

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

Interrogación 1

Fecha: 8 de mayo de 2026

1er semestre 2026 - Profesores: Eduardo Bustos - Christian Álvarez

---

INSTRUCCIONES

- Al finalizar la prueba debe **digitalizarla y subirla a Canvas** correctamente. No seguir esta instrucción los expone a un descuento de hasta 5 décimas.
- Cada hoja de respuesta debe contener su **nombre completo** y **número de lista** en la parte superior.
- Responda cada pregunta en una hoja diferente.
- Debe firmar la lista de asistencia.
- Duración 2 horas.

Pregunta 1 - Álgebra Relacional

La Fórmula 1 es la principal competencia automovilística del mundo, en ella participan equipos llamados **Escuderías**, en cada escudería corren **Pilotos**. Se realizan 24 rondas o **Carreras**, en cada una tanto las escuderías como los pilotos, van sumando puntos de acuerdo a la posición en que llegaron en cada carrera, solo los 10 primeros lugares dan puntaje. Sean las siguientes relaciones, que contienen información sobre el campeonato: **Pilotos** Contiene los datos del piloto y la escudería a la que pertenece, **Carreras** contiene el nombre y lugar donde se realiza la carrera,**Escuderías** contiene el nombre de la escudería y el país al que pertenece, **Resultados** regista la posición en la que el piloto terminó la carrera, los puntos obtenidos en ella y si dió la vuelta más rápida de la carrera.

En el siguiente esquema (las llaves primarias se encuentran **subrayadas**):

- **Pilotos**(IDP, Nombre, Nacionalidad, IDE)
- **Carreras**(IDC, NombreGP, Pais, Ciudad)
- **Resultados**(IDP, IDC, PosicionFinal, PuntosObtenidos, VueltaRapida)
- **Escuderias**(IDE, NombreE, PaisBase)

A continuación se muestran **ejemplos** de tuplas en las 4 tablas:

IDE	**NombreE**	**PaisBase**
RBR	Oracle Red Bull Racing	Reino Unido
FER	Scuderia Ferrari	Italia
AMF	Aston Martin Aramco	Reino Unido
AUD	Audi F1 Team	Alemania
MER	Mercedes-AMG PETRONAS	Reino Unido
...	...	...

Cuadro 1: Escuderías

IDP	**Nombre**	**Nacionalidad**	**IDE**
VER	Max Verstappen	Neerlandés	RBR
HAM	Lewis Hamilton	Británico	FER
LEC	Charles Leclerc	Monegasco	FER
ALO	Fernando Alonso	Español	AMF
HUL	Nico Hülkenberg	Alemán	AUD
...	...	..	...

Cuadro 2: Pilotos

IDC	NombreGP	Pais	Ciudad
C01	GP de Australia	Australia	Melbourne
C02	GP de España	España	Madrid
C03	GP de Italia	Italia	Monza
C04	GP de Mónaco	Mónaco	Montecarlo
C05	GP de Gran Bretaña	Reino Unido	Silverstone
...	...	...	...

Cuadro 3: Carreras

IDP	IDC	PosicionFinal	PuntosObtenidos	VueltaRapida
HAM	C01	1	25	0
HAM	C02	1	26	1
HAM	C05	4	12	0
VER	C01	2	18	1
VER	C05	1	25	0
ALO	C02	3	15	0
...	...	...	...	...

Cuadro 4: Resultados

Escriba las siguientes operaciones en álgebra relacional, utilizando la sintaxis vista en clases (puede encontrarla en el torpedo):

a) (3 pts) Obtener el nombre y la nacionalidad de aquellos pilotos que estuvieron en el podio (PosicionFinal ≤ 3) en el GP de Miami.

b) (3 pts) Encontrar el ID y nombre de los pilotos que han participado en al menos una carrera, pero que nunca han obtenido una vuelta rápida (VueltaRapida $= 1$ significa que el piloto realizó la vuelta más rápida en la carrera).

c) (4 pts) Obtener los IDs de los pilotos que han ganado al menos 2 carreras. Nota: no se pueden usar contadores ni truncar

d) (5 pts) Obtener el nombre de las escuderías que han logrado que todos sus pilotos terminen en el podio (PosicionFinal ≤ 3) en todas las carreras.

Pregunta 2 - Diagrama E/R y Esquema Relacional

El refugio *Patitas a la Obra* necesita modernizar su sistema de información. El refugio se encarga de rescatar animales, cuidarlos y gestionar su adopción responsable.

