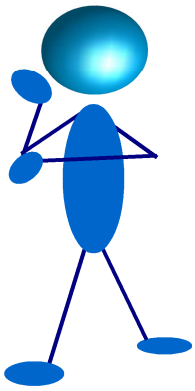
Fundamentos de SQL

¿SQL? ¿DDL? ¿DML? ¿TCL? ¿DCL?



Fundamentos de SQL

Definiciones

- SQL: Structured Query Language
  - Lenguaje de Consultas
- DDL: Data Definition Language
  - Definición de Objetos de la Base de Datos
- DML: Data Manipulation Language
  - Altas, Bajas, Cambios
- TCL: Transaction Control Language
  - Manejo de Transacciones
- DCL: Data Control Language
  - Permisos y Accesos

SQL - Structured Query Language (1/6)

- Select - From

```
SELECT ColumnName1, ColumnName2, …
FROM Table1 [As Alias]
```

- Where

```
Select ProductId, Name
From Production.Product
Where ProductId=1
```

- Operadores

Operator	Description
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	If you know the exact value you want to return for at least one of the columns

Select...

```
SELECT
  Campo1
  ,CASE
    WHEN Campo2<>'DET' THEN 'NO DETALLE'
    WHEN Campo2='DET' THEN 'DETALLE'
  END AS Campo2_Calculado
  ,Campo3+Campo4 As [Campo3-4]
  ,Funcion(Campo5) As Campo5Calculado
  ,T.Campo6 C6
FROM Tabla T
```


Where...

```
SELECT
  Campo1
FROM Tabla T
WHERE
  Campo1 Is Null
  ,OR ISNull(Campo2,0) = 0
  ,OR CAST(Campo3, varchar) = '0'
  ,OR LEFT(
       CONVERT(varchar, Campo4, 108), 4
     ) = '11/09'
```

SQL - Structured Query Language (2/6)

- Like
  - Comparación en base a patrones de similitud.

```
Select ProductId, Name
From Production.Product
Where Name Like '%Chain%'
```

- Top N
  - Muestra solamente los primeros N registros.

```
Select Top 5 ProductId, Name From
 Production.Product
```

- Distinct
  - Muestra los elementos únicos dado “Column1”

```
SELECT DISTINCT Column1FROM Table1
```

- ROWCOUNT
  - Define cuantos renglones se procesarán

```
SET ROWCOUNT 5
```

Fundamentos de SQL

SQL - Structured Query Language (3/6)

- Order By
  - Ordena la información resultante.

```
Select Name From Production.Product
Order By Name Asc
```

Fundamentos de SQL

Paginado con Offset

```
-- Brinca 3...
SELECT c.[Nombre del cliente]
FROM Cliente C
ORDER BY c.[Nombre del cliente]
OFFSET 3 ROWS;

-- Brinca 3 y lee los siguientes 5...
SELECT C.[Nombre del cliente]
FROM Cliente C
ORDER BY C.[Nombre del cliente]
OFFSET 3 ROWS FETCH NEXT 5 ROWS ONLY;
```

