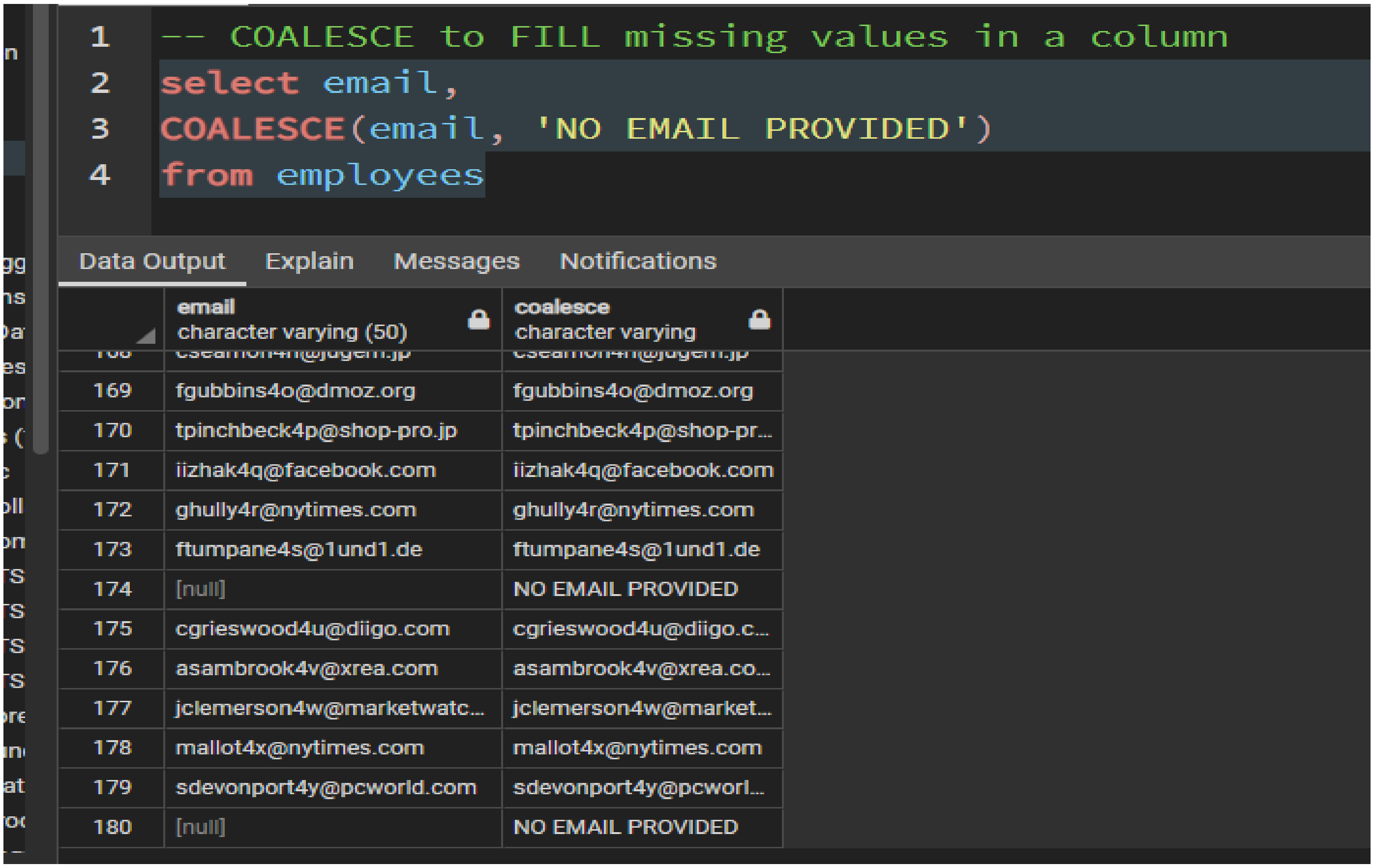
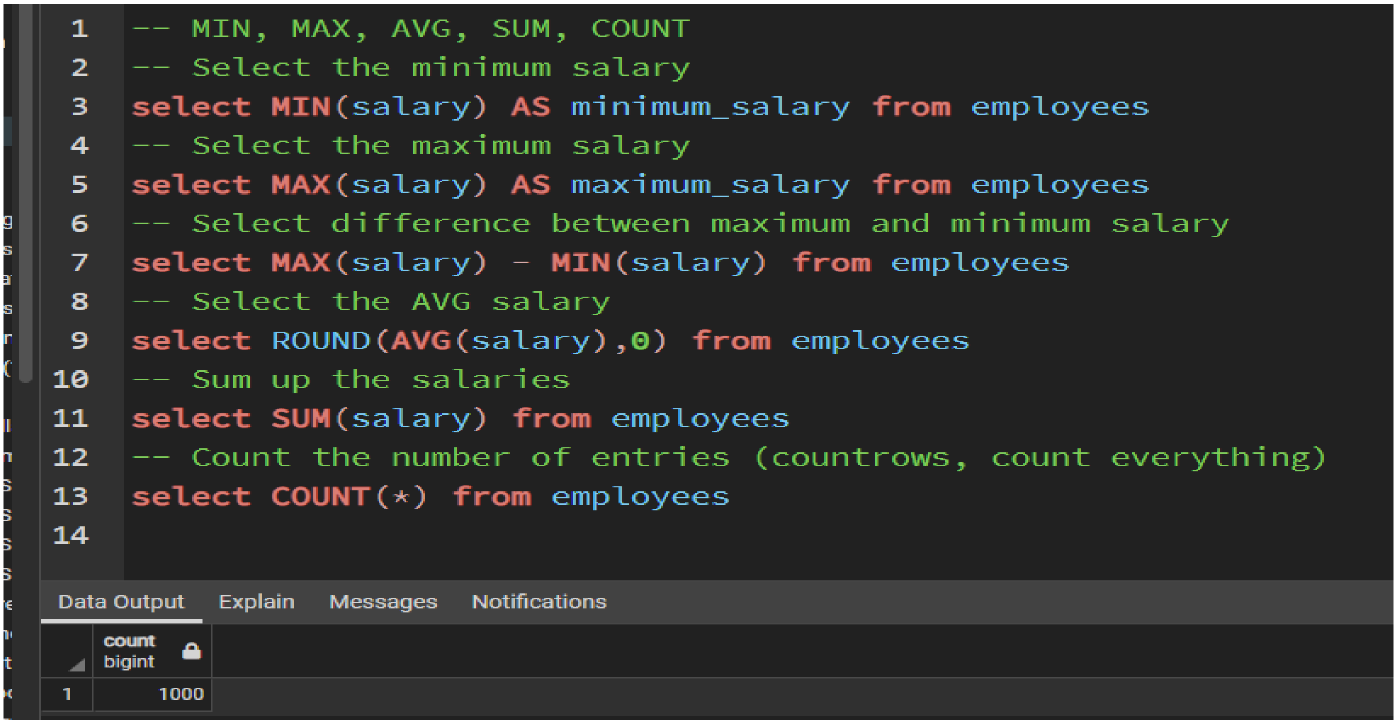


