POLITÉCNICO
DO PORTO
ESCOLA
SUPERIOR
DE MEDIA ARTES
E DESIGN

P.PORTO

BASES DE DADOS
SQL
Data Manipulation Language
Part II

TECNOLOGIAS E SISTEMAS DE INFORMAÇÃO PARA A WEB

Agenda

SQL – Structured Query Language – Data Manipulation Language

- ❖ SELECT with mutiple tables (*Join*)
  - ❖ Inner Join
  - ❖ Outer Joins
    - ❖ LEFT JOIN
    - ❖ RIGHT JOIN

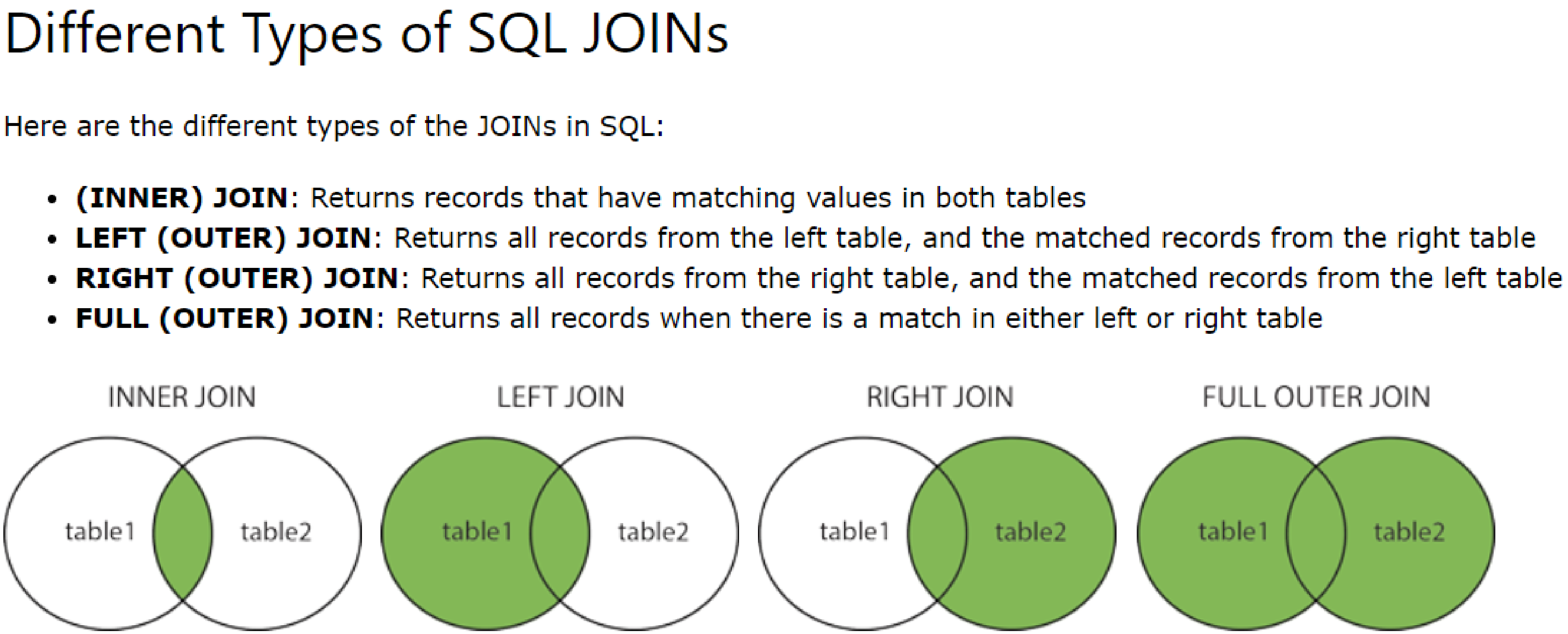
SQL

- Join concept
  - Join tables allows you to get data from more than one database table, as long as they relate to each other
  - A JOIN clause is used to combine rows from two or more tables, based on a related column between them
  - The join of two or more tables materializes through established relationships, these being expressed in the WHERE clause.

```
SELECT [distinct/all] field, field2, …, fieldn [*]
FROM table1, table2
[WHERE conditions]
[GROUP BY fields]
[HAVING conditions]
[ORDER BY fields]
```

SQL

- Join concept



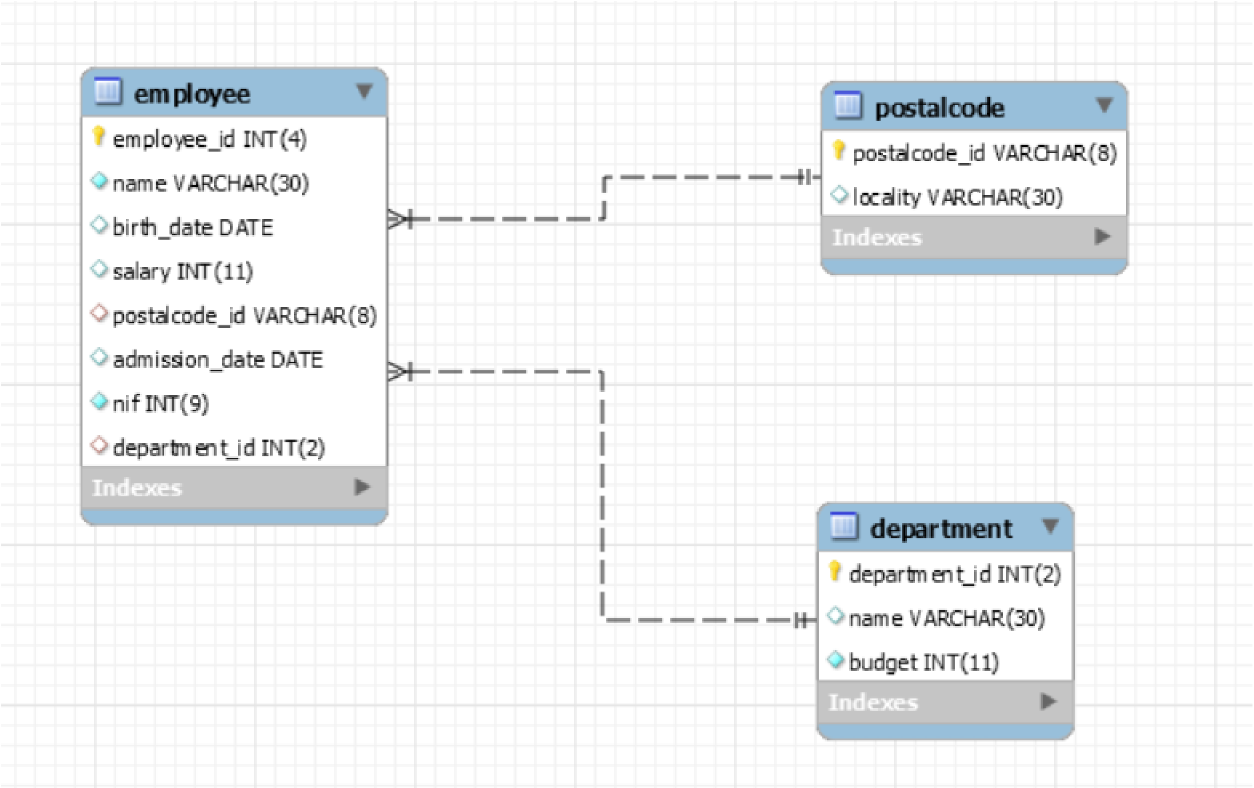
Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

- **(INNER) JOIN**: Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN**: Returns all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN**: Returns all records when there is a match in either left or right table

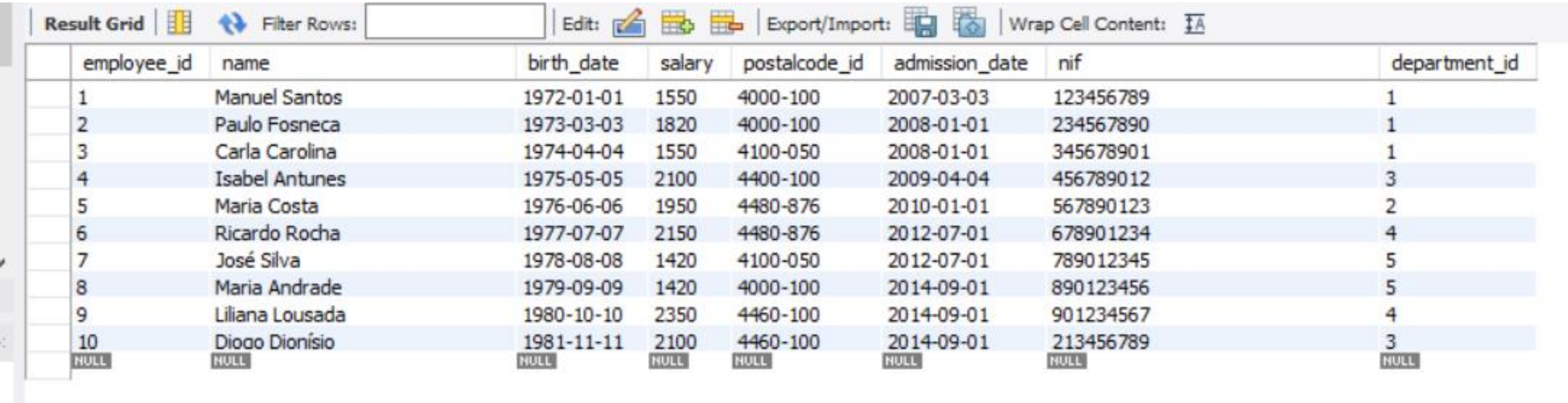
SQL

- Consider the following database schema :

