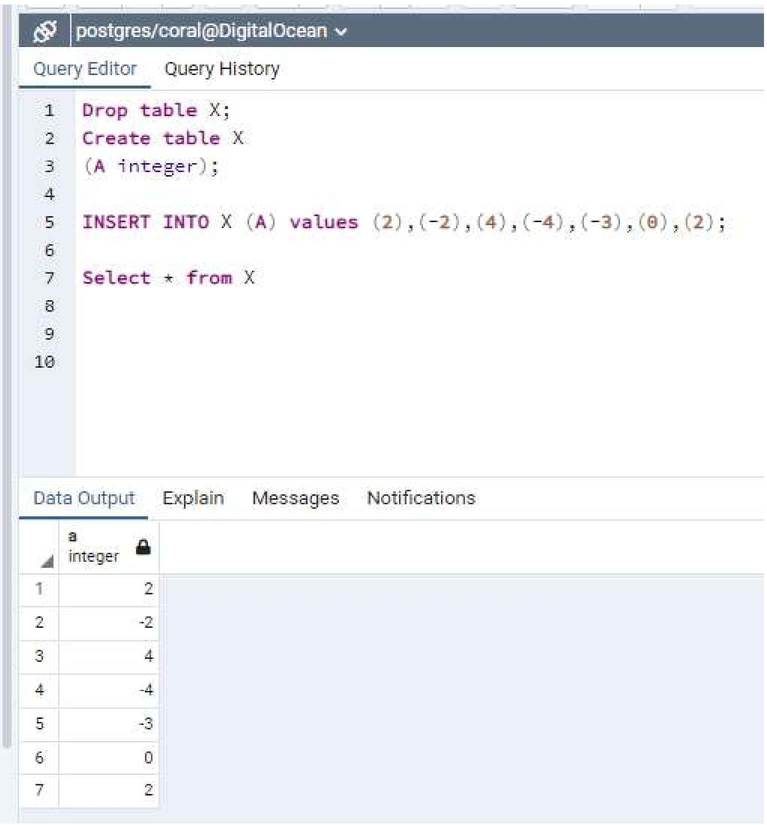
SQL - Results

Given the following table (X):

```
A
------
2
-2
4
-4
-3
0
2
```

1) **Write a single query to calculate the sum of all positive values and save it as sum_positive of and the sum of all negative values of and save it as sum_negative.**

Create a table



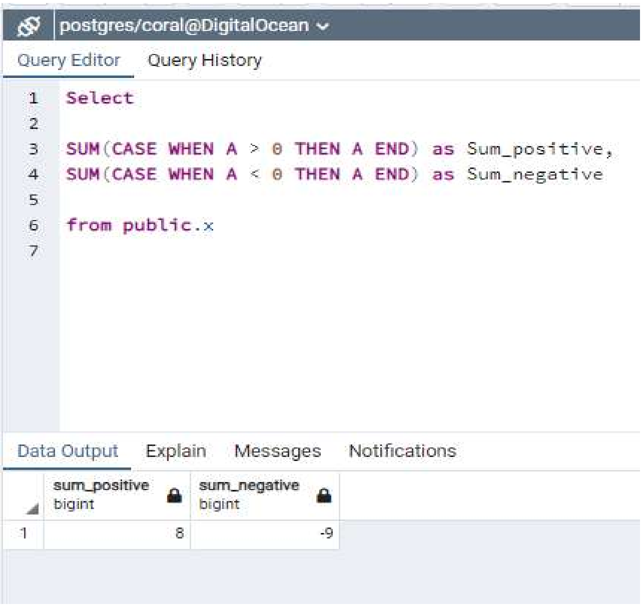
```sql
Drop table X;
Create table X
(A integer);

INSERT INTO X (A) values (2),(-2),(4),(-4),(-3),(0),(2);

Select * from X
```

	a integer
1	2
2	-2
3	4
4	-4
5	-3
6	0
7	2

Single Query



```sql
Select

SUM(CASE WHEN A > 0 THEN A END) as Sum_positive,
SUM(CASE WHEN A < 0 THEN A END) as Sum_negative

from public.x
```

	sum_positive bigint	sum_negative bigint
1	8	-9

Given the table (X):

