Join with the USING Clause

Natural Join

Join with the USING Clause

Inner Join

LEFT JOIN (or LEFT OUTER JOIN)

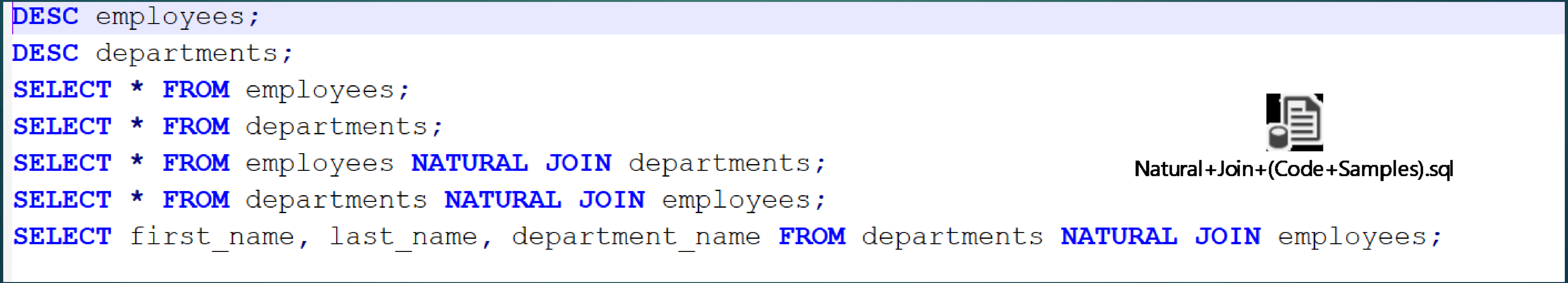
RIGHT JOIN (or RIGHT OUTER JOIN)

FULL JOIN (or FULL OUTER JOIN)

By Eslam Khder

Natural Join

A NATURAL JOIN in Oracle Database is a type of join that automatically joins tables based on columns with the same name and compatible data types in .both tables

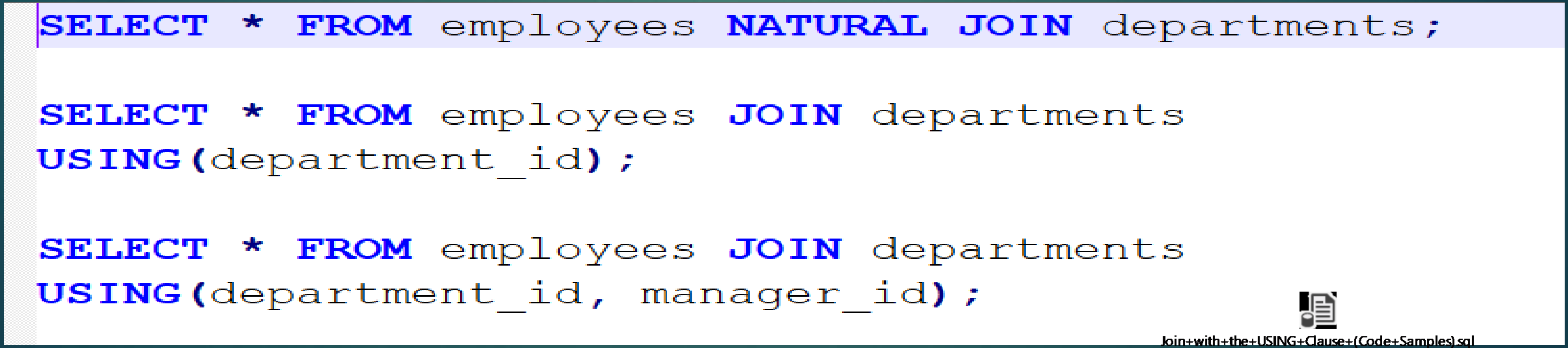


```sql
DESC employees;
DESC departments;
SELECT * FROM employees;
SELECT * FROM departments;
SELECT * FROM employees NATURAL JOIN departments;
SELECT * FROM departments NATURAL JOIN employees;
SELECT first_name, last_name, department_name FROM departments NATURAL JOIN employees;
```

