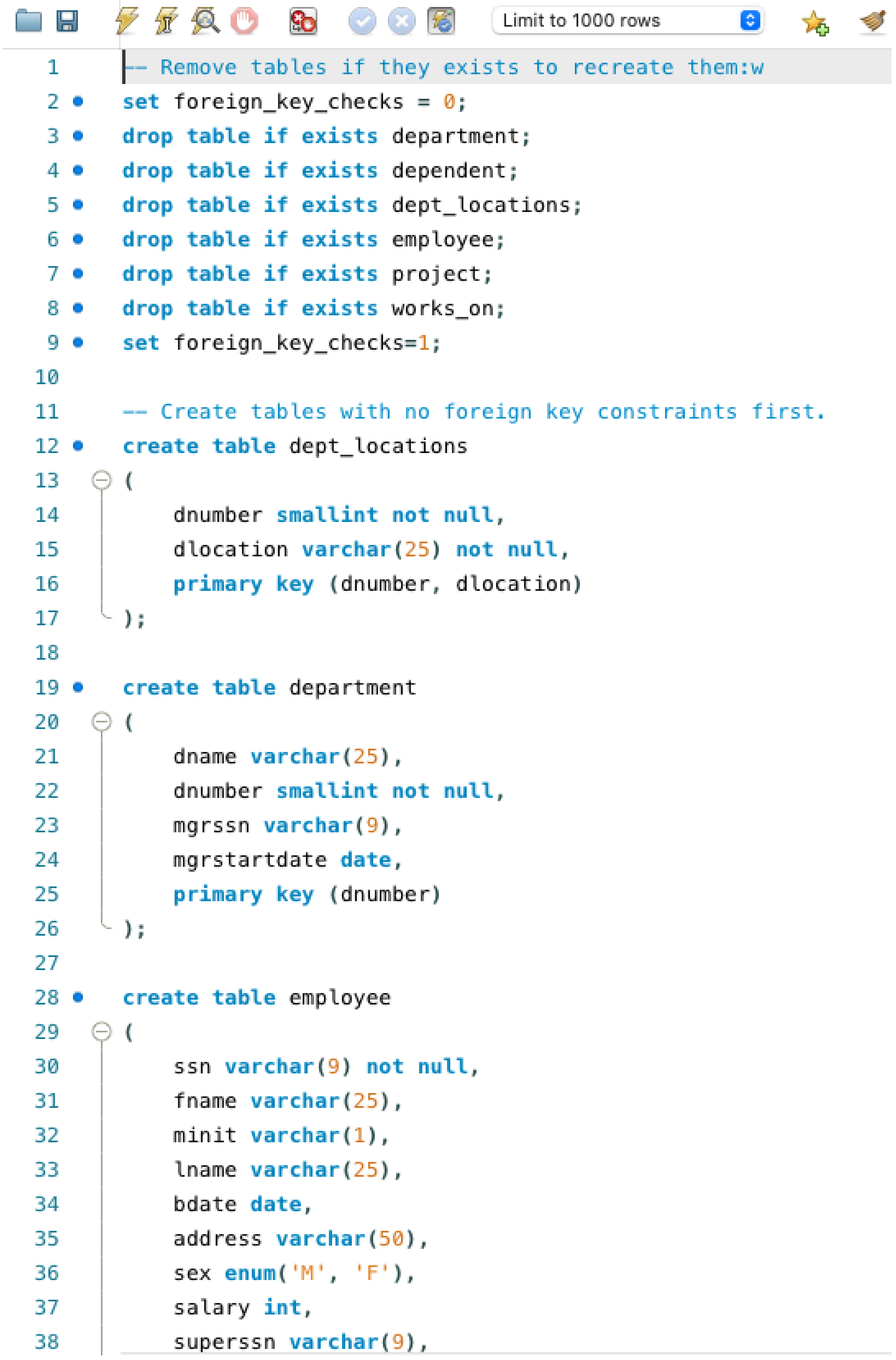
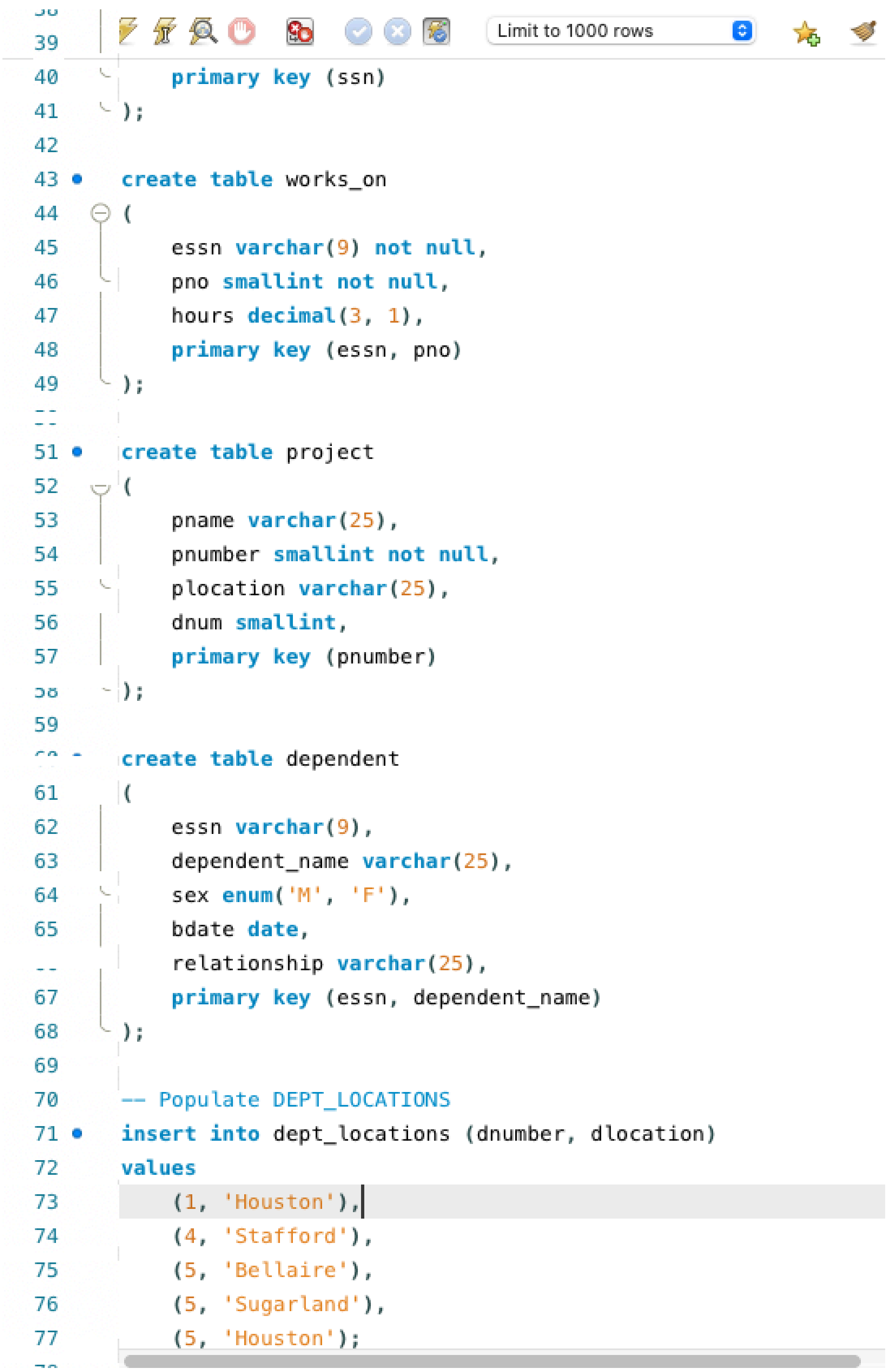
Devere Anthony Weaver
Homework 5

