



Parcial 1 - Base De Datos 2 - 2024-2C

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 Clave Primaria: No puedes insertar valores duplicados en una columna que esté definida como clave primaria.
- ☐ 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: No puedes insertar valores duplicados en columnas con restricciones de unicidad.
- ☐ Restricciones de Verificación: Los valores insertados 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:

- ☐ Claves Primarias: Puedes actualizar una clave primaria de manera que se convierta en un valor duplicado.
- ☐ Claves Foráneas: Los valores actualizados en una columna de clave foránea no deben seguir existiendo en la tabla referenciada, a menos que se actualicen las referencias en las tablas relacionadas.
- ☐ Restricciones de Unicidad: Debes actualizar valores en una columna que tengan una restricción de unicidad a fin de que se conviertan en valores duplicados.
- ☐ Restricciones de Verificación: Los nuevos valores 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 DELETE - Restricciones:

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

- ☐ Claves Foráneas: Si una fila que deseas eliminar no está referenciada por una clave foránea en otra tabla, SQL Server puede impedir la eliminación para evitar la violación de la integridad referencial.
- ☐ Claves Foráneas: Si una fila que deseas eliminar está referenciada por una clave foránea en otra tabla, si se han definido acciones ON DELETE CASCADE, la eliminación se propagará automáticamente a las filas dependientes.
- ☐ Restricciones de Integridad Referencial: Las restricciones de clave foránea pueden requerir que manejes cuidadosamente la eliminación de filas para mantener la integridad de los datos.
- ☒ Mas de una respuesta es Correcta.
- ☐ Todas las respuestas anteriores son correcta.
- ☐ Ninguna de las respuestas es correcta.



- ☐ 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 añade elementos adicionales que mejoran la funcionalidad y flexibilidad del lenguaje.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta 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.
- ☐ Funciones Definidas por el Usuario (UDFs): Puedes definir funciones personalizadas que pueden ser utilizadas en consultas, pero no para realizar cálculos complejos o transformaciones de datos.
- ☐ Triggers: T-SQL soporta triggers que son procedimientos que se ejecutan unicamente de forma manual por el DBA ante eventos específicos, como inserciones, actualizaciones o eliminaciones en una tabla.
- ☐ 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 consecutivo y se pueden agrupar por ser iguales.
- ☐ Puede haber una o dos claves primarias por tabla.



- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

Normalizacion de Base de Datos

1) Las BBDDs se Normalizan con el proposito de:

- ☐ Asegurar la redundancia de los datos.
- ☐ Aunmentar los volúmenes de datos 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 len:

- ☐ Convierte caracteres a numeros.
- ☐ Retorna un caracter en código ASCII del entero enviado como argumento.
- ☐ El segundo y tercer argumento son opcionales y deben ser positivos.
- ☒ 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 de Escaleres:

- ☒ Operan sobre un conjunto de valores y devuelven un único valor como resultado de una consulta.
- ☐ Operan sobre un conjunto de valores y devuelven un único valor (por ejemplo, SUM(), AVG(), MAX(), MIN()).
- ☐ Toman un solo valor y retornan una lista de valores.
- ☐ Devuelven un solo valor (por ejemplo, LEN(), GETDATE(), UPPER()).
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuesta es correcta.

Funciones para manejo de STRING - CADENAS

1) Funcion CHARINDEX:

- ☐ Devuelve una parte de la cadena especificada como primer argumento, empezando desde la posición especificada por el segundo argumento y de tantos caracteres de longitud como indica el tercer argumento.



- ☐ 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_ProductosMasVendidos que muestre el ID_Producto, Nombre_Producto, y el Total_Vendido (la suma de la cantidad de ventas por producto) para los productos que hayan sido vendidos más de 100 veces. La tabla Ventas contiene las columnas ID_Producto y Cantidad_Vendida, mientras que la tabla Productos contiene las columnas ID_Producto y Nombre_Producto.

- ☐ CREATE VIEW v_ProductosMasVendidos AS SELECT P.ID_Producto, P.Nombre_Producto, COUNT(V.ID_Producto) AS Total_Vendido FROM Productos P INNER JOIN Ventas V ON P.ID_Producto = V.ID_Producto GROUP BY P.ID_Producto, P.Nombre_Producto WHERE COUNT(V.ID_Producto) > 100;
- ☒ CREATE VIEW v_ProductosMasVendidos AS SELECT P.ID_Producto, P.Nombre_Producto, SUM(V.Cantidad_Vendida) AS Total_Vendido FROM Productos P JOIN Ventas V ON P.ID_Producto = V.ID_Producto GROUP BY P.ID_Producto HAVING Total_Vendido > 100;
- ☐ CREATE VIEW v_ProductosMasVendidos AS SELECT P.ID_Producto, P.Nombre_Producto, SUM(V.Cantidad_Vendida) AS Total_Vendido FROM Productos P LEFT JOIN Ventas V ON P.ID_Producto = V.ID_Producto GROUP BY P.ID_Producto, P.Nombre_Producto WHERE SUM(V.Cantidad_Vendida) > 100;
- ☐ CREATE VIEW v_ProductosMasVendidos AS SELECT P.ID_Producto, P.Nombre_Producto, SUM(V.Cantidad_Vendida) AS Total_Vendido FROM Productos P INNER JOIN Ventas V ON P.ID_Producto = V.ID_Producto GROUP BY P.ID_Producto, P.Nombre_Producto HAVING SUM(V.Cantidad_Vendida) > 100;

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;

Practica - Creacion DBs

1) Dado el siguiente requerimiento, crear la base de datos y su esquema: La base de datos debe llamarse Facturación. El esquema dentro de la base de datos debe llamarse Compras.

