CCT College Dublin

Employees - SQL

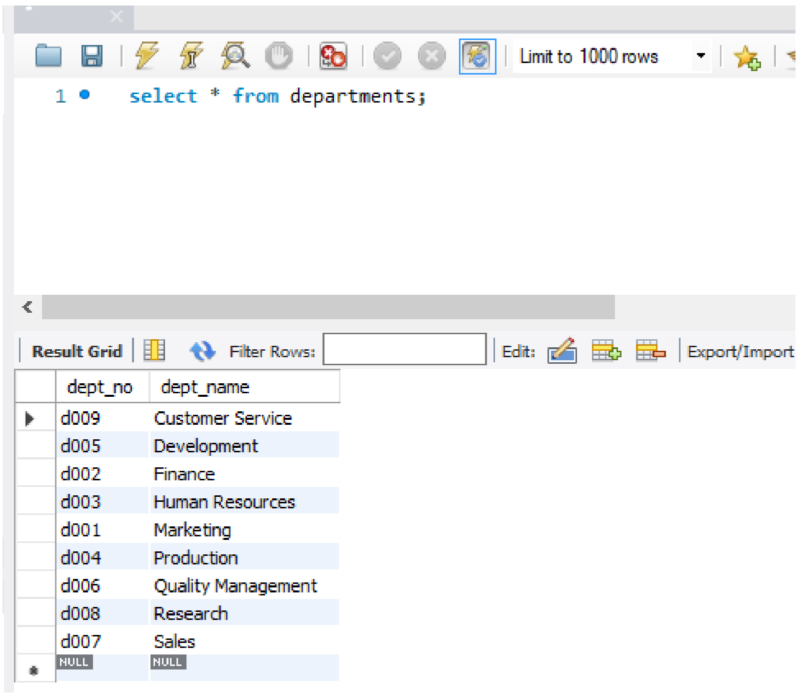
Databases

Juliana Garcia Alves
6/5/2019

SQL Statement

Part 1

1. all the information of the departments:



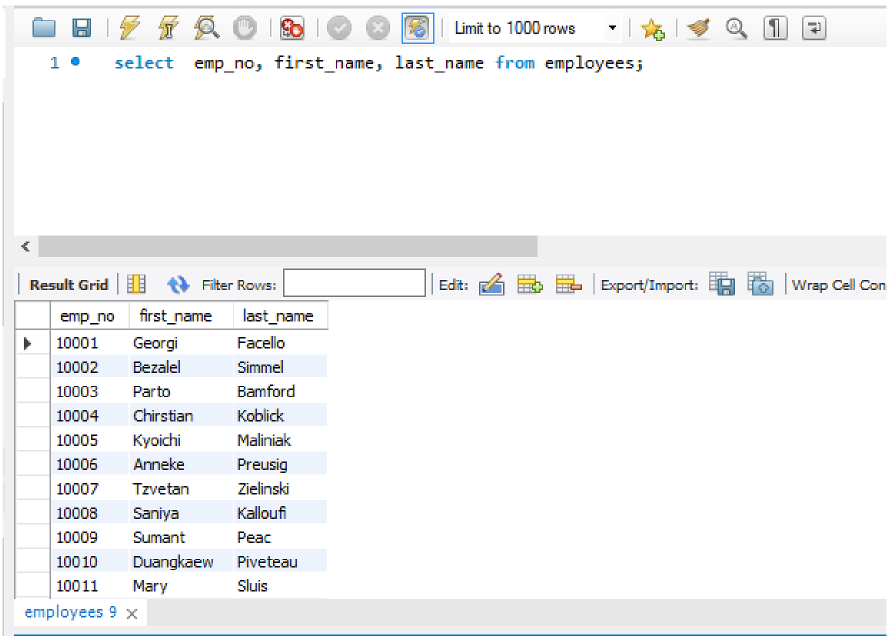
Limit to 1000 rows

```
select * from departments;
```

Result Grid | Filter Rows: | Edit: | Export/Import

	dept_no	dept_name
▶	d009	Customer Service
	d005	Development
	d002	Finance
	d003	Human Resources
	d001	Marketing
	d004	Production
	d006	Quality Management
	d008	Research
	d007	Sales
*	NULL	NULL

2. the employee number, first name and last name of the employees.



Limit to 1000 rows

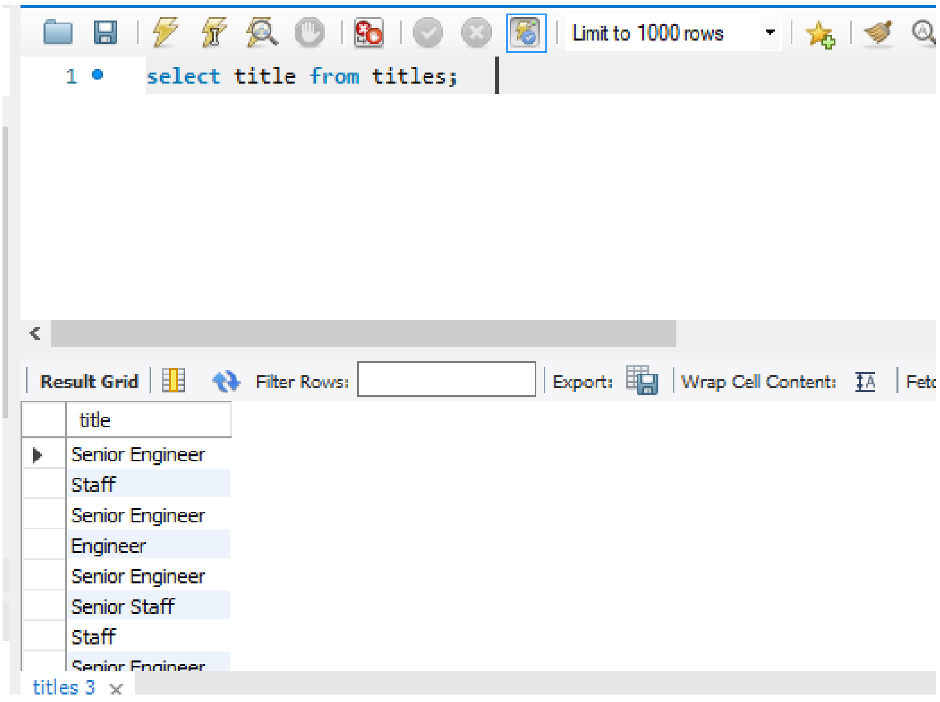
```
select emp_no, first_name, last_name from employees;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Con

	emp_no	first_name	last_name
▶	10001	Georgi	Facello
	10002	Bezalel	Simmel
	10003	Parto	Bamford
	10004	Chirstian	Koblick
	10005	Kyoichi	Maliniak
	10006	Anneke	Preusig
	10007	Tzvetan	Zielinski
	10008	Saniya	Kalloufi
	10009	Sumant	Peac
	10010	Duangkaew	Piveteau
	10011	Mary	Sluis

employees 9 ×

3. all the job titles in the database.



Limit to 1000 rows

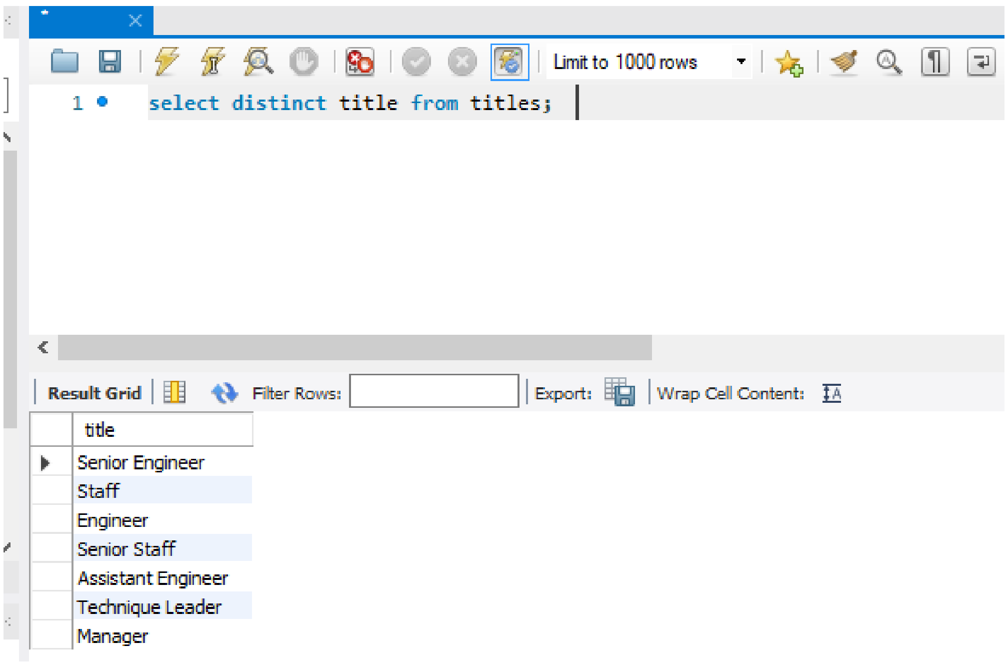
```
select title from titles;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetc

```
title
Senior Engineer
Staff
Senior Engineer
Engineer
Senior Engineer
Senior Staff
Staff
Senior Engineer
```

titles 3

4. all unique   job titles in the database.