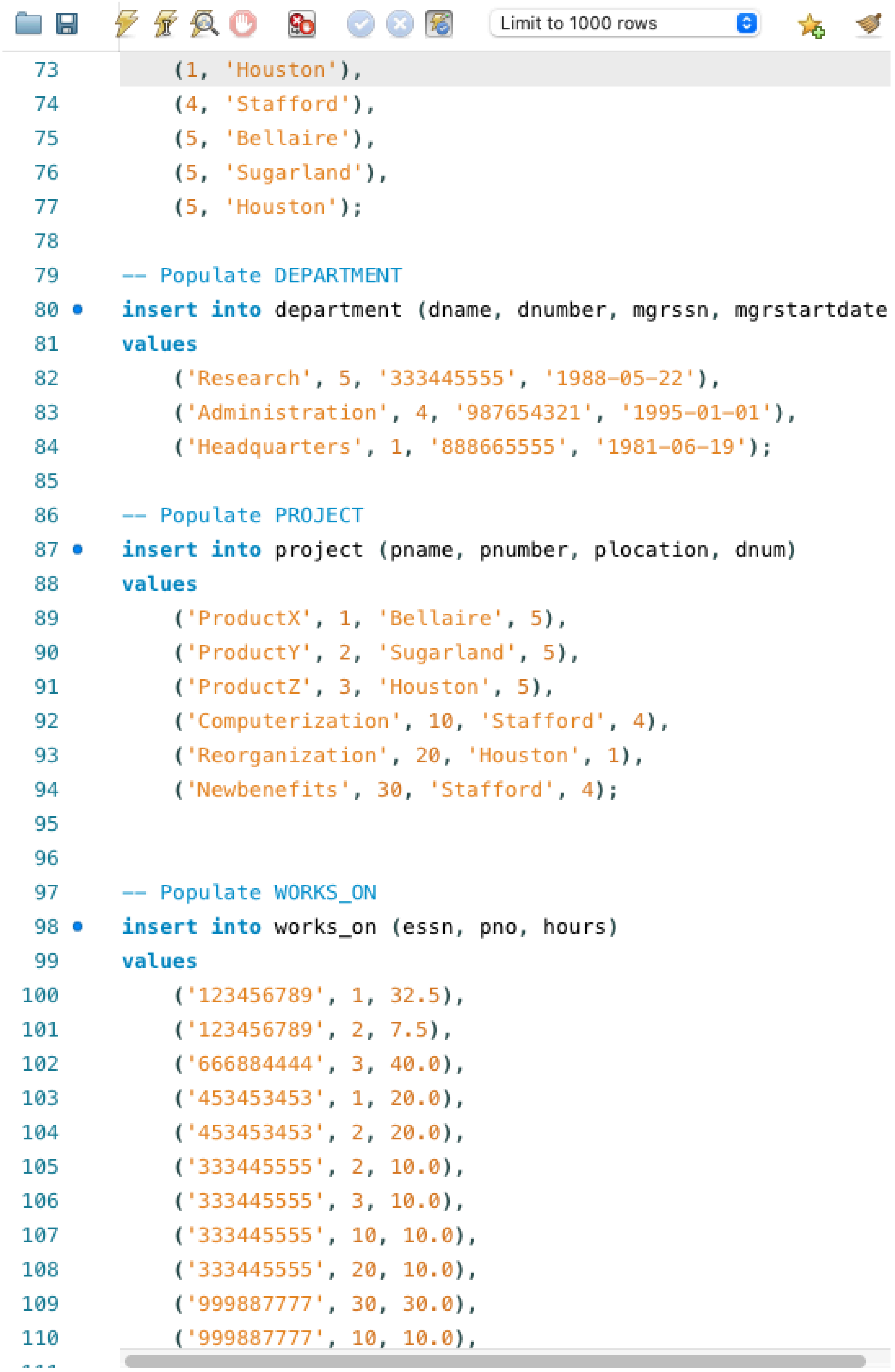
Build Database:



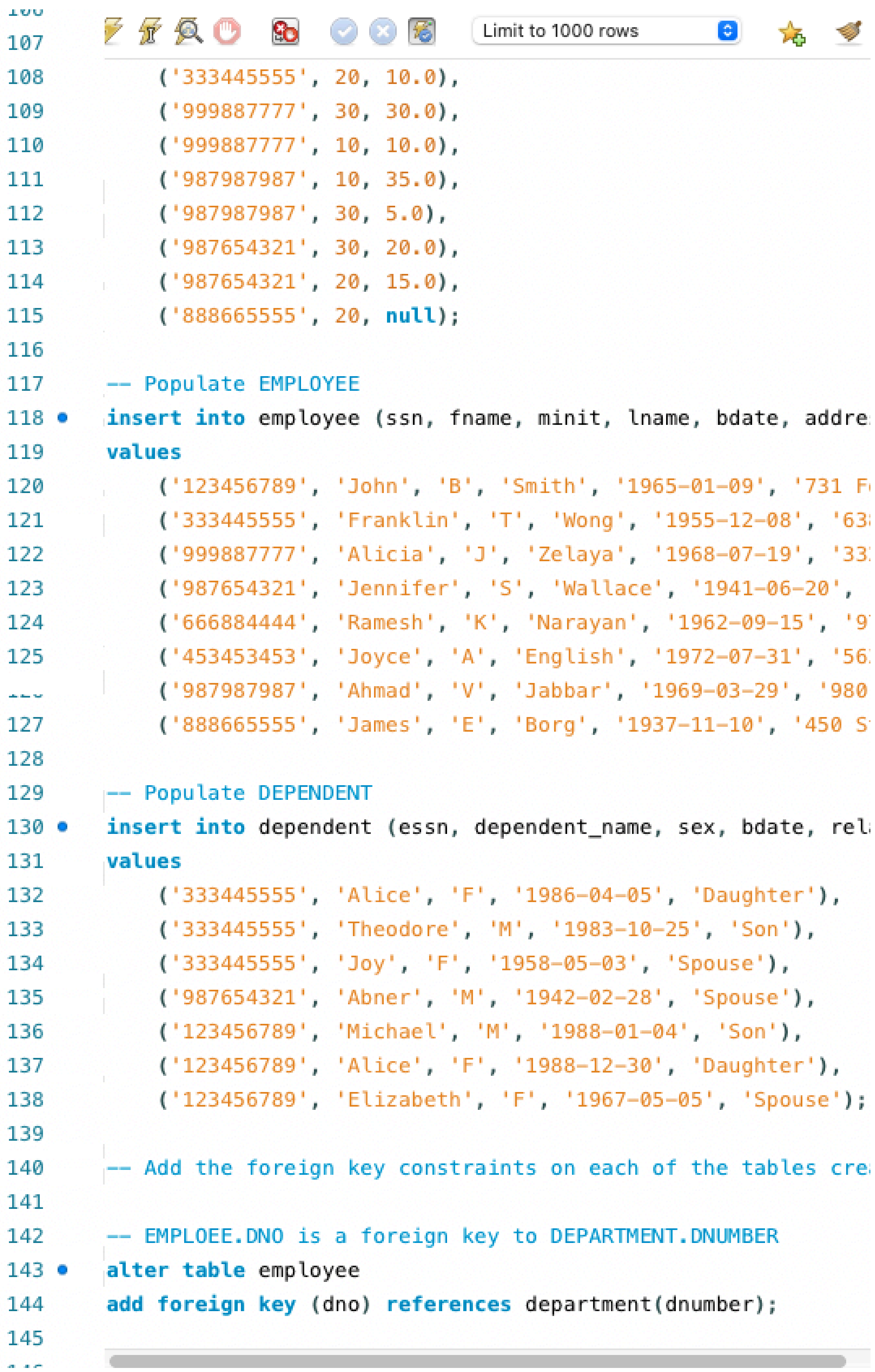
```
-- Remove tables if they exists to recreate them:w
set foreign_key_checks = 0;
drop table if exists department;
drop table if exists dependent;
drop table if exists dept_locations;
drop table if exists employee;
drop table if exists project;
drop table if exists works_on;
set foreign_key_checks=1;

-- Create tables with no foreign key constraints first.
create table dept_locations
(
    dnumber smallint not null,
    dlocation varchar(25) not null,
    primary key (dnumber, dlocation)
);

create table department
(
    dname varchar(25),
    dnumber smallint not null,
    mgrssn varchar(9),
    mgrstartdate date,
    primary key (dnumber)
);

create table employee
(
    ssn varchar(9) not null,
    fname varchar(25),
    minit varchar(1),
    lname varchar(25),
    bdate date,
    address varchar(50),
    sex enum('M', 'F'),
    salary int,
    superssn varchar(9),
```



