



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

BD-RA1: Reconoce los elementos de las bases de datos analizando sus funciones y valorando la utilidad de los sistemas gestores.

1 ¿Cuál es una de las principales limitaciones de la gestión de información mediante ficheros en comparación con las bases de datos?

- a. No permiten almacenar datos binarios como imágenes o audio.
- b. El acceso a la información es siempre más lento que en una base de datos.
- c. Presentan dificultades significativas en el manejo de la concurrencia (acceso de múltiples usuarios simultáneos) y la integridad.
- d. Obligan a utilizar estructuras de datos extremadamente complejas (por ejemplo, arrays)

2 En el modelo relacional, ¿cómo se organiza y relaciona la información?

- a. Mediante una estructura de árbol con relaciones de un único "padre" y varios "hijos".
- b. Usando tablas bidimensionales (filas y columnas) relacionadas mediante claves primarias y ajenas (foráneas).
- c. Almacenando exclusivamente objetos que contienen tanto datos como métodos.
- d. Utilizando grafos con conexiones múltiples sin una estructura fija.

3 ¿Qué caracteriza fundamentalmente a una base de datos centralizada?

- a. La información se reparte en varios servidores conectados por una red para mejorar la disponibilidad.
- b. Toda la información se almacena en un único equipo o servidor, lo que simplifica su gestión y seguridad.
- c. Requiere obligatoriamente políticas de fragmentación horizontal y vertical.
- d. Es el modelo ideal para grandes empresas con sedes en diferentes continentes que requieren acceso en tiempo real.

4 ¿Cuál es una de las utilidades esenciales que aporta un SGBD a una organización?

- a. Eliminar la necesidad de realizar copias de seguridad externas.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

- b. Sustituir por completo al sistema operativo en la gestión del hardware.
- c. Garantizar la integridad, seguridad y disponibilidad de la información, liberando a los usuarios de tareas técnicas complejas de almacenamiento físico.
- d. Permitir únicamente el acceso monousuario para evitar conflictos de datos.

5 Dentro de la arquitectura de un SGBD, ¿de qué se encarga específicamente el gestor de transacciones?

- a. De interpretar y optimizar las consultas escritas en lenguaje SQL.
- b. De definir los roles y privilegios de los usuarios que acceden al sistema.
- c. De controlar las operaciones que deben ejecutarse de forma conjunta y atómica para garantizar las propiedades ACID.
- d. De gestionar físicamente la lectura y escritura de datos en el disco duro.

6 Si clasificamos un SGBD según el número de usuarios a los que da servicio simultáneamente, ¿qué categorías encontramos?

- a. Relacionales y NoSQL.
- b. Monousuario y multiusuario.
- c. Centralizados y distribuidos.
- d. SQL y NoSQL.

7 ¿Cuál es una ventaja clave de utilizar una base de datos distribuida en una empresa con múltiples sedes?

- a. La reducción de la complejidad en la sincronización de los datos.
- b. La alta disponibilidad y tolerancia a fallos.
- c. Que el coste de mantenimiento es significativamente inferior a una centralizada.
- d. Que no requiere el uso de redes de comunicación entre nodos.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

8 ¿Qué caracteriza al concepto de Big Data?

- a. El uso exclusivo de bases de datos relacionales tradicionales.
- b. Las denominadas "Vs": Volumen, Velocidad, Variedad, Veracidad y Valor.
- c. La incapacidad total de procesar datos en tiempo real.
- d. Que solo es aplicable en el sector de las redes sociales.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

BD-RA2: Crea bases de datos definiendo su estructura y las características de sus elementos según el modelo relacional.

9 Para almacenar precios, con dos decimales, cuyo valor máximo es 9999.99, ¿qué definición es más adecuada en PostgreSQL?

- a. REAL(4,2)
- b. FLOAT(4,2)
- c. DECIMAL(6,2)
- d. MONEY

10 Se desea crear una tabla llamada detalle_pedido, cuya clave primaria esté compuesta por pedido_id y producto_id. ¿Qué definición es correcta?

- a. CREATE TABLE detalle_pedido (
 pedido_id INTEGER NOT NULL UNIQUE,
 producto_id INTEGER NOT NULL UNIQUE,
 cantidad INTEGER NOT NULL CHECK (cantidad > 0));
- b. CREATE TABLE detalle_pedido (
 pedido_id INTEGER PRIMARY KEY,
 producto_id INTEGER PRIMARY KEY,
 cantidad INTEGER NOT NULL CHECK (cantidad > 0));
- c. CREATE TABLE detalle_pedido (
 pedido_id INTEGER NOT NULL,
 producto_id INTEGER NOT NULL,
 cantidad INTEGER NOT NULL CHECK (cantidad > 0),
 PRIMARY KEY (pedido_id, producto_id));
- d. CREATE TABLE detalle_pedido (
 pedido_id INTEGER,
 producto_id INTEGER,



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

cantidad INTEGER NOT NULL CHECK (cantidad > 0),

UNIQUE (pedido_id, producto_id)

NOT NULL(pedido_id, producto_id));

11 Se desea crear la tabla pedidos con una clave foránea a clientes (la clave primaria de la tabla clientes es la columna "id"). Si se borra un cliente, se deben eliminar todos sus pedidos. ¿Qué opción es correcta?

a. CREATE TABLE pedidos (

id SERIAL PRIMARY KEY,

cliente_id INTEGER REFERENCES clientes(id),

fecha DATE NOT NULL);

b. CREATE TABLE pedidos (

id SERIAL PRIMARY KEY,

cliente_id INTEGER NOT NULL,

fecha DATE NOT NULL,

FOREIGN KEY (cliente_id) REFERENCES clientes(id) ON DELETE CASCADE);

c. CREATE TABLE pedidos (

id SERIAL PRIMARY KEY,

cliente_id INTEGER NOT NULL,

fecha DATE NOT NULL,

FOREIGN KEY (id) REFERENCES clientes(cliente_id) ON DELETE CASCADE);

d. CREATE TABLE pedidos (

id SERIAL PRIMARY KEY,

cliente_id INTEGER NOT NULL FOREIGN KEY(clientes),

fecha DATE NOT NULL,

ON DELETE(clientes) CASCADE(id));



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

12 Se quiere que el estado de un pedido admita únicamente pendiente, enviado o cancelado, con valor por defecto pendiente. ¿Qué secuencia de definiciones es correcta?

a. CREATE TYPE estado_pedido AS ENUM ('pendiente','enviado','cancelado');

CREATE TABLE pedidos (

id SERIAL PRIMARY KEY,

estado estado_pedido NOT NULL DEFAULT 'pendiente');

b. CREATE TABLE pedidos (

id SERIAL PRIMARY KEY,

estado VARCHAR(10)

CHECK(estado IN ('pendiente','enviado','cancelado')) DEFAULT 'pendiente');

c. CREATE TABLE pedidos (

id SERIAL PRIMARY KEY,

estado TEXT NOT NULL DEFAULT pendiente);

d. Las opciones a) y b) son válidas ambas.

13 Se desea crear el usuario appuser para trabajar (realizar operaciones CRUD) con tablas del esquema public, incluidas columnas SERIAL. ¿Qué conjunto de privilegios es mínimo y correcto?

a. CREATE USER appuser WITH PASSWORD 'pwd';

GRANT USAGE ON SCHEMA public TO appuser;

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO appuser;

GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO appuser;

b. CREATE USER appuser WITH PASSWORD 'pwd';

GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO appuser;

c. CREATE USER appuser WITH PASSWORD 'pwd';



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

GRANT USAGE ON SCHEMA public TO appuser;

GRANT SELECT ON ALL TABLES IN SCHEMA public TO appuser;

d. CREATE USER appuser WITH PASSWORD 'pwd';

GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA public TO appuser;

14 ¿Qué tipo es el más adecuado para almacenar una fecha y hora?

a. TIMESTAMP

b. TIMEDATE

c. DATE_TIME

d. b) y c) son ambas correctas aunque se recomienda DATE_TIME por optimización.

