

Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2413 - Bases de Datos

Interrogación 2

Fecha: 28 de octubre de 2025

2º semestre 2025 - Profesores: Eduardo Bustos - Christian Álvarez

---

Si Ud. no lee instrucciones, lea al menos estas INSTRUCCIONES

a) La prueba tiene 4 preguntas y una adicional de Bonus. Cada pregunta vale 15 puntos y el Bonus 6 puntos, en total 66 puntos. Con 60 puntos es un 7.

b) La prueba tiene una **Duración 2 horas**.

c) Lea toda la prueba primero, las preguntas NO ESTÁN en orden de dificultad

d) Al finalizar la prueba debe digitalizar cada pregunta mas el torpedo en formato PDF VERTICAL y subirlas a Canvas **correctamente**.

e) Cada plana debe contener su Nombre Completo y Número de Lista en la parte superior.

f) No mezcle respuestas de diferentes preguntas en la misma plana. Si puede responder una pregunta en una plana y otra en la otra plana de la misma hoja.

g) Debe firmar la lista de asistencia.

h) No seguir esta instrucción dificulta el trabajo de evaluación y por ende el tiempo de entrega de las notas, además los expone a un descuento de hasta 5 décimas de la nota final.

Pregunta 1 - Lógica en la BD

Una empresa necesita gestionar el inventario de sus activos (equipos de oficina, computadores, vehículos, etc.). Para ello, se cuenta con el siguiente modelo de datos simplificado:

- **Activo**
  - `id_activo` (PK): Identificador único del activo
  - `nombre`: Nombre del activo
  - `categoria`: Tipo de activo, puede tomar uno de los siguientes valores:
    - `'EQUIPO'` (computadores, impresoras, etc.)
    - `'MUEBLE'` (escritorios, sillas, etc.)
    - `'VEHICULO'` (autos, camionetas, etc.)
  - `valor`: Costo del activo en moneda local (debe ser mayor a 0)
  - `estado`: Situación actual del activo, por ejemplo:
    - `'EN USO'`
    - `'FUERA DE USO'`
    - `'EN MANTENCIÓN'`
- **Ubicacion**
  - `id_ubicacion` (PK): Identificador único de la ubicación
  - `nombre`: Nombre del lugar (ej. "Oficina Central")
  - `descripcion`: Detalle adicional de la ubicación
- **Movimiento**
  - `id_movimiento` (PK): Identificador único del movimiento
  - `id_activo` (FK): Activo que se mueve
  - `id_ubicacion` (FK): Ubicación del movimiento
  - `fecha_movimiento`: Fecha en que se realizó el movimiento
  - `tipo_movimiento`: Puede ser:
    - `'ENTRADA'`: El activo ingresa a una ubicación.
    - `'SALIDA'`: El activo sale de una ubicación.

a) Cree una vista llamada `vw_activos_detalle` que muestre: nombre del activo, categoría, valor, estado, última ubicación y fecha del último movimiento. (5 pts)

b) Cree un procedimiento almacenado llamado `sp_registrar_movimiento` que: (5 pts)

- Reciba como parámetros: `id_activo`, `id_ubicacion`, `tipo_movimiento`.
- Inserte un nuevo registro en **Movimiento** con la fecha actual.
- Actualice el estado del activo a:
  - `'EN USO'` si el movimiento es de tipo `ENTRADA`.
  - `'FUERA DE USO'` si el movimiento es de tipo `SALIDA`.

c) Cree un trigger llamado `trg_valor_activo` que: (5 pts)

- Se dispare **antes de insertar** en la tabla **Activo**.
- Valide que el valor del activo sea mayor a 0.
- Si no se cumple, debe impedir la inserción y mostrar un mensaje de error.