Natural+Join+(Code+Samples).sql

By Eslam Khder

Join with the USING Clause



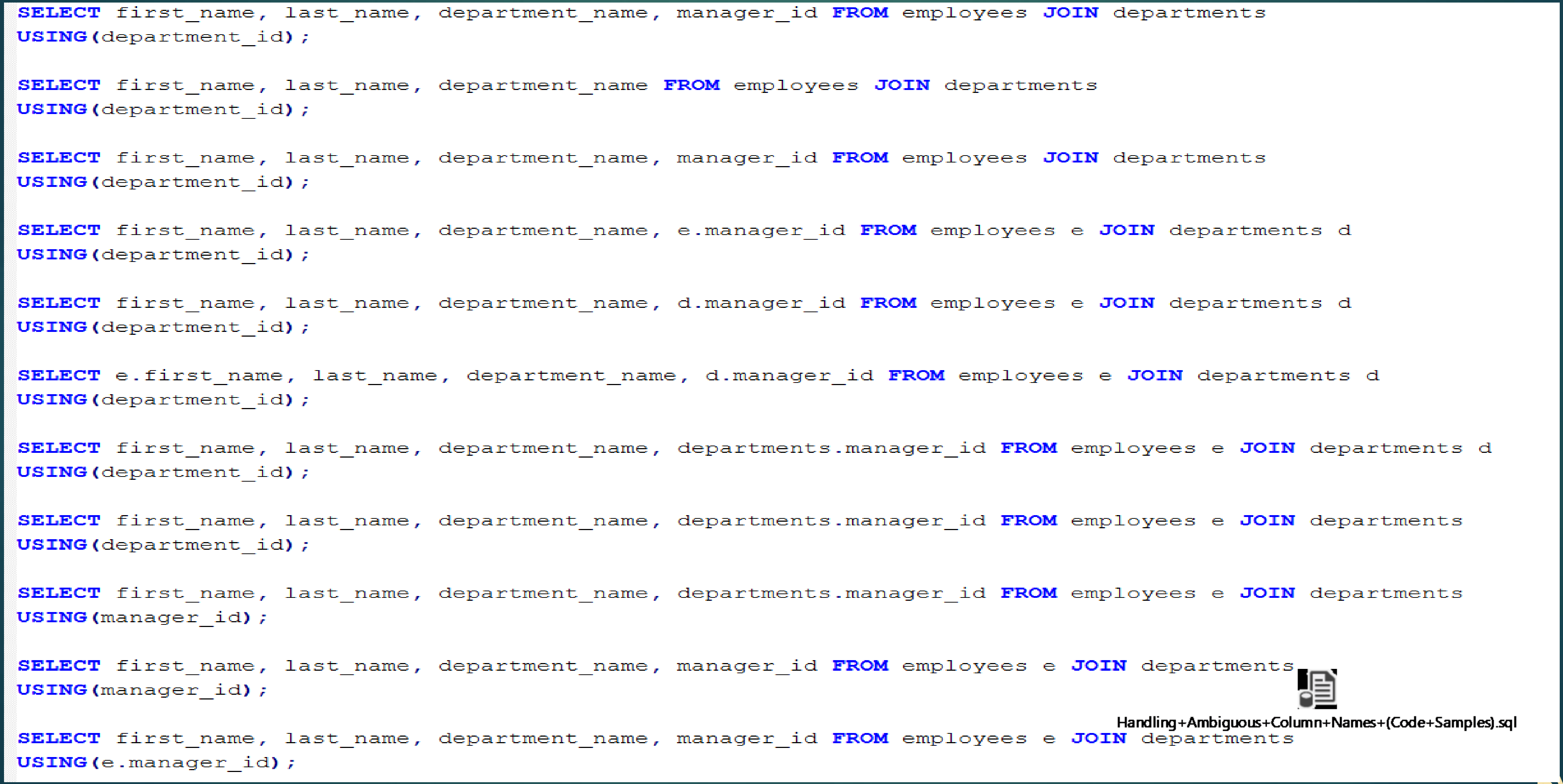
```sql
SELECT * FROM employees NATURAL JOIN departments;

SELECT * FROM employees JOIN departments
USING(department_id);

SELECT * FROM employees JOIN departments
USING(department_id, manager_id);
```