https://technet.microsoft.com/en-us/library/gg699618%28v=sql.110%29.aspx

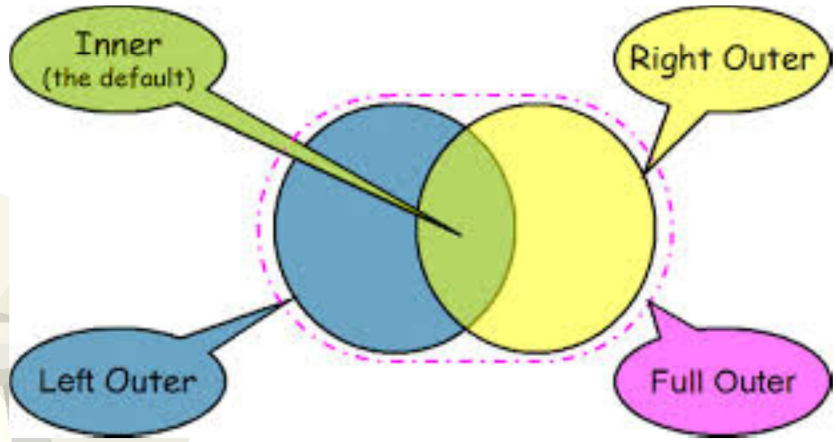
Fundamentos de SQL

Alumno\Ejercicios 01\Ejercicios 01.docx
Del 01 al 06

Fundamentos de SQL

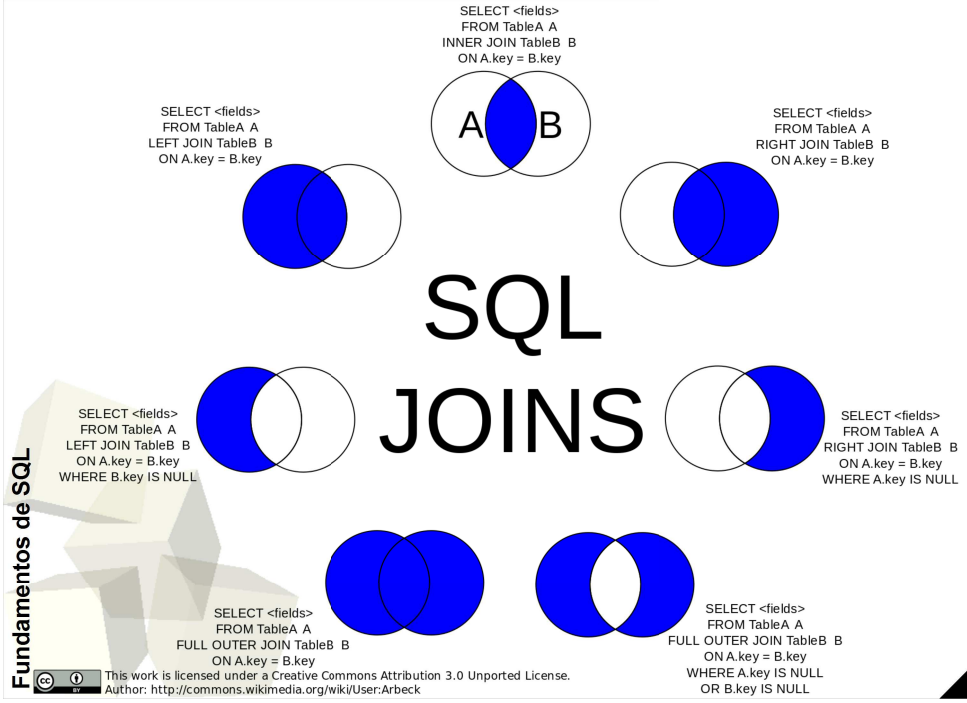
SQL - Structured Query Language (3/6)

- Joins
  - Une información de varias tablas, a través de la relación entre sus llaves correspondientes.



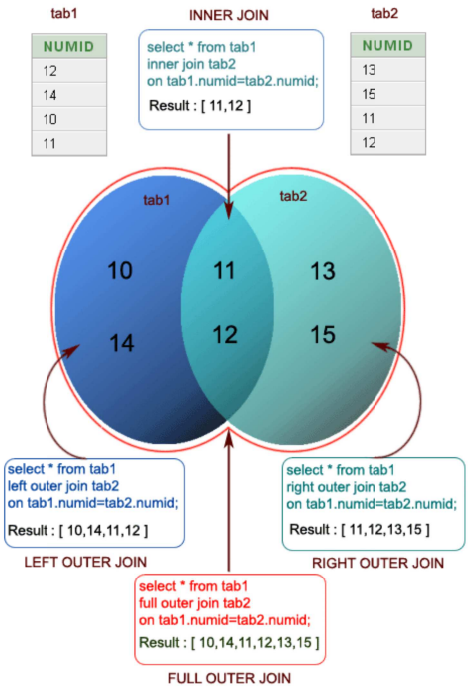
Fundamentos de SQL

SQL - Structured Query Language (3/6)



Fundamentos de SQL

SQL - Structured Query Language (3/6)



Fundamentos de SQL

SQL - Structured Query Language (4/6)

- Inner Join: Elementos coincidentes.
- Left Join: Dada una relación, trae los elementos de definidos en la tabla izquierda.
- Right Join: Dada una relación, trae los elementos de definidos en la tabla izquierda.
- Outter Join: Trae todos los elementos coincidentes, y nulos para el resto de los valores.

- Ver ejemplos en:
  - http://www.w3schools.com/sql/sql_join.asp
  - http://blog.codinghorror.com/a-visual-explanation-of-sql

