

[Área personal](#) ▶ [Cursos](#) ▶ [Exámenes](#) ▶ [BD-Tecn-Finales](#) ▶ [Finales Julio-Agosto 2025](#) ▶

[Final Bases de Datos 17/07/2025 Sede Tandil](#)

**Comenzado el** jueves, 17 de julio de 2025, 09:40

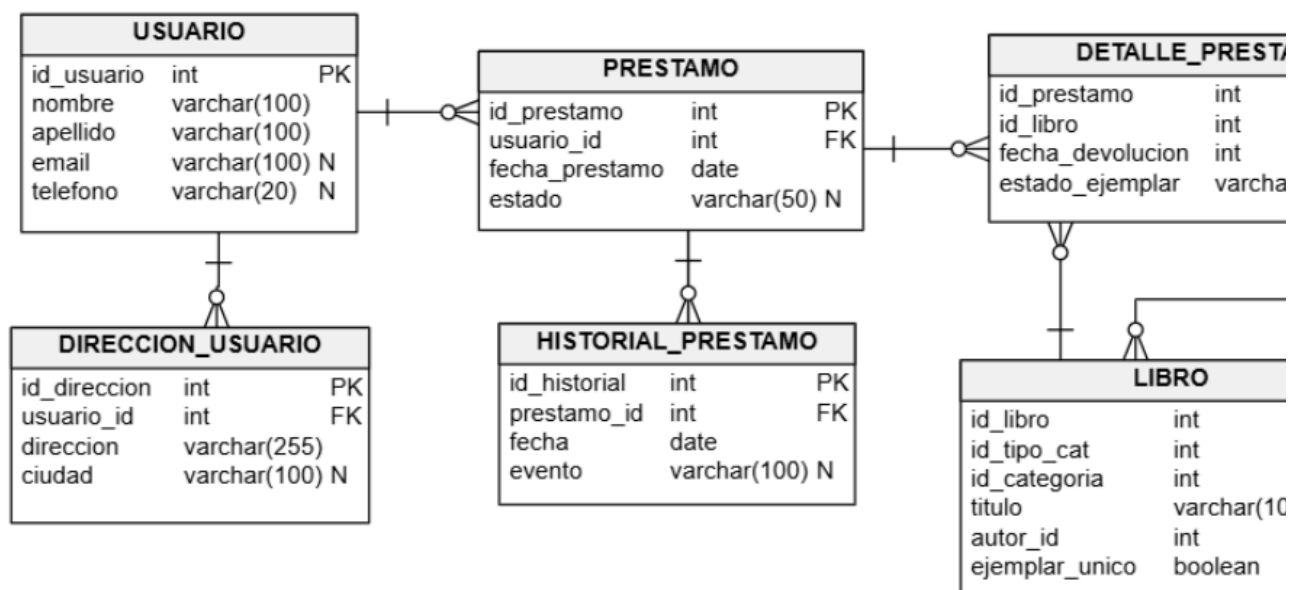
**Estado** Finalizado

**Finalizado en** jueves, 17 de julio de 2025, 12:34

**Tiempo empleado** 2 horas 53 minutos

Información

El siguiente esquema de base de datos ([link](#)) forma parte de un sistema de gestión de información relacionada con una biblioteca. La información con la que se cuenta es con los libros que posee dicha biblioteca junto a su autor y los préstamos realizados a cada usuario. Cada préstamo puede involucrar a varios libros y es posible hacer devoluciones parciales, es decir por ejemplo de un mismo préstamo devolver sólo un libro.