Join+with+the+USING+Clause+(Code+Samples).sql

By Eslam Khder

Handling Ambiguous Column Names

EraaSoft



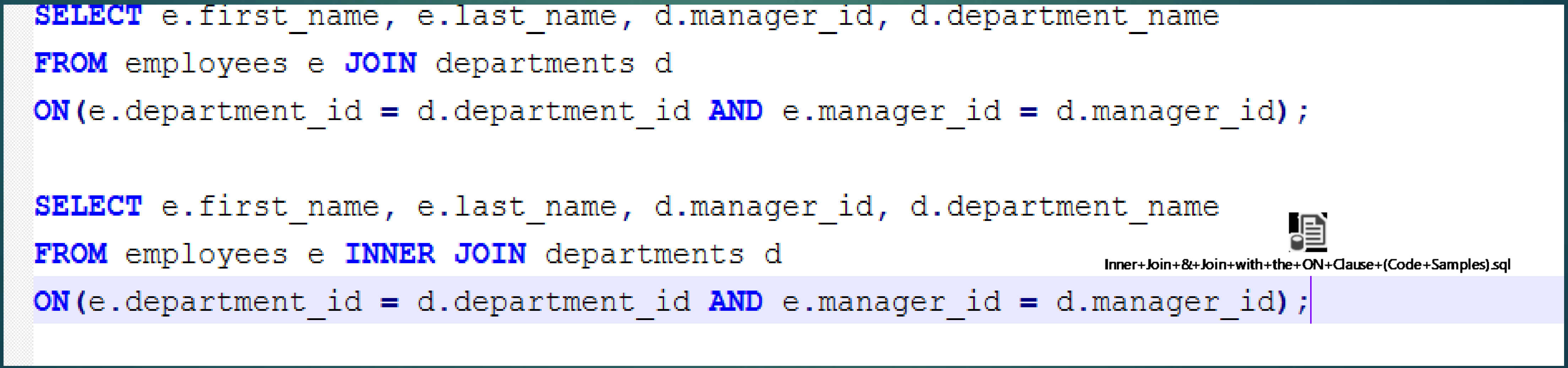
```sql
SELECT first_name, last_name, department_name, manager_id FROM employees JOIN departments
USING(department_id);

SELECT first_name, last_name, department_name FROM employees JOIN departments
USING(department_id);

SELECT first_name, last_name, department_name, manager_id FROM employees JOIN departments
USING(department_id);

SELECT first_name, last_name, department_name, e.manager_id FROM employees e JOIN departments d
USING(department_id);

SELECT first_name, last_name, department_name, d.manager_id FROM employees e JOIN departments d
USING(department_id);

SELECT e.first_name, last_name, department_name, d.manager_id FROM employees e JOIN departments d
USING(department_id);

SELECT first_name, last_name, department_name, departments.manager_id FROM employees e JOIN departments d
USING(department_id);

SELECT first_name, last_name, department_name, departments.manager_id FROM employees e JOIN departments
USING(department_id);

SELECT first_name, last_name, department_name, departments.manager_id FROM employees e JOIN departments
USING(manager_id);

SELECT first_name, last_name, department_name, manager_id FROM employees e JOIN departments
USING(manager_id);

SELECT first_name, last_name, department_name, manager_id FROM employees e JOIN departments
USING(e.manager_id);
```

Handling+Ambiguous+Column+Names+(Code+Samples).sql

By Eslam Khder

Inner Join Join with the ON Clause

An INNER JOIN returns rows when there is a match in both .tables. It is the most common type of join

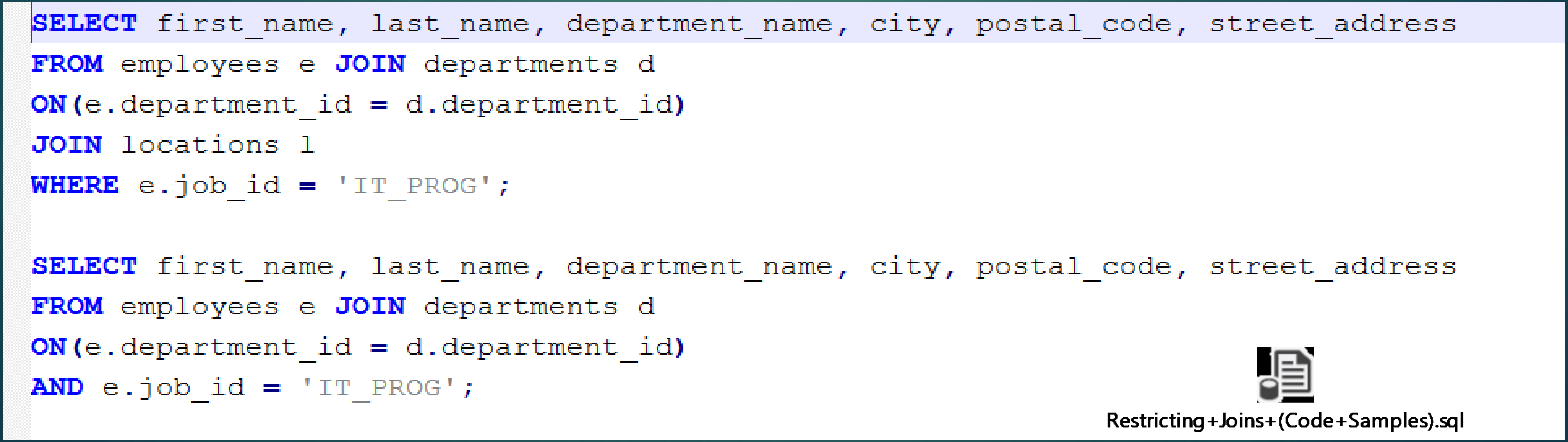


```
SELECT e.first_name, e.last_name, d.manager_id, d.department_name
FROM employees e JOIN departments d
ON(e.department_id = d.department_id AND e.manager_id = d.manager_id);

SELECT e.first_name, e.last_name, d.manager_id, d.department_name
FROM employees e INNER JOIN departments d
ON(e.department_id = d.department_id AND e.manager_id = d.manager_id);
```

