

Capítulo 4: Modelo físico relacional de base de datos

Capítulo 5: Introducción al lenguaje SQL

Capítulo 6: Implementando consultas básicas



5

Introducción al lenguaje SQL

SQL y modelamiento de base de datos - SQL Server 2016

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Objetivos

- Identificar la sintaxis del lenguaje SQL de base de datos.
- Aplicar las sentencias de definición de datos (DDL).
- Establecer las diferencias entre SQL Server 2016 y SQL.
- Diferenciar los dialectos del lenguaje según el manejador de base de datos.

5 - 2

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Agenda

- Definición del lenguaje SQL
- SQL-Server 2016 y Transact-SQL (T-SQL)
- Implementando la estructura de una base de datos con lenguaje T- SQL
- Importando datos desde diversos fuentes de información

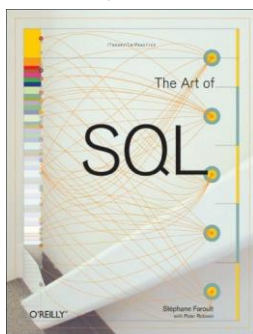
5 - 3

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Definición del lenguaje SQL ¿Qué es Structured Query Language o SQL?

- Es un lenguaje de gestión de datos de base de datos relacionales.
- No es un software, sino un dialecto con reglas gramaticales.
- Es declarativo y permite especificar diversas operaciones.



5 - 4

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Definición del lenguaje SQL

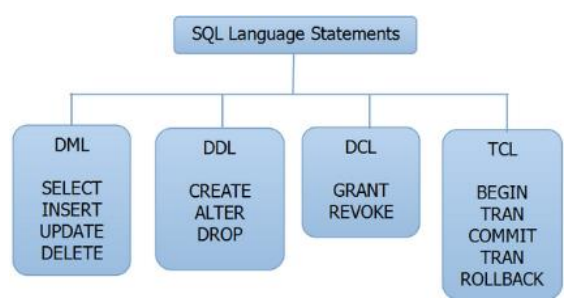
Clasificación

Lenguaje definición de datos (DDL):

- Estructura de una base de datos: create, drop y alter.

Lenguaje manipulación de datos (DML):

- Administración de los datos: select, insert, update y delete.



Definición del lenguaje SQL

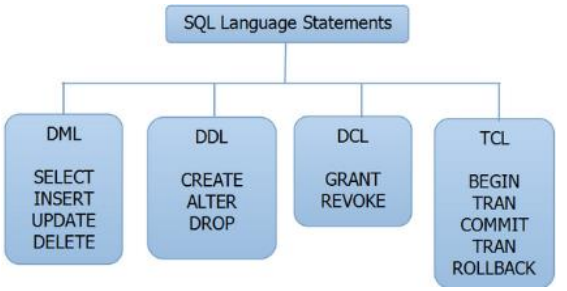
Clasificación

Lenguaje de control de datos (DCL):

- Privilegios de acceso a los datos: grant y revoke.

Lenguaje control transacciones (TCL):

- Administración de las transacciones: begin, tran, commit y rollback.



SQL-Server 2016 y Transact-SQL (T-SQL) Objetos de una base de datos

Estructura de datos:

- Base de datos
- Tablas
- Índices

Integridad referencial:

- Primary key
- Foreign key
- Defaults
- Checks



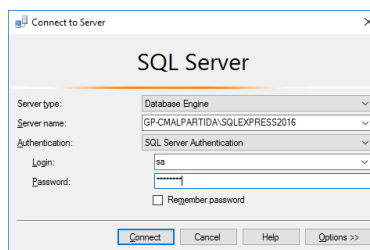
5 - 7

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos con lenguaje T-SQL

```
CREATE DATABASE [nombre]
ON
PRIMARY (NAME = DATA,
FILENAME = 'C:\DATA.MDF',
SIZE = 5,
MAXSIZE = 50,
FILEGROWTH = 15%)
LOG ON (NAME = LOG,
FILENAME = 'C:\LOG.LDF',
SIZE = 2,
MAXSIZE = 10,
FILEGROWTH = 15%)
GO
```



```
DROP DATABASE [nombre]
```