Desarrolle los ejercicios indicados a continuación:

## Pregunta 1

Finalizado

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 usuarios tengan registrado un teléfono o un email.*

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

- ☐ a. ALTER TABLE usuario ADD CONSTRAINT chk\_contacto  
CHECK (telefono IS NOT NULL AND email IS NOT NULL);
- ☒ b. ALTER TABLE usuario ADD CONSTRAINT chk\_contacto  
CHECK (NOT EXISTS (SELECT 1 from usuario  
WHERE telefono IS NULL AND email IS NULL));
- ☐ c. Ninguna de las opciones
- ☐ d. ALTER TABLE usuario ADD CONSTRAINT chk\_contacto  
CHECK (telefono IS NULL AND email IS NULL);
- ☐ e. ALTER TABLE usuario ADD CONSTRAINT chk\_contacto  
CHECK (telefono IS NOT NULL OR email IS NOT NULL);
- ☐ f. ALTER TABLE usuario ADD CONSTRAINT chk\_contacto  
CHECK (NOT EXISTS (SELECT 1 from usuario  
WHERE telefono IS NOT NULL OR email IS NOT NULL));
- ☐ g. ALTER TABLE usuario ADD CONSTRAINT chk\_contacto  
CHECK (telefono IS NULL OR email IS NULL);

Respuesta incorrecta.

La respuesta correcta es:

ALTER TABLE usuario ADD CONSTRAINT chk\_contacto  
CHECK (telefono IS NOT NULL OR email IS NOT NULL);

Comentario: No es necesario realizar un chequeo de tabla cuando lo podemos hacer (e implementar) con un chequeo de tupla

## Pregunta 2

Finalizado

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 controle que controle que todas las categorías de libros que contengan el string 'Edición Especial' deben tener descripción.*

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

- ☐ a. ALTER TABLE categoria\_libro  
ADD CONSTRAINT ck\_chequeo\_categoria\_libro  
CHECK ( EXISTS  
(SELECT 1  
FROM categoria\_libro  
WHERE UPPER(nombre) LIKE '%EDICION ESPECIAL%' AND descripcion IS NOT NULL ));
- ☒ b. ALTER TABLE categoria\_libro  
ADD CONSTRAINT ck\_chequeo\_categoria\_libro  
CHECK ( NOT EXISTS  
(SELECT 1  
FROM categoria\_libro  
WHERE UPPER(nombre) LIKE '%EDICION ESPECIAL%' AND descripcion IS NULL ));
- ☐ c. Ninguna de las opciones
- ☐ d. ALTER TABLE categoria\_libro  
ADD CONSTRAINT ck\_chequeo\_categoria\_libro  
CHECK ( ( UPPER(nombre) LIKE '%EDICION ESPECIAL%' AND descripcion IS NOT NULL));
- ☐ e. CREATE ASSERTION ck\_chequeo\_categoria\_libro  
CHECK ( EXISTS  
(SELECT 1  
FROM categoria\_libro  
WHERE UPPER(nombre) LIKE '%EDICION ESPECIAL%' AND descripcion IS NULL ));
- ☐ f. ALTER TABLE categoria\_libro  
ADD CONSTRAINT ck\_chequeo\_categoria\_libro  
CHECK ( ( UPPER(nombre) LIKE '%EDICION ESPECIAL%' AND descripcion IS NOT NULL)  
OR ( UPPER(nombre) NOT LIKE '%EDICION ESPECIAL%' ) );

- ☐ 9. CREATE ASSERTION ck\_chequeo\_categoria\_libro  
CHECK ( NOT EXISTS  
(SELECT 1  
FROM categoria\_libro  
WHERE UPPER(nombre) LIKE '%EDICION ESPECIAL%' AND descripcion IS NULL ));

Respuesta incorrecta.

La respuesta correcta es:

```
ALTER TABLE categoria_libro  
ADD CONSTRAINT ck_chequeo_categoria_libro  
CHECK ( ( UPPER(nombre) LIKE '%EDICION ESPECIAL%' AND descripcion IS NOT NULL)  
OR ( UPPER(nombre) NOT LIKE '%EDICION ESPECIAL%' ) );
```

Comentario: Al tener el NOT EXISTS la consulta me debe retornar todo lo que está mal, entonces para que todas las categorías de EDICION ESPECIAL tengan descripción debe cumplirse que la descripción NO SEA NULA, pero además no es necesario un chequeo de tabla

## 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.

- Cada préstamo sólo puede incluir un único libro que sea ejemplar único

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

**Nota:** No se olvide de justificar aquí las preguntas 1.a) y 1.b)

1a) - Controlar que los usuarios tengan registrado un teléfono o un email.

