

SQL Server



Tablas

Temario



8. Manipulación de tablas.

- 8.1 Creación de tablas
- 8.2. Eliminación de tablas
- 8.3. Modificación de tablas
- 8.4 Agregación de restricciones a las tablas

CREATE TABLE



- ❧ Se utiliza CREATE TABLE para crear tablas
- ❧ Sintaxis:

```
CREATE TABLE tabla  
    (columna tipo_de_datos[DEFAULT expr],  
     columna2 tipo_de_dato[DEFAULT expr]);
```

- ❧ Tabla: nombre de la tabla
- ❧ Columna: nombre de la columna
- ❧ Tipo_de_datos: tipo y longitud de la columna
- ❧ DEFAULT expr: especifica un valor por defecto si el valor es omitido en la instrucción INSERT. Previene la inserción de valores NULL. El valor por defecto puede ser una literal, una expresión o una función SQL; tales como GETDATE().

```

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 ] ]
< column_constraint > ::=
    [ CONSTRAINT constraint_name ]
    { [ NULL | NOT NULL ]
        | [ PRIMARY KEY | UNIQUE ]
        | REFERENCES ref_table [ ( ref_column ) ]
        [ ON DELETE { CASCADE | NO ACTION } ]
        [ ON UPDATE { CASCADE | NO ACTION } ]
    }
< table_constraint > ::=
    [ CONSTRAINT constraint_name ]
    { [ { PRIMARY KEY | UNIQUE }
        { ( column [ ,...n ] ) }
        ]
        | FOREIGN KEY
        ( column [ ,...n ] )
        REFERENCES ref_table [ ( ref_column [ ,...n ] ) ]
        [ ON DELETE { CASCADE | NO ACTION } ]
        [ ON UPDATE { CASCADE | NO ACTION } ]
    }

```

Ejemplos



```
CREATE TABLE depto  
  (noDep NUMERIC(2),  
   dNombre VARCHAR(14),  
   Loc VARCHAR(13));
```


Crear una tabla



- ❧ *table_name* Nombre de la nueva tabla. Los nombres de tabla deben cumplir las mismas reglas que los identificadores. El valor de *table_name* debe ser único dentro de la base de datos. Un valor de *table_name* puede contener un máximo de 128 caracteres.
- ❧ *column_name* , Nombre de una columna de la tabla Los nombres de columna deben cumplir las mismas reglas que los identificadores y deben ser exclusivos en la tabla.
- ❧ *data_type* , Especifica el tipo de datos de la columna. Para obtener información acerca de los tipos de datos, vea [Tipos de datos y RDA](#).
- ❧ DEFAULT , Especifica el valor que se asigna a la columna cuando no se ha suministrado ningún valor de forma explícita durante una acción de inserción. Se pueden aplicar definiciones con el valor DEFAULT a cualquier columna, excepto las definidas por la propiedad IDENTITY. Las definiciones DEFAULT desaparecen cuando la tabla se elimina. Se puede utilizar como predeterminado un valor constante.
- ❧ IDENTITY , Indica que la nueva columna es una columna de identidad. Cuando se agrega una nueva fila a la tabla, SQL Server Compact Edition proporciona un valor incremental único para la columna. Las columnas de identidad se suelen utilizar junto con restricciones PRIMARY KEY para servir como identificador de fila único de la tabla. La propiedad IDENTITY sólo se puede asignar a las columnas **int**. Sólo se puede crear una columna de identidad para cada tabla. Las restricciones DEFAULT y los valores predeterminados enlazados no se pueden utilizar en las columnas de identidad. Debe especificar tanto el valor de inicialización como el incremento, o bien ninguno de los dos. Si no se especifica ninguno, el valor predeterminado es (1,1).



- ❧ CONSTRAINT , Palabra clave opcional que indica el principio de una definición de restricción **PRIMARY KEY**, **UNIQUE** o **FOREIGN KEY**. Las restricciones son propiedades especiales que garantizan la integridad de los datos y crean tipos especiales de índices para la tabla y sus columnas.
- ❧ *constraint_name* , Nombre de una restricción. El valor de *constraint_name* es opcional y debe ser único dentro de la base de datos. Si no se especifica el valor de *constraint_name*, SQL Server Compact Edition genera un nombre de restricción.
- ❧ NULL | NOT NULL , Palabras clave que especifican si se permite el uso de valores Null en la columna. **NULL** no es estrictamente una restricción, pero se puede especificar del mismo modo que **NOT NULL**.
- ❧ PRIMARY KEY , Restricción que impone exclusividad de entidad para una o varias columnas determinadas mediante un índice único. Sólo se puede crear una restricción **PRIMARY KEY** por tabla.
- ❧ UNIQUE , Restricción que proporciona integridad a la entidad para una columna o columnas específicas mediante un índice exclusivo. Las columnas de una restricción **UNIQUE** pueden ser **NULL**, pero sólo se permite un valor NULL por columna. Una tabla puede tener varias restricciones **UNIQUE**.



