**PRUEBA TEÓRICA.**

1. ¿Qué es la Normalización?

La normalización de [bases de datos](https://es.wikipedia.org/wiki/Base_de_datos), es un proceso para designar y aplicar diferentes reglas a las relaciones que se obtienen del MER [modelo entidad-relación](https://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n) luego de pasar al [modelo relacional](https://es.wikipedia.org/wiki/Modelo_relacional), con el fin de minimizar o reducir la redundancia de los datos, mantener la integridad de los datos, garantizar la integridad referencial, reducir la problemática de actualización de datos.

2. ¿Qué es PL/SQL?

Structured Query Language

Procedural Language

Procedural Language for Structured Query Language

SQL

3. Unidades Léxicas

Parámetros para construir bloques de constructores

Son bloques de construcción

Bloques

Son bloques de construcción de cualquier bloque PL/SQL

4. Cuál de las siguientes instrucciones es la que está correctamente escrita:

Create table “Select” (Columna1 number(8));

Create table 1empleados (Columna1 number(8));

Create table update (Columna1 number(8));

Create table empleados (Columna1 number(8));

5. Directriz Lateral de PL/SQL

Fechas correspondientes a los datos almacenados en consulta

Valores alfanuméricos de datos almacenados

Los números pueden ser valores simples o científica

Fechas y valores únicos

6. Los identificadores se utilizan para:

Proporciona una estructura de bloques para las unidades de ejecutables

Proporciona varias estructuras ejecutables

Proporcionar una convención para nombres de variables

Manipulación de los datos almacenados en consulta

7. ¿Con qué variable puedo hacer uso de declaración?

DECLARE\_DBMS

DECLARE

DECLARE\_FUNCTION

DECALRE

8. ¿Con qué procedimiento puedo realizar una Salida de un bloque PL/SQL?

DBMS\_OUTPUT.PUT-LINE

DMS\_OUPUT.PUT\_LINE

DBM\_OUPUT.PUT\_LINE

DBMS\_OUTPUT.PUT\_LINE

9. ¿Cuál de las siguientes afirmaciones son comandos DML?

INSERT

UPDATE

GRANT

TRUNCATE

CREATE

**PRUEBA TEÓRICA Y TÉCNICA DE ORACLE**

10. ¿Cuál es la diferencia entre los comandos TRUNCATE, DELETE y DROP?

* El comando **TRUNCATE** permite dejar vacía una tabla, borrar todos los datos de una tabla pero mantiene la estructura
* El comando **DELETE** permite borrar registros de una tabla cuando tiene la clausula **WHERE** se condiciona al cumplimiento de la condición para borrar el registro, cuando no hay clausula **WHERE** borra todos los registros de la tabla.
* El comando **DROP** borra la tabla y su estructura.



De acuerdo al MER de proyecto de Jardinería, realice los siguientes querys.

1. Visualizar la ciudad y el código postal Colombia

SELECT ciudad, codigopostal

FROM OFICINAS

WHERE LOWER(pais) = 'colombia';

1. Visualizar el apellidos y nombre del jefe de la empresa

SELECT apellido1, apellido2, nombre

FROM empleados

WHERE codigojefe IS NULL;

1. Visualizar el nombre y cargo de los empleados que no sean directores de oficina.

SELECT nombre, puesto

FROM empleados

WHERE LOWER(puesto) <> 'director de oficina';

1. Visualizar la gama de productos(código) más vendidos. Implementar una vista

CREATE OR REPLACE VIEW gamas\_vendidas AS

SELECT p.gama, SUM(dp.cantidad) AS cantidad

FROM detallepedidos dp, productos p

WHERE p.codigoproducto = dp.codigoproducto

GROUP BY p.gama;

SELECT gv.gama, gv.cantidad

FROM gamas\_vendidas gv

WHERE gv.CANTIDAD = (select MAX(gv.cantidad)

FROM gamas\_vendidas gv);

1. Visualizar el país(cliente) donde menos pedidos se hacen. Implementar una Vista

CREATE OR REPLACE VIEW pedidos\_paises AS

SELECT c.pais, count(\*) AS num\_pedidos

FROM clientes c, pedidos p

WHERE c.codigocliente = p.codigocliente

GROUP BY c.pais;

SELECT pp.pais, pp.num\_pedidos

FROM PEDIDOS\_PAISES pp

WHERE pp.num\_pedidos = (SELECT MIN(num\_pedidos)

FROM PEDIDOS\_PAISES);

1. Visualizar los pedidos(código) donde se hayan vendido más de 10 productos

SELECT pe.codigopedido

FROM pedidos pe, detallepedidos dp

WHERE pe.codigopedido = dp.codigopedido

GROUP BY pe.codigopedido

HAVING COUNT(\*)>10;

1. Visualizar los pedidos(código) donde el precio del pedido sea superior a la media de todos los pedidos.

SELECT pe.codigopedido

FROM pedidos pe

WHERE

(SELECT SUM(dp.cantidad \* dp.PRECIOUNIDAD) AS total

FROM pedidos p, detallepedidos dp

WHERE p.codigopedido = dp.codigopedido and pe.codigopedido = p.codigopedido

GROUP BY p.codigopedido)

>

(SELECT AVG(totalp.total)

FROM (SELECT p.codigopedido, SUM(dp.cantidad \* dp.PRECIOUNIDAD) AS total

FROM pedidos p, detallepedidos dp

WHERE p.codigopedido = dp.codigopedido

GROUP BY p.codigopedido) totalp);

1. Visualizar la cantidad de veces y el código; que se ha pedido un producto al menos una vez.

SELECT SUM(dp.cantidad) AS "cantidad", p.codigoproducto

FROM productos p, detallepedidos dp

WHERE p.codigoproducto = dp.codigoproducto

GROUP BY p.codigoproducto;

1. Visualizar el producto(código) más vendido de la gama “Aromáticas”.

SELECT DISTINCT dp.codigoproducto

FROM pedidos pe, detallepedidos dp

WHERE pe.codigopedido = dp.codigopedido

AND dp.codigoproducto IN (SELECT codigoproducto

FROM productos

WHERE precioventa = (SELECT max(precioventa)

FROM productos p, gamasproductos g

WHERE p.gama = g.gama

AND LOWER(g.gama) = 'Aromáticas'))

1. Visualizar los pedidos(código) donde el precio del pedido sea superior a la media de todos los pedidos

SELECT pe.codigopedido

FROM pedidos pe

WHERE

(SELECT SUM(dp.cantidad \* dp.PRECIOUNIDAD) AS total

FROM pedidos p, detallepedidos dp

WHERE p.codigopedido = dp.codigopedido and pe.codigopedido = p.codigopedido

GROUP BY p.codigopedido)

>

(SELECT AVG(totalp.total)

FROM (SELECT p.codigopedido, SUM(dp.cantidad \* dp.PRECIOUNIDAD) AS total

FROM pedidos p, detallepedidos dp

WHERE p.codigopedido = dp.codigopedido

GROUP BY p.codigopedido) totalp);