Inner+Join+&+Join+with+the+ON+Clause+(Code+Samples).sql

By Eslam Khder

Restricting Joins



```sql
SELECT first_name, last_name, department_name, city, postal_code, street_address
FROM employees e JOIN departments d
ON(e.department_id = d.department_id)
JOIN locations l
WHERE e.job_id = 'IT_PROG';

SELECT first_name, last_name, department_name, city, postal_code, street_address
FROM employees e JOIN departments d
ON(e.department_id = d.department_id)
AND e.job_id = 'IT_PROG';
```

Restricting+Joins+(Code+Samples).sql

By Eslam Khder

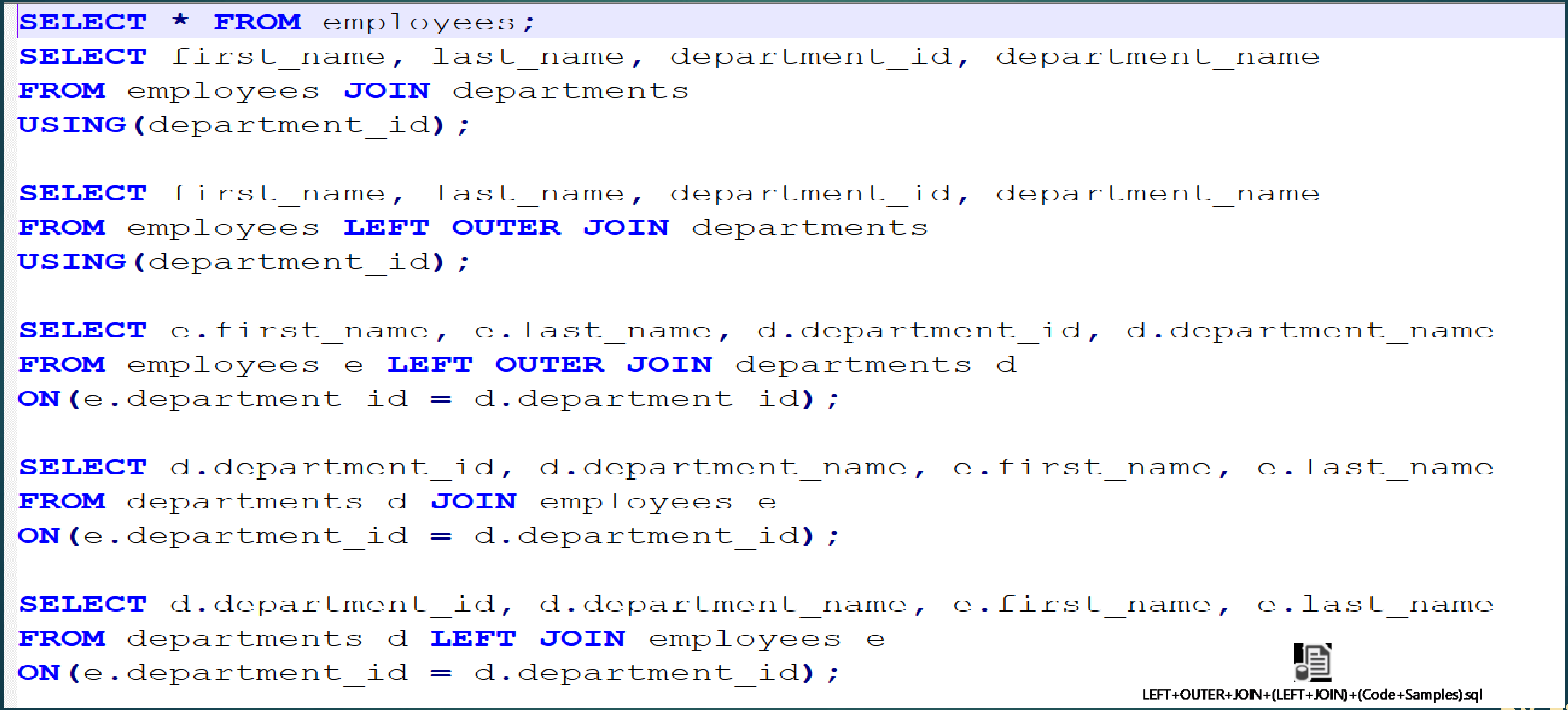
Outer Join

LEFT JOIN (or LEFT OUTER JOIN) RIGHT JOIN (or RIGHT OUTER JOIN) FULL JOIN (or FULL OUTER JOIN)

```
SELECT first_name, last_name, department_name
FROM employees JOIN departments
USING(department_id);

SELECT * FROM departments;

SELECT d.department_id, d.department_name, e.first_name, e.last_name
FROM departments d JOIN employees e
ON (d.manager_id = e.employee_id);
```

