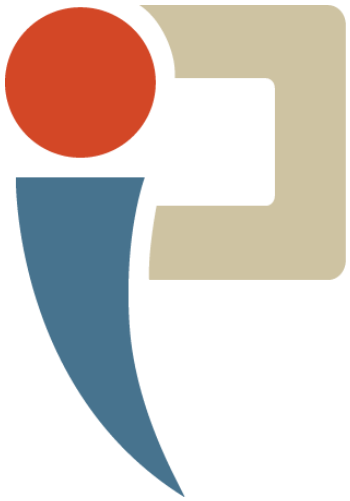
POLUDO INSTITUTE OF TECHNOLOGY AND MEDIA



SQL TRAINING

Module 1

Introduction to SQL and RDBMS

Jesse Silva – jessetsilva@gmail.com

2018

Module 1: Introduction to SQL and RDBMS

The student will be introduced to how client/server architecture works, and examine the various database and business tasks that can be performed by using the components of SQL Server. The student will also be introduced to SQL Server database concepts such as relational databases, normalization, and database objects. In addition, the student will learn how to use T-SQL to query databases and generate reports.

Lessons

- Introduction to SQL and its history
- SQL Process to execute tasks
- DDL – Data definition Language
- DML – Data manipulation Language
- DCL – Data Control Language
- DQL – Data Query Language
- What is RDBMS
- What is table, field, record row, column and Constraint
- SQL Data Types and Operators

Introduction to SQL and its history

Structured Query Language is a computer language for storing, manipulating and retrieving data stored in a relational database. SQL is the standard language for Relation Database System, which means that all relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix, postgres and SQL Server uses SQL as standard database language. Dr. E. F. "Ted" of IBM is known as the father of relational databases because he was the first one to describe a relational model for this. That's the earlies events regarding the consolidation of the SQL:

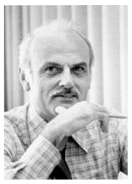
- 1970

  Dr. E. F. "Ted" of IBM is known as the father of relational databases. He described a relational model for databases.
- 1974

  Structured Query Language appeared.
- 1978

  IBM worked to develop Codd's ideas and released a product named System/R.
- 1986

  IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software (later becomes Oracle).



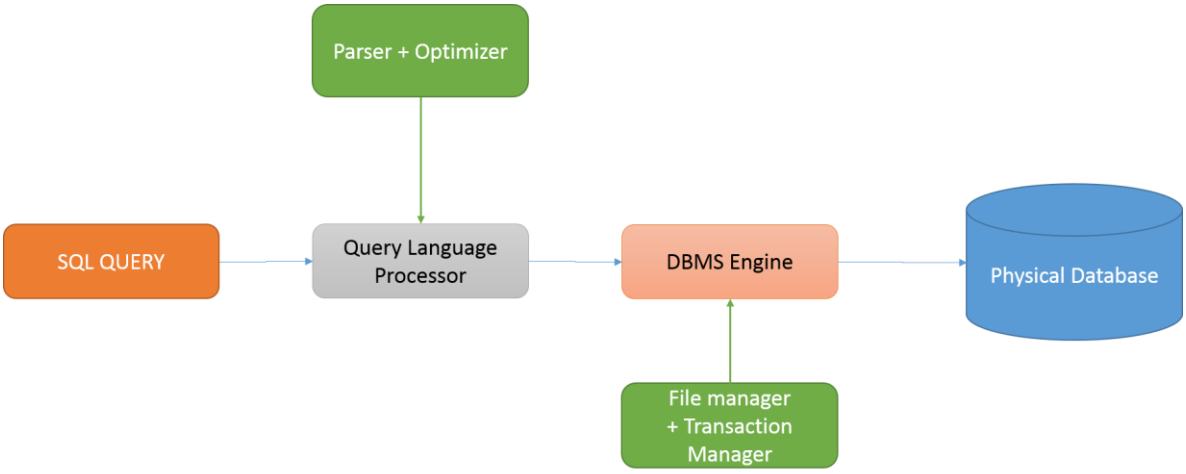
Edgar Frank Codd
1923 – 2003

Among a lot of great features the SQL has, we can spotlight some of the most important ones that is allowing the users to:

