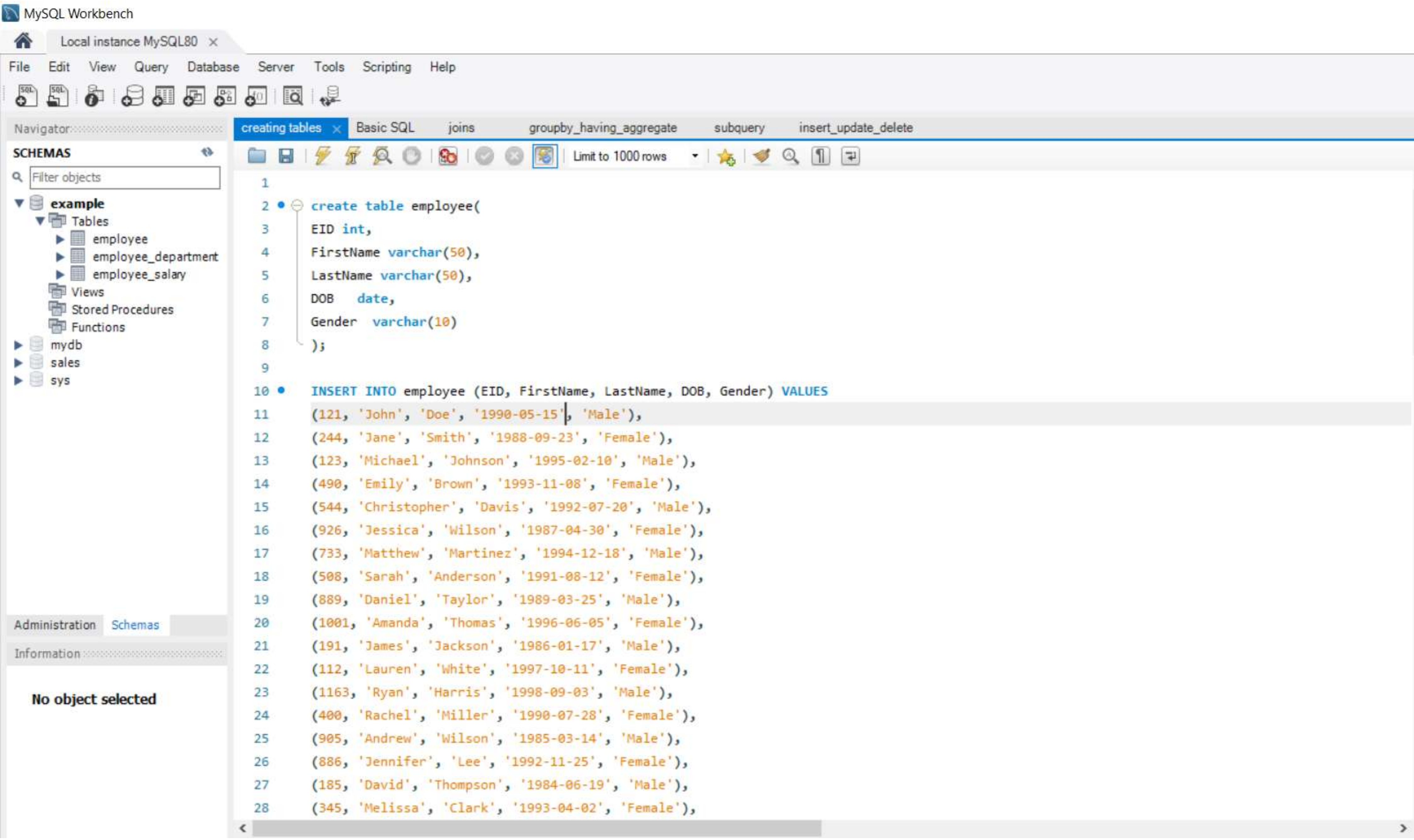
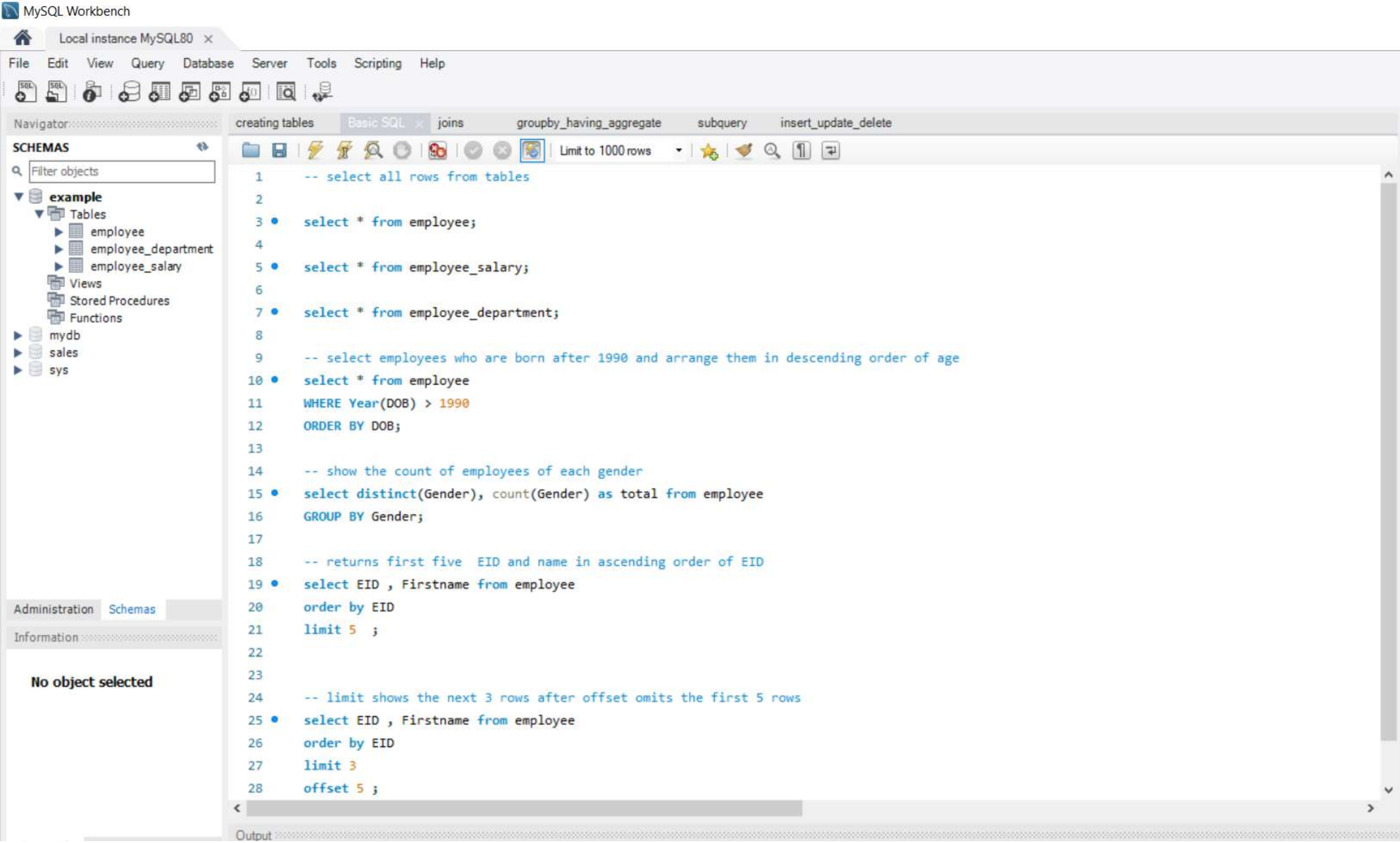
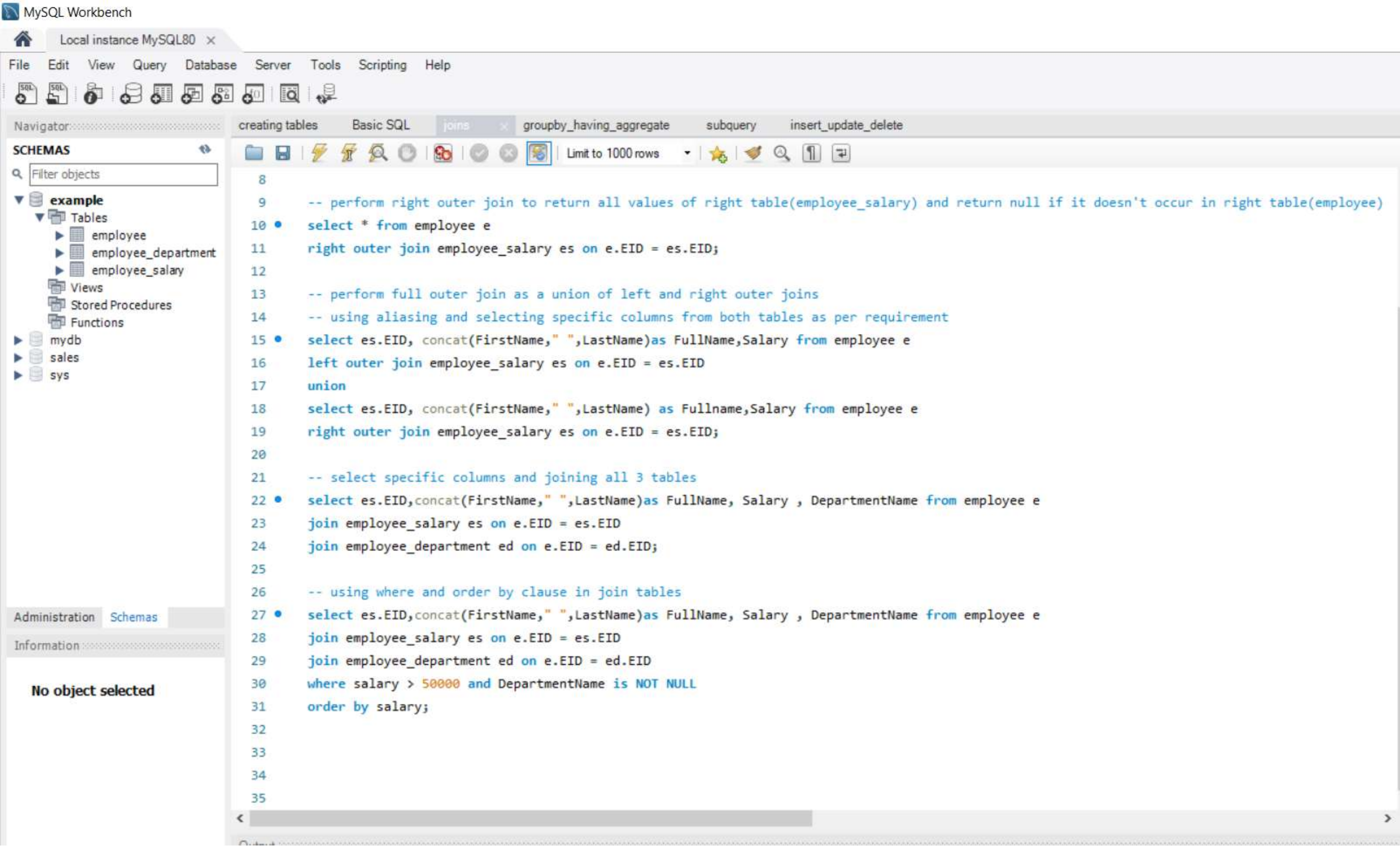


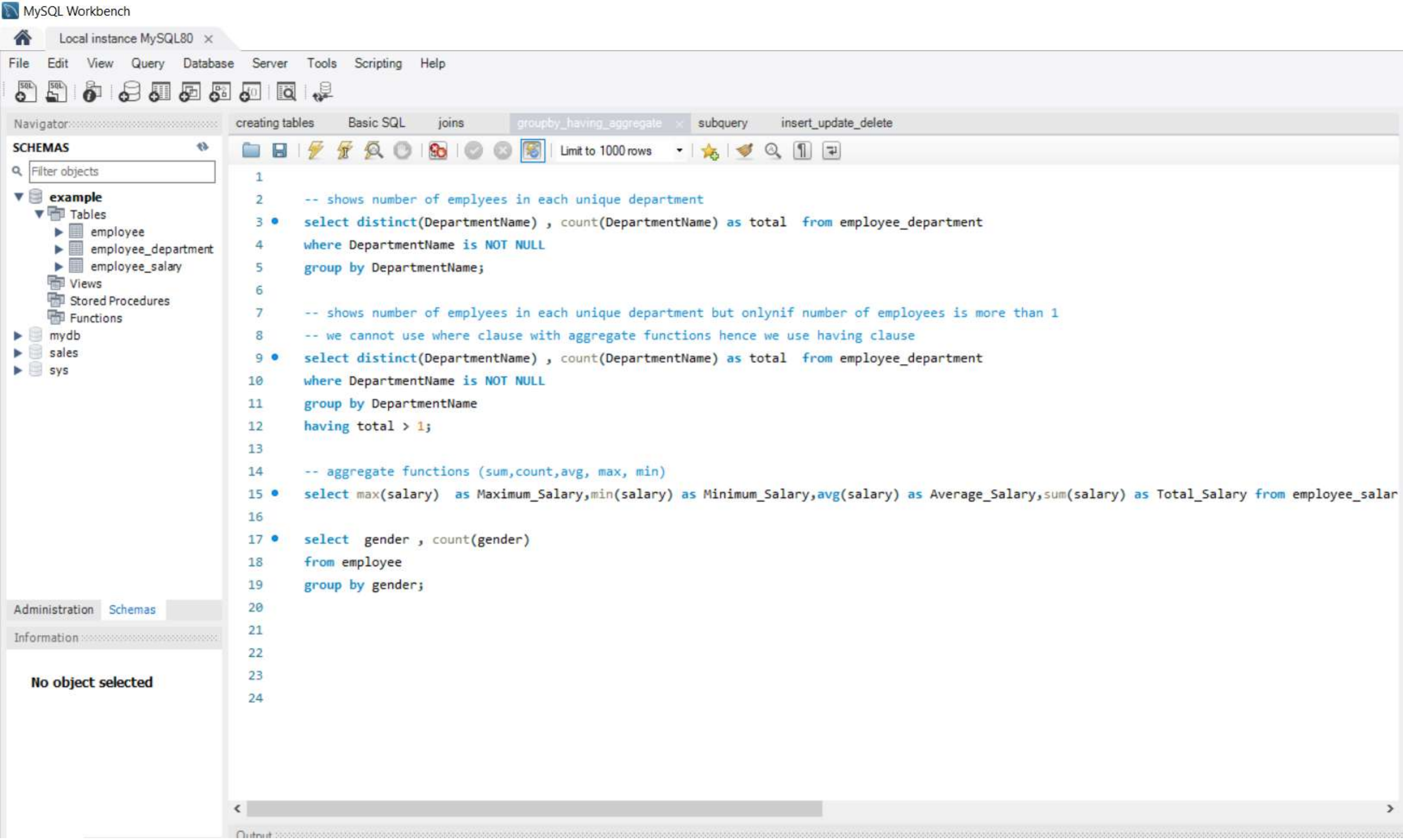
```sql

create table employee(
EID int,
FirstName varchar(50),
LastName varchar(50),
DOB  date,
Gender  varchar(10)
);

INSERT INTO employee (EID, FirstName, LastName, DOB, Gender) VALUES
(121, 'John', 'Doe', '1990-05-15', 'Male'),
(244, 'Jane', 'Smith', '1988-09-23', 'Female'),
(123, 'Michael', 'Johnson', '1995-02-10', 'Male'),
(490, 'Emily', 'Brown', '1993-11-08', 'Female'),
(544, 'Christopher', 'Davis', '1992-07-20', 'Male'),
(926, 'Jessica', 'Wilson', '1987-04-30', 'Female'),
(733, 'Matthew', 'Martinez', '1994-12-18', 'Male'),
(508, 'Sarah', 'Anderson', '1991-08-12', 'Female'),