OUTER+JOINS+(Code+Samples).sql

LEFT JOIN (or LEFT OUTER JOIN)



```sql
SELECT * FROM employees;
SELECT first_name, last_name, department_id, department_name
FROM employees JOIN departments
USING(department_id);

SELECT first_name, last_name, department_id, department_name
FROM employees LEFT OUTER JOIN departments
USING(department_id);

SELECT e.first_name, e.last_name, d.department_id, d.department_name
FROM employees e LEFT OUTER JOIN departments d
ON(e.department_id = d.department_id);

SELECT d.department_id, d.department_name, e.first_name, e.last_name
FROM departments d JOIN employees e
ON(e.department_id = d.department_id);

SELECT d.department_id, d.department_name, e.first_name, e.last_name
FROM departments d LEFT JOIN employees e
ON(e.department_id = d.department_id);
```

LEFT+OUTER+JOIN+(LEFT+JOIN)+(Code+Samples).sql

By Eslam Khder

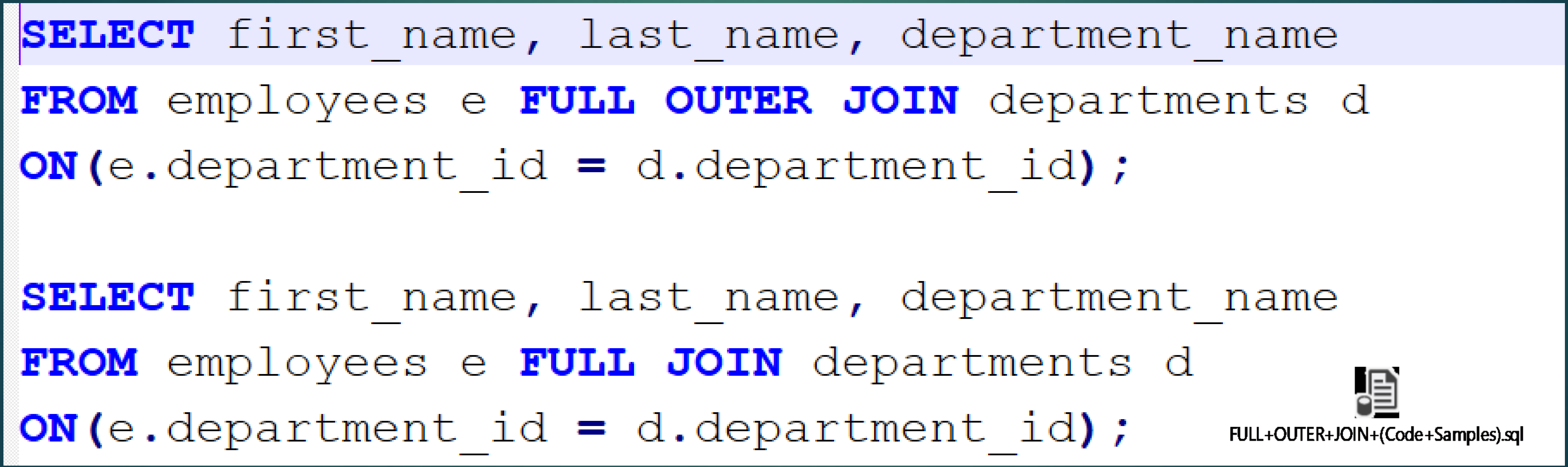
RIGHT JOIN (or RIGHT OUTER JOIN)

```
SELECT count(*) FROM employees;
SELECT count(*) FROM departments;

SELECT first_name, last_name, department_name
FROM employees e RIGHT OUTER JOIN departments d
ON(e.department_id = d.department_id);

SELECT first_name, last_name, department_name, e.department_id, d.department_id
FROM employees e RIGHT OUTER JOIN departments d
ON(e.department_id = d.department_id);

SELECT first_name, last_name, department_name, e.department_id, d.department_id
FROM employees e LEFT OUTER JOIN departments d
ON(e.department_id = d.department_id);

SELECT first_name, last_name, department_name, e.department_id, d.department_id
FROM departments d LEFT OUTER JOIN employees e
ON(e.department_id = d.department_id);
```