5 - 8

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos

Crear una tabla (table)

```
CREATE TABLE dbo.Territories
( TerritoryID      varchar(20)      NOT NULL,
  TerritoryDescription char(50)      NOT NULL,
  RegionID         int              NOT NULL
)
GO

CREATE TABLE dbo.Employees
( EmployeeID      int IDENTITY(1,1) NOT NULL,
  LastName         varchar(20)      NOT NULL,
  FirstName        varchar(10)      NOT NULL,
  Title            varchar(30)      NULL,
  TitleOfCourtesy  varchar(25)      NULL,
  BirthDate        datetime         NULL,
  HireDate         datetime         NULL,
  Address          varchar(60)      NULL,
  City             varchar(15)      NULL,
  Region           varchar(15)      NULL,
  PostalCode       varchar(10)      NULL,
  Country          varchar(15)      NULL,
  HomePhone        varchar(24)      NULL,
  Extension        varchar(4)       NULL,
  Photo            image            NULL,
  Notes            varchar(max)     NULL,
  ReportsTo        int              NULL,
  PhotoPath        varchar(255)     NULL
)
```

5 - 9

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos

Crear una tabla usando datos XML (table)

- Es posible incluir en un campo de una tabla datos en formato XML (eXtensible Markup Language).
- El tipo de datos XML permite almacenar datos que dependen del registro de la misma tabla y posiblemente evitar diseñar un maestro – detalle.

```
CREATE TABLE dbo.Proprietarios
( PropietariosCodigo char(10)      NOT NULL,
  PropietariosNombre  varchar(200) NOT NULL,
  Mascotas            XML          NULL,
  PropietariosTelefono varchar(30) NOT NULL,
  CONSTRAINT PK_Propietarios PRIMARY KEY (PropietariosCodigo)
)
go
```

5 - 10

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos

Crear una tabla temporal

- Las tablas temporales son utilizadas para almacenar cálculos intermedios en transacciones que requieren de grandes cantidades de datos.
- Las tablas temporales son de dos tipos:
 - Temporales locales
 - Temporales globales

```
CREATE TABLE #Prueba
( Codigo      char(04)      NOT NULL,
  Descripcion  varchar(100) NOT NULL,
  CONSTRAINT PK_Pueba PRIMARY KEY (Codigo)
)
go

CREATE TABLE ##DatosEmpresa
( Codigo      char(03)      NOT NULL,
  Descripcion  varchar(100) NOT NULL,
  Direccion    varchar(100) NOT NULL
)
go
```

5 - 11

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos

Crear una tabla usando IDENTITY

- IDENTITY es una propiedad que permite que un campo en una tabla se incremente de manera automática al insertar los registros en ella.
- Para el uso de la propiedad IDENTITY el tipo de dato debe ser INT.
- Es necesario definir un valor inicial y un valor de incremento.

```
CREATE TABLE dbo.Personas
( Codigo      int IDENTITY (1,1) NOT NULL,
  NombreCompleto  varchar(200) NOT NULL,
  CONSTRAINT PK_Personas PRIMARY KEY (Codigo)
)
go
```

5 - 12

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos

Tipo de dato definido por el usuario

- SQL Server permite crear tipos de datos que el usuario puede definir en base a los tipos de datos de SQL Server.

```
CREATE TYPE Codigo10 FROM char(10) NOT NULL
go
CREATE TYPE TextoObligatorio100 FROM varchar(10) NOT NULL
go
CREATE TYPE Precio FROM numeric(9,2) NULL
go

CREATE TABLE dbo.Agencias
(
  Codigo          Codigo10,
  Descripcion     TextoObligatorio100,
  Responsable     varchar(150),
  CONSTRAINT PK_Agencias PRIMARY KEY (Codigo)
)
GO
```



Implementando la estructura de base de datos

Uso de Secuencias

- Se puede definir una secuencia como un conjunto de valores que parten de un valor inicial, tienen un incremento o decremento.
- SQL Server permite la creación de secuencias que pueden ser utilizadas para la generación de códigos en las tablas.

