SQL SERVER IMPLEMENTACIÓN

Eric Gustavo Coronel Castillo
www.desarrollasoftware.com
gcoronelc@gmail.com

CORONEL
DESARROLLA SOFTWARE

MICROSOFT SQL SERVER

NIVEL I - IMPLEMENTACIÓN

Creación de Tablas y Restricciones de Integridad de Datos

Eric Gustavo Coronel Castillo
www.desarrollasoftware.com
gcoronelc@gmail.com

LOGRO ESPERADO

Se espera que al finalizar esta clase el participante pueda implementar una base de datos, esto quiere decir, sus tablas y con sus respectivas restricciones.

Tipos de Datos

- **CATEGORIAS**
  - Numéricos exactos
  - Numéricos aproximados
  - Fecha y hora
  - Cadenas de caracteres No Unicode
  - Cadenas de caracteres Unicode
  - Cadenas binarias
  - Otros tipos de datos

Tipos de Datos

- **Numéricos exactos**
  - bigint — 8 bytes
  - bit — 1, 0 o NULL
  - decimal(p,s) — Depende de la presición (p)
  - int — 4 bytes
  - money — 8 bytes
  - numeric(p,s) — Depende de la presición (p)
  - smallint — 2 bytes
  - smallmoney — 4 bytes
  - tinyint — 1 byte

Tipos de Datos

- **Numéricos aproximados**
  - float[(n)] Depende del valor de n.
  - real 4 bytes
- **Fecha y hora**
  - date de 0001-01-01 a 9999-12-31 3 bytes
  - datetime de 1753-01-01 a 9999-12-31 8 bytes
  - datetime2(n) 0001-01-01 a 9999-12-31 de 6 a 8 bytes
  - datetimeoffset(n) Fecha, hora y desplazamiento de 8 a 10 bytes
  - smalldatetime 1900-01-01 a 2079-06-06 4 bytes
  - time(n) HH:MI:SS[.nnnnnnn] de 3 a 5 bytes

Tipos de Datos

- **Cadenas de caracteres NO UNICODE**
  - char(n) De longitud fija hasta 8.000 caracteres.
  - varchar(n) De longitud variable hasta 8.000 caracteres.
  - text De longitud variable hasta 2.147.483.647 caracteres.
- **Cadenas de caracteres UNICODE**
  - nchar(n) De longitud fija hasta 4.000 caracteres.
  - nvarchar(n) De longitud variable hasta 4.000 caracteres.
  - ntext De longitud variable 1.073.741.823 caracteres.

Tipos de Datos

- **Cadenas binarias**
  - binary[(n)] De longitud fija hasta 8.000 bytes.
  - varbinary[(n)] De longitud variable hasta 8.000 bytes.
  - image De longitud variable hasta 2.147.483.647 bytes.
- **Otros tipos de datos**
  - cursor
    - Un tipo de datos para las variables o para los parámetros de resultado de los procedimientos almacenados que contiene una referencia a un cursor.
  - table
    - Es un tipo de datos especial que se puede utilizar para almacenar un conjunto de resultados para su procesamiento posterior. table se utiliza principalmente para el almacenamiento temporal de un conjunto de filas devuelto como el conjunto de resultados de una función con valores de tabla.
  - xml ([CONTENT | DOCUMENT] xml_schema_collection)
    - Es el tipo de datos que almacena datos de XML. Puede almacenar instancias de xml en una columna o una variable de tipo xml.

Creación de Tablas

- **SINTAXIS**

```
CREATE TABLE table_name (
    { < column_definition > | < table_constraint > } [ ,...n ]
)
< column_definition > ::=
    { column_name data_type }
    [ { DEFAULT constant_expression
        | [ IDENTITY [ ( seed , increment ) ] ]
    } ]
    [ ROWGUIDCOL ]
    [ < column_constraint > [ ...n ] ]
```

Creación de Tablas

```
USE tienda
GO
CREATE TABLE maestros.articulo(
        art_id INT NOT NULL IDENTITY(1,1),
        art_nombre VARCHAR(100) NOT NULL,
        art_pre_costo MONEY NOT NULL,
        art_pre_venta MONEY NOT NULL,
        art_stock INT NOT NULL
)
GO
```

Restricciones de Integridad

- **Definición de Integridad**
  - La Integridad es el término utilizado para decir que la información almacenada tiene calidad. El DBMS debe asegurar que los datos se almacenan de acuerdo a las políticas previamente determinadas por el DBA.
  - En otras palabras, el DBMS debe principalmente, a este respecto, comprobar las restricciones de integridad, controlar la correcta ejecución de las actualizaciones y recuperar la base de datos en caso de pérdida.
  - Un control de integridad o restricción es aquel que nos permite definir con precisión el rango de valores válidos para un elemento y/o las operaciones que serán consideradas válidas en la relación de tales elementos.