- ❧ FOREIGN KEY...REFERENCES , Restricción que proporciona integridad referencial para los datos de la columna. Las restricciones **FOREIGN KEY** requieren que cada valor de la columna exista en la columna especificada en la tabla referenciada.
- ❧ *ref_table* , Nombre de la tabla referenciada por la restricción **FOREIGN KEY**.
- ❧ (*ref_column* [,...*n*]) , Columna o lista de columnas de la tabla referenciada por la restricción **FOREIGN KEY**.
- ❧ ON DELETE {CASCADE | NO ACTION} , Especifica qué sucede con una fila de la tabla que se ha creado cuando dicha fila tiene una relación referencial y la fila a la que se hace referencia se elimina de la tabla primaria. El valor predeterminado es **NO ACTION**.
- ❧ Si se especifica **CASCADE**, se elimina una fila de la tabla que hace referencia cuando se elimina la fila referenciada correspondiente de la tabla primaria. Si se especifica **NO ACTION**, SQL Server Compact Edition devuelve un error y se revierte la acción de eliminación de la fila a la que se hace referencia en la tabla primaria.
- ❧ ON UPDATE {CASCADE | NO ACTION} Especifica qué sucede con una fila de la tabla que se ha creado cuando dicha fila tiene una relación referencial y la fila a la que se hace referencia se actualiza en la tabla primaria. El valor predeterminado es **NO ACTION**.
- ❧ Si se especifica **CASCADE**, se actualiza la fila de la tabla que hace referencia cuando se actualiza la fila referenciada correspondiente en la tabla primaria. Si se especifica **NO ACTION**, SQL Server Compact Edition devuelve un error y se revierte la acción de actualización de la fila a la que se hace referencia en la tabla primaria.
- ❧ *column* , Columna o lista de columnas, entre paréntesis, que se utiliza en las restricciones de tabla para indicar las columnas que se utilizan en la definición de restricción.

Restricciones



- ❧ Restricciones **PRIMARY KEY**
Una tabla sólo puede incluir una restricción PRIMARY KEY.
- ❧ Cada PRIMARY KEY genera un índice.
- ❧ Todas las columnas definidas en una restricción PRIMARY KEY se deben definir como NOT NULL. Si no se especifica la capacidad de aceptar valores Null, se establece la capacidad de aceptar valores Null de todas las columnas que participen en una restricción PRIMARY KEY en NOT NULL.
- ❧ Restricciones **UNIQUE**
Cada restricción UNIQUE genera un índice.
- ❧ Restricciones **FOREIGN KEY**
Si se especifica un valor distinto de NULL en la columna de una restricción FOREIGN KEY, el valor debe existir en la columna referenciada; de lo contrario, se devolverá un error de infracción de clave externa.
- ❧ Las restricciones FOREIGN KEY pueden hacer referencia a otra columna de la misma tabla, lo que se denomina autorreferencia.

Restricciones



- ❧ La cláusula REFERENCES de una restricción FOREIGN KEY de nivel de columna sólo puede listar una columna de referencia. Debe tener el mismo tipo de datos que la columna en la que se defina la restricción.
- ❧ Las restricciones FOREIGN KEY sólo pueden hacer referencia a columnas de las restricciones PRIMARY KEY o UNIQUE de la tabla referenciada.
- ❧ Información adicional sobre las restricciones
Un índice creado para una restricción no se puede quitar con la instrucción DROP INDEX. La restricción se debe quitar con la instrucción ALTER TABLE DROP CONSTRAINT.
- ❧ Los nombres de restricción deben cumplir las reglas de los identificadores, excepto la que indica que el nombre no puede empezar con una almohadilla (#). Si no se indica la palabra clave CONSTRAINT y constraint_name, se asignará a la restricción un nombre generado por el sistema.
- ❧ Si se infringe una restricción en una instrucción INSERT, UPDATE o DELETE, se detendrá la instrucción.