- Access data in relational database management systems.
- Describe the data.
- Define the data in database and manipulate that data.
- Embed within other languages using SQL modules, libraries & pre-compilers.
- Create and drop databases and tables.
- Create views, stored procedures and functions in a database.
- Set permissions on tables, procedures and views
- …

SQL Process to execute tasks

When executing an SQL command for any RDBMS, is due to the system the task to find the best way to carry out your request and to the SQL engine how to interpret the task. The architecture can be described as:



SQL Commands

There is a set of standard SQL Commands to interact with relational databases and they can be classified by their nature:

- DDL – Data Definition Language
    - Create
      Create a new database, view, table or other database object
    - Alter
      Modify an existing database object
    - Drop
      Delete an entire table, view or other database object
- DML – Data modeling Language
    - Insert
      Insert new records on the table
    - Update
      Update existing records on the table
    - Delete
      Delete objects (records) from a table
- DQL – Data Query Language
    - Select
      Retrieve information from tables based on a set of conditions

- DCL – Data Control Language

    - Grant

        Gives privileges for a user or a group
    - Revoke

        Removes privileges of an user or a group

- CRUD

Despite all the definitions, the four major Functions implemented in a database application can be defined as:

- Create (insert command) — NEW RECORDS
- Read or Retrieve (select command) — EXISTING RECORDS
- Update (update command) — EXISTING RECORDS
- Delete (delete command) — EXISTING RECORDS

RDBMS

Relational Database Management System is the base for the SQL and for all modern database systems. It is a DBMS (database management system) based on the relational model introduced by "*Ted*" Codd. Each RDBMS is made of database table rows that consists of one or more database table fields. RDBMS store the data into collection of tables, which might be related by common fields (table columns). RDBMS also provide relational operators to manipulate the data stored into the database tables. Several companies owns his own RDBMS Systems, like MS SQL Server, IBM DB2, Oracle, MySQL and MS Access. Here are some definitions

- Table

  Where all the data are stored in a database. It can be defined as a collection of well formatted related information consisting in rows and columns to organize it.

id	full_name	address	gender	birth_date	course	registration_date
11177898022	Bill Gates	879 4th Street	m	1965-11-22	Windows concepts	2014-03-20
09878965621	Edgar Frank Codd	Isle of Portland, England	m	1923-08-19	SQL advanced	2015-06-19
05255986933	Jesse Teixeira Silva	1327 maple Street - Vancouver	m	1983-06-08	SQL	2015-07-08
12345678901	Marie Curie	Warsaw, Poland	f	1867-11-07	NULL	NULL
98997867833	Martin Pert	234 Haro Street	m	1965-12-11	Java 1	2015-03-08

- Field

  Every table is broken up into smaller entities called fields. These fields are responsible to maintain all the information in the table. Usually we refer these fields as "table headers"

id	full_name	address	gender	birth_date	course	registration_date
11177898022	Bill Gates	879 4th Street	m	1965-11-22	Windows concepts	2014-03-20
09878965621	Edgar Frank Codd	Isle of Portland, England	m	1923-08-19	SQL advanced	2015-06-19
05255986933	Jesse Teixeira Silva	1327 maple Street - Vancouver	m	1983-06-08	SQL	2015-07-08
12345678901	Marie Curie	Warsaw, Poland	f	1867-11-07	NULL	NULL
98997867833	Martin Pert	234 Haro Street	m	1965-12-11	Java 1	2015-03-08

- Column

  It is the vertical entity in a table that contains all information associated with a specific field.

id	full_name	address	gender	birth_date	course	registration_date
11177898022	Bill Gates	879 4th Street	m	1965-11-22	Windows concepts	2014-03-20
09878965621	Edgar Frank Codd	Isle of Portland, England	m	1923-08-19	SQL advanced	2015-06-19
05255986933	Jesse Teixeira Silva	1327 maple Street - Vancouver	m	1983-06-08	SQL	2015-07-08
12345678901	Marie Curie	Warsaw, Poland	f	1867-11-07	NULL	NULL
98997867833	Martin Pert	234 Haro Street	m	1965-12-11	Java 1	2015-03-08

- Row

  Also known as record, represents each set of data in a table. It's important to keep in mind that sometimes we call "row" only what we get from a specific select as it represents the information we need in that exactly moment.