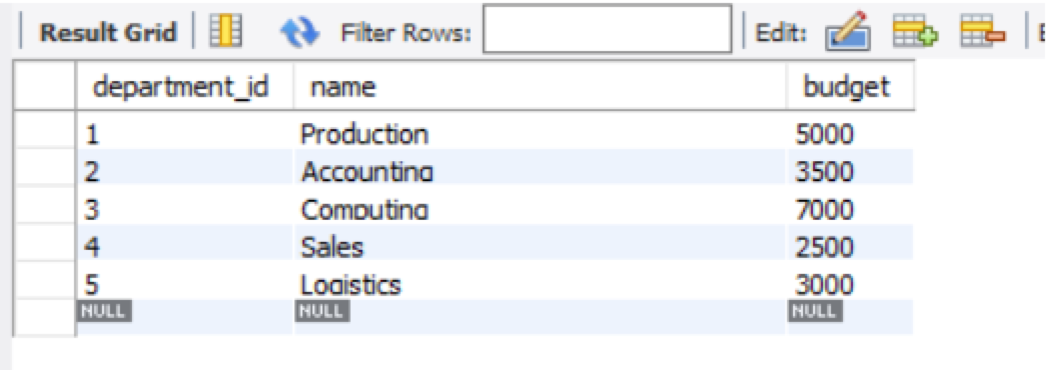


SQL

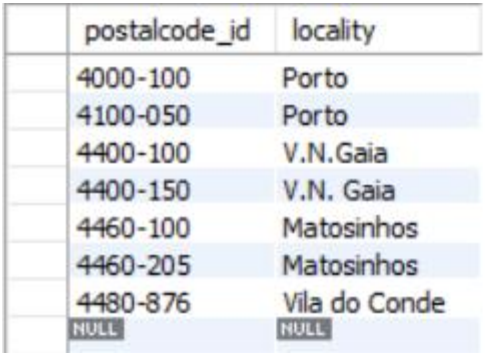
- With the following sample data:



employee_id	name	birth_date	salary	postalcode_id	admission_date	nif	department_id
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1
2	Paulo Fosneca	1973-03-03	1820	4000-100	2008-01-01	234567890	1
3	Carla Carolina	1974-04-04	1550	4100-050	2008-01-01	345678901	1
4	Isabel Antunes	1975-05-05	2100	4400-100	2009-04-04	456789012	3
5	Maria Costa	1976-06-06	1950	4480-876	2010-01-01	567890123	2
6	Ricardo Rocha	1977-07-07	2150	4480-876	2012-07-01	678901234	4
7	José Silva	1978-08-08	1420	4100-050	2012-07-01	789012345	5
8	Maria Andrade	1979-09-09	1420	4000-100	2014-09-01	890123456	5
9	Liliana Lousada	1980-10-10	2350	4460-100	2014-09-01	901234567	4
10	Diogo Dionísio	1981-11-11	2100	4460-100	2014-09-01	213456789	3
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



department_id	name	budget
1	Production	5000
2	Accounting	3500
3	Computing	7000
4	Sales	2500
5	Logistics	3000
NULL	NULL	NULL



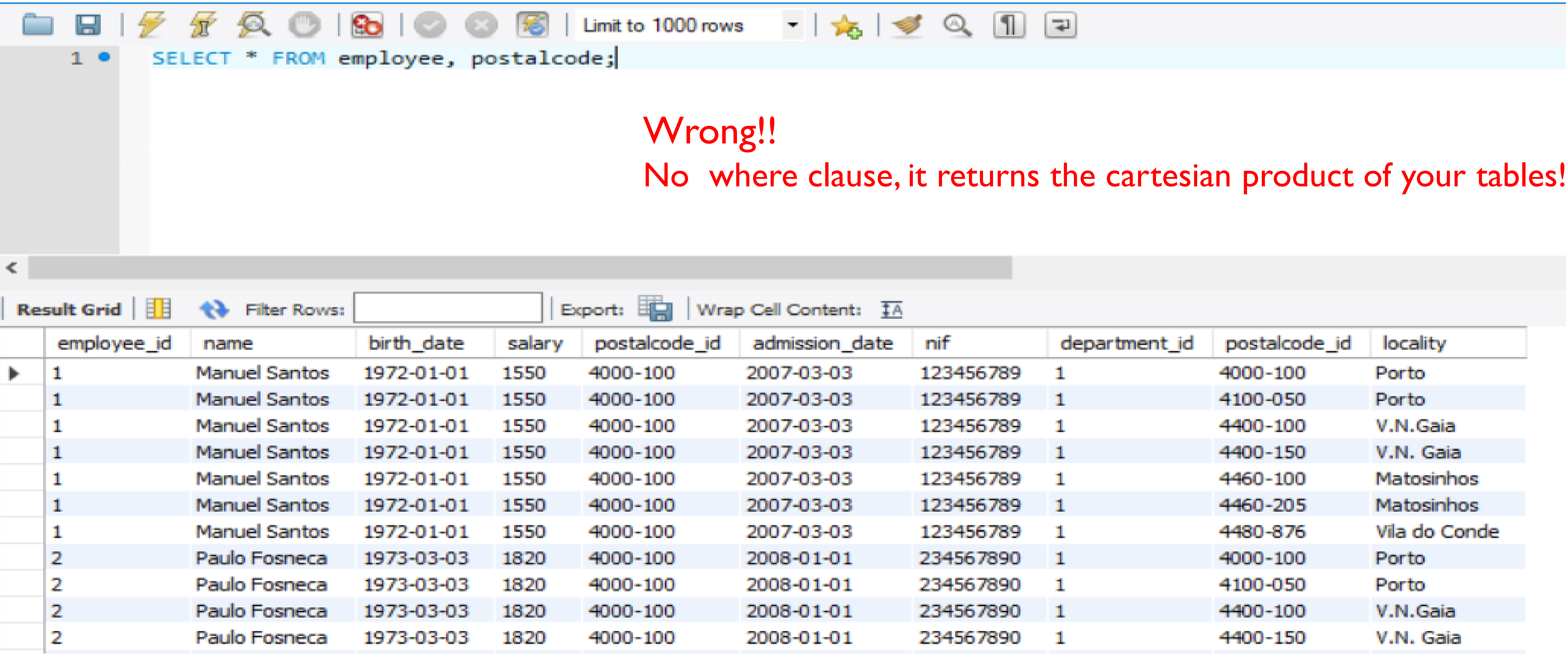
postalcode_id	locality
4000-100	Porto
4100-050	Porto
4400-100	V.N.Gaia
4400-150	V.N. Gaia
4460-100	Matosinhos
4460-205	Matosinhos
4480-876	Vila do Conde
NULL	NULL

SQL

The joining of two tables is done through the relationship between them, expressed in the

- WHERE clause
- INNER JOIN clause

If we do not, we get the cartesian product of your tables.



```sql
SELECT * FROM employee, postalcode;
```

Wrong!!
No where clause, it returns the cartesian product of your tables!

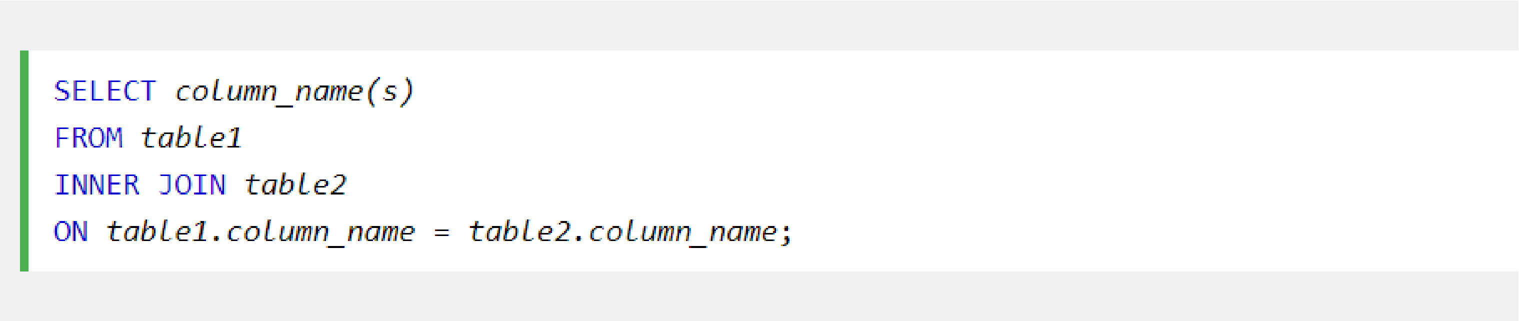
employee_id	name	birth_date	salary	postalcode_id	admission_date	nif	department_id	postalcode_id	locality
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1	4000-100	Porto
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1	4100-050	Porto
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1	4400-100	V.N.Gaia
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1	4400-150	V.N. Gaia
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1	4460-100	Matosinhos
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1	4460-205	Matosinhos
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1	4480-876	Vila do Conde
2	Paulo Fosneca	1973-03-03	1820	4000-100	2008-01-01	234567890	1	4000-100	Porto
2	Paulo Fosneca	1973-03-03	1820	4000-100	2008-01-01	234567890	1	4100-050	Porto
2	Paulo Fosneca	1973-03-03	1820	4000-100	2008-01-01	234567890	1	4400-100	V.N.Gaia
2	Paulo Fosneca	1973-03-03	1820	4000-100	2008-01-01	234567890	1	4400-150	V.N. Gaia

SQL

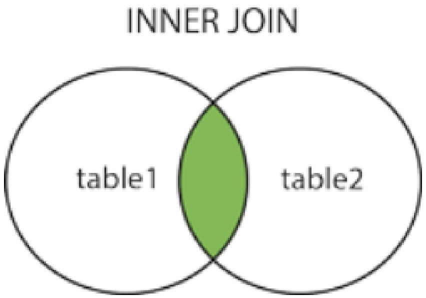
Inner Join

- Inner-Join: The join between 2 or more tables is done through the primary key of a table and the foreign key of the other table

