Databases with *MYSQL*

7. JOINS *Querys*

Dante – Digital Area for Networking Teachers and Educators

Funded by the Erasmus+ Programme of the European Union

Learning Outcomes

After this lesson, the learner will be able to use all the following join query commands to retrieve data from database tables:

- Simple Joins
- Complex Joins
- Cartesian Joins
- Outer Joins
- Joints with composite keys

Funded by the Erasmus+ Programme of the European Union

Joins between tables

he join tables are part of the FROM clause.

The join is performed in the WHERE clause
Several operators can be used to join tables, such as =, <, >, <>, <=, >=, !=, BETWEEN, LIKE and NOT;

The most common operator is =.

Funded by the Erasmus+ Programme of the European Union

Relational Algebra - Relational Operators
Cartesian Product or Cartesian Join

The Cartesian product of two relations R1 and R2 is a new relation R3 in which every tuple of R1 is added to every tuple of R2.

The result of the Cartesian product is of very limited usefulness since the records are associated without any criteria.

Funded by the Erasmus+ Programme of the European Union

Relational Algebra – Relational Operators Cartesian Junction (Exemple)

```
SELECT EMP.ENAME, EMP.DEPTNO, DEPT.DEPTNO, DEPT.DNAME FROM EMP, DEPT;
```

ENAME	DEPTNO	DEPTNO	DNAME
SMITH	20	10	ACCOUNTING
SMITH	20	20	RESEARCH
SMITH	20	30	SALES
SMITH	20	40	OPERATIONS
ALLEN	30	10	ACCOUNTING
ALLEN	30	20	RESEARCH
ALLEN	30	30	SALES
ALLEN	30	40	OPERATIONS
WARD	30	10	ACCOUNTING
WARD	30	20	RESEARCH
WARD	30	30	SALES
WARD	30	40	OPERATIONS
JONES	20	10	ACCOUNTING
JONES	20	20	RESEARCH
JONES	20	30	SALES
JONES	20	40	OPERATIONS
MARTIN	30	10	ACCOUNTING
MARTIN	30	20	RESEARCH
MARTIN	30	30	SALES
MARTIN	30	40	OPERATIONS
BLAKE	30	10	ACCOUNTING
BLAKE	30	20	RESEARCH
BLAKE	30	30	SALES
BLAKE	30	40	OPERATIONS
CLARK	10	10	ACCOUNTING
CLARK	10	20	RESEARCH

ENAME	DEPTNO	DEPTNO	DNAME
CLARK	10	30	SALES
CLARK	10	40	OPERATIONS
SCOTT	20	10	ACCOUNTING
SCOTT	20	20	RESEARCH
SC		30	SALES
SCOTT	20	40	OPERATIONS
KING	10	10	ACCOUNTING
KING	10	20	RESEARCH
KING	10	30	SALES
KING	10	40	OPERATIONS
TURNER	30	10	ACCOUNTING
TURNER	30	20	RESEARCH
TURNER	30	30	SALES
TURNER	30	40	OPERATIONS
ADAMS	20	10	ACCOUNTING
ADAMS	20	20	RESEARCH
ADAMS	20	30	SALES
ADAMS	20	40	OPERATIONS
JAMES	30	10	ACCOUNTING
JAMES	30	20	RESEARCH
JAMES	30	30	SALES
JAMES	30	40	OPERATIONS
FORD	20	10	ACCOUNTING
FORD	20	20	RESEARCH
FORD	20	30	SALES
FORD	20	40	OPERATIONS

ENAME	DEPTNO	DEPTNO	DNAME
MILLER	10	10	ACCOUNTING
MILLER	10	20	RESEARCH
MILLER	10	30	SALES
MILLER	10	40	OPERATIONS

Each employee will show in every department

Funded by the Erasmus+ Programme of the European Union

Relational Algebra – Relational Operators Cartesian Junction (Exemple)

```
SELECT EMP.ENAME, EMP.DEPTNO, DEPT.DEPTNO, DEPT.DNAME FROM EMP, DEPT;
```