Tipo de Dato	Valores
Tinyint	Rango de 0 a 255
Smallint	Rango -32,768 a 32,767
Int	Rango -2,147,483,648 a 2,147,483,647
Bigint	Rango -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 Este es el tipo de dato por defecto
Decimal y Numeric	Con una escala de CERO



Implementando la estructura de base de datos Uso de secuencias

```
CREATE SEQUENCE [Esquema.]NombreDeSecuencia
[AS [TipoEntero | TipoEnteroDefindoPorElUsuario]]
[START WITH <constante>]
[INCREMENT BY <constante>]
[{MINVALUE [<constant>]} | {NO MINVALUE}]
[{MAXVALUE [<constant>]} | {NO MAXVALUE}]
[CYCLE | {NO CYCLE}]
```

```
CREATE SEQUENCE ValoresPordefecto
```

```
CREATE SEQUENCE EquipoBasquet
AS int
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 12
CYCLE
```

5 - 15

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos Sinonimos

- Hacen referencia a los objetos de una base de datos de manera más entendible, clara, o simplemente para abreviar los nombres de tal manera que nos faciliten su uso en nuestras consultas.

```
CREATE SYNONYM [Esquema.]NombreSinonimo
FOR [Esquema.]NombreObjeto
go
```

5 - 16

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos Columnas calculadas

- Una columna calculada es una columna virtual que no está almacenada físicamente en la base de datos.

```
-- Crear la base de datos
CREATE DATABASE NORTHWIND
go

-- Activarla con USE
USE NORTHWIND
go

CREATE TABLE dbo.OrderDetails
( OrderID int NOT NULL,
  ProductID int NOT NULL,
  UnitPrice money NOT NULL,
  Quantity smallint NOT NULL,
  Total AS (UnitPrice * Quantity)
)
go
```

5 - 17

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos Crear un primary key (llave primaria)

```
ALTER TABLE dbo.Categories
ADD CONSTRAINT PK_Categories PRIMARY KEY
CLUSTERED (CategoryID)
GO

ALTER TABLE dbo.Customers
ADD CONSTRAINT PK_Customers PRIMARY KEY
CLUSTERED (CustomerID)
GO

ALTER TABLE dbo.Employees
ADD CONSTRAINT PK_Employees PRIMARY KEY
CLUSTERED (EmployeeID)
GO

ALTER TABLE dbo.EmployeeTerritories
ADD CONSTRAINT PK_EmployeeTerritories PRIMARY KEY
CLUSTERED (EmployeeID, TerritoryID)
GO
```

5 - 18

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos

Crear un índice (index)

```
CREATE NONCLUSTERED INDEX CategoryName
ON Categories (CategoryName ASC)
GO

CREATE NONCLUSTERED INDEX City
ON Customers (City ASC)
GO

CREATE NONCLUSTERED INDEX CompanyName
ON Customers (CompanyName ASC)
GO
|
CREATE NONCLUSTERED INDEX PostalCode
ON Customers (PostalCode ASC)
GO

CREATE NONCLUSTERED INDEX Region
ON Customers (Region ASC)
GO
```

5 - 19

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos

Crear foreign key (llave foránea)

```
ALTER TABLE dbo.Territories
ADD CONSTRAINT FK_Territories_Region FOREIGN KEY(RegionID)
REFERENCES dbo.Region (RegionID)
GO

ALTER TABLE dbo.EmployeeTerritories
ADD CONSTRAINT FK_EmployeeTerritories_Employees FOREIGN KEY(EmployeeID)
REFERENCES dbo.Employees (EmployeeID)
GO

ALTER TABLE dbo.EmployeeTerritories
ADD CONSTRAINT FK_EmployeeTerritories_Territories FOREIGN KEY(TerritoryID)
REFERENCES dbo.Territories (TerritoryID)
GO

ALTER TABLE dbo.Orders
ADD CONSTRAINT FK_Orders_Customers FOREIGN KEY(CustomerID)
REFERENCES dbo.Customers (CustomerID)
GO

ALTER TABLE dbo.Orders
ADD CONSTRAINT FK_Orders_Employees FOREIGN KEY(EmployeeID)
REFERENCES dbo.Employees (EmployeeID)
GO
```

