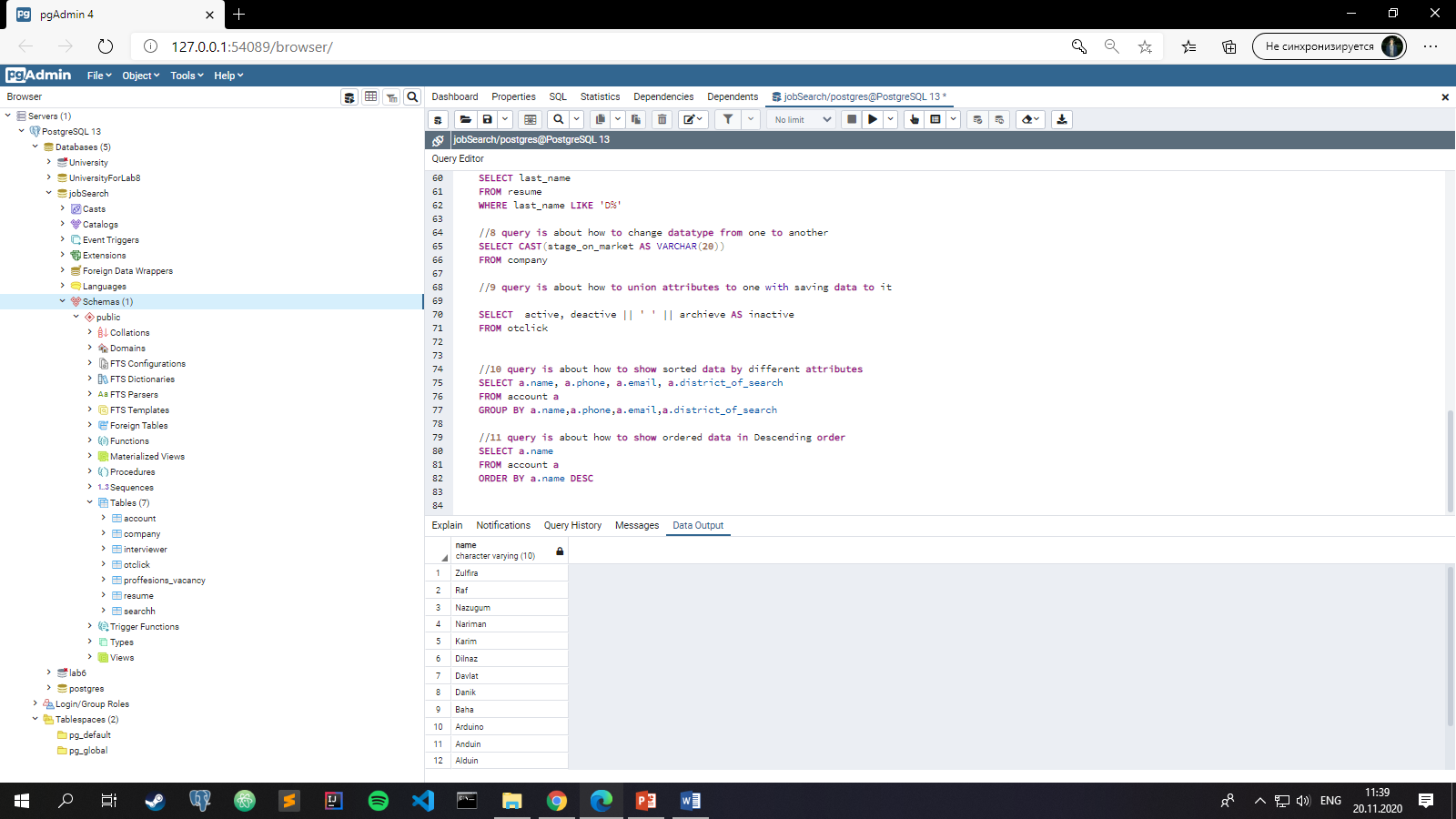


//11 query is about how to show ordered data in Descending order

SELECT a.name

FROM account a

ORDER BY a.name DESC



2.5 Nested queries (part 9)

--Add a short description before each statement and the screenshot of the result after it

…

///1 Well, we can use some kind of ariphmetic operations with the attribute that we want to use for achieve some definite result like this:

SELECT last\_name

FROM resume

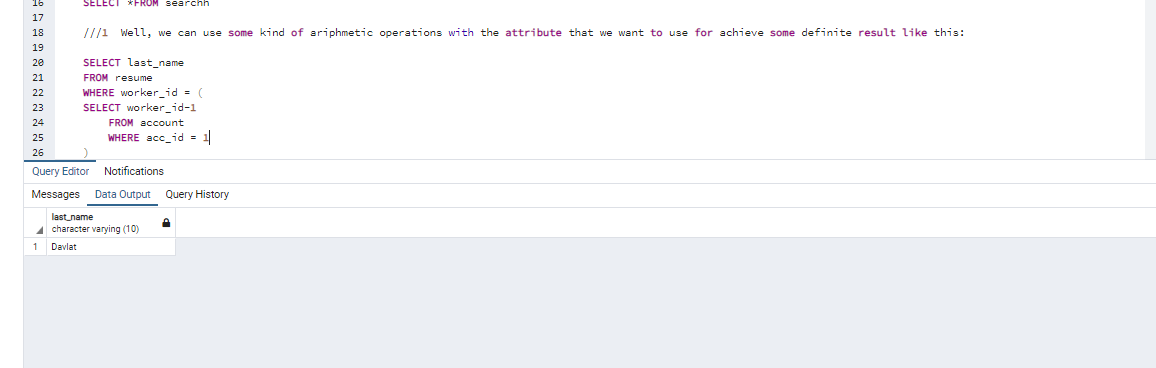
WHERE worker\_id = (

SELECT worker\_id-1

FROM account

WHERE acc\_id = 1

)



2) For example we need to find all names and sex of those people whose stage of work less than 5 to make some research work.

SELECT a.name, r.sex

FROM account a , resume r

WHERE a.worker\_id = r.worker\_id

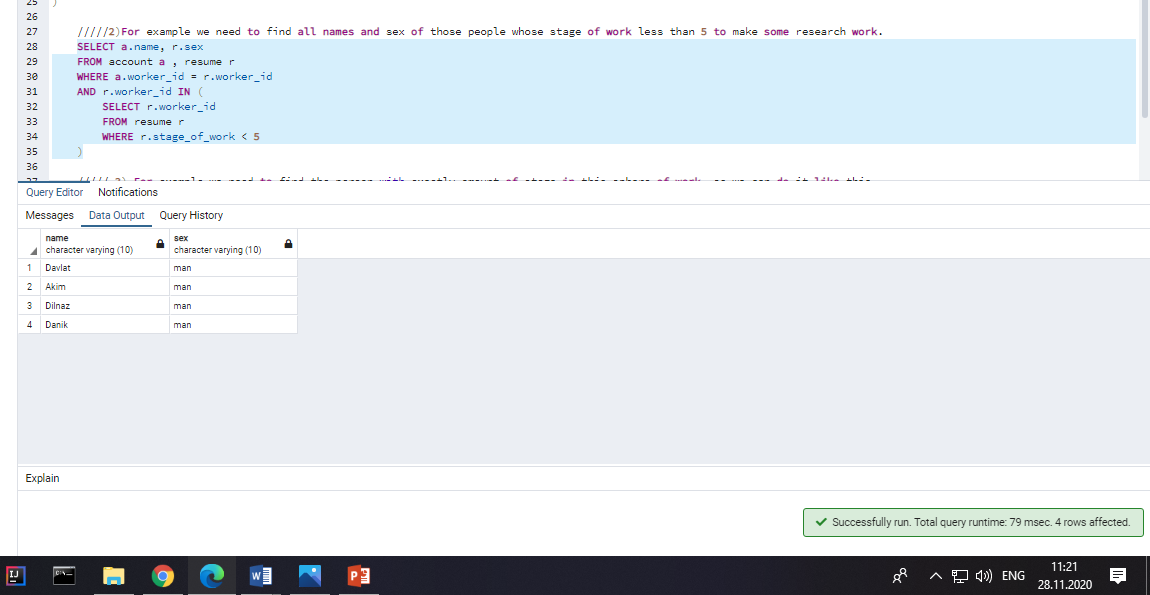
AND r.worker\_id IN (

SELECT r.worker\_id

FROM resume r

WHERE r.stage\_of\_work < 5

)



///// 3) For example we need to find the person with exactly amount of stage in this sphere of work, so we can do it like this