id	full_name	address	gender	birth_date	course	registration_date
11177898022	Bill Gates	879 4th Street	m	1965-11-22	Windows concepts	2014-03-20
09878965621	Edgar Frank Codd	Isle of Portland, England	m	1923-08-19	SQL advanced	2015-06-19
05255986933	Jesse Teixeira Silva	1327 maple Street - Vancouver	m	1983-06-08	SQL	2015-07-08
12345678901	Marie Curie	Warsaw, Poland	f	1867-11-07	NULL	NULL
98997867833	Martin Pert	234 Haro Street	m	1965-12-11	Java 1	2015-03-08

  - Null values

    It is the value in a field that appears to be blank, which means a field with a NULL value is a field with no value. Be careful, NULL value is different from zero or a field with only spaces. A field that is null was for some reason left in BLANK during an insert or update.

id	full_name	address	gender	birth_date	course	registration_date
11177898022	Bill Gates	879 4th Street	m	1965-11-22	Windows concepts	2014-03-20
09878965621	Edgar Frank Codd	Isle of Portland, England	m	1923-08-19	SQL advanced	2015-06-19
05255986933	Jesse Teixeira Silva	1327 maple Street - Vancouver	m	1983-06-08	SQL	2015-07-08
12345678901	Marie Curie	Warsaw, Poland	f	1867-11-07	NULL	NULL
98997867833	Martin Pert	234 Haro Street	m	1965-12-11	Java 1	2015-03-08

CONSTRAINTS

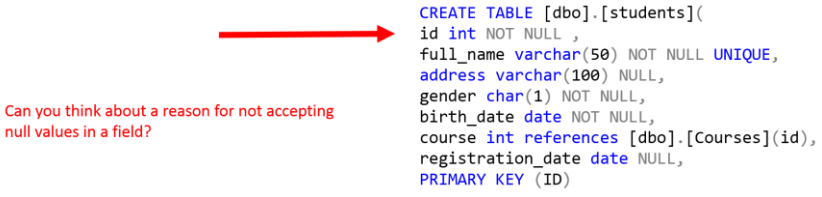
Constraints are the rules enforced on table data columns used to limit (restrict) the type of data that can go into a table and ensures the accuracy and reliability of the data in the database. We can divided them into three main categories:

- **Implicit constraints**, that are inherent in the data model.
- **Explicit level Constraints**, that can be directly expressed in schemas of the data model, typically by specifying them in the DDL
- **Application-based Constraints,** that cannot be directly expressed in the schemas of the data model and must be expressed and enforced by the application programs

The constraints can be applied only to one column (columns level constraints) or to the whole table (table level constraints). These are some constraints examples:

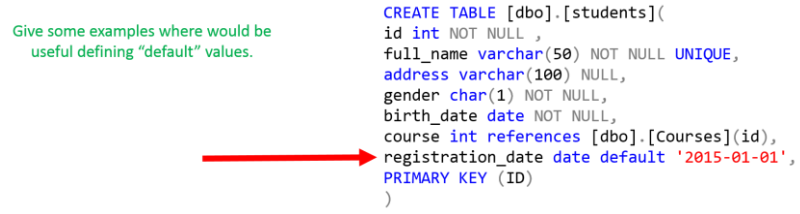
- NOT NULL

  This enforces that a column can't accept NULL values, what means that the field must always contains a valid value. Keep in mind that null values is different of blanks spaces or values like 0 (zero)



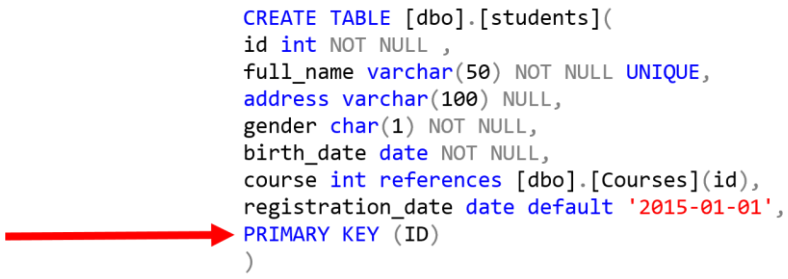
- DEFAULT

  It is used when you want to insert a default value into a columns when the user do not specify it. We must remember that not specifying a value is different from specifying it as null.