INNER JOIN Syntax



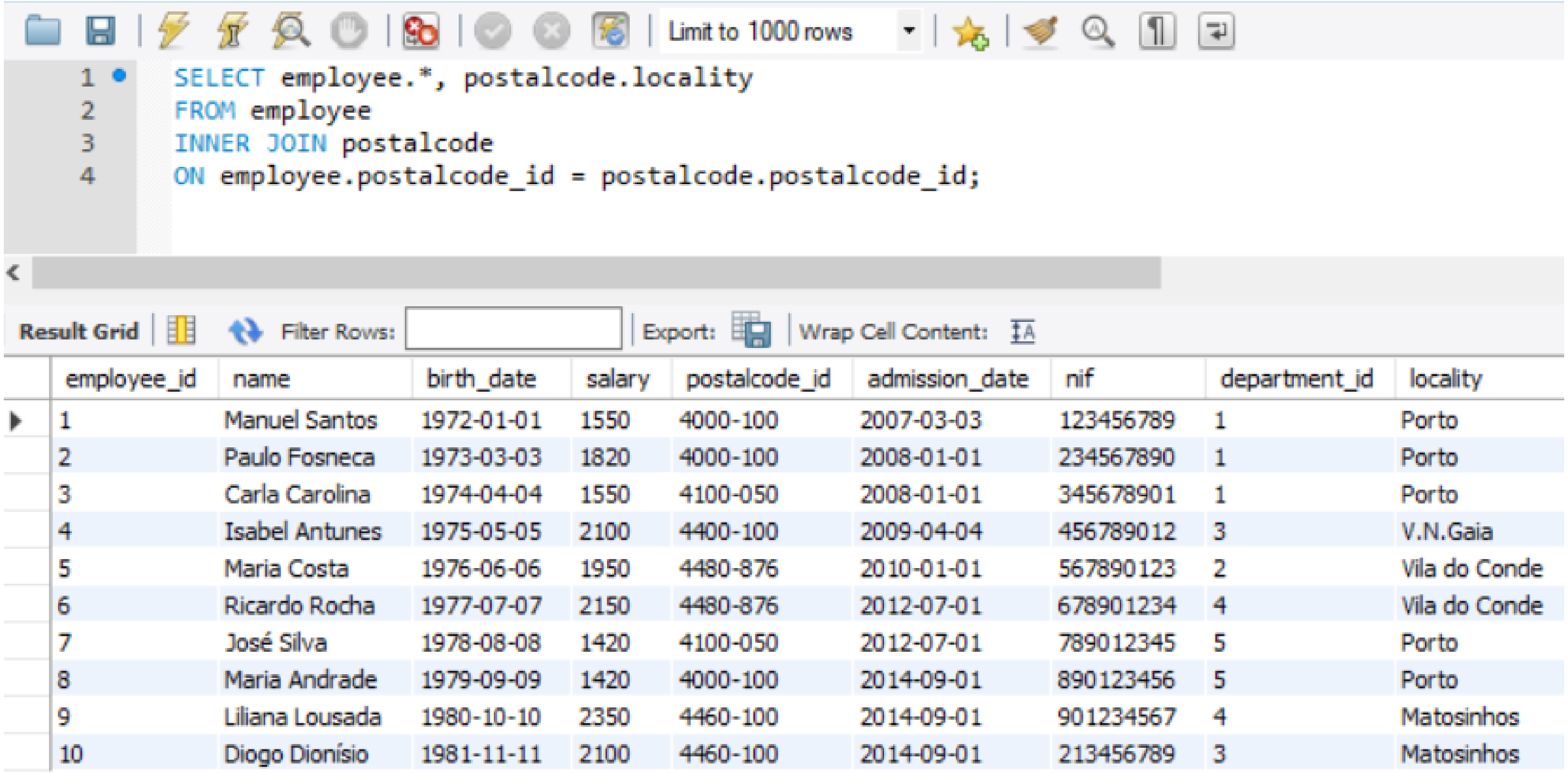
```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```



SQL

Inner Join

- Inner-Join: The join between 2 or more tables is done through the primary key of a table and the foreign key of the other table

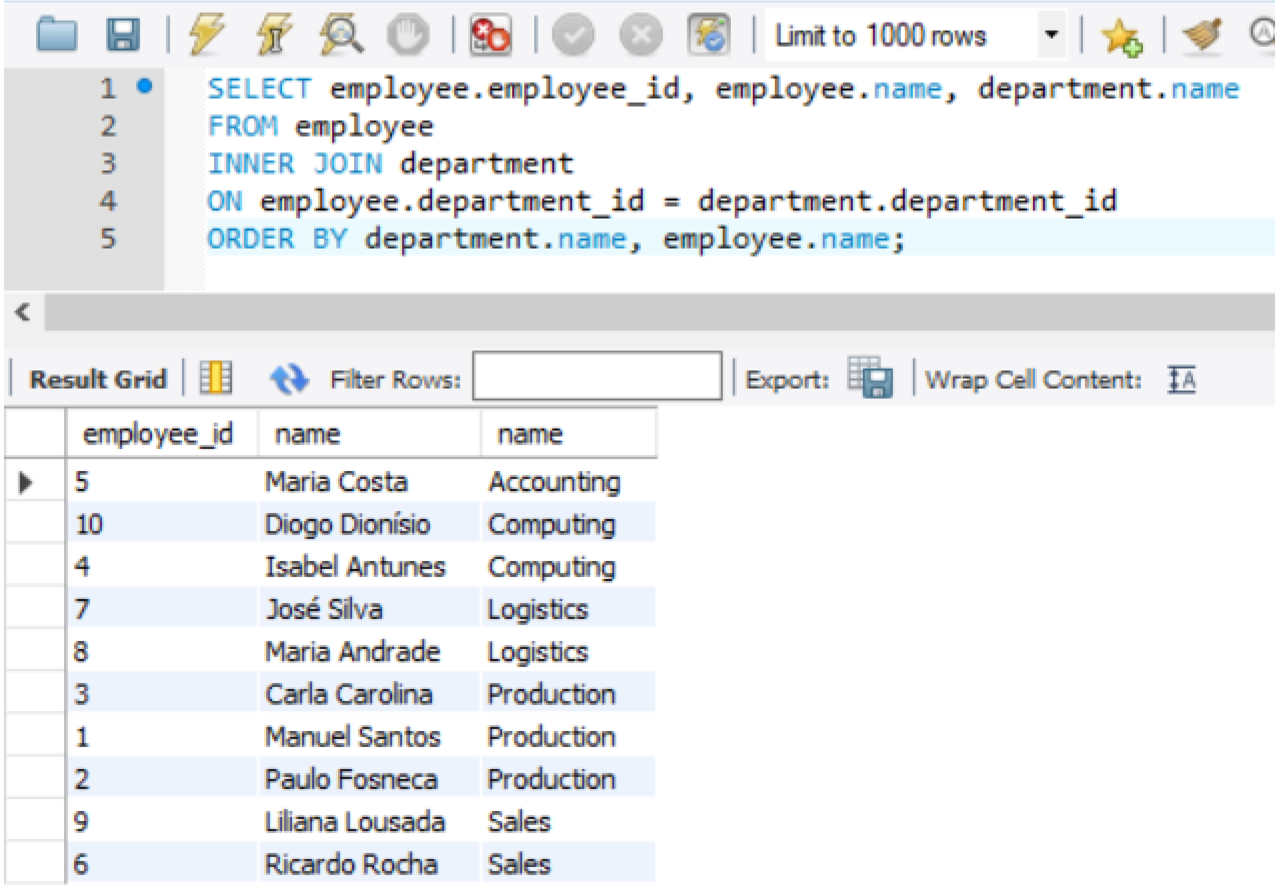