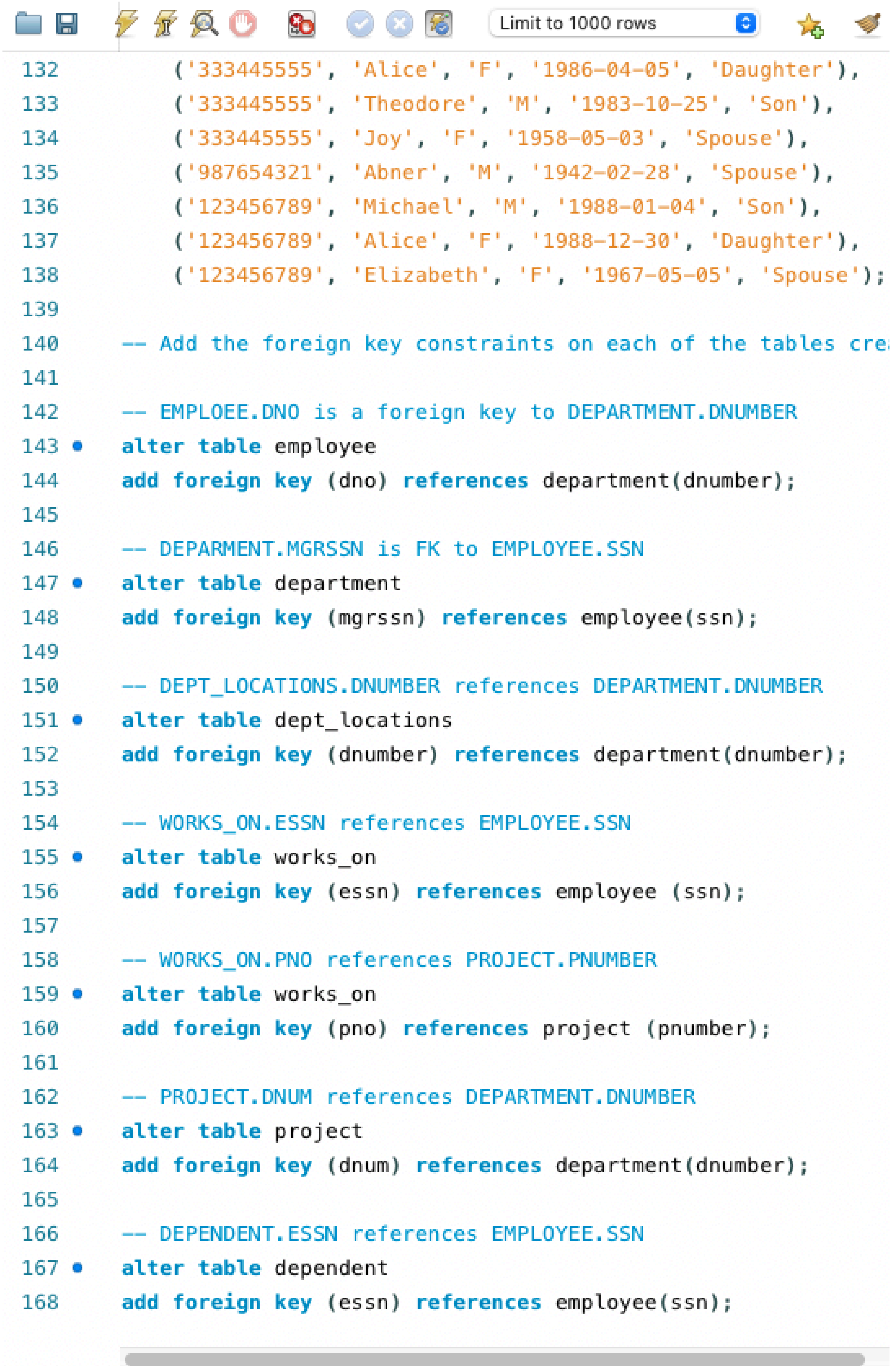
```
    primary key (ssn)
);

create table works_on
(
    essn varchar(9) not null,
    pno smallint not null,
    hours decimal(3, 1),
    primary key (essn, pno)
);

create table project
(
    pname varchar(25),
    pnumber smallint not null,
    plocation varchar(25),
    dnum smallint,
    primary key (pnumber)
);

create table dependent
(
    essn varchar(9),
    dependent_name varchar(25),
    sex enum('M', 'F'),
    bdate date,
    relationship varchar(25),
    primary key (essn, dependent_name)
);

-- Populate DEPT_LOCATIONS
insert into dept_locations (dnumber, dlocation)
values
    (1, 'Houston'),
    (4, 'Stafford'),
    (5, 'Bellaire'),
    (5, 'Sugarland'),
    (5, 'Houston');
```



```
    (1, 'Houston'),
    (4, 'Stafford'),
    (5, 'Bellaire'),
    (5, 'Sugarland'),
    (5, 'Houston');

-- Populate DEPARTMENT
insert into department (dname, dnumber, mgrssn, mgrstartdate
values
    ('Research', 5, '333445555', '1988-05-22'),
    ('Administration', 4, '987654321', '1995-01-01'),
    ('Headquarters', 1, '888665555', '1981-06-19');

-- Populate PROJECT
insert into project (pname, pnumber, plocation, dnum)
values
    ('ProductX', 1, 'Bellaire', 5),
    ('ProductY', 2, 'Sugarland', 5),
    ('ProductZ', 3, 'Houston', 5),
    ('Computerization', 10, 'Stafford', 4),
    ('Reorganization', 20, 'Houston', 1),
    ('Newbenefits', 30, 'Stafford', 4);


-- Populate WORKS_ON
insert into works_on (essn, pno, hours)
values
    ('123456789', 1, 32.5),
    ('123456789', 2, 7.5),
    ('666884444', 3, 40.0),
    ('453453453', 1, 20.0),
    ('453453453', 2, 20.0),
    ('333445555', 2, 10.0),
    ('333445555', 3, 10.0),
    ('333445555', 10, 10.0),
    ('333445555', 20, 10.0),
    ('999887777', 30, 30.0),
    ('999887777', 10, 10.0),
```

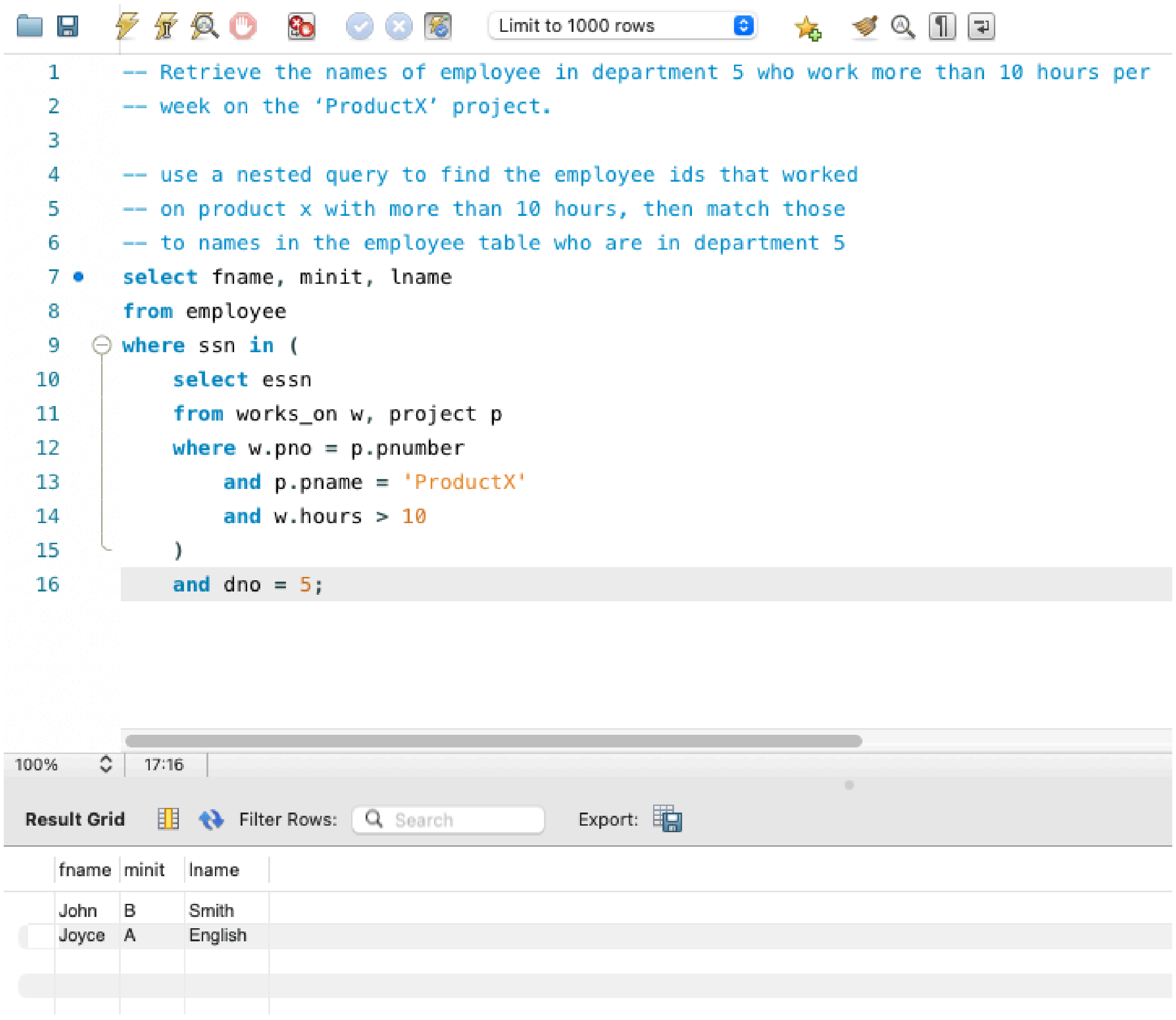


```
    ('333445555', 20, 10.0),
    ('999887777', 30, 30.0),
    ('999887777', 10, 10.0),
    ('987987987', 10, 35.0),
    ('987987987', 30, 5.0),
    ('987654321', 30, 20.0),
    ('987654321', 20, 15.0),
    ('888665555', 20, null);

-- Populate EMPLOYEE
insert into employee (ssn, fname, minit, lname, bdate, addre
values
    ('123456789', 'John', 'B', 'Smith', '1965-01-09', '731 F
    ('333445555', 'Franklin', 'T', 'Wong', '1955-12-08', '63
    ('999887777', 'Alicia', 'J', 'Zelaya', '1968-07-19', '33
    ('987654321', 'Jennifer', 'S', 'Wallace', '1941-06-20',
    ('666884444', 'Ramesh', 'K', 'Narayan', '1962-09-15', '9
    ('453453453', 'Joyce', 'A', 'English', '1972-07-31', '56
    ('987987987', 'Ahmad', 'V', 'Jabbar', '1969-03-29', '980
    ('888665555', 'James', 'E', 'Borg', '1937-11-10', '450 S

-- Populate DEPENDENT
insert into dependent (essn, dependent_name, sex, bdate, rel
values
    ('333445555', 'Alice', 'F', '1986-04-05', 'Daughter'),
    ('333445555', 'Theodore', 'M', '1983-10-25', 'Son'),
    ('333445555', 'Joy', 'F', '1958-05-03', 'Spouse'),
    ('987654321', 'Abner', 'M', '1942-02-28', 'Spouse'),
    ('123456789', 'Michael', 'M', '1988-01-04', 'Son'),
    ('123456789', 'Alice', 'F', '1988-12-30', 'Daughter'),
    ('123456789', 'Elizabeth', 'F', '1967-05-05', 'Spouse');

-- Add the foreign key constraints on each of the tables cre

-- EMPLOEE.DNO is a foreign key to DEPARTMENT.DNUMBER
alter table employee
add foreign key (dno) references department(dnumber);