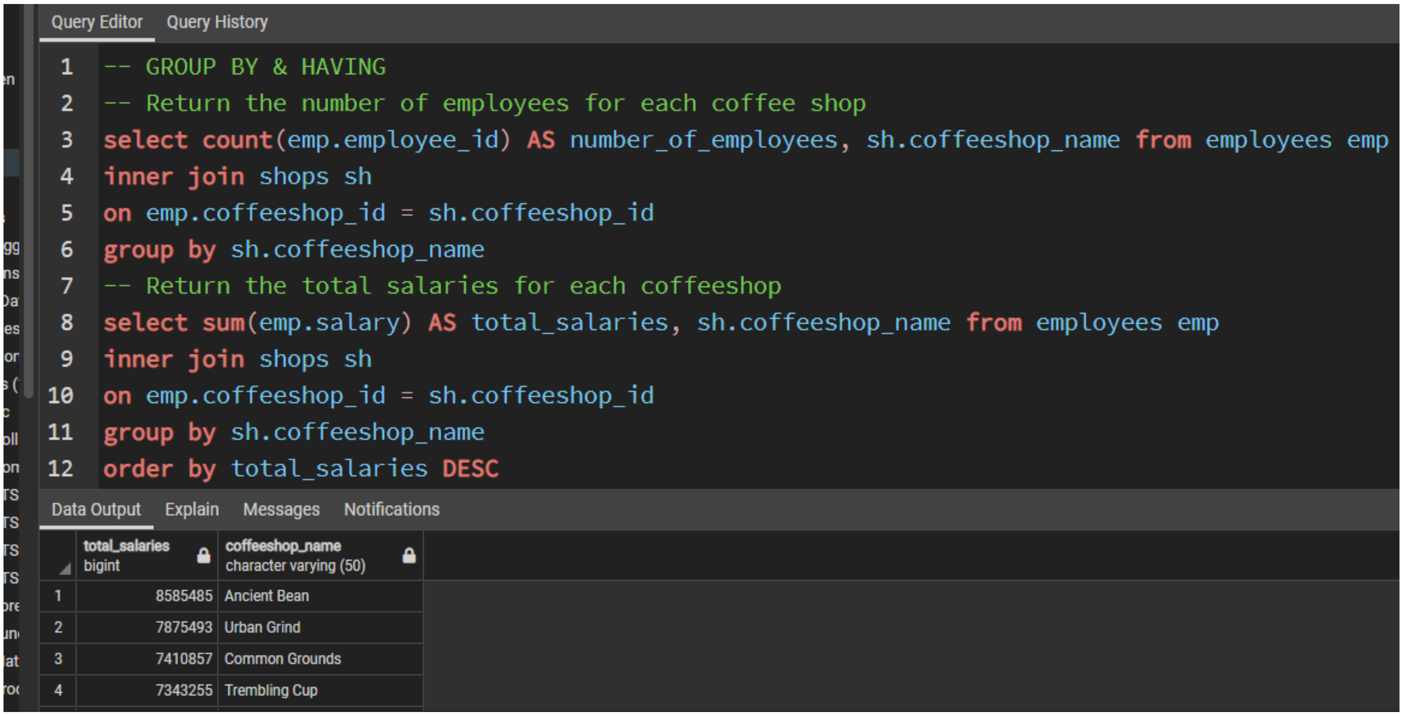
```sql
-- COALESCE to FILL missing values in a column
select email,
COALESCE(email, 'NO EMAIL PROVIDED')
from employees
```