(889, 'Daniel', 'Taylor', '1989-03-25', 'Male'),
(1001, 'Amanda', 'Thomas', '1996-06-05', 'Female'),
(191, 'James', 'Jackson', '1986-01-17', 'Male'),
(112, 'Lauren', 'White', '1997-10-11', 'Female'),
(1163, 'Ryan', 'Harris', '1998-09-03', 'Male'),
(400, 'Rachel', 'Miller', '1990-07-28', 'Female'),
(905, 'Andrew', 'Wilson', '1985-03-14', 'Male'),
(886, 'Jennifer', 'Lee', '1992-11-25', 'Female'),
(185, 'David', 'Thompson', '1984-06-19', 'Male'),
(345, 'Melissa', 'Clark', '1993-04-02', 'Female'),
```



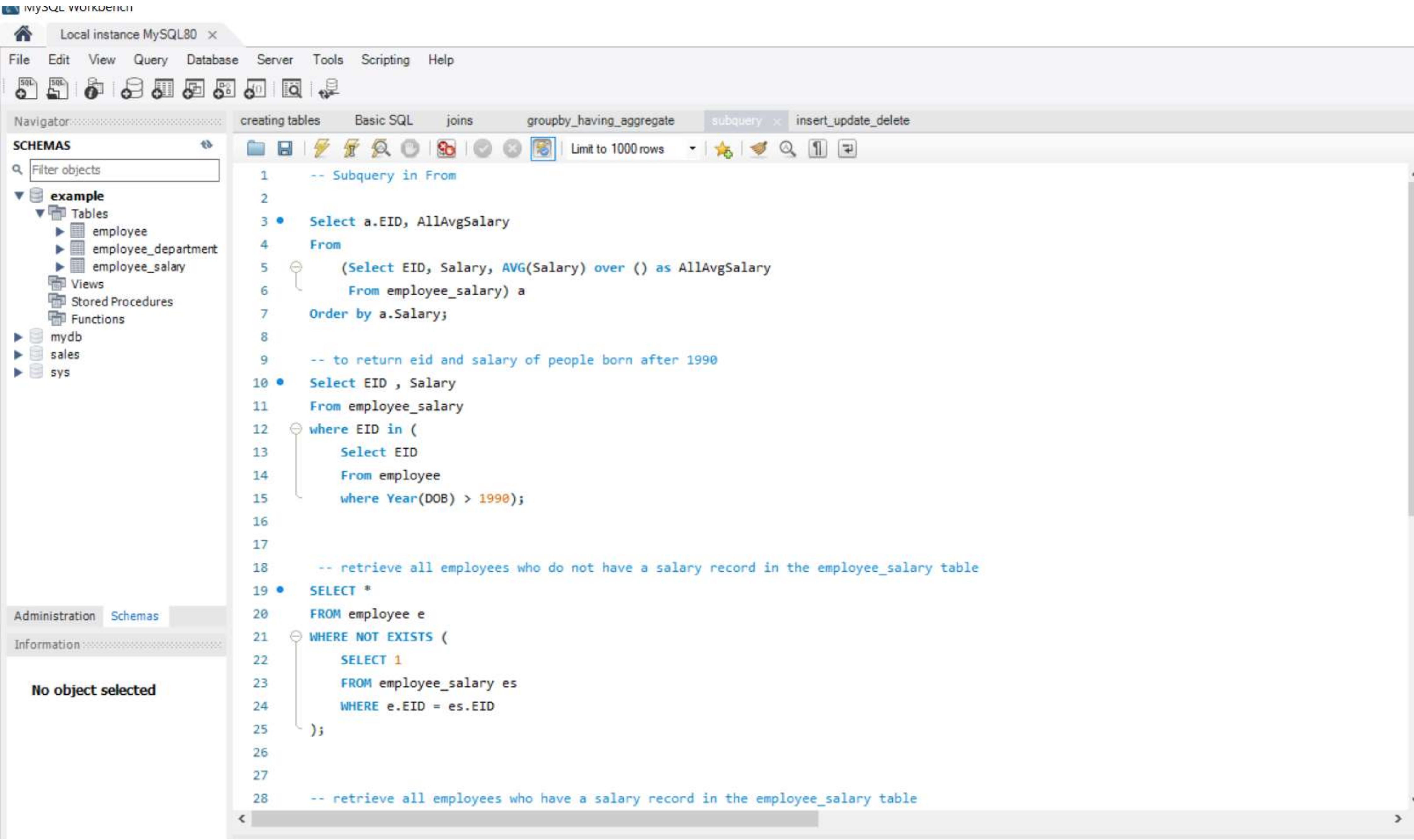
```
-- select all rows from tables

select * from employee;

select * from employee_salary;

select * from employee_department;

-- select employees who are born after 1990 and arrange them in descending order of age
