

## *Structured Query Language (SQL)*

**Structured Query Language (SQL):** is a database computer language designed for managing in relational database management systems (RDBMS), and originally based upon Relational Algebra.

SQL designed to organize and simplify the process of getting information out of a database in a usable form, and also used to reorganize data within databases. It is a set of statements to manage databases, tables, and data.

Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server and Microsoft Access.

SQL falls into two classes

1. Data Definition Language (DDL) - SQL for creating, altering and dropping tables
2. Data Manipulation Language (DML) - SQL for retrieving and storing data.

Data Definition Language (DDL) : a database schema is specified by a set of definitions expressed by the a language called (DDL).

It is the subset of SQL used for defining and examining the structure of a database. It used to define databases and their components.

Data Manipulation Language (DML): The data in the database are manipulated by the (DML). With (DML) we can

- Retrieve information from the database
- Inserting new information into the database
- Delete information from the database
- Updating the data in the database

### **Table Joining**

A JOIN is a means for combining fields from two tables by using values common to each.

Table Joining is a formal specification of which column(s) in a row in one table should be matched to a column(s) in a row in another table.

To combine tables, i.e. to perform a join, we have to select data from the tables, and relate the tables to each other with conditions on some attributes (often keys attributes).

Joining tables is used when retrieving information from the database is required .

*In SQL the word ( SELECT ) is used to display information from the database followed by the name of the tables.*

to explain the types of joins in the database, the following three tables are used as an example:

teachers table

id	name
1	Volker
2	Elke

(2 rows)

Projects table

id	name	duration	teacher
1	compiler	180	1
2	xpaint	120	1
3	game	250	2
4	Perl	80	4

(4 rows)

Assign table

project	stud	percentage
1	2	10
1	4	60
1	1	30
2	1	50
2	4	50
3	2	70
3	4	30

(7 rows)

## Cross Join

Each row from table one is arbitrarily combine with each row from table two.

This is known as a Cartesian product. In practical terms a cross join is a join without a join

Condition

Each row in teachers is arbitrarily combined with each row in projects.:

Example: Cross join of tables *teachers* and *project* will be

```
> SELECT * FROM teachers, projects;
```

id	name	id	name	duration	teacher
1	Volker	1	compiler	180	1
1	Volker	2	xpaint	120	1
1	Volker	3	game	250	2
1	Volker	4	Perl	80	4
2	Elke	1	compiler	180	1
2	Elke	2	xpaint	120	1
2	Elke	3	game	250	2
2	Elke	4	Perl	80	4

(8 rows)

### Inner Join :

Inner joins are the most common type of joins, they combine two or more tables using where clause as main condition with equality "=" sign or inequality "!=" .

A field from the first table is compared to another field from the second table and if they are equal ( or not equal ) the two records from the two tables are combined.

Example : inner join of tables *teachers* and *project* if the condition is `teachers.id = projects.id` will be

`SELECT * FROM teachers, projects where teachers.id = projects.id;`

id	name	id	name	duration	teacher
1	Volker	1	compiler	180	1
2	Elke	2	xpaint	120	1

Example : inner join of tables *teachers* and *project* if the condition is *teachers.id != projects.id* will be

`SELECT * FROM teachers, projects where teachers.id != projects.id;`

id	name	id	name	duration	teacher
1	Volker	2	xpaint	180	1
1	Volker	3	game	180	1
1	Volker	4	Perl	180	1
2	Elke	1	compiler	120	1
2	Elke	3	game	120	1
2	Elke	4	Perl	120	1

There is another format for inner join

`select T1.F1, F2,F3,...,FN from T1 inner join T2 on (T1.F1 = T2.F1)`

`inner join T3 on (T2.F1=T3.F1) inner join .....`

### **LEFT OUTER JOIN:**

this type of join will first select all shared records from those tables, and then will bring all records from left table which are not exist in the second table.

### **RIGHT OUTER JOIN**

this type of join will first select all shared records from those tables, and then will bring all

records from right table which are not exist in the first table

### **FULL OUTER JOIN**

this type of join will first select all shared records from those tables, and then will bring all

records exist in first table which are not exist in the second table and then bring all records exist in the second table which are not exist in first table.

select teachers.id , teachers.name , projects.id , projects.name from teachers full OUTER join projects on (teachers.id = projects.id);

```
id | name | id | name
---+---+---+---
 1 | Volker | 1 | compiler
 2 | Elke | 2 | xpaint
      |      | 4 | Perl
      |      | 3 | games
```