La respuesta b. es la correcta:

```
ALTER TABLE usuario ADD CONSTRAINT chk_contacto
```

```
CHECK (NOT EXISTS (SELECT 1 from usuario
```

```
WHERE telefono IS NULL AND email IS NULL));
```

ya que realiza un RI TUPLA porque involucra valores de las filas en la tabla USUARIO, además de que chequea que NO EXISTAN ambos atributos nulos con AND. Así nos aseguramos de que tengan registrados teléfono O email.

1b) - Verificar que controle que controle que todas las categorías de libros que contengan el string 'Edición

Comentario:

1.a) Mal Justificado y seleccionado

1.b) Mal Justificado y seleccionado

1.c) Regular, no hace falta la tabla prestamo en el chequeo

## Pregunta 4

Finalizado

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:

- *VI: listado de todos los préstamos del año 2025 que tengan más de 2 libros sin devolver (que no poseen fecha de devolución)*

```
CREATE OR REPLACE VIEW VI_P4 AS
SELECT p.id_prestamo
FROM PRESTAMO p
JOIN DETALLE_PRESTAMO dp ON p.id_prestamo = dp.id_prestamo
WHERE EXTRACT(YEAR FROM p.fecha_prestamo) = '2025'
AND dp.fecha_devolucion IS NULL
GROUP BY p.id_prestamo
HAVING COUNT > 2;
```

Comentario:

Mal, de que sirve que sólo liste el id\_prestamo ?

Hay hay errores de sintaxis. es count(\*)

## Pregunta 5

Finalizado

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: listado detallado de préstamos de libros que incluya el identificador y apellido del usuario, el identificador y título del libro, el identificador del préstamo y la fecha en que se realizó, así como la cantidad total de préstamos realizados por cada libro, ordenado en forma descendente según la cantidad de préstamos por libro.

```
CREATE OR REPLACE VIEW V2_P5 AS
SELECT u.id_usuario,
       u.apellido,
       dp.id_libro,
       l.titulo,
       dp.id_prestamo,
       p.fecha_prestamo,
       SUM OVER(PARTITION BY dp.id_libro) AS cantidad_total_prestamos_por_libro
FROM DETALLE_PRESTAMO dp
JOIN LIBRO l ON dp.id_libro = l.id_libro
JOIN PRESTAMO p ON dp.id_prestamo = p.id_prestamo
JOIN USUARIO u ON p.usuario_id = u.id_usuario
ORDER BY SUM OVER(PARTITION BY dp.id_libro) DESC;
```

Comentario:

Mal

- sum que... tiene que especificar que es lo que quiere sumar

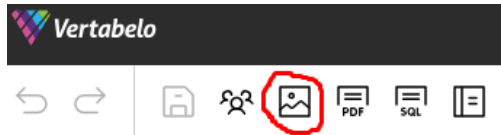
Pregunta 6

Finalizado

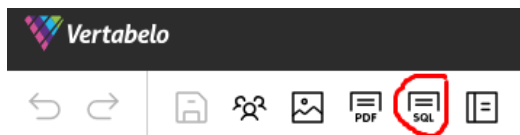
Se puntúa como 0 sobre 1,00

3) Dado el script de tablas ([link](#)) complete y/o corrija (en caso de ser necesario) en Vertabelo el esquema de tablas y restricciones de integridad (de claves, referenciales y de nulidad) para el siguiente diagrama de entidades y relaciones. Incluya los siguientes archivos:

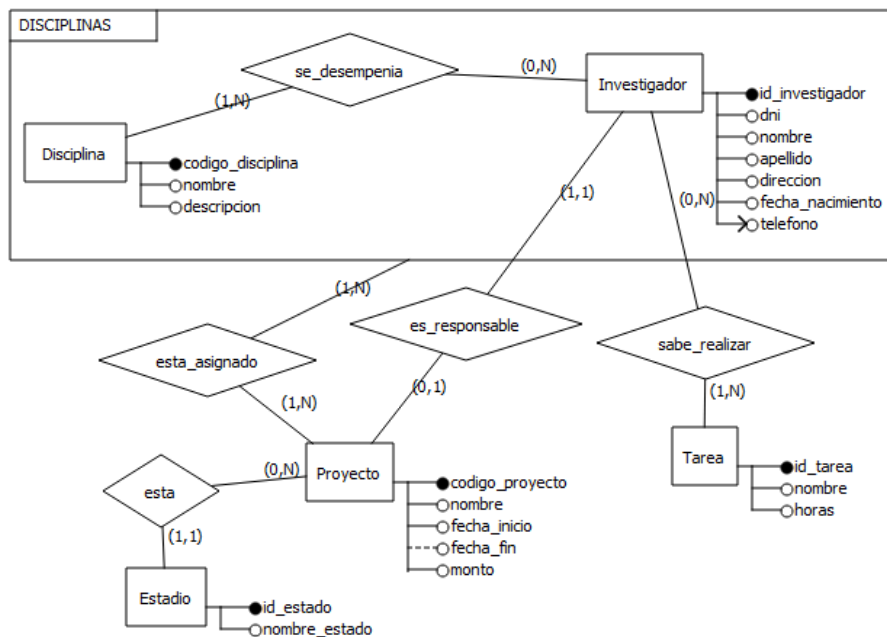
1. Imagen del esquema en PNG: nombrado como **DERE\_IMG\_XXX.png** (donde XXX son los últimos 3 nros de su documento): El tipo de imagen seleccionado debe ser PNG.



2. Script SQL generado por Vertabelo nombrado como **DERE\_ESQ\_XXX.sql** (donde XXX son los últimos 3 nros de su documento)



**Nota:** nombre adecuadamente los constraint de PK, FK, AK,



Adjunto ambos archivos solicitados.

[DERE\\_ESQ\\_720.png](#)



[DERE\\_ESQ\\_720.sql](#)

Comentario: Error PK tabal investigador\_telefono

Mal transformada relación está\_asignado

Debe quitar los prefijos de las tablas en las columnas que son FK

No respeta los nombres de los constraints de PK, FK, AK

#### Pregunta 7

Finalizado

Se puntúa como 0 sobre 1,00

4) Utilizando el [esquema dado](#) cree el recurso que crea más apropiado para generar un reporte que deberá quedar almacenado en la base de datos y que reciba un año por parámetro. El mismo debe ser para todos los usuarios y debe tener para cada usuario al menos:

- Nombre completo del usuario
- Total de préstamos realizados por cada usuario
- Cantidad de préstamos activos de cada usuario
- Cantidad de libros devueltos en mal estado. por cada usuario
- Fecha del último préstamo de cada usuario

\*Importante, no puede utilizar sentencias de bucle como por ejemplo for, while, etc.

No recuerdo como es la sintaxis correcta para generar el reporte, pero al pedirlo, le pasaría un atributo/variable anio por parámetro, sería el año en el que es generado, entonces lo primero que hace es imprimir: 'Reporte año ' || anio  
y luego hace un: SELECT 1 FROM V\_P7  
mostrando todos los datos solicitados.

```
CREATE OR REPLACE VIEW V_P7 AS
SELECT u.id_usuario,
       (u.nombre || ' ' || u.apellido) AS nombre_completo,
       SUM OVER(PARTITION BY p.id_prestamo) AS cantidad_prestamos_activos,
       SUM OVER(PARTITION BY dp.estado_ejemplar) AS cantidad_libros_devueltos_mal_estado,
       MAX OVER(PARTITION BY p.fecha_prestamo) AS fecha_ultimo_prestamo
LEFT JOIN PRESTAMO p ON u.id_usuario = p.id_usuario
LEFT JOIN DETALLE_PRESTAMO ON p.id_usuario = dp.id_usuario
WHERE p.estado = 'activo'
```

Comentario:

Mal

- Sum que ?

- Max que ?

