

Ejercicios de PosgreSQL

Este tutorial está pensado para que pueda ser desarrollado por una persona que conoce muy poco de SQL. Su objetivo es poder aprender PostgreSQL de forma sencilla con algunos ejercicios resueltos.

Origen: <http://postgresqlya.com.ar/>

Ejercicio 1

Necesita almacenar los datos de sus amigos en una tabla, los datos que guardará serán: apellido, nombre, domicilio y teléfono.

1. Intente crear una tabla llamada **"agenda"** (*aparece un mensaje de error porque usamos un caracter inválido "/" para el nombre*).
2. Cree una tabla llamada "agenda", debe tener los siguientes campos:
 - **apellido**, *varchar(30)*.
 - **nombre**, *varchar(20)*.
 - **domicilio**, *varchar(30)*.
 - **telefono**, *varchar(11)*.
3. Intente crearla nuevamente (*Aparece mensaje de error*).
4. Visualice la estructura de la tabla.
5. Elimine la tabla.
6. Intente eliminar nuevamente la tabla (*Debe aparecer un mensaje de error*).

Ejercicio 2

Necesita almacenar información referente a los libros de su biblioteca personal. Los datos que guardará serán: título del libro, nombre del autor y nombre de la editorial.

1. Cree una tabla llamada **"libros"** que tenga la siguiente estructura:
 - **título**, *varchar(20)*.
 - **autor**, *varchar(30)*.
 - **editorial**, *varchar(15)*.
2. Intente crearla nuevamente (*Aparece mensaje de error*).
3. Visualice la estructura de la tabla.
4. Elimine la tabla.
5. Intente eliminar la tabla nuevamente.

Ejercicio 3

Trabaje con la tabla "agenda" que almacena información de sus amigos.

1. Cree una tabla llamada **agenda** que tenga la siguiente estructura:
 - **apellido**, *cadena de 30*.
 - **nombre**, *cadena de 20*.
 - **domicilio**, *cadena de 30*.
 - **telefono**, *cadena de 11*.
2. Visualice la estructura de la tabla "agenda".
3. Ingrese los siguientes registros:
 - 'Moreno', 'Alberto', 'Colon 123', '4234567'.
 - 'Torres', 'Juan', 'Avellaneda 135', '4458787'.
4. Seleccione todos los registros de la tabla.
5. Elimine la tabla "agenda".
6. Intente eliminar la tabla nuevamente (*aparece un mensaje de error*)

Ejercicio 4

Trabaje con la tabla "libros" que almacena los datos de los libros de su propia biblioteca.

1. Cree una tabla llamada **libros** que tenga la siguiente estructura:
 - **titulo**, *cadena de 20*.
 - **autor**, *cadena de 30*.
 - **editorial**, *cadena de 15*.
2. Visualice la estructura de la tabla "libros".
3. Ingrese los siguientes registros:
 - 'El aleph', 'Borges', 'Planeta'.
 - 'Martin Fierro', 'José Hernández', 'Emece'.
 - 'Aprenda PHP', 'Mario Molina', 'Emece'.
4. Muestre todos los registros.

Ejercicio 5

Un videoclub que alquila películas en vídeo almacena la información de sus películas en una tabla llamada "películas".

1. Cree la tabla eligiendo el tipo de dato adecuado para cada campo.
 - **nombre**, *cadena de caracteres de 20 de longitud*.
 - **actor**, *cadena de caracteres de 20 de longitud*.
 - **duracion**, *valor numérico entero*.
 - **cantidadcopias**, *valor entero*.
2. Visualice la estructura de la tabla.
3. Ingrese los siguientes registros:
 - 'Misión imposible', 'Tom Cruise', 128, 3.
 - 'Misión imposible 2', 'Tom Cruise', 130, 2.
 - 'Mujer bonita', 'Julia Roberts', 118, 3.
 - 'Elsa y Fred', 'China Zorrilla', 110, 2.
4. Muestre todos los registros.

Ejercicio 6

Una empresa almacena los datos de sus empleados en una tabla "empleados" que guarda los siguientes datos: nombre, documento, sexo, domicilio, sueldobasico.

1. Cree la tabla eligiendo el tipo de dato adecuado para cada campo.
2. Visualice la estructura de la tabla.
3. Ingrese algunos registros:
 - 'Juan Pérez', '22333444', 'm', 'Sarmiento 123', 500.
 - 'Ana Acosta', '24555666', 'f', 'Colon 134', 650.
 - 'Bartolomé Barrios', '27888999', 'm', 'Urquiza 479', 800.
4. Seleccione todos los registros.

Ejercicio 7

Un videoclub que alquila películas en vídeo almacena la información de sus películas en alquiler en una tabla llamada "películas".

1. Cree una tabla con los siguientes campos:
 - **titulo**, *varchar(20)*.
 - **actor**, *varchar(20)*.
 - **duracion**, *integer*.
 - **cantidad**, *integer*.
2. Visualice la estructura de la tabla.
3. Ingrese los siguientes registros:
 - 'Misión imposible', 'Tom Cruise', 180, 3.
 - 'Misión imposible 2', 'Tom Cruise', 190, 2.
 - 'Mujer bonita', 'Julia Roberts', 118, 3.
 - 'Elsa y Fred', 'China Zorrilla', 110, 2.
4. Muestre el título y actor de todas las películas.
5. Muestre el título y duración de todas las películas.

- Muestre el título y la cantidad de copias.

Ejercicio 8

Una empresa almacena los datos de sus empleados en una tabla llamada "empleados".

- Cree una tabla con los siguientes campos:
 - nombre**, *varchar(20)*.
 - documento**, *varchar(8)*.
 - sexo**, *varchar(1)*.
 - domicilio**, *varchar(30)*.
 - sueldobasico**, *float*.
- Vea la estructura de la tabla
- Ingrese algunos registros:
 - 'Juan Juárez', '22333444', 'm', 'Sarmiento 123', 500.
 - 'Ana Acosta', '27888999', 'f', 'Colon 134', 700.
 - 'Carlos Caseres', '31222333', 'm', 'Urquiza 479', 850.
- Muestre todos los datos de los empleados.
- Muestre el nombre, documento y domicilio de los empleados.
- Muestre el documento, sexo y sueldo básico de todos los empleados.

Ejercicio 9

Trabaje con la tabla "agenda" en la que registra los datos de sus amigos.

- Cree una tabla con los siguientes campos:
 - apellido**, *cadena de 30*.
 - nombre**, *cadena de 20*.
 - domicilio**, *cadena de 30*.
 - telefono**, *cadena de 11*.
- Visualice la estructura de la tabla "agenda".
- Ingrese los siguientes registros:
 - Acosta, Ana, Colon 123, 4234567.
 - Bustamante, Betina, Avellaneda 135, 4458787.
 - López, Héctor, Salta 545, 4887788.
 - López, Luis, Urquiza 333, 4545454.
 - López, Marisa, Urquiza 333, 4545454..
- Seleccione todos los registros de la tabla
- Seleccione el registro cuyo nombre sea "Marisa" (1 registro)
- Seleccione los nombres y domicilios de quienes tengan apellido igual a "López" (3 registros)
- Muestre el nombre de quienes tengan el teléfono "4545454" (2 registros)

Ejercicio 10

Trabaje con la tabla "libros" de una librería que guarda información referente a sus libros disponibles para la venta.

- Cree una tabla llamada **libros** que posea los siguientes campos:
 - título**, *varchar(20)*.
 - autor**, *varchar(30)*.
 - editorial**, *varchar(15)*.
- Visualice la estructura de la tabla "libros".
- Ingrese los siguientes registros:
 - El (aleph), Borges, Emece.
 - Martin (Fierro), José Hernández, Emece.
 - Martin (Fierro), José Hernández, Planeta.
 - Aprenda (PHP), Mario Molina, Siglo XXI.
- Seleccione los registros cuyo autor sea "Borges" (1 registro)
- Seleccione los títulos de los libros cuya editorial sea "Emece" (2 registros)

6. Seleccione los nombres de las editoriales de los libros cuyo título sea "Martin Fierro" (2 registros)

Ejercicio 11

Un comercio que vende artículos de computación registra los datos de sus artículos en una tabla con ese nombre.

1. Cree la tabla, con la siguiente estructura:
 - **codigo**, integer.
 - **nombre**, varchar(20).
 - **descripcion**, varchar(30).
 - **precio**, float.
 - **cantidad**, integer.
2. Ingrese algunos registros:
 - 1, 'impresora', 'Epson Stylus C45', 400.80, 20.
 - 2, 'impresora', 'Epson Stylus C85', 500, 30.
 - 3, 'monitor', 'Samsung 14', 800, 10.
 - 4, 'teclado', 'ingles Biswal', 100, 50.
 - 5, 'teclado', 'español Biswal', 90, 50.
3. Seleccione los datos de las impresoras (2 registros)
4. Seleccione los artículos cuyo precio sea mayor o igual a 400 (3 registros)
5. Seleccione el código y nombre de los artículos cuya cantidad sea menor a 30 (2 registros)
6. Seleccione el nombre y descripción de los artículos que NO cuesten \$100 (4 registros)

Ejercicio 12

Un video club que alquila películas en video almacena la información de sus películas en alquiler en una tabla denominada "películas".

1. Cree la tabla eligiendo el tipo de dato adecuado para cada campo:
 - **titulo**, varchar(20).
 - **actor**, varchar(20).
 - **duracion**, integer.
 - **cantidad**, integer.
2. Ingrese los siguientes registros:
 - 'Misión imposible', 'Tom Cruise', 120, 3.
 - 'Misión imposible 2', 'Tom Cruise', 180, 4.
 - 'Mujer bonita', 'Julia R.', 90, 1.
 - 'Elsa y Fred', 'China Zorrilla', 80, 2.
3. Seleccione las películas cuya duración no supere los 90 minutos (2 registros)
4. Seleccione el título de todas las películas en las que el actor NO sea "Tom Cruise" (2 registros)
5. Muestre todos los campos, excepto "duracion", de todas las películas de las que haya más de 2 copias (2 registros)

Ejercicio 13

Trabaje con la tabla "agenda" que registra la información referente a sus amigos.

1. Cree la tabla con los siguientes campos:
 - **apellido**, varchar(30).
 - **nombre**, varchar(20).
 - **domicilio**, varchar(30).
 - **telefono**, varchar(11).
2. Ingrese los siguientes registros (insert into):
 - Álvarez, Alberto, Colon 123, 4234567.
 - Juárez, Juan, Avellaneda 135, 4458787.
 - López, María, Urquiza 333, 4545454.
 - López, José, Urquiza 333, 4545454.
 - Salas, Susana, Gral. Paz 1234, 4123456..
3. Elimine el registro cuyo nombre sea "Juan" (1 registro afectado)
4. Elimine los registros cuyo número telefónico sea igual a "4545454" (2 registros afectados)

5. Muestre la tabla.
6. Elimine todos los registros (*2 registros afectados*)
7. Muestre la tabla.

Ejercicio 14

Un comercio que vende artículos de computación registra los datos de sus artículos en una tabla con ese nombre.

1. Cree la tabla, con la siguiente estructura:
 - **codigo**, *integer*.
 - **nombre**, *varchar(20)*.
 - **descripcion**, *varchar(30)*.
 - **precio**, *float*.
 - **cantidad**, *integer*.
2. Ingrese algunos registros:
 - 1, 'impresora', 'Epson Stylus C45', 400.80, 20.
 - 2, 'impresora', 'Epson Stylus C85', 500, 30.
 - 3, 'monitor', 'Samsung 14', 800, 10.
 - 4, 'teclado', 'ingles Biswal', 100, 50.
 - 5, 'teclado', 'español Biswal', 90, 50.
3. Elimine los artículos cuyo precio sea mayor o igual a 500 (*2 registros*)
4. Elimine todas las impresoras (*1 registro*)
5. Elimine todos los artículos cuyo código sea diferente a 4 (*1 registro*)
6. Mostrar la tabla después de borrar cada registro.

Ejercicio 15

Trabaje con la tabla "agenda" que almacena los datos de sus amigos.

1. Cree una tabla con los siguientes campos:
 - **apellido**, *varchar(30)*.
 - **nombre**, *varchar(20)*.
 - **domicilio**, *varchar(30)*.
 - **telefono**, *varchar(11)*.
2. Ingrese los siguientes registros:
 - 'Acosta', 'Alberto', 'Colon 123', '4234567'.
 - 'Juárez', 'Juan', 'Avellaneda 135', '4458787'.
 - 'López', 'María', 'Urquiza 333', '4545454'.
 - 'López', 'José', 'Urquiza 333', '4545454'.
 - 'Suarez', 'Susana', 'Gral. Paz 1234', '4123456'.
3. Modifique el registro cuyo nombre sea "Juan" por "Juan José" (*1 registro afectado*)
4. Actualice los registros cuyo número telefónico sea igual a "4545454" por "4445566" (*2 registros afectados*)
5. Actualice los registros que tengan en el campo "nombre" el valor "Juan" por "Juan José" (*ningún registro afectado porque ninguno cumple con la condición del "where"*)
6. Tras cada actualización muestre los datos de la tabla.

Ejercicio 16

Trabaje con la tabla "libros" de una librería.

1. Créela con los siguientes campos:
 - **titulo**, *varchar(30)*.
 - **autor**, *varchar(20)*.
 - **editorial**, *varchar(15)*.
 - **precio**, *float*.
2. Ingrese los siguientes registros:
 - 'El aleph', 'Borges', 'Emece', 25.00.
 - 'Martin Fierro', 'José Hernández', 'Planeta', 35.50.

- 'Aprenda PHP', 'Mario Molina', 'Emece', 45.50.
 - 'Cervantes y el quijote', 'Borges', 'Emece', 25.
 - 'Matemática estas ahí', 'Paenza', 'Siglo XXI', 15.
3. Muestre todos los registros (5 registros).
 4. Modifique los registros cuyo autor sea igual a "Paenza", por "Adrián Paenza" (1 registro afectado)
 5. Modifique los registros cuyo autor sea igual a "Paenza", por "Adrián Paenza" (ningún registro afectado porque ninguno cumple la condición)
 6. Actualice el precio del libro de "Mario Molina" a 27 pesos (1 registro afectado)
 7. Actualice el valor del campo "editorial" por "Emece S.A.", para todos los registros cuya editorial sea igual a "Emece" (3 registros afectados)
 8. Tras cada actualización muestre los datos de la tabla.

Ejercicio 17

Una farmacia guarda información referente a sus medicamentos en una tabla llamada "medicamentos".