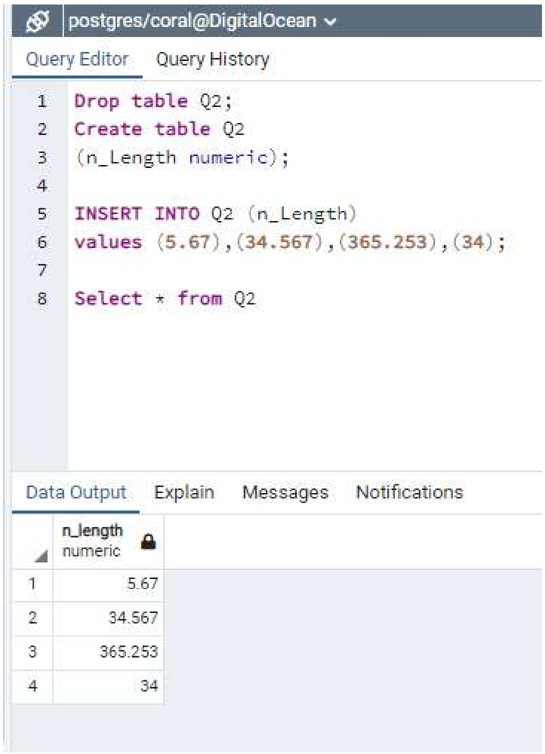
Length
5.67
34.567
365.253
34

2) **Write a query that produces the output:**

Length	m	cm
5.67	5	67
34.567	34	567
365.253	365	253
34	34	0

Create a table

Query - floor - case when

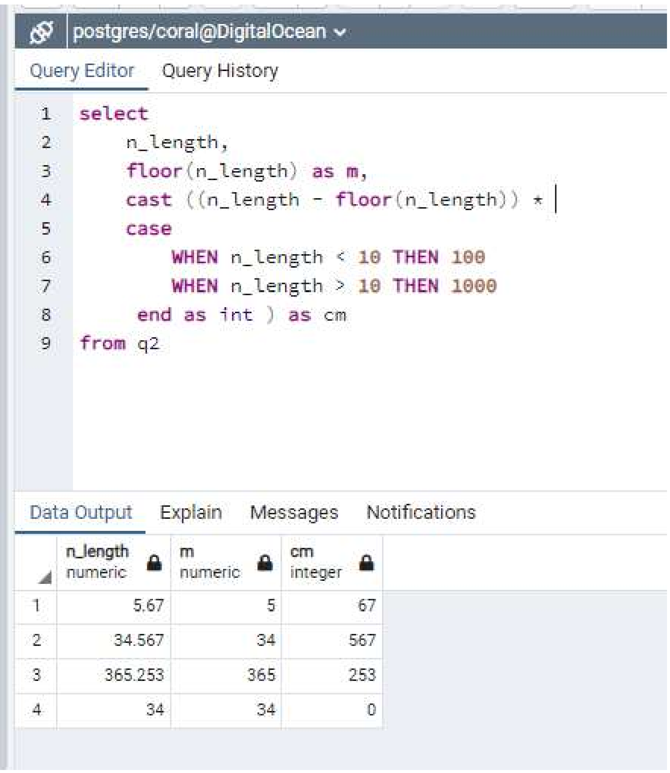


```sql
Drop table Q2;
Create table Q2
(n_Length numeric);

INSERT INTO Q2 (n_Length)
values (5.67),(34.567),(365.253),(34);

Select * from Q2
```

	n_length numeric
1	5.67
2	34.567
3	365.253
4	34



```sql
select
    n_length,
    floor(n_length) as m,
    cast ((n_length - floor(n_length)) *
    case
        WHEN n_length < 10 THEN 100
        WHEN n_length > 10 THEN 1000
     end as int ) as cm
from q2
```

	n_length numeric	m numeric	cm integer
1	5.67	5	67
2	34.567	34	567
3	365.253	365	253
4	34	34	0