- podía revisar la "sintaxis" en la página de PostgreSQL

## Pregunta 8

Finalizado

Se puntúa como 0 sobre 1,00

5) En el [esquema dado](#) se detectó un problema:

Algunos usuarios logran registrar múltiples direcciones residenciales idénticas, generando redundancia innecesaria en la tabla DIRECCION\_USUARIO.

Por otro lado, se quiere evitar que un usuario tenga más de una dirección registrada por ciudad, ya que esto genera conflictos logísticos en los envíos de correspondencia.

La política que se desea imponer es:

*- Un usuario no puede registrar más de una dirección en la misma ciudad, y no puede registrar direcciones idénticas (mismo texto en dirección) más de una vez.*

- a) Agregar una restricción UNIQUE(usuario\_id, direccion) en la tabla DIRECCION\_USUARIO.
- b) Agregar una restricción UNIQUE(usuario\_id, ciudad) en la tabla DIRECCION\_USUARIO.
- c) Combinar las restricciones UNIQUE(usuario\_id, direccion) y UNIQUE(usuario\_id, ciudad) en la tabla, aunque esto puede producir errores si se insertan direcciones distintas dentro de la misma ciudad.
- d) Crear un trigger BEFORE INSERT OR UPDATE que consulte si ya existe una fila para el mismo usuario\_id con la misma ciudad o la misma dirección exacta, y si es así, rechace la operación.
- e) Rediseñar la tabla DIRECCION\_USUARIO creando una entidad separada para CIUDAD, asociándola por clave foránea y asegurando unicidad en una tabla intermedia.
- f) Sólo se puede resolver a nivel de aplicación, ya que las bases de datos relacionales no permiten imponer múltiples restricciones condicionales sobre columnas diferentes.
- g) Ninguna opción es correcta

d) Crear un trigger BEFORE INSERT OR UPDATE que consulte si ya existe una fila para el mismo usuario\_id con la misma ciudad o la misma dirección exacta, y si es así, rechace la operación.

Comentario:

La rta correcta es c)

## Pregunta 9

Finalizado

Se puntúa como 0 sobre 1,00

6) Sobre el [esquema dado](#) se definen los usuarios y roles:

Usuario alice (gerente de biblioteca).

Usuario bob (analista datos).

Usuario charlie (bibliotecario).

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 libro TO 'alice';
```

```
GRANT SELECT ON detalle_prestamo TO 'analysts';
```

```
GRANT INSERT ON categoria_libro TO 'managers';
```

```
GRANT ALL PRIVILEGES ON detalle_prestamo TO 'bob' WITH GRANT OPTION;
```

```
GRANT SELECT ON libro, categoria_libro TO 'charlie';
```

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

a) alice puede leer y modificar la tabla libro.

☐ Si

b) bob puede otorgar permisos sobre la tabla detalle\_prestamo a otros usuarios.

☐ Si

c) charlie puede añadir nuevos registros a la tabla categoria\_libro.

☐ No

d) Los usuarios del rol managers tienen permisos para modificar el contenido de la tabla categoria\_libro.

☐ No

e) bob puede eliminar la tabla detalle\_prestamo.

☐ Si

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

```
REVOKE SELECT ON libro FROM 'charlie';
```

```
REVOKE INSERT ON categoria_libro FROM 'managers';
```

```
GRANT DELETE ON detalle_prestamo TO 'alice';
```

¿Qué permisos tendrá cada usuario después de estas modificaciones?

a) charlie conserva el permiso de SELECT en la tabla categoria\_libro pero no en libro.

☐ Si

b) alice puede eliminar registros en la tabla detalle\_prestamo pero no puede leerlos.

☐ Si

c) El rol managers ya no puede insertar nuevos registros en categoria\_libro.

☐ Si

d) bob pierde todos los permisos sobre detalle\_prestamo debido al cambio en alice.

☐ No

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

```
REVOKE SELECT ON detalle_prestamo FROM 'analysts';
```

¿Qué sucede con el acceso de alice a la tabla detalle\_prestamo?

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

b) alice pierde completamente el permiso de SELECT en detalle\_prestamo.

c) alice solo conserva el permiso de UPDATE en detalle\_prestamo.

d) alice no puede ejecutar ninguna acción sobre detalle\_prestamo.

6.4) El DBA ejecuta

