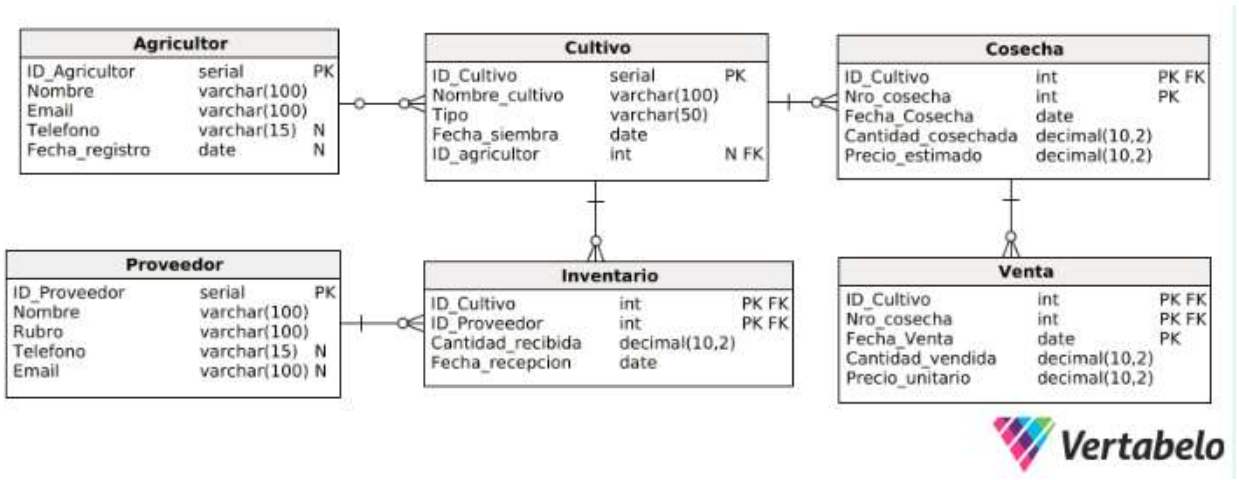
Comenzado el	jueves, 5 de diciembre de 2024, 09:35
Estado	Finalizado
Finalizado en	jueves, 5 de diciembre de 2024, 12:34
Tiempo empleado	2 horas 59 minutos

Información

El siguiente esquema de base de datos (link) forma parte de un sistema de gestión de información relacionada con la agricultura, abarcando desde la siembra hasta la venta de cultivos. Las tablas incluyen detalles sobre los agricultores (con sus datos personales y fecha de registro en el sistema), sobre los cultivos sembrados por cada agricultor (incluyendo tipo de cultivo, nombre y fecha de siembra), sobre las cosechas resultantes de cada cultivo (detallando la fecha de cosecha, la cantidad cosechada y el precio estimado) y sobre las ventas efectuadas para cada cosecha (incluyendo la cantidad vendida, el precio de venta por unidad y la fecha de venta). También se registran los proveedores de insumos agrícolas, así como el inventario con las entregas recibidas de parte de los proveedores para cultivos específicos (indicando la cantidad recibida y la fecha de recepción).



Desarrolle los ejercicios indicados a continuación.

Puede utilizar el servidor de la base vía PhpPgAdmin y consultar la documentación de PostgreSQL en línea (únicos recursos a disposición durante el examen)

Pregunta **1**

Correcta

Se puntúa como 0 sobre 1,00

1.a) En el esquema dado se requiere incorporar la siguiente restricción según SQL estándar utilizando el recurso declarativo más restrictivo posible (a nivel de atributo, de tupla, de tabla o general) y utilizando sólo las tablas/atributos necesarios.

- *Controlar que los proveedores tengan registrado un teléfono o un email.*

Seleccione la opción que considera correcta, de acuerdo a lo solicitado y justifique claramente (debajo de la pregunta 1.c)

a. ALTER TABLE Proveedor ADD CONSTRAINT chk_contacto
CHECK (Telefono IS NOT NULL OR Email IS NOT NULL);

b. ALTER TABLE Proveedor ADD CONSTRAINT chk_contacto
CHECK (Telefono IS NULL OR Email IS NULL);

c. ALTER TABLE Proveedor ADD CONSTRAINT chk_contacto
CHECK (NOT EXISTS (SELECT 1 from PROVEEDOR
WHERE Telefono IS NOT NULL OR Email IS NOT NULL));

d. ALTER TABLE Proveedor ADD CONSTRAINT chk_contacto
CHECK (Telefono IS NULL AND Email IS NULL);

e. Ninguna de las opciones

f. ALTER TABLE Proveedor ADD CONSTRAINT chk_contacto
CHECK (Telefono IS NOT NULL AND Email IS NOT NULL);

g. ALTER TABLE Proveedor ADD CONSTRAINT chk_contacto
CHECK (NOT EXISTS (SELECT 1 from PROVEEDOR
WHERE Telefono IS NULL AND Email IS NULL));