```



```
        ('333445555', 'Alice', 'F', '1986-04-05', 'Daughter'),
        ('333445555', 'Theodore', 'M', '1983-10-25', 'Son'),
        ('333445555', 'Joy', 'F', '1958-05-03', 'Spouse'),
        ('987654321', 'Abner', 'M', '1942-02-28', 'Spouse'),
        ('123456789', 'Michael', 'M', '1988-01-04', 'Son'),
        ('123456789', 'Alice', 'F', '1988-12-30', 'Daughter'),
        ('123456789', 'Elizabeth', 'F', '1967-05-05', 'Spouse');

-- Add the foreign key constraints on each of the tables cre

-- EMPLOEE.DNO is a foreign key to DEPARTMENT.DNUMBER
alter table employee
add foreign key (dno) references department(dnumber);

-- DEPARMENT.MGRSSN is FK to EMPLOYEE.SSN
alter table department
add foreign key (mgrssn) references employee(ssn);

-- DEPT_LOCATIONS.DNUMBER references DEPARTMENT.DNUMBER
alter table dept_locations
add foreign key (dnumber) references department(dnumber);

-- WORKS_ON.ESSN references EMPLOYEE.SSN
alter table works_on
add foreign key (essn) references employee (ssn);

-- WORKS_ON.PNO references PROJECT.PNUMBER
alter table works_on
add foreign key (pno) references project (pnumber);

-- PROJECT.DNUM references DEPARTMENT.DNUMBER
alter table project
add foreign key (dnum) references department(dnumber);

-- DEPENDENT.ESSN references EMPLOYEE.SSN
alter table dependent
add foreign key (essn) references employee(ssn);
```

(a) Retrieve the names of employees in department 5 who work more than 10 hours per week on the 'ProductX' project.



Limit to 1000 rows

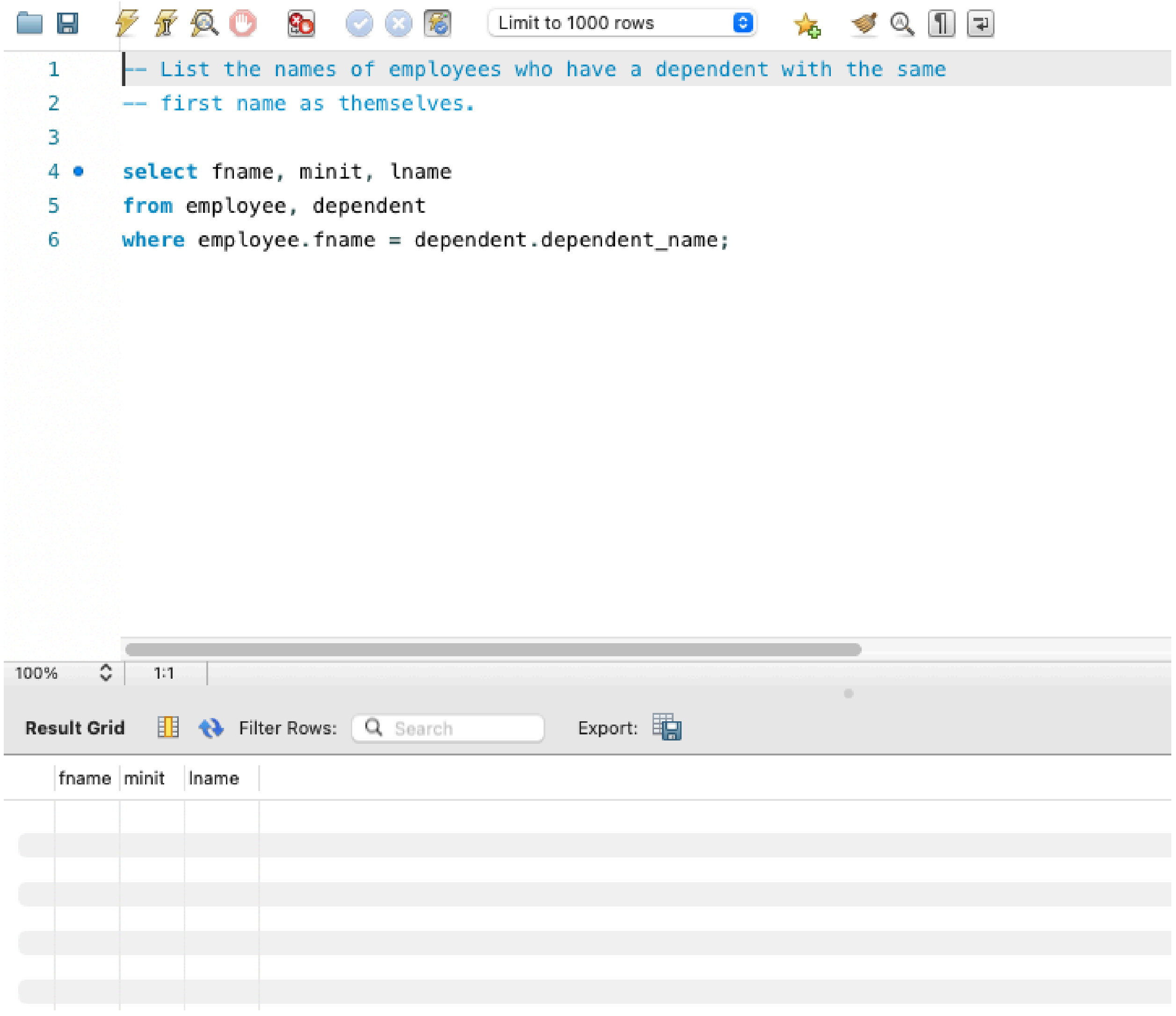
```sql
-- Retrieve the names of employee in department 5 who work more than 10 hours per
-- week on the 'ProductX' project.

-- use a nested query to find the employee ids that worked
-- on product x with more than 10 hours, then match those
-- to names in the employee table who are in department 5
select fname, minit, lname
from employee
where ssn in (
    select essn
    from works_on w, project p
    where w.pno = p.pnumber
        and p.pname = 'ProductX'
        and w.hours > 10
    )
    and dno = 5;
```

100% 17:16

Result Grid Filter Rows: Search Export:

fname	minit	lname
John	B	Smith
Joyce	A	English

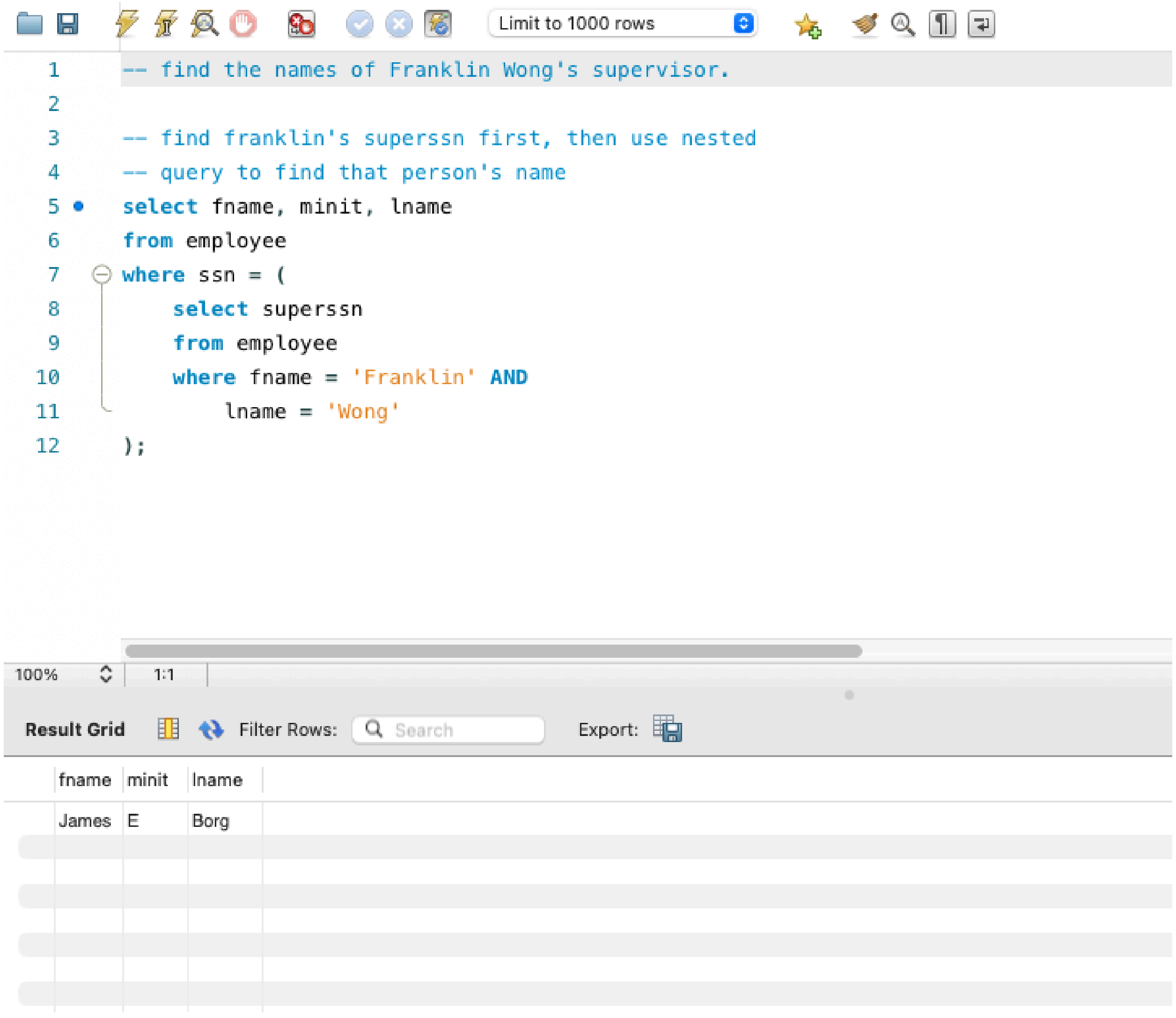
(b) List the names of employees who have a dependent with the same first name as themselves.



```sql
-- List the names of employees who have a dependent with the same
-- first name as themselves.

select fname, minit, lname
from employee, dependent
where employee.fname = dependent.dependent_name;
```

