# PRUEBA FINAL SQL - Full Stack (Principiante → Avanzado → Experto)

## Sección 1: Alternativas (15 preguntas)

1. ¿Cuál de las siguientes instrucciones crea una tabla en MySQL?  
a) MAKE TABLE  
b) CREATE TABLE  
c) NEW TABLE  
d) ADD TABLE

2. ¿Qué tipo de dato es más adecuado para guardar fechas?  
a) VARCHAR  
b) TEXT  
c) DATE  
d) CHAR

3. ¿Cuál instrucción se usa para eliminar datos sin borrar la estructura de la tabla?  
a) DROP TABLE  
b) TRUNCATE TABLE  
c) DELETE FROM  
d) REMOVE TABLE

4. ¿Qué operador se usa para buscar coincidencias parciales en texto?  
a) LIKE  
b) IN  
c) BETWEEN  
d) EXISTS

5. ¿Qué operador se usa para encontrar valores dentro de un rango?  
a) BETWEEN  
b) IN  
c) IS NULL  
d) NOT

6. ¿Qué comando garantiza que una columna no acepte valores duplicados?  
a) UNIQUE  
b) PRIMARY  
c) INDEX  
d) FOREIGN KEY

7. ¿Qué función SQL devuelve el número total de registros de una tabla?  
a) SUM()  
b) COUNT()  
c) AVG()  
d) TOTAL()

8. ¿Qué tipo de JOIN devuelve solo las filas que coinciden en ambas tablas?  
a) LEFT JOIN  
b) RIGHT JOIN  
c) INNER JOIN  
d) FULL JOIN

9. ¿Qué sentencia guarda los cambios realizados en una transacción?  
a) ROLLBACK  
b) COMMIT  
c) SAVE  
d) APPLY

10. ¿Cuál es una restricción que asegura que un valor no sea nulo?  
a) NOT NULL  
b) DEFAULT  
c) UNIQUE  
d) PRIMARY KEY

11. ¿Cuál de los siguientes comandos crea una vista?  
a) CREATE VISTA  
b) CREATE VIEW  
c) NEW VIEW  
d) MAKE VIEW

12. ¿Qué operador se usa para concatenar cadenas de texto?  
a) SUM()  
b) CONCAT()  
c) JOIN()  
d) GLUE()

13. ¿Cuál función encuentra el valor máximo de una columna?  
a) MIN()  
b) MAX()  
c) TOP()  
d) HIGHEST()

14. ¿Qué palabra clave crea un índice?  
a) INDEX  
b) CONSTRAINT  
c) DEFAULT  
d) VIEW

15. ¿Qué hace GROUP BY en una consulta SQL?  
a) Filtra registros  
b) Agrupa resultados por columnas específicas  
c) Elimina duplicados  
d) Combina tablas

## Sección 2: Verdadero o Falso (15 preguntas)

16. DELETE FROM elimina datos y la estructura de la tabla. (V/F) Falso

17. El operador IN sirve para buscar múltiples valores a la vez. (V/F) Verdadero

18. LEFT JOIN devuelve todos los registros de la tabla derecha. (V/F) Falso

19. TRUNCATE TABLE elimina registros uno por uno, por lo que es lento. (V/F) Falso

20. Puedes tener más de una PRIMARY KEY por tabla. (V/F) Falso

21. La palabra DEFAULT define un valor por defecto si no se inserta uno. (V/F) Verdadero

22. El operador != es equivalente a <> en MySQL. (V/F) Verdadero

23. IS NULL compara valores nulos correctamente. (V/F) Verdadero

24. HAVING se usa en consultas que agrupan resultados. (V/F) Verdadero

25. CASE es una estructura de control para hacer condiciones. (V/F) Verdadero

26. Se puede hacer una subconsulta en la cláusula FROM. (V/F) Falso

27. Las VISTAS ocupan espacio físico en la base de datos igual que una tabla. (V/F) Verdadero

28. Los TRIGGERS en MySQL se disparan de manera manual. (V/F) Falso

29. El comando ALTER TABLE permite modificar estructuras existentes. (V/F) Verdadero

30. UNION combina consultas y elimina duplicados automáticamente. (V/F) Verdadero

## Sección 3: Código SQL (15 ejercicios)

31. Crea una tabla Productos con los campos: id\_producto (PK, auto\_increment), nombre, precio.  
create table Productos (

id\_producto int auto\_increment,

nombre varchar (255),

precio int,

primary key (id\_producto)

);

32. Inserta tres productos: Laptop ($900), Mouse ($30), Teclado ($60).  
insert into productos (nombre, precio) values

('Laptop',900),

('Mouse',30),

('Teclado',60);

33. Consulta todos los productos que tengan precio mayor a 50.

select \* from productos where precio >50;