	email character varying (50)	coalesce character varying
168	cseamon4n@jugem.jp	cseamon4n@jugem.jp
169	fgubbins4o@dmoz.org	fgubbins4o@dmoz.org
170	tpinchbeck4p@shop-pro.jp	tpinchbeck4p@shop-pr...
171	iizhak4q@facebook.com	iizhak4q@facebook.com
172	ghully4r@nytimes.com	ghully4r@nytimes.com
173	ftumpane4s@1und1.de	ftumpane4s@1und1.de
174	[null]	NO EMAIL PROVIDED
175	cgrieswood4u@diigo.com	cgrieswood4u@diigo.c...
176	asambrook4v@xrea.com	asambrook4v@xrea.co...
177	jclemerson4w@marketwatc...	jclemerson4w@market...
178	mallot4x@nytimes.com	mallot4x@nytimes.com
179	sdevonport4y@pcworld.com	sdevonport4y@pcworl...
180	[null]	NO EMAIL PROVIDED



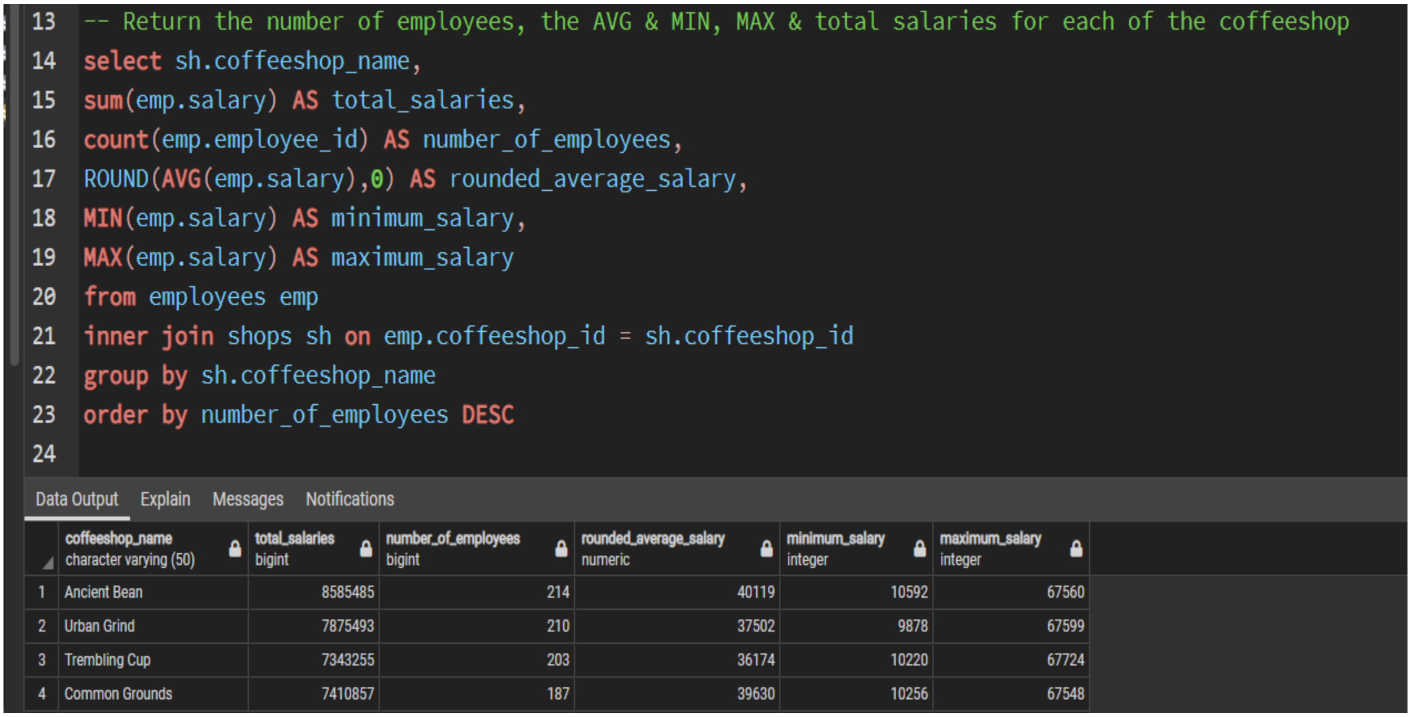
```sql
-- MIN, MAX, AVG, SUM, COUNT
-- Select the minimum salary
select MIN(salary) AS minimum_salary from employees
-- Select the maximum salary
select MAX(salary) AS maximum_salary from employees
-- Select difference between maximum and minimum salary
select MAX(salary) - MIN(salary) from employees
-- Select the AVG salary
select ROUND(AVG(salary),0) from employees
-- Sum up the salaries
select SUM(salary) from employees
-- Count the number of entries (countrows, count everything)
select COUNT(*) from employees