5 - 20

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos

Crear una restricción (check)

```
ALTER TABLE dbo.Employees WITH CHECK
    ADD CONSTRAINT CK_Birthdate CHECK (BirthDate < getdate())
GO

ALTER TABLE dbo.OrderDetails WITH CHECK
    ADD CONSTRAINT CK_Discount CHECK (Discount >= 0 and Discount <= 1)
GO

ALTER TABLE dbo.OrderDetails WITH CHECK
    ADD CONSTRAINT CK_Quantity CHECK (Quantity > 0)
GO

ALTER TABLE dbo.OrderDetails WITH CHECK
    ADD CONSTRAINT CK_UnitPrice CHECK (UnitPrice >= 0)
GO

ALTER TABLE dbo.Products WITH CHECK
    ADD CONSTRAINT CK_Products_UnitPrice CHECK (UnitPrice >= 0)
GO

ALTER TABLE dbo.Products WITH CHECK
    ADD CONSTRAINT CK_ReorderLevel CHECK (ReorderLevel >= 0)
GO
```

5 - 21

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Implementando la estructura de base de datos

Crear un valor por defecto (default)

```
ALTER TABLE dbo.OrderDetails
    ADD CONSTRAINT DF_Order_Details_Discount DEFAULT (0) FOR Discount
GO

ALTER TABLE dbo.OrderDetails
    ADD CONSTRAINT DF_Order_Details_Quantity DEFAULT (1) FOR Quantity
GO

ALTER TABLE dbo.OrderDetails
    ADD CONSTRAINT DF_Order_Details_UnitPrice DEFAULT (0) FOR UnitPrice
GO

ALTER TABLE dbo.Orders
    ADD CONSTRAINT DF_Orders_Freight DEFAULT (0) FOR Freight
GO

ALTER TABLE dbo.Products
    ADD CONSTRAINT DF_Products_Discontinued DEFAULT (0) FOR Discontinued
GO

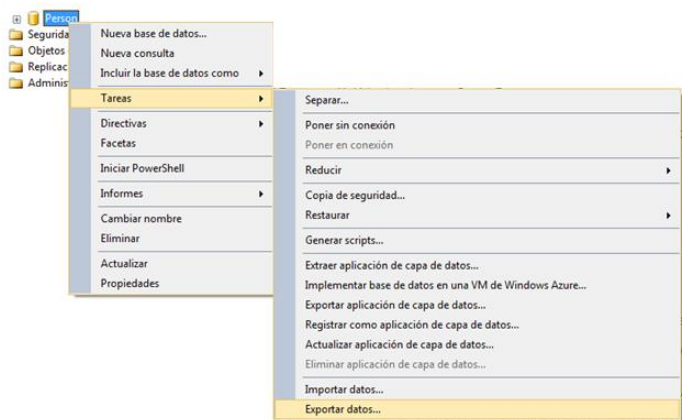
ALTER TABLE dbo.Products
    ADD CONSTRAINT DF_Products_ReorderLevel DEFAULT (0) FOR ReorderLevel
GO
```

5 - 22

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Importando datos desde diversas fuentes Microsoft Excel

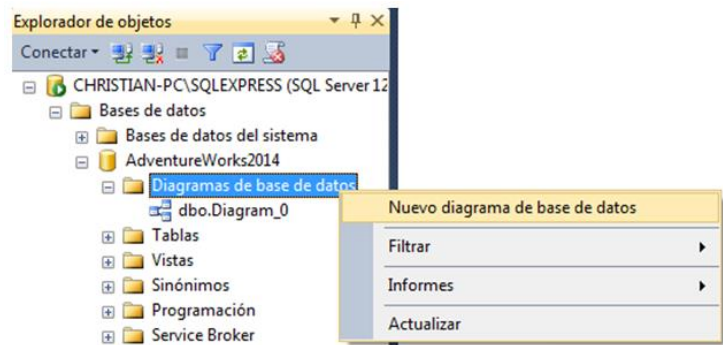


5 - 23

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Importando datos desde diversas fuentes Diagrama entidad relación



