

SQL - PRÁCTICA

- 1) Dadas las siguientes afirmaciones, indique cual es Verdadera
- a) El valor en un atributo que se encuentra definido como clave primaria de una tabla nunca puede ser modificado desde un UPDATE.
 - b) Para crear una vista debo utilizar una sentencia DML
 - c) No es posible insertar tuplas en una tabla con el resultado de un SELECT
 - d) No es posible, con un Stored Procedure, insertar filas en más de una tabla.
 - e) Ninguna de las opciones es correcta
- 2) ¿Cuál de las siguientes sentencias SQL se debe utilizar si desea modificar la tabla denominada "tabla1" de forma de agregar una nueva columna llamada "descripcion" de tipo varchar(300)?
- a) ALTER TABLE tabla1 ADD NEW COLUMN descripcion varchar(300)
 - b) ALTER TABLE tabla1 ADD descripcion varchar(300)
 - c) MODIFY TABLE tabla1 NEW COLUMN descripcion AS varchar(300)
 - d) Todas las opciones son correctas
 - e) Ninguna de las opciones es correcta

- 3) Dada la siguiente tabla: Empleado (legajo int, sueldo float) y Asignado (legajo_emp int, proyecto char(1), fecha date) y la siguiente función:

```
CREATE FUNCTION f_parcial2 (@sueldo float) RETURNS int
BEGIN
    DECLARE @cant int
    SET @cant = (SELECT count(*) FROM Empleado WHERE sueldo > @sueldo
    and NOT EXISTS (SELECT 1 FROM Asignado WHERE Asignado.legajo_emp = Empleado.legajo)
    )
    RETURN @cant
END
```

Indicar la opción correcta:

- a) La función tiene un error, ya que no se permite usar NOT EXISTS dentro de las mismas.
 - b) La función retorna la cantidad de empleados cuyo sueldo supera al enviado por parámetro y que, a su vez, no se encuentran asignados a ningún proyecto
 - c) La función retorna la cantidad de empleados asignados al menos a un proyecto y cuyo sueldo supera al enviado por parámetro.
 - d) La función creada debe invocarse desde el FROM de una consulta.
 - e) Ninguna de las opciones es correcta
- 4) Indique cuál de las siguientes sentencias SQL permite insertar filas en una tabla denominada "tabla1"? Nota: "tabla1" tiene la siguiente estructura (**id** int primary key, **descripcion** varchar(10)). "tabla2" tiene la misma estructura que "tabla1"
- a) insert into tabla1 values (1,'uno'),(2,'dos')
 - b) insert into tabla1 (id, descripcion) values (3,'tres')
 - c) insert into tabla1 select * from tabla2
 - d) Todas las opciones son correctas
 - e) Ninguna de las opciones es correcta

- 5) Dadas las siguientes tablas: Empleado (legajo int, sueldo float) y Asignado (legajo_emp int, proyecto char(1), fecha date) y el siguiente Trigger:

```
CREATE TRIGGER parcial2 ON Empleado INSTEAD OF UPDATE
AS
BEGIN
    UPDATE Empleado SET sueldo = sueldo * 1.20
    WHERE Empleado.legajo IN (SELECT legajo FROM inserted)
    AND NOT EXISTS
        (
            SELECT 1
            FROM Asignado
            WHERE Asignado.Legajo_emp = Empleado.legajo AND Asignado.Proyecto IN ('A','B','C')
        )
END
```

Seleccione la respuesta correcta

- a) El trigger permite actualizarles el sueldo solo a aquellos empleados que se encuentran asignados al proyecto A, B y C (Todos ellos)
- b) El trigger daría error porque en la consulta usada dentro del NOT EXISTS se hace referencia a la tabla "Empleado", la cual no puede ser invocada desde allí.
- c) El trigger aumenta el sueldo en un 20% para aquellos empleados que se están intentando modificar con el UPDATE que disparó el trigger y que, además, no se encuentran asignados al proyecto A, B ó C (Al menos uno de ellos)
- d) Al ser de tipo INSTEAD OF, finalmente, no se termina actualizando el sueldo de ningún empleado. Para que realmente funcione debería ser del tipo AFTER.
- e) Ninguna de las opciones es correcta