```sql
SELECT employee.*, postalcode.locality
FROM employee
INNER JOIN postalcode
ON employee.postalcode_id = postalcode.postalcode_id;
```

employee_id	name	birth_date	salary	postalcode_id	admission_date	nif	department_id	locality
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1	Porto
2	Paulo Fosneca	1973-03-03	1820	4000-100	2008-01-01	234567890	1	Porto
3	Carla Carolina	1974-04-04	1550	4100-050	2008-01-01	345678901	1	Porto
4	Isabel Antunes	1975-05-05	2100	4400-100	2009-04-04	456789012	3	V.N.Gaia
5	Maria Costa	1976-06-06	1950	4480-876	2010-01-01	567890123	2	Vila do Conde
6	Ricardo Rocha	1977-07-07	2150	4480-876	2012-07-01	678901234	4	Vila do Conde
7	José Silva	1978-08-08	1420	4100-050	2012-07-01	789012345	5	Porto
8	Maria Andrade	1979-09-09	1420	4000-100	2014-09-01	890123456	5	Porto
9	Liliana Lousada	1980-10-10	2350	4460-100	2014-09-01	901234567	4	Matosinhos
10	Diogo Dionísio	1981-11-11	2100	4460-100	2014-09-01	213456789	3	Matosinhos

SQL

Inner Join

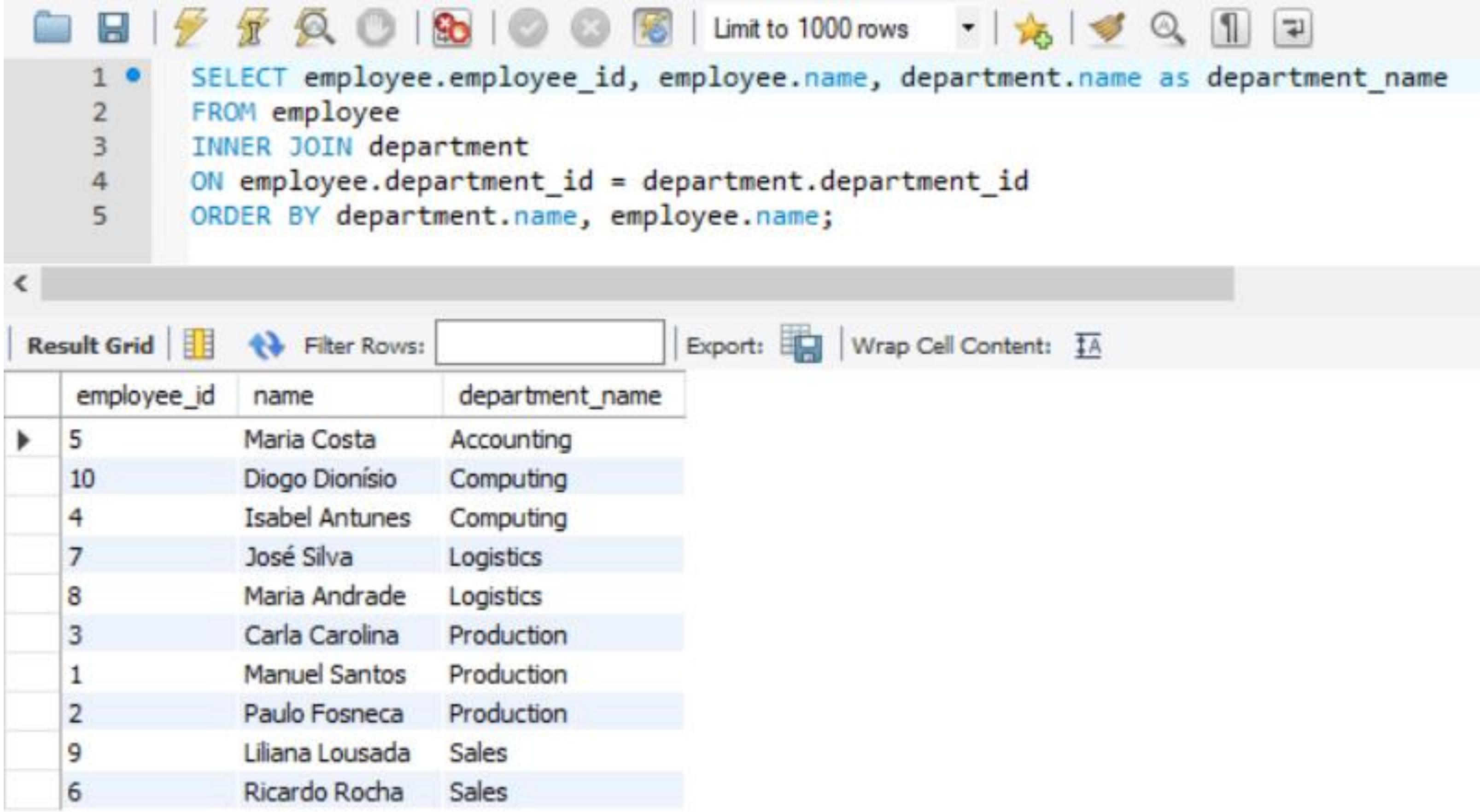