34. Ordena los productos de precio más alto a más bajo.

select \* from productos

order by precio desc;

35. Actualiza el precio del mouse a $35.

update productos

set precio = 35

where nombre = "Mouse";

36. Crea una nueva tabla Clientes con campos: id\_cliente, nombre, correo, pais.

create table Clientes (

id\_cliente int auto\_increment,

nombre varchar(255),

correo varchar(255),

pais varchar(255),

primary key (id\_cliente)

);

37. Inserta cuatro clientes de diferentes países.

insert into clientes (nombre, correo, pais) values

('Esteban','Esteban@gmail.com','Chile'),

('Marcos','Marcos@gmail.com','Bolivia'),

('Juan','Juan@gmail.com','Argentina'),

('Lucas','Lucas@gmail.com','Brasil');

38. Crea una tabla Ventas que registre id\_venta, id\_producto, id\_cliente, fecha.

create table Ventas (

id\_venta int auto\_increment,

id\_producto int,

id\_cliente int,

fecha date,

primary key (id\_venta),

foreign key (id\_producto) references productos (id\_producto),

foreign key (id\_cliente) references clientes (id\_cliente)

);

39. Realiza un INNER JOIN entre Clientes y Ventas.

select \* from clientes c

inner join Ventas v

on c.id\_cliente = v.id\_cliente;

40. Haz un LEFT JOIN para listar todos los clientes y los productos que han comprado (aunque no hayan comprado).

select c.nombre, v.id\_venta from clientes c

left join Ventas v

on c.id\_cliente = v.id\_cliente;

41. Usa CASE para mostrar si el cliente es de 'México' poner 'Cliente Mexicano', sino 'Cliente Internacional'.

select nombre, pais,

case

when pais = "México" then 'Cliente Mexicano'

else 'Cliente internacional'

end as nacionalidad

from clientes;

42. Usa CONCAT para mostrar nombre completo del cliente y su país en una misma columna.

select concat(nombre, ' ', pais) as cliente\_nacionalidad from clientes;

43. Muestra todos los productos cuyo nombre comience con 'T'.

select \* from productos

where nombre like 't%';

44. Usa GROUP BY para agrupar las ventas por país.

select count(id\_venta), pais from ventas v

join clientes c

on v.id\_cliente = c.id\_cliente

group by pais;

45. Haz una subconsulta para encontrar el nombre del cliente con más ventas.

## Sección 4: Preguntas abiertas (15 preguntas)

46. ¿Qué problema soluciona la normalización?

Evita la redundancia y dependencias anomalas

47. ¿Qué ocurre si olvidas poner WHERE en un DELETE?

Borrare todos los datos de la tabla

48. ¿Para qué sirve un índice? ¿Tiene alguna desventaja?

Optimiza la velocidad de las consultas, puede llegar a tener una desventaja si se aplican mal y consume algo de espacio

49. ¿Qué ventajas tiene usar procedimientos almacenados?

Evita tener que escribir una consulta que se utiliza constantemente

50. ¿En qué situaciones usarías un TRIGGER?

Cuando se aplcan consulta instrucciones ddl o dml

51. ¿Qué tipos de relaciones entre tablas existen en bases de datos?

1:n, 1:1 y n:m

52. ¿Qué función tiene ROLLBACK en una transacción?

Retroceder los cambios ejecutados cuando ejecutamos transacciones

53. ¿Cuál es la diferencia entre UNION y UNION ALL?

Union all trae los resultados de una consulta con duplicados y union no

54. ¿Cuándo es mejor usar LEFT JOIN en lugar de INNER JOIN?

Cuando se quiere evitar traer datos nulos para un analisis

55. ¿Qué hace exactamente el operador LIKE '%a%'?

Busca cualquier coincidencia que almenos contenga el caracter “a”

56. ¿Qué pasa si intento insertar un valor duplicado en una columna UNIQUE?

Dara error ya que esta palabra reservada Evita insertar datos duplicados

57. ¿Qué ocurre si intento hacer un JOIN entre tablas sin una condición ON adecuada?

Traera los resultados de la consulta pero sin ninguna utilidad para un analisis adecuado

58. ¿Cuál es la diferencia entre BETWEEN y usar >= y <=?

Between buscara entre un rango de numeros o fechas, mientras que >= y <= buscara valores numericos inferiors/superiores o iguales

59. ¿Qué diferencias existen entre una Vista y una Tabla en SQL?

La vista es una consulta cuyo fin es visualizar los datos, la table, en cambio, se podran manipular los datos dependiendo de la necesidad del usuario

60. ¿Qué pasa si eliminas una tabla que es referenciada por una Foreign Key?

No me dejara eliminarla ya que hace referencia a otra tabla