```

	count bigint
1	1000



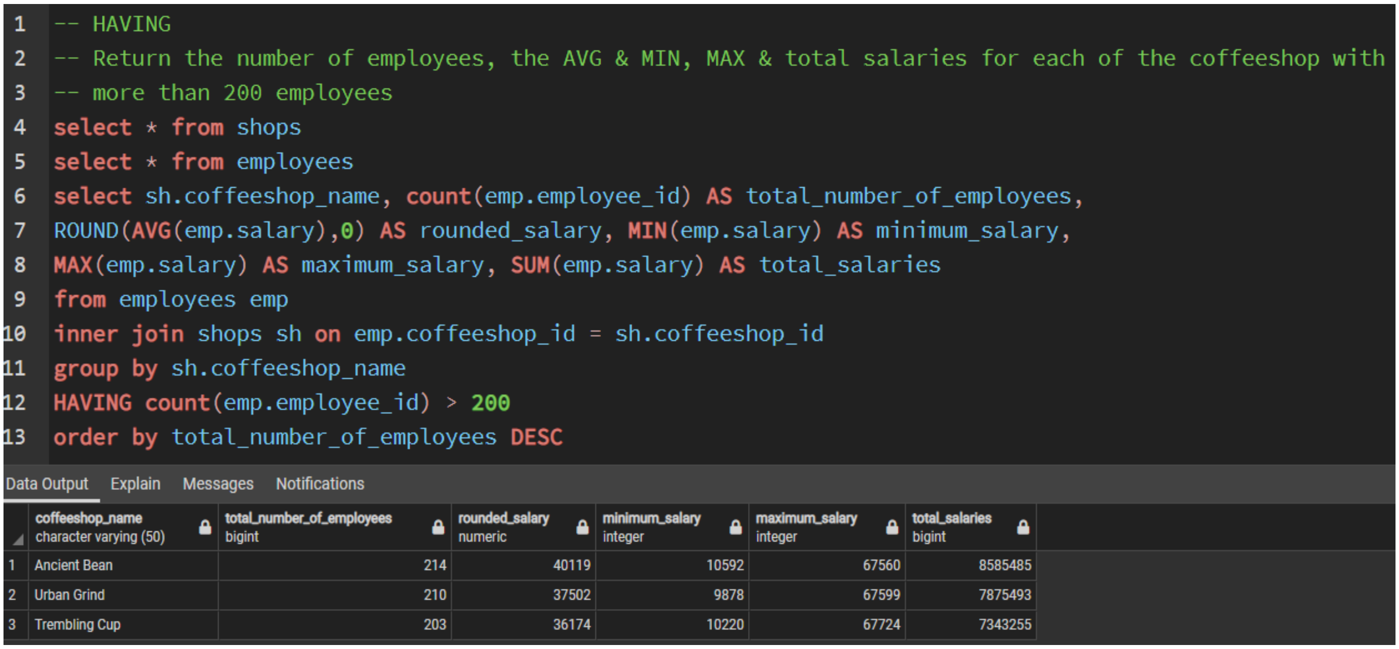
```sql
-- GROUP BY & HAVING
-- Return the number of employees for each coffee shop
select count(emp.employee_id) AS number_of_employees, sh.coffeeshop_name from employees emp
inner join shops sh
on emp.coffeeshop_id = sh.coffeeshop_id
group by sh.coffeeshop_name
-- Return the total salaries for each coffeeshop
select sum(emp.salary) AS total_salaries, sh.coffeeshop_name from employees emp
inner join shops sh
on emp.coffeeshop_id = sh.coffeeshop_id
group by sh.coffeeshop_name
order by total_salaries DESC
```