```sql
SELECT employee.employee_id, employee.name, department.name
FROM employee
INNER JOIN department
ON employee.department_id = department.department_id
ORDER BY department.name, employee.name;
```

employee_id	name	name
5	Maria Costa	Accounting
10	Diogo Dionísio	Computing
4	Isabel Antunes	Computing
7	José Silva	Logistics
8	Maria Andrade	Logistics
3	Carla Carolina	Production
1	Manuel Santos	Production
2	Paulo Fosneca	Production
9	Liliana Lousada	Sales
6	Ricardo Rocha	Sales

SQL

Inner Join



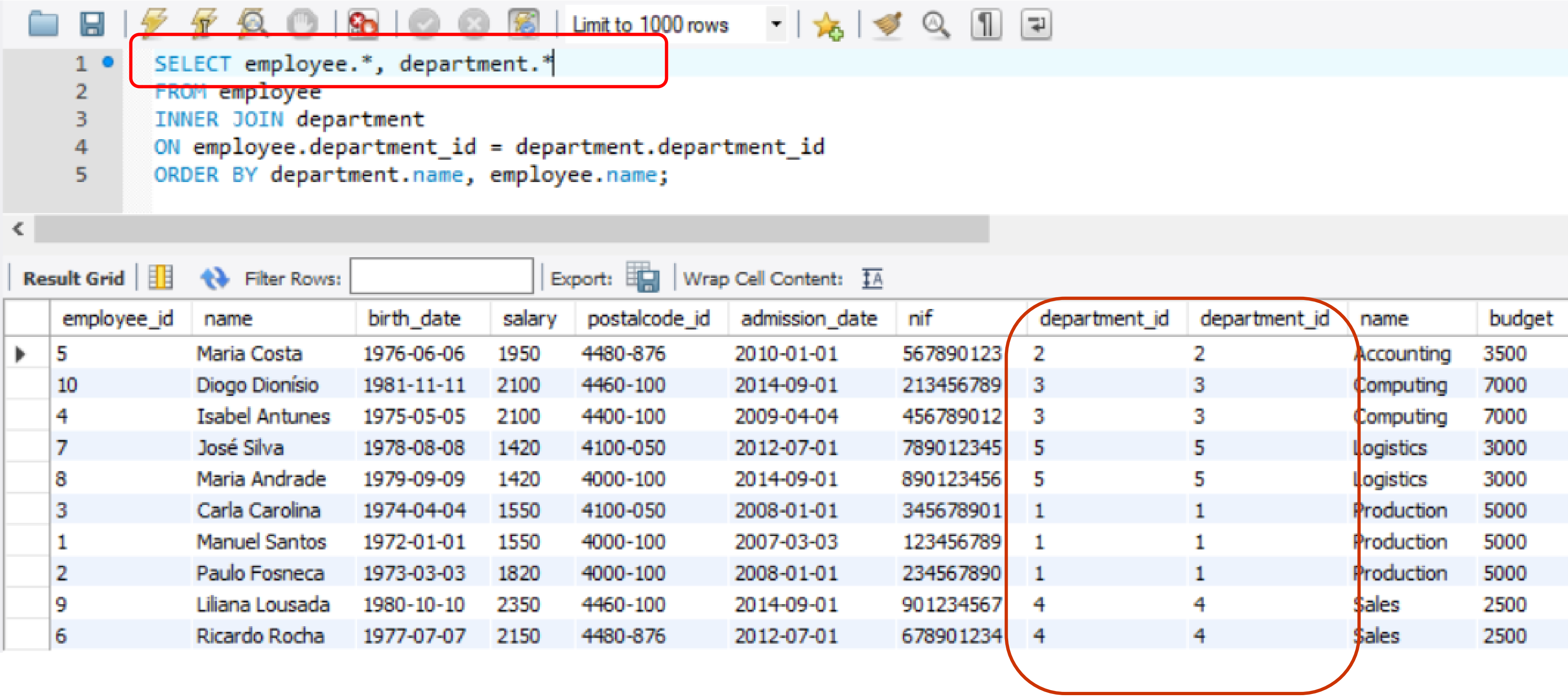
```sql
SELECT employee.employee_id, employee.name, department.name as department_name
FROM employee
INNER JOIN department
ON employee.department_id = department.department_id
ORDER BY department.name, employee.name;
```

employee_id	name	department_name
5	Maria Costa	Accounting
10	Diogo Dionísio	Computing
4	Isabel Antunes	Computing
7	José Silva	Logistics
8	Maria Andrade	Logistics
3	Carla Carolina	Production
1	Manuel Santos	Production
2	Paulo Fosneca	Production
9	Liliana Lousada	Sales
6	Ricardo Rocha	Sales

SQL

Inner Join

- Equi-Join: when we select all the columns of 2 or more tables and the connection between them is made by an equality, thus giving rise two columns of identical data



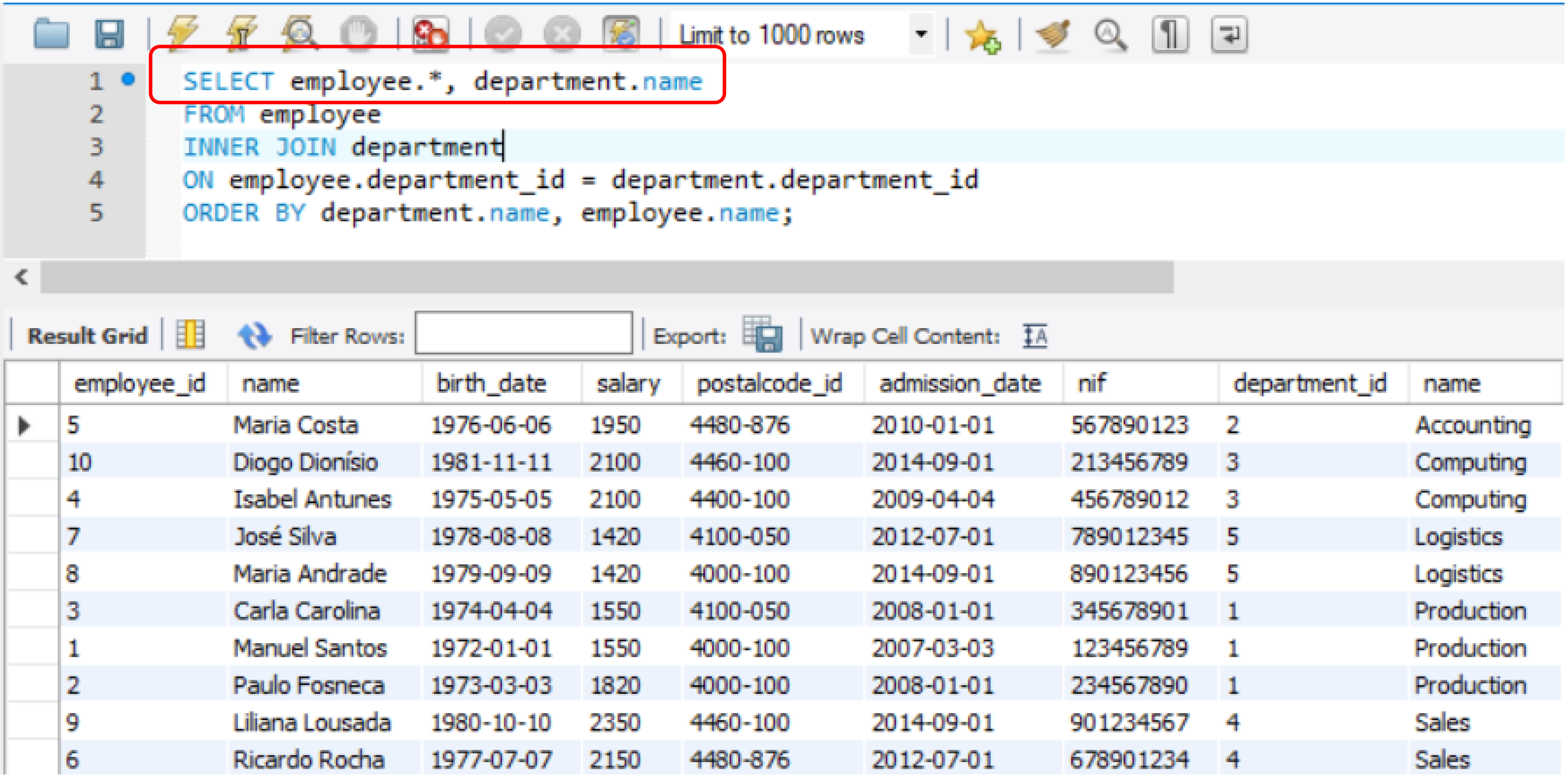
```
SELECT employee.*, department.*
FROM employee
INNER JOIN department
ON employee.department_id = department.department_id
ORDER BY department.name, employee.name;
```

employee_id	name	birth_date	salary	postalcode_id	admission_date	nif	department_id	department_id	name	budget
5	Maria Costa	1976-06-06	1950	4480-876	2010-01-01	567890123	2	2	Accounting	3500
10	Diogo Dionísio	1981-11-11	2100	4460-100	2014-09-01	213456789	3	3	Computing	7000
4	Isabel Antunes	1975-05-05	2100	4400-100	2009-04-04	456789012	3	3	Computing	7000
7	José Silva	1978-08-08	1420	4100-050	2012-07-01	789012345	5	5	Logistics	3000
8	Maria Andrade	1979-09-09	1420	4000-100	2014-09-01	890123456	5	5	Logistics	3000
3	Carla Carolina	1974-04-04	1550	4100-050	2008-01-01	345678901	1	1	Production	5000
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1	1	Production	5000
2	Paulo Fosneca	1973-03-03	1820	4000-100	2008-01-01	234567890	1	1	Production	5000
9	Liliana Lousada	1980-10-10	2350	4460-100	2014-09-01	901234567	4	4	Sales	2500
6	Ricardo Rocha	1977-07-07	2150	4480-876	2012-07-01	678901234	4	4	Sales	2500

SQL

Inner Join

- Natural-Join: when we select data from 2 or more tables and the connection between them is made by an equality, without repeating columns.



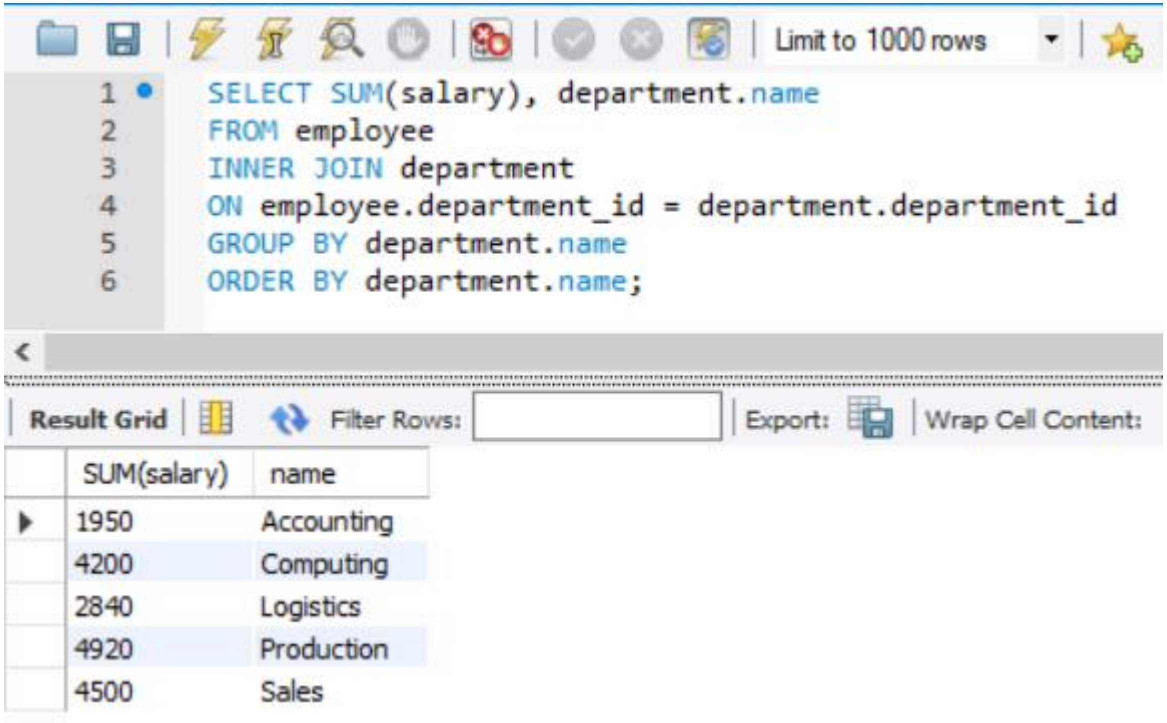
```
SELECT employee.*, department.name
FROM employee
INNER JOIN department
ON employee.department_id = department.department_id
ORDER BY department.name, employee.name;
```

employee_id	name	birth_date	salary	postalcode_id	admission_date	nif	department_id	name
5	Maria Costa	1976-06-06	1950	4480-876	2010-01-01	567890123	2	Accounting
10	Diogo Dionísio	1981-11-11	2100	4460-100	2014-09-01	213456789	3	Computing
4	Isabel Antunes	1975-05-05	2100	4400-100	2009-04-04	456789012	3	Computing
7	José Silva	1978-08-08	1420	4100-050	2012-07-01	789012345	5	Logistics
8	Maria Andrade	1979-09-09	1420	4000-100	2014-09-01	890123456	5	Logistics
3	Carla Carolina	1974-04-04	1550	4100-050	2008-01-01	345678901	1	Production
1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1	Production
2	Paulo Fosneca	1973-03-03	1820	4000-100	2008-01-01	234567890	1	Production
9	Liliana Lousada	1980-10-10	2350	4460-100	2014-09-01	901234567	4	Sales
6	Ricardo Rocha	1977-07-07	2150	4480-876	2012-07-01	678901234	4	Sales

SQL

Inner Join

- Inner Join between 2 tables, including a GROUP BY clause



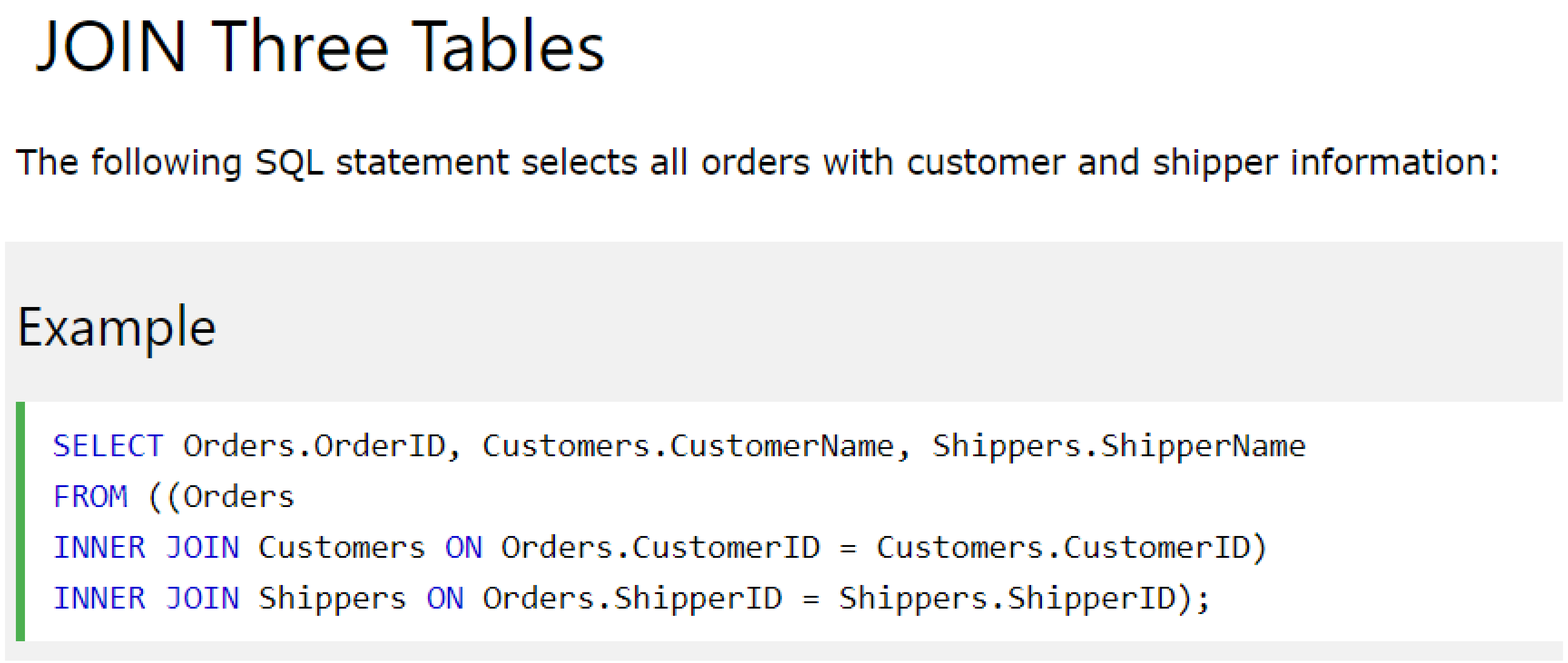
```sql
SELECT SUM(salary), department.name
FROM employee
INNER JOIN department
ON employee.department_id = department.department_id
GROUP BY department.name
ORDER BY department.name;
```

SUM(salary)	name
1950	Accounting
4200	Computing
2840	Logistics
4920	Production
4500	Sales

SQL

Inner Join

- Inner Join between 3 tables



JOIN Three Tables

The following SQL statement selects all orders with customer and shipper information:

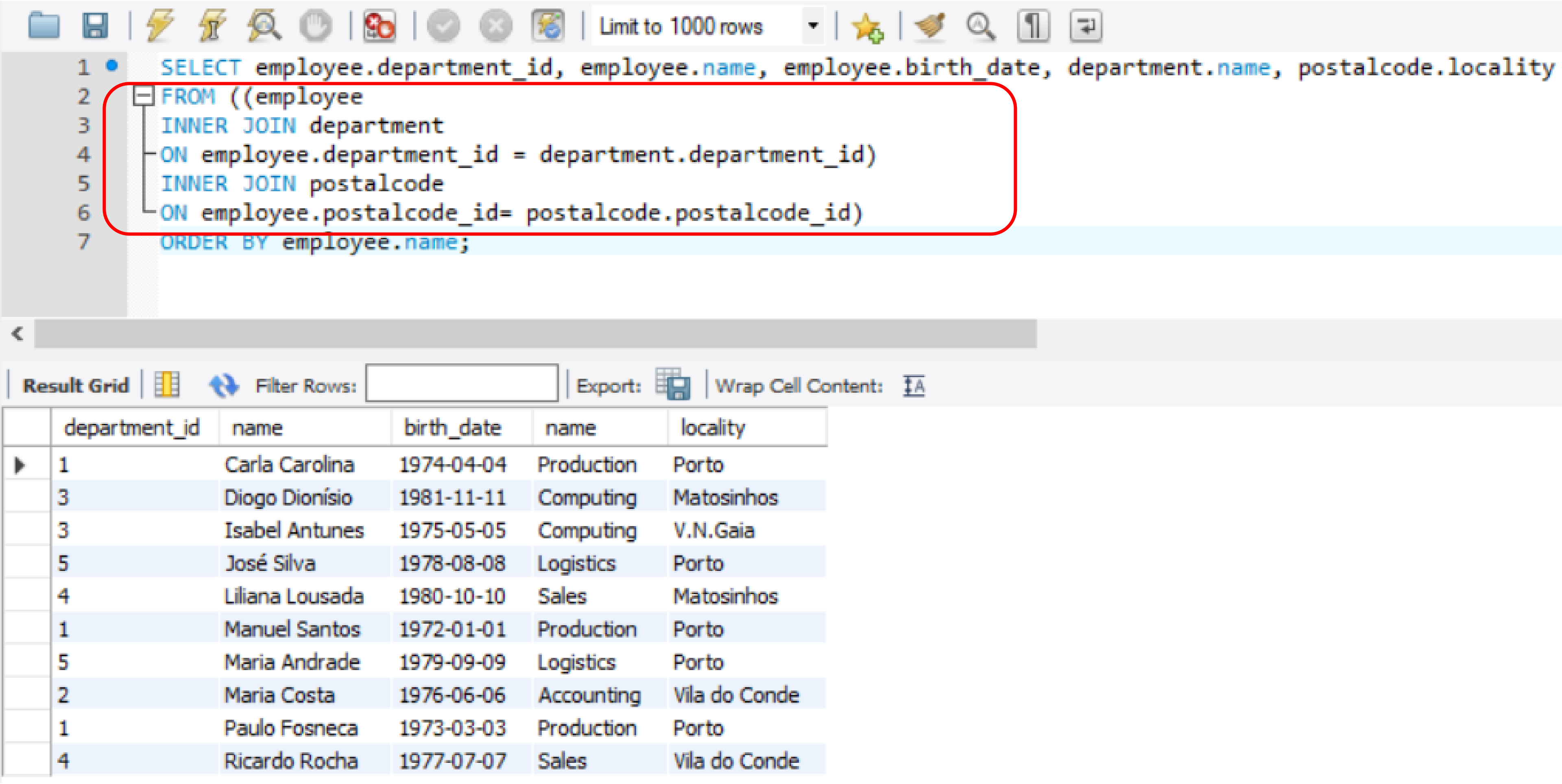
Example