Fundamentos de SQL

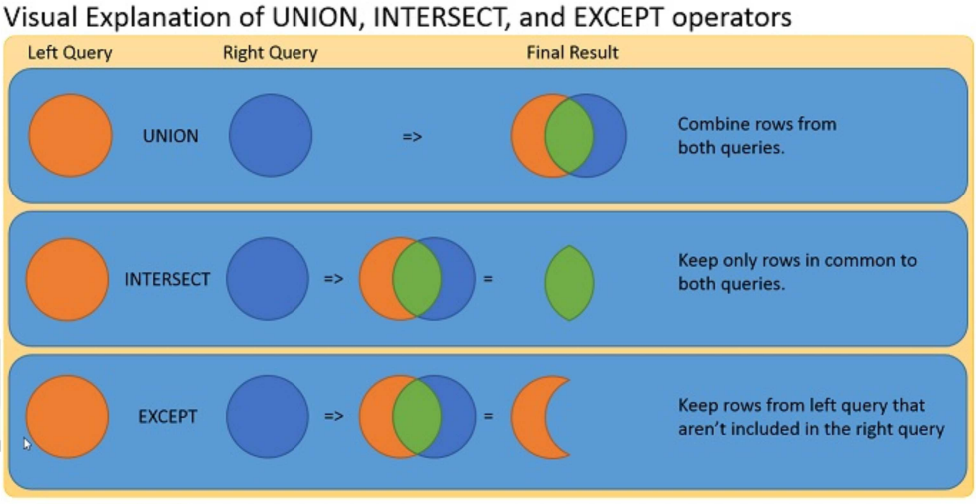
SQL - Structured Query Language (4/6)

- Union
  - Une los registros de varias tablas para presentarlas como una sola.

```
Select Top 1 EmployeeId As Id, LoginId As Login
From HumanResources.Employee
Union
Select Top 1 ContactId As Id, EmailAddress As Login
From Person.Contact
```

  - Ver ejemplos en:
    - http://www.w3schools.com/sql/sql_union.asp

SQL - Structured Query Language (4/6)



http://www.essentialsql.com/learn-to-use-union-intersect-and-except-clauses/

Fundamentos de SQL

SQL - Structured Query Language (5/6)

- Aggregate Functions
  - Count
    ```
    Select Count(ProductId) From Production.Product
    ```
  - AVG, MIN, MAX, SUM
    ```
    Select Avg(Weight) From Production.Product
    Select Min(Weight) From Production.Product
    Select Max(Weight) From Production.Product
    Select Sum(Weight) From Production.Product
    ```
- Group By
  ```
  -- Cuantos empleados administra un gerente...
  Select Count(EmployeeID) As CuantosEmpleados, ManagerID
  From HumanResources.Employee
  Group By ManagerID
  ```
- Having
  - Filtro sobre agrupamientos y/o funciones agregadas.
    - Para el caso anterior... agregar...
      ```
      Having ManagerID=3
      ```


SQL - Structured Query Language (6/6)

- Consultas Anidadas
  - MSSQL permite realizar una consulta dentro de otra consulta.
  - Ejemplo
    ```
    -- Trae todas las direcciones de las
     personas cuyo estado corresponde al
     territorio 1
    Select *
    From Person.Address
    Where StateProvinceID In
    (Select StateProvinceID From
     Person.StateProvince Where
     TerritoryID=1)
    ```

Fuente: http://www.sql-tutorial.com/sql-nested-queries-sql-tutorial/

Alumno\Ejercicios MSSQL\Ejercicios 01\
Ejercicios 01.docx, Del 7 al 18

DDL - Data Definition Language (1/4)

- Create
  - Crea objetos dentro de la base de datos.

```
Create Database <Nombre>
```

  - Tablas

```
CREATE TABLE TableName
(
Column1 DataType,
Column2 DataType,
Column3 DataType,
….)
```

- Alter
  - Modifica objetos dentro de la base de datos.
  - Alter Table <Nombre>
    - Ejemplo 1

```
ALTER TABLE TableName
ADD ColumnName DataType
```

    - Ejemplo 2

```
ALTER TABLE TableName
DROP ColumnName
```

Fundamentos de SQL

DDL - Data Definition Language (2/4)

- Drop
  - Se utiliza para borrar objetos de la base de datos.

```
Drop Database <Nombre,varchar,>
Go
Drop Table <Nombre,varchar,>
Go
```

- Uso de la sentencia “GO”
  - Le indica al motor de base de datos que ejecute el bloque de instrucciones como un solo lote (grupo de instrucciones).

