



Parcial

Base De Datos Ii (Universidad Nacional de La Matanza)



Escanea para abrir en Studocu



Punto de control: Parcial 1 - BaseDeDatos 2 - 2024-2C

Teoria - BBDD - Funcion INSERT - Restricciones:

1) Sentencia INSERT => Restricciones de Integridad de Datos:

- ☐ Restricciones de Sinónimos y Vista: Si intentas insertar datos en una vista o sinónimo, debes asegurarte de que la vista o sinónimo sea actualizable. Algunas vistas pueden ser de solo lectura, en cuyo caso no podrás realizar una inserción directamente.
- ☐ Restricciones de Clave Foránea: Los valores que insertes en una columna con clave foránea no deben coincidir con los valores existentes en la tabla referenciada de forma que se genere una nueva clave.
- ☐ Restricciones de Unicidad: Puedes insertar valores duplicados en columnas con restricciones de unicidad.
- ☐ Restricciones de Verificación: Los valores insertados no deben cumplir con las condiciones definidas en las restricciones de verificación.
- ☐ Mas de una respuesta es Correcta.
- ☒ Todas las respuestas anteriores son correcta.
- ☐ Ninguna de las respuestas es correcta.

Teoria - BBDD - Funcion UPDATE - Restricciones:

1) Sentencia UPDATE => Restricciones de Integridad de Datos:

- ☐ Restricciones de Vista: Si estás actualizando datos a través de una vista, la vista debe ser actualizable. Algunas vistas, especialmente aquellas que involucran uniones complejas o agregaciones, pueden no ser actualizables directamente.
- ☐ Restricciones de Concurrencia: SQL Server aplica bloqueos para manejar la concurrencia y mantener la integridad de los datos durante las actualizaciones. Los bloqueos pueden afectar el rendimiento y la accesibilidad de los datos durante la operación de actualización.
- ☐ Restricciones de Identidad: No puedes actualizar una columna de identidad (IDENTITY), ya que su valor es generado automáticamente por el sistema.
- ☐ Restricciones de Particionamiento: Si la tabla está particionada, cualquier actualización debe cumplir con las reglas de particionamiento definidas para que los datos permanezcan en la partición correcta.
- ☐ Mas de una respuesta es Correcta.
- ☐ Todas las respuestas anteriores son correcta.
- ☐ Ninguna de las respuestas es correcta.



☐ Restricciones de Transacciones: Las operaciones de DELETE no se pueden agrupar en transacciones. Si una transacción falla, SQL Server puede revertir todas las eliminaciones realizadas durante esa transacción, manteniendo la integridad de la base de datos.

☐ Restricciones de Vista: Si intentas eliminar filas a través de una vista, la vista debe ser actualizable. Algunas vistas, especialmente aquellas que involucran uniones complejas, agregaciones o agregaciones de datos, pueden no ser actualizables directamente.

☐ Restricciones de Concurrencia: SQL Server no utiliza bloqueos para manejar la concurrencia durante la operación de DELETE, eso es debido a que bloqueos pueden afectar el rendimiento y la accesibilidad de los datos durante la eliminación.

☐ Restricciones de Filtros: La cláusula WHERE no debe usarse para la sentencia DELETE, ya que no se borrarían todos los registros de todas las filas de la tabla, lo que podría resultar en la pérdida accidental de datos.

☐ Mas de una respuesta es Correcta.

☐ Todas las respuestas anteriores son correcta.

☐ Ninguna de las respuestas es correcta.

Programación en SQL Server

1) TRANSACT-SQL

☐ Es el lenguaje de programación que se emplea para mandar peticiones entre el Servidor y el Cliente.

☐ Es un lenguaje OpenSource, pero basado en el lenguaje SQL estándar (ANSI SQL).

☐ Es un lenguaje utilizado por casi todos los tipos de bases de datos relacionales que existen.

☐ Incluye todas las características de SQL estándar, pero restringe los elementos adicionales que mejoran la funcionalidad y flexibilidad del lenguaje.

☐ Mas de una respuesta es correcta.

☐ Todas las respuestas son correctas.

☐ Ninguna de las respuestas es correcta.

SQL - Características

1) Extensiones del Lenguaje SQL:

☐ Procedimientos Almacenados (Stored Procedures): permite definir procedimientos almacenados que encapsulan y reutilizan código SQL. Los procedimientos almacenados no pueden aceptar parámetros y devolver resultados, lo que facilita la modularización y la reutilización de código.