RIGHT+OUTER+JOIN+(RIGHT+JOIN)+(Code+Samples).sql

By Eslam Khder

FULL JOIN (or FULL OUTER JOIN)

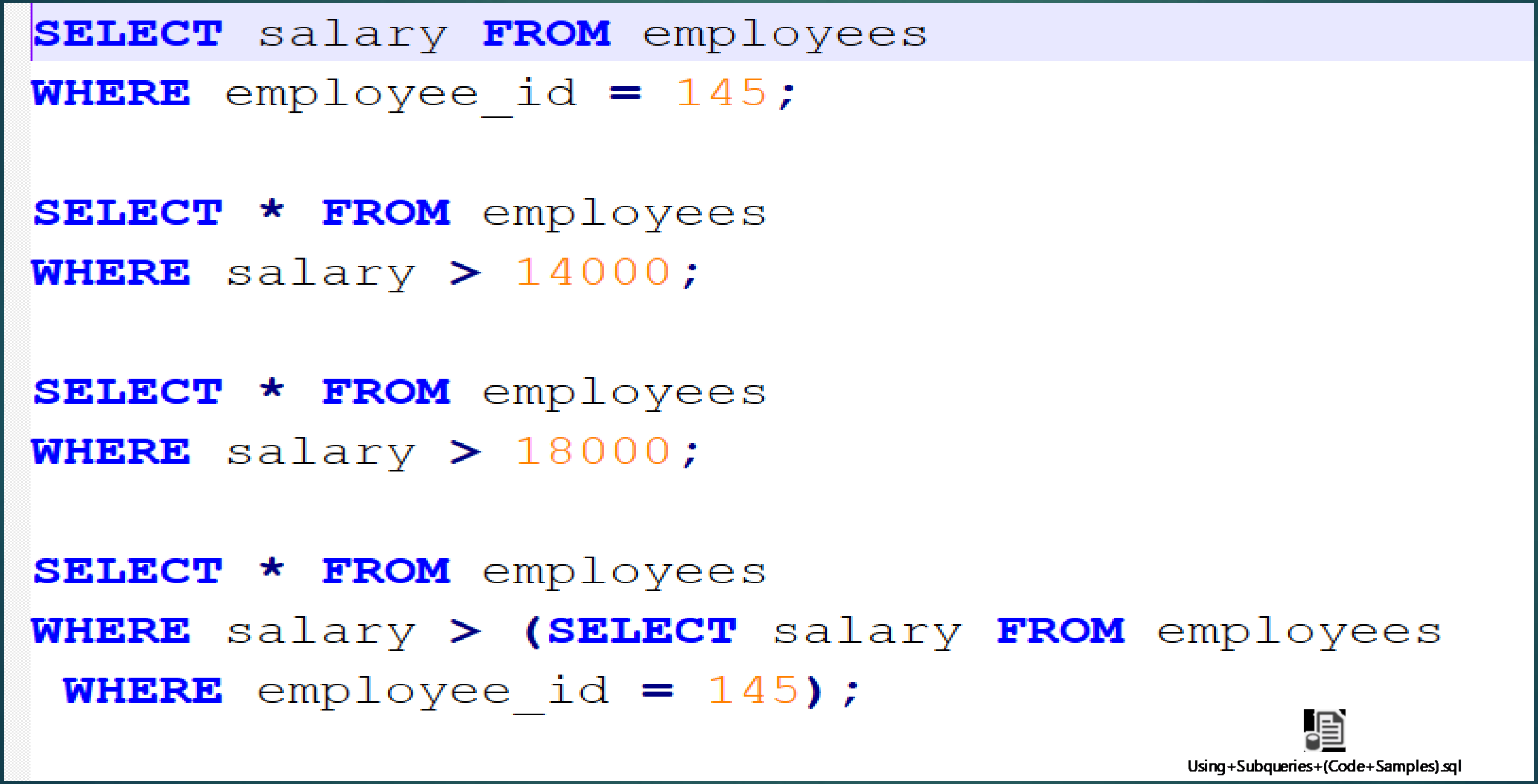


```sql
SELECT first_name, last_name, department_name
FROM employees e FULL OUTER JOIN departments d
ON(e.department_id = d.department_id);

SELECT first_name, last_name, department_name
FROM employees e FULL JOIN departments d
ON(e.department_id = d.department_id);
```