15 ¿Se desea permitir al usuario report_user ejecutar la consulta de la vista v_totales_pedido (el propietario de la vista ya tiene permisos sobre las tablas subyacentes). ¿Qué sentencia es correcta?

a. GRANT SELECT ON v_totales_pedido TO report_user;

b. GRANT USAGE ON SCHEMA public TO report_user;

c. GRANT EXECUTE ON v_totales_pedido TO report_user;

d. GRANT SELECT ON ALL TABLES IN SCHEMA public TO report_user;

16 Se tienen dos tablas: pedido y detalle_pedido (con la información, líneas, del pedido). Se realizan consultas frecuentes que unen la tabla de pedidos con la de detalle_pedido mediante detalle_pedido.pedido_id = pedido.id. ¿Cuál es el índice mínimo y más apropiado para acelerar este JOIN?

a. CREATE INDEX idx_pedidos_id ON pedidos(id);

b. CREATE INDEX idx_dp_pedidos ON detalle_pedido(id);

c. CREATE INDEX idx_dp_pedido ON detalle_pedido(pedido_id);

d. CREATE INDEX idx_dp_multi ON detalle_pedido(pedido_id, producto_id, cantidad);



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

17 ¿En la tabla pagos(metodo, tarjeta_num, importe), se quiere garantizar que si el método de pago es 'tarjeta', entonces tarjeta_num no sea nulo. ¿Qué sentencia es correcta?

a. ALTER TABLE pagos

ADD CONSTRAINT ck_tarjeta_num_requerido

CHECK (metodo = 'tarjeta' AND tarjeta_num NOT NULL);

b. ALTER TABLE pagos

ADD CONSTRAINT ck_tarjeta_num_requerido

CHECK (COALESCE(tarjeta_num, '') <> '' WHEN metodo = 'tarjeta');

c. ALTER TABLE pagos

ADD CONSTRAINT ck_tarjeta_num_requerido

CHECK (metodo IN ('tarjeta') -> tarjeta_num IS NOT NULL);

d. ALTER TABLE pagos

ADD CONSTRAINT ck_tarjeta_num_requerido

CHECK (metodo <> 'tarjeta' OR tarjeta_num IS NOT NULL);

18 ¿Cuál de las siguientes afirmaciones describe correctamente la diferencia entre CHAR(n) y VARCHAR(n)?

a. VARCHAR(n) rellena con espacios hasta n, mientras que CHAR(n) no rellena.

b. CHAR(n) es de longitud fija y VARCHAR(n) es de longitud variable.

c. CHAR(n) sin especificar n permite cadenas de cualquier longitud (equivalente a TEXT).

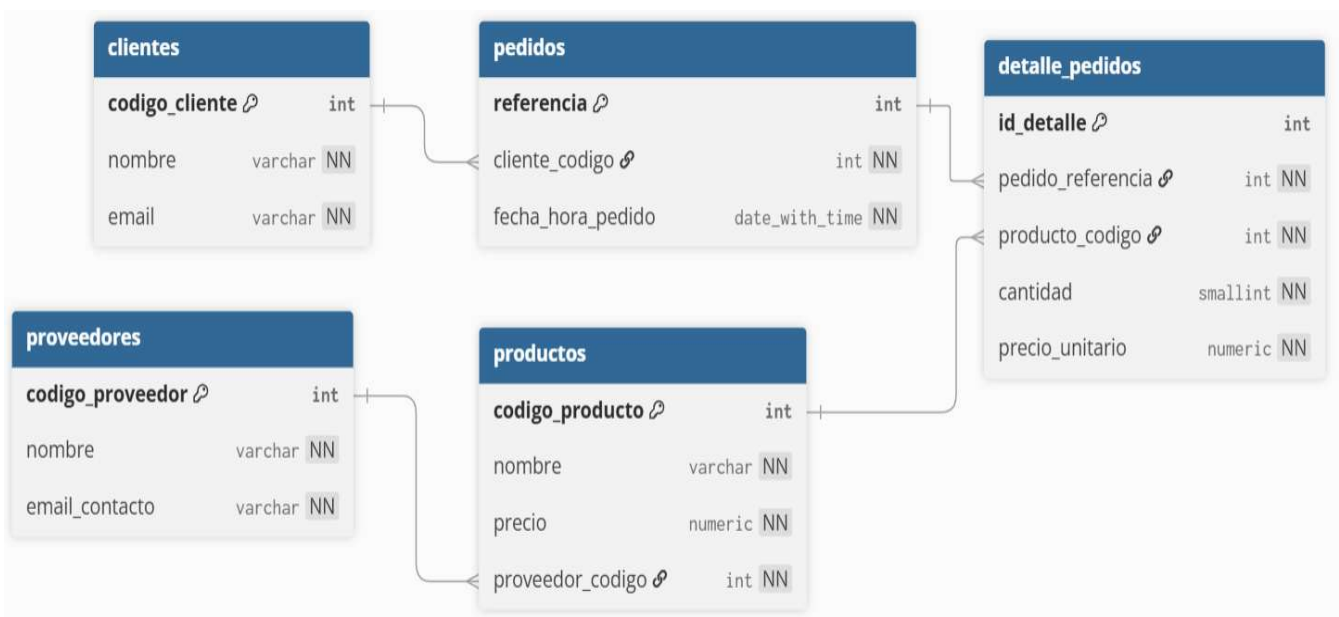
d. Ninguna de las anteriores respuestas es correcta.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

BD-RA3: Consulta la información almacenada en una base de datos empleando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.

Dado el siguiente diagrama relacional y las características que se indican, responde a las siguientes preguntas.



- El contexto es la gestión de pedidos pudiendo haber en cada pedido más de un producto.
- El "precio" de la tabla productos es el precio actual del producto.
- El "precio_unitario" de la tabla detalle_pedidos es el precio que tiene un producto en ese pedido en concreto.
- El campo fecha_hora_pedido almacena la fecha y la hora en la que se realiza el pedido.
- El símbolo de la llave indica que la columna forma parte de la clave primaria de la tabla.
- Las columnas marcadas con NN no permiten valores nulos

19 Indica qué consulta selecciona los emails de los clientes cuyo nombre comienza por 'A' y el nombre tiene, al menos, dos letras. Los emails deben aparecer ordenados alfabéticamente de la Z a la A. Por ejemplo, zaragoza@correo.edu debe aparecer antes que alicante@correo.edu

- SELECT email FROM clientes WHERE nombre LIKE 'A%' ORDER BY email ASC;
- SELECT email FROM clientes WHERE nombre LIKE 'A_%' ORDER BY email DESC;



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

c. SELECT email FROM clientes WHERE nombre LIKE '_A%' ORDER BY 1 DESC;

d. SELECT codigo_cliente FROM clientes WHERE nombre LIKE 'A_%' ORDER BY email ASC;

20 ¿Qué consulta obtiene los nombres de clientes que han realizado al menos un pedido (sin duplicados)?