Respuesta correcta

La respuesta correcta es: ALTER TABLE Proveedor ADD CONSTRAINT chk_contacto
CHECK (Telefono IS NOT NULL OR Email IS NOT NULL);

Pregunta **2**

Correcta

Se puntúa como 0 sobre 1,00

1.b) En el esquema dado se requiere incorporar la siguiente restricción según SQL estándar utilizando el recurso declarativo más restrictivo posible (a nivel de atributo, de tupla, de tabla o general) y utilizando sólo las tablas/atributos necesarios.

- *Verificar que la cantidad total vendida de cada cosecha no exceda la cantidad cosechada de la misma.*

Seleccione la opción que considera correcta, de acuerdo a lo solicitado y justifique claramente (debajo de la pregunta 1.c):

a. CREATE ASSERTION check_fecha_siembra
CHECK (not exists (select 1 from cosecha c join venta v using (id_cultivo, nro_cosecha)
group by id_cultivo, nro_cosecha
having SUM(v.cantidad_vendida) > c.cantidad_cosechada));

b. ALTER TABLE venta add constraint check_fecha_siembra
CHECK (not exists (select 1 from cosecha c join venta v using (id_cultivo)
group by id_cultivo
having SUM(v.cantidad_vendida) > c.cantidad_cosechada));

c. Ninguna de las opciones

d. CREATE ASSERTION check_fecha_siembra
CHECK (not exists (select 1 from cosecha c join venta v using (id_cultivo, nro_cosecha)
where SUM(v.cantidad_vendida) > c.cantidad_cosechada
group by nro_cosecha));

e. CREATE ASSERTION check_fecha_siembra
CHECK (not exists (select 1 from cosecha c join venta v using (id_cultivo)
group by id_cultivo
having SUM(v.cantidad_vendida) > c.cantidad_cosechada));

f. ALTER TABLE cosecha add constraint check_fecha_siembra
CHECK (not exists (select 1 from cosecha c join venta v using (nro_cosecha)
group by nro_cosecha
having SUM(v.cantidad_vendida) <= c.cantidad_cosechada));

g. CREATE ASSERTION check_fecha_siembra
CHECK (exists (select 1 from cosecha c join venta v using (id_cultivo, nro_cosecha)
group by id_cultivo, nro_cosecha
having SUM(v.cantidad_vendida) <= c.cantidad_cosechada));

Respuesta correcta

La respuesta correcta es: CREATE ASSERTION check_fecha_siembra
CHECK (not exists (select 1 from cosecha c join venta v using (id_cultivo, nro_cosecha)
group by id_cultivo, nro_cosecha
having SUM(v.cantidad_vendida) > c.cantidad_cosechada));

Pregunta **3**

Finalizado

Se puntúa como 0 sobre 1,00

1.c) En el esquema dado se requiere incorporar la siguiente restricción según SQL estándar utilizando el recurso declarativo más restrictivo posible (a nivel de atributo, de tupla, de tabla o general) y utilizando sólo las tablas/atributos necesarios.

- Para cada cultivo, los números de cosecha deben reflejar el orden cronológico de las fechas de cosecha; es decir, un número de cosecha mayor debe corresponder a una fecha de cosecha posterior para el mismo cultivo.