select * from employee
WHERE Year(DOB) > 1990
ORDER BY DOB;

-- show the count of employees of each gender
select distinct(Gender), count(Gender) as total from employee
GROUP BY Gender;

-- returns first five  EID and name in ascending order of EID
select EID , Firstname from employee
order by EID
limit 5  ;


-- limit shows the next 3 rows after offset omits the first 5 rows
select EID , Firstname from employee
order by EID
limit 3
offset 5 ;
```



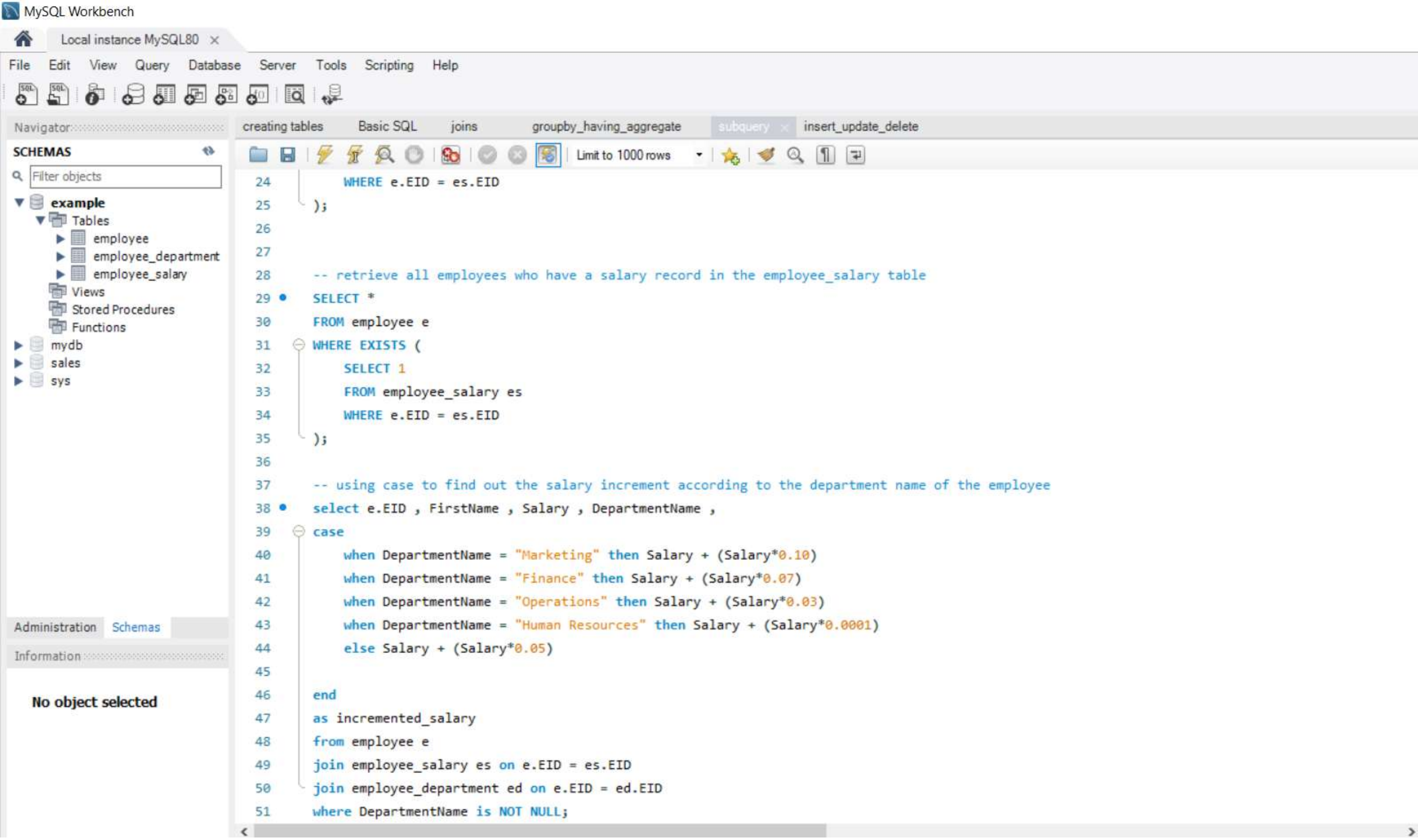
```sql

-- perform right outer join to return all values of right table(employee_salary) and return null if it doesn't occur in right table(employee)
select * from employee e
right outer join employee_salary es on e.EID = es.EID;

