

# SQL Practice – Employee Insights

Real-world business questions answered with PostgreSQL

Lilla Katona  
2025.05.09.

## Project Summary

This mini project demonstrates practical SQL skills through a fictional employee dataset. It includes two tables: *employees* and *performance\_reviews*. The queries aim to extract and clean HR-related data such as full employee names, salary formatting, department counts, and performance trends.

## Demonstrated Skills:

- CREATE TABLE, INSERT INTO
- CONCAT
- CAST
- COALESCE
- LPAD
- DISTINCT

## Dataset Creation:

### Query:

```
CREATE TABLE employees (  
  id SERIAL PRIMARY KEY,  
  first_name TEXT,  
  last_name TEXT,  
  department TEXT,  
  hire_date DATE,  
  termination_date DATE,  
  salary NUMERIC  
);
```

```
CREATE TABLE performance_reviews (  
  id SERIAL PRIMARY KEY,  
  employee_id INTEGER REFERENCES employees(id),  
  review_date DATE,  
  score INTEGER  
);
```

-- Employees table

```
INSERT INTO employees (first_name, last_name, department, hire_date,  
  termination_date, salary) VALUES  
( 'Alice', 'Smith', 'Marketing', '2020-02-15', NULL, 55000),  
( 'Bob', 'Johnson', 'Sales', '2019-06-10', '2023-01-01', 62000),  
( 'Charlie', 'Lee', 'IT', '2021-03-20', NULL, 70000),  
( 'Diana', 'Wang', 'HR', '2018-11-01', NULL, 58000),
```

```
('Ethan', 'Brown', 'Sales', '2022-05-05', NULL, NULL);
```

-- Performance Reviews table

```
INSERT INTO performance_reviews (employee_id, review_date, score) VALUES
(1, '2022-01-10', 80),
(1, '2023-01-10', 85),
(2, '2022-01-10', 70),
(3, '2023-03-15', 90),
(4, '2023-04-01', 88);
```

### Result:

1	Select	5 rows returned											
2	a.*,												
3	b.*												
4	FROM employees a	id	first_name	last_name	department	hire_date	termination_date	salary	id	employee_id	review_date	score	
5	LEFT JOIN performance_reviews b	integer	text	text	text	date	date	numeric	integer	integer	date	integer	
6	on a.id = b.id	1	1	Alice	Smith	Marketing	2020-02-15	null	55000	1	1	2022-01-10	80
7		2	2	Bob	Johnson	Sales	2019-06-10	2023-01-01	62000	2	1	2023-01-10	85
		3	3	Charlie	Lee	IT	2021-03-20	null	70000	3	2	2022-01-10	70
		4	4	Diana	Wang	HR	2018-11-01	null	58000	4	3	2023-03-15	90
		5	5	Ethan	Brown	Sales	2022-05-05	null	null	5	4	2023-04-01	88

## Queries:

### 1. CONCAT: Full name with department

```
SELECT
  CONCAT(first_name, ' ', last_name) AS full_name,
  department
FROM employees;
```

### Result:

5 rows returned		
	full_name	department
	text	text
1	Alice Smith	Marketing
2	Bob Johnson	Sales
3	Charlie Lee	IT
4	Diana Wang	HR
5	Ethan Brown	Sales

## 2. **CAST**: Format salary with currency

```
SELECT
  first_name,
  last_name,
  CAST(salary AS TEXT) || ' USD' AS formatted_salary
FROM employees;
```

### Result:

5 rows returned

	first_name text	last_name text	formatted_salary text
1	Alice	Smith	55000 USD
2	Bob	Johnson	62000 USD
3	Charlie	Lee	70000 USD
4	Diana	Wang	58000 USD
5	Ethan	Brown	<i>null</i>

### Alternative solution:

```
SELECT
  first_name,
  last_name,
  CONCAT(COALESCE(CAST(salary AS TEXT), 'N/A'), ' USD') AS formatted_salary
FROM employees;
```

### Result:

5 rows returned

	first_name text	last_name text	formatted_salary text
1	Alice	Smith	55000 USD
2	Bob	Johnson	62000 USD
3	Charlie	Lee	70000 USD
4	Diana	Wang	58000 USD
5	Ethan	Brown	N/A USD

### 3. COALESCE: Handle missing salaries

```
SELECT
  first_name,
  last_name,
  salary
  COALESCE(salary, 0) AS effective_salary
FROM employees;
```

#### Result:

5 rows returned

	first_name text	last_name text	salary numeric	effective_salary numeric
1	Alice	Smith	55000	55000
2	Bob	Johnson	62000	62000
3	Charlie	Lee	70000	70000
4	Diana	Wang	58000	58000
5	Ethan	Brown	null	0

### 4. LPAD: Show employee IDs as 5-digit numbers

```
SELECT
  LPAD(id::TEXT, 5, '0') AS padded_id,
  first_name,
  last_name
FROM employees;
```

#### Result:

5 rows returned

	padded_id text	first_name text	last_name text
1	00001	Alice	Smith
2	00002	Bob	Johnson
3	00003	Charlie	Lee
4	00004	Diana	Wang
5	00005	Ethan	Brown

## 5. **DISTINCT**: Count and list unique departments

```
SELECT  
    COUNT(DISTINCT department) AS unique_departments  
FROM employees;
```

**Result:**

1 row returned	
	unique_departments bigint
1	4

```
SELECT DISTINCT  
    department  
FROM employees;
```

**Result:**

4 rows returned	
	department text
1	Marketing
2	Sales
3	IT
4	HR