Resuelva según lo solicitado y justifique el tipo de chequeo utilizado.

```
1)a) la opcion seleccionada abarca los casos donde el proveedor tiene un telefono o tiene un
b) la opcion indicada cumple con lo que pide, ya que tiene que ser un assertion porque tomar
c) ALTER TABLE cocecha ADD CONSTRAINT chk_1_c CHECK (
      NOT EXISTS (
         SELECT 'X'
         FROM cosecha c
         WHERE EXISTS (
            SELECT 'Y'
            FROM cosecha
            WHERE c.id_cultivo = id_cultivo AND c.nro_cosecha > nro_cosecha
```

Comentario:
a) bien- incompleta: por qué corresponde esta opción y no g?
b) bien
c) bien

Pregunta **4**

Correcta

Se puntúa como 0 sobre 1,00

2.a) Sobre el esquema dado se requiere definir la siguiente vista, de manera que resulte automáticamente actualizable en PostgreSQL, siempre que sea posible:

- *V1: que contenga los datos de los cultivos sembrados durante el corriente año que no registren inventario de productos adquiridos al proveedor 'AgroPlus'.*

Considerando la siguiente definición para V1, seleccione la/s afirmación/es que considere correcta/s respecto de esta vista (*Nota*: tenga en cuenta que las opciones incorrectamente seleccionadas pueden restar puntaje) y justifíquela/s claramente (debajo de la pregunta 2.c).

```
CREATE VIEW V1 AS
SELECT * FROM cultivo
WHERE EXTRACT(YEAR FROM fecha_siembra) = EXTRACT(YEAR FROM CURRENT_DATE)
 AND id_cultivo IN ( SELECT id_cultivo FROM inventario
                          JOIN proveedor using (id_proveedor)
                          WHERE nombre <> 'AgroPlus' );
```

- [ ] a. no resulta automáticamente actualizable en PostgreSQL
- [x] b. para cumplir lo requerido hay que reformularla, cambiando IN por NOT IN y <> por =
- [ ] c. no es posible reformularla para que cumpla lo requerido (y sea automáticamente actualizable)
- [x] d. incluye cultivos de este año que tienen inventarios de proveedores distintos de 'AgroPlus'
- [ ] e. para cumplir lo requerido hay que reformularla, sólo cambiando <> por =
- [ ] f. para cumplir lo requerido hay que reformularla, sólo cambiando IN por NOT IN
- [x] g. no garantiza que los cultivos de este año no tengan inventario de productos de 'AgroPlus'
- [ ] h. filtra correctamente los cultivos de este año que no tienen inventario de 'AgroPlus'
- [x] i. es automáticamente actualizable en PostgreSQL
- [ ] j. ninguna de las opciones

Respuesta correcta

Las respuestas correctas son: incluye cultivos de este año que tienen inventarios de proveedores distintos de 'AgroPlus', no garantiza que los cultivos de este año no tengan inventario de productos de 'AgroPlus'

, es automáticamente actualizable en PostgreSQL

,

para cumplir lo requerido hay que reformularla, cambiando IN por NOT IN y <> por =

Pregunta **5**

Correcta

Se puntúa como 0 sobre 1,00

2.b) Sobre el esquema dado se requiere definir la siguiente vista, de manera que resulte automáticamente actualizable en PostgreSQL, siempre que sea posible:

- *V2: que contenga para cada cosecha con al menos 3 ventas realizadas, el identificador de la cosecha, la cantidad total vendida y la fecha de la última venta registrada.*

Considerando la siguiente definición para V2, seleccione la/s afirmación/es que considere correcta/s respecto de esta vista (*Nota*: tenga en cuenta que las opciones incorrectamente seleccionadas pueden restar puntaje) y justifíquela/s claramente (debajo de la pregunta 2.c)

```
CREATE VIEW V2 AS
SELECT nro_cosecha, id_cultivo,
    SUM(cantidad_vendida) AS total_vendido,
    MAX(fecha_venta) AS ultima_venta
FROM venta
GROUP BY nro_cosecha,id_cultivo
HAVING COUNT(*) >= 3;
```

- [ ] a. calcula incorrectamente la fecha de la última venta registrada y de la fecha de la última venta registrada porque no incluye una cláusula WHERE para filtrar valores nulos
- [ ] b. la cláusula HAVING incluida no permite asegurar que solo se incluyan cosechas con al menos tres ventas realizadas
- [x] c. calcula correctamente la cantidad total vendida y la fecha de la última venta registrada por cada cosecha diferente
- [ ] d. no resuelve lo requerido debido a que la cantidad total vendida y la fecha de la última venta registrada no se asocian adecuadamente a cada cosecha diferente
- [x] e. mediante la cláusula HAVING se asegura que solo se incluyan cosechas con al menos tres ventas realizadas
- [ ] f. puede convertirse en actualizable si se elimina la cláusula HAVING
- [ ] g. ninguna de las opciones
- [ ] h. resulta automáticamente actualizable para el estándar SQL
- [ ] i. resulta automáticamente actualizable en PostgreSQL
- [x] j. no es automáticamente actualizable al incluir agrupamiento en su definición

