

Práctica 4- sql

viernes, 27 de octubre de 2023

13:50

Ejercicio 1

Cliente(idCliente, nombre, apellido, DNI, telefono, direccion)

Factura (nroTicket, total, fecha, hora, idCliente (fk))

Detalle(nroTicket, idProducto, cantidad, preciounitario)

Producto(idProducto, descripcion, precio, nombreP, stock)

1. Listar datos personales de clientes cuyo apellido comience con el string 'Pe'. Ordenar por DNI
2. Listar nombre, apellido, DNI, teléfono y dirección de clientes que realizaron compras solamente durante 2017.
3. Listar nombre, descripción, precio y stock de productos vendidos al cliente con DNI:45789456, pero que no fueron vendidos a clientes de apellido 'Garcia'.
4. Listar nombre, descripción, precio y stock de productos no vendidos a clientes que tengan teléfono con característica: 221 (La característica está al comienzo del teléfono). Ordenar por nombre.
5. Listar para cada producto: nombre, descripción, precio y cuantas veces fué vendido. Tenga en cuenta que puede no haberse vendido nunca el producto.
6. Listar nombre, apellido, DNI, teléfono y dirección de clientes que compraron los productos con nombre 'prod1' y 'prod2' pero nunca compraron el producto con nombre 'prod3'.
7. Listar nroTicket, total, fecha, hora y DNI del cliente, de aquellas facturas donde se haya comprado el producto 'prod38' o la factura tenga fecha de 2019.
8. Agregar un cliente con los siguientes datos: nombre:'Jorge Luis', apellido:'Castor', DNI:40578999, teléfono:221-4400789, dirección:'11 entre 500 y 501 nro:2587' y el id de cliente: 500002. Se supone que el idCliente 500002 no existe.
9. Listar nroTicket, total, fecha, hora para las facturas del cliente 'Jorge Pérez' donde no haya comprado el producto 'Z'.
10. Listar DNI, apellido y nombre de clientes donde el monto total comprado, teniendo en cuenta todas sus facturas, supere \$10.000.000.

```
1- SELECT *
FROM cliente
WHERE(apellido LIKE 'Pe%')
ORDER BY DNI

2- SELECT c.nombre, c.apellido, c.DNI, c.telefono, c.direccion
FROM cliente c INNER JOIN factura f ON (c.idCliente=f.idCliente)
WHERE(fecha BETWEEN "1/1/2017" AND "31/12/2017")
EXCEPT
SELECT c.nombre, c.apellido, c.DNI, c.telefono, c.direccion
FROM cliente c INNER JOIN factura f ON (c.idCliente=f.idCliente)
WHERE(fecha NOT BETWEEN "1/1/2017" AND "31/12/2017")

3- SELECT p.nombre, p.descripcion, p.precio, p.stock
FROM cliente c NATURAL JOIN factura f NATURAL JOIN detalle d NATURAL JOIN producto p
WHERE(c.DNI='45789456')
EXCEPT
SELECT p.nombre, p.descripcion, p.precio, p.stock