fname	minit	lname

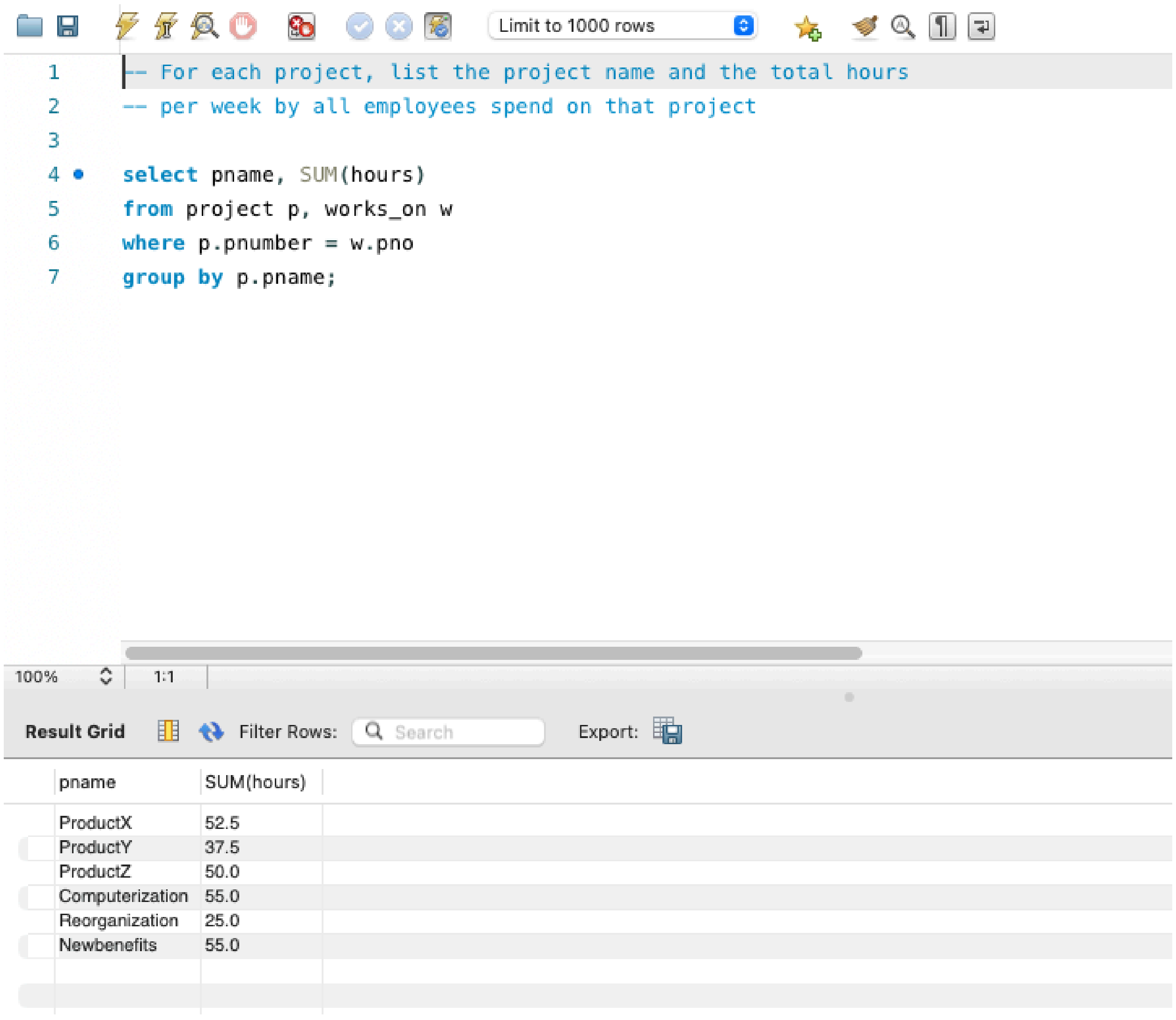
(c) Find the names of Franklin Wong's supervisor.



```sql
-- find the names of Franklin Wong's supervisor.

-- find franklin's superssn first, then use nested
-- query to find that person's name
select fname, minit, lname
from employee
where ssn = (
    select superssn
    from employee
    where fname = 'Franklin' AND
        lname = 'Wong'
);
```

fname	minit	lname
James	E	Borg

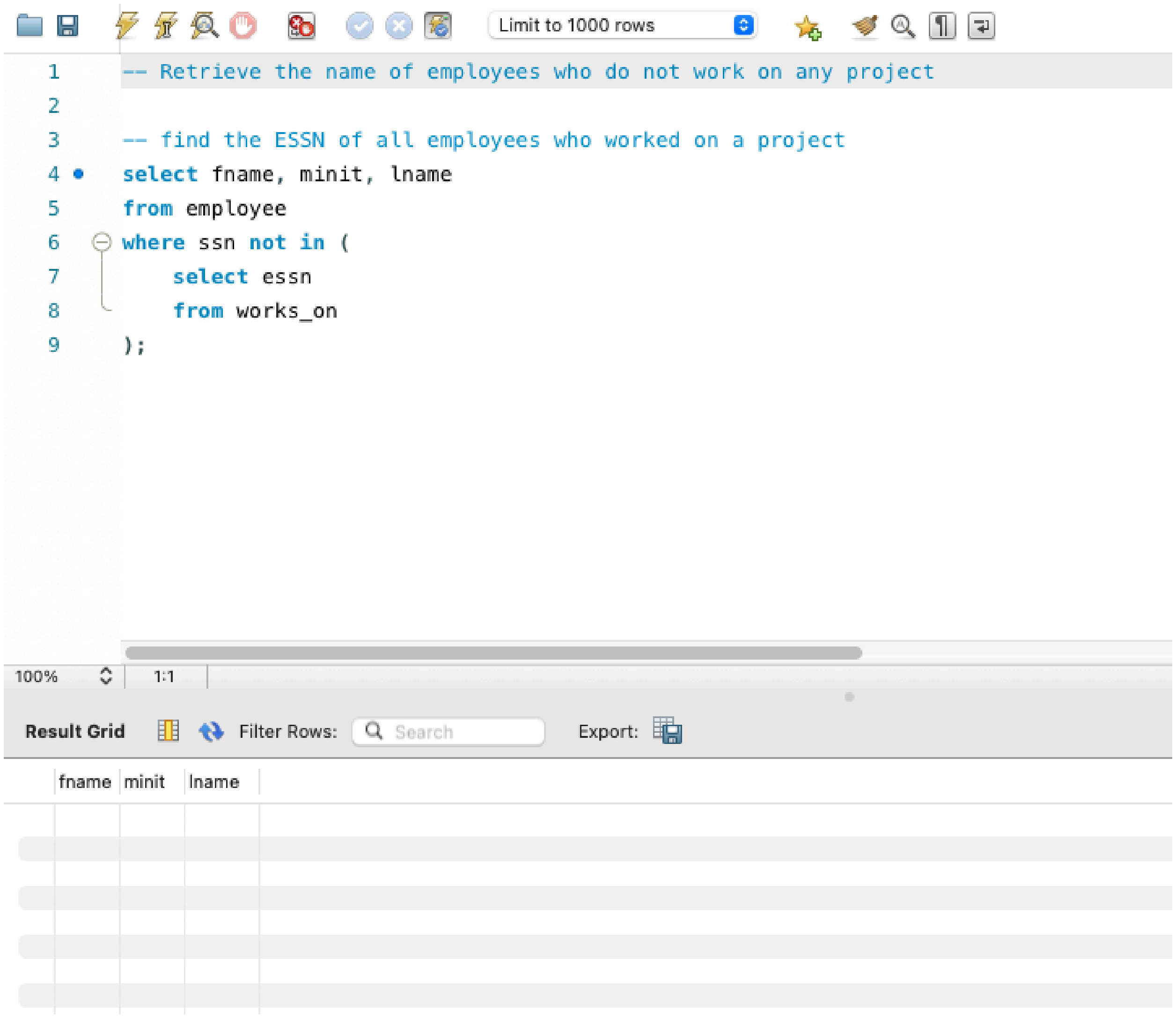
(d) For each project, list the project name and the total hours per week (by all employees) spent on that project.



```sql
-- For each project, list the project name and the total hours
-- per week by all employees spend on that project

select pname, SUM(hours)
from project p, works_on w
where p.pnumber = w.pno
group by p.pname;
```

pname	SUM(hours)
ProductX	52.5
ProductY	37.5
ProductZ	50.0
Computerization	55.0
Reorganization	25.0
Newbenefits	55.0