Restricciones de Integridad

- **Integridad de Entidad**
  - La integridad de entidad define una fila como entidad única para una tabla determinada.
  - La integridad de entidad exige la integridad de las columnas de los identificadores o la clave primaria de una tabla, mediante índices y restricciones UNIQUE, o restricciones PRIMARY KEY.

art_id	art_nombre	art_pre_costo	art_pre_venta	art_stock
1	TELEVISOR	790.00	1799.00	600
2	LAPTOP	1570.00	1399.00	200
3	IMPRESORA	175.00	350.00	300
4	CAMARA DIGITAL	489.00	988.00	377
5	MP4	570.00	1300.00	55
6	LAVADORA	677.00	1450.00	68

Restricciones de Integridad

- **Integridad de Dominio**
  - La integridad de dominio viene dada por la validez de las entradas para una columna determinada.
  - Puede exigir la integridad de dominio para restringir el tipo mediante tipos de datos, el formato mediante reglas y restricciones CHECK, o el intervalo de valores posibles mediante restricciones FOREIGN KEY, restricciones CHECK, definiciones DEFAULT, definiciones NOT NULL y reglas.

art_id	art_nombre	art_pre_costo	art_pre_venta	art_stock
1	TELEVISOR	790.00	1799.00	600
2	LAPTOP	1570.00	1399.00	200
3	IMPRESORA	175.00	350.00	300
4	CAMARA DIGITAL	489.00	988.00	377
5	MP4	570.00	1300.00	55
6	LAVADORA	677.00	1450.00	68

Restricciones de Integridad

- **Integridad Referencial**
  - La integridad referencial protege las relaciones definidas entre las tablas cuando se crean o se eliminan filas. En SQL Server la integridad referencial se basa en las relaciones entre claves foráneas y claves primarias, mediante restricciones FOREIGN KEY.
  - La integridad referencial garantiza que los valores de clave sean coherentes en las distintas tablas. Para conseguir esa coherencia, es preciso que no haya referencias a valores inexistentes y que, si cambia el valor de una clave, todas las referencias a ella se cambien en consecuencia en toda la base de datos.

Restricciones de Integridad

- ## Integridad Referencial

Tabla: Cliente

ClienteId	Nombre	Email
C1001	Sergio Matsukawa Maeda	smatsukawa@isil.pe
C1002	Hugo Valencia Morales	hvalencia@terra.com.pe
C1003	Gustavo Coronel Castillo	gcoronelc@gmail.com
C1004	Ricardo Marcelo Villalobos	ricardomarcelo@hotmail.com

Tabla: Pedido

PedidoId	Fecha	ClienteId	Importe
1001	04/Ene/2011	C1002	5,000.00
1002	15/Ene/2011	C1001	10,000.00
1003	13/Feb/2011	C1004	8,500.00
1004	07/Mar/2011	C1001	7,400.00

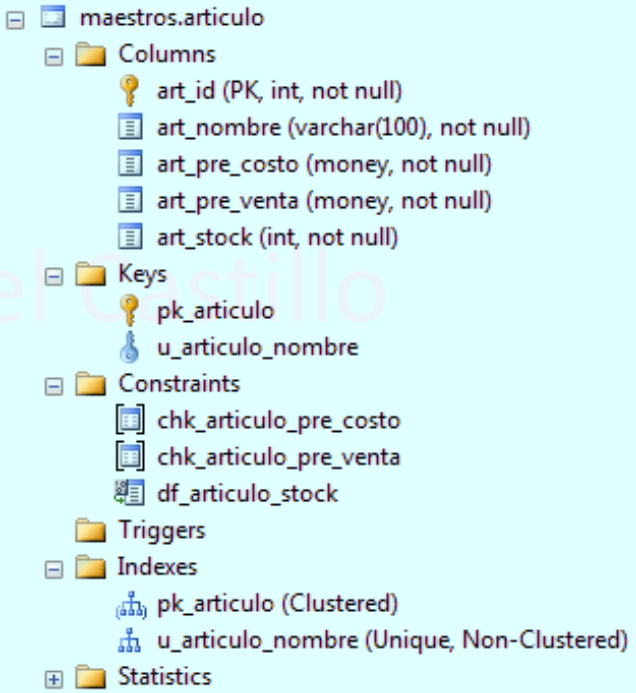
ClienteId en la tabla Pedido es una Clave Foránea.

Implementación de Restricciones

NIVEL DE INTEGRIDAD	TIPO DE RESTRICCIÓN	DESCRIPCIÓN
Dominio (columna)	DEFAULT	Especifica el valor que se mantendrá para la columna cuando un valor no se ha ingresado explícitamente en una sentencia INSERT.
	CHECK	Especifica los valores de los datos que son aceptables en la columna.
Entidad (fila)	PRIMARY KEY	Identifica cada registro o fila como única. Se crea un índice para mejorar el rendimiento. Los valores nulos no son permitidos.
	UNIQUE	Previene la duplicación de las llaves alternas, y se asegura que un índice se crea para que mejore el rendimiento. Se permiten valores nulos.
Referencial (relación)	FOREIGN KEY	Define la columna o combinación de columnas de una tabla secundaria, cuyos valores dependen de la llave primaria de una tabla primaria.