FROM cliente c NATURAL JOIN factura f NATURAL JOIN detalle d NATURAL JOIN producto p
WHERE(apellido='Garcia')

4- SELECT p.nombre, p.descripcion, p.precio, p.stock
FROM cliente c NATURAL JOIN factura f NATURAL JOIN detalle d NATURAL JOIN producto p
EXCEPT
SELECT p.nombre, p.descripcion, p.precio, p.stock
FROM cliente c NATURAL JOIN factura f NATURAL JOIN detalle d NATURAL JOIN producto p
WHERE(telefono LIKE '221%')
ORDER BY nombre

5- SELECT p.nombre, p.descripcion, p.precio SUM(d.cantidad)
FROM producto p LEFT JOIN detalle d ON (d.idProducto=p.idProducto)

6- SELECT c.nombre, c.apellido, c.DNI, c.telefono, c.direccion
FROM cliente c NATURAL JOIN factura f NATURAL JOIN detalle d NATURAL JOIN producto p
WHERE(p.nombre="producto1")
INTERSECT
SELECT c.nombre, c.apellido, c.DNI, c.telefono, c.direccion
FROM cliente c NATURAL JOIN factura f NATURAL JOIN detalle d NATURAL JOIN producto p
WHERE(p.nombre="Producto2")
EXCEPT
SELECT c.nombre, c.apellido, c.DNI, c.telefono, c.direccion
FROM cliente c NATURAL JOIN factura f NATURAL JOIN detalle d NATURAL JOIN producto p
WHERE(p.nombre="producto3")

6- SELECT c.nombre, c.apellido, c.DNI, c.telefono, c.direccion
FROM cliente c NATURAL JOIN factura f NATURAL JOIN detalle d NATURAL JOIN producto p
WHERE(p.nombre="producto1" AND c.dni IN
(
SELECT c2.dni
FROM cliente c2 NATURAL JOIN factura NATURAL JOIN detalle NATURAL JOIN producto p2
WHERE p2.nombre= "producto2"
)
AND c.dni NOT IN
(
SELECT c3.dni
FROM cliente c3 NATURAL JOIN factura NATURAL JOIN detalle NATURAL JOIN producto p3
WHERE p3.nombre= "producto3"
)
)

7- SELECT f.nroTicket, f.total, f.fecha, f.hora, c.dni
FROM cliente c INNER JOIN factura f ON (c.idcliente=f.idcliente)
INNER JOIN producto p ON (f.idproducto=p.idproducto)
WHERE (p.nombre= "prod38" OR f.fecha BETWEEN "1/1/2019" AND f.fecha "31/1/2019")

8- INSERT INTO cliente (nombre, apellido, dni, telefono, direccion, id) VALUES ('Jorge Luis', 'Castor',
4057, 22144, '11 entre bnla', 500002)

9- SELECT f.nroTicket, f.total, f.fecha, f.hora
FROM cliente c NATURAL JOIN factura f NATURAL JOIN detalle d NATURAL JOIN producto p
WHERE(c.nombre='Jorge' AND c.apellido=' Perez' AND f.nroTicket NOT IN)(
SELECT f.nroTicket, f.total, f.fecha, f.hora
FROM cliente c NATURAL JOIN factura f NATURAL JOIN detalle d NATURAL JOIN producto p
WHERE(p.nombre=z)
)

10- SELECT c.nombre, c.dni, c.apellido SUM (f.total)
FROM cliente c INNER JOIN factura f ON (c.idcliente=f.idcliente)
GROUP BY c.dni
HAVING SUM (f.total) >=10000000
```

