Politécnico Internacional de Occidente PIO

Curso Metodologías Ágiles para la Programación

Profesor Julio Cesar Castaño

Estudiante Laura Barona Saavedra

Grupo 6

Septiembre 2024

Comandos SQL

Comandos DDL (**Data Definition Language – Lenguaje Definición de Datos**) para creación, modificación y eliminación de la estructura y objetos de la base de datos.

CREATE – Utilizado para crear objetos (tablas, vistas, índices) en la base de datos

CREATE DATABASEcoleccion;

Al crear una tabla de base de datos debemos considerar

CREATE TABLE nombre\_tabla (

nombre\_columna tipo\_columna [ cláusula\_defecto ] [ vínculos\_de\_columna ]

[ , nombre\_columna tipo\_columna [ cláusula\_defecto ] [ vínculos\_de\_columna ] … ]

[ , [ vínculo\_de tabla] … ] )

**Nombre del campo** deben empezar por un carácter alfabético y ser lo más comprensible posible para entender que datos se almacenan en ese campo.

**Tipo del campo** (lista de los más importante):

Numérico: bigint, int, smallint, tinyint, numeric, bit, decimal, money, smallmoney.

Cadena de caracteres: char, varchar, text.

Fecha y hora: date, datetime, time, timestamp.

Cadenas binarias: binary, varbinary, image.

**Cláusula defecto** se le asigna valor por defecto al campo si no se le indica en el momento que se inserta una fila. Puede iniciarse con un valor o nulo.

DEFAULT { valor | NULL }

**Vínculos de integridad** que se aplica a cada campo

* NOT NULL no permite valor NULL
* PRIMARY KEY para indicar que es la clave primaria de la tabla. Puede formarse por más de un campo, directamente debe almacenar valor único y que no sea NULL
* FOREIGN KEY indica la clave foránea haciendo referencia a otra tabla, estableciendo la relación. Tiene las cláusulas ON DELETE y ON UPDATE indican que acción debe ejecutarse en el caso que la clave foránea (a quién hace referencia) es eliminada o borrada. Las acciones pueden ser:
  + CASCADE: elimina o modifica la tupa que tiene el campo referenciado
  + SET DEFAULT: asigna valor por defecto a la columna referenciada
  + SET NULL: asigna valor NULL a la columna referenciada
* Control de valor permite asignar o no un valor a la columna dependiendo del resultado de la condición

CHECK { expresión\_condicional }

**Vínculos de integridad que se pueden aplicar a más campos de la tabla**

* Clave primaria PRIMARY KEY ( columna1 [ , columna2 … ] )
* Clave foránea FOREIGN KEY ( columna1 [ , columna2 … ] )

CREATE TABLE consola (id\_consola bigint IDENTITY(1,1),

nombre varchar (50),

tipo varchar (15) DEFAULT ‘Sobremesa’,

marca varchar (50),

PRIMARY KEY (id\_consola)

)

CREATE TABLE juego(id\_juego bigint IDENTITY(1,1),

titulo varchar(50),

genero varchar(50),

PRIMARY KEY (id\_juego)

)

CREATE TABLE consola\_juego(id\_consola bigint,

id\_juego bigint,

PRIMARY KEY (id\_consola, id\_juego),

FOREIGN KEY (id\_consola) REFERENCES consola(id\_consola) ON UPDATE CASCADE ON DELETE CASCADE,

FOREIGN KEY (id\_juego) REFERENCES juego(id\_juego) ON UPDATE CASCADE ON DELETE CASCADE

)

También podemos crear índices, se usan para encontrar un dato más rápidamente mejorando la eficacia de la consulta

CREATE INDEX index\_marca

ON consola (marca)

CREATE UNIQUE INDEX index\_nombre\_marca

ON consola (nombre, marca)

ALTER – Utilizado para modificar las tablas agregando o cambiando la definición de los campos

ALTER TABLE consola ALTER COLUMN marca char (SQL Server)

ALTER TABLE consola MODIFY COLUMN marca char (MySQL)

ALTER TABLE consola ADD fecha\_lanzamiento date

ALTER TABLE consola DROP COLUMN fecha\_lanzamiento

¿Qué pasaría si creamos el campo fecha con valores por defecto y después intentamos eliminar el propio campo?

ALTER TABLE consola ADD fecha\_lanzamiento DATE NOT NULL DEFAULT ‘2017-01-01’

ALTER TABLE consola DROP COLUMN fecha\_lanzamiento

Msg 5074, Level 16, State 1, Line 1

The object ‘DF\_\_consola\_\_fecha\_l\_\_276EDEB3’ is dependent on column ‘fecha\_lanzamiento’.

Msg 4922, Level 16, State 9, Line 1

ALTER TABLE DROP COLUMN fecha\_lanzamiento failed because one or more objects access this column.