De cada animal se registra un código único, nombre, especie (perro, gato, ave, etc.), raza, sexo, fecha de nacimiento estimada y la fecha en que ingresó al refugio.

- **Especialización:** Si el animal es un **perro**, se debe registrar si conoce comandos básicos de obediencia. Si es un **gato**, se debe registrar si es positivo a Leucemia o SIDA felino.
- **Salud:** Se debe llevar un registro de las vacunas aplicadas. Una vacuna se identifica por su nombre (ej: "Séxtuple", "Antirrábica") y la fecha de aplicación. Una misma vacuna puede aplicarse varias veces al mismo animal (refuerzos).

Los animales son traídos por rescatistas. De ellos se registra su RUT, nombre, teléfono y dirección. Un animal puede ser traído por uno o más rescatistas en conjunto, y un rescatista puede haber traído muchos animales.

Existen casas de acogida donde los animales pasan un tiempo hasta que sean adoptados. De estas casas se registra el RUT del dueño, dirección y capacidad máxima. Se debe registrar el periodo (fecha inicio y fecha término) que un animal pasa en una casa de acogida específica. Los adoptantes se registran con su RUT, nombre y calificación de confiabilidad. Un adoptante puede adoptar a varios animales, pero cada animal solo puede ser adoptado por una única persona. Se debe registrar la fecha de la adopción y el monto de la donación realizada.

a) (10 pts) Diseñe el Diagrama Entidad/Relación de *Patitas a la Obra*. Identifique claramente las entidades, entidades débiles, relaciones, cardinalidades, atributos, y llaves primarias y parciales.

b) (5 pts) A partir del Diagrama E/R propuesto, genere el Esquema Relacional asociado. Para cada tabla, indique los atributos, sus dominios, las llaves primarias, foráneas, y cualquier otra aclaración respecto a la representación de sus relaciones.

Pregunta 3 - SQL y Lógica en la BD

Considere el esquema de la aerolínea *FlyPoint* que va a continuación.