FULL+OUTER+JOIN+(Code+Samples).sql

By Eslam Khder

Using Subqueries



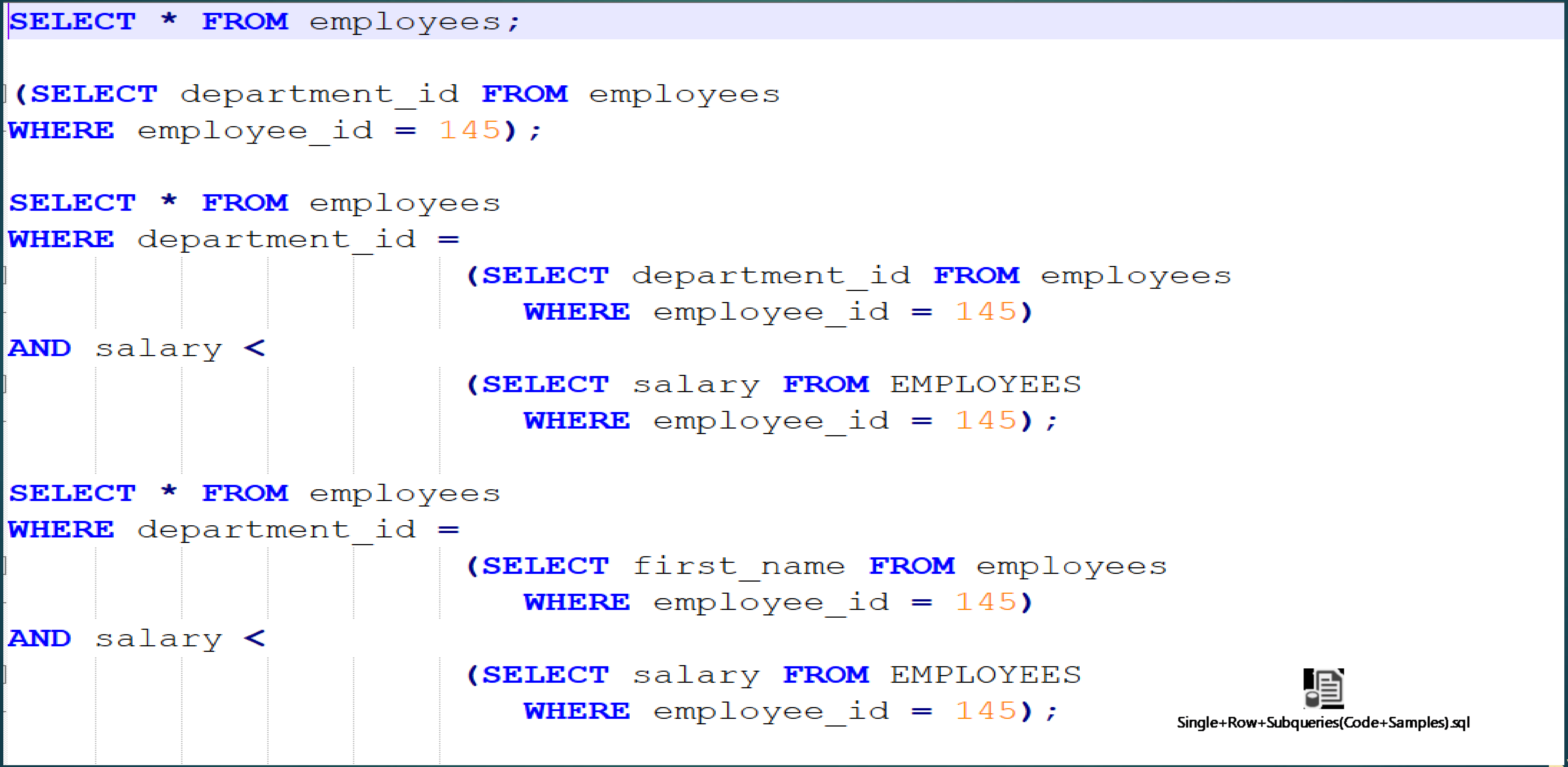
```
SELECT salary FROM employees
WHERE employee_id = 145;

SELECT * FROM employees
WHERE salary > 14000;

SELECT * FROM employees
WHERE salary > 18000;

SELECT * FROM employees
WHERE salary > (SELECT salary FROM employees
 WHERE employee_id = 145);
```