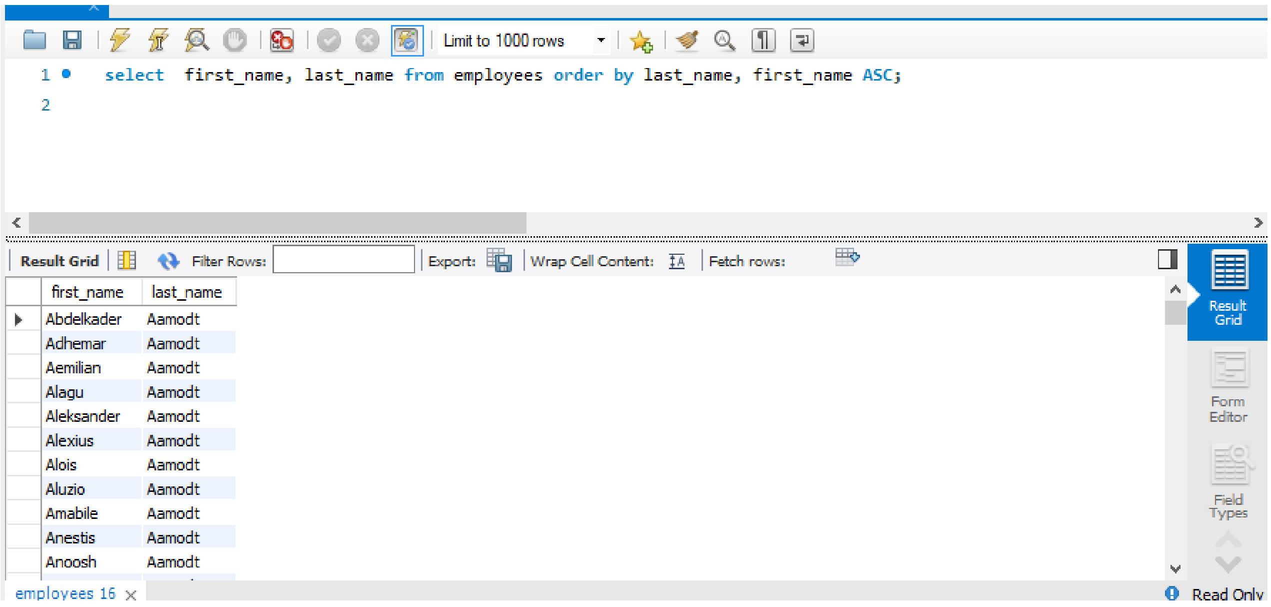
Limit to 1000 rows

```
select distinct title from titles;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
title
Senior Engineer
Staff
Engineer
Senior Staff
Assistant Engineer
Technique Leader
Manager
```

5.all employees names ordered alphabetically in ascending order (note first name and last nam
e are alphabetically ordered).

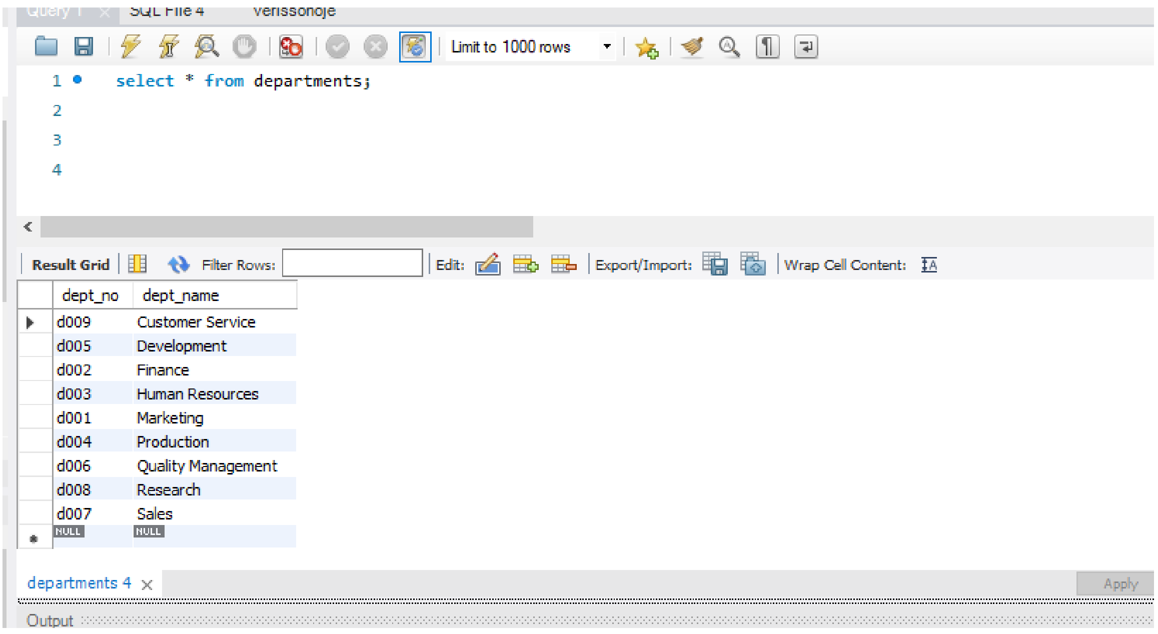


```
select  first_name, last_name from employees order by last_name, first_name ASC;
```

first_name	last_name
Abdelkader	Aamodt
Adhemar	Aamodt
Aemilian	Aamodt
Alagu	Aamodt
Aleksander	Aamodt
Alexius	Aamodt
Alois	Aamodt
Aluzio	Aamodt
Amabile	Aamodt
Anestis	Aamodt
Anoosh	Aamodt

Part 2

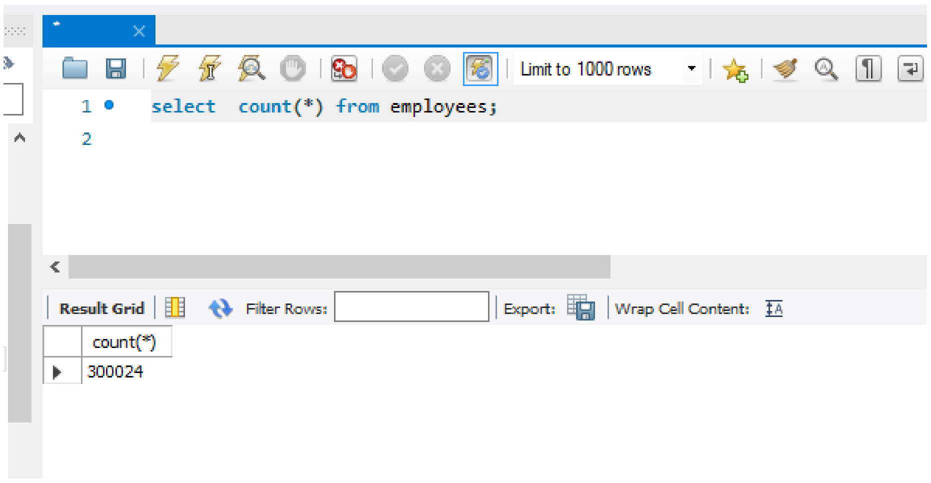
1. all the departments in the database



```
select * from departments;
```

dept_no	dept_name
d009	Customer Service
d005	Development
d002	Finance
d003	Human Resources
d001	Marketing
d004	Production
d006	Quality Management
d008	Research
d007	Sales
NULL	NULL