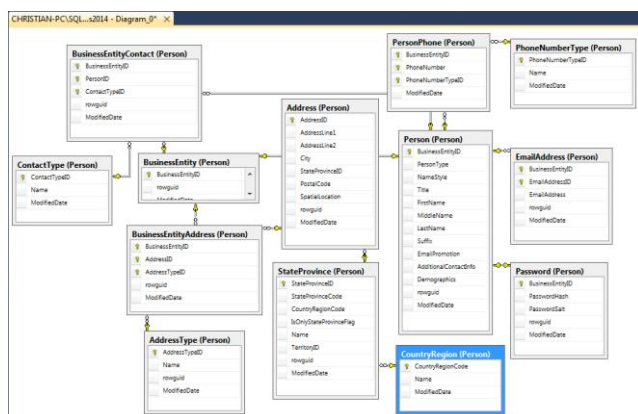
5 - 24

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Importando datos desde diversas fuentes

Diagrama entidad relación



Microsoft
SQL Server
2016

5 - 25

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Ejercicio Nº 5.1: Utilizar las principales sentencias de definición de datos (DDL)

Al finalizar el laboratorio, el alumno logrará:

- Crear una base de datos.

5 - 26

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Ejercicio Nº 5.2: Utilizar las principales sentencias de definición de datos (DDL)

Al finalizar el laboratorio, el alumno logrará:

- Crear los tipos de datos definidos por el usuario de una base de datos.

5 - 27

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Ejercicio Nº 5.3: Utilizar las principales sentencias de definición de datos (DDL)

Al finalizar el laboratorio, el alumno logrará:

- Crear las tablas de una base de datos.

5 - 28

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Ejercicio Nº 5.4: Utilizar las principales sentencias de definición de datos (DDL)

Al finalizar el laboratorio, el alumno logrará:

- Crear los sinónimos de una base de datos.

5 - 29

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Ejercicio Nº 5.5: Utilizar las principales sentencias de definición de datos (DDL)

Al finalizar el laboratorio, el alumno logrará:

- Crear las llaves primarias de una base de datos.

5 - 30

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Ejercicio Nº 5.6: Utilizar las principales sentencias de definición de datos (DDL)

Al finalizar el laboratorio, el alumno logrará:

- Crear las llaves foráneas de una base de datos.

5 - 31

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Ejercicio Nº 5.7: Utilizar las principales sentencias de definición de datos (DDL)

Al finalizar el laboratorio, el alumno logrará:

- Crear reglas de negocio a las tablas.

5 - 32

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Ejercicio N° 5.8: Utilizar las principales sentencias de definición de datos (DDL)

Al finalizar el laboratorio, el alumno logrará:

- Importar datos desde Microsoft Excel y Microsoft Access.

5 - 33

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Lecturas adicionales

Para obtener información adicional, puede consultar los siguientes enlaces:

- <http://quidel.inele.ufro.cl/~pvalenzu/tutoriales/sql/sql0.html>
- http://latam.softlinegroup.com/rs/232-AER-142/images/Microsoft_Press_eBook_Intro_SQL%20Server_2016_Preview_Ed_PDF.pdf
- <http://www.formacionprofesional.info/tutoriales-y-manuales-de-sql-server-2016/>

5 - 34

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Resumen

En este capítulo, usted aprendió que:

- El lenguaje SQL es necesario para la gestión de información de base de datos en SQL Server 2016.
 - Select
 - Update
 - Insert
 - Delete

5 - 35

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Resumen

En este capítulo, usted aprendió:

- El lenguaje T-SQL es necesario para la creación de objetos de base de datos en SQL Server 2016.
 - Table
 - Primary key
 - Foreign key
 - Index
 - Constraint
 - Default
 - Synonym

5 - 36

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Tarea N° 5: Implementar una base de datos aplicando los conceptos aprendidos

Aplicar los conceptos del modelado de base de datos y T-SQL.

Al finalizar el laboratorio, el alumno logrará:

- Diseñar un modelo de base de datos a partir de un caso.
- Establecer los objetos de datos necesarios del negocio:
 - Tablas
 - Campos
 - Funciones
 - Procedimientos
 - Triggers