Ejemplos



VER CODIGO SQLBASICO

Nota: para ver la estructura de la tabla creada
EXEC sp_help 'NombreTabla';

8.2 Eliminación de tablas



- ❧ La instrucción DROP TABLE elimina una tabla de la BD.
- ❧ Cuando se elimina una tabla, se pierden todos los datos y los índices asociados con ella
- ❧ La instrucción DROP TABLE, una vez realizada, es irreversible. Microsoft SQL Server no pregunta si desea confirmar esta acción.

Eliminación de tablas



❧ Sintaxis

❧ DROP TABLE <nombre_tabla>

❧ Notas:

- ❧ No se puede utilizar para eliminar una tabla a la que se hace referencia en una restricción FOREIGN KEY.
- ❧ Solución: eliminar la restricción FOREIGN KEY de la tabla a la que se hace referencia

8.3 Modificación de tablas

- ❧ Se utiliza la instrucción ALTER TABLE para:
 - ❧ Agregar nuevas columnas
 - ❧ Eliminar una columna ya existente
 - ❧ Definir un valor por defecto para una nueva columna
 - ❧ Eliminar o agregar constraints

Agregar columnas



- ✧ En el caso de que le hayan faltado columnas se puede utilizar la instrucción ALTER TABLE con la cláusula ADD
- ✧ Sintaxis:

```
ALTER TABLE tabla  
ADD columna tipo_de_datos[DEFAULT expr]  
[,columna tipo_de_datos]...;
```

- ✧ tabla= nombre de la tabla a modificar
- ✧ DEFAULT expr= especifica un valor por defecto si el valor es omitido en la instrucción INSERT
- ✧ Columna= nombre de la columna
- ✧ Tipo_de_datos= tipo y longitud de la columna

Ejemplo



```
ALTER TABLE EMP  
ADD AGE VARCHAR(9);
```

La nueva columna será la última

EMPNO	ENAME	SAL	HIREDATE	AGE
7698	BLAKE	34200	01-MAY-81	
7654	MARTIN	15000	28-SEP-81	
7499	ALLEN	19200	20-FEB-81	
7844	TURNER	18000	08-SEP-81	
...				
6 rows selected.				

Nota: si la tabla ya contiene registros, esta nueva columna será inicializada con NULL en todo los registros existentes.

Eliminar columnas



- Se puede eliminar columnas existentes en la tabla utilizando la instrucción `ALTER TABLE` con la cláusula `DROP`
- Sintaxis:

```
ALTER TABLE tabla  
DROP COLUMN columna;
```

8.4 Agregación de restricciones a las tablas

- ❧ Reglas para la creación de CONSTRAINTS (restricciones)
 - ❧ Todas las restricciones son almacenadas en el diccionario de datos.
 - ❧ Es importante asignar un nombre adecuado para su identificación
 - ❧ Si no se asigna un nombre, Microsoft SQL Server le asignará un nombre interno, que puede llegar a confundir al momento de obtener errores
 - ❧ Las restricciones se pueden definir al momento en que se crea la tabla o después de haberla creada
 - ❧ Hay restricciones a nivel de columnas o de tablas

Restricciones



- ⌘ Previenen borrados de una tabla si existen dependencias

Restricciones



❧ Constraints implícitos

- ❧ NOT NULL NN

- ❧ UNIQUE Key UK

- ❧ PRIMARY KEY PK

- ❧ FOREIGN KEY FK

❧ Se denominan implícitos porque la regla de validación ya está dada por el manejador

❧ Constraints explícitos

- ❧ CHEK CK

Restricciones



❧ Constraint NOT NULL

- ❧ El constraint NOT NULL asegura que no se permitan valores nulos en una columna. Las columnas que no tengan esa restricción pueden grabar valores nulos.
- ❧ Únicamente puede ser especificado a nivel columna

```
CREATE TABLE emp (  
  empno    NUMERIC(4),  
  ename    VARCHAR(10) NOT NULL,  
  job      VARCHAR(9),  
  mgr      NUMERIC(4),  
  hiredate DATETIME,  
  sal      NUMERIC(7,2),  
  comm     NUMERIC(7,2),  
  deptno   NUMERIC(2) NOT NULL);
```

```
... DEPTNO    NUMERIC(2),  
CONSTRAINT   emp_deptno_nn NOT NULL...
```