-- perform full outer join as a union of left and right outer joins
-- using aliasing and selecting specific columns from both tables as per requirement
select es.EID, concat(FirstName," ",LastName)as FullName,Salary from employee e
left outer join employee_salary es on e.EID = es.EID
union
select es.EID, concat(FirstName," ",LastName) as Fullname,Salary from employee e
right outer join employee_salary es on e.EID = es.EID;

-- select specific columns and joining all 3 tables
select es.EID,concat(FirstName," ",LastName)as FullName, Salary , DepartmentName from employee e
join employee_salary es on e.EID = es.EID
join employee_department ed on e.EID = ed.EID;

-- using where and order by clause in join tables
select es.EID,concat(FirstName," ",LastName)as FullName, Salary , DepartmentName from employee e
join employee_salary es on e.EID = es.EID
join employee_department ed on e.EID = ed.EID
where salary > 50000 and DepartmentName is NOT NULL
order by salary;
```



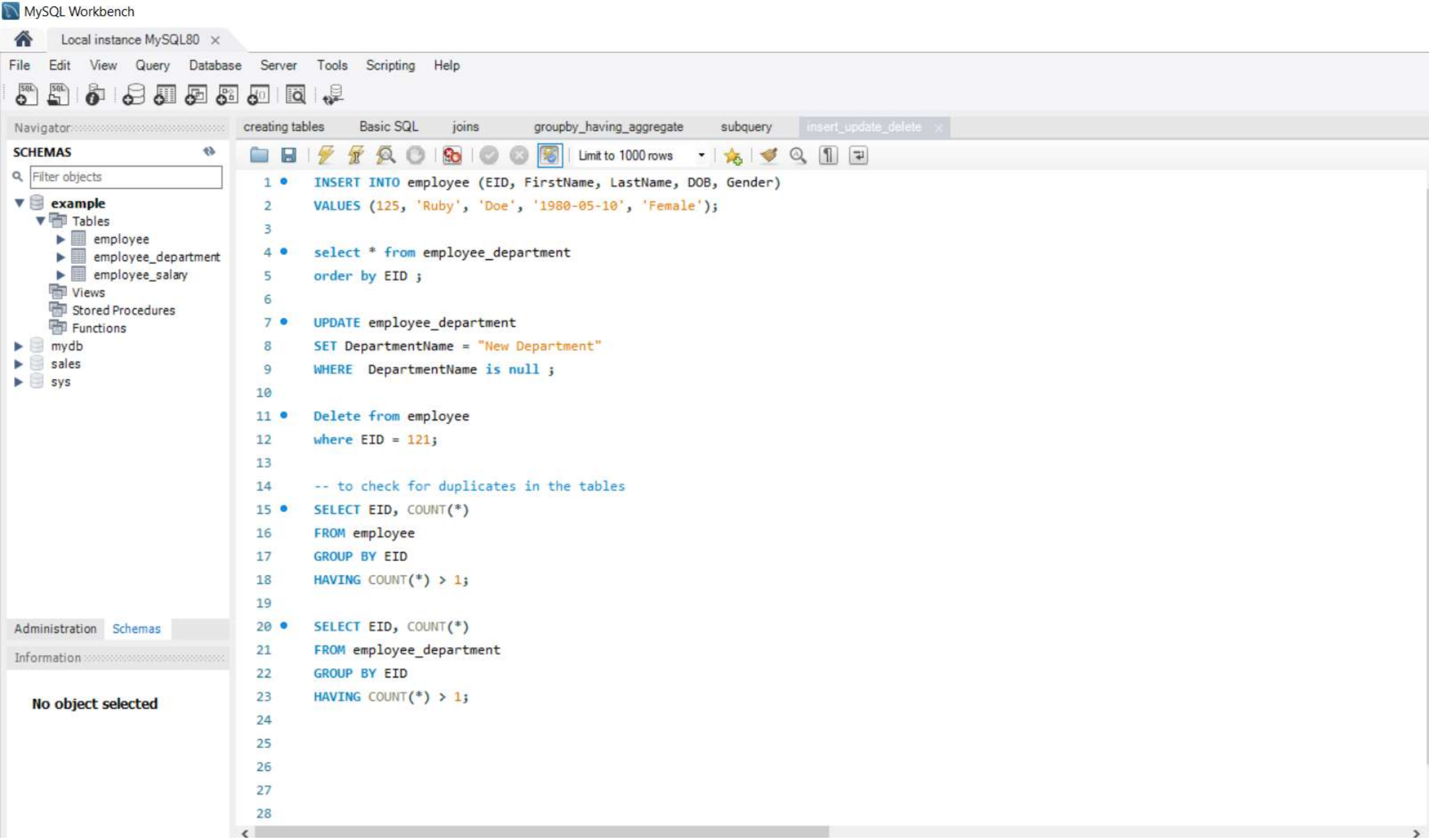
```

-- shows number of emplyees in each unique department
select distinct(DepartmentName) , count(DepartmentName) as total  from employee_department
where DepartmentName is NOT NULL
group by DepartmentName;

-- shows number of emplyees in each unique department but onlynif number of employees is more than 1
-- we cannot use where clause with aggregate functions hence we use having clause
select distinct(DepartmentName) , count(DepartmentName) as total  from employee_department
where DepartmentName is NOT NULL
group by DepartmentName