DECLARE @ConstraintName nvarchar(MAX)

SELECT @ConstraintName = Name FROM SYS.DEFAULT\_CONSTRAINTS

WHERE PARENT\_OBJECT\_ID = OBJECT\_ID(‘consola’)

AND PARENT\_COLUMN\_ID = (SELECT column\_id FROM sys.columns

WHERE NAME = N’fecha\_lanzamiento’

AND object\_id = OBJECT\_ID(N’consola’))

IF @ConstraintName IS NOT NULL

EXEC(‘ALTER TABLE consola DROP CONSTRAINT ‘ + @ConstraintName)

ALTER TABLE consola DROP COLUMN fecha\_lanzamiento

DROP – Utilizado para eliminar objetos en la base de datos

Podemos eliminar un índice

DROP INDEX consola.index\_nombre\_marca (SQL Server)

ALTER TABLE consola DROP INDEX consola.index\_nombre\_marca (MySql)

O la tabla entera

DROP TABLE consola

**TRUNCATE** – Borra todo el contenido de una tabla

TRUNCATE TABLE consola

Msg 4712, Level 16, State 1, Line 1

Cannot truncate table ‘consola’ because it is being referenced by a FOREIGN KEY constraint.

En este caso no nos lo permitirá al haber una clave foránea que hace referencia a la tabla consola

**COMMENT** – Agregar comentarios al diccionario de datos sobre un objeto de la base de datos

COMMENT ON COLUMN consola.marca

IS ‘desarrolladora del producto’; (ORACLE)

COMMENT ON COLUMN consola.marca (ORACLE)

IS ‘ ‘;

**RENAME** – Renombrar la tabla de la base de datos

ALTER TABLE consola RENAME TO consola1 (ORACLE y MySQL)

sp\_rename ‘consola’,’consola1′; (SQL SERVER)

Caution: Changing any part of an object name could break scripts and stored procedures.

**Si modificamos el nombre, los script podrían dejar de funcionar. Hay que tener en cuenta también las relaciones que hay en la tabla, en Oracle por ejemplo si una tabla tiene una clave foránea, generará error.**

Comandos DML (**Data Manipulation Language – Lenguaje Manipulación de Datos**) para recuperar y trabajar con datos.

**SELECT** – Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado

SELECT [ ALL | DISTINCT ] lista\_elementos\_selección

FROM lista\_referencias\_tabla

[ WHERE expresión\_condicional ]

[ GROUP BY lista\_columnas ]

[ HAVING expresión\_condicional ]

[ ORDER BY lista\_columnas ]

Se puede buscar todos los valores de un campo o varios campos de una tabla, o con DISTINCT eliminar aquellas filas cuyo campo se repite. Si no se indica ALL | DISTINCT por defecto se realizará ALL. Nunca se debe usar DISTINCT en la clave primaria, ni en las candidatas ya que de por sí son únicas

WHERE se utiliza para generar una condición que debe cumplirse, aquellos campos que no la cumplan no se seleccionarán.

GROUP BY para agrupar en una sola línea el campo o campos especificados.

HAVING del resultado de GROUP BY elimina las que no cumplan su expresión condicional.

ORDER BY para ordenar el orden de las columnas de formas ascendente (ASC) o descendente (DESC). Por defecto el orden se hace de forma ascendente.

Además de la sintaxis, podemos encontrarnos con consultas que utilicen las siguientes funciones de agregado: COUNT, SUM, AVG, MAX, MIN.

Operadores lógicos: AND, OR, NOT.

Operadores de comparación: >, <, , =, =, !=, !, BETWEEN para intervalo de valores, LIKE para comparación entre cadenas de caracteres con los pattern ‘%’ y ‘\_’.

IN para especificar registros de una base de datos.

Para ver un ejemplo, vamos a ver las tablas de nuestra base de datos agregando campo STOCK a la tabla CONSOLA\_JUEGO, para saber cuántas unidades hay de cada juego y por consola.

Tabla

Descripción generada automáticamente

Consulta: Mostrar los tipos de consolas que hay registrados

SELECT DISTINCT tipo

FROM consola

Consulta: Nombre de los juegos que sólo están en 3DS

SELECT titulo

FROM juego

WHERE juego.id\_juego IN(

SELECT id\_juego

FROM consola\_juego

WHERE consola\_juego.id\_consola IN(

SELECT id\_consola

FROM consola

WHERE consola.nombre=’3DS’)

)

Consulta: Stock total de aquellos juegos cuyo género sea de Acción

SELECT SUM(consola\_juego.stock)

FROM consola\_juego

WHERE consola\_juego.id\_juego IN(

SELECT id\_juego

FROM juego

WHERE juego.genero=’accion’)

Consulta: Titulo de aquellos juegos con stock total que supere las 20 unidades

SELECT juego.titulo

FROM juego

WHERE juego.id\_juego IN(

SELECT id\_juego

FROM consola\_juego

GROUP BY id\_juego

HAVING SUM(consola\_juego.stock) > 20)