SELECT a.name, r.sex

FROM account a , resume r

WHERE a.worker\_id = r.worker\_id

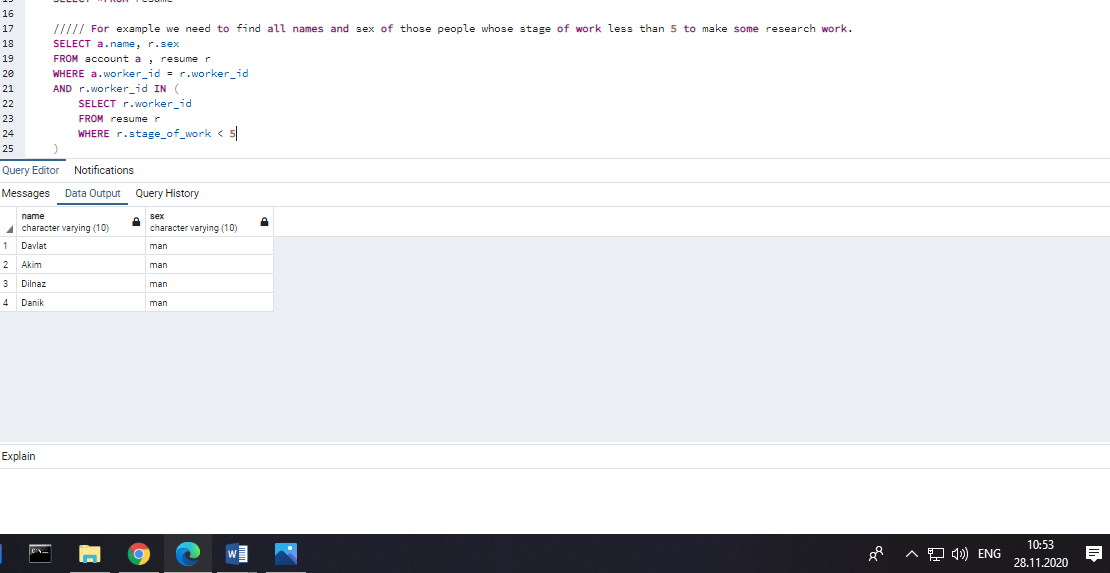
AND EXISTS(

SELECT\*

FROM account ,resume

WHERE r.stage\_of\_work = 10

)



///// 4) Let`s consider this situation when we need to find Any people from Almaty for example

SELECT a.name, r.city\_of\_living ,r.sex

FROM account a , resume r

WHERE a.worker\_id = r.worker\_id

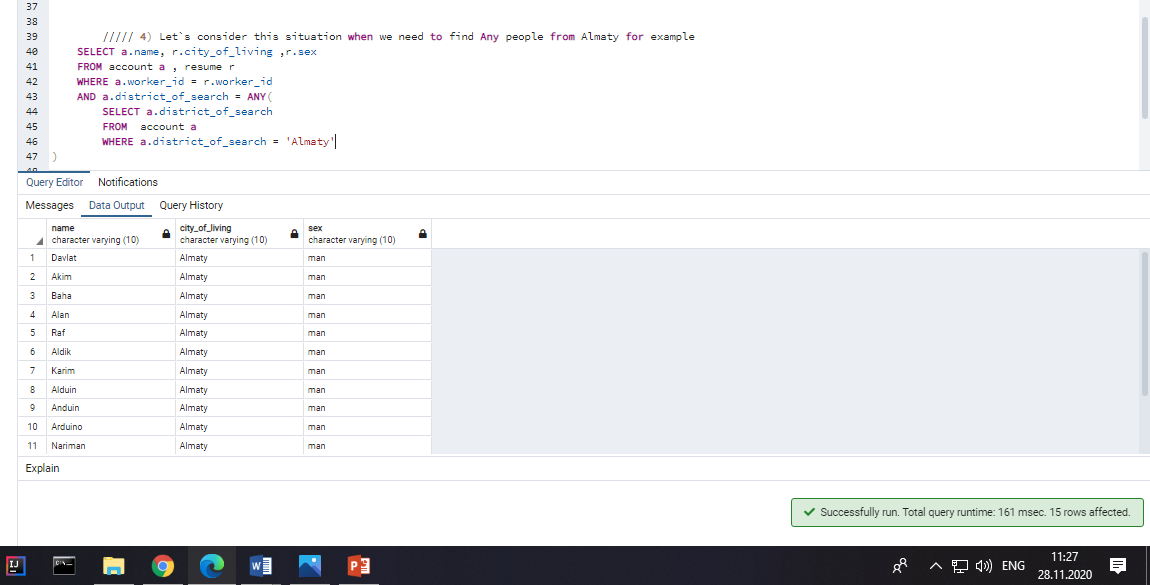
AND a.district\_of\_search = ANY(

SELECT a.district\_of\_search

FROM account a

WHERE a.district\_of\_search = 'Almaty'