Given the two tables (Employee, Reward):

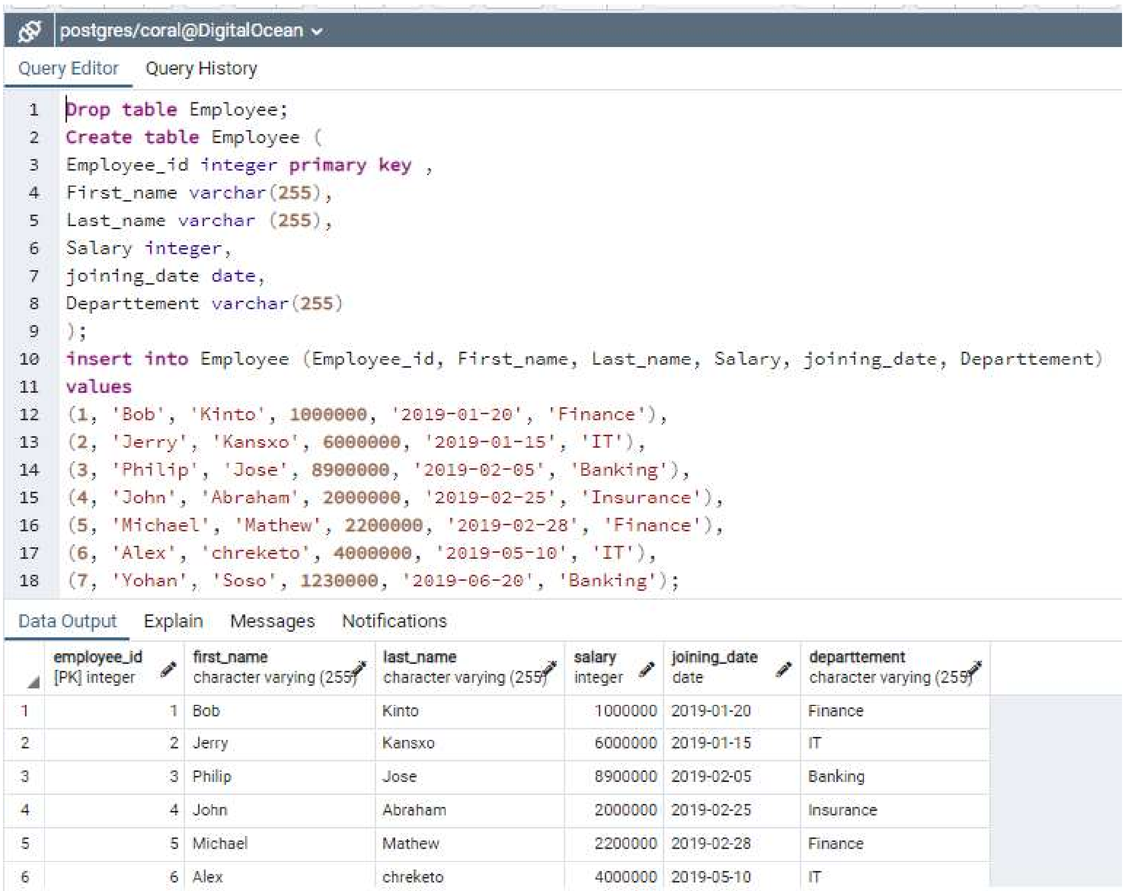
Employee:

Employee_id	First_name	Last_name	Salary	Joining_date	Departement
1	Bob	Kinto	1000000	2019-01-20	Finance
2	Jerry	Kansxo	6000000	2019-01-15	IT
3	Philip	Jose	8900000	2019-02-05	Banking
4	John	Abraham	2000000	2019-02-25	Insurance
5	Michael	Mathew	2200000	2019-02-28	Finance
6	Alex	chreketo	4000000	2019-05-10	IT
7	Yohan	Soso	1230000	2019-06-20	Banking