having total > 1;

-- aggregate functions (sum,count,avg, max, min)
select max(salary)  as Maximum_Salary,min(salary) as Minimum_Salary,avg(salary) as Average_Salary,sum(salary) as Total_Salary from employee_salar

select  gender , count(gender)
from employee
group by gender;
```



```
-- Subquery in From

Select a.EID, AllAvgSalary
From
    (Select EID, Salary, AVG(Salary) over () as AllAvgSalary
     From employee_salary) a
Order by a.Salary;

-- to return eid and salary of people born after 1990
Select EID , Salary
From employee_salary
where EID in (
    Select EID
    From employee
    where Year(DOB) > 1990);


 -- retrieve all employees who do not have a salary record in the employee_salary table
SELECT *
FROM employee e
WHERE NOT EXISTS (
    SELECT 1
    FROM employee_salary es
    WHERE e.EID = es.EID
);


-- retrieve all employees who have a salary record in the employee_salary table
```



```
    WHERE e.EID = es.EID
);


-- retrieve all employees who have a salary record in the employee_salary table
SELECT *
FROM employee e
WHERE EXISTS (
    SELECT 1
    FROM employee_salary es
    WHERE e.EID = es.EID
);

-- using case to find out the salary increment according to the department name of the employee
select e.EID , FirstName , Salary , DepartmentName ,
case
    when DepartmentName = "Marketing" then Salary + (Salary*0.10)
    when DepartmentName = "Finance" then Salary + (Salary*0.07)
    when DepartmentName = "Operations" then Salary + (Salary*0.03)
    when DepartmentName = "Human Resources" then Salary + (Salary*0.0001)
    else Salary + (Salary*0.05)

end
as incremented_salary
from employee e
join employee_salary es on e.EID = es.EID
join employee_department ed on e.EID = ed.EID
where DepartmentName is NOT NULL;
```



```sql
INSERT INTO employee (EID, FirstName, LastName, DOB, Gender)
VALUES (125, 'Ruby', 'Doe', '1980-05-10', 'Female');

select * from employee_department
order by EID ;

UPDATE employee_department
SET DepartmentName = "New Department"
WHERE  DepartmentName is null ;

Delete from employee
where EID = 121;

-- to check for duplicates in the tables
SELECT EID, COUNT(*)
FROM employee
GROUP BY EID
HAVING COUNT(*) > 1;

SELECT EID, COUNT(*)
FROM employee_department
GROUP BY EID
HAVING COUNT(*) > 1;
```