ENAME	DEPTNO	DEPTNO	DNAME
SMITH	20	10	ACCOUNTING
SMITH	20	20	RESEARCH
SMITH	20	30	SALES
SMITH	20	40	OPERATIONS
ALLEN	30	10	ACCOUNTING
ALLEN	30	20	RESEARCH
ALLEN	30	30	SALES
ALLEN	30	40	OPERATIONS
WARD	30	10	ACCOUNTING
WARD	30	20	RESEARCH
WARD	30	30	SALES
WARD	30	40	OPERATIONS
JONES	20	10	ACCOUNTING
JONES	20	20	RESEARCH
JONES	20	30	SALES
JONES	20	40	OPERATIONS
MARTIN	30	10	ACCOUNTING
MARTIN	30	20	RESEARCH
MARTIN	30	30	SALES
MARTIN	30	40	OPERATIONS
BLAKE	30	10	ACCOUNTING
BLAKE	30	20	RESEARCH
BLAKE	30	30	SALES
BLAKE	30	40	OPERATIONS
CLARK	10	10	ACCOUNTING
CLARK	10	20	RESEARCH

ENAME	DEPTNO	DEPTNO	DNAME
CLARK	10	30	SALES
CLARK	10	40	OPERATIONS
SCOTT	20	10	ACCOUNTING
SCOTT	20	20	RESEARCH
SCOTT	20	30	SALES
SCOTT	20	40	OPERATIONS
KING	10	10	ACCOUNTING
KING	10	20	RESEARCH
KING	10	30	SALES
KING	10	40	OPERATIONS
TURNER	30	10	ACCOUNTING
TURNER	30	20	RESEARCH
TURNER	30	30	SALES
TURNER	30	40	OPERATIONS
ADAMS	20	10	ACCOUNTING
ADAMS	20	20	RESEARCH
ADAMS	20	30	SALES
ADAMS	20	40	OPERATIONS
JAMES	30	10	ACCOUNTING
JAMES	30	20	RESEARCH
JAMES	30	30	SALES
JAMES	30	40	OPERATIONS
FORD	20	10	ACCOUNTING
FORD	20	20	RESEARCH
FORD	20	30	SALES
FORD	20	40	OPERATIONS

ENAME	DEPTNO	DEPTNO	DNAME
MILLER	10	10	ACCOUNTING
MILLER	10	20	RESEARCH
MILLER	10	30	SALES
MILLER	10	40	OPERATIONS

Each employee will should only show in every department that is reistered in the EMPR table

EMP.DEPTNO should have the same value as DEPT.DEPTNO

Funded by the Erasmus+ Programme of the European Union

Simple JOIN
Equi-JOIN

The most common simple join is the equi-join, also called an inner join.

It is the joining of two tables through the = operator.

Equi-join joins two tables using a common column (primary key and foreign key).

Funded by the Erasmus+ Programme of the European Union

Simple JOIN
Equi-JOIN

Sintax:

```
SELECT TABLE1.COLUMN1, TABLE2.COLUMN2...FROM TABLE1,
TABLE2 [, TABLE3 ] WHERE
TABLE1. NAME_COLUMN = TABLE2. NAME_COLUMN
[AND TABLE1. NAME_COLUMN = TABLE3. NAME_COLUMN]
```

Funded by the Erasmus+ Programme of the European Union

Simple JOIN
Equi-JOIN

```
SELECT EMP.ENAME, EMP.DEPTNO, DEPT.DEPTNO, DEPT.DNAME FROM EMP,
DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO;
```

ENAME	DEPTNO	DEPTNO	DNAME
CLARK	10	10	ACCOUNTING
KING	10	10	ACCOUNTING
MILLER	10	10	ACCOUNTING
SMITH	20	20	RESEARCH
JONES	20	20	RESEARCH
SCOTT	20	20	RESEARCH
ADAMS	20	20	RESEARCH
FORD	20	20	RESEARCH
ALLEN	30	30	SALES
WARD	30	30	SALES
MARTIN	30	30	SALES
BLAKE	30	30	SALES
TURNER	30	30	SALES
JAMES	30	30	SALES

Each employee show in the department that is reistered in the EMPR table

EMP.DEPTNO has the same value as DEPT.DEPTNO

Funded by the Erasmus+ Programme of the European Union

Aliases for table names

To simplify the coding of querys Alias for table names may be used.

```
SELECT E.ENAME, E.DEPTNO, D.DEPTNO, D.DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO=D.DEPTNO;
```

For the EMP table, the alias **E** was used.