a. SELECT c.nombre

FROM clientes c

INNER JOIN pedidos p ON p.cliente_codigo = c.codigo_cliente;

b. SELECT DISTINCT c.nombre

FROM clientes c

LEFT JOIN pedidos p ON p.cliente_codigo = c.codigo_cliente

WHERE p.referencia IS NULL;

c. SELECT DISTINCT c.nombre

FROM clientes c

INNER JOIN pedidos p ON p.referencia = c.codigo_cliente;

d. SELECT DISTINCT c.nombre

FROM clientes c

INNER JOIN pedidos p ON p.cliente_codigo = c.codigo_cliente;

21 ¿Qué consulta muestra el código y nombre de los clientes que han hecho más de 5 pedidos?

a. SELECT c.codigo_cliente, c.nombre

FROM clientes c

WHERE (

SELECT SUM(dp.cantidad)

FROM pedidos p



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

INNER JOIN detalle_pedidos dp ON dp.pedido_referencia = p.referencia

WHERE p.cliente_codigo = c.codigo_cliente) > 5;

b. SELECT c.codigo_cliente, c.nombre

FROM clientes c

WHERE (

SELECT AVG(dp.cantidad)

FROM pedidos p

INNER JOIN detalle_pedidos dp ON dp.pedido_referencia = p.referencia

WHERE p.cliente_codigo = c.codigo_cliente) > 5;

c. SELECT c.codigo_cliente, c.nombre

FROM clientes c

WHERE (

SELECT COUNT(*)

FROM pedidos p

WHERE p.cliente_codigo = c.codigo_cliente) > 5;

d. SELECT c.codigo_cliente, c.nombre

FROM clientes c, pedidos p

GROUP BY c.codigo_cliente, c.nombre

HAVING COUNT(*) >= 5;

22 ¿Qué consulta permite mostrar la referencia del pedido y el importe total de cada pedido?

a. SELECT dp.pedido_referencia AS pedido,

SUM(dp.cantidad * dp.precio_unitario) AS importe_total

FROM detalle_pedidos dp

GROUP BY dp.pedido_referencia;



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

b. SELECT pe.referencia, SUM(dp.precio_unitario)

FROM pedidos pe

LEFT JOIN detalle_pedidos dp ON dp.pedido_referencia = pe.referencia;

c. SELECT dp.pedido_referencia, COUNT(DISTINCT dp.cantidad * dp.precio_unitario)

FROM detalle_pedidos dp

GROUP BY dp.pedido_referencia;

d. Ninguna de las anteriores respuestas es correcta.

23 ¿Qué consulta permite obtener una lista con el código de cliente, su nombre y el número de pedidos que ha realizado? Se deben mostrar también los clientes que no tienen pedidos

a. SELECT c.codigo_cliente, c.nombre, COUNT(p.referencia) AS total_pedidos

FROM clientes c

INNER JOIN pedidos p ON p.cliente_codigo = c.codigo_cliente

GROUP BY c.codigo_cliente, c.nombre

ORDER BY c.codigo_cliente;

b. SELECT c.codigo_cliente, c.nombre, SUM(p.referencia) AS total_pedidos

FROM clientes c

RIGHT JOIN pedidos p ON p.cliente_codigo = c.codigo_cliente

WHERE p.referencia IS NOT NULL

GROUP BY c.codigo_cliente, c.nombre

ORDER BY c.codigo_cliente;

c. SELECT c.codigo_cliente, c.nombre, COUNT(p.referencia) AS total_pedidos

FROM clientes c

LEFT JOIN pedidos p ON p.cliente_codigo = c.codigo_cliente

GROUP BY c.codigo_cliente, c.nombre



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

ORDER BY c.codigo_cliente;

d. SELECT c.codigo_cliente, c.nombre, SUM(p.referencia) AS total_pedidos

FROM clientes c

INNER JOIN pedidos p ON p.cliente_codigo = c.codigo_cliente

GROUP BY c.codigo_cliente

ORDER BY c.codigo_cliente;

24 ¿Qué consulta permite obtener el código y nombre de los productos que no han sido pedidos nunca?

a. SELECT pr.codigo_producto, pr.nombre

FROM productos pr

LEFT JOIN detalle_pedidos dp ON dp.producto_codigo = pr.codigo_producto

WHERE dp.producto_codigo = NULL;

b. SELECT pr.codigo_producto, pr.nombre

FROM detalle_pedidos dp

RIGHT JOIN productos pr ON dp.producto_codigo = pr.codigo_producto

WHERE dp.producto_codigo IS NULL;

c. SELECT pr.codigo_producto, pr.nombre

FROM detalle_pedidos dp

INNER JOIN productos pr ON dp.producto_codigo = pr.codigo_producto

WHERE pr.codigo_producto = NULL;

d. SELECT pr.codigo_producto, pr.nombre

FROM detalle_pedidos dp

LEFT JOIN productos pr



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

ON dp.producto_codigo = pr.codigo_producto

WHERE pr.codigo_producto IS NULL;

25 ¿Qué consulta muestra la referencia e importe total de los 3 pedidos más caros (los de mayor importe)?

a. SELECT pe.referencia AS pedido, SUM(dp.precio_unitario) AS importe_total

FROM pedidos pe

INNER JOIN detalle_pedidos dp ON dp.pedido_referencia = pe.referencia

GROUP BY pe.referencia

ORDER BY importe_total ASC TAKE 3;

b. SELECT dp.pedido_referencia AS pedido,

SUM(dp.cantidad * dp.precio_unitario) AS importe_total

FROM detalle_pedidos dp

GROUP BY dp.pedido_referencia

ORDER BY importe_total DESC

LIMIT 3;

c. SELECT dp.pedido_referencia AS pedido,

COUNT(dp.cantidad * dp.precio_unitario) AS importe_total

FROM detalle_pedidos dp

GROUP BY dp.pedido_referencia

ORDER BY importe_total ASC

LIMIT 3;

d. SELECT dp.pedido_referencia AS pedido,

COUNT(dp.cantidad * dp.precio_unitario) AS importe_total

FROM detalle_pedidos dp



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

ORDER BY importe_total DESC

LIMIT 3;

26 ¿Qué consulta muestra el código y nombre de los productos que han sido vendidos en los últimos 90 días (al menos una línea/fila de detalle_pedidos en ese periodo)?

a. SELECT pr.codigo_producto, pr.nombre

FROM productos pr

WHERE pr.codigo_producto IN (

SELECT dp.producto_codigo

FROM detalle_pedidos dp

INNER JOIN pedidos pe ON pe.referencia = dp.pedido_referencia);

b. SELECT pr.codigo_producto, pr.nombre

FROM productos pr

WHERE EXISTS (

SELECT 1

FROM detalle_pedidos dp

INNER JOIN pedidos pe ON pe.referencia = dp.pedido_referencia

WHERE pe.fecha_hora_pedido < NOW() - INTERVAL '90 days');

c. SELECT pr.codigo_producto, pr.nombre

FROM productos pr

WHERE EXISTS (

SELECT 1

FROM detalle_pedidos dp

WHERE dp.pedido_referencia = pr.codigo_producto);

d. SELECT pr.codigo_producto, pr.nombre

FROM productos pr



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

WHERE EXISTS (

SELECT 1

FROM detalle_pedidos dp

INNER JOIN pedidos pe ON pe.referencia = dp.pedido_referencia

WHERE dp.producto_codigo = pr.codigo_producto

AND pe.fecha_hora_pedido >= NOW() - INTERVAL '90 days');

27 ¿Qué consulta muestra los proveedores que tienen, al menos, 2 productos y que el precio medio de sus productos es mayor que 50?