Respuesta correcta

Las respuestas correctas son: calcula correctamente la cantidad total vendida y la fecha de la última venta registrada por cada cosecha diferente, mediante la cláusula HAVING se asegura que solo se incluyan cosechas con al menos tres ventas realizadas

, no es automáticamente actualizable al incluir agrupamiento en su definición

Pregunta **6**

Finalizado

Se puntúa como 0 sobre 1,00

2.c) Sobre el esquema dado se requiere definir la siguiente vista, de manera que resulte automáticamente actualizable en PostgreSQL, siempre que sea posible, y que se verifique que no haya migración de tuplas de la vista:. Resuelva según lo solicitado y justifique su solución.

- *V3: que contenga los datos de los cultivos que han tenido el mayor promedio de cantidad vendida el año actual.*

```
2)a) esta vista resulta automaticamente actualizable ya que la vista no contiene nada que la
b) esta vista no resulta automaticamente actualizable ya que contiene funciones de agregaci
c)
CREATE VIEW V3 AS (
   SELECT c.*
   FROM cosecha c
   WHERE (SELECT AVG(cantidad_vendida)
          FROM venta
          WHERE c.id_cultivo = id_cultivo AND EXTRACT(YEAR FROM(fecha_venta)) = EXTRACT(YEAR
         )
```

Comentario:

a) bien

b) bien

c) Reg. La consulta da error, falta paréntesis en el 2° extract en ambos casos; se piden datos de los cultivos, no de las cosechas. No verifica que no haya migración de tuplas

Pregunta **7**

Finalizado

Se puntúa como 0 sobre 1,00

3) Para el esquema dado, se ha creado la tabla *cultivos_agricultor* donde se requiere registrar la siguiente información para todos los agricultores que están registrados en la base:

id_agricultor, nombre, fecha_registro, cantidad_cultivos, fecha_ultima_siembra

donde, para cada agricultor:
- cantidad_cultivos corresponde a la cantidad de cultivos que registra
- fecha_ultima_siembra es la fecha correspondiente a la última siembra que realizó

Nota: en caso que un agricultor no registre cultivos, se deberá indicar apropiadamente.

a) Implemente el método más adecuado en PostgreSQL que permita completar dicha tabla con la información de todos los agricultores a partir de los datos existentes en la base. Explique su solución e incluya la sentencia que debería utilizar un usuario para la ejecución del mismo.

Nota: no puede utilizar sentencias de bucle (for, loop, etc.) para resolverlo.

```
3)a)
CREATE PROCEDURE pr_actualizar_cultivos_agricultor()
LANGUAGE 'plpgsql' AS
$$
DELETE FROM cultivos_agricultor; --no se nos indica que la tabla este vacia
INSERT INTO cultivos_agricultor (
   SELECT a.id_agricultor, a.nombre, a.fecha_registro,
  (SELECT COUNT(*) FROM cultivo WHERE a.id_agricultor = id_agricultor),
  COALESCE((SELECT MAX(fecha_siembra) FROM WHERE a.id_agricultor = id_agricultor), '00-00-0(
   FROM agricultor a
```

Comentario:
Regular

No existe la fecha 00-00-0000, porque no revisa la documentación ?

Y es bastante ineficiente, sería mucho, mucho mas eficiente si hace un ensamble y agrupa

Pregunta **8**

Finalizado

Se puntúa como 0 sobre 1,00

3.b) Indique y justifique todos los eventos críticos necesarios para mantener los datos actualizados en la tabla *cultivos_agricultor* cuando se produzcan actualizaciones en la base. Incluya la declaración de los triggers correspondientes en PostgreSQL y escriba la implementación de la/s función/es requerida/s para operaciones de insert.

```
3)b) eventos criticos a controlar:
UPDATE id_agricultor y fecha_siembra en tabla cultivo
DELETE en tabla cultivo
INSERT en tabla cultivo

--datos desactualizados del agricultor en la tabla--

INSERT en agricultor
UPDATE id_agricultor, nombre y fecha_registro en agricultor
DELETE agricultor
```

Comentario:

Regular

La fecha 00-00-0000 no existe, de donde sacó que eso funciona ?