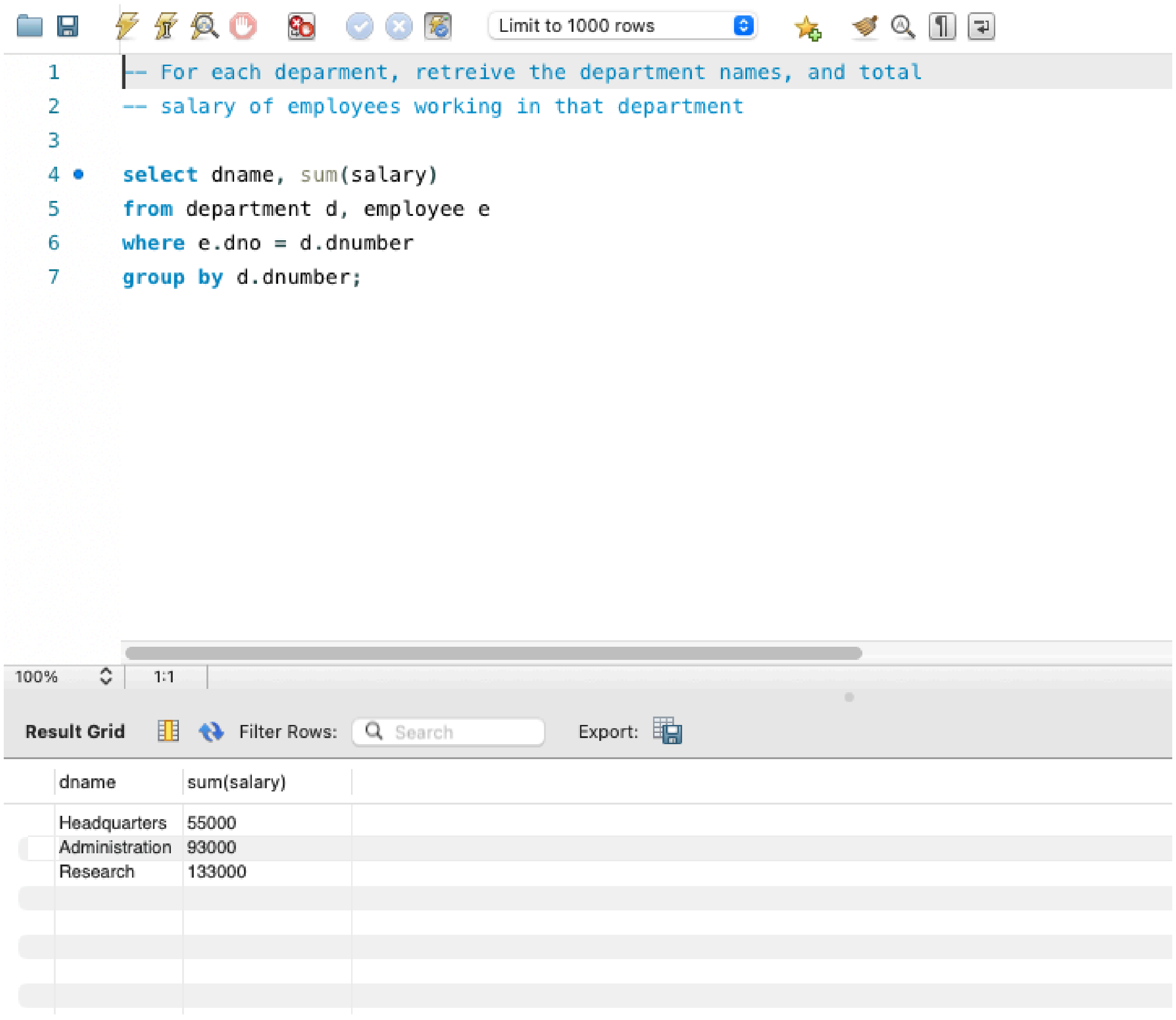
(f) Retrieve the names of employees who do not work on any project.



```
-- Retrieve the name of employees who do not work on any project

-- find the ESSN of all employees who worked on a project
select fname, minit, lname
from employee
where ssn not in (
    select essn
    from works_on
);
```

```
fname | minit | lname
```

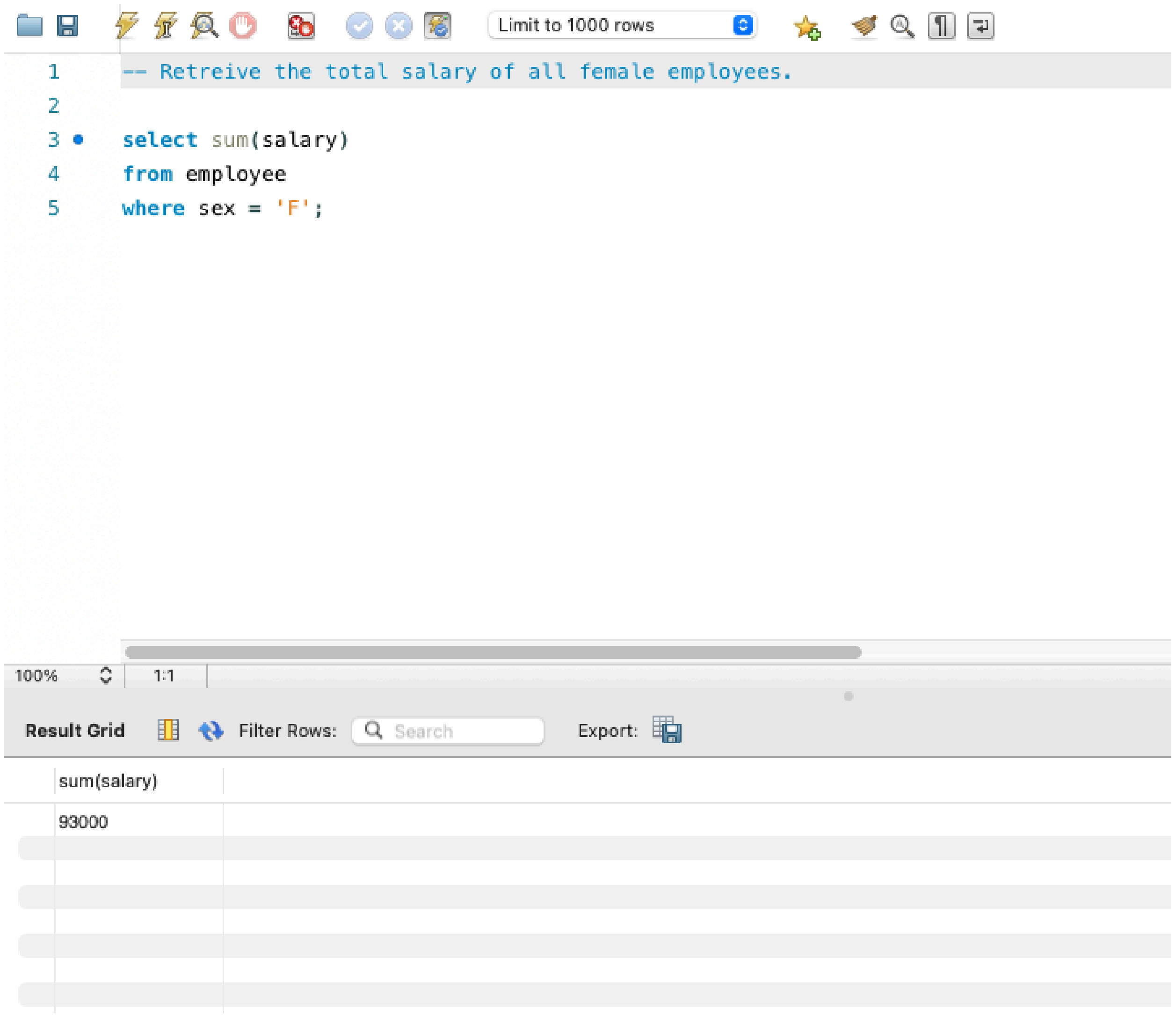
(g) For each department, retrieve the department name, and the total salary of employees working in that department.



```
-- For each deparment, retreive the department names, and total
-- salary of employees working in that department

select dname, sum(salary)
from department d, employee e
where e.dno = d.dnumber
group by d.dnumber;
```

```
dname           sum(salary)
Headquarters    55000
Administration  93000
Research        133000
```