a. SELECT pr.codigo_proveedor, pr.nombre

FROM proveedores pr

INNER JOIN productos pd ON pd.proveedor_codigo = pr.codigo_proveedor

GROUP BY pr.codigo_proveedor, pr.nombre

HAVING COUNT(pd.codigo_producto) > 1 AND AVG(pd.precio) > 50

ORDER BY pr.codigo_proveedor;

b. SELECT pr.codigo_proveedor, pr.nombre

FROM proveedores pr

LEFT JOIN productos pd ON pd.proveedor_codigo = pr.codigo_proveedor

GROUP BY pr.codigo_proveedor, pr.nombre

WHERE COUNT(pd.codigo_producto) >= 2 AND AVG(pd.precio) > 50;

c. SELECT pr.codigo_proveedor, pr.nombre

FROM proveedores pr

INNER JOIN productos pd ON pd.proveedor_codigo = pr.codigo_proveedor

WHERE COUNT(pd.codigo_producto) > 1 AND AVG(pd.precio) > 50

GROUP BY pr.codigo_proveedor, pr.nombre;



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

d. SELECT pr.codigo_proveedor, pr.nombre

FROM proveedores pr

OUTER JOIN productos pd ON pd.proveedor_codigo = pr.codigo_proveedor

GROUP BY pr.nombre

HAVING COUNT() >= 2 AND AVG(DISTINCT precio) > 50;

28 ¿Qué consulta selecciona nombre del producto, nombre del cliente y la cantidad de cada línea de pedido (detalle_pedidos)?

a. SELECT pr.nombre AS producto, c.nombre AS cliente, dp.cantidad

FROM detalle_pedidos dp

INNER JOIN productos pr ON pr.codigo_producto = dp.producto_codigo

INNER JOIN pedidos pe ON pe.referencia = dp.pedido_referencia

INNER JOIN clientes c ON c.codigo_cliente = pe.cliente_codigo;

b. SELECT pr.nombre, c.nombre, dp.cantidad

FROM detalle_pedidos dp

INNER JOIN pedidos pe ON pe.referencia = dp.pedido_referencia

INNER JOIN productos pr ON pr.codigo_producto = dp.producto_codigo

INNER JOIN clientes c ON c.codigo_cliente = pe.cliente_codigo

GROUP BY pe.referencia;

c. SELECT pr.nombre, c.nombre, dp.cantidad

FROM productos pr, detalle_pedidos dp, pedidos pe, clientes c

WHERE pr.codigo_producto = dp.producto_codigo

AND pe.cliente_codigo = c.codigo_cliente;

d. Ninguna respuesta es correcta.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

29 Suponiendo que en la tabla de Pedidos existe una columna estado (con posibles valores pendiente, enviado o cancelado), se desea obtener un listado que clasifique los pedidos en activos (pendiente, enviado) e inactivos (cancelado), mostrando las columnas referencia, estado y una columna categoría con los literales 'activo' o 'inactivo'. ¿Cuál de las siguientes consultas SQL produce exactamente ese resultado?

- a. `SELECT referencia, 'activo' AS categoria FROM pedidos`
`WHERE estado IN ('pendiente','enviado')`
`UNION ALL`
`SELECT referencia, 'inactivo' AS categoria FROM pedidos`
`WHERE estado = 'cancelado';`
- b. `SELECT referencia, estado, 'activo' AS categoria FROM pedidos`
`WHERE estado IN ('pendiente','enviado')`
`UNION`
`SELECT referencia, 'inactivo' AS categoria FROM pedidos`
`WHERE estado = 'cancelado';`
- c. `SELECT referencia, estado, 'activo' AS categoria FROM pedidos`
`WHERE estado IN ('pendiente','enviado')`
`UNION`
`SELECT referencia, estado, 'inactivo' AS categoria FROM pedidos`
`WHERE estado = 'cancelado';`
- d. `(SELECT referencia, estado FROM pedidos`
`WHERE estado IN ('pendiente','enviado'))`
`UNION`
`(SELECT referencia, estado, 'inactivo' FROM pedidos`
`WHERE estado = 'cancelado');`



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

30 Se quieren listar los pedidos del día de hoy. ¿Qué condición de filtrado permite hacerlo aprovechando un índice sobre pedidos(fecha_hora_pedido)?

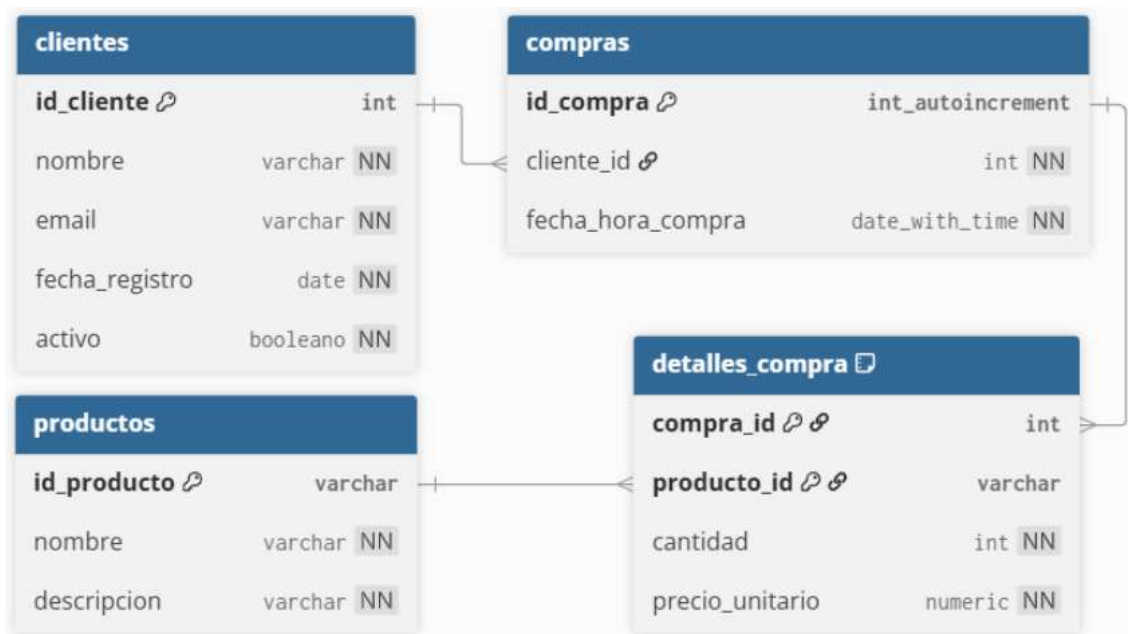
- a. WHERE DATE(pe.fecha_hora_pedido + INTERVAL '1 day') = CURRENT_DATE;
- b. WHERE pe.fecha_hora_pedido >= CURRENT_DATE
AND pe.fecha_hora_pedido < CURRENT_DATE + INTERVAL '1 day';
- c. WHERE CAST(pe.fecha_hora_pedido, 'YYYY-MM-DD') = CAST(NOW(), 'YYYY-MM-DD');
- d. WHERE EXTRACT(DAY FROM pe.fecha_hora_pedido)::int
BETWEEN EXTRACT(DAY FROM CURRENT_DATE)::int
AND EXTRACT(DAY FROM CURRENT_DATE - INTERVAL '1 day');



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

BD-RA4: Modifica la información almacenada en la base de datos utilizando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.

Dado el siguiente diagrama relacional y las características que se indican, responde a las siguientes preguntas.



- El contexto para las siguientes preguntas es sobre una tienda online en la que los clientes compran productos. En la misma compra se puede comprar más de un producto.
- La columna id_compra es un entero que se incrementa automáticamente con cada inserción de una nueva compra.
- El "precio_unitario" de la tabla detalles_compra es el precio que tiene un producto en esa compra en concreto.
- El campo fecha_hora_compra almacena la fecha y la hora en la que se produce la compra.
- El símbolo de la llave indica que la columna forma parte de la clave primaria de la tabla.
- Ninguna de las columnas permiten valores nulos (NN).
- Se parte de una base de datos en la que se acaban de crear las tablas.