Pregunta **9**

Correcta

Se puntúa como 0 sobre 1,00

4) Dada la siguiente vista creada en PostgreSQL sobre el esquema dado:.

```
create view V_Ventas_Cosecha as
 select v.*, c.Fecha_Cosecha, c.Cantidad_cosechada
 from Venta v join Cosecha using (ID_Cultivo, Nro_cosecha)
 where Cantidad_vendida > 10000 ;
```

4.a) Complete convenientemente la implementación del trigger a continuación para que permita eliminar tuplas de la vista

```
create or replace function fn_del_vista()
returns trigger as $$
begin
    delete from Venta
      where ID_Cultivo=old.ID_Cultivo and Nro_cosecha=old.Nro_cosecha and Fecha_Venta= old.Fecha_Venta
 ;
    return old
 ;
end$$
language 'plpgsql';

create trigger tr_del_vista
instead of delete on V_Ventas_Cosecha
for each row
execute function fn_instal_serv() ;
```

Respuesta correcta

La respuesta correcta es:

4) Dada la siguiente vista creada en PostgreSQL sobre el esquema dado:.

```
create view V_Ventas_Cosecha as
 select v.*, c.Fecha_Cosecha, c.Cantidad_cosechada
 from Venta v join Cosecha using (ID_Cultivo, Nro_cosecha)
 where Cantidad_vendida > 10000 ;
```

4.a) Complete convenientemente la implementación del trigger a continuación para que permita eliminar tuplas de la vista

```
create or replace function fn_del_vista()
[returns trigger as $$]
begin
    delete from [Venta]
      where [ID_Cultivo=old.ID_Cultivo and Nro_cosecha=old.Nro_cosecha and Fecha_Venta= old.Fecha_Venta];
   [return old];
end$$
language 'plpgsql';
```

```
create trigger tr_del_vista
[instead of delete on V_Ventas_Cosecha]
for each [row]
execute function fn_instal_serv() ;
```

Pregunta **10**

Finalizado

Se puntúa como 0 sobre 1,00

4.b) Detalle cómo implementaría actualizaciones sobre la vista para el resto de las operaciones.

```
b) las actualizaciones las haria todas sobre la tabla ventas, ya que conserva toda la clave p
   Por ende cada vez que se inserten o actualicen tuplas en la vista, todos los cambios los v
   Ademas para hacer el insert se deberia tomar todos los datos de la tupla a insertar (NEW)
   Para el caso de update buscaria la tupla que se quiere actualizar (filtrando con el NEW) y
   Para ambos casos haria un trigger instead of y unaa funcion diferente para ambos
```

Comentario:
Bien

Pregunta **11**

Incorrecta

Se puntúa como 0 sobre 1,00

5) La siguiente consulta sobre el esquema dado se ha planteado para recuperar los agricultores que han vendido sus cultivos en el último año y que supere la cantidad vendida a 1000 kilogramos indicando lo vendido, lo cosechado y cuando ha sido sembrado ese cultivo.

```
SELECT a.Nombre, cl.Fecha_siembra, c.Cantidad_cosechada, v.Cantidad_vendida
FROM Venta v  JOIN Cosecha c ON c.ID_Cultivo = v.ID_Cultivo and c.Nro_cosecha=v.Nro_cosecha
JOIN Cultivo cl ON cl.ID_Cultivo = c.ID_Cultivo
JOIN Agricultor a ON a.ID_Agricultor = cl.ID_agricultor
WHERE v.Cantidad_vendida > 1000 and v.Fecha_Venta > now() - '1 year'::interval ;
```

Determine cuál de los siguientes índices emplearía para optimizarla. Abajo justifique su elección y como el DBMS resuelve la recuperación de los datos de la consulta.

- a. create index idx_indice on Venta(Cantidad_vendida, Nro_cosecha);
- b. Ninguna de las opciones
- c. create index idx_indice on Venta(Cantidad_vendida);
- d. create index idx_indice on Venta(Fecha_Venta, Nro_cosecha);
- e. create index idx_indice on Venta(Fecha_Venta, Cantidad_vendida);
- f. create index idx_indice on Venta(Fecha_Venta);

Respuesta incorrecta.

La respuesta correcta es:
create index idx_indice on Venta(Fecha_Venta, Cantidad_vendida);

Pregunta **12**

Finalizado

Se puntúa como 0 sobre 1,00

Justifique su elección anterior y cómo el DBMS resuelve la recuperación de los datos de la consulta.