3:

Club=(codigoClub, nombre, anioFundacion, codigoCiudad(FK))

Ciudad=(codigoCiudad, nombre)

Estadio=(codigoEstadio, codigoClub(FK), nombre, direccion)

Jugador=(DNI, nombre, apellido, edad, codigoCiudad(FK))

ClubJugador=(codigoClub, DNI, desde, hasta)

- Reportar nombre y anioFundacion de aquellos clubes de la ciudad de La Plata que no poseen estadio.
- Listar nombre de los clubes que no hayan tenido ni tengan jugadores de la ciudad de Berisso.
- Mostrar DNI, nombre y apellido de aquellos jugadores que jugaron o juegan en el club Gimnasia y Esgrima La Plata.
- Mostrar DNI, nombre y apellido de aquellos jugadores que tengan más de 29 años y hayan jugado o juegan en algún club de la ciudad de Córdoba.
- Mostrar para cada club, nombre de club y la edad promedio de los jugadores que juegan actualmente en cada uno.
- Listar para cada jugador: nombre, apellido, edad y cantidad de clubes diferentes en los que jugó. (incluido el actual)
- Mostrar el nombre de los clubes que nunca hayan tenido jugadores de la ciudad de Mar del Plata.
- Reportar el nombre y apellido de aquellos jugadores que hayan jugado en todos los clubes.
- Agregar con codigoClub 1234 el club "Estrella de Berisso" que se fundó en 1921 y que pertenece a la ciudad de Berisso. Puede asumir que el codigoClub 1234 no existe en la tabla Club.

```

1- SELECT c.nombre, c.aniofundacion
FROM club c INNER JOIN ciudad ci ON (c.codigoCiudad=ci.codigoCiudad)
WHERE ci.nombre=laplata AND NOT EXISTS IN(
SELECT c.nombre, c.aniofundacion
FROM club c INNER JOIN estadio e ON (c.codigoClub= e.codigoClub)

1- SELECT c.nombre, c.aniofundacion
FROM club c INNER JOIN ciudad ci ON (c.codigoCiudad=ci.codigoCiudad)
WHERE (ci.nombre=laplata)
EXCEPT
SELECT c.nombre, c.aniofundacion
FROM club c INNER JOIN estadio e ON (e.codigoClub= e.codigoClub)

2- SELECT c.nombre
FROM club
EXCEPT
(SELECT c.nombre
FROM club c INNER JOIN clubJugador cb ON (c.cod=cb.cod)
INNER JOIN jugador j ON (cb.dni= j.dni)
INNER JOIN ciudad ci ON (j.codCiudad=ci.codCiudad)
WHERE ci.nombre=berisso)

3- Select j.dni, j.nombre, j.apellido
FROM jugador j INNER JOIN club jugador cj ON (j.dni=cj.dni)
Inner JOIN club c ON (cj.codigoClub=c.codigoClub)
Where c.nombre=esgrima

4- Select j.dni, j.nombre, j.apellido
FROM jugador j INNER JOIN club jugador cj ON (j.dni=cj.dni)
Inner JOIN club c ON (cj.codigoClub=c.codigoClub)
Inner join ciudad ci ON (c.codigoCiudad= ci.codigoCiudad)
Where ci.nombre=cordoba AND j.edad>= 29

5- Select c.nombre AVG (j.edad) as promedio
FROM jugador j INNER JOIN club jugador cj ON (j.dni=cj.dni)
Inner JOIN club c ON (cj.codigoClub=c.codigoClub)
WHERE cj.hasta IS NULL
GROUP BY c.idclub, c.nombre

6- Select j.nombre, j.apellido, j.edad COUNT (*) as cantidad
FROM jugador j INNER JOIN club jugador cj ON (j.dni=cj.dni)
GROUP BY j.dni, j.nombre, j.apellido, j.edad

7- SELECT c.nombre
FROM club
EXCEPT
(SELECT c.nombre
FROM club c INNER JOIN clubJugador cb ON (c.cod=cb.cod)
INNER JOIN jugador j ON (cb.dni= j.dni)
INNER JOIN ciudad ci ON (j.codCiudad=ci.codCiudad)
WHERE ci.nombre=mar del plata)

8- SELECT j.nombre, j.apellido
FROM jugador j
WHERE NOT EXISTS IN (
SELECT *
FROM club c
WHERE NOT EXISTS IN(
SELECT *
FROM clubJugador cj
WHERE cj.idclub=c.idclub AND cj.DNI=j.dni
)
)

9-INSERT INTO club (nombre,año, ciudad) VALUES(estrella, 1921, berisso)

```

Ejercicio 4:

PERSONA = (DNI, Apellido, Nombre, Fecha_Nacimiento, Estado_Civil, Genero)

ALUMNO = (DNI, Legajo, Año_Ingreso)

PROFESOR = (DNI, Matricula, Nro_Expediente)

TITULO = (Cod_Titulo, Nombre, Descripción)

TITULO-PROFESOR = (Cod_Titulo, DNI, Fecha)

CURSO = (Cod_Curso, Nombre, Descripción, Fecha_Creacion, Duracion)

ALUMNO-CURSO = (DNI, Cod_Curso, Año, Desempeño, Calificación)

PROFESOR-CURSO = (DNI, Cod_Curso, Fecha_Desde, Fecha_Hasta)