Pregunta 2 - ACID, Transacciones y Locks

Una base de datos maneja transacciones concurrentes. Considera las siguientes operaciones realizadas por tres transacciones **T1**, **T2** y **T3** sobre los mismos datos.

a) Explica los siguientes conceptos y da un ejemplo para cada uno de ellos: (3 pts)

- Lectura sucia
- Lectura no repetible
- Actualización perdida

b) Para el siguiente cronograma, determina si ocurren alguno de los conflictos de la pregunta **a**. Justifique su respuesta. (3 pts)

T1	**T2**	**T3**
R(X)		
		R(X)
W(X)		
	R(Y)	
	W(Y)	
		R(Z)
		W(X)
W(Y)		
	W(Z)	
		W(Z)

Cuadro 1: Schedule S

c) Verifica si el cronograma `Cuadro 1` es *Conflict Serializable* usando el método del *Grafo de precedencia*. Justifique su respuesta. (4 pts)

d) Aplica el método de las *Permutaciones* para verificar si el cronograma del `Cuadro 2` es *Conflict serializable*. (5 pts)

T1	**T2**	**T3**
R(X)		
		R(Z)
		W(X)
	W(Y)	
		R(X)
W(X)		
		W(Z)
	R(Y)	
W(Y)		
	W(Z)	

Cuadro 2: Schedule S2

Pregunta 3 - ACID, Log y Recuperación de Fallas

Responda cada una de las siguientes preguntas:

a) Analice las siguientes afirmaciones sobre *Checkpoint* y responda la pregunta (5 pts)

I. Los *Checkpoint* son estados de la BD que son consistentes.

II. En *Undo Logging* al iniciar un checkpoint se marcan todas las transacciones activas.

III. En *Redo Logging* al iniciar un checkpoint se marcan todas las transacciones activas que hayan realizado commit.

IV. Para marcar `<END CKPT>` en *Undo Logging* todas las transacciones activas deben finalizar.

V. Para marcar `<END CKPT>` en *Redo Logging* todas las transacciones activas con `COMMIT` deben finalizar.

Seleccione la alternativa correcta.

i. Sólo I, II y IV son verdaderas

ii. Solo I, III y V son verdaderas

iii. Solo I, II, IV y V son verdaderas

iv. Todas las afirmaciones son verdaderas

v. Todas las afirmaciones son falsas

b) Considera el siguiente contenido del log antes de la falla:

⟨START T1⟩
⟨T1, X, 10⟩
⟨T1, Y, 30⟩
⟨START T2⟩
⟨COMMIT T1⟩
⟨T2, Z, 50⟩

Si la falla ocurre **después** de escribir el último registro, pero **antes** de que T2 haga COMMIT. ¿Qué acciones se realizan usando *Undo Logging*? (5 pts)

c) Considera el siguiente contenido del log antes de la falla

⟨START T1⟩
⟨START T2⟩
⟨T1, X, 10⟩
⟨T2, Z, 50⟩
⟨START T3⟩
⟨T1, Y, 30⟩
⟨T3, A, 20⟩
⟨COMMIT T2⟩
⟨COMMIT T1⟩
⟨T2 END⟩

Si la falla ocurre **después** de escribir el último registro ¿Qué acciones se realizan usando *Redo Logging*? (5 pts)

Pregunta 4 - EEDD, Almacenamiento e Índices

Sea T(a,b,c,d) una tabla con 2.000.000 de tuplas. Si cada página puede almacenar P punteros o M tuplas.

a) Si existe un ínidce *B+ unclustered* sobre el campo *c*. ¿Cuál es el costo de $I/0$ de la siguiente consulta? Justifique su respuesta. (5 pts)

```
Select a,b from T where c <12
```

b) Si existe un índice *B+ Clustered* sobre el campo *b* ¿Cuál es el costo de I/O de la siguiente consulta? Justifique su respuesta. (5 pts)

```
Select a, b from T where d=5
```

c) Si existe un índice *Hash clustered* sobre el campo *a*. ¿Cuál es el costo de I/O de la siguiente consulta? Justifique su respuesta. (5 pts)

```
Select a, b, c from T where a=457
```

Bonus - 6 puntos

a) Analice las siguientes afirmaciones sobre *Fallas en la ejecución de las transacciones* y responda la pregunta. (2 pts)

I. La *limpieza de datos* es una forma para solucionar problemas de *Datos Erróneos*

II. *Transaction Manager* es el módulo que previene *Fallas del Sistema* SABD, ya sea por fallos de sofware, hardware o errores humanos.