6) Dada la siguiente consulta SQL, indique su correspondiente enunciado:

EMPLEADO (legajo, nombre, apellido, salario)
CARGO (cod_cargo, descripcion)
ASIGNADO_A (legajo, cod_cargo, fecha_desde, fecha_hasta)

```
CREATE VIEW Vista1 AS
SELECT a.legajo
FROM Asignado_a a, Cargo c
WHERE a.cod_cargo = c.cod_cargo
AND c.descripcion = 'Programador Senior'
AND a.fecha_hasta IS NULL
```

```
CREATE VIEW Vista2 AS
SELECT DISTINCT a.legajo, a.cargo
FROM Vista1 v1, Asignado_a a
WHERE v1.legajo = a.legajo
```

```
SELECT legajo, count(*)
FROM Vista2
GROUP BY legajo
HAVING COUNT(*) = 3
```

- a) Listar los Empleados que fueron asignados a tres cargos distintos y que el primero de ellos fue Programador Senior.
- b) Listar los Empleados que actualmente tienen el cargo de Programador Senior y que previamente ocuparon otros 2 cargos.
- c) Listar los Empleados que actualmente tienen el cargo de Programador Senior y que previamente ocuparon otros 3 cargos.
- d) Listar los Empleados que actualmente tienen el cargo de Programador Senior y que hace 3 años que ocupan ese puesto.
- e) Ninguna de las anteriores

7) Dada la siguiente consulta SQL, indique su correspondiente enunciado:

ALUMNO (legajo, nombre, apellido)
MATERIA (cod_mat, nombre, cod_carrera)
FINAL (cod_mat, legajo, fecha, nota)

```
SELECT a.legajo
FROM Alumno a
WHERE NOT EXISTS ( SELECT *
                  FROM Final f
                  WHERE f.legajo = a.legajo
                  AND NOT EXISTS ( SELECT *
                                FROM Materia m
                                WHERE f.cod_mat = m.cod_mat
                                AND m.cod_carrera = 201 ))
```

- a) Listar los Alumnos que solamente rindieron Final de materias de la carrera 201 y nunca rindieron final de alguna materia de otra carrera.
- b) Listar los alumnos que rindieron final de todas las materias de la carrera 201.
- c) Listar los alumnos que aprobaron todas las materias de la carrera 201.
- d) Listar los alumnos que nunca rindieron un final de alguna materia de la carrera 201.
- e) Ninguna de las anteriores

8) Dadas las siguientes dos tablas, para esas determinadas filas, hay 4 consultas que devuelven el mismo resultado y una consulta que devuelve un resultado distinto. Indique cuál de ellas es la distinta.

Tabla1

A	B
1	10
2	20
3	30
4	40

Tabla2

C	D
20	200
20	201
30	300
50	500

- SELECT * FROM Tabla1, Tabla2 WHERE B = C AND A > 1
- SELECT * FROM Tabla1, Tabla2 WHERE B = C
- SELECT * FROM Tabla1 INNER JOIN Tabla2 ON (B = C)
- SELECT * FROM Tabla1 LEFT JOIN Tabla2 ON (B = C) WHERE A > 1
- SELECT * FROM Tabla1 LEFT JOIN Tabla2 ON (B = C) WHERE A NOT IN (1,4)