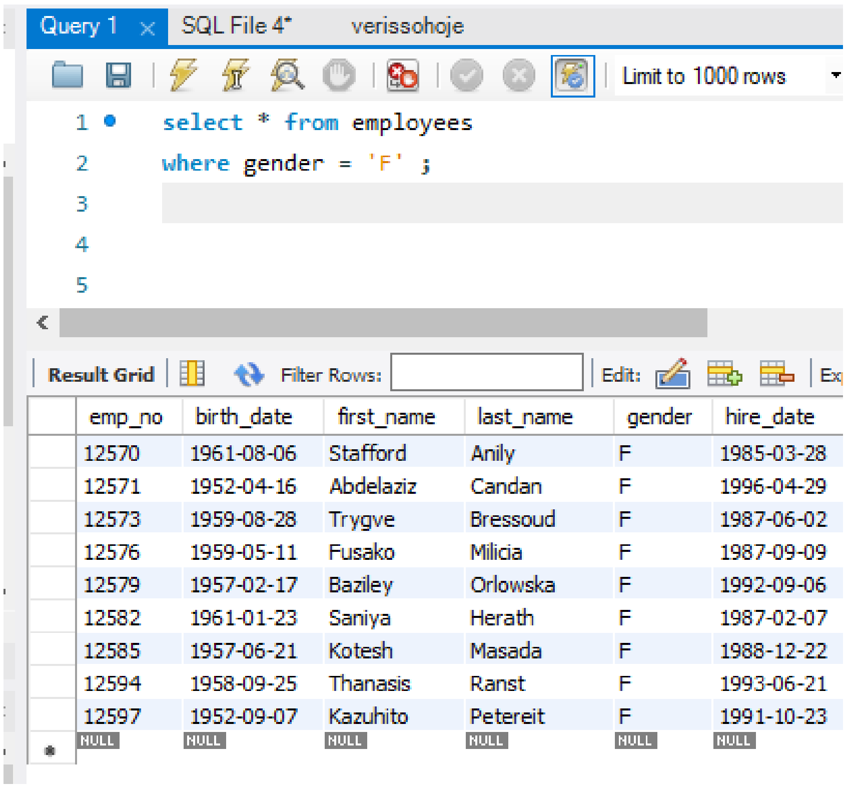
2. the number of employees in our database



```
select count(*) from employees;
```

	count(*)
▶	300024

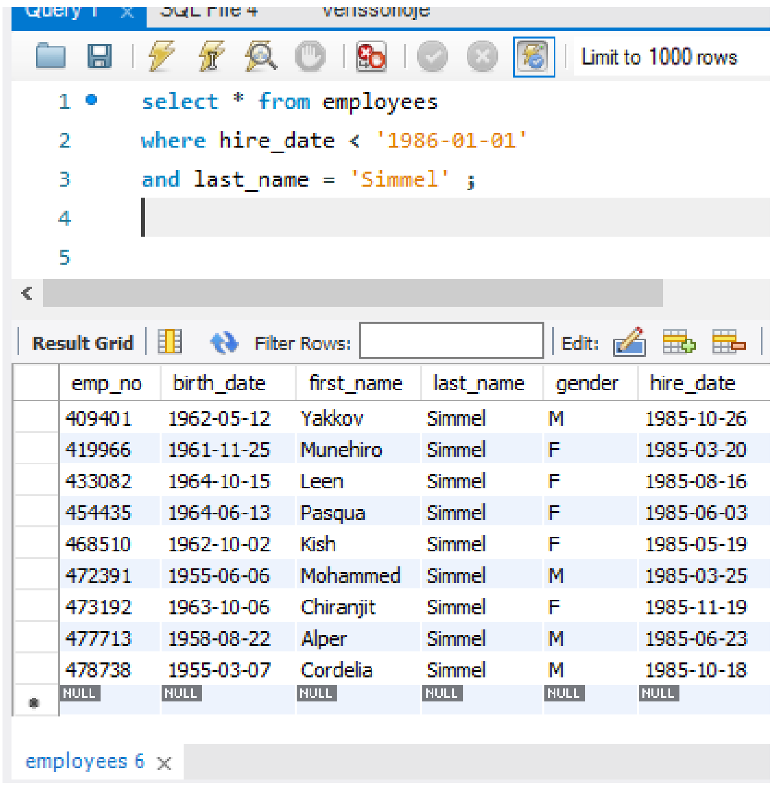
3. all details for female employees only (all columns).



```
select * from employees
where gender = 'F' ;
```

emp_no	birth_date	first_name	last_name	gender	hire_date
12570	1961-08-06	Stafford	Anily	F	1985-03-28
12571	1952-04-16	Abdelaziz	Candan	F	1996-04-29
12573	1959-08-28	Trygve	Bressoud	F	1987-06-02
12576	1959-05-11	Fusako	Milicia	F	1987-09-09
12579	1957-02-17	Baziley	Orlowska	F	1992-09-06
12582	1961-01-23	Saniya	Herath	F	1987-02-07
12585	1957-06-21	Kotesh	Masada	F	1988-12-22
12594	1958-09-25	Thanasis	Ranst	F	1993-06-21
12597	1952-09-07	Kazuhito	Petereit	F	1991-10-23
NULL	NULL	NULL	NULL	NULL	NULL

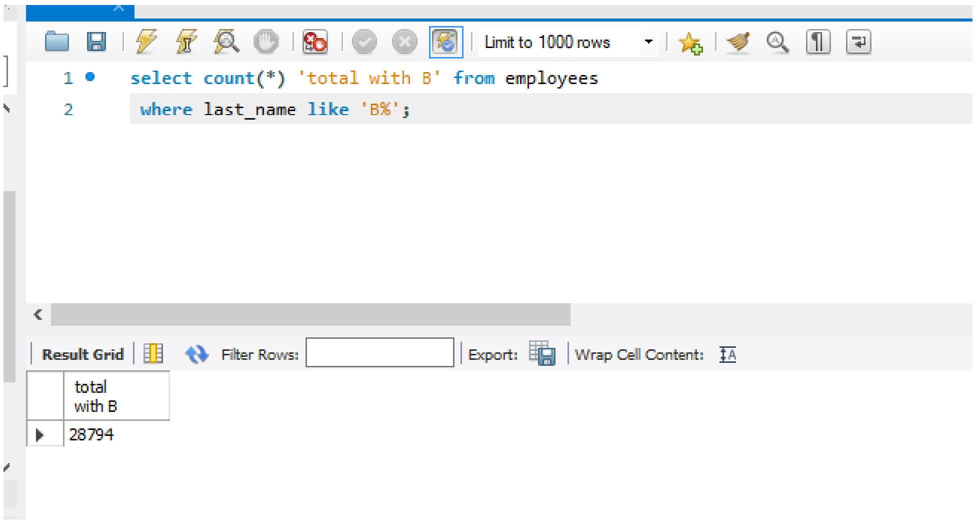
4. the employees that joined before ‘1986-1-1’ and have a last name of “Simmel”



```sql
select * from employees
where hire_date < '1986-01-01'
and last_name = 'Simmel' ;
```

emp_no	birth_date	first_name	last_name	gender	hire_date
409401	1962-05-12	Yakkov	Simmel	M	1985-10-26
419966	1961-11-25	Munehiro	Simmel	F	1985-03-20
433082	1964-10-15	Leen	Simmel	F	1985-08-16
454435	1964-06-13	Pasqua	Simmel	F	1985-06-03
468510	1962-10-02	Kish	Simmel	F	1985-05-19
472391	1955-06-06	Mohammed	Simmel	M	1985-03-25
473192	1963-10-06	Chiranjit	Simmel	F	1985-11-19
477713	1958-08-22	Alper	Simmel	M	1985-06-23
478738	1955-03-07	Cordelia	Simmel	M	1985-10-18
NULL	NULL	NULL	NULL	NULL	NULL

5. how many employees are in the database whose last name begins with the letter B. Use an alias (table title) as ‘total with B’ to output your results.