	total_salaries bigint	coffeeshop_name character varying (50)
1	8585485	Ancient Bean
2	7875493	Urban Grind
3	7410857	Common Grounds
4	7343255	Trembling Cup



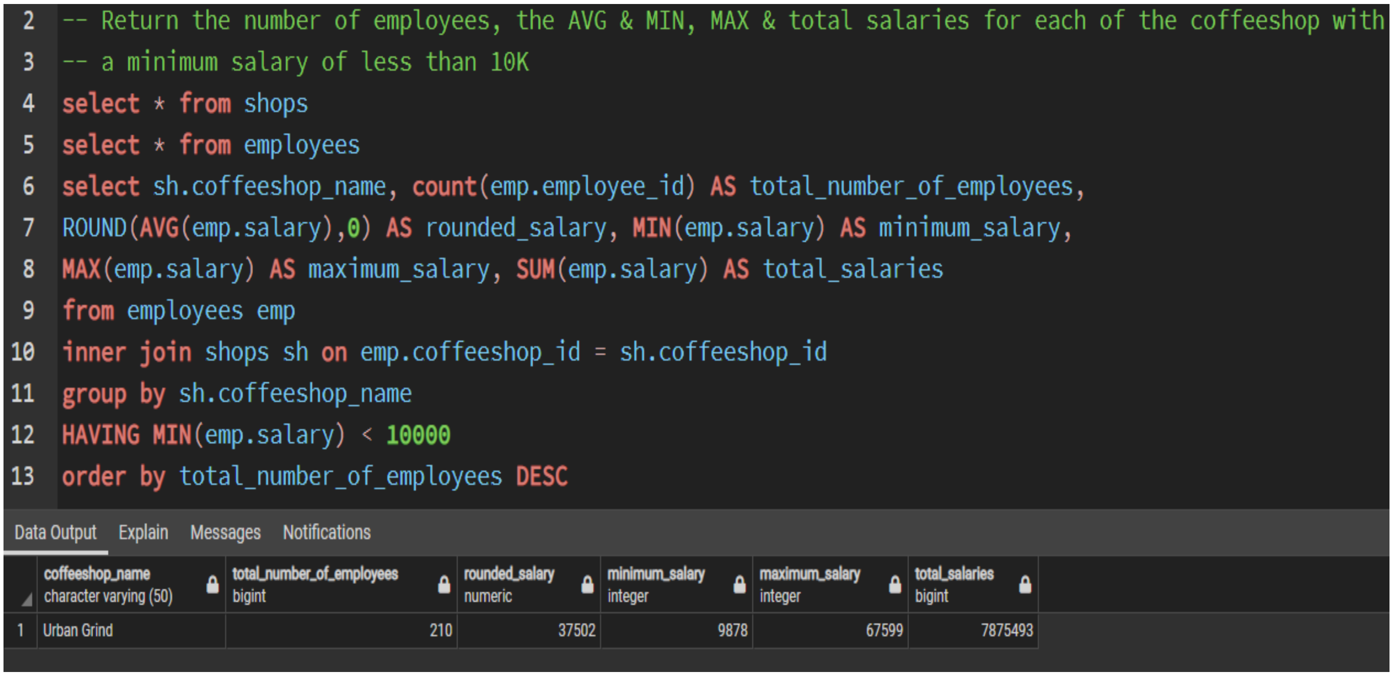
```sql
-- Return the number of employees, the AVG & MIN, MAX & total salaries for each of the coffeeshop
select sh.coffeeshop_name,
sum(emp.salary) AS total_salaries,
count(emp.employee_id) AS number_of_employees,
ROUND(AVG(emp.salary),0) AS rounded_average_salary,
MIN(emp.salary) AS minimum_salary,
MAX(emp.salary) AS maximum_salary
from employees emp
inner join shops sh on emp.coffeeshop_id = sh.coffeeshop_id
group by sh.coffeeshop_name
order by number_of_employees DESC
```

	coffeeshop_name character varying (50)	total_salaries bigint	number_of_employees bigint	rounded_average_salary numeric	minimum_salary integer	maximum_salary integer
1	Ancient Bean	8585485	214	40119	10592	67560
2	Urban Grind	7875493	210	37502	9878	67599
3	Trembling Cup	7343255	203	36174	10220	67724
4	Common Grounds	7410857	187	39630	10256	67548