SQL - TEORÍA

- ¿Indique cuál de las siguientes NO es una sentencia DML de SQL?
 - delete
 - update
 - alter
 - insert
 - select
- Las sentencias SQL pueden clasificarse en:
 - DCL, DRL y DDL
 - DLL, DML y DCL
 - DML, DDL y DCL
 - Solo b y c son correctas
 - Ninguna de las opciones es correcta
- Indique cuál de las siguientes opciones es una desventaja de los Stored Procedures
 - Permiten la reutilización de código
 - Aumentan considerablemente el tráfico de datos en la red
 - Nos obligan a enviarles al menos un parámetro, cuando muchas veces no es necesario hacerlo
 - Todas son correctas
 - Ninguna es correcta
- Indique cuál de las siguientes afirmaciones sobre las funciones, no es válida:
 - En el caso de las funciones con valores de tabla en línea de base, obligatoriamente, definir una tabla y cargarla con datos para finalmente retornarla como resultado.
 - No es obligatorio enviarles parámetros
 - Existen 2 tipos de funciones: Las que retornan un escalar y las que retornan una tabla
 - Todas son correctas
 - Ninguna es correcta
- Indique cuál de las siguientes afirmaciones es FALSA respecto a los triggers:
 - No pueden existir 2 triggers de tipo INSTEAD OF sobre una determinada tabla y para un mismo evento.
 - Solo pueden existir muchos triggers de tipo FOR/AFTER sobre una tabla, siempre y cuando cada uno de ellos sea definido para diferentes eventos.
 - Las tablas INSERTED y DELETED son tablas virtuales que se pueden utilizar dentro de un Trigger
 - Todas son FALSAS
 - Ninguna es FALSA
- Indique cuál de las siguientes afirmaciones es Falsa:
 - En el ORDER BY se puede usar el Alias de las columnas. Por ejemplo:
SELECT salario AS sueldo FROM Empleado ORDER BY sueldo
 - Todas las columnas del SELECT que no sean funciones agregadas deben colocarse en el GROUP BY.
Por ejemplo, si ejecutamos la siguiente sentencia daría error:
SELECT apellido, categoria, count(*) FROM Empleado GROUP BY apellido
 - Todas las columnas que se colocan en el GROUP BY deben colocarse en el SELECT
Por ejemplo, si ejecutamos la siguiente sentencia daría error:
SELECT apellido, count(*) FROM Empleado GROUP BY apellido, categoria
 - Es posible colocar una columna en el ORDER BY que no esté en el SELECT.
Por ejemplo: SELECT apellido FROM Empleado ORDER BY legajo

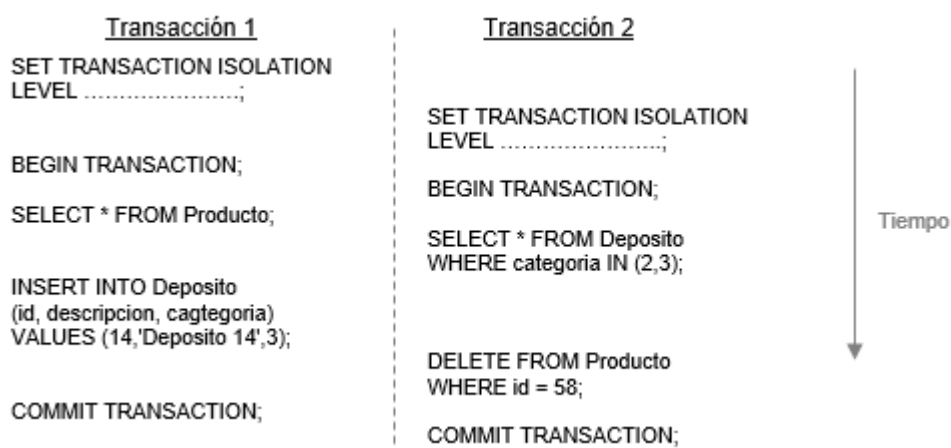
- e) Es posible colocar una subconsulta en el FROM, siempre y cuando se le coloque un alias.
 Por ejemplo: `SELECT * FROM (SELECT legajo, apellido FROM Empleado WHERE cod_depto = 3) a`

15) Indique cuál de las siguientes afirmaciones es Verdadera:

- a) No es posible usar la cláusula HAVING en una consulta que no tenga GROUP BY.
- b) No es posible renombrar las columnas cuando se crea una Vista.
- c) No es posible utilizar una Vista dentro de otra Vista.
- d) No es posible hacer un UPDATE sobre ninguna Vista para modificar los datos.
- e) No es posible usar una función de agregación en el SELECT (por ejemplo, SUM()) si la consulta no tiene un GROUP BY.