```
SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName
FROM ((Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);
```

SQL

Inner Join

- Inner Join between 3 tables



```
SELECT employee.department_id, employee.name, employee.birth_date, department.name, postalcode.locality
FROM ((employee
INNER JOIN department
ON employee.department_id = department.department_id)
INNER JOIN postalcode
ON employee.postalcode_id= postalcode.postalcode_id)
ORDER BY employee.name;
```

department_id	name	birth_date	name	locality
1	Carla Carolina	1974-04-04	Production	Porto
3	Diogo Dionísio	1981-11-11	Computing	Matosinhos
3	Isabel Antunes	1975-05-05	Computing	V.N.Gaia
5	José Silva	1978-08-08	Logistics	Porto
4	Liliana Lousada	1980-10-10	Sales	Matosinhos
1	Manuel Santos	1972-01-01	Production	Porto
5	Maria Andrade	1979-09-09	Logistics	Porto
2	Maria Costa	1976-06-06	Accounting	Vila do Conde
1	Paulo Fosneca	1973-03-03	Production	Porto
4	Ricardo Rocha	1977-07-07	Sales	Vila do Conde

SQL

Outer Join

- LEFT JOIN

The LEFT JOIN keyword returns all records from the left table, even if there are no matches in the right table.

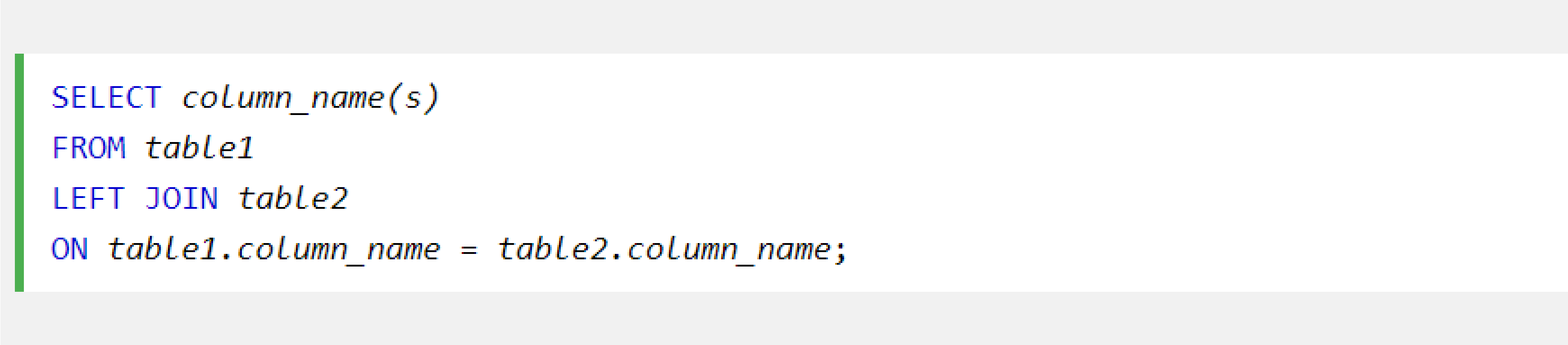
- RIGHT JOIN

The RIGHT JOIN keyword returns all records from the right table, even if there are no matches in the left table.

SQL

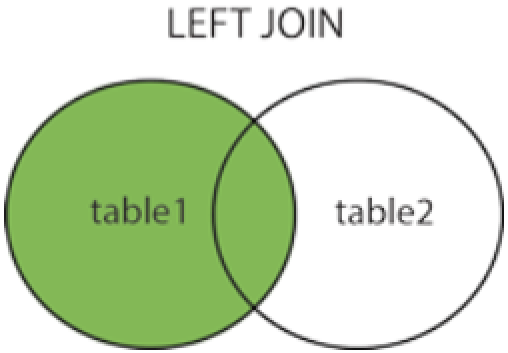
LEFT JOIN

LEFT JOIN Syntax



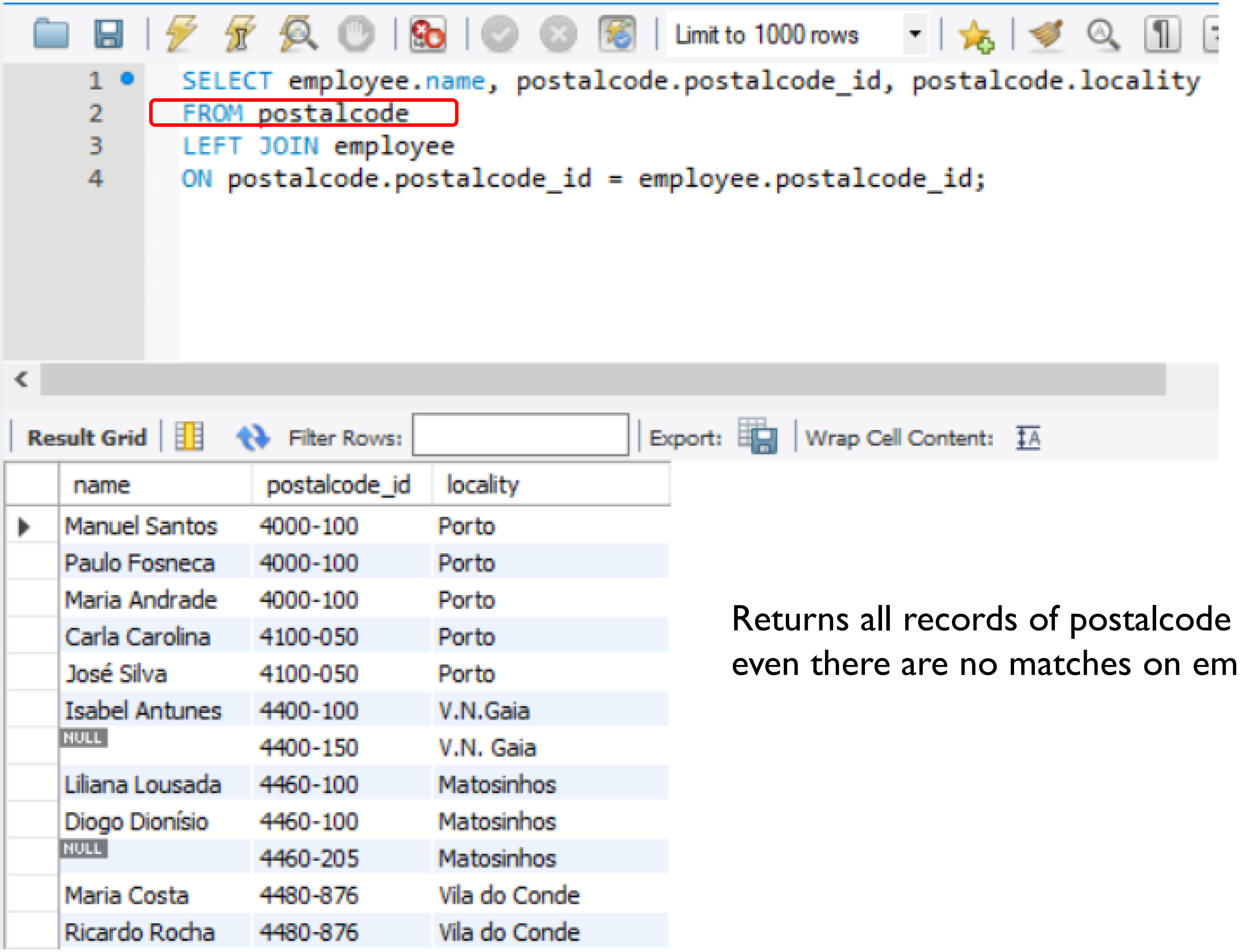
```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