III. Las *Catástrofes* como incendios, terremotos o inundaciones pueden ser solucionadas usando *RAID*

Seleccione la alternativa correcta y justifique su respuesta.

A. Sólo I y III son verdaderas

B. Sólo I y II son verdaderas

C. Sólo I es verdadera

D. Sólo III es verdadera

E. Todas las afirmaciones son verdaderas

b) Analice las siguientes afirmaciones sobre *ACID* y responda la pregunta. (2 pts)

I. La *Atomicidad* es una propiedad que asegura que toda transacción se ejecute independiente de otras transacciones.

II. La Durabilidad es asegurada por el módulo *Log y Recovery Manager*

III. Si una BD no respeta la propiedad de la *Consistencia*, se pueden violar las llaves primarias o foráneas.

Seleccione la alternativa correcta y justifique su respuesta.

A. Sólo I y III son verdaderas

B. Sólo II y III son verdaderas

C. Sólo I y II son verdaderas

D. Todas las afirmaciones son verdaderas

E. Todas las afirmaciones son falsas

c) ¿Cuál de las siguientes afirmaciones describe correctamente la diferencia entre un índice basado en hashing y un índice basado en árboles B+? (2 pts)

Seleccione la alternativa correcta y justifique su respuesta.

A. Los índices hash son más eficientes para consultas por rangos que los árboles B+.

B. Los árboles B+ no permiten búsquedas exactas, solo consultas por rangos.

C. Los índices hash organizan los datos en nodos ordenados, mientras que los árboles B+ usan funciones de dispersión.

D. Los índices hash son ideales para búsquedas exactas, mientras que los árboles B+ son mejores para consultas por rangos.

E. Ambos tipos de índices ofrecen el mismo rendimiento para búsquedas exactas y por rangos.

Torpedo

Sintaxis consulta SQL

```
SELECT [ALL | DISTINCT] * | [COUNT, MAX, MIN, SUM, AVG](col1), col2, ...
       [AS alias_column]
FROM table1
     [JOIN table2 ON table1.common_field = table2.common_field]
WHERE condition1 [AND | OR condition2]
      [BETWEEN value1 AND value2]
      [IN (value1, value2, ...)]
      [LIKE pattern]
      [= | <> | <= | >= | < | >] value
GROUP BY column1, column2, ...
HAVING aggregate_condition [COUNT, MAX, MIN, SUM, AVG]
ORDER BY column1 [ASC | DESC]
UNION | INTERSECT | EXCEPT
[SELECT column1 FROM another_table]
LIMIT row_count;
```

Sintaxis Stored Procedure

```
CREATE OR REPLACE FUNCTION <nombre_función> (<argumentos>)
RETURNS [VOID | TABLE(...) ] AS $$
DECLARE
    <declaración_variables>
BEGIN
    <sentencias_SQL>
END; $$ language plpgsql
```

Sintaxis Trigger

```
CREATE TRIGGER <nombre_trigger>
[BEFORE | AFTER | INSTEAD OF] [ DELETE | UPDATE | INSERT] ON <tabla>
[FOR EACH ROW | FOR EACH STATEMENT]
BEGIN
    UPDATE <tabla_a_actualizar>
    SET <atributo> = <valor>
    WHERE <condición>
END;
```

Sintaxis View

```
CREATE VIEW <nombre_view> AS (
    <sentencias_SQL>
);
```

Sintaxis PL/pgSQL

```
BEGIN
    IF <condición> THEN
        <cuerpo_if>
    ELSE
        <cuerpo_else>
    END IF;
    INSERT INTO <tabla> VALUES (<tupla_valores>);
    FOR <variable> IN <iterable>
    LOOP
        <cuerpo_loop>
    END LOOP;
    RETURN QUERY <consulta_SQL>;
END;
```