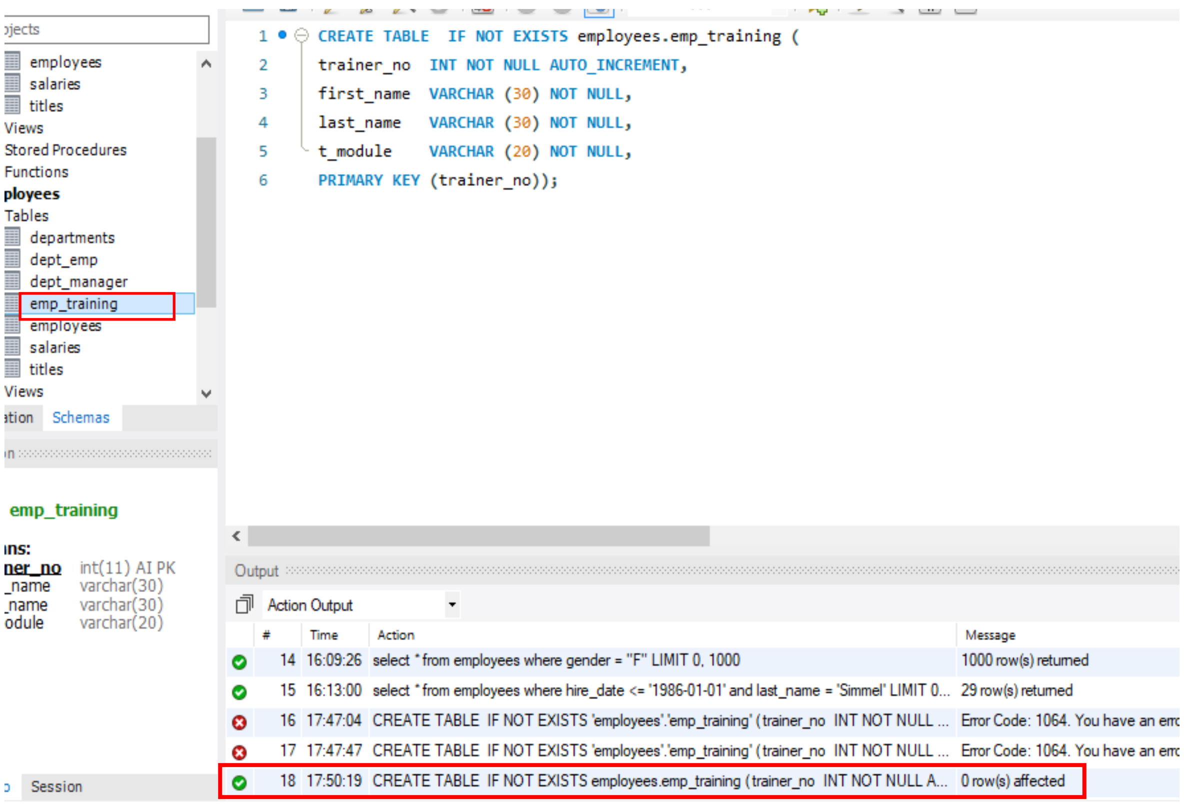


```sql
select count(*) 'total with B' from employees
where last_name like 'B%';
```

total with B
28794

6. Create a new table called emp_training with 3 columns:

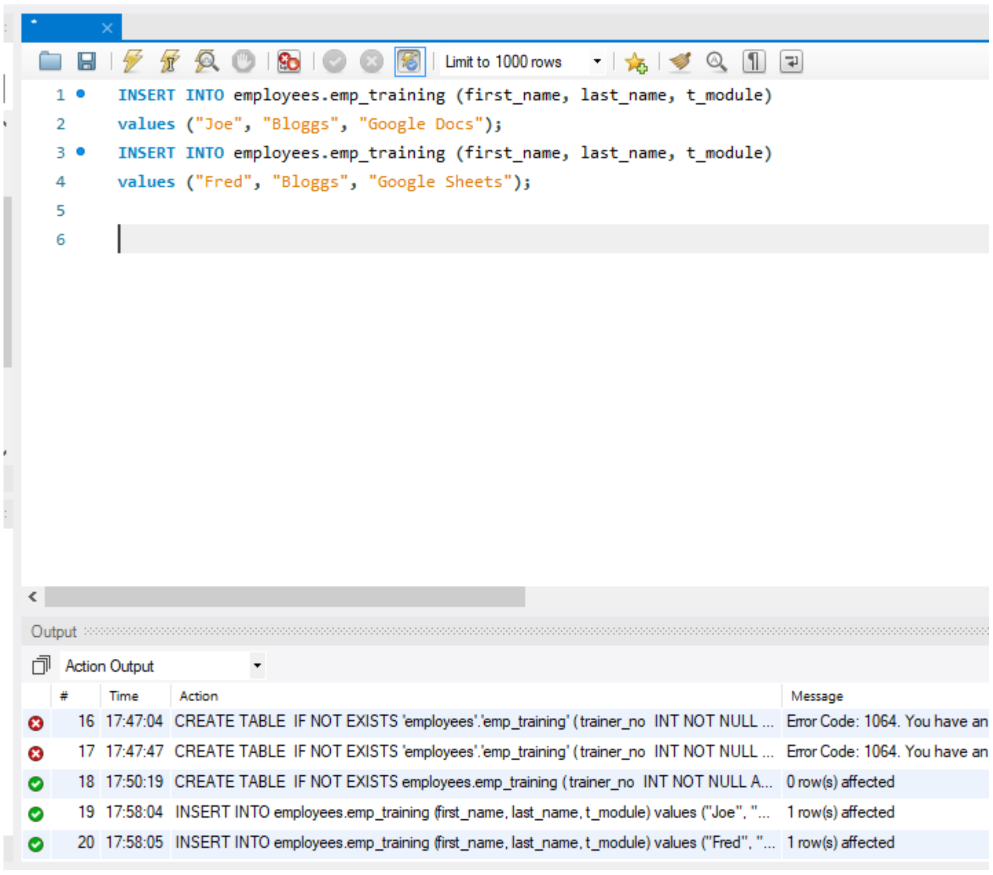
- trainer_no: this should be the primary key and is of type integer and needs to impleme nted as an auto-increment.
- first_name: this data type is varchar (30) and should not be NULL
- last_name: this data type is varchar (30) and should not be NULL
- t_module: this data type is varchar (20).



```
CREATE TABLE  IF NOT EXISTS employees.emp_training (
trainer_no  INT NOT NULL AUTO_INCREMENT,
first_name  VARCHAR (30) NOT NULL,
last_name   VARCHAR (30) NOT NULL,
t_module    VARCHAR (20) NOT NULL,
PRIMARY KEY (trainer_no));
```

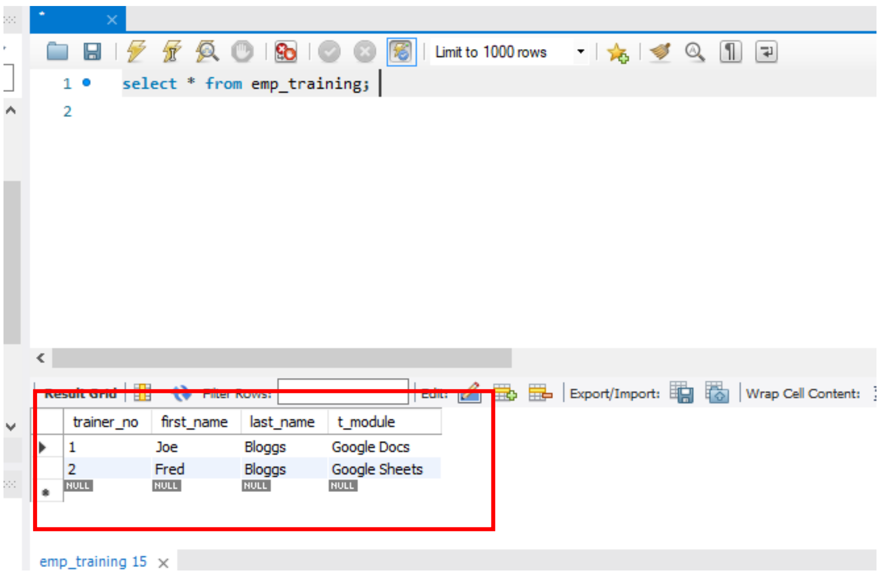
```
#   Time      Action                                                                        Message
14  16:09:26  select * from employees where gender = "F" LIMIT 0, 1000                      1000 row(s) returned
15  16:13:00  select * from employees where hire_date <= '1986-01-01' and last_name = 'Simmel' LIMIT 0...  29 row(s) returned
16  17:47:04  CREATE TABLE  IF NOT EXISTS 'employees'.'emp_training' (trainer_no  INT NOT NULL ...  Error Code: 1064. You have an erro
17  17:47:47  CREATE TABLE  IF NOT EXISTS 'employees'.'emp_training' (trainer_no  INT NOT NULL ...  Error Code: 1064. You have an erro
18  17:50:19  CREATE TABLE  IF NOT EXISTS employees.emp_training (trainer_no  INT NOT NULL A...  0 row(s) affected
```