)



///// 5) Let`s consider this situation when we need to find All men from the list of candidates

SELECT a.name, r.sex

FROM account a , resume r

WHERE a.worker\_id = r.worker\_id

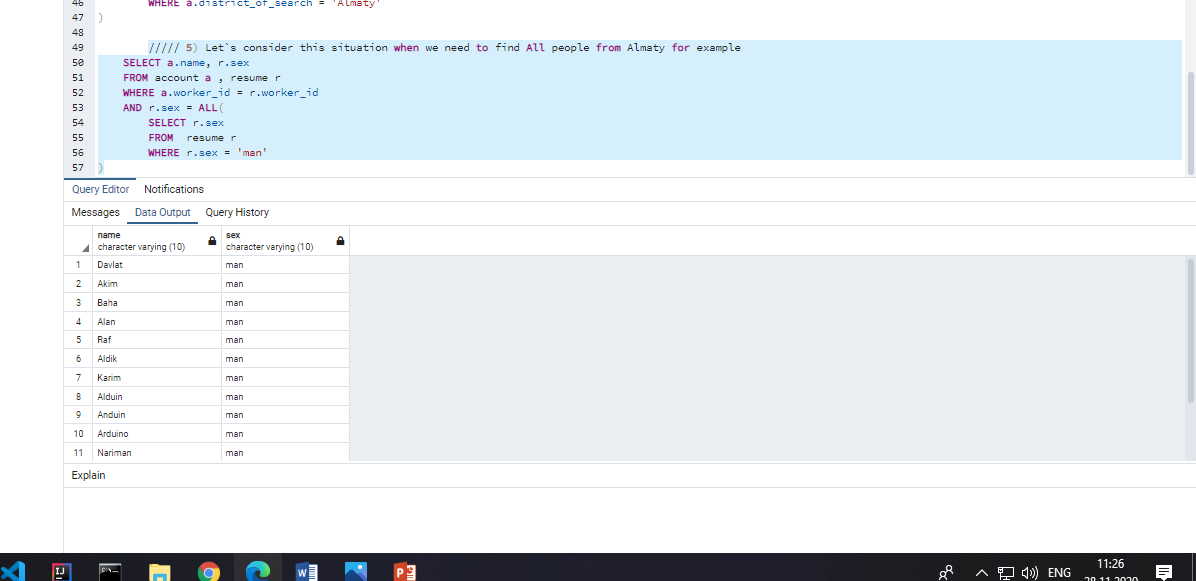
AND r.sex = ALL(

SELECT r.sex

FROM resume r

WHERE r.sex = 'man'

)



// 6) Look, we can use subquery in FROM operator like this and for help of it we can choose what we need to find using alias of FROM operator