1. Cree la tabla con la siguiente estructura:
 - **codigo**, integer. not null
 - **nombre**, varchar(20). not null
 - **laboratorio**, varchar(20).
 - **precio**, float.
 - **cantidad**, integer. not null
2. Visualice la estructura de la tabla "medicamentos" indicando si el campo admite valores null.
3. Ingrese algunos registros con valores "null" para los campos que lo admitan:
 - *1, 'Sertal gotas', null, null, 100.
 - *2, 'Sertal compuesto', null, 8.90, 150.
 - *3, 'Buscapina', 'Roche', null, 200.
4. Vea todos los registros.
5. Ingrese un registro con valor "0" para el precio y cadena vacía para el laboratorio.
6. Ingrese un registro con valor "0" para el código y cantidad y cadena vacía para el nombre.
7. Muestre todos los registros.
8. Intente ingresar un registro con valor nulo para un campo que no lo admite (aparece un mensaje de error).
9. Recupere los registros que contengan valor "null" en el campo "laboratorio", luego los que tengan una cadena vacía en el mismo campo. Note que el resultado es diferente.
10. Recupere los registros que contengan valor "null" en el campo "precio", luego los que tengan el valor 0 en el mismo campo. Note que el resultado es distinto.
11. Recupere los registros cuyo laboratorio no contenga una cadena vacía, luego los que sean distintos de "null". Note que la salida de la primera sentencia no muestra los registros con cadenas vacías y tampoco los que tienen valor nulo; el resultado de la segunda sentencia muestra los registros con valor para el campo laboratorio (incluso cadena vacía).
12. Recupere los registros cuyo precio sea distinto de 0, luego los que sean distintos de "null".

Note que la salida de la primera sentencia no muestra los registros con valor 0 y tampoco los que tienen valor nulo; el resultado de la segunda sentencia muestra los registros con valor para el campo precio (incluso el valor 0).

Ejercicio 18

Trabaje con la tabla que almacena los datos sobre películas, llamada "películas".

1. Créela con la siguiente estructura:
 - **codigo**, int. not null
 - **titulo**, varchar(40). not null
 - **actor**, varchar(20).
 - **duracion**, int.
2. Visualice la estructura de la tabla note que el campo "codigo" y "titulo", en la columna "ins_nullable" muestra "NO" y los otros campos "YES".
3. Ingrese los siguientes registros:
 - 1, 'Misión imposible', 'Tom Cruise', 120.
 - 2, 'Harry Potter y la piedra filosofal', null, 180.
 - 3, 'Harry Potter y la camara secreta', 'Daniel R.', null.
 - 0, 'Misión imposible 2', '', 150.
 - 4, '', 'L. Di Caprio', 220.
 - 5, 'Mujer bonita', 'R. Gere-J. Roberts', 0.

- Recupere todos los registros para ver cómo PostgreSQL los almacenó.
- Intente ingresar un registro con valor nulo para campos que no lo admiten (*aparece un mensaje de error*)
- Muestre los registros con valor nulo en el campo "actor" y luego los que guardan una cadena vacía (*note que la salida es distinta*) (1 registro)
- Modifique los registros que tengan valor de duración desconocido (*nulo*) por "120" (1 registro actualizado)
- Coloque 'Desconocido' en el campo "actor" en los registros que tengan una cadena vacía en dicho campo (1 registro afectado)
- Muestre todos los registros. Note que el cambio anterior no afectó a los registros con valor nulo en el campo "actor".
- Elimine los registros cuyo título sea una cadena vacía (1 registro)

Ejercicio 19

Trabaje con la tabla "libros" de una librería.

- Créela con los siguientes campos, estableciendo como clave primaria el campo "codigo":
 - codigo**, *int*. not null
 - titulo**, *varchar(40)*. not null
 - autor**, *varchar(20)*.
 - editorial**, *varchar(15)*.
 - primary key**, *codigo*.
- Ingrese los siguientes registros:
 - 1, 'El aleph', 'Borges', 'Emece'.
 - 2, 'Martin Fierro', 'José Hernández', 'Planeta'.
 - 3, 'Aprenda PHP', 'Mario Molina', 'Nuevo Siglo'.
- Ingrese un registro con código repetido (*aparece un mensaje de error*)
- Intente ingresar el valor "null" en el campo "codigo"
- Intente actualizar el código del libro "Martin Fierro" a "1" (*mensaje de error*)

Ejercicio 20

Un instituto de enseñanza almacena los datos de sus estudiantes en una tabla llamada "alumnos".

- Intente crear la tabla con la siguiente estructura (*no lo permite*):
 - legajo**, *varchar(4)* not null.
 - documento**, *varchar(8)*.
 - nombre**, *varchar(30)*.
 - domicilio**, *varchar(30)*.
 - primary key**, *documento*.
 - primary key**, *legajo*.
- Cree la tabla con la siguiente estructura:
 - legajo**, *varchar(4)* not null.
 - documento**, *varchar(8)*.
 - nombre**, *varchar(30)*.
 - domicilio**, *varchar(30)*.
 - primary key**, *documento*.
- Verifique que el campo "documento" no admite valores nulos (*sí los admite*).
- Ingrese los siguientes registros:
 - 'A233', '22345345', 'Pérez Mariana', 'Colon 234'.
 - 'A567', '23545345', 'Morales Marcos', 'Avellaneda 348'.
- Intente ingresar un alumno con número de documento existente (*no lo permite*).
- Intente ingresar un alumno con documento nulo (*no lo permite*).

Ejercicio 21

Una farmacia guarda información referente a sus medicamentos en una tabla llamada "medicamentos".

- Cree la tabla con la siguiente estructura:
 - codigo**, *serial*.
 - nombre**, *varchar(20)*.

- **laboratorio**, *varchar(20)*.
 - **precio**, *float*.
 - **cantidad**, *integer*.
 - **primary key**, *codigo*.
2. Visualice la estructura de la tabla "medicamentos".
 3. Ingrese los siguientes registros:
 - 'Sertal', 'Roche', 5.2, 100.
 - 'Buscapina', 'Roche', 4.10, 200.
 - 'Amoxidal 500', 'Bayer', 15.60, 100.
 4. Verifique que el campo "código" generó los valores de modo automático.

Ejercicio 22

Un videoclub almacena información sobre sus películas en una tabla llamada "películas".

1. Créela con la siguiente estructura:
 - **codigo**, *serial*.
 - **titulo**, *cadena de 40*.
 - **actor**, *cadena de 20*.
 - **duracion**, *entero*.
 - **clave primaria**, *codigo*.
2. Visualice la estructura de la tabla "películas".
3. Ingrese los siguientes registros:
 - 'Misión imposible', 'Tom Cruise', 120.
 - 'Harry Potter y la piedra filosofal', 'xxx', 180.
 - 'Harry Potter y la camara secreta', 'xxx', 190.
 - 'Misión imposible 2', 'Tom Cruise', 120.
 - 'La vida es bella', 'zzz', 220.
4. Seleccione todos los registros y verifique la carga automática de los códigos.
5. Actualice las películas cuyo código es 3 colocando en "actor" 'Daniel R.'
6. Elimine la película 'La vida es bella'.
7. Elimine todas las películas cuya duración sea igual a 120 minutos.
8. Visualice los registros.
9. Ingrese el siguiente registro, sin valor para la clave primaria.
 - 'Mujer bonita', 'Richard Gere', 120.

Note que sigue la secuencia tomando el último valor generado, aunque ya no esté.

Ejercicio 23

Una farmacia guarda información referente a sus medicamentos en una tabla llamada "medicamentos".

1. Cree la tabla con la siguiente estructura:
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **laboratorio**, *varchar(20)*.
 - **precio**, *float*.
 - **cantidad**, *integer*.
 - **primary key**, *codigo*.
2. Ingrese los siguientes registros:
 - 'Sertal', 'Roche', 5.2, 100.
 - 'Buscapina', 'Roche', 4.10, 200.
 - 'Amoxidal 500', 'Bayer', 15.60, 100.
3. Elimine todos los registros con "DELETE".
4. Inserte los siguientes registros en la tabla:
 - 'Sertal', 'Roche', 5.2, 100.
 - 'Amoxidal 500', 'Bayer', 15.60, 100.
5. Vea los registros para verificar que continuó la secuencia al generar el valor para "codigo"
6. Vacíe la tabla con TRUNCATE TABLE reiniciando el índice (*RESTART IDENTITY*).

- Ingrese el siguiente registro:
 - 'Buscapina', 'Roche', 4.10, 200.
- Vea los registros para verificar que al cargar el código reinició la secuencia en 1.

Ejercicio 24

Una concesionaria de autos vende autos usados y almacena los datos de los autos en una tabla llamada "autos".

- Cree la tabla eligiendo el tipo de dato adecuado para cada campo, estableciendo el campo "patente" como clave primaria:
 - patente**, *char(6)*.
 - marca**, *varchar(20)*.
 - modelo**, *char(4)*.
 - precio**, *float*.
 - primary key**, *patente*.
- Ingrese los siguientes registros:
 - 'ACD123', 'Fiat 128', '1970', 15000.
 - 'ACG234', 'Renault 11', '1990', 40000.
 - 'BCD333', 'Peugeot 505', '1990', 80000.
 - 'GCD123', 'Renault Clio', '1990', 70000.
 - 'BCC333', 'Renault Megane', '1998', 95000.
 - 'BVF543', 'Fiat 128', '1975', 20000.
- Seleccione todos los autos del año 1990.
- Borre la tabla.
- Crearla nuevamente con la misma estructura pero utilizando las otras palabras claves para los tipos de datos char y varchar.
- Ingrese un registro.
- Mostrar el contenido de la tabla.

Hemos definido el campo "patente" de tipo "char" y no "varchar" porque la cadena de caracteres siempre tendrá la misma longitud (6 caracteres). Lo mismo sucede con el campo "modelo", en el cual almacenaremos el año, necesitamos 4 caracteres fijos.

Ejercicio 25

Una empresa almacena los datos de sus clientes en una tabla llamada "clientes".

- Créela eligiendo el tipo de dato más adecuado para cada campo:
 - documento**, *char(8)*.
 - apellido**, *varchar(20)*.
 - nombre**, *varchar(20)*.
 - domicilio**, *varchar(30)*.
 - telefono**, *varchar*. (11)
- Analice la definición de los campos. Se utiliza char(8) para el documento porque siempre constará de 8 caracteres. Para el número telefónico se usar "varchar" y no un tipo numérico porque si bien es un número, con él no se realizarán operaciones matemáticas.
- Ingrese algunos registros:
 - '2233344', 'Pérez', 'Juan', 'Sarmiento 980', '4342345'.
 - '2333344', 'Pérez', 'Ana', 'Colon 234'.
 - '2433344', 'García', 'Luis', 'Avellaneda 1454', '4558877'.
 - '2533344', 'Juárez', 'Ana', 'Urquiza 444', '4789900'.
- Seleccione todos los clientes de apellido "Pérez" (2 registros)

Ejercicio 26

Un banco tiene registrados las cuentas corrientes de sus clientes en una tabla llamada "cuentas".

La tabla contiene estos datos:

Nº de Cuenta	Documento	Nombre	Saldo
1234	25666777	Pedro Pérez	500000.60

2234	27888999	Juan López	-250000
3344	27888999	Juan López	4000.50
3346	32111222	Susana Molina	1000

1. Cree la tabla eligiendo el tipo de dato adecuado para almacenar los datos descritos arriba:
 - Número de cuenta: entero, no puede haber valores repetidos, clave primaria
 - Documento (del) propietario de la cuenta: cadena de caracteres de 8 de longitud (*siempre 8*), no nulo
 - Nombre (del) propietario de la cuenta: cadena de caracteres de 30 de longitud
 - Saldo (de) la cuenta: valores altos con decimales.
2. Ingrese los siguientes registros:
 - '1234', '25666777', 'Pedro Pérez', 500000.60.
 - '2234', '27888999', 'Juan López', -250000.
 - '3344', '27888999', 'Juan López', 4000.50.
 - '3346', '32111222', 'Susana Molina', 1000.
3. Seleccione todos los registros cuyo saldo sea mayor a "4000" (2 registros)
4. Muestre el número de cuenta y saldo de todas las cuentas cuyo propietario sea "Juan López" (2 registros)
5. Muestre las cuentas con saldo negativo (1 registro)
6. Muestre todas las cuentas cuyo número es igual o mayor a "3000" (2 registros).

Note que hay dos cuentas, con distinto número de cuenta, de la misma persona.

Ejercicio 27

Una empresa almacena los datos de sus empleados en una tabla "empleados" que guarda los siguientes datos: nombre, documento, sexo, domicilio, sueldobasico.

1. Cree la tabla eligiendo el tipo de dato adecuado para cada campo:
 - **nombre**, *varchar(30)*.
 - **documento**, *char(8)*.
 - **sexo**, *char(1)*.
 - **domicilio**, *varchar(30)*.
 - **sueldobasico**, *decimal(7, 2)*, --máximo estimado 99999.99.
 - **cantidadhijos**, *smallint*. --no superará los 255
2. Ingrese algunos registros:
 - 'Juan Pérez', '22333444', 'm', 'Sarmiento 123', 500, 2.
 - 'Ana Acosta', '24555666', 'f', 'Colon 134', 850, 0.
 - 'Bartolomé Barrios', '27888999', 'm', 'Urquiza 479', 10000.80, 4.
3. Ingrese un valor de "sueldobasico" con más decimales que los definidos (*redondea los decimales al valor más cercano 800.89*).
 - 'Susana Molina', '29000555', 'f', 'Salta 876', 800.888, 3.
4. Intente ingresar un sueldo que supere los 7 dígitos (*no lo permite*)
5. Muestre todos los empleados cuyo sueldo no supere los 900 pesos (1 registro).
6. Seleccione los nombres de los empleados que tengan hijos (3 registros).

Ejercicio 28

Una facultad almacena los datos de sus alumnos en una tabla denominada "alumnos".

1. Cree la tabla eligiendo el tipo de dato adecuado para cada campo:
 - **apellido**, *varchar(30)*.
 - **nombre**, *varchar(30)*.
 - **documento**, *char(8)*.
 - **domicilio**, *varchar(30)*.
 - **fechaingreso**, *date*.
 - **fechanacimiento**, *date*.
2. Setee el formato para entrada de datos de tipo fecha para que acepte valores "día-mes-año"
3. Ingrese un alumno empleando distintos separadores para las fechas.
4. Ingrese otro alumno empleando solamente un dígito para día y mes y 2 para el año.

5. Ingrese un alumnos empleando 2 dígitos para el año de la fecha de ingreso y "null" en "fechanacimiento".
6. Intente ingresar un alumno con fecha de ingreso correspondiente a "15 de marzo de 1990" pero en orden incorrecto "03-15-90": aparece un mensaje de error porque lo lee con el formato día, mes y año y no reconoce el mes 15.
7. Muestre todos los alumnos que ingresaron antes del '1-1-91'. 1 registro.
8. Muestre todos los alumnos que tienen "null" en "fechanacimiento". 1 registro.

Ejercicio 29

Un comercio que tiene un stand en una feria registra en una tabla llamada "visitantes" algunos datos de las personas que visitan o compran en su stand para luego enviarle publicidad de sus productos.

1. Cree la tabla con la siguiente estructura:
 - **nombre**, *varchar(30)*.
 - **edad**, *smallint*.
 - **sexo**, *char(1)*. default 'f'
 - **domicilio**, *varchar(30)*.
 - **ciudad**, *varchar(20)*. default 'Córdoba'
 - **telefono**, *varchar(11)*.
 - **mail**, *varchar(30)*. default 'no tiene'
 - **montocompra**, *decimal*. (6, 2)
2. Vea la información de las columnas "column_default" y "is_nullable"
3. Ingrese algunos registros sin especificar valores para algunos campos para ver cómo opera la cláusula "default".
 - 'Susana Molina', 'Colon 123', 59.80.
 - 'Marcos Torres', 29, 'Carlos Paz', 'marcostorres@hotmail.com'.
4. Use la palabra "default" para ingresar valores en un insert.
5. Ingrese un registro con "default values".