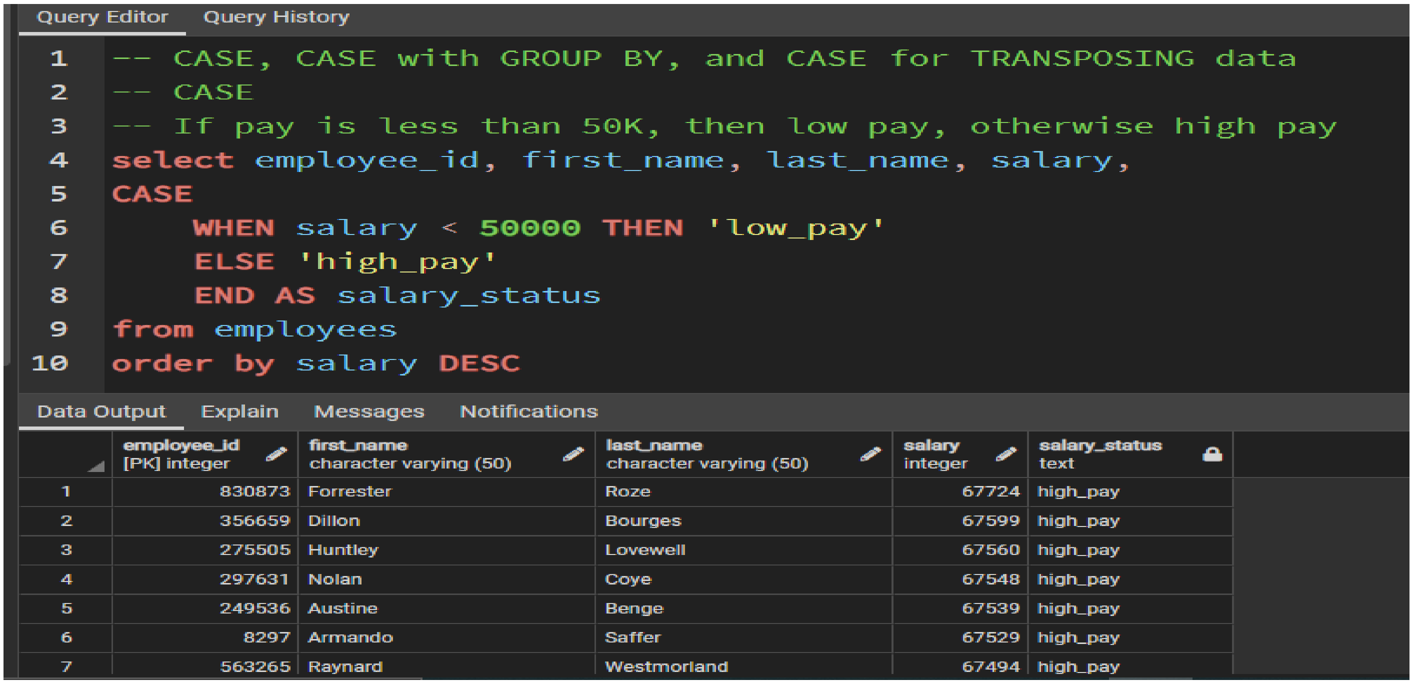
```sql
-- HAVING
-- Return the number of employees, the AVG & MIN, MAX & total salaries for each of the coffeeshop with
-- more than 200 employees
select * from shops
select * from employees
select sh.coffeeshop_name, count(emp.employee_id) AS total_number_of_employees,
ROUND(AVG(emp.salary),0) AS rounded_salary, MIN(emp.salary) AS minimum_salary,
MAX(emp.salary) AS maximum_salary, SUM(emp.salary) AS total_salaries
from employees emp
inner join shops sh on emp.coffeeshop_id = sh.coffeeshop_id
group by sh.coffeeshop_name
HAVING count(emp.employee_id) > 200
order by total_number_of_employees DESC
```

	coffeeshop_name character varying (50)	total_number_of_employees bigint	rounded_salary numeric	minimum_salary integer	maximum_salary integer	total_salaries bigint
1	Ancient Bean	214	40119	10592	67560	8585485
2	Urban Grind	210	37502	9878	67599	7875493
3	Trembling Cup	203	36174	10220	67724	7343255