TRANSACCIONES - PRÁCTICA

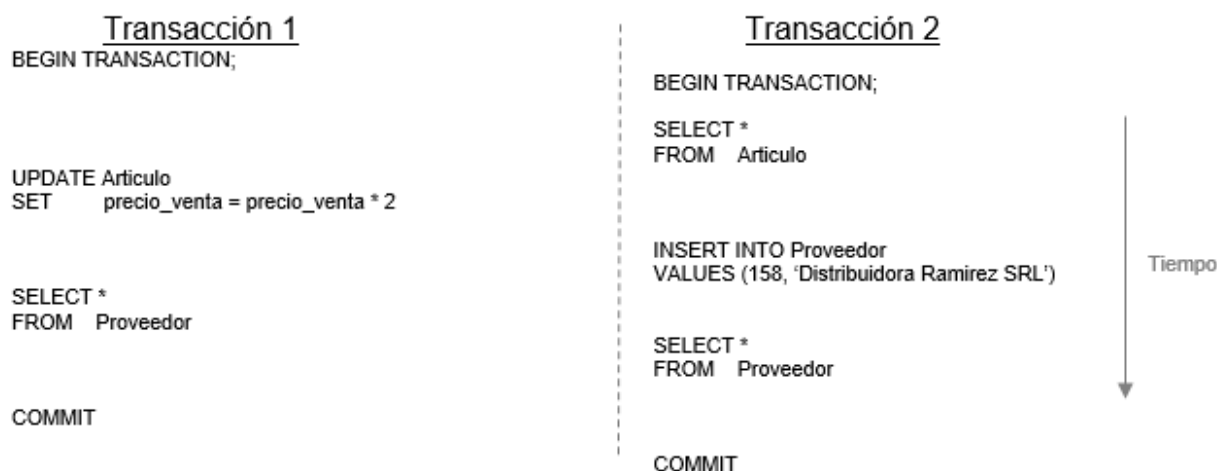
16) Dadas las siguientes dos transacciones concurrentes:



¿Con cuál de los siguientes niveles de aislamiento se generaría Deadlock?

- | | |
|--------------------------|-----------------------|
| a) T1 = Read Uncommitted | T2 = Read Uncommitted |
| b) T1 = Repeatable Read | T2 = Repeatable Read |
| c) T1 = Serializable | T2 = Repeatable Read |
| d) T1 = Repeatable Read | T2 = Serializable |
| e) T1 = Read Committed | T2 = Read Committed. |

17) Dadas las siguientes dos transacciones concurrentes (en el nivel de aislamiento por defecto):



¿Cuál de las siguientes afirmaciones es verdadera?

- a) La Transacción 1 finaliza antes que la Transacción 2.
- b) La Transacción 2 finaliza antes que la Transacción 1 ya que la Transacción 1 queda bloqueada esperando a que la Transacción 2 finalice.
- c) Ambas transacciones finalizan exactamente al mismo tiempo.
- d) Ninguna de las transacciones logra finalizar porque quedan bloqueadas mutuamente.

e) Ninguna de las anteriores afirmaciones es verdadera.

TRANSACCIONES - TEORÍA

18) Existen cuatro propiedades deseables que todo motor de base de datos debería cumplir con las Transacciones, cual de las siguientes no es una de ellas?

- a) Conservación de la Consistencia
- b) Seguridad de la información
- c) Durabilidad
- d) Atomicidad
- e) Aislamiento

19) El nivel de aislamiento Read Committed:

- a) Asegura que no tendremos "lecturas sucias".
- b) Asegura que no tendremos "lecturas fantasmas".
- c) Asegura que no tendremos deadlocks.
- d) Todas las anteriores.
- e) Ninguna de las anteriores

20) El LOG de transacciones de una base de datos tiene:

- a) El registro de todos los cambios confirmados
- b) El registro de todos los cambios no confirmados
- c) El registro de todas las sentencias SQL ejecutadas (ya sean cambios o consultas)
- d) El registro de todos los cambios confirmados y no confirmados
- e) Ninguna de las anteriores.

RESPUESTAS

	A	B	C	D	E
1					X
2		X			
3		X			
4				X	
5			X		
6		X			
7	X				
8				X	
9			X		
10			X		
11					X
12	X				
13		X			
14			X		
15	X				
16				X	
17		X			
18		X			
19	X				
20				X	