Ejercicio 30

Una pequeña biblioteca de barrio registra los préstamos de sus libros en una tabla llamada "prestamos". En ella almacena la siguiente información: título del libro, documento de identidad del socio a quien se le presta el libro, fecha de préstamo, fecha en que tiene que devolver el libro y si el libro ha sido o no devuelto.

1. Cree una tabla con los siguientes campos:
 - **título**, *varchar(40)*. not null
 - **documento**, *char(8)*. not null
 - **fechaprestamo**, *date*. not null
 - **fechadevolucion**, *date*.
 - **devuelto**, *char(1)*. default 'n'
2. Ingrese algunos registros omitiendo el valor para los campos que lo admiten.
 - 'Manual de 1 grado', '23456789', '2006-12-15', '2006-12-18'.
 - 'Alicia en el pais de las maravillas', '23456789', '2006-12-16'.
 - 'El aleph', '22543987', '2006-12-16', '2006-08-19'.
 - 'Manual de geografía 5 grado', '25555666', '2006-12-18', 's'.
3. Seleccione todos los registros
4. Ingrese un registro colocando "default" en los campos que lo admiten y vea cómo se almacenó.
5. Intente ingresar un registro con "default values" y analice el mensaje de error (*no se puede*)

Ejercicio 31

Un comercio que vende artículos de computación registra los datos de sus artículos en una tabla con ese nombre.

1. Cree una tabla con los siguientes campos:
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **descripcion**, *varchar(30)*.
 - **precio**, *decimal(9, 2)*
 - **cantidad**, *smallint*. default 0

- **primary key**, *codigo*.
2. Ingrese algunos registros:
 - *'impresora', 'Epson Stylus C45', 400.80, 20*.
 - *'impresora', 'Epson Stylus C85', 500*.
 - *'monitor', 'Samsung 14', 800*.
 - *'teclado', 'ingles Biswal', 100, 50*.
 3. El comercio quiere aumentar los precios de todos sus artículos en un 15%. Actualice todos los precios empleando operadores aritméticos.
 4. Vea el resultado
 5. Muestre todos los artículos, concatenando el nombre y la descripción de cada uno de ellos separados por coma.
 6. Reste a la cantidad de todos los teclados, el valor 5, empleando el operador aritmético menos ("-")

Ejercicio 32

Trabaje con la tabla "libros" de una librería.

1. Cree una tabla con los siguientes campos:
 - **codigo**, *serial*.
 - **titulo**, *varchar(40)*. not null
 - **autor**, *varchar(20)*. default 'Desconocido'
 - **editorial**, *varchar(20)*.
 - **precio**, *decimal(6., 2)*
 - **cantidad**, *smallint*. default 0
 - **primary key**, *codigo*.
2. Ingrese algunos registros:
 - *'El aleph', 'Borges', 'Emece', 25*.
 - *'Java en 10 minutos', 'Mario Molina', 'Siglo XXI', 50.40, 100*.
 - *'Alicia en el pais de las maravillas', 'Lewis Carroll', 'Emece', 15, 50*.
3. Muestre todos los campos de los libros y un campo extra, con el encabezado "monto_total" en la que calcule el monto total en dinero de cada libro (*precio por cantidad*)
4. Muestre el título, autor y precio de todos los libros de editorial "Emece" y agregue dos columnas extra en las cuales muestre el descuento de cada libro, con el encabezado "descuento" y el precio con un 10% de descuento con el encabezado "precio_final".
5. Muestre una columna con el título y el autor concatenados con el encabezado "título_y_autor"

Ejercicio 33

Trabaje con la tabla que almacena los datos de clientes.

1. Créela con la siguiente estructura:
 - **documento**, *char(8)*.
 - **apellido**, *varchar(20)*.
 - **nombre**, *varchar(20)*.
 - **domicilio**, *varchar(30)*.
 - **telefono**, *varchar*. (11)
2. Ingresar algunos registros:
 - *'2233344', 'Pérez', 'Juan', 'Sarmiento 980', '4342345'*.
 - *'2333344', 'Pérez', 'Ana', 'Colon 234', '2345123'*.
 - *'2433344', 'García', 'Luis', 'Avellaneda 1454', '4558877'*.
 - *'2533344', 'Juárez', 'Ana', 'Urquiza 444', '4789900'*.
3. Mostrar todos los registros disponiendo en mayúsculas el apellido y el nombre.
4. Mostrar el primer carácter del nombre.

Ejercicio 34

Una empresa tiene registrados sus clientes en una tabla llamada "clientes".

1. Créela con la siguiente estructura:
 - **codigo**, *serial*.
 - **nombre**, *varchar(30)*. not null

- **domicilio**, *varchar(30)*.
 - **ciudad**, *varchar(20)*.
 - **provincia**, *varchar*. (20)
 - **credito**, *decimal(9., 2)*
 - **primary key**, *codigo*.
2. Ingrese 5 registros:
 - 'López Marcos', 'Colon 111', 'Córdoba', 'Córdoba', 1900.56.
 - 'Pérez Ana', 'San Martín 222', 'Cruz del Eje', 'Córdoba', 450.33.
 - 'García Juan', 'Rivadavia 333', 'Villa del Rosario', 'Córdoba', 190.
 - 'Olmos Luis', 'Sarmiento 444', 'Rosario', 'Santa Fe', 670.22.
 - 'Pereyra Lucas', 'San Martín 555', 'Cruz del Eje', 'Córdoba', 500.55.
 3. Muestre todos los registros.
 4. Mostrar el campo crédito redondeado hacia arriba.

Ejercicio 35

Una facultad almacena los datos de sus alumnos en una tabla denominada "alumnos".

1. Cree la tabla eligiendo el tipo de dato adecuado para cada campo:
 - **apellido**, *varchar(30)*.
 - **nombre**, *varchar(30)*.
 - **documento**, *char(8)*.
 - **domicilio**, *varchar(30)*.
 - **fechaingreso**, *date*.
 - **fechanacimiento**, *date*.
2. Setee el formato para entrada de datos de tipo fecha para que acepte valores "día-mes-año"
3. Ingrese un alumno empleando distintos separadores para las fechas
4. Ingrese otro alumno empleando solamente un dígito para día y mes y 2 para el año
5. Ingrese un alumno empleando 2 dígitos para el año de la fecha de ingreso y "null" en "fechanacimiento"
6. Muestre todos los alumnos que ingresaron antes del '1-1-91'.
7. Muestre todos los alumnos que tienen "null" en "fechanacimiento".
8. Muestre el año de nacimiento de todos los alumnos.

Ejercicio 36

En una página web se guardan los siguientes datos de las visitas: número de visita, nombre, mail, país, fecha.

1. Créela con la siguiente estructura:
 - **numero**, *serial*.
 - **nombre**, *varchar(30)*. default 'Anónimo'
 - **mail**, *varchar(50)*.
 - **pais**, *varchar*. (20)
 - **fecha**, *timestamp*.
 - **primary key**, *numero*.
2. Ingrese algunos registros:
 - 'Ana María López', 'AnaMaria@hotmail.com', 'Argentina', '2006-10-10 10:10'.
 - 'Gustavo González', 'GustavoGGonzalez@hotmail.com', 'Chile', '2006-10-10 21:30'.
 - 'Juancito', 'JuanJosePerez@hotmail.com', 'Argentina', '2006-10-11 15:45'.
 - 'Fabiola Martínez', 'MartinezFabiola@hotmail.com', 'México', '2006-10-12 08:15'.
 - 'Fabiola Martínez', 'MartinezFabiola@hotmail.com', 'México', '2006-09-12 20:45'.
 - 'Juancito', 'JuanJosePerez@hotmail.com', 'Argentina', '2006-09-12 16:20'.
 - 'Juancito', 'JuanJosePerez@hotmail.com', 'Argentina', '2006-09-15 16:25'.
3. Ordene los registros por fecha, en orden descendente.
4. Muestre el nombre del usuario, país y el número de mes, ordenado por país (*ascendente*) y número de mes (*descendente*)
5. Muestre el país, el mes, el día y la hora y ordene las visitas por nombre del mes, del día y la hora.
6. Muestre los mail, país, ordenado por país, de todos los que visitaron la página en octubre (4 registros)

Ejercicio 37

Trabaje con la tabla llamada "medicamentos" de una farmacia.

1. Cree la tabla con la siguiente estructura:
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **laboratorio**, *varchar(20)*.
 - **precio**, *decimal(5., 2)*.
 - **cantidad**, *smallint*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros:
 - 'Sertal', 'Roche', 5.2, 100.
 - 'Buscapina', 'Roche', 4.10, 200.
 - 'Amoxidal 500', 'Bayer', 15.60, 100.
 - 'Paracetamol 500', 'Bago', 1.90, 200
 - *'Bayaspirina', 'Bayer', 2.10, 150); *
 - *'Amoxidal jarabe', 'Bayer', 5.10, 250); *
3. Recupere los códigos y nombres de los medicamentos cuyo laboratorio sea 'Roche' y cuyo precio sea menor a 5 (*1 registro cumple con ambas condiciones*)
4. Recupere los medicamentos cuyo laboratorio sea 'Roche' o cuyo precio sea menor a 5 (*4 registros*) Note que el resultado es diferente al del punto 4, hemos cambiado el operador de la sentencia anterior.
5. Muestre todos los medicamentos cuyo laboratorio NO sea "Bayer" y cuya cantidad sea=100 (*1 registro*)
6. Muestre todos los medicamentos cuyo laboratorio sea "Bayer" y cuya cantidad NO sea=100 (*2 registros*) Analice estas 2 últimas sentencias. El operador "not" afecta a la condición a la cual antecede, no a las siguientes. Los resultados de los puntos 6 y 7 son diferentes.
7. Elimine todos los registros cuyo laboratorio sea igual a "Bayer" y su precio sea mayor a 10 (*1 registro eliminado*)
8. Cambie la cantidad por 200, a todos los medicamentos de "Roche" cuyo precio sea mayor a 5 (*1 registro afectado*)
9. Borre los medicamentos cuyo laboratorio sea "Bayer" o cuyo precio sea menor a 3 (*3 registros borrados*)

Ejercicio 38

Trabajamos con la tabla "peliculas" de un vídeo club que alquila películas en vídeo.

1. Créela con la siguiente estructura:
 - **codigo**, *serial*.
 - **titulo**, *varchar(40)*. not null
 - **actor**, *varchar(20)*.
 - **duracion**, *smallint*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros:
 - 'Misión imposible', 'Tom Cruise', 120.
 - 'Harry Potter y la piedra filosofal', 'Daniel R.', 180.
 - 'Harry Potter y la cámara secreta', 'Daniel R.', 190.
 - 'Misión imposible 2', 'Tom Cruise', 120.
 - 'Mujer bonita', 'Richard Gere', 120.
 - 'Tootsie', 'D. Hoffman', 90.
 - 'Un oso rojo', 'Julio Chávez', 100.
 - 'Elsa y Fred', 'China Zorrilla', 110.
3. Recupere los registros cuyo actor sea "Tom Cruise" or "Richard Gere" (*3 registros*)
4. Recupere los registros cuyo actor sea "Tom Cruise" y duración menor a 100 (*ninguno cumple ambas condiciones*)
5. Cambie la duración a 200, de las películas cuyo actor sea "Daniel R." y cuya duración sea 180 (*1 registro afectado*)
6. Borre todas las películas donde el actor NO sea "Tom Cruise" y cuya duración sea mayor o igual a 100 (*2 registros eliminados*)

Ejercicio 39

Trabajamos con la tabla "peliculas" de un vídeo club que alquila películas en vídeo.

1. Créela con la siguiente estructura:
 - **codigo**, *serial*.
 - **titulo**, *varchar(40)*. not null

- **actor**, *varchar(20)*.
 - **duracion**, *smallint*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros:
 - 'Misión imposible', 'Tom Cruise', 120.
 - 'Harry Potter y la piedra filosofal', 'Daniel R.', null.
 - 'Harry Potter y la cámara secreta', 'Daniel R.', 190.
 - 'Misión imposible 2', 'Tom Cruise', 120.
 - 'Mujer bonita', null, 120.
 - 'Tootsie', 'D. Hoffman', 90.
 - 'Un oso rojo'.
 3. Recupere las películas cuyo actor sea nulo (2 registros)
 4. Cambie la duración a 0, de las películas que tengan duración igual a "null" (2 registros)
 5. Borre todas las películas donde el actor sea "null" y cuya duración sea 0 (1 registro)

Ejercicio 40

En una página web se guardan los siguientes datos de las visitas: número de visita, nombre, mail, pais, fechayhora de la visita.

1. Créela con la siguiente estructura:
 - **numero**, *serial*.
 - **nombre**, *varchar(30)*. default 'Anónimo'
 - **mail**, *varchar(50)*.
 - **pais**, *varchar*. (20)
 - **fechayhora**, *timestamp*.
 - **primary key**, *numero*.
2. Ingrese algunos registros:
 - 'Ana María López', 'AnaMaria@hotmail.com', 'Argentina', '2006-10-10 10:10'.
 - 'Gustavo González', 'GustavoGGonzalez@gotmail.com', 'Chile', '2006-10-10 21:30'.
 - 'Juancito', 'JuanJosePerez@hotmail.com', 'Argentina', '2006-10-11 15:45'.
 - 'Fabiola Martínez', 'MartinezFabiola@hotmail.com', 'México', '2006-10-12 08:15'.
 - 'Fabiola Martínez', 'MartinezFabiola@hotmail.com', 'México', '2006-09-12 20:45'.
 - 'Juancito', 'JuanJosePerez@gmail.com', 'Argentina', '2006-09-12 16:20'.
 - 'Juancito', 'JuanJosePerez@hotmail.com', 'Argentina', '2006-09-15 16:25'.
 - 'Federico1', 'federicogarcia@xaxamail.com', 'Argentina'.
3. Seleccione los usuarios que visitaron la página entre el '2006-09-12' y '2006-10-11' (5 registros) Note que incluye los de fecha mayor o igual al valor mínimo y menores al valor máximo, y que los valores null no se incluyen.
4. Recupere las visitas cuyo número se encuentra entre 2 y 5 (4 registros) Note que incluye los valores límites.

Ejercicio 41

Una concesionaria de autos vende autos usados y almacena la información en una tabla llamada "autos".

1. Cree la tabla con la siguiente estructura:
 - **patente**, *char(6)*.
 - **marca**, *varchar(20)*.
 - **modelo**, *char(4)*.
 - **precio**, *decimal(8., 2)*.
 - **primary key**, *patente*.
2. Ingrese algunos registros:
 - 'ACD123', 'Fiat 128', '1970', 15000.
 - 'ACG234', 'Renault 11', '1980', 40000.
 - 'BCD333', 'Peugeot 505', '1990', 80000.
 - 'GCD123', 'Renault Clio', '1995', 70000.
 - 'BCC333', 'Renault Megane', '1998', 95000.
 - 'BVF543', 'Fiat 128', '1975', 20000.
3. Seleccione todos los autos cuyo modelo se encuentre entre '1970' y '1990' usando el operador "between" y ordénelos por dicho campo(4 registros)

4. Seleccione todos los autos cuyo precio esté entre 50000 y 100000.

Ejercicio 42

Trabaje con la tabla llamada "medicamentos" de una farmacia.