For the DEPT table, the alias **D** was used.

Funded by the Erasmus+ Programme of the European Union

Simple Equi-JOIN

```
SELECT E.*, D.DNAME FROM EMP E, DEPT D
WHERE E.DEPTNO=D.DEPTNO;
```

(*) represents all columns of the table

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DNAME
7782	CLARK	MANAGER	7839	2023-06-09	2450.00	NULL	10	ACCOUNTING
7839	KING	PRESIDENT	NULL	2023-11-17	5000.00	NULL	10	ACCOUNTING
7934	MILLER	CLERK	7782	2024-01-23	1300.00	NULL	10	ACCOUNTING
7369	SMITH	CLERK	7902	2022-12-17	800.00	NULL	20	RESEARCH
7566	JONES	MANAGER	7839	2023-04-02	2975.00	NULL	20	RESEARCH
7788	SCOTT	ANALYST	7566	2025-04-19	3000.00	NULL	20	RESEARCH
7876	ADAMS	CLERK	7788	2025-05-23	1100.00	NULL	20	RESEARCH
7902	FORD	ANALYST	7566	2023-12-03	3000.00	NULL	20	RESEARCH
7499	ALLEN	SALESMAN	7698	2023-02-20	1600.00	300.00	30	SALES
7521	WARD	SALESMAN	7698	2023-02-22	1250.00	500.00	30	SALES
7654	MARTIN	SALESMAN	7698	2023-09-28	1250.00	1400.00	30	SALES
7698	BLAKE	MANAGER	7839	2023-05-01	2850.00	NULL	30	SALES
7844	TURNER	SALESMAN	7698	2023-09-08	1500.00	0.00	30	SALES
7900	JAMES	CLERK	7698	2023-12-03	950.00	NULL	30	SALES

Funded by the Erasmus+ Programme of the European Union

Complex JOIN

In addition to the join, you can add more conditions in the WHERE clause.

```
SELECT E.*, D.DNAME FROM EMP E, DEPT D WHERE
E.DEPTNO=D.DEPTNO AND JOB='MANAGER';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DNAME
7566	JONES	MANAGER	7839	2023-04-02	2975.00	NULL	20	RESEARCH
7698	BLAKE	MANAGER	7839	2023-05-01	2850.00	NULL	30	SALES
7782	CLARK	MANAGER	7839	2023-06-09	2450.00	NULL	10	ACCOUNTING

Funded by the Erasmus+ Programme of the European Union

NATURAL JOIN

NATURAL JOIN is very similar to EQUI-JOIN;

It differs from EQUI-JOIN in that it does not duplicate the Join columns.

The join columns must have the same name and type in both tables.

Funded by the Erasmus+ Programme of the European Union

Natural Join

```
SELECT EMPNO, ENAME, DNAME FROM EMP NATURAL JOIN DEPT;
```

	EMPNO	ENAME	DNAME
▶	7782	CLARK	ACCOUNTING
	7839	KING	ACCOUNTING
	7934	MILLER	ACCOUNTING
	7369	SMITH	RESEARCH
	7566	JONES	RESEARCH
	7788	SCOTT	RESEARCH
	7876	ADAMS	RESEARCH
	7902	FORD	RESEARCH
	7499	ALLEN	SALES
	7521	WARD	SALES
	7654	MARTIN	SALES
	7698	BLAKE	SALES
	7844	TURNER	SALES
	7900	JAMES	SALES

Funded by the Erasmus+ Programme of the European Union

Outer JOIN

An outer join is used to return **all rows that exist in one of the tables, even if the corresponding rows do not exist in the other joining table.**

The outer join can be divided into left outer join, right outer join and full outer join.

Funded by the Erasmus+ Programme of the European Union

RIGHT Outer JOIN