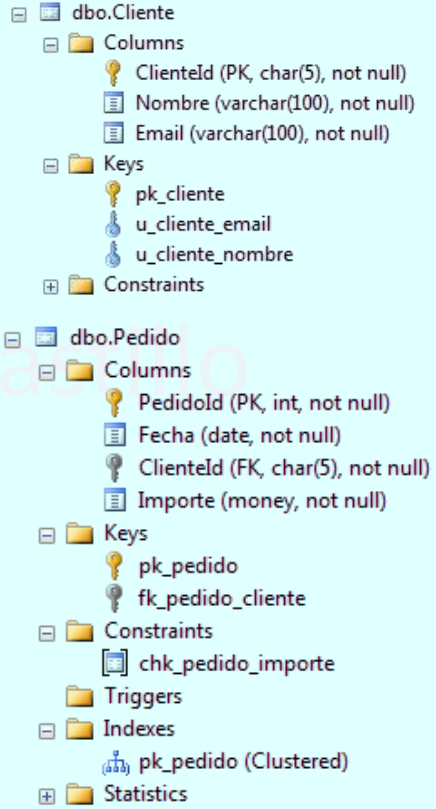
Implementación de Restricciones

```
CREATE TABLE maestros.articulo(
    art_id INT NOT NULL IDENTITY(1,1),
    art_nombre VARCHAR(100) NOT NULL,
    art_pre_costo MONEY NOT NULL,
    art_pre_venta MONEY NOT NULL,
    art_stock INT NOT NULL CONSTRAINT df_articulo_stock DEFAULT 0,
    CONSTRAINT pk_articulo PRIMARY KEY(art_id),
    CONSTRAINT u_articulo_nombre UNIQUE (art_nombre),
    CONSTRAINT chk_articulo_pre_costo CHECK(art_pre_costo > 0),
    CONSTRAINT chk_articulo_pre_venta CHECK(art_pre_venta > art_pre_costo)
)
GO
```



Implementación de Restricciones

```
CREATE TABLE Cliente(
    ClienteId CHAR(5) NOT NULL  CONSTRAINT pk_cliente PRIMARY KEY,
    Nombre VARCHAR(100) NOT NULL CONSTRAINT u_cliente_nombre UNIQUE,
    Email VARCHAR(100) NOT NULL CONSTRAINT u_cliente_email UNIQUE
)
GO
CREATE TABLE Pedido(
    PedidoId INT NOT NULL IDENTITY(1000,1) CONSTRAINT pk_pedido PRIMARY KEY,
    Fecha DATE NOT NULL, ClienteId CHAR(5) NOT NULL,
    Importe MONEY NOT NULL CONSTRAINT chk_pedido_importe CHECK(Importe>0),
    CONSTRAINT fk_pedido_cliente FOREIGN KEY (ClienteId)
        REFERENCES Cliente ON DELETE NO ACTION ON UPDATE NO ACTION
)
GO
```

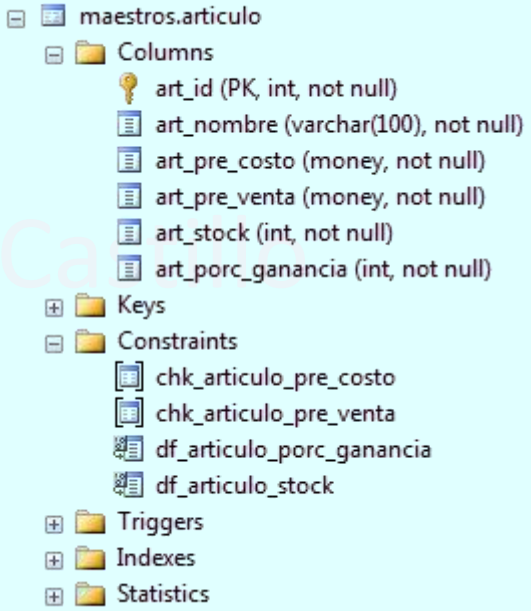


Modificar de la Definición de una Tabla

```
ALTER TABLE [ database_name . ] [ schema_name . ] table_name {
    ALTER COLUMN column_name
        type_name [ ( precision [ , scale ] ) ]
        [ NULL | NOT NULL ]
        [ WITH { CHECK | NOCHECK } ]
    | ADD
        {
            <column_definition>
            | <table_constraint>
        } [ ,...n ]
    | DROP
        {
            [ CONSTRAINT ] constraint_name
            | COLUMN column_name
        } [ ,...n ]
```

Modificar de la Definición de una Tabla

```
ALTER TABLE maestros.articulo
ADD art_porc_ganancia INT NOT NULL
   CONSTRAINT df_articulo_porc_ganancia
   DEFAULT 0
GO
```



GRACIAS TOTALES

udemy

FUNDAMENTOS DE PROGRAMACIÓN CON JAVA

Inicia tu aprendizaje, utilizando las mejores prácticas de programación

PROGRAMACIÓN DE BASE DE DATOS ORACLE CON PL/SQL

Aprende a obtener el mejor rendimiento de tú base de datos

CURSO PROFESIONAL DE JAVA ORIENTADA A OBJETOS

Aprende programación en capas, patrones y buenas prácticas

PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JDBC

Aprende a programar correctamente con JDBC