- PRIMARY Key

  This constraint uniquely identifies each record in a database table, what means that it must contains unique values. You can inform the value every time you insert a record or just let the system handle that, as example, by incrementing numbers. You can't have more than one primary key for table, but you can use, as example, two fields to generate a primary key.



- FOREIGN Key

  Its' used to point to another table primary key (used when you have table relationships). Using that constraint we can avoid some actions that can corrupt our data. Imagine, for example, a table orders that contains all the client purchases. That table must have something that identifies the client, and that's the reason we use foreign key, because doing that we can point that record for another table, in this case clients, that contains all the

personal information. All the foreign key are validate, what means that you can't create a relation between a Order and a client that do not exist.

```
CREATE TABLE [dbo].[students](
id int NOT NULL ,
full_name varchar(50) NOT NULL UNIQUE,
address varchar(100) NULL,
gender char(1) NOT NULL,
birth_date date NOT NULL,
course int references [dbo].[Courses](id),
registration_date date default '2015-01-01',
PRIMARY KEY (ID)
)
```

id	description
1	Java 1
2	Java 2
3	Java 3
7	PHP
5	Project Management
6	Quality assurance
4	SQL Concepts

- UNIQUE

Ensures that all values in a column are different (not replicate). It works in a similar way of the primary key constraint, providing the guarantee that duplicates records will not exist, but different from the primary key, you can have as much UNIQUE constraints as you want.

```
CREATE TABLE [dbo].[students](
id int NOT NULL ,
full_name varchar(50) NOT NULL UNIQUE,
address varchar(100) NULL,
gender char(1) NOT NULL,
birth_date date NOT NULL,
course int references [dbo].[Courses](id),
registration_date date default '2015-01-01',
PRIMARY KEY (ID)
)
```

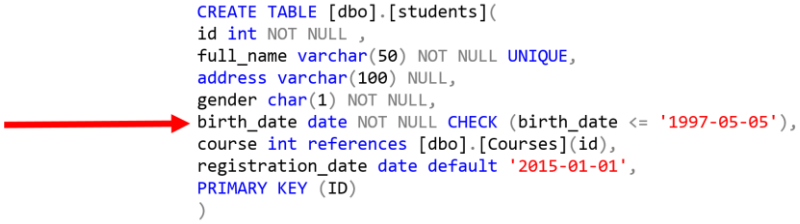
- INDEX

The INDEX is used to create and retrieve data from the database very quickly and can be created by using single or group of columns in a table. When the index is created, it is assigned a ROWID for each row before it sorts out the data.

```
CREATE INDEX idIndex ON students (full_name)
```

- CHECK

We use this constraint to limit the values inserted on a column. We can, for example, limit a numerical column to accept only values between 5 and 10. We can even do validations based on other records or field values.



```
CREATE TABLE [dbo].[students](
id int NOT NULL ,
full_name varchar(50) NOT NULL UNIQUE,
address varchar(100) NULL,
gender char(1) NOT NULL,
birth_date date NOT NULL CHECK (birth_date <= '1997-05-05'),
course int references [dbo].[Courses](id),
registration_date date default '2015-01-01',
PRIMARY KEY (ID)
)
```

SQL Data Types

In SQL Server, each column, local variable, expression, and parameter has an attribute that specifies the type of data that the object can hold, called data type. Data types in SQL Server are organized into the following categories (source: https://msdn.microsoft.com):

- Exact numerics

DATA TYPE	FROM	TO
bigint	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
int	-2,147,483,648	2,147,483,647
smallint	-32,768	32,767
tinyint	0	255
bit	0	1
decimal	-10^38 +1	10^38 -1
numeric	-10^38 +1	10^38 -1
money	-922,337,203,685,477.5808	+922,337,203,685,477.5807
smallmoney	-214,748.3648	+214,748.3647

- Approximate numerics

DATA TYPE	FROM	TO
float	**-1.79E + 308**	**1.79E + 308**
real	**-3.40E + 38**	**3.40E + 38**

- Date and time

DATA TYPE	FROM	TO
datetime	Jan 1, 1753	Dec 31, 9999
smalldatetime	Jan 1, 1900	Jun 6, 2079

  - We also have types to store only the date or time, separated

DATA TYPE	DESCRIPTION
date	Stores a date without the time
time	Stores a time without the date part.