Reward:

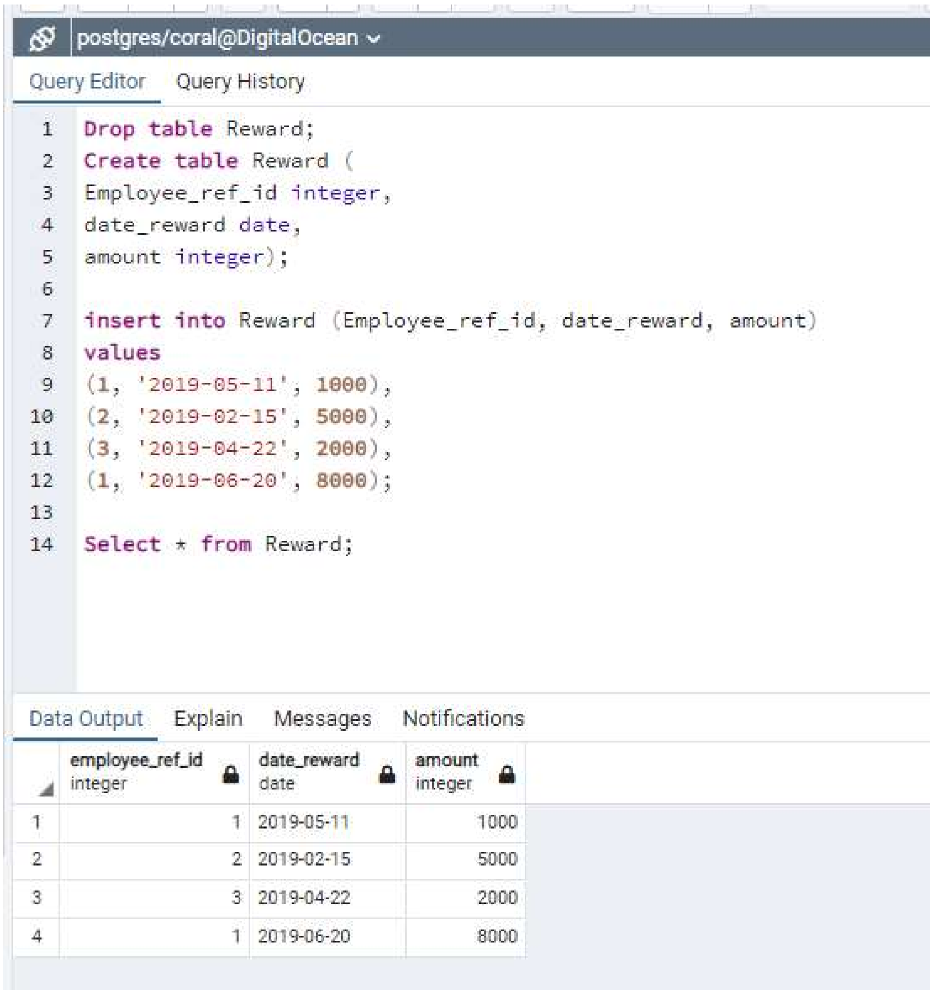
Employee_ref_id	date_reward	amount
1	2019-05-11	1000
2	2019-02-15	5000
3	2019-04-22	2000
1	2019-06-20	8000

Create two tables



```sql
Drop table Employee;
Create table Employee (
Employee_id integer primary key ,
First_name varchar(255),
Last_name varchar (255),
Salary integer,
joining_date date,
Departtement varchar(255)
);
insert into Employee (Employee_id, First_name, Last_name, Salary, joining_date, Departtement)
values
(1, 'Bob', 'Kinto', 1000000, '2019-01-20', 'Finance'),
(2, 'Jerry', 'Kansxo', 6000000, '2019-01-15', 'IT'),
(3, 'Philip', 'Jose', 8900000, '2019-02-05', 'Banking'),
(4, 'John', 'Abraham', 2000000, '2019-02-25', 'Insurance'),
(5, 'Michael', 'Mathew', 2200000, '2019-02-28', 'Finance'),
(6, 'Alex', 'chreketo', 4000000, '2019-05-10', 'IT'),
(7, 'Yohan', 'Soso', 1230000, '2019-06-20', 'Banking');
```