- Listar DNI, legajo y apellido y nombre de todos los alumnos que tegan año ingreso inferior a 2014.
- Listar DNI, matricula, apellido y nombre de los profesores que dictan cursos que tengan más 100 horas de duración. Ordenar por DNI
- Listar el DNI, Apellido, Nombre, Género y Fecha de nacimiento de los alumnos inscriptos al curso con nombre "Diseño de Bases de Datos" en 2019.
- Listar el DNI, Apellido, Nombre y Calificación de aquellos alumnos que obtuvieron una calificación superior a 9 en los cursos que dicta el profesor "Juan Garcia". Dicho listado deberá estar ordenado por Apellido.
- Listar el DNI, Apellido, Nombre y Matricula de aquellos profesores que posean más de 3 títulos. Dicho listado deberá estar ordenado por Apellido y Nombre.
- Listar el DNI, Apellido, Nombre, Cantidad de horas y Promedio de horas que dicta cada profesor. La cantidad de horas se calcula como la suma de la duración de todos los cursos que dicta.
- Listar Nombre, Descripción del curso que posea más alumnos inscriptos y del que posea menos alumnos inscriptos durante 2019.
- Listar el DNI, Apellido, Nombre, Legajo de alumnos que realizaron cursos con nombre conteniendo el string 'BD' durante 2018 pero no realizaron ningún curso durante 2019.
- Agregar un profesor con los datos que prefiera y agregarle el título con código: 25.
- Modificar el estado civil del alumno cuyo legajo es '2020/09', el nuevo estado civil es divorciado.
- Dar de baja el alumno con DNI 30568989. Realizar todas las bajas necesarias para no dejar el conjunto de relaciones en estado inconsistente.

```

1- SELECT a.dni, a.legajo, p.apellido, p.nombre
FROM persona p INNER JOIN alumno a ON (a.dni=p.dni)
WHERE a.añoingreso<1/1/2014

2- SELECT p.dni, p.matricula, pe.apellido, pe.nombre
FROM persona pe INNER JOIN profesor p ON (p.dni=pe.dni)
INNER JOIN profesorcurso pc ON (p.dni=pc.dni)
INNER JOIN curso c ON (pc.codcurso=c.codcurso)
WHERE (c.duracion > 100)
ORDER BY p.dni

3- SELECT a.dni, p.apellido, p.nombre, p.genero, p.nacimiento
FROM persona p INNER JOIN alumno a ON (a.dni=p.dni)
INNER JOIN alumnocurso ac ON (ac.dni=p.dni)
INNER JOIN curso c ON (c.codcurso=ac.codcurso)
WHERE c.nombre='dbd' AND ac.año=2019

4- SELECT a.dni, p.apellido, p.nombre, ac.calificación
FROM persona p INNER JOIN alumno a ON (a.dni=p.dni)
INNER JOIN alumnocurso ac ON (ac.dni=p.dni)
WHERE ac.NOTA>9 AND EXISTS(SELECT *
FROM persona pe INNER JOIN profesor pr ON (pe.dni=pr.dni)
INNER JOIN profesor curso pc ON (pr.dni= pc.dni)
WHERE (ac.curso=pc.curso AND PE.nombre='JUAN'))

5- SELECT p.dni, p.apellido, p.nombre
FROM persona p INNER JOIN profesor pr ON (p.dni=pr.dni)
INNER JOIN titulo profesor tp ON (p.dni=tp.dni)
INNER JOIN titulo t on (tp.codtitulo=codtitulo)
GROUP BY p.dni, p.apellido, p.nombre
HAVING COUNT (p.dni)>3

6- SELECT p.dni, p.apellido, p.nombre, SUM(c.duracion), AVG (c.duracion)
FROM persona p INNER JOIN profesor pr ON (p.dni=pr.dni)
INNER JOIN profesorcurso pc ON (p.dni=pc.dni)

```

```

INNER JOIN curso c ON (p.codcurso=c.codcurso)
GROUP BY p.dni, p.apellido, p.nombre

7- SELECT c.nombre, c.descripcion
FROM alumnocurso ac INNER JOIN curso c ON (ac.cod=c.cod)
WHERE (ac.año=2019)
GROUP BY c.cod, c.nombre, c.descripcion
HAVING count (*) >= ALL (
    SELECT COUNT (*)
    FROM alumnocurso ac INNER JOIN curso c ON (ac.cod=c.cod)
    WHERE (ac.año=2019)
    GROUP BY c.cod, c.nombre, c.descripcion
)
UNION
SELECT c.nombre, c.descripcion
FROM alumnocurso ac INNER JOIN curso c ON (ac.cod=c.cod)
WHERE (ac.año=2019)
GROUP BY c.cod, c.nombre, c.descripcion
HAVING count (*) <= ALL (
    SELECT COUNT (*)
    FROM alumnocurso ac INNER JOIN curso c ON (ac.cod=c.cod)
    WHERE (ac.año=2019)
    GROUP BY c.cod, c.nombre, c.descripcion
)

8- SELECT p.dni, p.nombre, p.apellido, a.legajo
FROM persona p INNER JOIN alumno a ON(a.dni=p.dni)
INNER JOIN alumnocurso ac ON (ac.dni=a.dni)
INNER JOIN curso c ON (ac.codcurso=c.codcurso)
WHERE (c.nombre like "%BD%" AND ac.año=2018)
EXCEPT (
    SELECT p.dni, p.nombre, p.apellido, a.legajo
    FROM persona p INNER JOIN alumno a ON(a.dni=p.dni)
    INNER JOIN alumnocurso ac ON (ac.dni=a.dni)
    WHERE (ac.año=2019)
)

9- INSERT INTO profesores (dni, matricula, nro expediente) VALUES (111,321,222)
INSERT INTO titulo-profesor (25,111,22/2/2022)

10- UPDATE alumno set estadocivil='divorciado' WHERE legajo=2020/09

11- DELETE FROM alumno WHERE dni=30
DELETE FROM alumno-cursa WHERE dni=30
DELETE FROM persona WHERE dni=30

```