1. Cree la tabla con la siguiente estructura:
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **laboratorio**, *varchar(20)*.
 - **precio**, *decimal(6., 2)*.
 - **cantidad**, *smallint*.
 - **fechavencimiento**, *date*. not null
 - **primary key**, *codigo*.
2. Ingrese algunos registros:
 - 'Sertal', 'Roche', 5.2, 1, '2005-02-01'.
 - 'Buscapina', 'Roche', 4.10, 3, '2006-03-01'.
 - 'Amoxidal 500', 'Bayer', 15.60, 100, '2007-05-01'.
 - 'Paracetamol 500', 'Bago', 1.90, 20, '2008-02-01'.
 - '*Bayaspirina', 'Bayer', 2.10, 150, '2009-12-01'); *.
 - '*Amoxidal jarabe', 'Bayer', 5.10, 250, '2010-10-01'); *.
3. Recupere los nombres y precios de los medicamentos cuyo laboratorio sea "Bayer" o "Bago" empleando el operador "in" (4 registros)
4. Seleccione los remedios cuya cantidad se encuentre entre 1 y 5 empleando el operador "between" y luego el operador "in" (2 registros)

Note que es más conveniente emplear, en este caso, el operador ""between".

Ejercicio 43

Una empresa almacena los datos de sus empleados en una tabla "empleados".

1. Cree una tabla con los siguientes campos:
 - **nombre**, *varchar(30)*.
 - **documento**, *char(8)*.
 - **domicilio**, *varchar(30)*.
 - **fechaingreso**, *date*.
 - **seccion**, *varchar(20)*.
 - **sueldo**, *decimal(6., 2)*.
 - **primary key**, *documento*.
2. Ingrese algunos registros:
 - 'Juan Pérez', '22333444', 'Colon 123', '1990-10-08', 'Gerencia', 900.50.
 - 'Ana Acosta', '23444555', 'Caseros 987', '1995-12-18', 'Secretaria', 590.30.
 - 'Lucas Duarte', '25666777', 'Sucre 235', '2005-05-15', 'Sistemas', 790.
 - 'Pamela González', '26777888', 'Sarmiento 873', '1999-02-12', 'Secretaria', 550.
 - 'Marcos Juárez', '30000111', 'Rivadavia 801', '2002-09-22', 'Contaduría', 630.70.
 - 'Yolanda Pérez', '35111222', 'Colon 180', '1990-10-08', 'Administración', 400.
 - 'Rodolfo Pérez', '35555888', 'Coronel Olmedo 588', '1990-05-28', 'Sistemas', 800.
3. Muestre todos los empleados con apellido "Pérez" empleando el operador "like" (3 registros)
4. Muestre todos los empleados cuyo domicilio comience con "Co" y tengan un "8" (2 registros)
5. Muestre todos los nombres y sueldos de los empleados cuyos sueldos incluyen centavos (3 registros)
6. Muestre los empleados que hayan ingresado en "1990" (3 registros)

Ejercicio 44

Trabaje con la tabla llamada "medicamentos" de una farmacia.

1. Cree la tabla con la siguiente estructura:
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **laboratorio**, *varchar(20)*.

- **precio**, *decimal(6., 2)*
 - **cantidad**, *smallint*.
 - **fechavencimiento**, *date*. not null
 - **numerolote**, *int*. default null
 - **primary key**, *codigo*.
2. Ingrese algunos registros:
 - 'Sertal', 'Roche', 5.2, 1, '2005-02-01', null.
 - 'Buscapina', 'Roche', 4.10, 3, '2006-03-01', null.
 - 'Amoxidal 500', 'Bayer', 15.60, 100, '2007-05-01', null.
 - 'Paracetamol 500', 'Bago', 1.90, 20, '2008-02-01', null
 - '*Bayaspirina', null, 2.10, null, '2009-12-01', null); *.
 - '*Amoxidal jarabe', 'Bayer', null, 250, '2009-12-15', null); *.
 3. Muestre la cantidad de registros empleando la función "count()" (6 registros*)
 4. Cuente la cantidad de medicamentos que tienen laboratorio conocido (5 registros)
 5. Cuente la cantidad de medicamentos que tienen precio distinto a "null" y que tienen cantidad distinto a "null", disponer alias para las columnas.
 6. Cuente la cantidad de remedios con precio conocido, cuyo laboratorio comience con "B" (2 registros)
 7. Cuente la cantidad de medicamentos con número de lote distinto de "null" (0 registros)

Ejercicio 45

Una empresa almacena los datos de sus empleados en una tabla "empleados".

1. Cree una tabla con los siguientes campos:
 - **nombre**, *varchar(30)*.
 - **documento**, *char(8)*.
 - **domicilio**, *varchar(30)*.
 - **seccion**, *varchar(20)*.
 - **sueldo**, *decimal(6., 2)*
 - **cantidadhijos**, *smallint*.
 - **primary key**, *documento*.
2. Ingrese algunos registros:
 - 'Juan Pérez', '22333444', 'Colon 123', 'Gerencia', 5000, 2.
 - 'Ana Acosta', '23444555', 'Caseros 987', 'Secretaria', 2000, 0.
 - 'Lucas Duarte', '25666777', 'Sucre 235', 'Sistemas', 4000, 1.
 - 'Pamela González', '26777888', 'Sarmiento 873', 'Secretaria', 2200, 3.
 - 'Marcos Juárez', '30000111', 'Rivadavia 801', 'Contaduría', 3000, 0.
 - 'Yolanda Pérez', '35111222', 'Colon 180', 'Administración', 3200, 1.
 - 'Rodolfo Pérez', '35555888', 'Coronel Olmedo 588', 'Sistemas', 4000, 3.
 - 'Martina Rodríguez', '30141414', 'Sarmiento 1234', 'Administración', 3800, 4.
 - 'Andrés Costa', '28444555', default, 'Secretaria', null, null.
3. Muestre la cantidad de empleados usando "count" (9 empleados)
4. Muestre la cantidad de empleados con sueldo no nulo de la sección "Secretaria" (2 empleados)
5. Muestre el sueldo más alto y el más bajo colocando un alias (5000 y 2000)
6. Muestre el valor mayor de "cantidadhijos" de los empleados "Pérez" (3 hijos)
7. Muestre el promedio de sueldos de todo los empleados (3400. Note que hay un sueldo nulo y no es tenido en cuenta)
8. Muestre el promedio de sueldos de los empleados de la sección "Secretaría" (2100)
9. Muestre el promedio de hijos de todos los empleados de "Sistemas" (2)

Ejercicio 46

Un comercio que tiene un stand en una feria registra en una tabla llamada "visitantes" algunos datos de las personas que visitan o compran en su stand para luego enviarle publicidad de sus productos.

1. Cree la tabla con la siguiente estructura:
 - **nombre**, *varchar(30)*.
 - **edad**, *smallint*.
 - **sexo**, *char(1)*. default 'f'

- **domicilio**, *varchar(30)*.
 - **ciudad**, *varchar(20)*. default 'Córdoba'
 - **telefono**, *varchar(11)*.
 - **mail**, *varchar(30)*. default 'no tiene'
 - **montocompra**, *decimal*. (6, 2)
2. Ingrese algunos registros:
 - 'Susana Molina', 35, default, 'Colon 123', default, null, null, 59.80.
 - 'Marcos Torres', 29, 'm', default, 'Carlos Paz', default, 'marcostorres@hotmail.com', 150.50.
 - 'Mariana Juárez', 45, default, default, 'Carlos Paz', null, default, 23.90.
 - 'Fabián Pérez', 36, 'm', '4556677', 'fabianperez@xaxamail.com'.
 - 'Alejandra González', 'La Falda', 280.50.
 - 'Gastón Pérez', 29, 'm', 'Carlos Paz', 'gastonperez1@gmail.com', 95.40.
 - 'Liliana Torres', 40, default, 'Sarmiento 876', default, default, default, 85.
 - 'Gabriela Duarte', 21, null, null, 'Rio Tercero', default, 'gabrielaltorres@hotmail.com', 321.50.
 3. Queremos saber la cantidad de visitantes de cada ciudad utilizando la cláusula "group by" (4 filas devueltas)
 4. Queremos la cantidad visitantes con teléfono no nulo, de cada ciudad (4 filas devueltas)
 5. Necesitamos el total del monto de las compras agrupadas por sexo (3 filas)
 6. Se necesita saber el máximo y mínimo valor de compra agrupados por sexo y ciudad (6 filas)
 7. Calcule el promedio del valor de compra agrupados por ciudad (4 filas)
 8. Cuente y agrupe por ciudad sin tener en cuenta los visitantes que no tienen mail (3 filas)

Ejercicio 47

Una empresa almacena los datos de sus empleados en una tabla "empleados".

1. Cree una tabla con los siguientes campos:
 - **nombre**, *varchar(30)*.
 - **documento**, *char(8)*.
 - **domicilio**, *varchar(30)*.
 - **seccion**, *varchar(20)*.
 - **sueldo**, *decimal(6., 2)*
 - **cantidadhijos**, *smallint*.
 - **fechaingreso**, *date*.
 - **primary key**, *documento*.
2. Ingrese algunos registros:
 - 'Juan Pérez', '22333444', 'Colon 123', 'Gerencia', 5000, 2, '1980-05-10'.
 - 'Ana Acosta', '23444555', 'Caseros 987', 'Secretaria', 2000, 0, '1980-10-12'.
 - 'Lucas Duarte', '25666777', 'Sucre 235', 'Sistemas', 4000, 1, '1985-05-25'.
 - 'Pamela González', '26777888', 'Sarmiento 873', 'Secretaria', 2200, 3, '1990-06-25'.
 - 'Marcos Juárez', '30000111', 'Rivadavia 801', 'Contaduría', 3000, 0, '1996-05-01'.
 - 'Yolanda Pérez', '35111222', 'Colon 180', 'Administración', 3200, 1, '1996-05-01'.
 - 'Rodolfo Pérez', '35555888', 'Coronel Olmedo 588', 'Sistemas', 4000, 3, '1996-05-01'.
 - 'Martina Rodríguez', '30141414', 'Sarmiento 1234', 'Administración', 3800, 4, '2000-09-01'.
 - 'Andrés Costa', '28444555', default, 'Secretaria', null, null, null.
3. Cuente la cantidad de empleados agrupados por sección (5 filas)
4. Calcule el promedio de hijos por sección (5 filas)
5. Cuente la cantidad de empleados agrupados por año de ingreso (6 filas)
6. Calcule el promedio de sueldo por sección de los empleados con hijos (4 filas)

Ejercicio 48

Una empresa tiene registrados sus clientes en una tabla llamada "clientes".

1. Créela con la siguiente estructura:
 - **codigo**, *serial*.
 - **nombre**, *varchar(30)*. not null
 - **domicilio**, *varchar(30)*.
 - **ciudad**, *varchar(20)*.

- **provincia**, *varchar*. (20)
- **telefono**, *varchar*(11).
- **primary key**, *codigo*.

2. Ingrese algunos registros:

- 'López Marcos', 'Colon 111', 'Córdoba', 'Córdoba', 'null'.
- 'Pérez Ana', 'San Martin 222', 'Cruz del Eje', 'Córdoba', '4578585'.
- 'García Juan', 'Rivadavia 333', 'Villa del Rosario', 'Córdoba', '4578445'.
- 'Pérez Luis', 'Sarmiento 444', 'Rosario', 'Santa Fe', null.
- 'Pereyra Lucas', 'San Martin 555', 'Cruz del Eje', 'Córdoba', '4253685'.
- 'Gómez Inés', 'San Martin 666', 'Santa Fe', 'Santa Fe', '0345252525'.
- 'Torres Fabiola', 'Alem 777', 'Villa del Rosario', 'Córdoba', '4554455'.
- 'López Carlos', null, 'Cruz del Eje', 'Córdoba', null.
- 'Ramos Betina', 'San Martin 999', 'Córdoba', 'Córdoba', '4223366'.
- 'López Lucas', 'San Martin 1010', 'Posadas', 'Misiones', '0457858745'.

3. Obtenga el total de los registros agrupados por ciudad y provincia (6 filas)

4. Obtenga el total de los registros agrupados por ciudad y provincia sin considerar los que tienen menos de 2 clientes (3 filas)

Ejercicio 49

Un comercio que tiene un stand en una feria registra en una tabla llamada "visitantes" algunos datos de las personas que visitan o compran en su stand para luego enviarle publicidad de sus productos.

1. Créela con la siguiente estructura:

- **nombre**, *varchar*(30).
- **edad**, *smallint*.
- **sexo**, *char*(1).
- **domicilio**, *varchar*(30).
- **ciudad**, *varchar*(20).
- **telefono**, *varchar*(11).
- **montocompra**, *decimal*(6., 2) not null

2. Ingrese algunos registros:

- '**Susana Molina', 28, 'f', null, 'Córdoba', null, 45.50); *.
- 'Marcela Mercado', 36, 'f', 'Avellaneda 345', 'Córdoba', '4545454', 22.40.
- '**Alberto García', 35, 'm', 'Gral. Paz 123', 'Alta Gracia', '03547123456', 25); *.
- 'Teresa García', 33, 'f', default, 'Alta Gracia', '03547123456', 120.
- 'Roberto Pérez', 45, 'm', 'Urquiza 335', 'Córdoba', '4123456', 33.20.
- 'Marina Torres', 22, 'f', 'Colon 222', 'Villa Dolores', '03544112233', 95.
- 'Julieta Gómez', 24, 'f', 'San Martin 333', 'Alta Gracia', null, 53.50.
- 'Roxana López', 20, 'f', 'null', 'Alta Gracia', null, 240.
- 'Liliana García', 50, 'f', 'Paso 999', 'Córdoba', '4588778', 48.
- 'Juan Torres', 43, 'm', 'Sarmiento 876', 'Córdoba', null, 15.30.

3. Obtenga el total de las compras agrupados por ciudad y sexo de aquellas filas que devuelvan un valor superior a 50 (3 filas)

4. Agrupe por ciudad y sexo, muestre para cada grupo el total de visitantes, la suma de sus compras y el promedio de compras, ordenado por la suma total y considerando las filas con promedio superior a 30 (3 filas)

Ejercicio 50

Una empresa tiene registrados sus clientes en una tabla llamada "clientes".

1. Créela con la siguiente estructura:

- **codigo**, *serial*.
- **nombre**, *varchar*(30). not null
- **domicilio**, *varchar*(30).
- **ciudad**, *varchar*(20).
- **provincia**, *varchar*. (20)
- **primary key**, *codigo*.

2. Ingrese algunos registros:

- 'López Marcos', 'Colon 111', 'Córdoba', 'Córdoba'.

- 'Pérez Ana', 'San Martin 222', 'Cruz del Eje', 'Córdoba'.
 - 'García Juan', 'Rivadavia 333', 'Villa del Rosario', 'Córdoba'.
 - 'Pérez Luis', 'Sarmiento 444', 'Rosario', 'Santa Fe'.
 - 'Pereyra Lucas', 'San Martin 555', 'Cruz del Eje', 'Córdoba'.
 - 'Gómez Inés', 'San Martin 666', 'Santa Fe', 'Santa Fe'.
 - 'Torres Fabiola', 'Alem 777', 'Villa del Rosario', 'Córdoba'.
 - 'López Carlos', null, 'Cruz del Eje', 'Córdoba'.
 - 'Ramos Betina', 'San Martin 999', 'Córdoba', 'Córdoba'.
 - 'López Lucas', 'San Martin 1010', 'Posadas', 'Misiones'.
3. Obtenga las provincias sin repetir (3 registros)
 4. Cuento las distintas provincias.
 5. Se necesitan los nombres de las ciudades sin repetir (6 registros)
 6. Obtenga la cantidad de ciudades distintas.
 7. Combine con "where" para obtener las distintas ciudades de la provincia de Córdoba (3 registros).
 8. Contamos las distintas ciudades de cada provincia empleando "group by" (3 registros).