	employee_id [PK] integer	first_name character varying (255)	last_name character varying (255)	salary integer	joining_date date	departtement character varying (255)
1	1	Bob	Kinto	1000000	2019-01-20	Finance
2	2	Jerry	Kansxo	6000000	2019-01-15	IT
3	3	Philip	Jose	8900000	2019-02-05	Banking
4	4	John	Abraham	2000000	2019-02-25	Insurance
5	5	Michael	Mathew	2200000	2019-02-28	Finance
6	6	Alex	chreketo	4000000	2019-05-10	IT



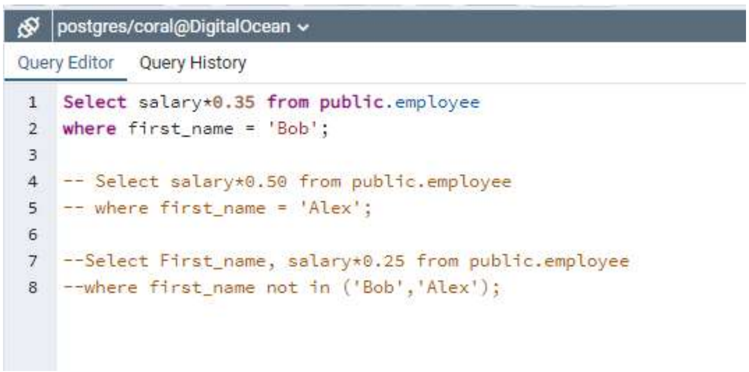
```
Drop table Reward;
Create table Reward (
Employee_ref_id integer,
date_reward date,
amount integer);

insert into Reward (Employee_ref_id, date_reward, amount)
values
(1, '2019-05-11', 1000),
(2, '2019-02-15', 5000),
(3, '2019-04-22', 2000),
(1, '2019-06-20', 8000);

Select * from Reward;
```

	employee_ref_id integer	date_reward date	amount integer
1	1	2019-05-11	1000
2	2	2019-02-15	5000
3	3	2019-04-22	2000
4	1	2019-06-20	8000

3) Get 35% of Bob's salary, 50% of Alex's salary, and 25% of other employees' salaries.

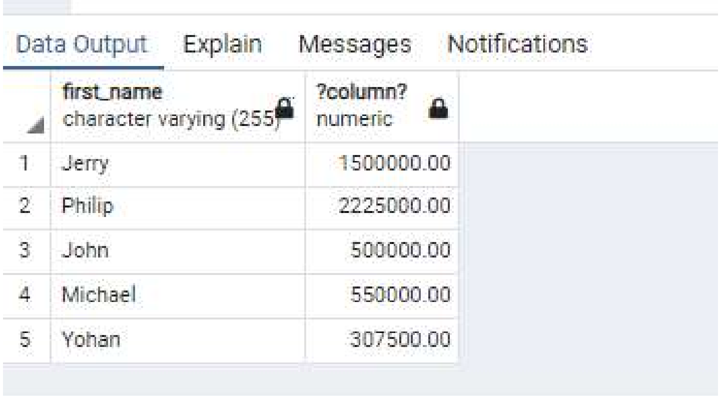


```
Select salary*0.35 from public.employee
where first_name = 'Bob';

-- Select salary*0.50 from public.employee
-- where first_name = 'Alex';

--Select First_name, salary*0.25 from public.employee
--where first_name not in ('Bob','Alex');
```