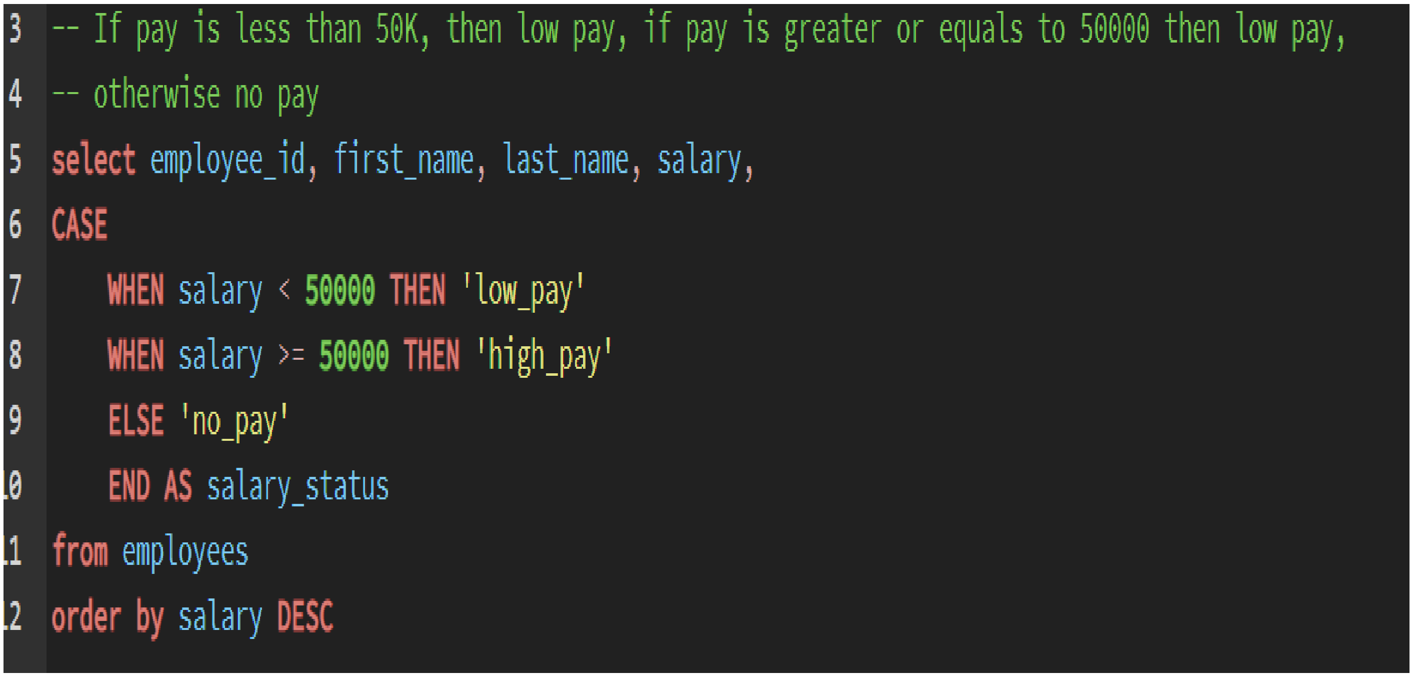
```sql
-- Return the number of employees, the AVG & MIN, MAX & total salaries for each of the coffeeshop with
-- a minimum salary of less than 10K
select * from shops
select * from employees
select sh.coffeeshop_name, count(emp.employee_id) AS total_number_of_employees,
ROUND(AVG(emp.salary),0) AS rounded_salary, MIN(emp.salary) AS minimum_salary,
MAX(emp.salary) AS maximum_salary, SUM(emp.salary) AS total_salaries
from employees emp
inner join shops sh on emp.coffeeshop_id = sh.coffeeshop_id
group by sh.coffeeshop_name
HAVING MIN(emp.salary) < 10000
order by total_number_of_employees DESC
```

	coffeeshop_name character varying (50)	total_number_of_employees bigint	rounded_salary numeric	minimum_salary integer	maximum_salary integer	total_salaries bigint
1	Urban Grind	210	37502	9878	67599	7875493