31 ¿Qué sentencia permitiría realizar un borrado lógico del cliente con código 1234?

a. DELETE FROM clientes WHERE id_cliente = 1234;



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

b. UPDATE clientes SET activo = FALSE WHERE id_cliente = 1234;

c. DELETE * FROM clientes WHERE id_cliente = 1234;

d. DELETE ALL FROM clientes WHERE id_cliente = 1234 WITHOUT ERASE;

32 ¿Qué sentencia elimina de detalles_compra todas las líneas correspondientes a compras anteriores al 1 de enero de 2022 realizados por clientes cuyo email termina en @correo.edu?

a. DELETE FROM detalles_compra dc

USING compras co, clientes cl

WHERE dc.compra_id = co.id_compra

AND co.cliente_id = cl.id_cliente

AND co.fecha_hora_compra < '2022-01-01'

AND cl.email LIKE '%@correo.edu';

b. DELETE detalles_compra dc

JOIN compras co ON co.id_compra = dc.compra_id

JOIN clientes cl ON cl.id_cliente = co.cliente_id

WHERE co.fecha_hora_compra < '2022-01-01'

AND cl.email = '%@correo.edu';

c. DELETE FROM detalles_compra dc

USING clientes cl

WHERE cl.email LIKE '%@correo.edu'

AND dc.compra_id IS NOT NULL;

d. DROP FROM detalles_compra dc, compras co, clientes cl

WHERE dc.compra_id = co.id_compra

AND co.fecha_hora_compra < '2022-01-01'

AND cl.email LIKE '%@correo.edu';



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

33 ¿Qué sentencia permitiría realizar un borrado físico del cliente con código 1234?

- a. UPDATE clientes SET activo = FALSE WHERE id_cliente = 1234;
- b. DELETE ALL FROM clientes WHERE id_cliente = 1234 WITH CASCADE;
- c. DELETE * FROM clientes WHERE id_cliente = 1234;
- d. DELETE FROM clientes WHERE id_cliente = 1234;

34 Si se desea que en la base de datos aparezca el siguiente pedido, indica qué opción es la correcta: "El cliente Pepe con email pepe@correo.edu y fecha de registro 31 de diciembre del 2024 (que tiene asociado el código 25) realiza una compra de cuatro monitores del mismo tipo (identificados por el id_producto 'MON01') por un precio de 299,99 € (cada uno de ellos) y un portátil (identificado por el id_producto 'P1000') que vale 1700 €".

a. INSERT INTO compras(id_compra, cliente_id, fecha_hora_compra)

VALUES (1, 25, CURRENT_DATETIME);

INSERT INTO clientes(id_cliente, nombre, email, fecha_registro, activo)

VALUES(25, 'Pepe', 'pepe@correo.edu', '2024-12-31', true);

INSERT INTO productos(id_producto, nombre, descripcion)

VALUES('MON01', 'Monitor', 'Puro 4K'), ('P1000', 'Portátil', 'El más potente del mercado');

INSERT INTO detalles_compra(compra_id, producto_id, cantidad, precio_unitario)

VALUES (1, 'MON01', 4, 299.99), (1, 'P1000', 1, 1700);

b. INSERT INTO clientes(id_cliente, nombre, email, fecha_registro, activo)

VALUES(25, 'Pepe', 'pepe@correo.edu', '2024-12-31', true);

INSERT INTO productos(id_producto, nombre, descripcion)

VALUES('MON01', 'Monitor', 'Puro 4K'), ('P1000', 'Portátil', 'El más potente del mercado');

INSERT INTO compras(cliente_id, fecha_hora_compra)

VALUES (25, CURRENT_TIMESTAMP);

INSERT INTO detalles_compra(compra_id, producto_id, cantidad, precio_unitario)

VALUES (1, 'MON01', 4, 299.99), (1, 'P1000', 1, 1700);



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

c. INSERT INTO productos(id_producto, nombre, descripcion)

VALUES('MON01', 'Monitor', 'Puro 4K'), ('P1000', 'Portátil', 'El más potente del mercado');

INSERT INTO clientes(id_cliente, nombre, email, fecha_registro, activo)

VALUES(25, 'Pepe', 'pepe@correo.edu', '31-12-2024', true);

INSERT INTO compras(cliente_id, fecha_hora_compra)

VALUES (25, NOW());

INSERT INTO detalles_compra(compra_id, producto_id, cantidad, precio_unitario)

VALUES (1, 'MON01', 4, 299,99), (1, 'P1000', 1, 1700);

d. Las soluciones de la opción b) y de la opción c) son válidas ambas.

35 ¿Cuál de las siguientes consultas actualiza el campo "descripcion" de los productos cuyo nombre contiene la cadena "Nuevo" para que diga "Producto actualizado"?

a. UPDATE productos

SET descripcion = 'Producto actualizado'

WHERE descripcion LIKE '%Nuevo%';

b. UPDATE productos

SET descripcion = 'Producto actualizado'

WHERE nombre = '%Nuevo';

c. UPDATE productos

SET descripcion = 'Producto actualizado'

WHERE nombre LIKE '%Nuevo%';

d. UPDATE productos

SET descripcion = 'Producto actualizado'

WHERE nombre LIKE 'Nuevo%';



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

36 ¿Cuál de las siguientes consultas inserta los resultados de una consulta en una tabla, que ya existe, llamada nueva_tabla?

- a. CREATE TABLE nueva_tabla INSERT INTO nueva_tabla SELECT * FROM otra_tabla;
- b. CREATE nueva_tabla AS FROM otra_tabla;
- c. INSERT TO nueva_tabla SELECT * FROM otra_tabla;
- d. INSERT INTO nueva_tabla SELECT * FROM otra_tabla;

37 ¿Qué sentencia marca como activos (clientes.activo = TRUE) a todos los clientes que tengan al menos un registro/fila/tupla en compras?

- a. UPDATE clientes cl
SET activo = TRUE
WHERE EXISTS (SELECT 1 FROM compras co WHERE co.cliente_id = cl.id_cliente);
- b. UPDATE clientes cl
FROM compras co
SET activo = TRUE
WHERE co.cliente_id = cl.id_cliente;
- c. UPDATE clientes cl
SET activo = TRUE
FROM (SELECT DISTINCT id_compra FROM compras) co
WHERE cl.id_cliente = co.cliente_id;
- d. UPDATE clientes cl
SET activo = TRUE
FULL OUTER JOIN compras co ON co.cliente_id = cl.id_cliente;



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

- 38 En una base de datos relacional, ¿qué describe mejor una transacción?
- Es un proceso de copia de seguridad que se ejecuta en segundo plano y se termina con SAVEPOINT.
 - Es la ejecución de una única sentencia SELECT, que no puede modificar datos ni deshacerse.
 - Es un conjunto de operaciones que se ejecutan como una unidad atómica.
 - Es un mecanismo de bloqueo que solo impide que otros usuarios lean datos, sin afectar a las escrituras.
- 39 ¿Cuál de las siguientes afirmaciones sobre el uso de ROLLBACK es correcta?
- Confirma (guarda) los cambios realizados desde el último COMMIT.
 - Puede deshacer instrucciones DDL pero no puede deshacer instrucciones DML.
 - Inicia una transacción doble que persistirá los datos de forma automática al finalizar la ejecución de una tarea (siempre y cuando no ocurra una excepción en la tarea).
 - Deshace todos los cambios de la transacción actual y libera los bloqueos adquiridos por esa transacción.
- 40 ¿Cuál de las siguientes afirmaciones es correcta respecto al aislamiento de transacciones en PostgreSQL?
- El nivel de aislamiento READ COMMITTED asegura que una transacción solo vea datos confirmados al inicio de la transacción.
 - El nivel de aislamiento REPEATABLE READ asegura que una transacción vea un conjunto consistente de datos durante toda su duración.
 - El nivel de aislamiento SERIALIZABLE permite que las transacciones se ejecuten en paralelo sin ningún tipo de bloqueo.
 - Ninguna de las otras respuestas es correcta.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

BD-RA5: Desarrolla procedimientos almacenados evaluando y utilizando las sentencias del lenguaje incorporado en el sistema gestor de bases de datos.