Note: In some databases LEFT JOIN is called LEFT OUTER JOIN.



SQL

LEFT JOIN



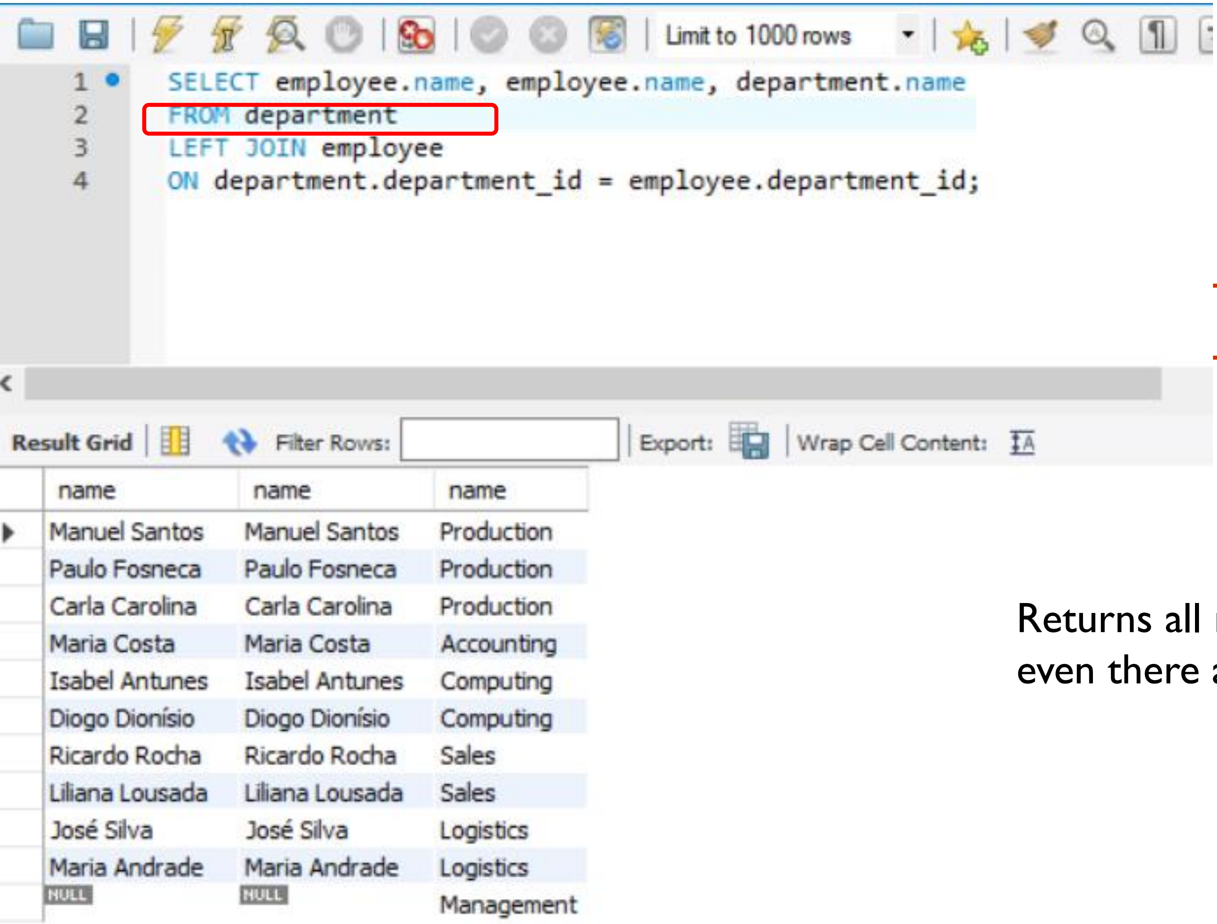
```
SELECT employee.name, postalcode.postalcode_id, postalcode.locality
FROM postalcode
LEFT JOIN employee
ON postalcode.postalcode_id = employee.postalcode_id;
```

name	postalcode_id	locality
Manuel Santos	4000-100	Porto
Paulo Fosneca	4000-100	Porto
Maria Andrade	4000-100	Porto
Carla Carolina	4100-050	Porto
José Silva	4100-050	Porto
Isabel Antunes	4400-100	V.N.Gaia
NULL	4400-150	V.N. Gaia
Liliana Lousada	4460-100	Matosinhos
Diogo Dionísio	4460-100	Matosinhos
NULL	4460-205	Matosinhos
Maria Costa	4480-876	Vila do Conde
Ricardo Rocha	4480-876	Vila do Conde

Returns all records of postalcode
even there are no matches on employee’s table

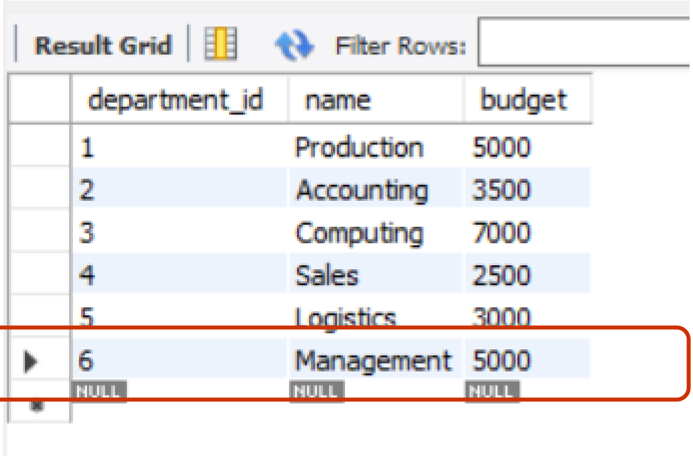
SQL

LEFT JOIN



```sql
SELECT employee.name, employee.name, department.name
FROM department
LEFT JOIN employee
ON department.department_id = employee.department_id;
```

name	name	name
Manuel Santos	Manuel Santos	Production
Paulo Fosneca	Paulo Fosneca	Production
Carla Carolina	Carla Carolina	Production
Maria Costa	Maria Costa	Accounting
Isabel Antunes	Isabel Antunes	Computing
Diogo Dionísio	Diogo Dionísio	Computing
Ricardo Rocha	Ricardo Rocha	Sales
Liliana Lousada	Liliana Lousada	Sales
José Silva	José Silva	Logistics
Maria Andrade	Maria Andrade	Logistics
NULL	NULL	Management



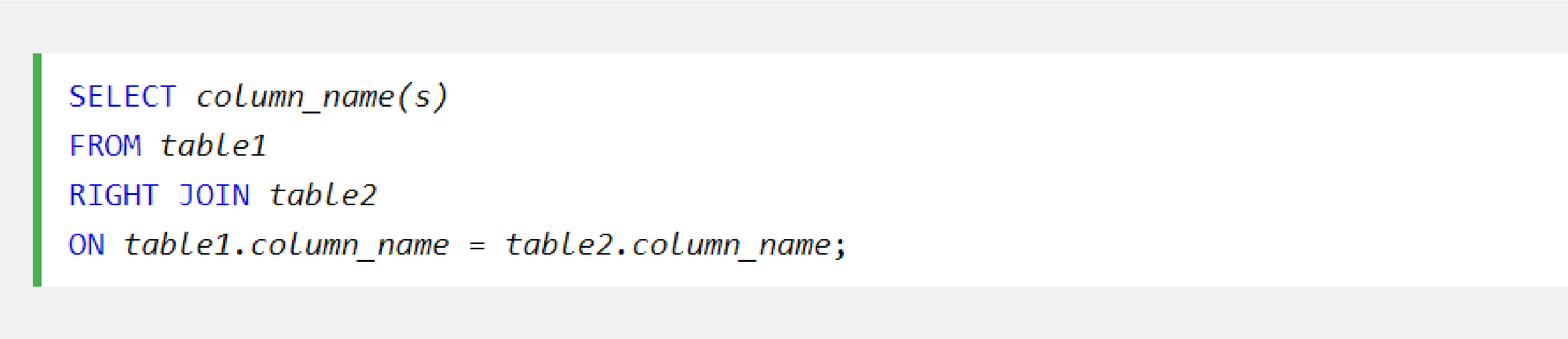
department_id	name	budget
1	Production	5000
2	Accounting	3500
3	Computing	7000
4	Sales	2500
5	Logistics	3000
6	Management	5000
NULL	NULL	NULL