```
REVOKE GRANT OPTION FOR ALL PRIVILEGES ON detalle_prestamo FROM 'bob';
```

¿Cuál será el efecto de esta sentencia?

a) bob ya no puede otorgar permisos sobre detalle\_prestamo pero conserva todos sus privilegios.

b) bob pierde todos los permisos sobre detalle\_prestamo.

c) Todos los usuarios a los que bob otorgó permisos sobre detalle\_prestamo pierden acceso.

d) bob conserva únicamente los permisos de lectura sobre detalle\_prestamo.

Respuesta parcialmente correcta.

Ha seleccionado correctamente 12.

La respuesta correcta es:

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

Usuario alice (gerente de biblioteca).

Usuario bob (analista datos).

Usuario charlie (bibliotecario).

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 libro TO 'alice';
```

```
GRANT SELECT ON detalle_prestamo TO 'analysts';
```

```
GRANT INSERT ON categoria_libro TO 'managers';
```

```
GRANT ALL PRIVILEGES ON detalle_prestamo TO 'bob' WITH GRANT OPTION;
```

```
GRANT SELECT ON libro, categoria_libro TO 'charlie';
```

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

a) alice puede leer y modificar la tabla libro.

b) bob puede otorgar permisos sobre la tabla detalle\_prestamo a otros usuarios.

c) charlie puede añadir nuevos registros a la tabla categoria\_libro.

d) Los usuarios del rol managers tienen permisos para modificar el contenido de la tabla categoria\_libro.

e) bob puede eliminar la tabla detalle\_prestamo.

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

```
REVOKE SELECT ON libro FROM 'charlie';
```

```
REVOKE INSERT ON categoria_libro FROM 'managers';
```

```
GRANT DELETE ON detalle_prestamo TO 'alice';
```

¿Qué permisos tendrá cada usuario después de estas modificaciones?

- a) charlie conserva el permiso de SELECT en la tabla categoria\_libro pero no en libro. [Si]
- b) alice puede eliminar registros en la tabla detalle\_prestamo pero no puede leerlos. [No]
- c) El rol managers ya no puede insertar nuevos registros en categoria\_libro. [Si]
- d) bob pierde todos los permisos sobre detalle\_prestamo debido al cambio en alice. [No]

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

REVOKE SELECT ON detalle\_prestamo FROM 'analysts';

¿Qué sucede con el acceso de alice a la tabla detalle\_prestamo?

- a) alice conserva el permiso de SELECT porque se otorga directamente. [Si]
- b) alice pierde completamente el permiso de SELECT en detalle\_prestamo. [No]
- c) alice solo conserva el permiso de UPDATE en detalle\_prestamo. [No]
- d) alice no puede ejecutar ninguna acción sobre detalle\_prestamo. [No]

6.4) El DBA ejecuta

REVOKE GRANT OPTION FOR ALL PRIVILEGES ON detalle\_prestamo FROM 'bob';

¿Cuál será el efecto de esta sentencia?

- a) bob ya no puede otorgar permisos sobre detalle\_prestamo pero conserva todos sus privilegios. [Si]
- b) bob pierde todos los permisos sobre detalle\_prestamo. [No]
- c) Todos los usuarios a los que bob otorgó permisos sobre detalle\_prestamo pierden acceso. [No]
- d) bob conserva únicamente los permisos de lectura sobre detalle\_prestamo. [No]

#### Actividad previa

◀ Avisos

Ir a...

## 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](tel:+5402494385650) Conmutador: int. 2000

✉ [moodle@exa.unicen.edu.ar](mailto:moodle@exa.unicen.edu.ar)



📱 Descargar la app para dispositivos móviles

[Facultad de Ciencias Exactas](#) – [UNICEN](#)

Contacto administradores plataforma: E-mail [moodle@exa.unicen.edu.ar](mailto:moodle@exa.unicen.edu.ar) – Tel. [+54 0249 4385650](tel:+5402494385650) int. 2098