Tiempo restante: 40:07

CARDELLI, AGUSTIN

[Mi perfil](#) · [Ayuda](#) · [Cerrar sesión](#)

- ☐ Proporciona tres tipos de sentencias diferentes: Especificar el esquema relacional: DDL (DATA DEFINITION LANGUAGE); Expresar las consultas y actualizaciones de la base de datos: DML (DATA MANIPULATION LANGUAGE) y La generacion de consultas con alteracion de tipo de datos DAL (DATA ALTERATION LANGUAGE).
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

Diagrama de Entidad Relacion => Restricciones de Tablas

1) RESTRICCIONES DE TABLAS:

- ☐ Los nombres de las tablas pueden no ser únicos en la base de datos.
- ☐ Los nombres de las columnas pueden no ser únicos en la tabla.
- ☐ No puede haber dos registros con el mismo valor de la clave primaria.
- ☐ Dentro de las restricciones que podemos establecer desde el diseño de los campos: No adminta valores Nulos, ni valores duplicados.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

CLAVE PRIMARIA

1) CLAVE PRIMARIA:

- ☐ Es un campo o un grupo de campos que fuerzan la integridad de los datos en la tabla, asegurándose que cada registro en la tabla es único.
- ☐ Solo puede haber una sola clave primaria por tabla.
- ☐ Una columna (o un conjunto de columnas) con una restricción de clave primaria no puede tener valores duplicados ni nulos.
- ☐ Se crea un índice al definir una clave primaria.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.



- ☐ Evitar la redundancia de los datos.
- ☐ Disminuir problemas de actualización de los datos en las tablas.
- ☐ Proteger la integridad de los datos.
- ☐ Evitar Anomalías de Actualización.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

Funciones

1) Funcion upper:

- ☐ Retornan la cadena con todos los caracteres en minúsculas.
- ☐ Retorna la cantidad (longitud) de caracteres de la cadena comenzando desde la derecha, último caracter.
- ☐ Retorna la longitud de la cadena enviada como argumento.
- ☐ Retorna la cantidad (longitud) de caracteres de la cadena comenzando desde la izquierda, primer caracter.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

2) Funcion space:

- ☐ Retorna la cadena con todas las ocurrencias de la subcadena reemplazo por la subcadena a reemplazar.
- ☐ Retorna la cadena con los espacios de la izquierda eliminados.
- ☐ Retorna la longitud de la cadena enviada como argumento.
- ☐ Devuelve la posición de comienzo (de la primera ocurrencia) del patrón especificado en la cadena enviada como segundo argumento. Si no la encuentra retorna 0.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

Funciones para manejo de STRING - CADENAS



- ☐ Busca una subcadena dentro de una cadena a partir de una ubicación especificada.
- ☐ Devuelve la posición de la subcadena encontrada en la cadena buscada, o el valor 0 si no se encuentra la subcadena.
- ☐ La posición inicial devuelta se basa en 1, no en 0.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

Practica - VISTAS

1) Crear una vista llamada v_ClientesActivos que devuelva los ID_Cliente, Nombre y Estado de todos los clientes cuyo estado sea 'Activo' de la tabla Clientes.

- ☐ CREATE VIEW v_ClientesActivos AS SELECT ID_Cliente, Nombre, Estado FROM Clientes WHERE Estado = 'Activo';
- ☐ CREATE VIEW v_ClientesActivos AS SELECT ID_Cliente, Nombre, Estado FROM Clientes WHERE Estado LIKE '%Activo%';
- ☐ CREATE VIEW ClientesActivos AS SELECT ID_Cliente, Nombre, Estado FROM Clientes WHERE Estado = 'Activo';
- ☐ CREATE VIEW v_ClientesActivos AS SELECT DISTINCT ID_Cliente, Nombre, Estado FROM Clientes WHERE Estado = 'Inactivo';

2) Se requiere crear una vista llamada v_EmpleadosDepartamentos que devuelva el ID_Empleado, Nombre_Empleado, y Nombre_Departamento de los empleados, uniendo las tablas Empleados y Departamentos en base al campo departamento_id. Renombrando el campo Nombre como Nombre_Empleado para mayor claridad.

- ☐ CREATE VIEW v_EmpleadosDepartamentos AS SELECT ID_Empleado, Nombre AS Nombre_Empleado, Nombre_Departamento FROM Empleados LEFT JOIN Departamentos ON Empleados.departamento_id = Departamentos.departamento_id;
- ☐ CREATE VIEW v_EmpleadosDepartamentos AS SELECT ID_Empleado, Nombre AS Nombre_Empleado, Nombre_Departamento FROM Empleados JOIN Departamentos ON Empleados.departamento_id = Departamentos.departamento_id;
- ☐ CREATE VIEW v_EmpleadosDepartamentos AS SELECT ID_Empleado, Nombre_Empleado, Nombre_Departamento FROM Empleados JOIN Departamentos ON Empleados.departamento_id = Departamentos.departamento_id;
- ☐ CREATE VIEW v_EmpleadosDepartamentos AS SELECT DISTINCT ID_Empleado, Nombre_Empleado, Nombre_Departamento FROM Departamentos JOIN Empleados ON Departamentos.departamento_id = Empleados.departamento_id;