SELECT name\_of\_worker

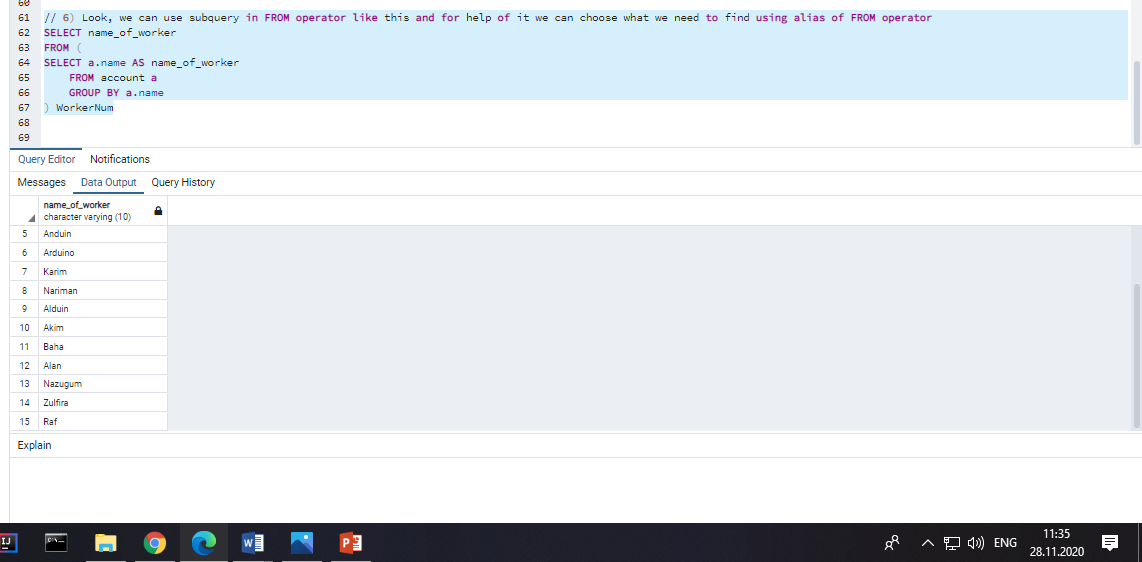
FROM (

SELECT a.name AS name\_of\_worker

FROM account a

GROUP BY a.name

) WorkerNum



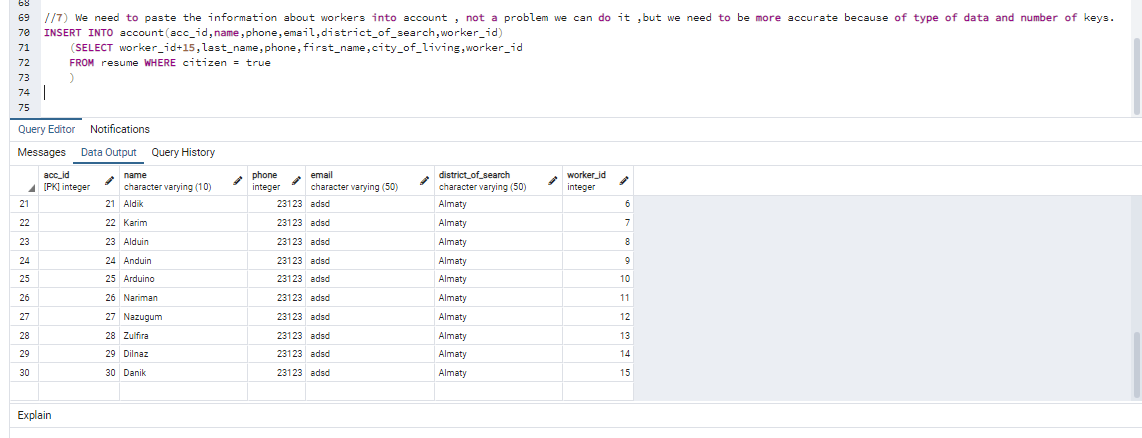
//7) We need to paste the information about workers who are citizens into account , not a problem we can do it ,but we need to be more accurate because of type of data and number of keys.

INSERT INTO account(acc\_id,name,phone,email,district\_of\_search,worker\_id)

(SELECT worker\_id+15,last\_name,phone,first\_name,city\_of\_living,worker\_id

FROM resume WHERE citizen = true

)



/////8) We need to delete some account because user decided to remove it from our site, well we can do it in one query with his id and account table

DELETE

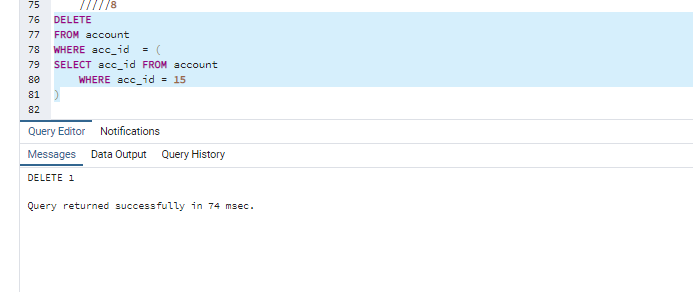
FROM account

WHERE acc\_id = (

SELECT acc\_id FROM account

WHERE acc\_id = 15

)



/// 9) For example we need to update id of some sort of users and we can do it like this

UPDATE account

SET acc\_id = acc\_id +101

WHERE acc\_id = ANY(

SELECT acc\_id

FROM account

WHERE acc\_id > 15

)