Como primero se recuperan las tuplas de las tablas afectadas en el WHERE, que en este caso s
luego de hacer el filtrado nos damos cuenta que ninguna de las dos tienen un indice asociado
como cantidad_vendida es la primera que se analiza necesitariamos un indice sobre esta para
no seleccionamos el indice de (fecha_venta,nro_cosecha) porque nuestra consulta no respeta e
v.Fecha_Venta > now() - '1 year'::interval AND v.Cantidad_vendida > 1000; el indice por (fe

Comentario:
Mal, el orden del WHERE no interfiere en la elección del índice.

Pregunta **13**

Correcta

Se puntúa como 0 sobre 1,00

6) Determine cuál es la mejor estructura que se debería usar en cada uno de los siguientes casos, para cada uno de los índices planteados de acuerdo a la consulta dada:

a) select * from Venta where v.Cantidad_vendida > 1000;

create index idx_indice1 on Venta(Cantidad_vendida);

Rta: BTree

b) select * from Venta where ID_Cultivo = 100 and Nro_cosecha = 50;

create index idx_indice1 on Venta(ID_Cultivo,Nro_cosecha);

Rta: Hash

c) select * from Venta where v.Cantidad_vendida between 1000 and 3000;

create index idx_indice1 on Venta(Cantidad_vendida);

Rta: BTree

Respuesta correcta

La respuesta correcta es:
6) Determine cuál es la mejor estructura que se debería usar en cada uno de los siguientes casos, para cada uno de los índices planteados de acuerdo a la consulta dada:

a) select * from Venta where v.Cantidad_vendida > 1000;

create index idx_indice1 on Venta(Cantidad_vendida);

Rta: [BTree]

b) select * from Venta where ID_Cultivo = 100 and Nro_cosecha = 50;

create index idx_indice1 on Venta(ID_Cultivo,Nro_cosecha);

Rta: [Hash]

c) select * from Venta where v.Cantidad_vendida between 1000 and 3000;

create index idx_indice1 on Venta(Cantidad_vendida);

Rta: [BTree]

Pregunta **14**

Finalizado

Se puntúa como 0 sobre 1,00

Justifique de modo claro y conciso su elección en cada caso.

a) se utiliza una busqueda por valores mayores a este y no iguales, por ende en este caso un
b) por el contrario en la consulta del inciso b) podemos notar que se realiza una busqueda po
c) mismo caso que en el inciso a), se utiliza un arbol binario o Btree para realizar la busqu

Comentario:
todas corrrectas.

Pregunta **15**

Parcialmente correcta

Se puntúa como 0 sobre 1,00

7) Sobre el esquema dado se definen los usuarios y roles:

Usuario alice (gerente de proyectos).

Usuario bob (analista financiero).

Usuario charlie (interno).

Rol managers (incluye a alice).

Rol analysts (incluye a bob).

El DBA ejecuta las siguientes sentencias SQL para configurar los permisos iniciales:

```
GRANT SELECT, UPDATE ON cosecha TO 'alice';
GRANT SELECT ON venta TO 'analysts';
GRANT INSERT ON agricultor TO 'managers';
GRANT ALL PRIVILEGES ON venta TO 'bob' WITH GRANT OPTION;
GRANT SELECT ON cosecha, agricultor TO 'charlie';
```

7.1) Después de ejecutar estos comandos, ¿cuáles de las siguientes afirmaciones son ciertas?

a) alice puede leer y modificar la tabla cosecha. Si

b) bob puede otorgar permisos sobre la tabla venta a otros usuarios. Si

c) charlie puede añadir nuevos empleados a la tabla agricultor. No

d) Los usuarios del rol managers tienen permisos para modificar el contenido de la tabla agricultor. No

e) bob puede eliminar la tabla venta. No

7.2) El DBA realiza los siguientes cambios en los permisos:

```
REVOKE SELECT ON cosecha FROM 'charlie';
REVOKE INSERT ON agricultor FROM 'managers';
GRANT DELETE ON venta TO 'alice';
```

¿Qué permisos tendrá cada usuario después de estas modificaciones?

a) charlie conserva el permiso de SELECT en la tabla agricultor pero no en cosecha. No

b) alice puede eliminar registros en la tabla venta pero no puede leerlos. Si

c) El rol managers ya no puede insertar nuevos empleados en agricultor. Si

d) bob pierde todos los permisos sobre venta debido al cambio en alice. No

7.3) Si alice también es añadida al rol analysts y el DBA ejecuta

REVOKE SELECT ON venta FROM 'analysts';