Ejercicio 51

La provincia almacena en una tabla llamada "inmuebles" los siguientes datos de los inmuebles y sus propietarios para cobrar impuestos.

1. Créela con la siguiente estructura:
 - **documento**, varchar(8). not null
 - **apellido**, varchar(30).
 - **nombre**, varchar(30).
 - **domicilio**, varchar(20).
 - **barrio**, varchar(20).
 - **ciudad**, varchar(20).
 - **tipo**, char(1)., --b=baldío, e: edificado.
 - **superficie**, decimal. (8, 2)
2. Ingrese algunos registros:
 - '11000000', 'Pérez', 'Alberto', 'San Martin 800', 'Centro', 'Córdoba', 'e', 100.
 - '11000000', 'Pérez', 'Alberto', 'Sarmiento 245', 'Gral. Paz', 'Córdoba', 'e', 200.
 - '12222222', 'López', 'María', 'San Martin 202', 'Centro', 'Córdoba', 'e', 250.
 - '13333333', 'García', 'Carlos', 'Paso 1234', 'Alberdi', 'Córdoba', 'b', 200.
 - '13333333', 'García', 'Carlos', 'Güemes 876', 'Alberdi', 'Córdoba', 'b', 300.
 - '14444444', 'Pérez', 'Mariana', 'Caseros 456', 'Flores', 'Córdoba', 'b', 200.
 - '15555555', 'López', 'Luis', 'San Martin 321', 'Centro', 'Carlos Paz', 'e', 500.
 - '15555555', 'López', 'Luis', 'López y Planes 853', 'Flores', 'Carlos Paz', 'e', 350.
 - '16666666', 'Pérez', 'Alberto', 'Sucre 1877', 'Flores', 'Córdoba', 'e', 150.
3. Muestre los distintos apellidos de los propietarios, sin repetir (3 registros)
4. Muestre los distintos documentos de los propietarios, sin repetir (6 registros)
5. Cuento, sin repetir, la cantidad de propietarios de inmuebles de la ciudad de Córdoba (5)
6. Cuento la cantidad de inmuebles con domicilio en 'San Martin', sin repetir la ciudad (2)
7. Muestre los apellidos y nombres, sin repetir (5 registros) Note que hay 2 personas con igual nombre y apellido que aparece una sola vez.
8. Muestre la cantidad de inmuebles que tiene cada propietario agrupando por documento, sin repetir barrio (6 registros)

Ejercicio 52

Un consultorio médico en el cual trabajan 3 médicos registra las consultas de los pacientes en una tabla llamada "consultas".

1. La tabla contiene los siguientes datos:
 - **fechayhora**: timestamp not null, fecha y hora de la consulta
 - **médico**: varchar(30), not null, nombre del médico (Pérez, López, Duarte).
 - **documento**: char(8) not null, documento del paciente
 - **paciente**: varchar(30), nombre del paciente
 - **obrasocial**: varchar(30), nombre de la obra social (IPAM, PAMI, etc.)..
2. Un médico sólo puede atender a un paciente en una fecha y hora determinada. En una fecha y hora determinada, varios médicos atienden a distintos pacientes. Cree la tabla definiendo una clave primaria compuesta:

- **fechayhora**, *timestamp*. not null
 - **medico**, *varchar(30)*. not null
 - **documento**, *char(8)*. not null
 - **paciente**, *varchar(30)*.
 - **obrasocial**, *varchar(30)*.
 - **primary key**, (*fechayhora.*, *medico*).
3. Ingrese varias consultas para un mismo médico en distintas horas el mismo día.
 4. Ingrese varias consultas para diferentes médicos en la misma fecha y hora.
 5. Intente ingresar una consulta para un mismo médico en la misma hora el mismo día.

Ejercicio 53

Un club dicta clases de distintos deportes. En una tabla llamada "inscritos" almacena la información necesaria.

1. La tabla contiene los siguientes campos:
 - **documento del socio alumno**: *char(8)* not null
 - **nombre del socio**: *varchar(30)*
 - **nombre del deporte** (tenis, futbol, natación, básquet): *varchar(15)* not null.
 - **año de inscripción**: *smallint*
 - **matricula**: si la matrícula ha sido o no pagada ('s' o 'n').
2. Necesitamos una clave primaria que identifique cada registro. Un socio puede inscribirse en varios deportes en distintos años. Un socio no puede inscribirse en el mismo deporte el mismo año. Varios socios se inscriben en un mismo deporte en distintos años. Cree la tabla con una clave compuesta:
 - **documento**, *char(8)*. not null,
 - **nombre**, *varchar(30)*.
 - **deporte**, *varchar(15)*. not null
 - **año** (*date*)
 - **matricula**, *char(1)*.
 - **primary key**, (*documento.*, *deporte*, *año*).
3. Inscriba a varios alumnos en el mismo deporte en el mismo año
4. Inscriba a un mismo alumno en varios deportes en el mismo año
5. Ingrese un registro con el mismo documento de socio en el mismo deporte en distintos años
6. Intente inscribir a un socio alumno en un deporte en el cual ya esté inscrito.
7. Intente actualizar un registro para que la clave primaria se repita.

Ejercicio 54

Una empresa tiene registrados datos de sus empleados en una tabla llamada "empleados".

1. Créela con la siguiente estructura:
 - **documento**, *varchar(8)*.
 - **nombre**, *varchar(30)*.
 - **fechanacimiento**, *date*.
 - **cantidadhijos**, *smallint*.
 - **seccion**, *varchar(20)*.
 - **sueldo**, *decimal(6,2)*.
2. Agregue una restricción "check" para asegurarse que no se ingresen valores negativos para el sueldo.
3. Ingrese algunos registros válidos:
 - '22222222','Alberto Lopez','1965/10/05',1,'Sistemas',1000.
 - '33333333','Beatriz Garcia','1972/08/15',2,'Administracion',3000.
 - '34444444','Carlos Caseres','1980/10/05',0,'Contaduría',6000.
4. Intente agregar otra restricción "check" al campo sueldo para asegurar que ninguno supere el valor 5000 (*La sentencia no se ejecuta porque hay un sueldo que no cumple la restricción*).
5. Elimine el registro infractor y vuelva a crear la restricción.
6. Establezca una restricción para controlar que la fecha de nacimiento que se ingresa no supere la fecha actual.
7. Establezca una restricción "check" para "cantidadhijos" que permita solamente valores entre 0 y 15.
8. Vea todas las restricciones de la tabla (5 filas)
9. Intente agregar un registro que vaya contra alguna de las restricciones al campo "sueldo" (*Mensaje de error porque se infringe la restricción*)

"CK_empleados_sueldo_positivo").

10. Intente agregar un registro con fecha de nacimiento futura (*Mensaje de error*).
11. Intente modificar un registro colocando en "cantidadhijos" el valor "21". (*Mensaje de error*).

Ejercicio 55

Una playa de estacionamiento almacena los datos de los vehículos que ingresan en la tabla llamada "vehiculos".

1. Cree una tabla con los siguientes campos:
 - **numero**, *serial*.
 - **patente**, *char(6)*.
 - **tipo**, *char(4)*.
 - **fechahoraentrada**, *timestamp*.
 - **fechahorasalida**, *timestamp*.
 - **primary key**, *numero*.
2. Ingresamos algunos registros:
 - 'AIC124', 'auto', '2007/01/17 8:05', '2007/01/17 12:30'.
 - 'CAA258', 'auto', '2007/01/17 8:10', null.
 - 'DSE367', 'moto', '2007/01/17 8:30', '2007/01/17 18:00'.
3. Agregue una restricción "check" para asegurarse que la fecha de entrada a la playa no sea posterior a la fecha y hora actual
4. Agregue otra restricción "check" al campo "fechahoraentrada" que establezca que sus valores no sean posteriores a "fechahorasalida"
5. Intente ingresar un valor que no cumpla con la primera restricción establecida en el campo "fechahoraentrada"
6. Intente modificar un registro para que la salida sea anterior a la entrada Mensaje de error.
7. Vea todas las restricciones para la tabla "vehiculos".
8. Vea todos los registros

Ejercicio 56

Una empresa tiene registrados datos de sus empleados en una tabla llamada "empleados".

1. Créela con la siguiente estructura:
 - **documento**, *varchar(8)*. not null
 - **nombre**, *varchar(30)*.
 - **seccion**, *varchar(20)*.
2. Ingrese algunos registros, dos de ellos con el mismo número de documento:
 - '22222222', 'Alberto López', 'Sistemas'.
 - '23333333', 'Beatriz García', 'Administración'.
 - '23333333', 'Carlos Fuentes', 'Administración'.
3. Intente establecer una restricción "primary key" para la tabla para que el documento no se repita ni admita valores nulos No lo permite porque la tabla contiene datos que no cumplen con la restricción, debemos eliminar (*o modificar*) el registro que tiene documento duplicado.
4. Establezca la restricción "primary key" del punto 3
5. Intente actualizar un documento para que se repita. No lo permite porque va contra la restricción.
6. Intente establecer otra restricción "primary key" con el campo "nombre". No lo permite, sólo puede haber una restricción "primary key" por tabla.
7. Intente ingresar un registro con valor nulo para el documento. No lo permite porque la restricción no admite valores nulos.
8. Vea las restricciones de la tabla empleados (2 filas)

Ejercicio 57

Una empresa de remises tiene registrada la información de sus vehículos en una tabla llamada "remis".

1. Cree la tabla con la siguiente estructura:
 - **numero**, *serial*.
 - **patente**, *char(6)*.
 - **marca**, *varchar(15)*.
 - **modelo**, *char(4)*.
2. Ingrese algunos registros sin repetir patente:
3. Definir una restricción "primary key" para el campo "patente".

4. Establezca una restricción "primary key" para el campo "numero". (*No lo permite ya que hay una "primary key"*)
5. Vea la información de las restricciones

Ejercicio 58

Una empresa de remises tiene registrada la información de sus vehículos en una tabla llamada "remis".

1. Cree la tabla con la siguiente estructura:
 - **numero**, *serial*.
 - **patente**, *char(6)*.
 - **marca**, *varchar(15)*.
 - **modelo**, *char(4)*.
2. Ingrese algunos registros, 2 de ellos con patente repetida y alguno con patente nula:
3. Intente agregar una restricción "unique" para asegurarse que la patente del remis no tomará valores repetidos. No se puede porque hay valores duplicados.
4. Elimine el registro con patente duplicada y establezca la restricción. Note que hay 1 registro con valor nulo en "patente".
5. Intente ingresar un registro con patente repetida (*no lo permite*)
6. Ingresar un registro con valor nulo para el campo "patente". Lo permite.
7. Muestre la información de las restricciones

Ejercicio 59

Una playa de estacionamiento almacena cada día los datos de los vehículos que ingresan en la tabla llamada "vehiculos".

1. Cree una tabla con los siguientes campos:
 - **patente**, *char(6)*. not null
 - **tipo**, *char(1)*., --'a'=auto, 'm'=moto.
 - **horallegada**, *timestamp*. not null
 - **horasalida**, *timestamp*.
2. Agregue una restricción "primary key" que incluya los campos "patente" y "horallegada"
3. Ingrese un vehículo.
4. Intente ingresar un registro repitiendo la clave primaria. No se permite.
5. Ingrese un registro repitiendo la patente pero no la hora de llegada.
6. Ingrese un registro repitiendo la hora de llegada pero no la patente.
7. Vea todas las restricciones para la tabla "vehiculos"
8. Elimine la restricción "primary key".
9. Vea si se han eliminado

Ejercicio 60

1. Cree la tabla con la siguiente estructura:
 - **apellido**, *varchar(30)*.
 - **nombre**, *varchar(20)*. not null
 - **domicilio**, *varchar(30)*.
 - **telefono**, *varchar(11)*.
 - **mail**, *varchar(30)*.
2. Ingrese los siguientes registros:
 - 444', '0354545256', 'perezluisalberto@hotmail.com'.
3. Cree un índice común por el campo apellido.
4. Cree un índice único por el mail.
5. Borre los dos índices.

Ejercicio 61

Trabaje con la tabla "agenda" que registra la información referente a sus amigos.

1. Cree la tabla con la siguiente estructura:
 - **apellido**, *varchar(30)*.

- **nombre**, *varchar(20)*. not null
 - **domicilio**, *varchar(30)*.
 - **telefono**, *varchar(11)*.
 - **mail**, *varchar(30)*.
2. Ingrese 5 registros.
 3. Realice una consulta limitando la salida a sólo 3 registros.
 4. Muestre los registros desde el 2 al 4.
 5. Muestre 4 registros a partir del 2 ordenado por apellido.

Ejercicio 62

Una empresa tiene registrados sus clientes en una tabla llamada "clientes", también tiene una tabla "provincias" donde registra los nombres de las provincias.

1. Créelas con las siguientes estructuras:
 - clientes
 - **codigo**, *serial*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **ciudad**, *varchar(20)*.
 - **codigoprovincia**, *smallint*. not null
 - **primary key**, *codigo*.
 - provincias
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros para ambas tablas:
 - 'López Marcos', 'Colon 111', 'Córdoba', 1.
 - 'Pérez Ana', 'San Martin 222', 'Cruz del Eje', 1.
 - 'García Juan', 'Rivadavia 333', 'Villa María', 1.
 - 'Pérez Luis', 'Sarmiento 444', 'Rosario', 2.
 - 'Pereyra Lucas', 'San Martin 555', 'Cruz del Eje', 1.
 - 'Gómez Inés', 'San Martin 666', 'Santa Fe', 2.
 - 'Torres Fabiola', 'Alem 777', 'Ibera', 3.
3. Obtenga los datos de ambas tablas, usando alias
4. Obtenga la misma información anterior pero ordenada por nombre de provincia.
5. Recupere los clientes de la provincia "Santa Fe" (2 registros devueltos)

Ejercicio 63

Un club dicta clases de distintos deportes. Almacena la información en una tabla llamada "inscritos" que incluye el documento, el nombre, el deporte y si la matricula esta paga o no y una tabla llamada "inasistencias" que incluye el documento, el deporte y la fecha de la inasistencia.

1. Cree las tablas:
 - inscritos
 - **nombre**, *varchar(30)*.
 - **documento**, *char(8)*.
 - **deporte**, *varchar(15)*.
 - **matricula**, *char(1)*., --'s'=paga 'n'=impaga
 - **primary key**, (*documento*., *deporte*).
 - inasistencias*
 - **documento**, *char(8)*.
 - **deporte**, *varchar(15)*.
 - **fecha**, *date*.
2. Ingrese algunos registros para ambas tablas.
3. Muestre el nombre, el deporte y las fechas de inasistencias, ordenado por nombre y deporte.
4. Obtenga el nombre, deporte y las fechas de inasistencias de un determinado inscripto en un determinado deporte (3 registros)
5. Obtenga el nombre, deporte y las fechas de inasistencias de todos los inscritos que pagaron la matricula (4 registros)

Note que la condición es compuesta porque para identificar los registros de la tabla "inasistencias" necesitamos ambos campos.