Listing 1: Definición de tablas SQL

```sql
CREATE TABLE Pasajeros (
    rut VARCHAR(12) PRIMARY KEY,
    nombre VARCHAR(100),
    millas_acumuladas INT DEFAULT 0,
    categoria VARCHAR(20) -- 'Bronce', 'Plata', 'Oro'
);

CREATE TABLE Vuelos (
    id_vuelo INT PRIMARY KEY,
    origen VARCHAR(50),
    destino VARCHAR(50),
    distancia_millas INT,
    estado VARCHAR(20) -- 'Programado', 'En Vuelo', 'Aterrizado',
    'Cancelado'
);

CREATE TABLE Reservas (
    id_reserva INT PRIMARY KEY,
    rut_pasajero VARCHAR(12),
    id_vuelo INT,
    millas_gastadas INT,
    estado_checkin BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (rut_pasajero) REFERENCES Pasajeros(rut),
    FOREIGN KEY (id_vuelo) REFERENCES Vuelos(id_vuelo)
);

CREATE TABLE AuditoriaMillas (
    id_log SERIAL PRIMARY KEY,
    rut_pasajero VARCHAR(12),
    operacion VARCHAR(20),
    cantidad INT,
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

a) (5 pts) Cree un procedimiento llamado `sp_cancelar_vuelo(id_vuelo_param INT)`. Este procedimiento debe:

- Cambiar el estado del vuelo a 'Cancelado'.
- Identificar a todos los pasajeros que tenían una reserva en dicho vuelo.

- Por cada pasajero que usó millas para pagar (suponga que las millas se restaron previamente), estas deben devolverse. Para efectos de este ejercicio, sume un bono de 500 millas a cada pasajero afectado por la cancelación.
- Registrar la devolución en la tabla `AuditoriaMillas`.

Nota: La tabla `AuditoriaMillas` actúa como un *log* de transacciones. Al implementar la lógica de cancelación, es obligatorio que por cada pasajero afectado se realice un `INSERT` en esta tabla. Por ejemplo, si al pasajero con RUT `'12.345.678-9'` se le devuelven 3.500 millas, la tabla de auditoría deberá registrar una nueva entrada como la siguiente:

id_log	**rut_pasajero**	**operacion**	**cantidad**	**fecha**
1024	12.345.678-9	DEVOLUCION_CANCEL	3500	2026-05-06 15:20:00

b) (5 pts) Cree un trigger llamado `tr_vuelo_aterrizado` que se active cuando el estado de un vuelo cambie a 'Aterrizado'. El trigger debe:

- Buscar a todos los pasajeros que realizaron el `checkin` correctamente para ese vuelo específico.
- Sumar la `distancia_millas` del vuelo a las `millas_acumuladas` del pasajero.
- **Regla de Categoría:** Si el pasajero es categoría 'Oro', suma 2 veces las millas del vuelo. Si es 'Plata', suma 1.5 veces las millas del vuelo . Si es 'Bronce', suma sólo las millas del vuelo.
- Insertar el registro correspondiente en `AuditoriaMillas`.

c) (5 pts) Cree un trigger que impida realizar un `checkin` (`estado_checkin = TRUE`) si el vuelo asociado tiene un estado distinto a 'Programado'. El sistema debe lanzar un error personalizado.

Pregunta 4 - Formas Normales

La aerolínea *FlyPoint* ha estado registrando la asignación de personal a los vuelos en una planilla única de Excel. Usted ha sido contratado para migrar estos datos a un modelo relacional robusto.

A continuación, se presenta una muestra de la tabla desnormalizada **ASIGNACIONES**:

ID_Vuelo	**ID_Emp**	**Nombre_Emp**	**Rol**	**Avion_Reg**	**Modelo**	**Capacidad**
V101	E50	Juan Soto	Piloto	CC-ABC	Airbus A320	180
V101	E62	Ana Luz	Copiloto	CC-ABC	Airbus A320	180
V102	E50	Juan Soto	Piloto	CC-DEF	Boeing 737	160
V205	E88	Luis Jara	Tripulante	CC-GHI	Airbus A321	220

Supuestos adicionales:

- Un vuelo tiene un único avión asignado.
- Un empleado tiene un único nombre y un rol fijo en la empresa.
- Cada matrícula de avión (`Avion_Reg`) corresponde a un único modelo.
- La capacidad depende exclusivamente del modelo del avión.

Se solicita:

a) (3 pts) Identifique la clave primaria de la tabla **ASIGNACIONES** y todas las Dependencias Funcionales (DF).

b) (4 pts) Explique por qué la tabla **no** está en Segunda Forma Normal (2NF).

c) (5 pts) Proponga un esquema descompuesto en Tercera Forma Normal (3NF).

d) (3 pts) ¿Se encuentra su propuesta en Forma Normal de Boyce-Codd (BCNF)? Justifique.

Bonus

a) (1 pt.) ¿Qué significa que un itinerario (schedule) sea Conflict Serializable?

b) (1 pt.) ¿Cuál es la diferencia entre Atomicidad y Consistencia?

c) (1 pt.) Si un itinerario es serial, ¿podemos asegurar que no presenta conflictos de lectura/escritura?

d) (1 pt.)Explique cómo se produce una long transaction y en qué afecta al DBMS.

e) (1 pt.) Explique en qué consiste un deadlock y en que afecta al sistema.

Torpedo

Álgebra Relacional

- **Sintaxis general:**

$\pi_{a_1,a_2,\ldots,a_n}\left(\sigma_{\text{condition1}[\wedge\,|\,\vee\,\text{condition2}]\ [<,\leq,>,\geq,=,\neq]}\left(R \bowtie_{\text{condition}} S\right)\right)$

$[\times, \cup, \cap, -]$

$\pi_{b_1,b_2,\ldots,b_n}\left(\sigma_{\text{condition}}(\rho((a_1 \to b_1, a_2 \to b_2, \ldots, a_n \to b_n), T))\right)$

SQL

- **Sintaxis general para la creación de una tabla:**

```
CREATE TABLE <Nombre> (<atr1> <tipo> PRIMARY KEY, <atr2> <tipo>
DEFAULT <valor> NOT NULL UNIQUE, <atr3> <tipo>, FOREIGN KEY
(<atr3>) REFERENCES <NombreA>(<atrA>), ...);
```

- **Sintaxis general para realizar una consulta:**

```
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 attribute [condition1 AND | OR condition2]
      [BETWEEN value1 AND value2]
      [IN (value1, value2, ...)]
      [LIKE pattern]
      [= | <> | <= | >= | < | >]
GROUP BY column1, column2, ...
HAVING aggregate_condition [COUNT, MAX, MIN, SUM, AVG]
ORDER BY column1 [ASC | DESC], column2
UNION | INTERSECT | EXCEPT
[SELECT column1 FROM another_table]
LIMIT rowCount;
```

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;
```