Using+Subqueries+(Code+Samples).sql

By Eslam Khder

EraaSoft

Single Row Subqueries

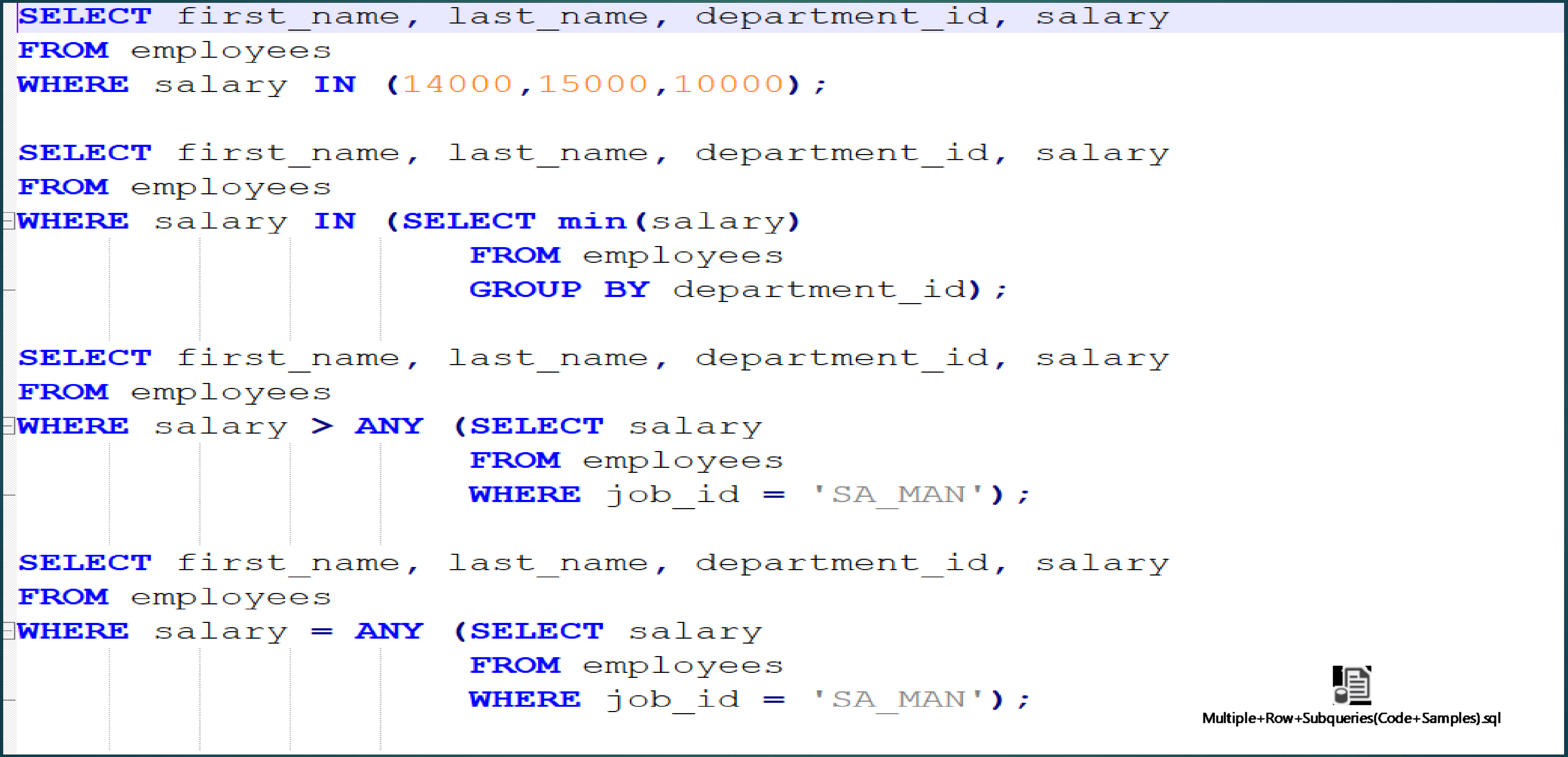


```
SELECT * FROM employees;

(SELECT department_id FROM employees
WHERE employee_id = 145);

SELECT * FROM employees
WHERE department_id =
                        (SELECT department_id FROM employees
                           WHERE employee_id = 145)
AND salary <
                        (SELECT salary FROM EMPLOYEES
                           WHERE employee_id = 145);

SELECT * FROM employees
WHERE department_id =
                        (SELECT first_name FROM employees
                           WHERE employee_id = 145)
AND salary <
                        (SELECT salary FROM EMPLOYEES
                           WHERE employee_id = 145);
```

Single+Row+Subqueries(Code+Samples).sql

By Eslam Khder

Multiple Row Subqueries



```
SELECT first_name, last_name, department_id, salary
FROM employees
WHERE salary IN (14000,15000,10000);

SELECT first_name, last_name, department_id, salary
FROM employees
WHERE salary IN (SELECT min(salary)
                 FROM employees
                 GROUP BY department_id);

SELECT first_name, last_name, department_id, salary
FROM employees
WHERE salary > ANY (SELECT salary
                    FROM employees
                    WHERE job_id = 'SA_MAN');

SELECT first_name, last_name, department_id, salary
FROM employees
WHERE salary = ANY (SELECT salary
                    FROM employees
                    WHERE job_id = 'SA_MAN');
```

Multiple+Row+Subqueries(Code+Samples).sql

By Eslam Khder

By Eslam Khder