DDL - Data Definition Language (3/4)

- Creación de Indices
  - Un índice es una herramienta que nos permite marcar o crear un directorio para acceder rápidamente registros en la base de datos.
  - Ejemplo:

```
CREATE INDEX <Nombre, varchar,>
ON <Tabla, varchar,> (<Campo,
 varchar,>)
```

DDL - Data Definition Language (4/4)

- Creación de Vistas
  - Una vista es una “tabla virtual” derivada de una expresión SQL.
  - Ejemplo:

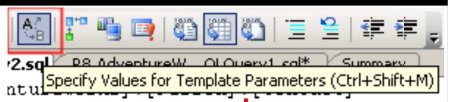
```
Create View MiVistaDireccion As
Select Top 5
       [AddressID]
      ,[AddressLine1]
      ,[AddressLine2]
FROM [AdventureWorks].[Person].[Address]
Go

Select * From MiVistaDireccion
Go
```

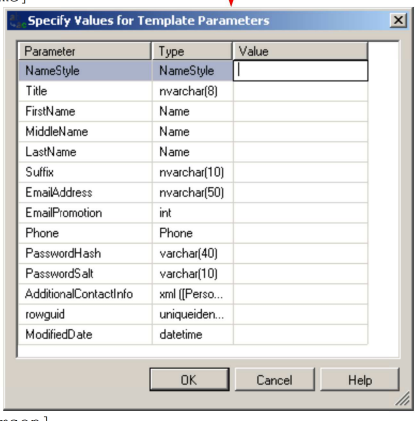
Fundamentos de SQL

DML – Data Manipulation Language (1/2)

- INSERT INTO
  - Agrega registros a una tabla.




```
INSERT INTO [AdventureWorks].[Person].[Contact]
           ([NameStyle]
           ,[Title]
           ,[FirstName],[MiddleName],[LastName]
           ,[Suffix]
           ,[EmailAddress]
           ,[EmailPromotion]
           ,[Phone]
           ,[PasswordHash]
           ,[PasswordSalt]
           ,[AdditionalContactInfo]
           ,[rowguid]
           ,[ModifiedDate])
     VALUES
           (<NameStyle, NameStyle,>
           ,<Title, nvarchar(8),>
           ,<FirstName, Name,>
           ,<MiddleName, Name,>
           ,<LastName, Name,>
           ,<Suffix, nvarchar(10),>
           ,<EmailAddress, nvarchar(50),>
           ,<EmailPromotion, int,>
           ,<Phone, Phone,>
           ,<PasswordHash, varchar(40),>
           ,<PasswordSalt, varchar(10),>
           ,<AdditionalContactInfo, xml ([Person].
   [AdditionalContactInfoSchemaCollection]),>
           ,<rowguid, uniqueidentifier,>
           ,<ModifiedDate, datetime,>)
```



Fundamentos de SQL

DML – Data Manipulation Language (2/2)

- UPDATE

```
UPDATE Table1
SET Column1 = Value1, Column2 = Value2, …
```

- DELETE
  - Borra registros de una tabla, opcionalmente se puede utilizar el comando WHERE para borrar registros específicos.

```
DELETE FROM Table1
```

- TRUNCATE TABLE
  - Borra todos los registros de una tabla.

```
TRUNCATE TABLE Person.Contact
```


Alumno\Ejercicios MSSQL\Ejercicios 01\
Ejercicios 01.docx, Del 19 al 23

Alumno\Ejercicios MSSQL\Ejercicios 02\
Ejercicios 02.docx, Del 01 al 07

Fundamentos de SQL

TCL – Transaction Control Language (1/3)

- Transacciones
  - Proceso que garantiza la ejecución correcta de las operaciones de lectura/escritura sobre una base de datos.
  - Propiedades Mínimas de una Transacción (ACID)
    - (A)tomic
      - Garantiza la realización de la transacción como un *todo,* sin dejar ninguna operación a medias. Todo se ejecuta o todo falla.
    - (C)onsistency
      - Permite completar la ejecución de una transacción consistente y de acuerdo a las restricciones de la base de datos (ej. Relaciones, llaves primarias, etc.)
    - (I)solated
      - Las transacciones no son afectadas por otras transacciones y/o procesos de lectura/escritura.
    - (D)urability
      - Las transacciones se guardan física y correctamente en el disco.

Fuente: http://es.wikipedia.org/wiki/ACID

TCL – Transaction Control Language (2/3)

- Transacciones en MSSQL
  - "AutoCommit"
    - Todas las instrucciones en MSSQL ejecutan son ejecutadas con "autocommit", es decir se autocompleta y se guardan los resultados.
  - Para manejar varias instrucciones dentro de una transacción se tiene que utilizar:
    - Begin Tran
      - Inicia una transacción.
    - Commit Tran
      - Da por terminada una transacción.
    - Rollback Tran
      - Deshace las acciones ejecutadas por la transacción.
  - Usualmente utilizamos transacciones desde un procedimiento almacenado (store procedure) o un lenguaje de programación.

TCL – Transaction Control Language (3/3)

- Ejemplo

```
USE xtreme_es
GO

Select Count(*) From Cliente
GO

BEGIN TRAN

TRUNCATE TABLE dbo.Cliente

ROLLBACK TRAN

Select Count(*) From Cliente
GO
```