41 ¿Qué hace la función COALESCE?

- a. Concatena cadenas sin separador.
- b. Devuelve el primer valor no nulo de una lista de expresiones.
- c. Redondea números a un número de decimales.
- d. Cuenta filas no nulas de una columna.

42 ¿Cuál de las siguientes afirmaciones acerca de funciones y procedimientos es correcta para PostgreSQL?

- a. Un procedimiento siempre devuelve una tabla.
- b. Un procedimiento puede usarse dentro de una expresión SELECT como si fuera una función.
- c. Una función no puede tener parámetros.
- d. Una función debe declarar un tipo de retorno (por ejemplo, RETURNS INTEGER).

43 ¿Cuál sería el tipo de trigger más adecuado para registrar información en una tabla de auditoría cada vez que se produce una actualización en la tabla clientes?

- a. BEFORE SELECT ON clientes
- b. BEFORE UPDATE ON clientes
- c. AFTER UPDATE ON clientes
- d. AFTER COMMIT ON clientes

44 ¿Cuál es el orden correcto de uso de un cursor en entornos procedurales (por ejemplo, PL/pgSQL o PL/SQL)?

- a. DECLARE → OPEN → FETCH → CLOSE
- b. OPEN → DECLARE → FETCH → CLOSE



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

c. OPEN → FETCH → DECLARE → CLOSE

d. DECLARE → FETCH → OPEN → CLOSE

45 En PL/pgSQL (PostgreSQL), ¿cómo se manejan excepciones dentro de un bloque?

a. Mediante un bloque EXCEPTION ... WHEN ... THEN ... dentro de BEGIN ... END

b. Con TRY ... CATCH nativo del lenguaje.

c. Con RESCUE tras un BEGIN

d. Con CATCH(Exception e) como en Java

46 ¿Cuál de las siguientes afirmaciones es cierta respecto a este FOR?

```
FOR r IN  
  SELECT id, total  
  FROM pedidos  
  WHERE estado = 'enviado'  
LOOP  
  CALL registrar_auditoria(r.id, r.total);  
END LOOP;
```

a. Requiere cerrar manualmente el cursor con CLOSE r;

b. No requiere declarar, abrir ni cerrar cursor; PL/pgSQL lo gestiona implícitamente.

c. Ejecuta el cuerpo del bucle en paralelo por cada fila.

d. Solo funciona si la consulta devuelve exactamente una fila.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

47 Suponiendo que se tiene el siguiente código en un procedimiento almacenado y que vip es de tipo booleano, total es numérico, las variables de promoción están definidas, el valor de total es 350, el cliente no es VIP y la fecha de compra está dentro del rango promocional, ¿qué valor toma descuento?

```
IF total IS NULL THEN
    RAISE EXCEPTION 'Total no puede ser NULL';
END IF;
IF total >= 300
    AND fecha_compra BETWEEN fecha_inicio_promocion AND fecha_fin_promocion THEN
    descuento := 0.20;
ELSIF total >= 100 THEN
    IF EXISTS (SELECT 1 FROM clientes WHERE id = v_cliente_id AND vip) THEN
        descuento := 0.15;
    ELSE descuento := 0.10;
    END IF;
ELSE descuento := 0.00;
END IF;
```

- a. 0.00
- b. 0.10
- c. 0.15
- d. 0.20

48 Sabiendo que el email es la clave primaria de la tabla usuarios, ¿qué hace exactamente el siguiente procedimiento almacenado?

```
CREATE OR REPLACE PROCEDURE proc_usuario(p_email text, p_nombre text)
LANGUAGE plpgsql
AS $$
```



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

BEGIN

INSERT INTO usuarios (email, nombre) VALUES (p_email, p_nombre)

ON CONFLICT (email) DO UPDATE

SET nombre = EXCLUDED.nombre,

updated_at = NOW();

END;

\$\$;

a. Inserta la fila si no existe y, si ya existe ese email, actualiza solo nombre, dejando updated_at sin cambios.

b. Siempre inserta la fila con el nuevo usuario y, si ya existe ese email, actualiza nombre y updated_at.

c. Inserta la fila si no existe un usuario con ese email y, si ya existe ese email, actualiza nombre y establece updated_at al momento actual.

d. La sentencia falla si hay conflicto de email. No se puede insertar y actualizar en una sola sentencia.

49 Se desea impedir que el stock quede negativo al insertar/actualizar líneas en detalle_pedido. ¿Cuál es la característica clave del siguiente diseño que lo hace adecuado para bloquear la operación cuando no hay stock suficiente?

CREATE OR REPLACE FUNCTION chk_stock_no_negativo()

RETURNS trigger

LANGUAGE plpgsql

AS \$\$

DECLARE

stock_actual INT;

BEGIN



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

```
SELECT p.stock INTO stock_actual  
  
FROM productos p  
  
WHERE p.id = NEW.producto_id  
  
FOR UPDATE; /* Bloqueo de la fila para que no se lea mientras dure la transacción */  
  
IF stock_actual - NEW.cantidad < 0 THEN  
    RAISE EXCEPTION 'Stock insuficiente para producto %', NEW.producto_id;  
END IF;  
  
RETURN NEW;  
  
END;  
  
$$;
```

```
CREATE TRIGGER tg_chk_stock  
  
BEFORE INSERT OR UPDATE ON detalle_pedido  
  
FOR EACH ROW  
  
EXECUTE FUNCTION chk_stock_no_negativo();
```

- Es un trigger BEFORE por lo que puede validar el stock previamente y abortar con RAISE EXCEPTION.
- Se usa una FUNCTION por lo que se pueden validar todas las filas (todos los stocks) a la vez a diferencia de lo que sucedería con un PROCEDURE.
- Devuelve un trigger (RETURNS trigger) que hace validación automática.
- Usa FOR EACH ROW para prevenir dejar sin stock cualquier producto.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

50 ¿Qué valor devuelve la siguiente función cuando se le pasa por parámetro el valor 4? Se asume que todos los valores de "id" que hay insertados en la tabla test_table(id, valor) son impares y "valor" es de tipo entero.

```
CREATE FUNCTION func_iota(i INTEGER)
RETURNS INTEGER
LANGUAGE plpgsql
AS $$
DECLARE resultado INTEGER;
BEGIN
    IF EXISTS (SELECT 1 FROM test_table WHERE id = i) THEN
        UPDATE test_table SET valor = valor * 2 WHERE id = i;
    ELSE INSERT INTO test_table (id, valor) VALUES (i, i * 10);
    END IF;
    SELECT valor INTO resultado FROM test_table WHERE id = i;
    RETURN resultado;
END;
$$
```

a. 20

b. 40

c. 8

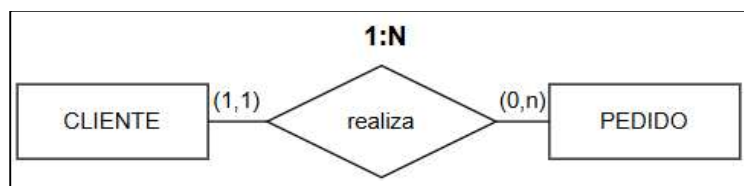
d. Con la información proporcionada no se puede dar una respuesta correcta ya que el código de la función es ambiguo y depende de la información de la tabla test_table.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