7. Insert two new rows into the emp_training table: Record 1 – fname: "Joe" lname: "Bloggs" module: "Google Docs" Record 2 – fname: "Fred" lname: "Bloggs" module: "Google Sheets"



```
INSERT INTO employees.emp_training (first_name, last_name, t_module)
values ("Joe", "Bloggs", "Google Docs");
INSERT INTO employees.emp_training (first_name, last_name, t_module)
values ("Fred", "Bloggs", "Google Sheets");
```

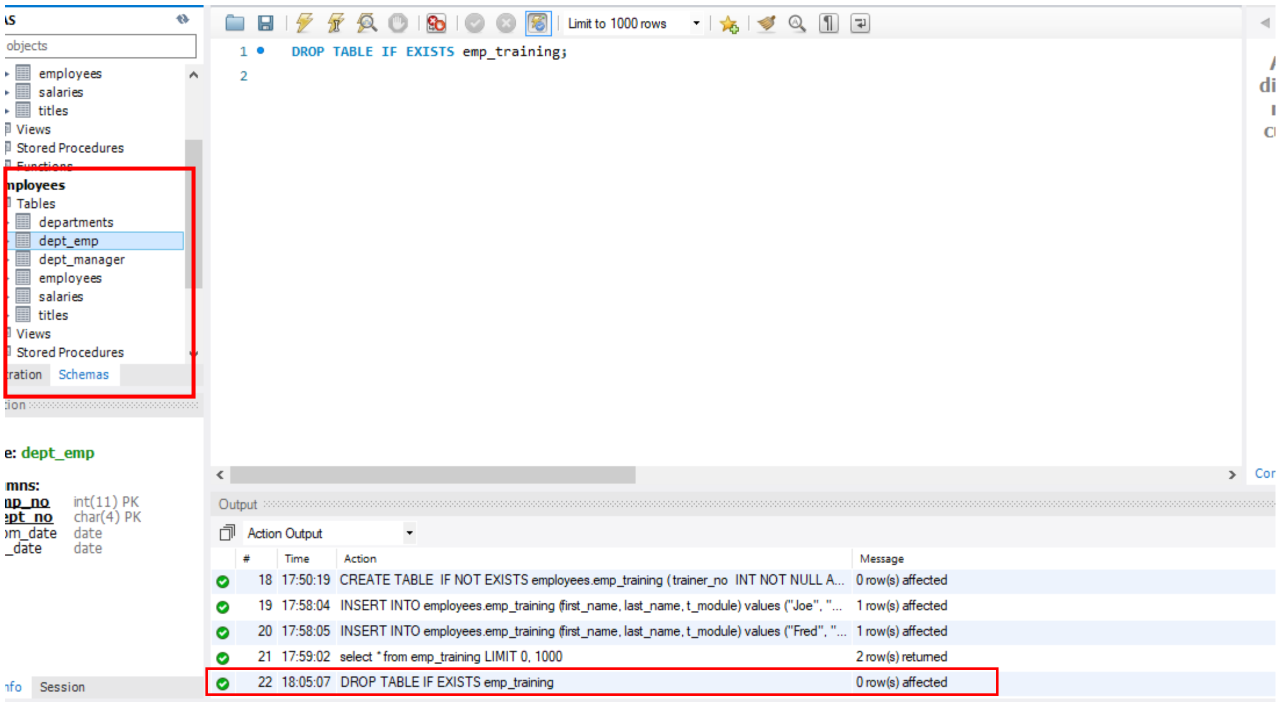
```
#   Time      Action                                                                              Message
16  17:47:04  CREATE TABLE  IF NOT EXISTS 'employees'.'emp_training' (trainer_no  INT NOT NULL ...   Error Code: 1064. You have an
17  17:47:47  CREATE TABLE  IF NOT EXISTS 'employees'.'emp_training' (trainer_no  INT NOT NULL ...   Error Code: 1064. You have an
18  17:50:19  CREATE TABLE  IF NOT EXISTS employees.emp_training (trainer_no  INT NOT NULL A...      0 row(s) affected
19  17:58:04  INSERT INTO employees.emp_training (first_name, last_name, t_module) values ("Joe", "...   1 row(s) affected
20  17:58:05  INSERT INTO employees.emp_training (first_name, last_name, t_module) values ("Fred", "...  1 row(s) affected
```



```
select * from emp_training;
```

trainer_no	first_name	last_name	t_module
1	Joe	Bloggs	Google Docs
2	Fred	Bloggs	Google Sheets
NULL	NULL	NULL	NULL

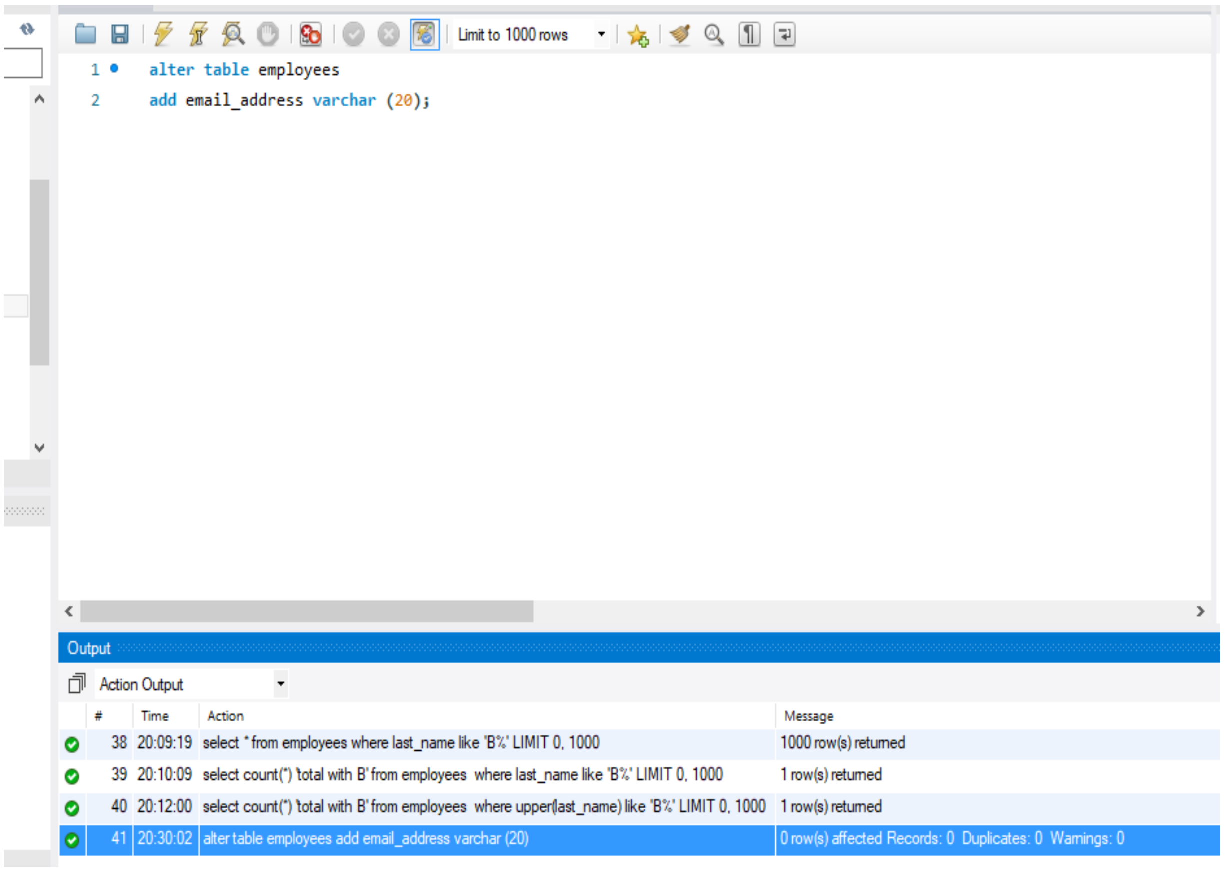
8. The organization no longer wishes to record the employees training within the database. Therefore, delete the newly created emp_training table.



```
DROP TABLE IF EXISTS emp_training;
```

