EJERCITACION:

1. **TABLAS**
2. Ejecutar la siguiente instrucción:

*create table USUARIOS(*

*nombre varchar2(10),*

*domicilio varchar2(10)*

*);*

-Agregarle el campo “clave”, varchar2(10)

-Modificar el campo “nombre”, varchar2(30)

-Eliminar el campo “domicilio”

*-- Verificar con: select \*from usuarios;*

B) Cree una tabla llamada "agenda", debe tener los siguientes campos: apellido, varchar2(30); nombre, varchar2(20); domicilio, varchar2 (30) y telefono, varchar2(11)

*-- Verificar con: select \*from agenda;*

1. **INSERT**
2. Ingrese los siguientes registros:

*insert into agenda (apellido, nombre, domicilio, telefono)*

*values ('Moreno','Alberto','Colon 123','4234567');*

*insert into agenda (apellido,nombre, domicilio, telefono)*

*values ('Torres','Juan','Avellaneda 135','4458787');*

*-- Verificar con: select \*from agenda;*

1. Ingrese un usuario con nombre “Lorena”.

*-- Verificar con: select \*from usuarios;*

1. **SELECT**

Realice un "select" mostrando solamente el nombre y el telefono de todas las agendas.

1. **WHERE**

Ingrese los siguientes registros:

*insert into agenda(apellido,nombre,domicilio,telefono) values*

*('Acosta', 'Ana', 'Colon 123', '4234567');*

*insert into agenda(apellido,nombre,domicilio,telefono) values*

*('Bustamante', 'Betina', 'Avellaneda 135', '4458787');*

*insert into agenda(apellido,nombre,domicilio,telefono) values*

*('Lopez', 'Hector', 'Salta 545', '4887788');*

*insert into agenda(apellido,nombre,domicilio,telefono) values*

*('Lopez', 'Luis', 'Urquiza 333', '4545454');*

*insert into agenda(apellido,nombre,domicilio,telefono) values*

*('Lopez', 'Marisa', 'Urquiza 333', '4545454');*

1. Seleccione el registro cuyo nombre sea "Marisa" (1 registro)
2. Seleccione los nombres y domicilios de quienes tengan apellido igual a "Lopez" (3 registros)
3. Seleccione los nombres y domicilios de quienes tengan apellido igual a "lopez" (en minúsculas)

**5 – OPERADORES**

Ejecute las siguientes instrucciones:

*create table articulos(*

*codigo number(5),*

*nombre varchar2(20),*

*descripcion varchar2(30),*

*precio number(6,2),*

*cantidad number(3)*

*);*

*insert into articulos (codigo, nombre, descripcion, precio,cantidad)*

*values (1,'impresora','Epson Stylus C45',400.80,20);*

*insert into articulos (codigo, nombre, descripcion, precio,cantidad)*

*values (2,'impresora','Epson Stylus C85',500,30);*

*insert into articulos (codigo, nombre, descripcion, precio,cantidad)*

*values (3,'monitor','Samsung 14',800,10);*

*insert into articulos (codigo, nombre, descripcion, precio,cantidad)*

*values (4,'teclado','ingles Biswal',100,50);*

*insert into articulos (codigo, nombre, descripcion, precio,cantidad)*

*values (5,'teclado','español Biswal',90,50);*

Operadores Relacionales:

1. Seleccione los datos de las impresoras (2 registros)
2. Seleccione los artículos cuyo precio sea mayor o igual a 400 (3 registros)
3. Selecciones el nombre y descripción de los artículos que NO cuesten $100 (4 registros)

Operadores Aritméticos:

1. El comercio quiere aumentar los precios de todos sus artículos en un 15%. Actualice todos los precios empleando operadores aritméticos.

Operadores de Concatenación:

1. Muestre todos los artículos, concatenando el nombre y la descripción de cada uno de ellos separados por coma.

Operadores Lógicos:

1. Muestre los artículos que sean teclado o impresoras.
2. Muestre los artículos cuyo precio sea mayor a 450 y tengan más de 25 unidades en stock.
3. **UPDATE**

A) Modifique el registro de la agenda cuyo nombre sea "Juan" por "Juan Jose" (1 registro actualizado)

B) Actualice los registros cuyo número telefónico sea igual a "4545454" por "4445566" (2 registros)

**7 – NULL y IS NULL**

1. Seleccione utilizando “IS NULL”, los usuarios cuya clave sea nula.

**8 - PRIMARY KEY**

1. Elimine la Tabla USUARIOS
2. Cree nuevamente la tabla usuarios utilizando como primary key el nombre.
3. Ingrese los siguientes registros a la tabla USUARIOS:

|  |  |
| --- | --- |
| Nombre | Clave |
| juanperez | Boca |
| raulgarcia | River |
| juanperez | Independiente |

**9 – CHECK**

1. Agregar un constraint a la tabla artículos que verifique que el precio sea distinto a 0.
2. Ejecutar la siguiente instrucción:

*insert into articulos (codigo, nombre, descripcion, precio,cantidad)*

*values (6,'microfono','Logitech', 0, 1);*

**10 – FOREIGN KEY**

|  |  |
| --- | --- |
| USUARIOS | |
| nombre (PK) | Varchar2(30) |
| clave | Varchar2(10) |

1. Crear una tabla CONEXIONES que se conecte a la tabla USUARIOS mediente una foreign key como muestran el siguiente gráfico:

|  |  |
| --- | --- |
| CONEXIONES | |
| id\_Conexion (PK) | Varchar2(10) |
| usuario (FK) | Varchar2(30) |
| fecha | Date |

1. Ejecute las siguientes instrucciones:

*insert into conexiones (id\_conexion, usuario, fecha)*

*values ('0', 'Lorena', '01/01/2020' );*

*insert into conexiones (id\_conexion, usuario, fecha)*

*values ('1', 'Marcelo', '01/01/2020' );*

**11- ORDER BY**

1. Mostrar el nombre y el precio de los artículos ordenados por el precio de menor a mayor.

**12- IN**

1. Utilizando IN, traer los artículos cuyo nombre sea “impresora” y “teclado”.

**13- COUNT**

1. Utilizando count, mostrar la cantidad de registros que hay en la tabla ARTICULS.
2. Mostrar la cantidad de registros de la tabla AGENDA cuyo apellido sea ‘Lopez’.

**14- GROUP BY**

1. Mostrar la cantidad de artículos agrupados por nombre.

**15- HAVING**

1. Mostrar la cantidad de artículos agrupados por nombre cuya cantidad sea mayor a 1.

**16- DISTINCT**

1. Ejecutar las siguientes instrucciones:

*insert into agenda(apellido,nombre,domicilio,telefono) values*

*('Lopez', 'Hector', 'Salta 545', '4887788');*

*select count(\*) from agenda;*

*select distinct count(\*) from agenda;*

**17- JOIN**

1. Ejecutar las siguientes instrucciones:

insert into usuarios values('Juan Perez','22222222');

insert into usuarios values('Maria Lopez','23333333');

insert into usuarios values('Agustin Juarez','24444444');

insert into usuarios values('Marta Garcia','25555555');

insert into inscriptos values('Juan Perez','22222222');

insert into inscriptos values('Maria Lopez','23333333');

insert into conexiones values('1', 'Juan Perez', '01/12/2006');

insert into conexiones values('2', 'Juan Perez', '08/12/2006');

insert into conexiones values('3', 'Maria Lopez', '01/12/2006');

insert into conexiones values('4', 'Agustin Juarez', '08/12/2006');

insert into conexiones values('5', 'Juan Perez', '02/12/2006');

insert into conexiones values('6', 'Maria Lopez', '02/12/2006');

1. Muestre los nombres, las claves y la fecha de conexión de cada usuario ordenados por fecha.
2. Muestre lo mismo utilizando un "right join".
3. Obtenga la misma salida que la consulta anterior pero empleando un "left join".

**18 – UNION, INTERSECCTION y MINUS**

1. Ejecutar las siguientes instrucciones:

insert into medicos values(1,'20111222','Ana Acosta','Avellaneda 111','clinica');

insert into medicos values(2,'21222333','Betina Bustos','Bulnes 222','clinica');

insert into medicos values(3,'22333444','Carlos Caseros','Colon 333','pediatria');

insert into medicos values(4,'23444555','Daniel Duarte','Duarte Quiros 444','oculista');

insert into medicos values(5,'24555666','Estela Esper','Esmeralda 555','alergia');

insert into pacientes values('24555666','Estela Esper','Esmeralda 555','IPAM');

insert into pacientes values('23444555','Daniel Duarte','Duarte Quiros 444','OSDOP');

insert into pacientes values('30111222','Fabiana Fuentes','Famatina 666','PAMI');

insert into pacientes values('31222333','Gaston Gonzalez','Guemes 777','PAMI');

1. La clínica necesita el nombre y domicilio de médicos y pacientes para enviarles una tarjeta de invitación a la inauguración de un nuevo establecimiento. Emplee el operador "union" para obtener dicha información de ambas tablas. Note que existen dos médicos que también están presentes en la tabla "pacientes"; tales registros aparecen una sola vez en el resultado de "union".
2. La clínica necesita el nombre y domicilio de los pacientes que también son médicos para enviarles una tarjeta de descuento para ciertas prácticas. Emplee el operador "intersect" para obtener dicha información de ambas tablas.
3. La clínica necesita los nombres de los pacientes que no son médicos. Realice una operación de resta.

**19 – SUBCONSULTAS**

1. Mostrar el nombre y el precio del articulo más costoso utilizando una subconsulta.
2. Obtener el nombre de los artículos cuya descripción sea "*Epson Stylus*" utilizando una subconsulta.

**20 – VISTAS**

1. Crear una vista “vista\_usuarios\_conexiones” que muestre los nombres, claves y fechas de conexión de los usuarios.
2. Editar la vista para que muestre el id de conexión.
3. Realizar un update a la vista del registro con id de conexión ‘5’, modificando la clave a nulo.