- ☐ CREATE SCHEMA Compras; CREATE DATABASE Facturación;
- ☐ CREATE DATABASE Facturación; CREATE SCHEMA Compras;
- ☒ CREATE DATABASE Facturación; USE Facturación; CREATE SCHEMA Compras;
- ☐ USE Facturación; CREATE SCHEMA Compras; CREATE DATABASE Facturación;

Practica Funciones

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



- ☐ 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', 'SQL Server CHARINDEX Estrada Web Group') as position;
```

- ☒ posicion = 0
- ☐ no devuelve valor CHARINDEX es sensible a mayúsculas y minúsculas.
- ☐ posicion = 1
- ☐ posicion = 3

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, World!')
```

- ☒ hello, world!
- ☐ Hello, World!
- ☐ HELLO, WORLD!
- ☐ hELLO, wORLD!



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

- ☐ resultado = resultado ventas anual
- ☐ resultado = rRESULTADO vENTAS aNUAL
- ☐ resultado = Resultado Ventas Anual
- ☒ resultado = RESULTADO VENTAS ANUAL

Practica Funciones - REPLACE

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

```
select right('Bienvenidos a SQL Server',8) as resultado;
```

- ☐ resultado = Bienvenidos a SQ
- ☒ resultado = L Server
- ☐ resultado = Bienveni
- ☐ resultado = Bienvenidos

Practica Funciones - left

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

```
select left('Bienvenidos a SQL Server',8) as resultado;
```

- ☐ resultado = Bienvenidos a
- ☒ resultado = Bienveni
- ☐ resultado = L Server
- ☐ resultado = Bienvenidos



KEY (categoria_id) REFERENCES categorias(categoria_id));

- ☐ 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: SQL Server accede a los datos de las siguientes maneras:

- ☐ Recorriendo las tablas; comenzando desde el principio y extrayendo los registros que cumplen las condiciones de la consulta.
- ☐ Recorriendo la estructura de forma secuencial del índice para localizar los registros y extrayendo los que cumplen las condiciones de la consulta.
- ☐ Empleando índices; recorriendo la estructura de árbol del índice para localizar los registros y extrayendo los que cumplen las condiciones de la consulta.
- ☒ Recorriendo las tablas; comenzando por el valor del índice mas grande y extrayendo los registros que cumplen las condiciones de la consulta.
- ☐ 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.
- ☐ El tamaño medio de un índice agrupado es aproximadamente el 10% del tamaño de la tabla mas grande de la Base.
- ☐ Un índice clustered define el orden en el cual los resultados de las consultas son procesadas con los datos en la tabla.
- ☐ Los datos en la tabla se almacenan en el disco en el mismo orden 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



- ☐ Se emplea cuando se realizan el mismo tipos de búsquedas de forma frecuentemente, con campos en los que los datos no 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) ¿Qué resultado devuelve la siguiente consulta SQL?

```
SELECT DISTINCT ciudad  
FROM clientes;
```

- ☐ Devuelve todas las ciudades, incluyendo los duplicados.
- ☒ Devuelve las ciudades únicas, eliminando los duplicados.
- ☐ Devuelve todas las ciudades que aparecen más de una vez en la tabla.
- ☐ Devuelve solo las ciudades que están asociadas con más de un cliente.

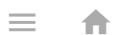
Practica - SQL Inversa

1) En base a la consulta SQL, determinar que enunciado dio origen a la consulta siguiente:

```
Dado el siguiente esquema:  
Request (NoRequest, IP, Fecha, Hora, IDMetodo)  
Page (IP, WebPage, IDAmbiente)  
Metodo (ID, Clase, Metodo)  
Ambiente (ID, Descripcion)  
  
insert into Page  
select IP, 'Web ' + IDMetodo, '?'  
from request R  
where not exists (Select 1  
from Page P  
where R.IP=P.IP  
)  
and IDMetodo in (select ID from Metodo)  
and fecha>=date()-30
```

- ☐ Actualizar las entradas existentes en la tabla Page con nuevas direcciones IP y métodos de la tabla Request.
- ☐ Eliminar las entradas de la tabla Request que no tienen un método válido o que son más antiguas de 30 días.
- ☐ Seleccionar todas las solicitudes de los últimos 30 días y mostrarlas en la tabla Page, incluyendo todas las direcciones IP y métodos.
- ☒ Insertar nuevas entradas en la tabla Page para direcciones IP que no existen en dicha tabla, utilizando la dirección IP de la tabla Request, concatenando 'Web ' con el ID del método, y asegurando que el ID del método esté presente en la tabla Metodo, considerando solo las solicitudes de los últimos 30 días.

Teoria - BBDD - Vistas



- ☐ Las tablas consultadas en una vista se llaman tablas base.
- ☒ Es una estructura en la base de datos que almacena una datos de las tablas en la Base.
- ☐ Mas de una respuesta es correcta.
- ☐ Todas las respuestas son correctas.
- ☐ Ninguna de las respuestas es correcta.

FINALIZAR EXAMEN