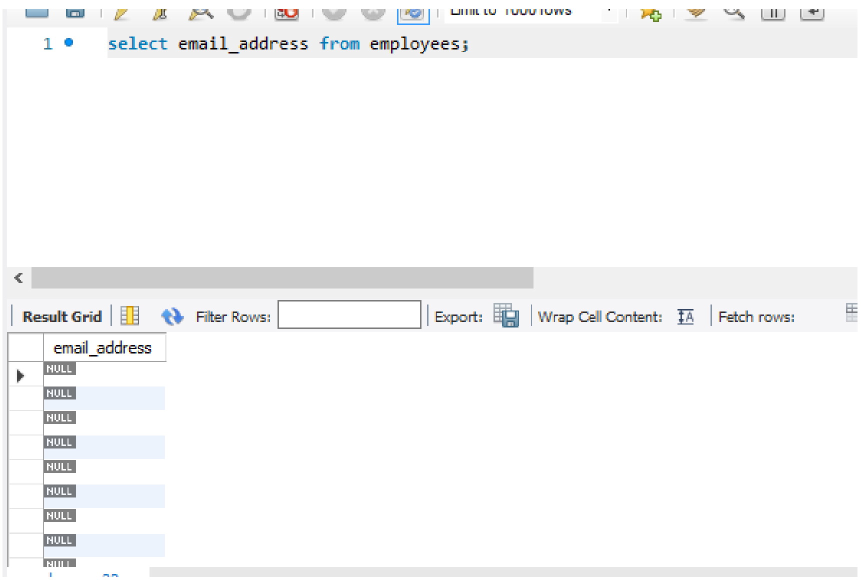
```
#   Time      Action                                                                              Message
18  17:50:19  CREATE TABLE IF NOT EXISTS employees.emp_training (trainer_no INT NOT NULL A...    0 row(s) affected
19  17:58:04  INSERT INTO employees.emp_training (first_name, last_name, t_module) values ("Joe", "...   1 row(s) affected
20  17:58:05  INSERT INTO employees.emp_training (first_name, last_name, t_module) values ("Fred", "...  1 row(s) affected
21  17:59:02  select * from emp_training LIMIT 0, 1000                                          2 row(s) returned
22  18:05:07  DROP TABLE IF EXISTS emp_training                                                 0 row(s) affected
```

9. Alter the employees table to include an email_address field with a data type of varchar(20).



```
alter table employees
add email_address varchar (20);
```

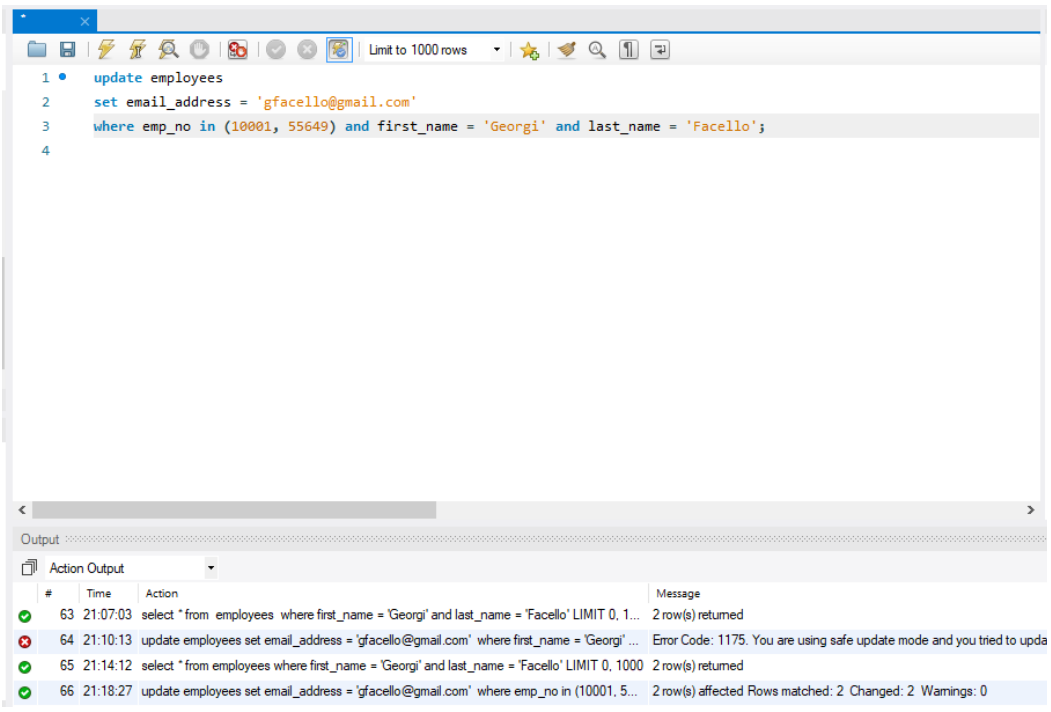
```
#   Time      Action                                                                          Message
38  20:09:19  select * from employees where last_name like 'B%' LIMIT 0, 1000                  1000 row(s) returned
39  20:10:09  select count(*) total with B' from employees where last_name like 'B%' LIMIT 0, 1000   1 row(s) returned
40  20:12:00  select count(*) total with B' from employees where upper(last_name) like 'B%' LIMIT 0, 1000   1 row(s) returned
41  20:30:02  alter table employees add email_address varchar (20)                            0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
```



```
select email_address from employees;
```

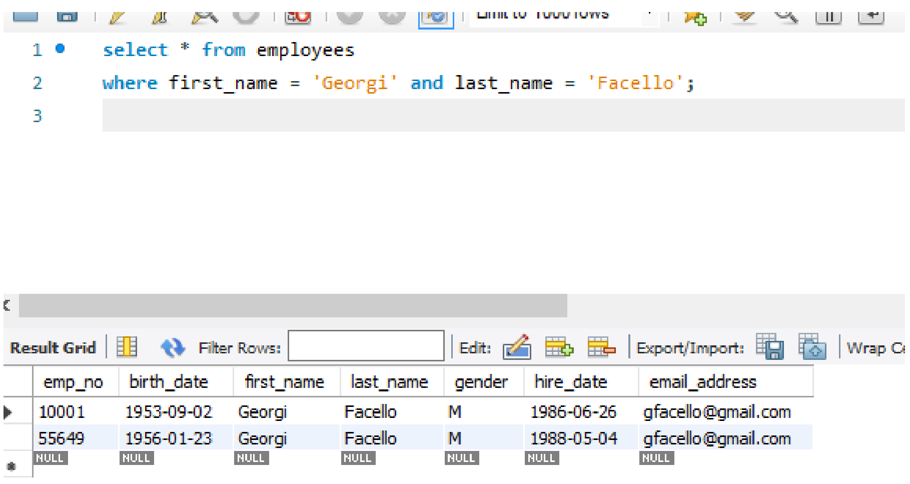
email_address
NULL
NULL
NULL
NULL
NULL
NULL
NULL
NULL
NULL

10. Update the email address of Georgi Facello to gfacello@gmail.com.



```
update employees
set email_address = 'gfacello@gmail.com'
where emp_no in (10001, 55649) and first_name = 'Georgi' and last_name = 'Facello';
```

	#	Time	Action	Message
✓	63	21:07:03	select * from employees where first_name = 'Georgi' and last_name = 'Facello' LIMIT 0, 1...	2 row(s) returned
✗	64	21:10:13	update employees set email_address = 'gfacello@gmail.com' where first_name = 'Georgi' ...	Error Code: 1175. You are using safe update mode and you tried to upda
✓	65	21:14:12	select * from employees where first_name = 'Georgi' and last_name = 'Facello' LIMIT 0, 1000	2 row(s) returned
✓	66	21:18:27	update employees set email_address = 'gfacello@gmail.com' where emp_no in (10001, 5...	2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0