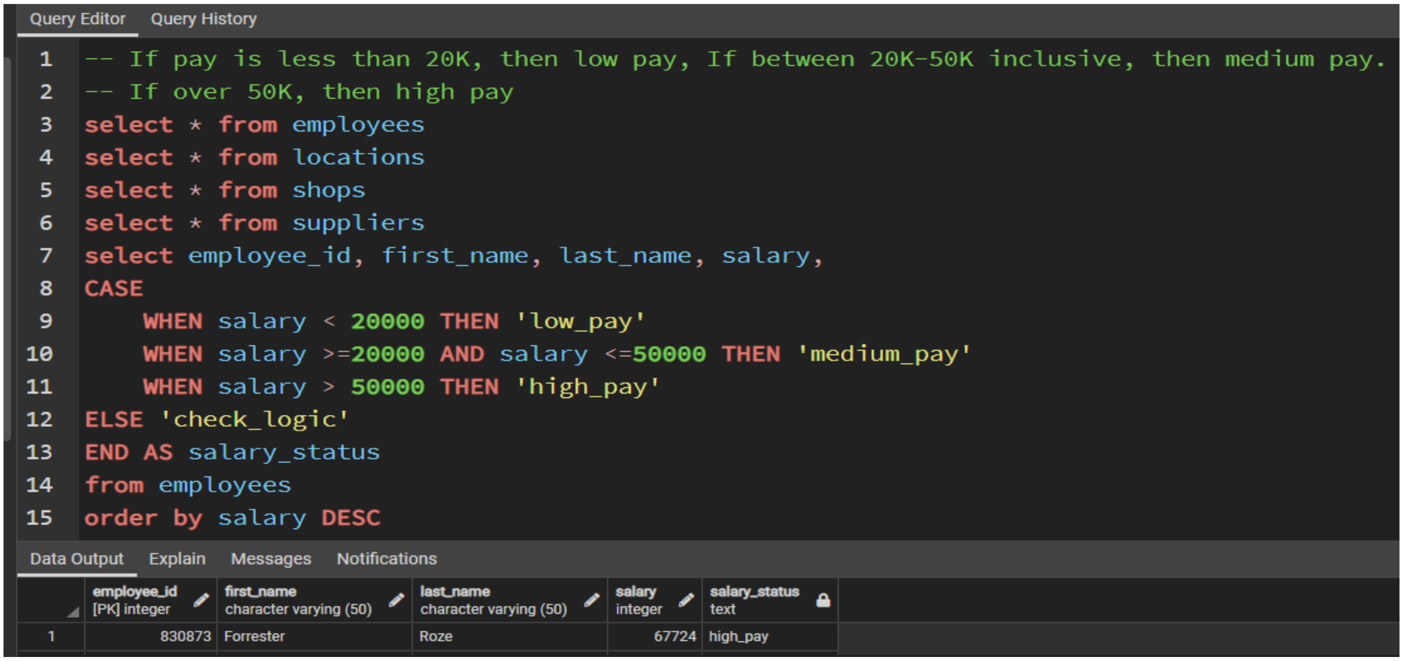


```sql
-- CASE, CASE with GROUP BY, and CASE for TRANSPOSING data
-- CASE
-- If pay is less than 50K, then low pay, otherwise high pay
select employee_id, first_name, last_name, salary,
CASE
     WHEN salary < 50000 THEN 'low_pay'
     ELSE 'high_pay'
     END AS salary_status
from employees
order by salary DESC
```

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	salary_status text
1	830873	Forrester	Roze	67724	high_pay
2	356659	Dillon	Bourges	67599	high_pay
3	275505	Huntley	Lovewell	67560	high_pay
4	297631	Nolan	Coye	67548	high_pay
5	249536	Austine	Benge	67539	high_pay
6	8297	Armando	Saffer	67529	high_pay
7	563265	Raynard	Westmorland	67494	high_pay



```sql
-- If pay is less than 50K, then low pay, if pay is greater or equals to 50000 then low pay,
-- otherwise no pay
select employee_id, first_name, last_name, salary,
CASE
    WHEN salary < 50000 THEN 'low_pay'
    WHEN salary >= 50000 THEN 'high_pay'
    ELSE 'no_pay'
    END AS salary_status
from employees
order by salary DESC
```



```sql
-- If pay is less than 20K, then low pay, If between 20K-50K inclusive, then medium pay.
-- If over 50K, then high pay
select * from employees
select * from locations
select * from shops
select * from suppliers
select employee_id, first_name, last_name, salary,
CASE
    WHEN salary < 20000 THEN 'low_pay'
    WHEN salary >=20000 AND salary <=50000 THEN 'medium_pay'
    WHEN salary > 50000 THEN 'high_pay'
ELSE 'check_logic'
END AS salary_status
from employees
order by salary DESC
```

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	salary_status text
1	830873	Forrester	Roze	67724	high_pay