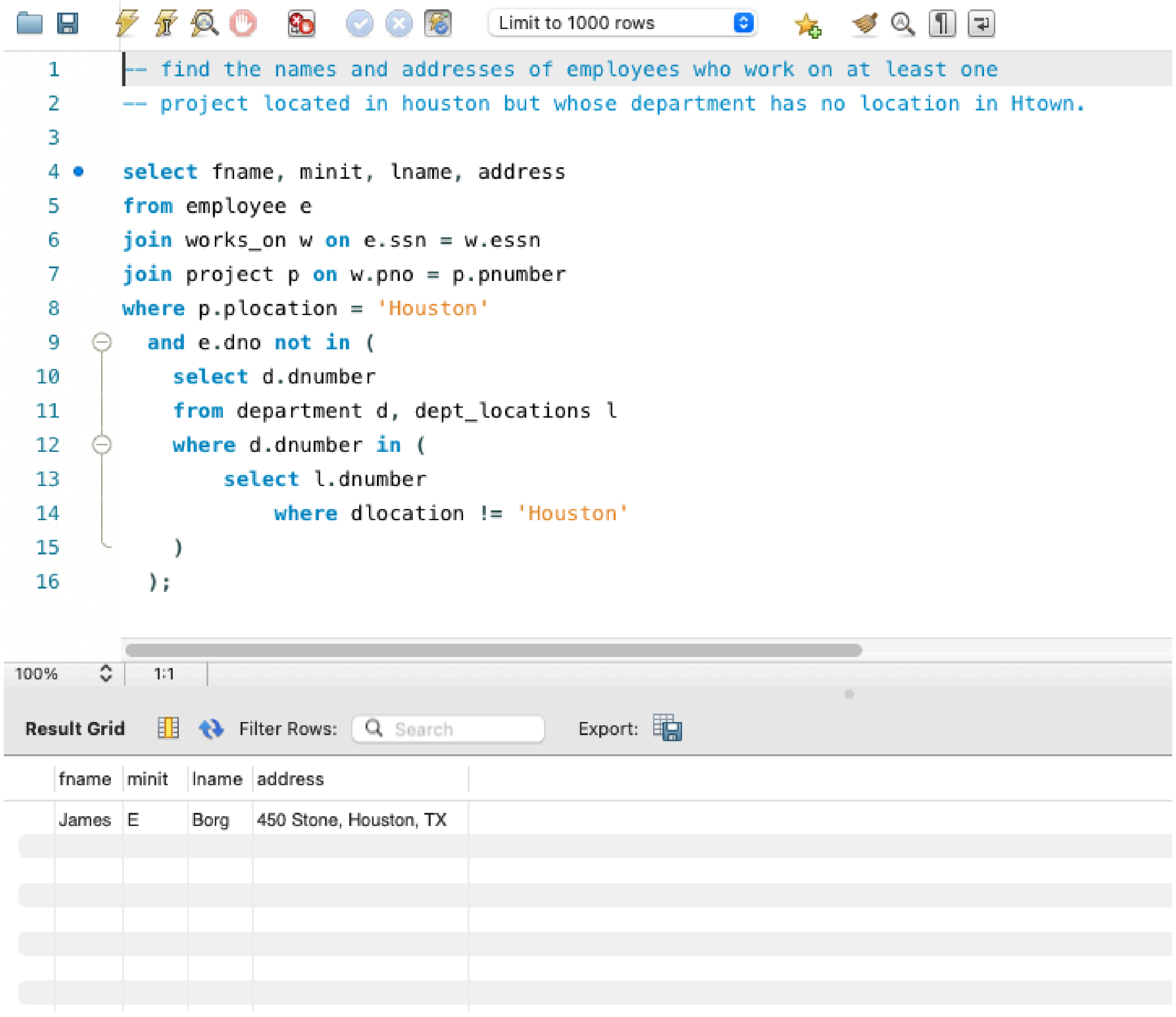
(h) Retrieve the total salary of all female employees.



```
-- Retreive the total salary of all female employees.

select sum(salary)
from employee
where sex = 'F';
```

```
sum(salary)
93000
```

(i) Find the names and addresses of employees who work on at least one project located in Houston but whose department has no location in Houston.



Limit to 1000 rows

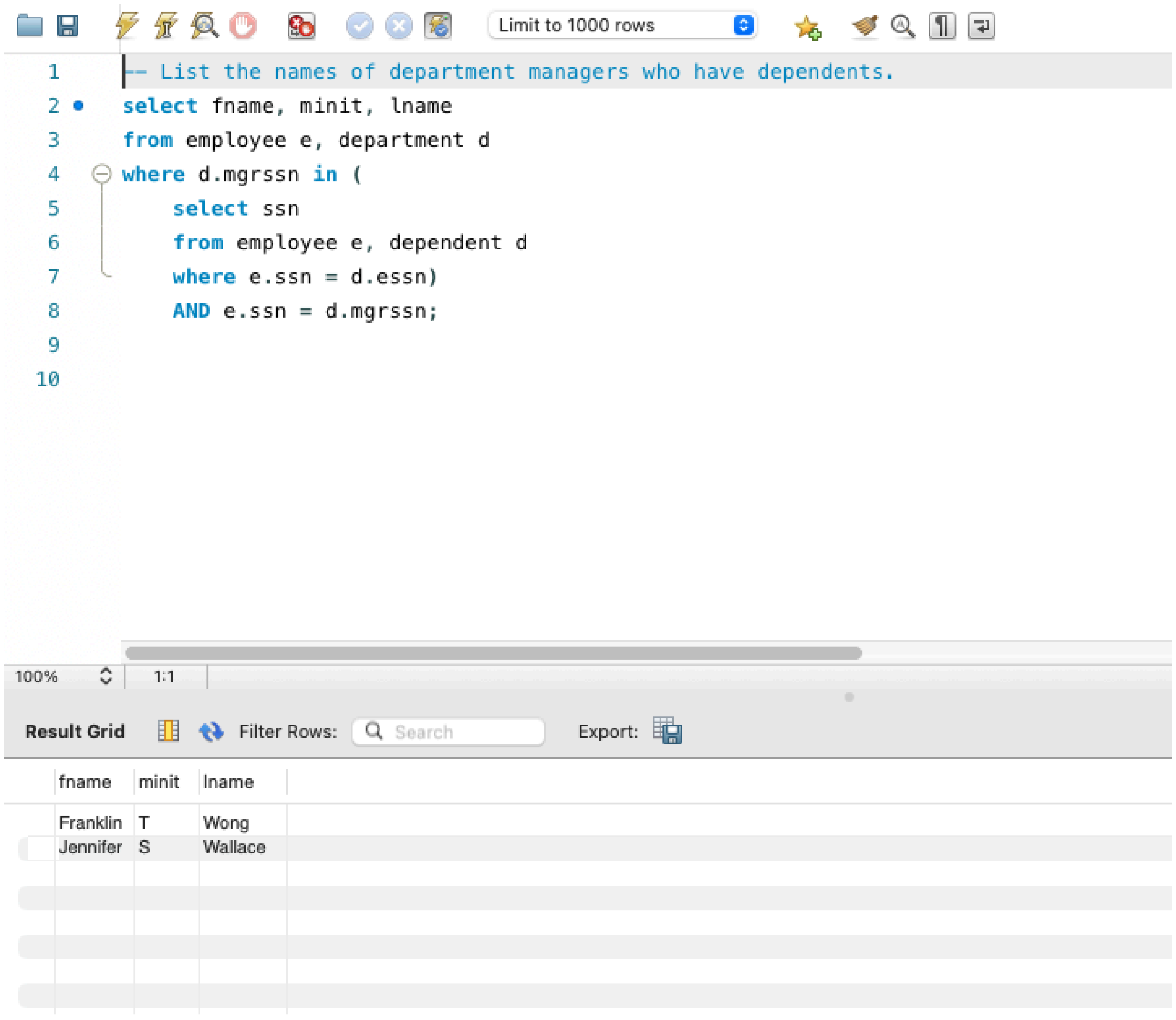
```
-- find the names and addresses of employees who work on at least one
-- project located in houston but whose department has no location in Htown.

select fname, minit, lname, address
from employee e
join works_on w on e.ssn = w.essn
join project p on w.pno = p.pnumber
where p.plocation = 'Houston'
  and e.dno not in (
    select d.dnumber
    from department d, dept_locations l
    where d.dnumber in (
        select l.dnumber
            where dlocation != 'Houston'
    )
  );
```