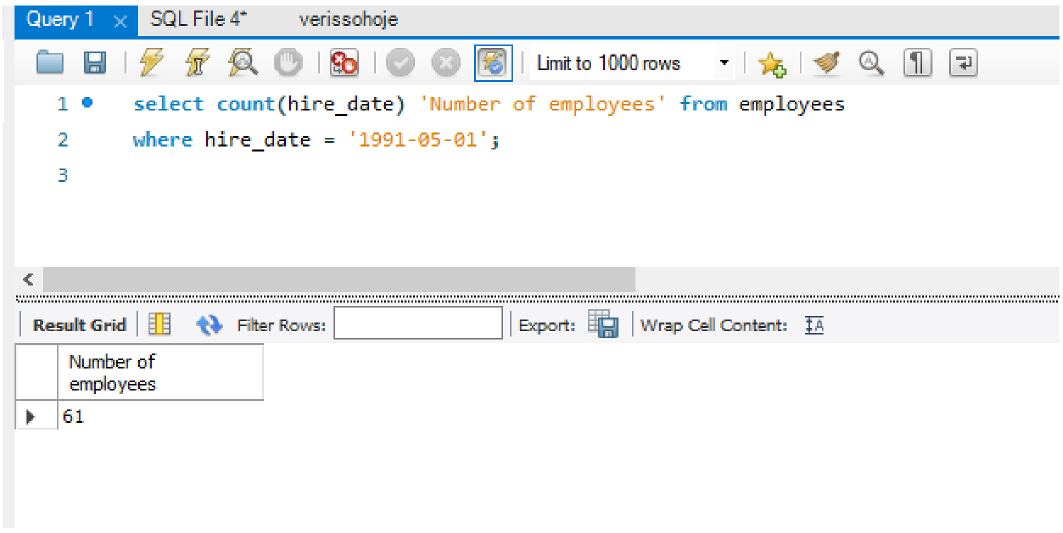


```
select * from employees
where first_name = 'Georgi' and last_name = 'Facello';
```

emp_no	birth_date	first_name	last_name	gender	hire_date	email_address
10001	1953-09-02	Georgi	Facello	M	1986-06-26	gfacello@gmail.com
55649	1956-01-23	Georgi	Facello	M	1988-05-04	gfacello@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Part 3

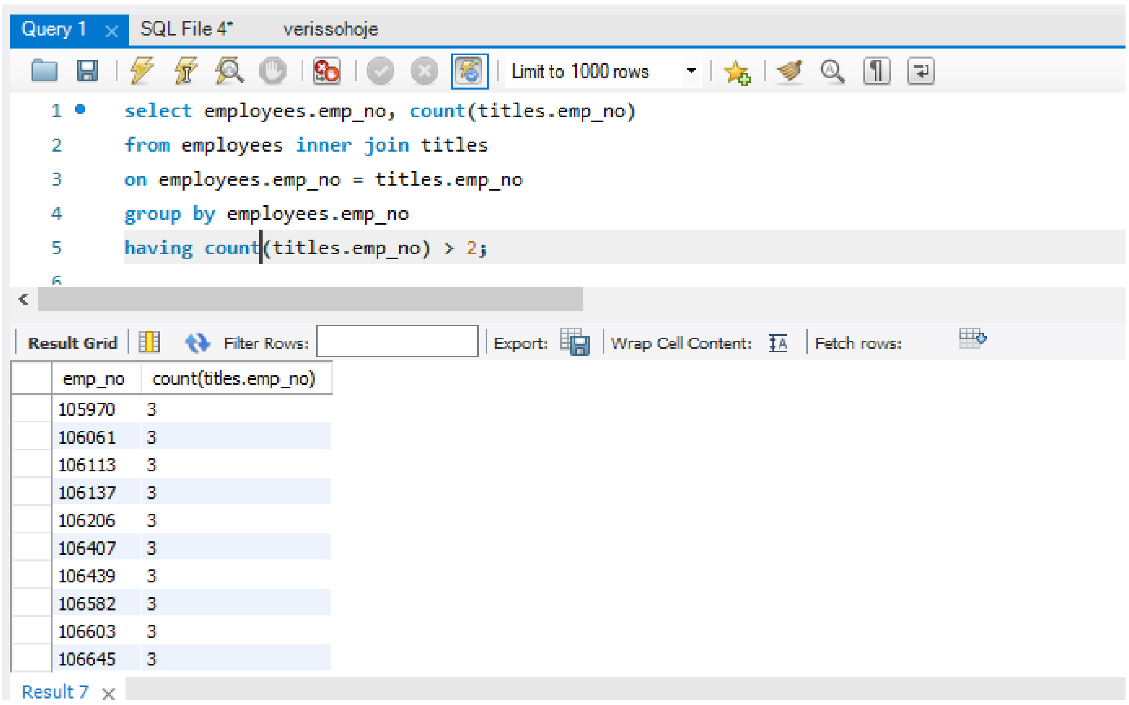
1. the number of employees that started on ‘1991-05-01’



```sql
select count(hire_date) 'Number of employees' from employees
where hire_date = '1991-05-01';
```

Number of employees
61

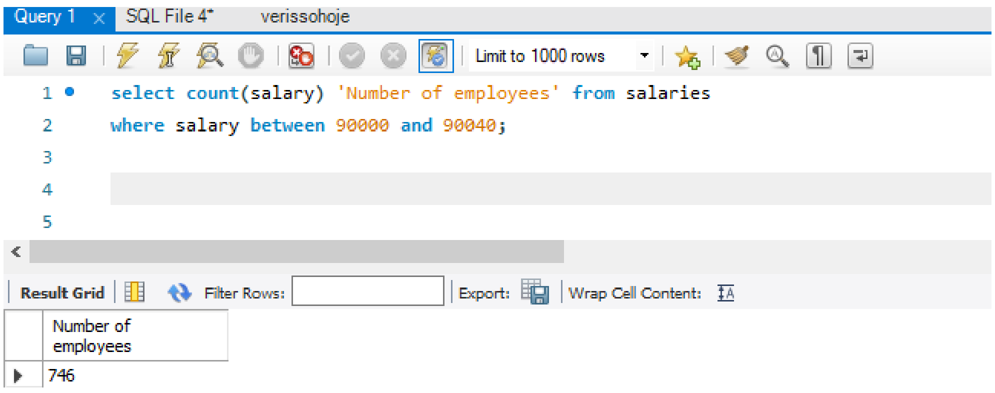
2. list all employee IDs (emp_no) who have had more than 2 title and show the number of titles they have had.



```sql
select employees.emp_no, count(titles.emp_no)
from employees inner join titles
on employees.emp_no = titles.emp_no
group by employees.emp_no
having count(titles.emp_no) > 2;
```

emp_no	count(titles.emp_no)
105970	3
106061	3
106113	3
106137	3
106206	3
106407	3
106439	3
106582	3
106603	3
106645	3

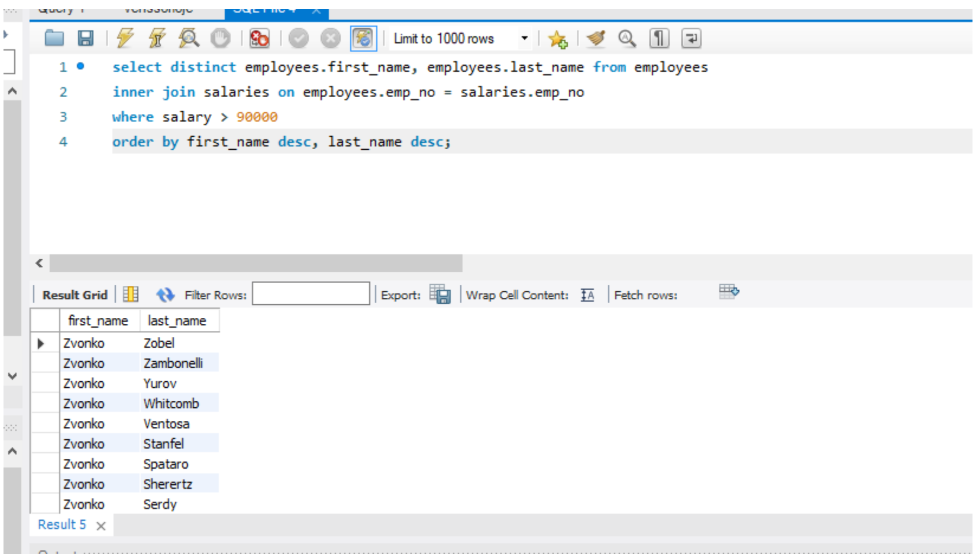
3. the number of employees who have a salary between 90000 and 90040.



```sql
select count(salary) 'Number of employees' from salaries
where salary between 90000 and 90040;
```