```
SELECT EMPNO, E.DEPTNO, D.DEPTNO, DNAME FROM EMP E RIGHT OUTER JOIN  DEPT
D ON  E.DEPTNO=D.DEPTNO;
```

Equivalent to:

```
SELECT EMPNO, DEPTNO, DNAME FROM EMP RIGHT JOIN  DEPT USING (DEPTNO);
```

	EMPNO	DEPTNO	DEPTNO	DNAME
▶	7782	10	10	ACCOUNTING
	7839	10	10	ACCOUNTING
	7934	10	10	ACCOUNTING
	7369	20	20	RESEARCH
	7566	20	20	RESEARCH
	7788	20	20	RESEARCH
	7876	20	20	RESEARCH
	7902	20	20	RESEARCH
	7499	30	30	SALES
	7521	30	30	SALES
	7654	30	30	SALES
	7698	30	30	SALES
	7844	30	30	SALES
	7900	30	30	SALES
	NULL	NULL	40	OPERATIONS

To show all Departments regardless if they have employees (DEPT is in the right in the query.

Although the OPERATIONS department does not have any emplyees, it will show using an OUTER JOIN

Funded by the Erasmus+ Programme of the European Union

LEFT Outer Junction

```
SELECT EMPNO, E.DEPTNO, D.DEPTNO, DNAME FROM DEPT D LEFT OUTER JOIN EMP E
ON  E.DEPTNO=D.DEPTNO;
```

Equivalent to:

```
SELECT EMPNO, DEPTNO, DNAME FROM DEPT LEFT JOIN  EMP USING (DEPTNO);
```

	EMPNO	DEPTNO	DEPTNO	DNAME
▶	7782	10	10	ACCOUNTING
	7839	10	10	ACCOUNTING
	7934	10	10	ACCOUNTING
	7369	20	20	RESEARCH
	7566	20	20	RESEARCH
	7788	20	20	RESEARCH
	7876	20	20	RESEARCH
	7902	20	20	RESEARCH
	7499	30	30	SALES
	7521	30	30	SALES
	7654	30	30	SALES
	7698	30	30	SALES
	7844	30	30	SALES
	7900	30	30	SALES
	NULL	NULL	40	OPERATIONS

To show all Departments regardless if they have employees (DEPT is in the left in the query).

Funded by the Erasmus+ Programme of the European Union

Non Equi-Join

Non-Equi-join is the joining of two or more tables across specific columns using an operator other than =.

Funded by the Erasmus+ Programme of the European Union

Non Equi-Join

```
SELECT * FROM SALGRADE;
```

GRADE	LOSAL	HISAL
1	700.00	1200.00
2	1201.00	1400.00
3	1401.00	2000.00
4	2001.00	3000.00
5	3001.00	9999.00

```
SELECT * FROM EMP;
```

empno	ename	sal
7369	SMITH	800.00
7499	ALLEN	1600.00
7521	WARD	1250.00
7566	JONES	2975.00
7654	MARTIN	1250.00
7698	BLAKE	2850.00
7782	CLARK	2450.00
7788	SCOTT	3000.00
7839	KING	5000.00
7844	TURNER	1500.00
7876	ADAMS	1100.00
7900	JAMES	950.00
7902	FORD	3000.00
7934	MILLER	1300.00

The SALGRADE table contains information about the grade or level of the salary of the employee.

If the employee earns between 700 and 1200, he is grade 1; If the employee earns between 1201 and 1400, he is grade 2; ...

For exemple, employee SMITH and JAMES are GRADE 1. SMITH earns 800, which is between 700 and 1200 that are LOSAL and HISAL in the SALGRADE table.

Funded by the Erasmus+ Programme of the European Union

Non Equi-Join