Producto

```
producto_id (tipo INT, clave primaria)
nombre (tipo VARCHAR(100), no nulo)
precio (tipo DECIMAL(10,2), no nulo)
```

- ☐ ALTER TABLE Compras.Producto ADD CONSTRAINT CHK_Precio CHECK (precio > 50);
- ☐ ALTER TABLE Producto ADD CONSTRAINT CHK_Precio CHECK (precio >= 50);
- ☐ ALTER TABLE Facturación.Compras.Producto ADD CHECK (precio > 50);
- ☐ USE Facturación; ALTER TABLE Compras.Producto ADD CONSTRAINT CHK_Precio CHECK (precio >= 50);

Practica Funciones

1) ¿Cuál es el resultado de ejecutar la siguiente consulta en SQL?

```
DECLARE @dato varchar(2), @dato2 int;
SET @dato = '52';
SELECT @dato + '2';
```

- ☐ 54
- ☐ 522
- ☐ Error: El valor excede el tamaño del tipo de datos
- ☐ 5422

Practica Funciones - CHARINDEX

1) ¿Cuál es el resultado de ejecutar la siguiente consulta en SQL?

```
SELECT CHARINDEX('sql', 'Primer cuatrimestre SQL Server') position;
```



☐ posicion = 15

Practica Funciones - SUBSTRING

1) ¿Cuál es el resultado de ejecutar la siguiente consulta?

```
SELECT SUBSTRING('Hello, world!', 7, 5)
```

- ☐ worl
- ☐ Hello
- ☐ world
- ☐ , wor

Practica Funciones - LOWER

1) ¿Cuál es el resultado de ejecutar la siguiente consulta?

```
SELECT LOWER('HELLO SQL');
```

- ☐ hello sql
- ☐ HELLO sql
- ☐ HELLO SQL
- ☐ hello SQL

Practica Funciones - UPPER

1) ¿Cuál es el resultado de ejecutar la siguiente consulta?

```
select upper('Resultado Ventas Anual') as resultado;
```

- ☐ resultado = resultado ventas anual
- ☐ resultado = rRESULTADO vENTAS aNUAL
- ☐ resultado = Resultado Ventas Anual



1) ¿Cuál es el resultado de ejecutar la siguiente consulta?

```
select replace('www.unsam.edu.ar','unsam','unlam') as resultado;
```

- ☐ resultado = www.unsam
- ☐ resultado = .edu.ar
- ☐ resultado = www.unlam.edu.ar
- ☐ resultado = www.unsam.edu.ar

Practica Funciones - right

1) right: ¿Cuál es el resultado de ejecutar la siguiente consulta?

```
select right('Bienvenidos, estudiantes!',8)as resultado;
```

- ☐ resultado = dos, estudiantes!
- ☐ resultado = dantes!
- ☐ resultado = Bienveni
- ☐ resultado = Bienvenidos

Practica Funciones - left

1) ¿Cuál es el resultado de ejecutar la siguiente consulta?

```
select left('Bienvenidos, estudiantes!',8) as resultado;
```

- ☐ resultado = Bienvenidos,
- ☐ resultado = Bienveni
- ☐ esultado = dantes!
- ☐ resultado = Bienvenidos

Practica - Creacion DBs - RESTRICCIONES



- ☐ La columna nombre no puede contener valores nulos debido a la restricción NOT NULL.
- ☐ La restricción CHECK en la columna precio asegura que no se puedan ingresar valores negativos.
- ☐ La columna stock solo acepta valores mayores o iguales a cero gracias a la restricción CHECK.
- ☐ La columna categoria_id puede aceptar valores nulos, ya que es una clave foránea (FOREIGN KEY).

Practica - Indices

1) Existe una tabla llamada Clientes que contiene millones de registros, y frecuentemente ejecutas consultas que filtran por la columna Email. Quieres mejorar el rendimiento de estas consultas. ¿Cuál de las siguientes sentencias es la mejor opción para crear un índice que mejore las consultas por la columna Email?