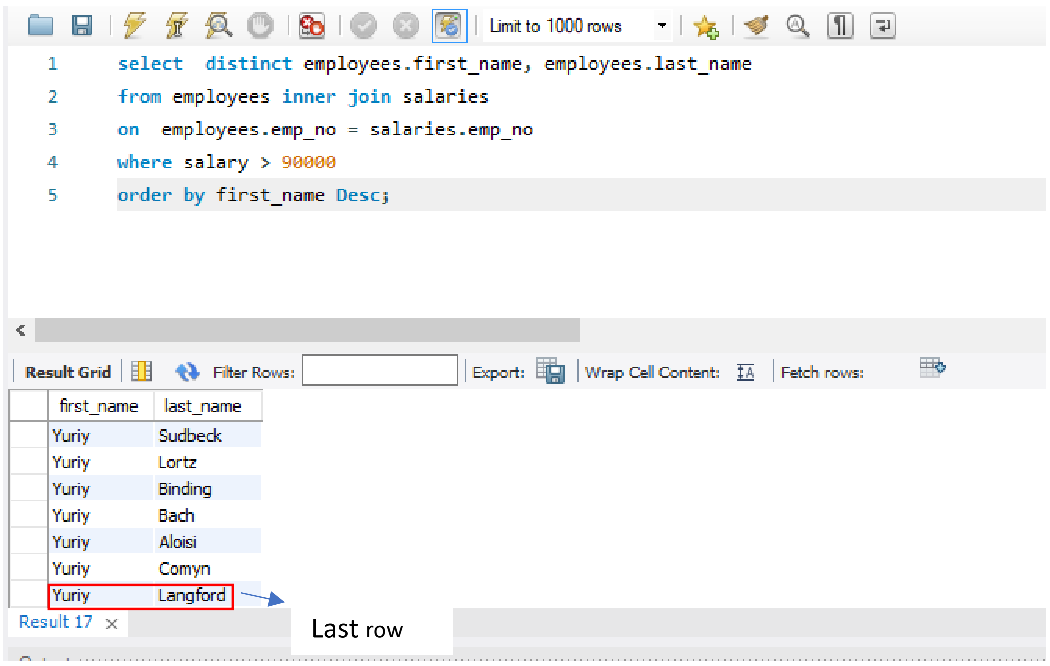
Number of employees
746

4. a list of only unique employee first and last names who have a salary greater than 90000 , and order this in descending order (using the INNER JOIN method).



```sql
select distinct employees.first_name, employees.last_name from employees
inner join salaries on employees.emp_no = salaries.emp_no
where salary > 90000
order by first_name desc, last_name desc;
```

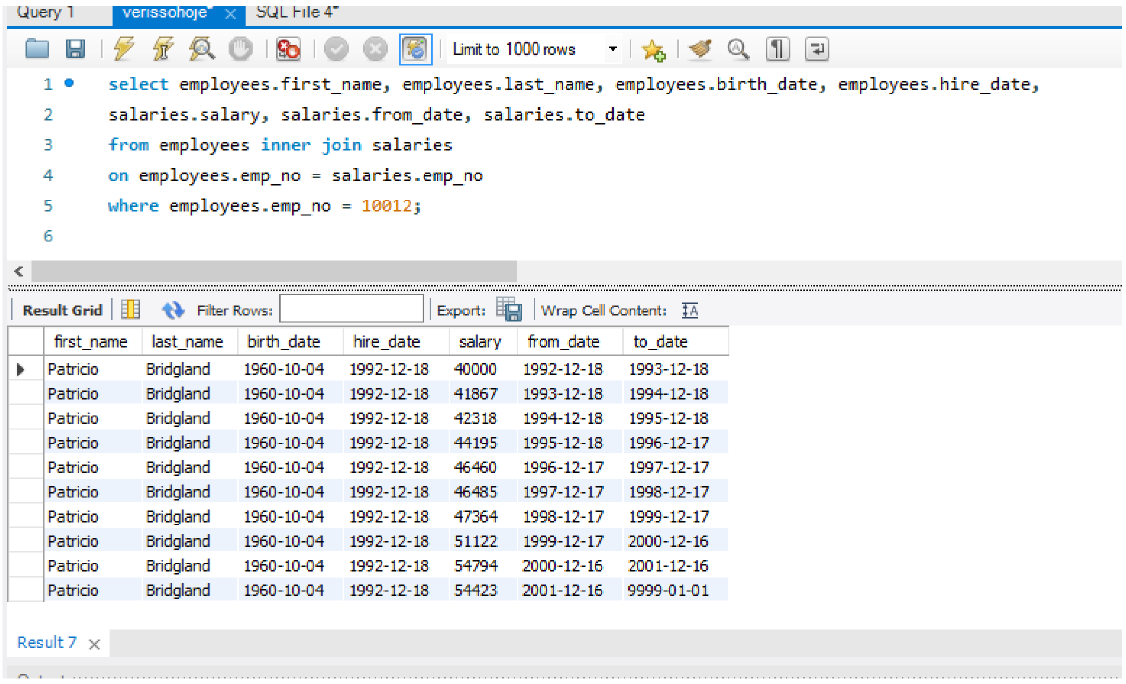
first_name	last_name
Zvonko	Zobel
Zvonko	Zambonelli
Zvonko	Yurov
Zvonko	Whitcomb
Zvonko	Ventosa
Zvonko	Stanfel
Zvonko	Spataro
Zvonko	Sherertz
Zvonko	Serdy



```sql
select  distinct employees.first_name, employees.last_name
from employees inner join salaries
on  employees.emp_no = salaries.emp_no
where salary > 90000
order by first_name Desc;
```

first_name	last_name
Yuriy	Sudbeck
Yuriy	Lortz
Yuriy	Binding
Yuriy	Bach
Yuriy	Aloisi
Yuriy	Comyn
Yuriy	Langford

5. first name, last name, dates and salaries for the employee whose employee number is "10012".



```
select employees.first_name, employees.last_name, employees.birth_date, employees.hire_date,
salaries.salary, salaries.from_date, salaries.to_date
from employees inner join salaries
on employees.emp_no = salaries.emp_no
where employees.emp_no = 10012;
```

first_name	last_name	birth_date	hire_date	salary	from_date	to_date
Patricio	Bridgland	1960-10-04	1992-12-18	40000	1992-12-18	1993-12-18
Patricio	Bridgland	1960-10-04	1992-12-18	41867	1993-12-18	1994-12-18
Patricio	Bridgland	1960-10-04	1992-12-18	42318	1994-12-18	1995-12-18
Patricio	Bridgland	1960-10-04	1992-12-18	44195	1995-12-18	1996-12-17
Patricio	Bridgland	1960-10-04	1992-12-18	46460	1996-12-17	1997-12-17
Patricio	Bridgland	1960-10-04	1992-12-18	46485	1997-12-17	1998-12-17
Patricio	Bridgland	1960-10-04	1992-12-18	47364	1998-12-17	1999-12-17
Patricio	Bridgland	1960-10-04	1992-12-18	51122	1999-12-17	2000-12-16
Patricio	Bridgland	1960-10-04	1992-12-18	54794	2000-12-16	2001-12-16
Patricio	Bridgland	1960-10-04	1992-12-18	54423	2001-12-16	9999-01-01

6. In relation to the table named ***salaries*** in Figure 1 above:
   a. what is the degree of this table?

   Columns: emp_no, salary, from_date and to_date

   b. what column(s), if any, make(s) up the primary key?

   Columns: emp_no and from_date.

   c. What column(s), if any, make(s) up the foreign key?

   Column: emp_no

7. In the given schema, the tables dept_emp, dept_manager, salaries, titles all have composite keys. Explain for each table why this is the case.

Composite key, or composite primary key, refers to cases where more than one column is used to specify the primary key of a table. In these tables, each of the rows individually can not uniquely identify each record, but together the combination of all them does uniquely identify each record.

a. Dept_manager:
   Composite Key: emp_no, dept_no
   A manager is a manager and is also an employee. A manager can have more than one department as a department can have more than one manager.
b. Dept_emp:

Composite Key: emp_no, dept_no. In this case, an employee can work in more than one department and a department can have more than one employee.

c. Titles:
   Composite key: emp_no, title, from_date: in this case, an employee may have more than one title, at different times

d. Salaries:
   Composite Key: emp_no, from_date
   An employee may change the salary, many times (not only once).