¿Qué sucede con el acceso de alice a la tabla venta?

a) alice conserva el permiso de SELECT porque se otorga directamente. No

b) alice pierde completamente el permiso de SELECT en venta. Si

c) alice solo conserva el permiso de UPDATE en venta. No

d) alice no puede ejecutar ninguna acción sobre venta. No

7.4) El DBA ejecuta

REVOKE GRANT OPTION FOR ALL PRIVILEGES ON venta FROM 'bob';

¿Cuál será el efecto de esta sentencia?

a) bob ya no puede otorgar permisos sobre venta pero conserva todos sus privilegios. Si

b) bob pierde todos los permisos sobre venta. No

c) Todos los usuarios a los que bob otorgó permisos sobre venta pierden acceso. No

d) bob conserva únicamente los permisos de lectura sobre venta. No

Respuesta parcialmente correcta.

Ha seleccionado correctamente 13.
La respuesta correcta es:

7) Sobre el esquema dado se definen los usuarios y roles:

- Usuario alice (gerente de proyectos).
- Usuario bob (analista financiero).
- Usuario charlie (interno).
- Rol managers (incluye a alice).
- Rol analysts (incluye a bob).

El DBA ejecuta las siguientes sentencias SQL para configurar los permisos iniciales:

```
GRANT SELECT, UPDATE ON cosecha TO 'alice';
GRANT SELECT ON venta TO 'analysts';
GRANT INSERT ON agricultor TO 'managers';
GRANT ALL PRIVILEGES ON venta TO 'bob' WITH GRANT OPTION;
GRANT SELECT ON cosecha, agricultor TO 'charlie';
```

7.1) Después de ejecutar estos comandos, ¿cuáles de las siguientes afirmaciones son ciertas?

a) alice puede leer y modificar la tabla cosecha. [Si]

b) bob puede otorgar permisos sobre la tabla venta a otros usuarios. [Si]

c) charlie puede añadir nuevos empleados a la tabla agricultor. [No]

d) Los usuarios del rol managers tienen permisos para modificar el contenido de la tabla agricultor. [No]

e) bob puede eliminar la tabla venta. [No]

7.2) El DBA realiza los siguientes cambios en los permisos:

REVOKE SELECT ON cosecha FROM 'charlie';

REVOKE INSERT ON agricultor FROM 'managers';

GRANT DELETE ON venta TO 'alice';

¿Qué permisos tendrá cada usuario después de estas modificaciones?

a) charlie conserva el permiso de SELECT en la tabla agricultor pero no en cosecha. [Si]

b) alice puede eliminar registros en la tabla venta pero no puede leerlos. [No]

c) El rol managers ya no puede insertar nuevos empleados en agricultor. [Si]

d) bob pierde todos los permisos sobre venta debido al cambio en alice. [No]

7.3) Si alice también es añadida al rol analysts y el DBA ejecuta

REVOKE SELECT ON venta FROM 'analysts';

¿Qué sucede con el acceso de alice a la tabla venta?

a) alice conserva el permiso de SELECT porque se otorga directamente. [Si]

b) alice pierde completamente el permiso de SELECT en venta. [No]

c) alice solo conserva el permiso de UPDATE en venta. [No]

d) alice no puede ejecutar ninguna acción sobre venta. [No]

7.4) El DBA ejecuta

REVOKE GRANT OPTION FOR ALL PRIVILEGES ON venta FROM 'bob';

¿Cuál será el efecto de esta sentencia?

a) bob ya no puede otorgar permisos sobre venta pero conserva todos sus privilegios. [Si]

b) bob pierde todos los permisos sobre venta. [No]

c) Todos los usuarios a los que bob otorgó permisos sobre venta pierden acceso. [No]

d) bob conserva únicamente los permisos de lectura sobre venta. [No]

Ir a...

Siguiente actividad

Final BDI 11/07/2024 ►

Mantente en contacto

Facultad, Pabellón Central Paraje Arroyo Seco. Campus Universitario. (B7001BBO) Tandil. Buenos Aires, Argentina

- https://exa.unicen.edu.ar/
- (+54) (0249) 438-5650 Conmutador: int. 2000
- moodle@exa.unicen.edu.ar

Descargar la app para dispositivos móviles