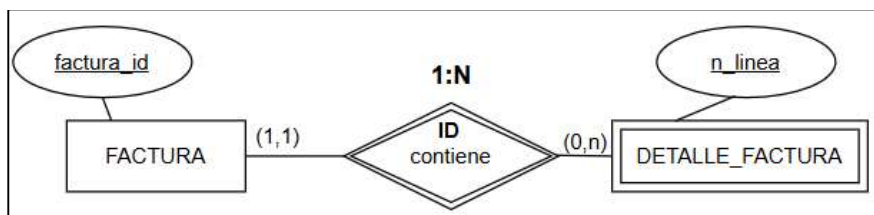
BD-RA6: Diseña modelos relacionales normalizados interpretando diagramas entidad/relación

51 En un diagrama E/R hay una relación 1:N entre Cliente (1) y Pedido (N). La participación de Pedido en la relación es total. ¿Cómo se refleja esto en el modelo relacional?



- Se añade una clave foránea en Cliente, permitiendo NULL.
- Se crea una tabla intermedia Cliente_Pedido con dos claves foráneas.
- Se añade una clave foránea en Pedido con NOT NULL.
- Se duplica la clave primaria de Cliente como parte de la clave primaria de Pedido.

52 Una entidad débil Detalle_Factura depende de Factura y tiene como discriminante n_linea. ¿Cuál es el mapeo correcto al modelo relacional?



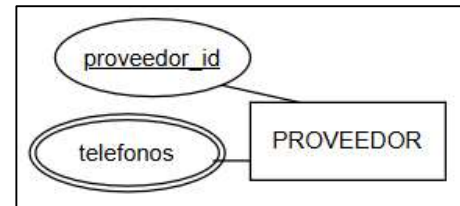
- Tabla Detalle_Factura con clave primaria n_linea y sin claves foráneas.
- Tabla Detalle_Factura con columnas factura_id y n_linea. La clave primaria será solamente factura_id.
- Tabla Detalle_Factura con clave primaria compuesta (factura_id, n_linea) y factura_id como clave foránea a Factura.
- Fusionar Detalle_Factura y Factura en una única tabla con la información completa sobre la factura y su detalle.



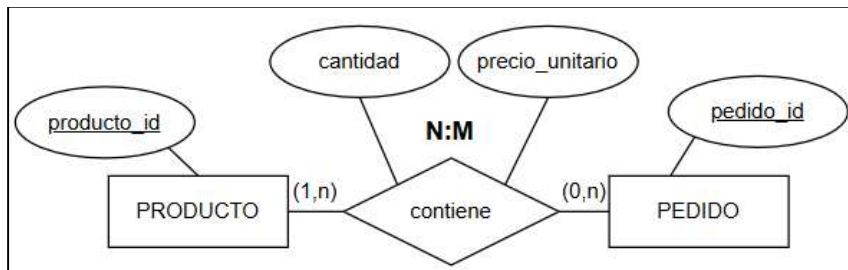
DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

53 El atributo telefono de Proveedor es multivaluado (un proveedor puede tener varios teléfonos). ¿Cuál es la transformación correcta?

- Crear tabla Proveedor_Telefono con clave primaria compuesta (proveedor_id, telefono).
- Añadir varias columnas telefono1, telefono2, telefono3 en Proveedor.
- Guardar los teléfonos como una lista separada por comas en una sola columna.
- Convertir telefono en clave alternativa de Proveedor.



54 Entre Producto y Pedido hay una relación N:M con atributos propios como cantidad y precio_unitario. ¿Cómo se mapea de forma óptima?



- Duplicar productos en la tabla Pedido para no perder información y producto_id como clave primaria única y no nula.
- Añadir pedido_id como clave foránea en Producto.
- Convertir la relación en dos relaciones 1:N sin tabla intermedia, con cantidad y precio_unitario haciendo de clave primaria compuesta.
- Crear tabla asociativa Detalle_Pedido que contenga pedido_id y producto_id, ambas claves foráneas y formando parte de la clave primaria.

55 Dada la tabla Pedido(pedido_id, cliente_id, cliente_nombre, fecha), siendo pedido_id la clave primaria y asumiendo que cliente_nombre depende de cliente_id y fecha depende de pedido_id, ¿qué incumplimiento hay (si es que hay alguno) y cómo resolverlo?

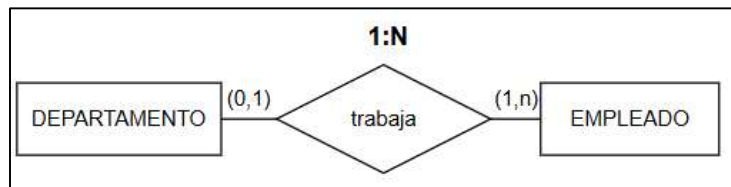
- Incumple 1FN. Hay que agregar también una columna (por ejemplo, dir_envio) para saber a qué dirección se enviará el pedido y que la tabla quede normalizada formalmente.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

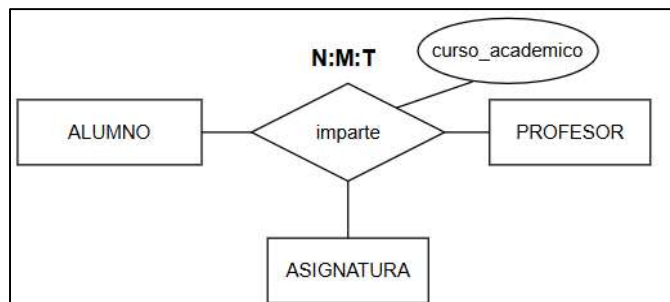
- b. Incumple 2FN por dependencia de clave compuesta. Hay que hacer clave primaria compuesta con cliente_id.
- c. Incumple 3FN por dependencia transitiva. Hay que mover cliente_nombre a una tabla Cliente cuya clave primaria sea cliente_id.
- d. No incumple ninguna forma normal.

56 En un diagrama E/R existe una relación Departamento (1) - Empleado (N) y la participación de Empleado es opcional. ¿Cuál es la decisión adecuada?



- a. Crear tabla intermedia Departamento_Empleado con dos claves foráneas.
- b. Añadir empleado_id en Departamento como clave foránea NOT NULL.
- c. Duplicar la clave primaria de Departamento dentro de la clave primaria de Empleado.
- d. Añadir departamento_id en Empleado como clave foránea que permita nulos.

57 Se modela una relación ternaria entre Alumno, Asignatura y Profesor con el atributo curso_academico. ¿Cuál es el mapeo correcto?



- a. Se crearían las columnas alumno_id y asignatura_id como claves foráneas en Profesor.
- b. Se crearía una tabla asociativa "imparte" con claves foráneas alumno_id, asignatura_id y profesor_id.
- c. Se crearían las columnas profesor_id y alumno_id como claves foráneas en Asignatura y se establecería a UNIQUE asignatura_id en Asignatura.
- d. Tanto la opción a) como la opción b) son correctas pero se recomienda, por optimización en futuras consultas SELECT, la opción b).