- Character String

DATA TYPE	DESCRIPTION
char	Max of 8,000 characters.
varchar	Max of 8,000 characters.
varchar(max)	Maximum length of 231characters
text	Max of 2,147,483,647 characters.

- Unicode Character String

DATA TYPE	DESCRIPTION
nchar	Max of 4,000 characters
nvarchar	Max of 4,000 characters.
nvarchar(max)	Max of 231characters (SQL Server 2005 only)
Ntext	Max of 1,073,741,823 characters. (Variable length Unicode)

- Binary

DATA TYPE	DESCRIPTION
Binary	Max of 8,000 bytes
Varbinary	Max of 8,000 bytes.
varbinary(max)	Max of 231 bytes
Image	Max of 2,147,483,647 bytes.

- Miscellaneous types

DATA TYPE	DESCRIPTION
sql_variant	Stores values of various SQL Server-supported data types, except text, ntext, and timestamp.
timestamp	Stores a database-wide unique number that gets updated every time a row gets updated
uniqueidentifier	Stores a globally unique identifier
xml	For SQL server 2005. Stores XML data. You can store xml instances in a column or a variable
cursor	References to a cursor object (we see it on module 8)
table	Stores a result set for later processing

SQL OPERATORS

In the module 2 of this course we will learn how to select records in our database tables. This selections relies on criterions that must contains the WHERE clause (don't worry, we will see it in details there). The Where clause uses reserved words called operators, which alone or combined can be used to specify conditions in an SQL statement and act as conjunctions for multiple conditions. The operators can be divide in four major groups:

- Arithmetic Operators
  - +
    - Sum the given values
  - –
    - Subtracts the given values
  - *
    - Multiplies the given values
  - /
    - Divides the left operand by the right hand operand.
  - %
    - Divides the left operand by the right hand operand and gives us the remainder.
- Comparison operators
  - =
    - Returns true if the given values are equals. Otherwise returns false.

- != or <>
  - Returns true if the given values are different. Otherwise returns false

- >
  - Returns true if the left operand is greater than the right operand.

- <
  - Returns true if the left operand is less than the right operand.

- >=
  - Returns true if the left operand is greater than or equal the right operand.

- <=
  - Returns true if the left operand is less than or equals the right operand.

- !<
  - Returns true if the left operand is not less than the right operand.

- !>
  - Returns true if the left operand is not greater than the right operand.

- Logical Operators
    - ALL
        - Compares a given value against all the values in a value set.
    - AND
        - Allows the existence of multiple conditions on a *where* clause
    - ANY
        - Compares a given value to any applicable value in the list according to the condition.
    - BETWEEN
        - Given the minimum and maximum values, returns all the values between the ranges.
    - EXISTS
        - Verify if a row in a specific table exists given a criteria.
    - IN
        - Used to compare a value to a list of literal values that have been specified. Returns true if the list contains the value.
    - LIKE
        - Used to compare a value to similar values using wildcard operators. We will see more about it on module 2.

- NOT
  - Used to deny the meaning of the logical operator in which it is used. For example NOT in will return true only if the value does not exist in the given list.

- OR
  - Similar with the AND, it is used to combine multiple conditions in a where clause.

- IS NULL
  - Returns true if a given value is null. Remember that null values are different from blanks space or 0 (zero).

- UNIQUE
  - Searches every row of a specified table for uniqueness (no duplicates).

PRACTICING

1. Our course is based on Relational databases, but do we know the differences between it and a transactional database? Make a Research based on internet or books and highlight the points you think is important to discuss with you colleagues next class!

REFERENCES

- **Books**
  - Fundamentals of database systems – 6th edition
  - Head First SQL - Lynn Beighley
- **Internet**
  - W3Schools @ http://www.w3schools.com/sql
  - Structured Query Language @ http://home.hit.no/~hansha/documents/database/documents/Structured%20Query%20Language.pdf
  - SQL Server tutorial @ Lynda.com http://www.lynda.com/SQL-Server-training-tutorial
  - Tutorial: SQL Server Management Studio @ https://technet.microsoft.com/en-us/library/bb934498(v=sql.110).aspx
  - SQL Server tutorials @ https://msdn.microsoft.com/en-us/library/ms167593(v=sql.105).aspx