```
SELECT EMPNO, ENAME, SAL, GRADE FROM EMP, SALGRADE
WHERE SAL BETWEEN LOSAL AND HISAL;
```

	EMPNO	ENAME	SAL	GRADE
▶	7369	SMITH	800.00	1
	7499	ALLEN	1600.00	3
	7521	WARD	1250.00	2
	7566	JONES	2975.00	4
	7654	MARTIN	1250.00	2
	7698	BLAKE	2850.00	4
	7782	CLARK	2450.00	4
	7788	SCOTT	3000.00	4
	7839	KING	5000.00	5
	7844	TURNER	1500.00	3
	7876	ADAMS	1100.00	1
	7900	JAMES	950.00	1
	7902	FORD	3000.00	4
	7934	MILLER	1300.00	2

Using the Non Equi-Join SAL BETWEEN LOSAL and HISAL, the GRADE of each emplyee can be seen.

Funded by the Erasmus+ Programme of the European Union

Pure Join

```
SELECT EMPNO, ENAME, MGR FROM EMP;
```

EMPNO	ENAME	MGR
7369	SMITH	7902
7499	ALLEN	7698
7521	WARD	7698
7566	JONES	7839
7654	MARTIN	7698
7698	BLAKE	7839
7782	CLARK	7839
7788	SCOTT	7566
7839	KING	NULL
7844	TURNER	7698
7876	ADAMS	7788
7900	JAMES	7698
7902	FORD	7566
7934	MILLER	7782
NULL	NULL	NULL

Each is employee has a manager (MGR) who is also an employee.

For exemple employee 7369, who is SMITH, has a manager 7902, who is FORD.

Funded by the Erasmus+ Programme of the European Union

Pure Join

A join from a table to that same table, as if that table were two tables.

Table **EMP E** will be the table of all employes

EMPNO	ENAME	MGR
7369	SMITH	7902
7499	ALLEN	7698
7521	WARD	7698
7566	JONES	7839
7654	MARTIN	7698
7698	BLAKE	7839
7782	CLARK	7839
7788	SCOTT	7566
7839	KING	NULL
7844	TURNER	7698
7876	ADAMS	7788
7900	JAMES	7698
7902	FORD	7566
7934	MILLER	7782
NULL	NULL	NULL

Table **EMP M** will be the table of all employes

EMPNO	ENAME	MGR
7369	SMITH	7902
7499	ALLEN	7698
7521	WARD	7698
7566	JONES	7839
7654	MARTIN	7698
7698	BLAKE	7839
7782	CLARK	7839
7788	SCOTT	7566
7839	KING	NULL
7844	TURNER	7698
7876	ADAMS	7788
7900	JAMES	7698
7902	FORD	7566
7934	MILLER	7782
NULL	NULL	NULL

We will use the Equi-JOIN

E.MGR=M.EMPNO

Funded by the Erasmus+ Programme of the European Union

Pure Join

```
SELECT E.EMPNO "Employee Number", E.ENAME "Employee Name", E.MGR
"Employee Manager", M.ENAME  "Manager Name"
FROM EMP E, EMP M
WHERE E.MGR=M.EMPNO;
```

Employee Number	Employee Name	Employee Manager	Manager Name
7369	SMITH	7902	FORD
7499	ALLEN	7698	BLAKE
7521	WARD	7698	BLAKE
7566	JONES	7839	KING
7654	MARTIN	7698	BLAKE
7698	BLAKE	7839	KING
7782	CLARK	7839	KING
7788	SCOTT	7566	JONES
7844	TURNER	7698	BLAKE
7876	ADAMS	7788	SCOTT
7900	JAMES	7698	BLAKE
7902	FORD	7566	JONES
7934	MILLER	7782	CLARK

Funded by the Erasmus+ Programme of the European Union

Joins with composite keys

A primary key can consist of more than one column, just like a foreign key.

A composite primary key must always be referenced by a composite foreign key.

Composite key joins have to compare all key components.