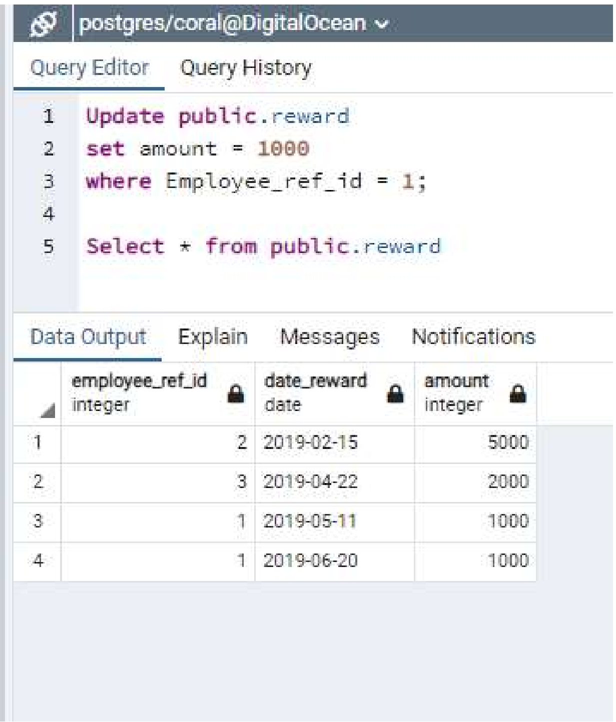
Bob	350000
Alex	2000000



	first_name character varying (255)	?column? numeric
1	Jerry	1500000.00
2	Philip	2225000.00
3	John	500000.00
4	Michael	550000.00
5	Yohan	307500.00

4) Update the reward of “Bob” to 1000.

Query - update - set



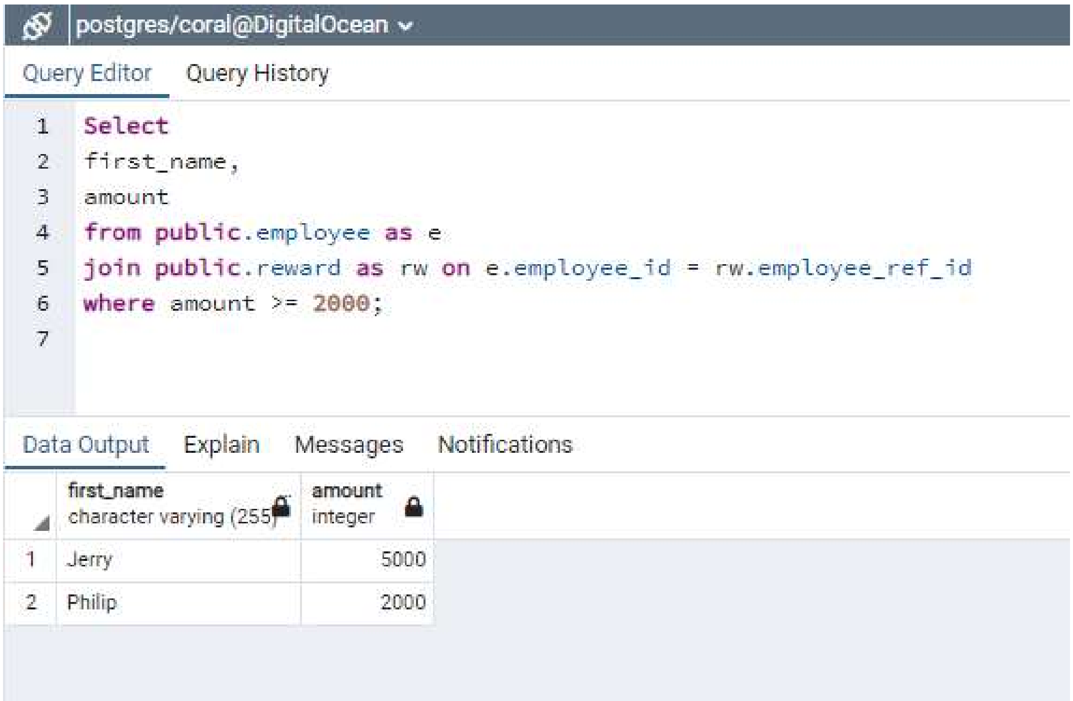
```sql
Update public.reward
set amount = 1000
where Employee_ref_id = 1;

Select * from public.reward
```

	employee_ref_id integer	date_reward date	amount integer
1	2	2019-02-15	5000
2	3	2019-04-22	2000
3	1	2019-05-11	1000
4	1	2019-06-20	1000

5) Get the first name, the reward amount for employees who have rewards with an amount greater than, equal to 2000.