Ejercicio 64

Una empresa tiene registrados sus clientes en una tabla llamada "clientes", también tiene una tabla "provincias" donde registra los nombres de las provincias.

1. Cree las tablas:
 - clientes
 - **codigo**, *serial*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **ciudad**, *varchar(20)*.
 - **codigoprovincia**, *smallint*. not null
 - **primary key**, *codigo*.
 - provincias
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros para ambas tablas:
 - 'López Marcos', 'Colon 111', 'Córdoba', 1.
 - 'Pérez Ana', 'San Martin 222', 'Cruz del Eje', 1.
 - 'García Juan', 'Rivadavia 333', 'Villa María', 1.
 - 'Pérez Luis', 'Sarmiento 444', 'Rosario', 2.
 - 'Gómez Inés', 'San Martin 666', 'Santa Fe', 2.
 - 'Torres Fabiola', 'Alem 777', 'La Plata', 4.
 - 'García Luis', 'Sucre 475', 'Santa Rosa', 5.
3. Muestre todos los datos de los clientes, incluido el nombre de la provincia
4. Realice la misma consulta anterior pero alterando el orden de las tablas
5. Muestre solamente los clientes de las provincias que existen en "provincias" (5 registros)
6. Muestre todos los clientes cuyo código de provincia NO existe en "provincias" ordenados por nombre del cliente (2 registros)
7. Obtenga todos los datos de los clientes de "Córdoba" (3 registros)

Ejercicio 65

Una empresa tiene registrados sus clientes en una tabla llamada "clientes", también tiene una tabla "provincias" donde registra los nombres de las provincias.

1. Cree las tablas:
 - clientes
 - **codigo**, *serial*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **ciudad**, *varchar(20)*.
 - **codigoprovincia**, *smallint*. not null
 - **primary key**, *codigo*.
 - provincias
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros para ambas tablas:
 - 'López Marcos', 'Colon 111', 'Córdoba', 1.
 - 'Pérez Ana', 'San Martin 222', 'Cruz del Eje', 1.
 - 'García Juan', 'Rivadavia 333', 'Villa María', 1.
 - 'Pérez Luis', 'Sarmiento 444', 'Rosario', 2.
 - 'Gómez Inés', 'San Martin 666', 'Santa Fe', 2.
 - 'Torres Fabiola', 'Alem 777', 'La Plata', 4.
 - 'García Luis', 'Sucre 475', 'Santa Rosa', 5.

3. Muestre todos los datos de los clientes, incluido el nombre de la provincia empleando un "right join".
4. Obtenga la misma salida que la consulta anterior pero empleando un "left join".
5. Empleando un "right join", muestre solamente los clientes de las provincias que existen en "provincias" (5 registros)
6. Muestre todos los clientes cuyo código de provincia NO existe en "provincias" ordenados por ciudad (2 registros)

Ejercicio 66

Un club dicta clases de distintos deportes. Almacena la información en una tabla llamada "deportes" en la cual incluye el nombre del deporte y el nombre del profesor y en otra tabla llamada "inscritos" que incluye el documento del socio que se inscribe, el deporte y si la matricula está paga o no.

1. Cree las tablas:
 - deportes
 - **codigo**, serial.
 - **nombre**, varchar(30).
 - **profesor**, varchar(30).
 - **primary key**, codigo.
 - inscritos
 - **documento**, char(8).
 - **codigodeporte**, smallint. not null
 - **matricula**, char(1). --'s'=paga 'n'=impaga
2. Ingrese algunos registros para ambas tablas.
3. Muestre todos la información de la tabla "inscritos", y consulte la tabla "deportes" para obtener el nombre de cada deporte (6 registros)
4. Empleando un "left join" con "deportes" obtenga todos los datos de los inscritos (7 registros)
5. Obtenga la misma salida anterior empleando un "right join".
6. Muestre los deportes para los cuales no hay inscritos, empleando un "left join" (1 registro)
7. Muestre los documentos de los inscritos a deportes que no existen en la tabla "deportes" (1 registro)
8. Emplee un "full join" para obtener todos los datos de ambas tablas, incluyendo las inscripciones a deportes inexistentes en "deportes" y los deportes que no tienen inscritos (8 registros)

Ejercicio 67

Una agencia matrimonial almacena la información de sus clientes de sexo femenino en una tabla llamada "mujeres" y en otra la de sus clientes de sexo masculino llamada "varones".

1. Cree las tablas:
 - mujeres
 - **nombre**, varchar(30).
 - **domicilio**, varchar(30).
 - **edad**, int.
 - varones
 - **nombre**, varchar(30).
 - **domicilio**, varchar(30).
 - **edad**, int.
2. Ingrese los siguientes registros.
3. La agencia necesita la combinación de todas las personas de sexo femenino con las de sexo masculino. Use un "cross join" (12 registros)
4. Realice la misma combinación pero considerando solamente las personas mayores de 40 años (6 registros)
5. Forme las parejas pero teniendo en cuenta que no tengan una diferencia superior a 10 años (8 registros)

Ejercicio 68

Una empresa de seguridad almacena los datos de sus guardias de seguridad en una tabla llamada "guardias". También almacena los distintos sitios que solicitaron sus servicios en una tabla llamada "tareas".

1. Cree las tablas:
 - guardias
 - **documento**, char(8).
 - **nombre**, varchar(30).

- **sexo**, **char(1), /'f' o 'm' */*
- **domicilio**, *varchar(30)*.
- **primary key**, *documento*.
- tareas
 - **codigo**, *serial*.
 - **domicilio**, *varchar(30)*.
 - **descripcion**, *varchar(30)*.
 - **horario**, *char(2), /'AM' o 'PM' /*
 - **primary key**, *codigo*.
- 2. Ingrese los siguientes registros:
 - 'Colon 1111', 'vigilancia exterior', 'AM'.
 - 'Urquiza 234', 'vigilancia exterior', 'PM'.
 - 'Perú 345', 'vigilancia interior', 'AM'.
 - 'Avellaneda 890', 'vigilancia interior', 'PM'.
- 3. La empresa quiere que todos sus empleados realicen todas las tareas. Realice una "cross join" (20 registros)
- 4. En este caso, la empresa quiere que todos los guardias de sexo femenino realicen las tareas de "vigilancia interior" y los de sexo masculino de "vigilancia exterior". Realice una "cross join" con un "where" que controle tal requisito (10 registros)

Ejercicio 69

Una agencia matrimonial almacena la información de sus clientes en una tabla llamada "clientes".

1. Cree una tabla con los siguientes campos:
 - **nombre**, *varchar(30)*.
 - **sexo**, *char(1), --'f'=femenino, 'm'=masculino*.
 - **edad**, *int*.
 - **domicilio**, *varchar(30)*.
2. Ingrese los siguientes registros.
3. La agencia necesita la combinación de todas las personas de sexo femenino con las de sexo masculino. Use un "cross join" (12 registros)
4. Obtenga la misma salida anterior pero realizando un "join".
5. Realice la misma autocombinación que el punto 3 pero agregue la condición que las parejas no tengan una diferencia superior a 5 años (5 registros)

Ejercicio 70

Varios clubes de barrio se organizaron para realizar campeonatos entre ellos. La tabla llamada "equipos" guarda la información de los distintos equipos que jugarán.

1. Cree una tabla con los siguientes campos:
 - **nombre**, *varchar(30)*.
 - **barrio**, *varchar(20)*.
 - **domicilio**, *varchar(30)*.
 - **entrenador**, *varchar(30)*.
2. Ingrese los siguientes registros.
3. Cada equipo jugará con todos los demás 2 veces, una vez en cada sede. Realice un "cross join" para combinar los equipos teniendo en cuenta que un equipo no juega consigo mismo (12 registros)
4. Obtenga el mismo resultado empleando un "join".
5. Realice un "cross join" para combinar los equipos para que cada equipo juegue con cada uno de los otros una sola vez (6 registros)

Ejercicio 71

Un comercio que tiene un stand en una feria registra en una tabla llamada "visitantes" algunos datos de las personas que visitan o compran en su stand para luego enviarle publicidad de sus productos y en otra tabla llamada "ciudades" los nombres de las ciudades.

1. Cree las tablas:
 - visitantes
 - **nombre**, *varchar(30)*.
 - **edad**, *smallint*.

- **sexo**, *char(1)*. default 'f'
 - **domicilio**, *varchar(30)*.
 - **codigociudad**, *smallint*. not null
 - **mail**, *varchar(30)*.
 - **montocompra**, *decimal*. (6, 2)
 - ciudades
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros:
 - 'Susana Molina', 35, 'f', 'Colon 123', 1, null, 59.80.
 - 'Marcos Torres', 29, 'm', 'Sucre 56', 1, 'marcostorres@hotmail.com', 150.50.
 - 'Mariana Juárez', 45, 'f', 'San Martin 111', 2, null, 23.90.
 - 'Fabián Pérez', 36, 'm', 'Avellaneda 213', 3, 'fabianperez@xaxamail.com', 0.
 - 'Alejandra García', 28, 'f', null, 2, null, 280.50.
 - 'Gastón Pérez', 29, 'm', null, 5, 'gastonperez1@gmail.com', 95.40.
 - 'Mariana Juárez', 33, 'f', null, 2, null, 90.
 3. Cuente la cantidad de visitas por ciudad mostrando el nombre de la ciudad (3 filas)
 4. Muestre el promedio de gastos de las visitas agrupados por ciudad y sexo (4 filas)
 5. Muestre la cantidad de visitantes con mail, agrupados por ciudad (3 filas)
 6. Obtenga el monto de compra más alto de cada ciudad (3 filas)

Ejercicio 72

Un club dicta clases de distintos deportes. En una tabla llamada "socios" guarda los datos de los socios, en una tabla llamada "deportes" la información referente a los diferentes deportes que se dictan y en una tabla denominada "inscritos", las inscripciones de los socios a los distintos deportes.

Un socio puede inscribirse en varios deportes el mismo año. Un socio no puede inscribirse en el mismo deporte el mismo año. Distintos socios se inscriben en un mismo deporte en el mismo año.

1. Cree las tablas con las siguientes estructuras:
 - Socios
 - **documento**, *char(8)*. not null,
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **primary key**, *documento*.
 - Deportes
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **profesor**, *varchar(15)*.
 - **primary key**, *codigo*.
 - Inscritos
 - **documento**, *char(8)*. not null,
 - **codigodeporte**, *smallint*. not null
 - **anio**, *char(4)*.
 - **matricula**, *char(1)*., --'s'=paga, 'n'=impaga.
 - ***primary key**, (*documento*., *codigodeporte*, *anio*).
2. Ingrese algunos registros en "socios".
3. Ingrese algunos registros en "deportes".
4. Inscriba a varios socios en el mismo deporte en el mismo año.
5. Inscriba a un mismo socio en el mismo deporte en distintos años.
6. Inscriba a un mismo socio en distintos deportes el mismo año.
7. Ingrese una inscripción con un código de deporte inexistente y un documento de socio que no exista en "socios".
8. Muestre el nombre del socio, el nombre del deporte en que se inscribió y el año empleando diferentes tipos de join.
9. Muestre todos los datos de las inscripciones (*excepto los códigos*) incluyendo aquellas inscripciones cuyo código de deporte no existe en "deportes" y cuyo documento de socio no se encuentra en "socios".
10. Muestre todas las inscripciones del socio con documento "22222222".

Ejercicio 73

Una empresa tiene registrados sus clientes en una tabla llamada "clientes", también tiene una tabla "provincias" donde registra los nombres de las provincias.

1. Cree las tablas "clientes" y "provincias":
 - clientes
 - **codigo**, *serial*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **ciudad**, *varchar(20)*.
 - **codigoprovincia**, *smallint*.
 - **primary key**, *codigo*.
 - provincias
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **primary key**, *codigo*.
2. Intente agregar una restricción "foreign key" a la tabla "clientes" que haga referencia al campo "codigo" de "provincias" (*No se puede porque "provincias" no tiene restricción "primary key" "unique"*)
3. Establezca una restricción "primary key" al campo "codigo" de "provincias"
4. Ingrese algunos registros para ambas tablas:
 - 'Pérez Juan', 'San Martín 123', 'Carlos Paz', 1.
 - 'Moreno Marcos', 'Colón 234', 'Rosario', 2.
 - 'Acosta Ana', 'Avellaneda 333', 'Posadas', 3.
 - 'Luisa López', 'Juárez 555', 'La Plata', 6.
5. Intente agregar la restricción "foreign key" del punto 2 a la tabla "clientes". No se puede porque hay un registro en "clientes" cuyo valor de "codigoprovincia" no existe en "provincias".
6. Elimine el registro de "clientes" que no cumple con la restricción y establezca la restricción nuevamente.
7. Intente agregar un cliente con un código de provincia inexistente en "provincias". No se puede.
8. Intente eliminar el registro con código 3, de "provincias". No se puede porque hay registros en "clientes" al cual hace referencia.
9. Elimine el registro con código "4" de "provincias". Se permite porque en "clientes" ningún registro hace referencia a él.
10. Intente modificar el registro con código 1, de "provincias". No se puede porque hay registros en "clientes" al cual hace referencia.

En este ejemplo, el campo "codigoprovincia" de "clientes" es una clave foránea, se emplea para enlazar la tabla "clientes" con "provincias".

Ejercicio 74

Una empresa registra los datos de sus clientes en una tabla llamada "clientes". Dicha tabla contiene un campo que hace referencia al cliente que lo recomendó denominado "referenciadopor". Si un cliente no ha sido referenciado por ningún otro cliente, tal campo almacena "null".

1. Creemos la tabla:
 - **codigo**, *int*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **ciudad**, *varchar(20)*.
 - **referenciadopor**, *int*.
 - **primary key**, *codigo*.
2. Ingresamos algunos registros.
3. Intente agregar una restricción "foreign key" para evitar que en el campo "referenciadopor" se ingrese un valor de código de cliente que no exista. No se permite porque existe un registro que no cumple con la restricción que se intenta establecer.
4. Cambie el valor inválido de "referenciadopor" del registro que viola la restricción por uno válido.
5. Agregue la restricción "foreign key" que intentó agregar en el punto 3.
6. Intente agregar un registro que infrinja la restricción. No lo permite.
7. Intente modificar el código de un cliente que está referenciado en "referenciadopor". No se puede.
8. Intente eliminar un cliente que sea referenciado por otro en "referenciadopor". No se puede.
9. Cambie el valor de código de un cliente que no referenció a nadie.
10. Elimine un cliente que no haya referenciado a otros.

Ejercicio 75

Una empresa tiene registrados sus clientes en una tabla llamada "clientes", también tiene una tabla "provincias" donde registra los nombres de las provincias.