Returns all records of department entity
even there are no matches on employee’s table

SQL

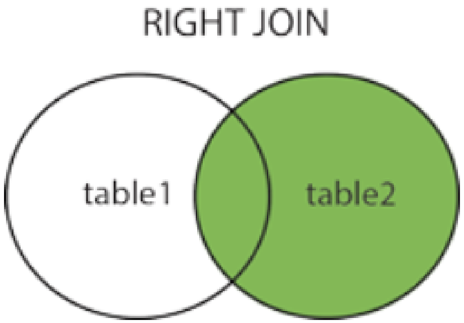
RIGHT JOIN

RIGHT JOIN Syntax



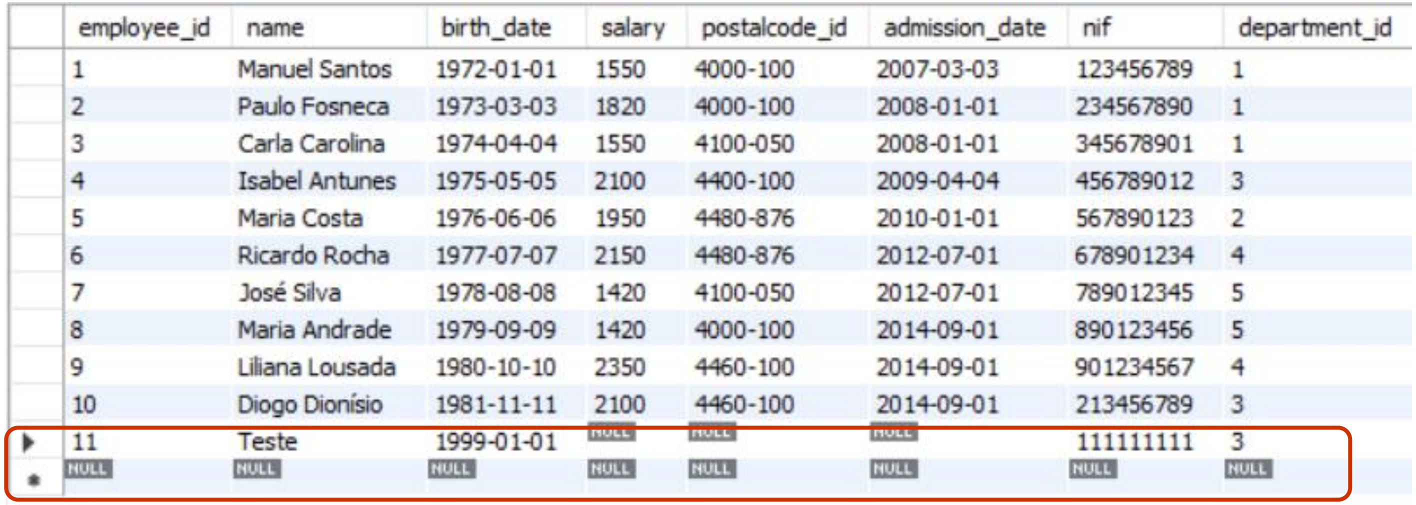
```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

Note: In some databases RIGHT JOIN is called RIGHT OUTER JOIN.



SQL

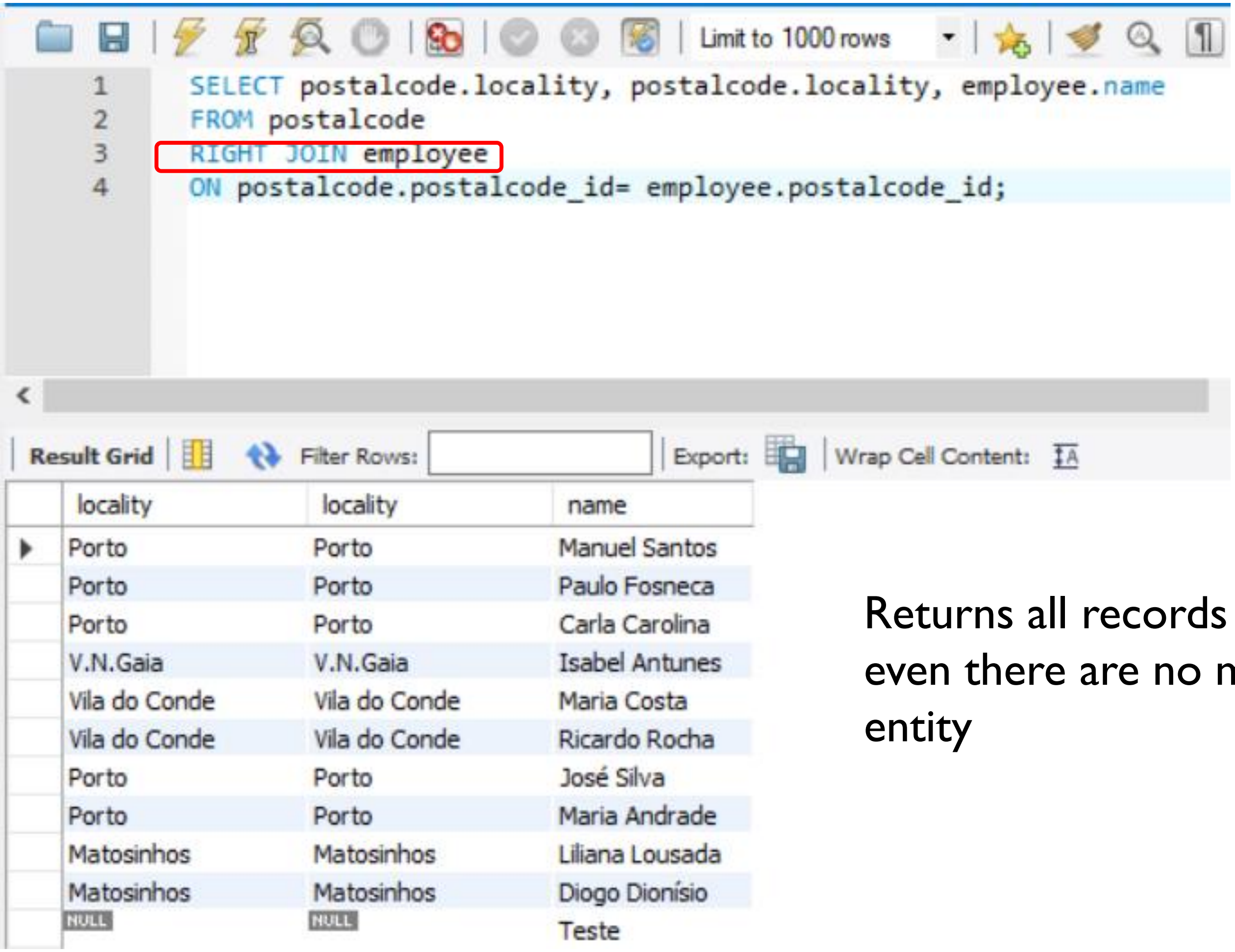
RIGHT JOIN



	employee_id	name	birth_date	salary	postalcode_id	admission_date	nif	department_id
	1	Manuel Santos	1972-01-01	1550	4000-100	2007-03-03	123456789	1
	2	Paulo Fosneca	1973-03-03	1820	4000-100	2008-01-01	234567890	1
	3	Carla Carolina	1974-04-04	1550	4100-050	2008-01-01	345678901	1
	4	Isabel Antunes	1975-05-05	2100	4400-100	2009-04-04	456789012	3
	5	Maria Costa	1976-06-06	1950	4480-876	2010-01-01	567890123	2
	6	Ricardo Rocha	1977-07-07	2150	4480-876	2012-07-01	678901234	4
	7	José Silva	1978-08-08	1420	4100-050	2012-07-01	789012345	5
	8	Maria Andrade	1979-09-09	1420	4000-100	2014-09-01	890123456	5
	9	Liliana Lousada	1980-10-10	2350	4460-100	2014-09-01	901234567	4
	10	Diogo Dionísio	1981-11-11	2100	4460-100	2014-09-01	213456789	3
▶	11	Teste	1999-01-01	NULL	NULL	NULL	111111111	3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SQL

RIGHT JOIN



```sql
SELECT postalcode.locality, postalcode.locality, employee.name
FROM postalcode
RIGHT JOIN employee
ON postalcode.postalcode_id= employee.postalcode_id;
```

locality	locality	name
Porto	Porto	Manuel Santos
Porto	Porto	Paulo Fosneca
Porto	Porto	Carla Carolina
V.N.Gaia	V.N.Gaia	Isabel Antunes
Vila do Conde	Vila do Conde	Maria Costa
Vila do Conde	Vila do Conde	Ricardo Rocha
Porto	Porto	José Silva
Porto	Porto	Maria Andrade
Matosinhos	Matosinhos	Liliana Lousada
Matosinhos	Matosinhos	Diogo Dionísio
NULL	NULL	Teste

Returns all records of employees entity, even there are no matches on postalcode entity