Query - join - alias



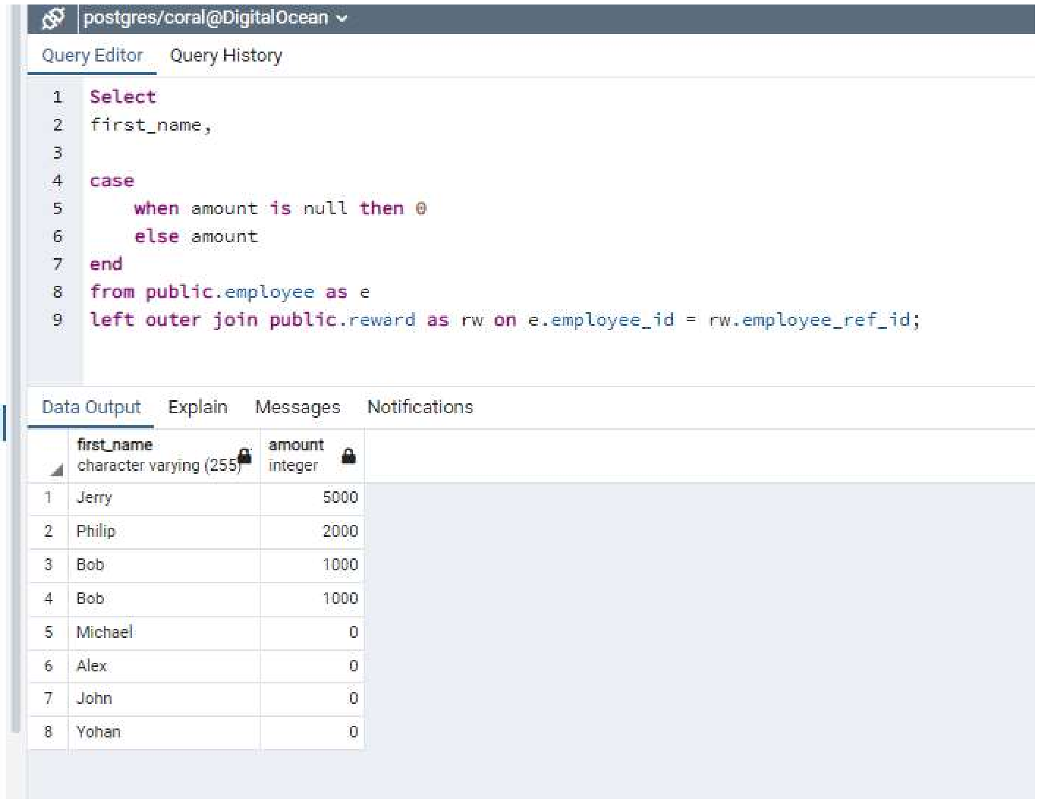
```sql
Select
first_name,
amount
from public.employee as e
join public.reward as rw on e.employee_id = rw.employee_ref_id
where amount >= 2000;

```

	first_name character varying (255)	amount integer
1	Jerry	5000
2	Philip	2000

6) Get the first name, the reward amount for employees even if they did not receive any rewards, and set a reward amount equal to 0 for the employees who did not receive rewards.

Query - case when - join



```sql
Select
first_name,

case
    when amount is null then 0
    else amount
end
from public.employee as e
left outer join public.reward as rw on e.employee_id = rw.employee_ref_id;
```

	first_name character varying (255)	amount integer
1	Jerry	5000
2	Philip	2000
3	Bob	1000
4	Bob	1000
5	Michael	0
6	Alex	0
7	John	0
8	Yohan	0