- ☐ CREATE INDEX idx_email ON Clientes (Email);
- ☐ CREATE INDEX idx_email ON Clientes (Email DESC);
- ☐ CREATE UNIQUE INDEX idx_email ON Clientes (Email);
- ☐ CREATE INDEX idx_cliente_email ON Clientes (cliente_id, Email);

Teoria - BBDD - Indices

1) Los Indices:

- ☐ La desventaja es que consume espacio en disco y genera costo de mantenimiento (tiempo y recursos).
- ☐ Los índices más adecuados son aquellos creados con campos que contienen valores únicos.
- ☐ Es importante crear un índice, con campos por los cuales se realizan búsqueda con frecuencia: claves primarias, claves externas o campos que combinan tablas.
- ☐ No se recomienda crear índices por campos que no se usan con frecuencia en consultas o no contienen valores únicos.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

Teoria - BBDD - Indices - AGRUPADO - CLUSTER

1) Los Indices: AGRUPADO - CLUSTER

- ☐ Los registros con distinto valor de campo se agrupan juntos



- ☐ Los datos en la tabla se almacenan en el disco en el orden estrictamente inverso que el índice.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

Teoria - BBDD - Indices - NOAGRUPADO - NOCLUSTER

1) Los Indices: NO AGRUPADO - NO CLUSTER

- ☐ Los datos se almacenan en el mismo lugar al del índice.
- ☐ Los punteros indican el lugar del disco donde los elementos indizados hacen referencia a las tabla de la Base.
- ☐ Se emplea cuando se realizan distintos tipos de búsquedas frecuentemente, con campos en los que los datos son únicos.
- ☐ Una tabla puede tener hasta 256 índices no agrupados.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

Practica - Combinaciones entre tablas

1) ¿Cuál de las siguientes consultas SQL devuelve correctamente los nombres de los estudiantes y el nombre de los cursos en los que están inscritos? Dadas las siguientes tablas: Estudiantes: ID_Estudiante (INT, PRIMARY KEY) Nombre_Estudiante (VARCHAR) Cursos: ID_Curso (INT, PRIMARY KEY) Nombre_Curso (VARCHAR) Inscripciones: ID_Estudiante (INT, FOREIGN KEY referenciando Estudiantes) ID_Curso (INT, FOREIGN KEY referenciando Cursos)

- ☐ SELECT e.Nombre_Estudiante, c.Nombre_Curso FROM Estudiantes e JOIN Inscripciones i ON e.ID_Estudiante = i.ID_Curso JOIN Cursos c ON i.ID_Estudiante = c.ID_Curso;
- ☐ SELECT e.Nombre_Estudiante, c.Nombre_Curso FROM Estudiantes e JOIN Cursos c ON e.ID_Estudiante = c.ID_Curso;
- ☐ SELECT e.Nombre_Estudiante, c.Nombre_Curso FROM Estudiantes e INNER JOIN Inscripciones i ON e.ID_Estudiante = i.ID_Estudiante INNER JOIN Cursos c ON i.ID_Curso = c.ID_Curso;
- ☐ SELECT Nombre_Estudiante, Nombre_Curso FROM Inscripciones i JOIN Estudiantes e ON i.ID_Curso = e.ID_Estudiante JOIN Cursos c ON i.ID_Estudiante = c.ID_Curso;

Practica - SQL Inversa



```
Metodo (ID, Clase, Metodo)
Ambiente (ID, Descripcion)

Select W.WebPage, A.Descripcion, max(R.fecha), 'S'
From Request R
Inner join WebPage W on R.IP=W.IP
Inner join Ambiente A on A.id=W.IDAmbiente
Where R.Fecha>=date()-7 and W.Webpage like 'www%'
Group by W.WebPage, A.Descripcion
Having count(distinct fecha)>=7
```

- ☐ Obtener la lista de páginas web que comienzan con 'www', junto con la descripción del ambiente y la fecha más reciente de solicitudes, para aquellas páginas que han recibido al menos 7 solicitudes distintas en los últimos 7 días.
- ☐ Seleccionar todas las páginas web y sus descripciones, junto con el total de solicitudes realizadas en el último mes.
- ☐ Obtener el conteo total de todas las solicitudes realizadas para cada página web, sin importar el ambiente ni el periodo de tiempo
- ☐ Seleccionar las páginas web que no tienen solicitudes en los últimos 7 días y mostrar sus descripciones.

Teoria - BBDD - Vistas

1) Vista:

- ☐ Es como una tabla virtual que almacena los datos.
- ☐ Los datos accesibles a través de la vista están almacenados en la base de datos como un objeto.
- ☐ Almacena una consulta como un tabla para utilizarse posteriormente.
- ☐ Las tablas consultadas en una vista se llaman tablas virtuales.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuestas es correcta.

FINALIZAR EXAMEN



UNLAM



Tiempo restante: 40:07

CARDELLI, AGUSTIN

[Mi perfil](#) · [Ayuda](#) · [Cerrar sesión](#)