100% 1:1

Result Grid Filter Rows: Search Export:

fname	minit	lname	address
James	E	Borg	450 Stone, Houston, TX

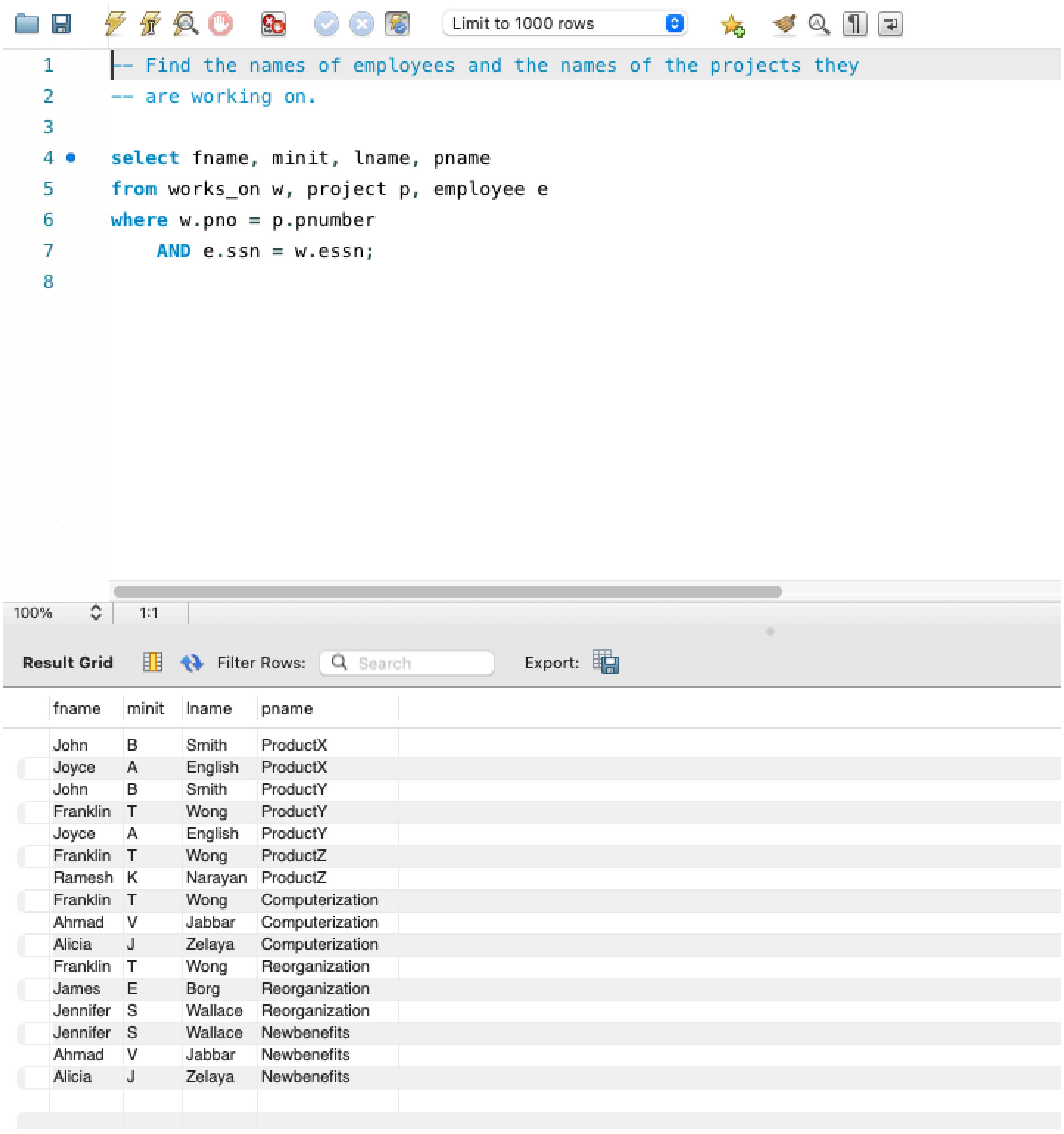
(j) List the last names of department managers who have dependents.



```sql
-- List the names of department managers who have dependents.
select fname, minit, lname
from employee e, department d
where d.mgrssn in (
    select ssn
    from employee e, dependent d
    where e.ssn = d.essn)
    AND e.ssn = d.mgrssn;
```

fname	minit	lname
Franklin	T	Wong
Jennifer	S	Wallace

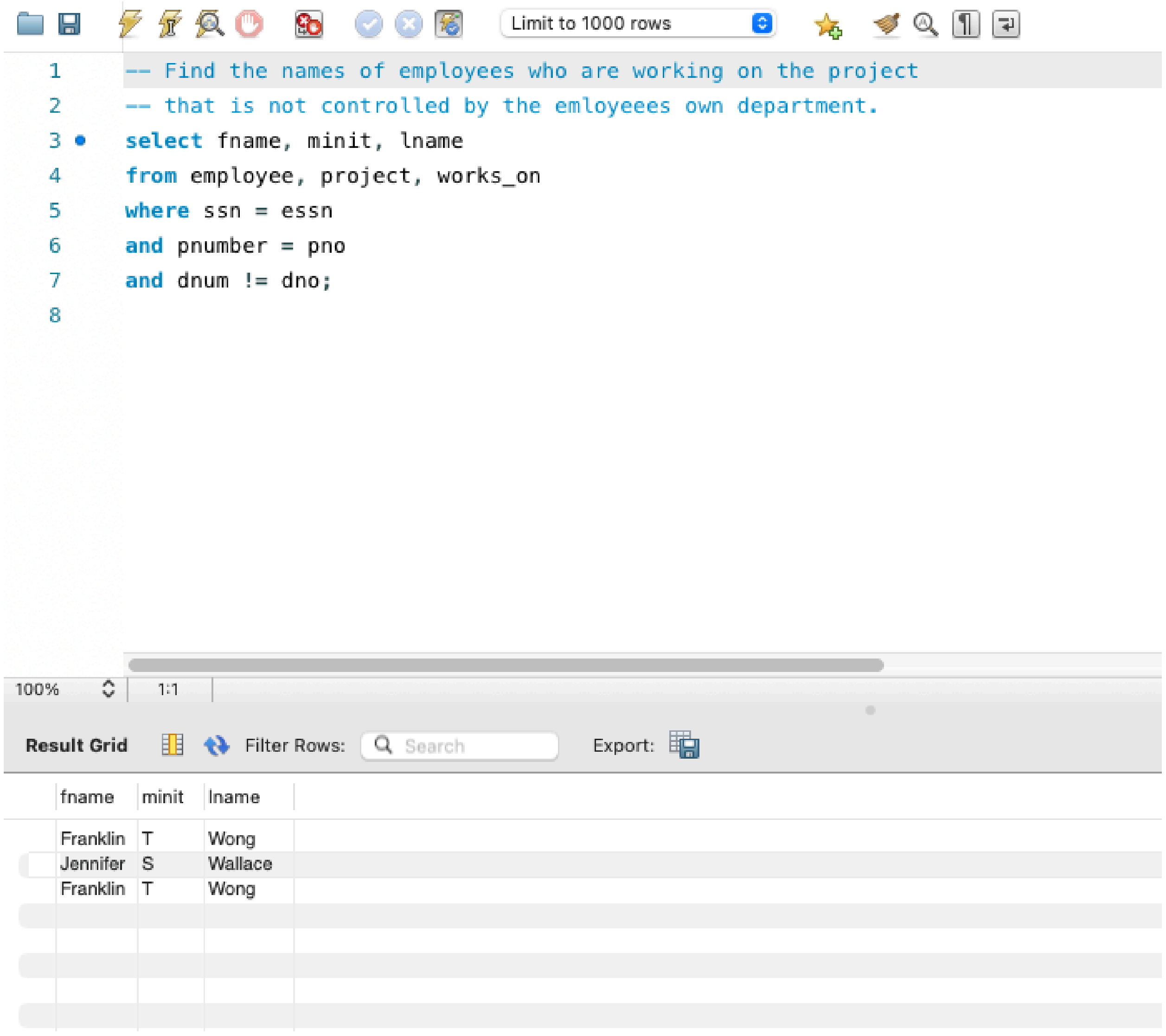
(k) Find the names of employees and the names of the projects that the employees are working on.



```sql
-- Find the names of employees and the names of the projects they
-- are working on.

select fname, minit, lname, pname
from works_on w, project p, employee e
where w.pno = p.pnumber
    AND e.ssn = w.essn;
```

fname	minit	lname	pname
John	B	Smith	ProductX
Joyce	A	English	ProductX
John	B	Smith	ProductY
Franklin	T	Wong	ProductY
Joyce	A	English	ProductY
Franklin	T	Wong	ProductZ
Ramesh	K	Narayan	ProductZ
Franklin	T	Wong	Computerization
Ahmad	V	Jabbar	Computerization
Alicia	J	Zelaya	Computerization
Franklin	T	Wong	Reorganization
James	E	Borg	Reorganization
Jennifer	S	Wallace	Reorganization
Jennifer	S	Wallace	Newbenefits
Ahmad	V	Jabbar	Newbenefits
Alicia	J	Zelaya	Newbenefits

(l) Find the names of employees who are working on the project that is not controlled by the employee's own department.



```sql
-- Find the names of employees who are working on the project
-- that is not controlled by the emloyeees own department.
select fname, minit, lname
from employee, project, works_on
where ssn = essn
and pnumber = pno
and dnum != dno;
```

fname	minit	lname
Franklin	T	Wong
Jennifer	S	Wallace
Franklin	T	Wong