1. Cree las tablas "clientes" y "provincias":
 - clientes
 - **codigo**, *serial*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **ciudad**, *varchar(20)*.
 - **codigoprovincia**, *smallint*.
 - **primary key**, *codigo*.
 - provincias
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros para ambas tablas:
 - 'Pérez Juan', 'San Martín 123', 'Carlos Paz', 1.
 - 'Moreno Marcos', 'Colón 234', 'Rosario', 2.
 - 'Acosta Ana', 'Avellaneda 333', 'Posadas', 3.
3. Establezca una restricción "foreign key" especificando la acción "en cascade" para actualizaciones y "no_action" para eliminaciones.
4. Intente eliminar el registro con código 3, de "provincias". No se puede porque hay registros en "clientes" al cual hace referencia y la opción para eliminaciones se estableció como "no action".
5. Modifique el registro con código 3, de "provincias".
6. Verifique que el cambio se realizó en cascada, es decir, que se modificó en la tabla "provincias" y en "clientes".
7. Intente modificar la restricción "foreign key" para que permita eliminación en cascada. Mensaje de error, no se pueden modificar las restricciones.
8. Intente eliminar la tabla "provincias". No se puede eliminar porque una restricción "foreign key" hace referencia a ella.

Ejercicio 76

Un supermercado almacena en una tabla denominada "proveedores" los datos de las compañías que le proveen de mercaderías; en una tabla llamada "clientes", los datos de los comercios que le compran y en otra tabla "empleados" los datos de los empleados.

1. Cree las tablas.
 - proveedores
 - **codigo**, *serial*.
 - **nombre**, *varchar*. (30)
 - **domicilio**, *varchar(30)*.
 - **primary key**, *codigo*.
 - clientes
 - **codigo**, *serial*.
 - **nombre**, *varchar*. (30)
 - **domicilio**, *varchar(30)*.
 - **primary key**, *codigo*.
 - empleados
 - **documento**, *char(8)*. not null
 - **nombre**, *varchar(20)*.
 - **apellido**, *varchar(20)*.
 - **domicilio**, *varchar(30)*.
 - **primary key**, *documento*.
2. Ingrese algunos registros.
3. El supermercado quiere enviar una tarjeta de salutación a todos los proveedores, clientes y empleados y necesita el nombre y domicilio de todos ellos. Emplee el operador "union" para obtener dicha información de las tres tablas.
4. Agregue una columna con un literal para indicar si es un proveedor, un cliente o un empleado y ordene por dicha columna.

Ejercicio 77

Un profesor almacena el documento, nombre y la nota final de cada alumno de su clase en una tabla llamada "alumnos".

1. Créela
 - **documento**, *char(8)*.
 - **nombre**, *varchar(30)*.
 - **nota**, *decimal(4., 2)*
 - **primary key**, *documento*.
2. Ingrese algunos registros:
3. Obtenga todos los datos de los alumnos con la nota más alta, empleando subconsulta (*2 registros*).
4. Realice la misma consulta anterior pero intente que la consulta interna retorne, además del máximo valor de precio, el título. Mensaje de error, porque la lista de selección de una subconsulta que va luego de un operador de comparación puede incluir sólo un campo o expresión (*excepto si se emplea "exists" o "in"*).
5. Muestre los alumnos que tienen una nota menor al promedio, su nota, y la diferencia con el promedio (*3 registros*).
6. Cambie la nota del alumno que tiene la menor nota por 4 (*1 registro modificado*).
7. Elimine los alumnos cuya nota es menor al promedio (*3 registros eliminados*).

Ejercicio 78

Una empresa tiene registrados sus clientes en una tabla llamada "clientes", también tiene una tabla "ciudades" donde registra los nombres de las ciudades.

1. Cree la tabla "clientes" (*codigo, nombre, domicilio, ciudad, codigociudad*) y "ciudades" (*codigo, nombre*). Agregue una restricción "primary key" para el campo "codigo" de ambas tablas.
 - clientes
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **primary key**, *codigo*.
 - ciudades
 - **codigo**, *serial*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **codigociudad**, *smallint*. not null
 - **primary key**, *codigo*.
2. Ingrese algunos registros para ambas tablas:
3. Necesitamos conocer los nombres de las ciudades de aquellos clientes cuyo domicilio es en calle "San Martin", empleando subconsulta (*3 registros*).
4. Obtenga la misma salida anterior pero empleando join.
5. Obtenga los nombres de las ciudades de los clientes cuyo apellido no comienza con una letra específica, empleando subconsulta (*2 registros*).
6. Pruebe la subconsulta del punto 5 separada de la consulta exterior para verificar que retorna una lista de valores de un solo campo.

Ejercicio 79

Un club dicta clases de distintos deportes a sus socios. El club tiene una tabla llamada "inscritos" en la cual almacena el número de "socio", el código del deporte en el cual se inscribe y la cantidad de cuotas pagas (*desde 0 hasta 10 que es el total por todo el año*), y una tabla denominada "socios" en la que guarda los datos personales de cada socio.

1. Cree las tablas:
 - inscritos
 - **numero**, *serial*.
 - **documento**, *char(8)*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **primary key**, *numero*.
 - socios
 - **numerosocio**, *int*. not null
 - **deporte**, *varchar(20)*. not null
 - **cuotas**, *smallint*.

- **primary key**, (*numerosocio*., *deporte*).
2. Ingrese algunos registros:
 3. Muestre el número de socio, el nombre del socio y el deporte en que está inscripto con un join de ambas tablas.
 4. Muestre los socios que serán compañeros en tenis y también en natación (*empleando subconsulta*) (3 filas devueltas).
 5. Vea si el socio 1 se ha inscripto en algún deporte en el cual se haya inscripto el socio 2 (3 filas.)
 6. Obtenga el mismo resultado anterior pero empleando join.
 7. Muestre los deportes en los cuales el socio 2 pagó más cuotas que ALGUN deporte en los que se inscribió el socio (2 registros).
 8. Muestre los deportes en los cuales el socio 2 pagó más cuotas que TODOS los deportes en que se inscribió el socio (1 registro).
 9. Cuando un socio no ha pagado la matrícula de alguno de los deportes en que se ha inscripto, se lo borra de la inscripción de todos los deportes. Elimine todos los socios que no pagaron ninguna cuota en algún deporte (7 registros).

Ejercicio 80

Un club dicta clases de distintos deportes a sus socios. El club tiene una tabla llamada "inscritos" en la cual almacena el número de "socio", el código del deporte en el cual se inscribe y la cantidad de cuotas pagas (*desde 0 hasta 10 que es el total por todo el año*), y una tabla denominada "socios" en la que guarda los datos personales de cada socio.

1. Cree las tablas:
 - inscritos
 - **numero**, *serial*.
 - **documento**, *char(8)*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **primary key**, *numero*. *socios
 - **numerosocio**, *int*. not null
 - **deporte**, *varchar(20)*. not null
 - **cuotas**, *smallint*.
 - **primary key**, (*numerosocio*., *deporte*).
2. Ingrese algunos registros:
3. Se necesita un listado de todos los socios que incluya nombre y domicilio, la cantidad de deportes a los cuales se ha inscripto, empleando subconsulta (4 registros).
4. Se necesita el nombre de todos los socios, el total de cuotas que debe pagar (10 por cada deporte) y el total de cuotas pagas, empleando subconsulta (4 registros).
5. Obtenga la misma salida anterior empleando join.

Ejercicio 81

Un club dicta clases de distintos deportes a sus socios. El club tiene una tabla llamada "inscritos" en la cual almacena el número de "socio", el código del deporte en el cual se inscribe y la cantidad de cuotas pagas (*desde 0 hasta 10 que es el total por todo el año*), y una tabla denominada "socios" en la que guarda los datos personales de cada socio.

1. Cree las tablas:
 - inscritos
 - **numero**, *serial*.
 - **documento**, *char(8)*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **primary key**, *numero*.
 - socios
 - **numerosocio**, *int*. not null
 - **deporte**, *varchar(20)*. not null
 - **cuotas**, *smallint*.
 - **primary key**, *(*numerosocio*., *deporte*)
2. Ingrese algunos registros:
3. Emplee una subconsulta con el operador "exists" para devolver la lista de socios que se inscribieron en 'natacion' (3 registros).
4. Busque los socios que NO se han inscripto en 'natacion' empleando "not exists" (1 registro).
5. Muestre todos los datos de los socios que han pagado todas las cuotas (1 registro).

Ejercicio 82

Un club dicta clases de distintos deportes a sus socios. El club tiene una tabla llamada "deportes" en la cual almacena el nombre del deporte, el nombre del profesor que lo dicta, el día de la semana que se dicta y el costo de la cuota mensual.

1. Cree una tabla con los siguientes campos:
 - **nombre**, *varchar(15)*.
 - **profesor**, *varchar(30)*.
 - **día**, *varchar(10)*.
 - **cuota**, *decimal(5, 2)*
2. Ingrese algunos registros. Incluya profesores que dicten más de un curso:
3. Muestre los nombres de los profesores que dictan más de un deporte empleando subconsulta.
4. Obtenga el mismo resultado empleando join.
5. Buscamos todos los deportes que se dictan el mismo día que un determinado deporte (*natacion*) empleando subconsulta.
6. Obtenga la misma salida empleando "join".

Ejercicio 83

Un club dicta clases de distintos deportes. En una tabla llamada "socios" guarda los datos de los socios, en una tabla llamada "deportes" la información referente a los diferentes deportes que se dictan y en una tabla denominada "inscritos", las inscripciones de los socios a los distintos deportes.

Un socio puede inscribirse en varios deportes el mismo año. Un socio no puede inscribirse en el mismo deporte el mismo año. Distintos socios se inscriben en un mismo deporte en el mismo año.

1. Cree las tablas con las siguientes estructuras:
 - socios
 - **documento**, *char(8)*. not null,
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **primary key**, *documento*.
 - deportes
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **profesor**, *varchar(15)*.
 - **primary key**, *codigo*.
 - inscritos
 - **documento**, *char(8)*. not null,
 - **codigodeporte**, *smallint*. not null
 - **año** *char(4)*
 - **matricula**, *char(1)*., --'s'=paga, 'n'=impaga.
 - ***primary key**, (*documento*., *codigodeporte*, *año*).
2. Ingrese algunos registros en las 3 tablas:
3. Realice una consulta en la cual muestre todos los datos de las inscripciones, incluyendo el nombre del deporte y del profesor. Esta consulta es un join.
4. Utilice el resultado de la consulta anterior como una tabla derivada para emplear en lugar de una tabla para realizar un "join" y recuperar el nombre del socio, el deporte en el cual está inscripto, el año, el nombre del profesor y la matrícula.

Ejercicio 84

Un club dicta clases de distintos deportes a sus socios. El club tiene una tabla llamada "inscritos" en la cual almacena el número de "socio", el código del deporte en el cual se inscribe y si la matrícula está o no paga, y una tabla denominada "socios" en la que guarda los datos personales de cada socio.

1. Cree las tablas:
 - socios
 - **numero**, *serial*.
 - **documento**, *char(8)*.
 - **nombre**, *varchar(30)*.

- **domicilio**, *varchar(30)*.
- **primary key**, *numero*.
- inscritos
 - **numerosocio**, *int*. not null
 - **deporte**, *varchar(20)*. not null
 - **matricula**, *char(1)*., -- 'n' o 's'
 - **primary key**, *(numerosocio., deporte)
- 2. Ingrese algunos registros:
- 3. Actualizamos la cuota ('s') de todas las inscripciones de un socio determinado (*por documento*) empleando subconsulta.
- 4. Elimine todas las inscripciones de los socios que deben alguna matrícula (*5 registros eliminados*).

Ejercicio 85

Un comercio que vende artículos de librería y papelería almacena la información de sus ventas en una tabla llamada "facturas" y otra "clientes".

1. Cree las tablas:
 - facturas
 - **codigo**, *serial*.
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
 - **primary key**, *codigo*.
 - clientes
 - **numero**, *int*. not null
 - **fecha**, *date*.
 - **codigocliente**, *int*. not null
 - **total**, *decimal(6., 2)*
 - **primary key**, *numero*.
2. Ingrese algunos registros:
3. El comercio necesita una tabla llamada "clientespref" en la cual quiere almacenar el nombre y domicilio de aquellos clientes que han comprado hasta el momento más de 500 pesos en mercaderías. Créela la tabla:
 - **nombre**, *varchar(30)*.
 - **domicilio**, *varchar(30)*.
4. Ingrese los registros en la tabla "clientespref" seleccionando registros de la tabla "clientes" y "facturas".
5. Vea los registros de "clientespref":

Ejercicio 86

Un club dicta cursos de distintos deportes. Almacena la información en varias tablas. El director no quiere que los empleados de administración conozcan la estructura de las tablas ni algunos datos de los profesores y socios, por ello se crean vistas a las cuales tendrán acceso.

1. Crear las tablas:
 - Socios
 - **documento**, *char(8)*. not null
 - **nombre**, *varchar(40)*.
 - **domicilio**, *varchar(30)*.
 - **primary key**, *documento*.
 - Profesores
 - **documento**, *char(8)*. not null
 - **nombre**, *varchar(40)*.
 - **domicilio**, *varchar(30)*.
 - **primary key**, *documento*.
 - Cursos
 - **numero**, *serial*.
 - **deporte**, *varchar(20)*.
 - **dia**, *varchar(15)*.
 - **documentoprofesor**, *char(8)*.
 - **primary key**, *numero*.
 - Inscritos

- **documentosocio**, *char(8)*. not null
 - **numero**, *smallint*. not null
 - **matricula**, *char(1)*.
 - **primary key**, *documentosocio, numero*.
2. Ingrese algunos registros para todas las tablas:
 3. Cree una vista en la que aparezca el nombre y documento del socio, el deporte, el día y el nombre del profesor.
 4. Muestre la información contenida en la vista.
 5. Realice una consulta a la vista donde muestre la cantidad de socios inscritos en cada deporte ordenados por cantidad.
 6. Muestre (*consultando la vista*) los cursos (*deporte y día*) para los cuales no hay inscritos.
 7. Muestre los nombres de los socios que no se han inscripto en ningún curso (*consultando la vista*)
 8. Muestre (*consultando la vista*) los profesores que no tienen asignado ningún deporte aún.
 9. Muestre (*consultando la vista*) el nombre y documento de los socios que deben matrículas.
 10. Consulte la vista y muestre los nombres de los profesores y los días en que asisten al club para dictar sus clases.
 11. Muestre la misma información anterior pero ordenada por día.
 12. Muestre todos los socios que son compañeros en tenis los lunes.

Ejercicio 87

Una empresa registra los datos de sus empleados en una tabla llamada "empleados".

1. Cree la secuencia "sec_legajoempleados" estableciendo el valor mínimo (1), máximo (999), valor inicial (100), valor de incremento (2) y no circular.
2. Cree la tabla:
 - **legajo**, *bigint*. default nextval('sec_legajoempleados')
 - **documento**, *char(8)*. not null
 - **nombre**, *varchar(30)*. not null
 - **primary key**, *legajo*.
3. Ingrese algunos registros:
 - '22333444', 'Ana Acosta'.
 - '23444555', 'Betina Bustamante'.
 - '24555666', 'Carlos Caseros'.
 - '25666777', 'Diana Domínguez'.
 - '26777888', 'Estela Esper'.
4. Recupere los registros de la tabla empleados.
5. Efectue un #####
6. Elimine la secuencia y la tabla asociada a dicha secuencia.

Ejercicio 88

Una empresa organiza un curso de computación (*dispone de dos aulas*), almacenar en una tabla inscritos los datos del estudiante.