Restricciones



❧ UNIQUE Key

- ❧ Requiere que cada valor en la columna o conjunto de columnas sea único, es decir, dos renglones de la misma tabla no pueden tener el valor duplicado en la columna o columnas especificadas como UNIQUE
- ❧ Permite la inserción de valores nulos a menos que defina una constraint NOT NULL para la misma columna
- ❧ Si hay valores repetidos anteriormente, la modificación de una columna al tipo UNIQUE falla

Restricciones



PRIMARY KEY

- ❧ Crea la llave primaria de la tabla. Sólo puede existir una llave primaria por cada tabla.
- ❧ Puede ser una columna o un conjunto de ellas, e identifican de forma única a cada registro en la tabla.
- ❧ Forza a que los valores de la columna o columnas sean únicos y que no contengan valores nulos

PRIMARY KEY		
DEPT	DEPTNO	DNAME
	10	ACCOUNTING
	20	RESEARCH
	30	SALES
	40	OPERATIONS

Insert into

20	MARKETING	DALLAS
	FINANCE	NEW YORK

No permitido (DEPTNO-20 ya existe)

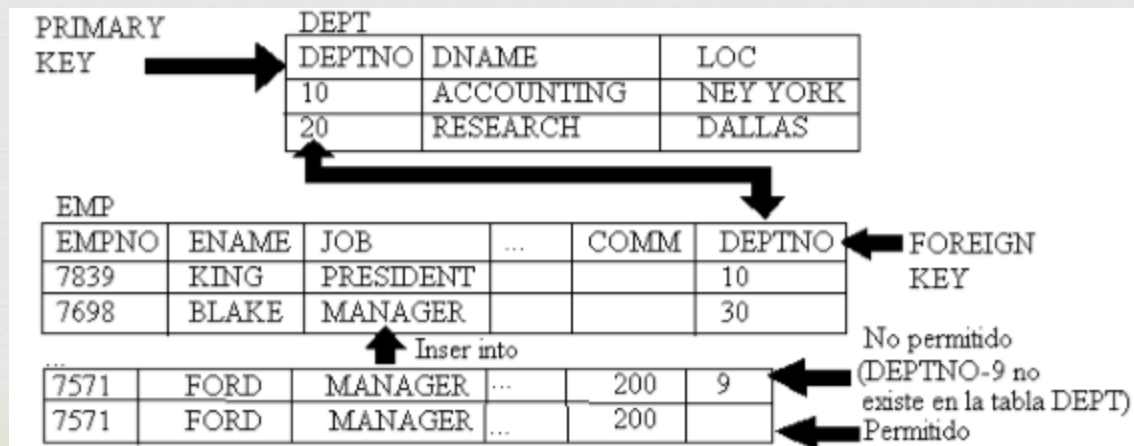
No permitido (DEPTNO es null)

Restricciones



FOREIGN KEY

- Designa una columna o combinación de ellas como una llave foránea y establece una relación entre una llave primaria en la misma tabla o en una tabla diferente
- El valor de una llave foránea debe coincidir con un valor existente en la tabla padre o ser nulo.



Restricciones



☞ CHECK

- ☞ Define una condición que cada renglón debe cumplir
- ☞ La condición puede ser definida al igual que las condiciones de las consultas
- ☞ Una columna puede tener múltiples CHECK

```
..., deptno NUMERIC(2),  
    CONSTRAINT emp_deptno_ok  
    ...CHECK(DEPTNO BETWEEN 10 AND 99), ...
```

Restricciones



⌘ Agregar constraints

- ⌘ Se puede agregar constraints a una tabla existente utilizando la excepción ALTER TABLE con la clausula ADD

```
ALTER TABLE tabla  
ADD [CONSTRAINT constraint_nombre] tipo (columna);
```

Sintaxis

<i>tabla</i>	<i>es el nombre de la tabla</i>
<i>columna</i>	<i>es el nombre de la columna afectada</i>
<i>tipo</i>	<i>es el tipo de constraint</i>

```
ALTER TABLE emp  
ADD CONSTRAINT emp_mgr_fk  
FOREIGN KEY(mgr) REFERENCES emp(empno);
```

Restricciones



✧ Eliminando un constraint

- ✧ Para eliminar un constraint se debe identificar el nombre con el que fue asignado, entonces utilizar la instrucción ALTER TABLE con la cláusula DROP

```
ALTER TABLE emp  
DROP CONSTRAINT emp_mgr_fk;
```

- ✧ sp_helpconstraint nombre_tabla;