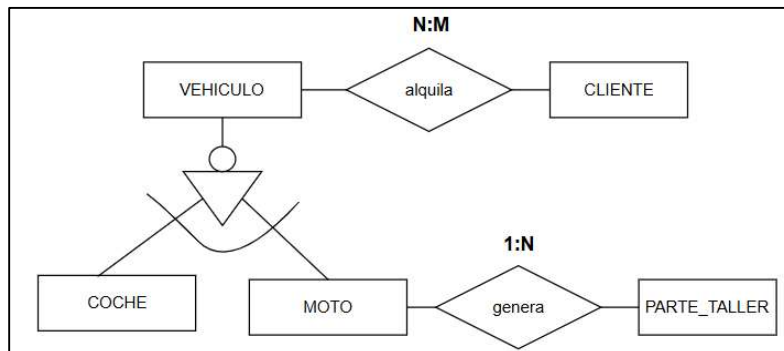


DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./ N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

58 En la tabla Cliente existen las columnas: cliente_id (PK), dni (único y no nulo), email (único y no nulo). ¿Qué afirmación es correcta?

- dni y email no se pueden considerar claves candidatas ya que, para eso, deberían previamente formar parte de la clave primaria en el diseño propuesto.
- dni y email son claves candidatas al haber sido seleccionado cliente_id como clave primaria.
- Únicamente el dni se puede considerar clave candidata ya que el email puede cambiar en el futuro.
- Ninguna columna con UNIQUE (actual o futura) puede ser clave candidata.

59 En una empresa de alquiler de vehículos se tiene Vehículo como supertipo y subtipos Coche y Moto. La especialización es disjunta y total (todo vehículo es coche o moto). Los vehículos son alquilados por los clientes y las motos acumularán diferentes partes de taller durante su vida útil. ¿Qué mapeo respeta mejor la semántica y necesidades del modelo sin crear tablas innecesarias?



- Tres tablas (Vehiculo, Coche y Moto) con una restricción que fuerce pertenecer a exactamente a un subtipo.
- Una única tabla Vehiculo con todas las columnas de Coche y Moto, usando NULL donde no aplique para optimizar el espacio de almacenamiento y la velocidad en la obtención de la información.
- Únicamente Coche y Moto. No se necesita Vehiculo ya que todo vehículo únicamente puede ser Coche o Moto y es lo único que pueden alquilar los clientes.
- Una única tabla Vehículo con una columna "subtipo" para diferenciar de qué subtipo es el vehículo.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

60 La tabla Matricula tiene las columnas (alumno_id, grupo_id, tutor_id, aula). La clave primaria está compuesta por las columnas (alumno_id, grupo_id) y, además, se sabe que a través del grupo_id se puede obtener el tutor_id y el aula: {grupo_id → tutor_id, aula}. ¿Qué ocurre y cuál es la mejor corrección para este diseño?

- No hay problema. La PK compuesta evita redundancias y hace que la tabla esté en 3FN.
- Incumple 2FN por dependencias parciales. Se soluciona descomponiendo en Grupo(grupo_id, tutor_id, aula) y Matricula(alumno_id, grupo_id).
- Incumple 1FN ya que aula debería dividirse en planta y número para que se aplique el principio de atomicidad.
- Incumple 3FN. Se soluciona añadiendo una clave sustituta en Matricula formada por una única columna. Dicha columna deberá ser de tipo UNIQUE, NOT NULL y entero autoincremental (secuencial).

61 La entidad Cliente tiene el atributo compuesto Direccion(calle, número, codigo_postal). No es multivaluado ni se reutiliza entre clientes. ¿Cómo se recomienda transformar?

- Reemplazarlo por columnas simples en Cliente: calle, numero y codigo_postal.
- Crear una tabla Direccion independiente y poner una FK en Cliente obligatoria.
- Guardar todo en una sola columna direccion separada por comas.
- Convertir el atributo Direccion en clave primaria alternativa de Cliente.

62 La entidad Pedido tiene el atributo importe_total derivado de la suma de sus líneas (cantidad * precio_unitario). ¿Qué práctica de diseño es más adecuada desde el punto de vista de la normalización?

- Guardarlo siempre como columna normal y actualizarlo manualmente en cada inserción/actualización.
- No almacenarlo físicamente y calcularlo en consulta (vista) o, si se necesita rendimiento, usar una vista materializada.
- Eliminar las líneas y almacenar solo importe_total para evitar cálculos.
- Guardarlo como texto para aprovechar los índices específicos de los VARCHAR.



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

BD-RA7: Gestiona la información almacenada en bases de datos no relacionales, evaluando y utilizando las posibilidades que proporciona el sistema gestor.

63 Cuál de las siguientes afirmaciones caracteriza mejor a las bases de datos no relacionales (NoSQL) frente a las relacionales?

- a. Se basan en esquemas rígidos de tablas y escalan principalmente de forma vertical.
- b. Están orientadas exclusivamente a cumplir ACID con normalización estricta.
- c. Están diseñadas para escalar horizontalmente admitiendo modelos flexibles (esquema dinámico).
- d. Únicamente sirven como caché en memoria y no almacenan datos persistentes. Si se apaga el servidor, se borran los datos.

64 ¿En qué situación es más razonable optar por NoSQL frente a una base de datos relacional?

- a. Cuando se necesita uniones complejas entre tablas y transacciones multitable muy estrictas.
- b. Cuando todos los datos son del mismo tipo y en forma de tabla.
- c. Cuando se requieren escrituras/lecturas masivas, datos semiestructurados y escalado horizontal global con alta disponibilidad.
- d. Cuando se exige integridad referencial con múltiples claves foráneas y normalización 3FN.

65 ¿Cómo se clasifica MongoDB dentro de los tipos de bases de datos NoSQL?

- a. Grafo.
- b. Clave—valor.
- c. Columna ancha.
- d. Documentos.

66 En MongoDB, ¿qué elemento actúa como identificador único obligatorio en cada documento?

- a. Clave primaria definida por el usuario (PK).



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

b. Índice TTL.

c. OID generado por la aplicación REST.

d. El campo _id.

67 En bases de datos de documentos como MongoDB, ¿qué descripción encaja mejor con el concepto de documento?

- a. Una fila estrictamente tabular con atributos y tipos fijos (siempre los mismos).
- b. Una estructura semiestructurada (por ejemplo, JSON/BSON) con campos anidados y arrays.
- c. Un grafo con nodos y aristas etiquetadas.
- d. Una clave y un valor.

68 En el modelo de documentos, ¿cuándo se recomienda embebido (embed) frente a referencias (reference)?

- a. Cuando las entidades tienen una relación N:M con crecimiento impredecible.
- b. Cuando el acceso a los datos es mayoritariamente separado y por ID.
- c. Cuando los datos se consultan juntos con mucha frecuencia.
- d. Cuando se requiere estricta normalización y múltiples uniones.

69 En MongoDB, ¿qué herramienta permite importar datos desde archivos JSON/CSV a una colección?

- a. mongoimport
- b. pgadmin
- c. mysql-workbench
- d. sql-management-studio



DATOS DEL ALUMNO			FIRMA
APELLIDOS:			
Nombre:	D.N.I./N.I.E.	Fecha:	
Código del ciclo: (1) IFCS02	Denominación completa del título: (1) Técnico Superior en Desarrollo de Aplicaciones Multiplataforma		
Clave o código del módulo: (1) 0484	Denominación completa del módulo profesional: (1) Base de datos		

70 ¿Qué cliente de línea de comandos se utiliza para ejecutar consultas y administrar bases de datos y colecciones en MongoDB?

- a. mongosh
- b. psql
- c. mysqlsh
- d. sql-developer