Cada vez que se inscribe un alumno asignarlo a un aula en forma alternada (*primero a la 1 y luego a la 2, luego nuevamente a la 1 y así sucesivamente*).

1. Crear una secuencia sec_codigoaulainscriptos (*valor inicial 1, incremento 1, valor máximo 2 y debe ser circular*).
2. Crear la tabla inscritos:
 - **documento**, *char(8)* not null.
 - **nombre**, *varchar(30)* not null.
 - **codigocurso**, *int* default nextval('sec_codigoaulainscriptos').
 - **primary key**, *documento*.
3. Insertar algunos registros:
 - '20000000', 'Rodriguez Pablo'.
 - '30000000', 'Mercado Ana'.
 - '40000000', 'Morello Luis'.
 - '50000000', 'Prado Juan'.
 - '00000', 'Solis Maria'.
4. Imprimir todos los alumnos del curso 1.
5. Imprimir todos los alumnos del curso 2.

Ejercicio 89

Trabaje con la tabla llamada "medicamentos" de una farmacia.

1. Cree la tabla con la siguiente estructura:
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **laboratorio**, *varchar(20)*.
 - **precio**, *decimal(5., 2)*
 - **cantidad**, *smallint*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros:
 - 'Sertal', 'Roche', 5.2, 100.
 - 'Buscapina', 'Roche', 4.10, 200.
 - 'Amoxidal 500', 'Bayer', 15.60, 100.
 - 'Paracetamol 500', 'Bago', 1.90, 200.
 - *Bayaspirina', 'Bayer', 2.10, 150); *.
 - *Amoxidal jarabe', 'Bayer', 5.10, 250); *.
3. Implementar una función que retorne el precio promedio de la tabla medicamentos.
4. Imprimir el precio promedio de los medicamentos.
5. Imprimir los medicamentos que tienen un precio mayor al promedio.

Ejercicio 90

Trabaje con la tabla llamada "medicamentos" de una farmacia.

1. Cree la tabla con la siguiente estructura:
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **laboratorio**, *varchar(20)*.
 - **precio**, *decimal(5., 2)*
 - **cantidad**, *smallint*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros:
 - 'Sertal', 'Roche', 5.2, 100.
 - 'Buscapina', 'Roche', 4.10, 200.
 - 'Amoxidal 500', 'Bayer', 15.60, 100.
 - 'Paracetamol 500', 'Bago', 1.90, 200.
 - *Bayaspirina', 'Bayer', 2.10, 150); *.
 - *Amoxidal jarabe', 'Bayer', 5.10, 250); *.
3. Implementar una función que reciba el código de un medicamento y proceda a borrarlo. La función no retorna dato.
4. Proceder a llamar a la función.
5. Imprimir la tabla medicamentos.

Ejercicio 91

Trabaje con la tabla llamada "medicamentos" de una farmacia.

1. Cree la tabla con la siguiente estructura:
 - **codigo**, *serial*.
 - **nombre**, *varchar(20)*.
 - **laboratorio**, *varchar(20)*.
 - **precio**, *decimal(5., 2)*
 - **cantidad**, *smallint*.
 - **primary key**, *codigo*.
2. Ingrese algunos registros:
 - 'Sertal', 'Roche', 5.2, 100.
 - 'Buscapina', 'Roche', 4.10, 200.

- 'Amoxidal 500', 'Bayer', 15.60, 100.
- 'Paracetamol 500', 'Bago', 1.90, 200.
- *'Bayaspirina', 'Bayer', 2.10, 150); *.
- *'Amoxidal jarabe', 'Bayer', 5.10, 250); *.

3. Implementar una función que retorne el registro completo del medicamento más caro.

4. Proceder a llamar a la función.

```
/** EJERCICIO 1
***** /

CREATE TABLE "/agenda" (
    apellido VARCHAR(30),
    nombre VARCHAR(20),
    domicilio VARCHAR(30),
    telefono VARCHAR(11)
);

DROP TABLE IF EXISTS agenda;

CREATE TABLE agenda(
    apellido VARCHAR(30),
    nombre VARCHAR(20),
    domicilio VARCHAR(30),
    telefono VARCHAR(11)
); -- Should fail

SELECT table_name,column_name,udt_name,character_maximum_length
FROM information_schema.columns
WHERE table_name = 'agenda';

DROP TABLE agenda;

DROP TABLE agenda; -- Should fail

/** EJERCICIO 2
***** /

DROP TABLE IF EXISTS libros;

CREATE TABLE libros(
    titulo VARCHAR(20),
    autor VARCHAR(30),
    editorial VARCHAR(15)
);

CREATE TABLE libros(
    titulo VARCHAR(20),
    autor VARCHAR(30),
    editorial VARCHAR(15)
); -- Should fail

SELECT table_name,column_name,udt_name,character_maximum_length
FROM information_schema.columns
WHERE table_name = 'libros';

DROP TABLE libros;

DROP TABLE libros; -- Should fail

/** EJERCICIO 3
***** /
```

```
DROP TABLE IF EXISTS agenda;
```

```
CREATE TABLE agenda (  
    apellido VARCHAR(30),  
    nombre VARCHAR(20),  
    domicilio VARCHAR(30),  
    telefono VARCHAR(11)  
);  
  
SELECT table_name,column_name,udt_name,character_maximum_length  
FROM information_schema.columns  
WHERE table_name = 'agenda';
```

```
INSERT INTO agenda (apellido, nombre, domicilio, telefono)  
VALUES ('Moreno','Alberto','Colon 123','4234567'),  
('Torres','Juan','Avellaneda 135','4458787');
```

```
SELECT * FROM agenda;
```

```
DROP TABLE agenda;
```

```
DROP TABLE agenda; -- Should fail
```

```
/** EJERCICIO 4
```

```
*****/
```

```
DROP TABLE IF EXISTS libros;
```

```
CREATE TABLE libros (  
    titulo VARCHAR(20),  
    autor VARCHAR(30),  
    editorial VARCHAR(15)  
);  
  
SELECT table_name,column_name,udt_name,character_maximum_length  
FROM information_schema.columns  
WHERE table_name = 'agenda';  
  
INSERT INTO libros (titulo,autor,editorial)  
VALUES ('El aleph','Borges','Planeta'),  
('Martin Fierro','Jose Hernandez','Emece'),  
('Aprenda PHP','Mario Molina','Emece');
```

```
SELECT * FROM libros;
```

```
/** EJERCICIO 5
```

```
*****/
```

```
DROP TABLE IF EXISTS peliculas;
```

```
CREATE TABLE peliculas (  
    nombre VARCHAR(20),  
    actor VARCHAR(20),  
    duracion INTEGER,  
    cantidad INTEGER  
);
```

```
SELECT table_name,column_name,udt_name,character_maximum_length
FROM information_schema.columns
WHERE table_name = 'peliculas';
```

```
INSERT INTO peliculas (nombre, actor, duracion, cantidad)
VALUES ('Mision imposible','Tom Cruise',128,3),
('Mision imposible 2','Tom Cruise',130,2),
('Mujer bonita','Julia Roberts',118,3),
('Elsa y Fred','China Zorrilla',110,2);
```

```
SELECT *FROM peliculas;
```

```
/** EJERCICIO 6
```

```
*****/
```

```
DROP TABLE IF EXISTS empleados;
```

```
CREATE TABLE empleados (
nombre VARCHAR(20),
documento VARCHAR(8),
sexo VARCHAR(1),
domicilio VARCHAR(30),
sueldobasico FLOAT
);
```

```
SELECT table_name,column_name,udt_name,character_maximum_length
FROM information_schema.columns
WHERE table_name = 'empleados';
```

```
INSERT INTO empleados (nombre, documento, sexo, domicilio, sueldobasico)
VALUES ('Juan Perez','22333444','m','Sarmiento 123',500),
('Ana Acosta','24555666','f','Colon 134',650),
('Bartolome Barrios','27888999','m','Urquiza 479',800);
```

```
SELECT *FROM empleados;
```

```
/** EJERCICIO 7
```

```
*****/
```

```
DROP TABLE IF EXISTS peliculas;
```

```
CREATE TABLE peliculas (
titulo VARCHAR(20),
actor VARCHAR(20),
duracion INTEGER,
cantidad INTEGER
);
```

```
SELECT table_name,column_name,udt_name,character_maximum_length
FROM information_schema.columns
WHERE table_name = 'peliculas';
```

```
INSERT INTO peliculas (titulo, actor, duracion, cantidad)
VALUES ('Mision imposible','Tom Cruise',180,3),
('Mision imposible 2','Tom Cruise',190,2),
('Mujer bonita','Julia Roberts',118,3),
```



```
( 'Elsa y Fred', 'China Zorrilla', 110, 2);
```

```
SELECT titulo,actor FROM peliculas;
SELECT titulo,duracion FROM peliculas;
SELECT titulo,cantidad FROM peliculas;
```

```
/** EJERCICIO 8
```

```
*****/
```

```
DROP TABLE IF EXISTS empleados;
```

```
CREATE TABLE empleados (
    nombre VARCHAR(20),
    documento VARCHAR(8),
    sexo VARCHAR(1),
    domicilio VARCHAR(30),
    sueldobasico FLOAT
);
SELECT table_name,column_name,udt_name,character_maximum_length
FROM information_schema.columns
WHERE table_name = 'empleados';
```

```
INSERT INTO empleados (nombre, documento, sexo, domicilio, sueldobasico)
VALUES ( 'Juan Juarez', '22333444', 'm', 'Sarmiento 123', 500),
( 'Ana Acosta', '27888999', 'f', 'Colon 134', 700),
( 'Carlos Caseres', '31222333', 'm', 'Urquiza 479', 850);
```

```
SELECT * FROM empleados;
SELECT nombre,documento,domicilio FROM empleados;
SELECT documento,sexo,sueldobasico FROM empleados;
```

```
/** EJERCICIO 9
```

```
*****/
```

```
DROP TABLE IF EXISTS agenda;
```

```
CREATE TABLE agenda (
    apellido VARCHAR(30),
    nombre VARCHAR(20),
    domicilio VARCHAR(30),
    telefono VARCHAR(11)
);
SELECT table_name,column_name,udt_name,character_maximum_length
FROM information_schema.columns
WHERE table_name = 'agenda';
INSERT INTO agenda(apellido,nombre,domicilio,telefono)
VALUES ( 'Acosta', 'Ana', 'Colon 123', '4234567'),
( 'Bustamante', 'Betina', 'Avellaneda 135', '4458787'),
( 'Lopez', 'Hector', 'Salta 545', '4887788'),
( 'Lopez', 'Luis', 'Urquiza 333', '4545454'),
( 'Lopez', 'Marisa', 'Urquiza 333', '4545454');
```

```
SELECT * FROM agenda;
```

```
SELECT * FROM agenda
```

```
WHERE nombre='Marisa';

SELECT nombre,domicilio FROM agenda
WHERE apellido='Lopez';

SELECT nombre FROM agenda
WHERE telefono='4545454';

/** EJERCICIO 10
***** /

DROP TABLE IF EXISTS libros;

CREATE TABLE libros (
    titulo VARCHAR(20),
    autor VARCHAR(30),
    editorial VARCHAR(15)
);

SELECT table_name,column_name,udt_name,character_maximum_length
FROM information_schema.columns
WHERE table_name = 'libros';

INSERT INTO libros(titulo,autor,editorial)
VALUES ('El aleph','Borges','Emece'),
('Martin Fierro','Jose Hernandez','Emece'),
('Martin Fierro','Jose Hernandez','Planeta'),
('Aprenda PHP','Mario Molina','Siglo XXI');

SELECT * FROM libros
WHERE autor='Borges';

SELECT titulo FROM libros
WHERE editorial='Emece';

SELECT editorial FROM libros
WHERE titulo='Martin Fierro';

/** EJERCICIO 11
***** /

DROP TABLE IF EXISTS articulos;

CREATE TABLE articulos (
    codigo INTEGER,
    nombre VARCHAR(20),
    descripcion VARCHAR(30),
    precio FLOAT,
    cantidad INTEGER
);

INSERT INTO articulos (codigo, nombre, descripcion, precio,cantidad)
VALUES (1,'impresora','Epson Stylus C45',400.80,20),
(2,'impresora','Epson Stylus C85',500,30),
(3,'monitor','Samsung 14',800,10),
(4,'teclado','ingles Biswal',100,50),
```

```
(5,'teclado','español Biswal',90,50);
```

```
SELECT * FROM articulos
WHERE nombre='impresora';
```

```
SELECT * FROM articulos
WHERE precio <>400;
```

```
SELECT codigo,nombre
FROM articulos
WHERE cantidad < 30;
```

```
SELECT nombre, descripcion
FROM articulos
WHERE precio <> 100;
```

```
/** EJERCICIO 12
```

```
*****/
```

```
DROP TABLE IF EXISTS peliculas;
```

```
CREATE TABLE peliculas (
    titulo VARCHAR(20),
    actor VARCHAR(20),
    duracion INTEGER,
    cantidad INTEGER
);
```

```
INSERT INTO peliculas (titulo, actor, duracion, cantidad)
VALUES ('Mision imposible','Tom Cruise',120,3),
('Mision imposible 2','Tom Cruise',180,4),
('Mujer bonita','Julia R.',90,1),
('Elsa y Fred','China Zorrilla',80,2);
```

```
SELECT * FROM peliculas
WHERE duracion <>90;
```

```
SELECT * FROM peliculas
WHERE actor <> 'Tom Cruise';
```

```
SELECT titulo,actor,cantidad
FROM peliculas
WHERE cantidad > 2;
```

```
/** EJERCICIO 13
```

```
*****/
```

```
DROP TABLE IF EXISTS agenda;
```

```
CREATE TABLE agenda (
    apellido VARCHAR(30),
    nombre VARCHAR(20),
    domicilio VARCHAR(30),
    telefono VARCHAR(11)
);
```

```
INSERT INTO agenda (apellido,nombre,domicilio,telefono)
VALUES  ('Alvarez','Alberto','Colon 123','4234567'),
        ('Juarez','Juan','Avellaneda 135','4458787'),
        ('Lopez','Maria','Urquiza 333','4545454'),
        ('Lopez','Jose','Urquiza 333','4545454'),
        ('Salas','Susana','Gral. Paz 1234','4123456');
```

```
DELETE FROM agenda
WHERE nombre='Juan';
```

```
DELETE FROM agenda
WHERE telefono='4545454';
```

```
SELECT * FROM agenda;
```

```
DELETE FROM agenda;
```

```
SELECT * FROM agenda;
```

```
/** EJERCICIO 14
```

```
*****/
```

```
DROP TABLE IF EXISTS articulos;
```

```
CREATE TABLE articulos (
    codigo INTEGER,
    nombre VARCHAR(20),
    descripcion VARCHAR(30),
    precio FLOAT,
    cantidad INTEGER
);
```

```
INSERT INTO articulos (codigo, nombre, descripcion, precio,cantidad)
VALUES  (1,'impresora','Epson Stylus C45',400.80,20),
        (2,'impresora','Epson Stylus C85',500,30),
        (3,'monitor','Samsung 14',800,10),
        (4,'teclado','ingles Biswal',100,50),
        (5,'teclado','espaniol Biswal',90,50);
```

```
DELETE FROM articulos
WHERE precio <>500;
```

```
SELECT * FROM articulos;
```

```
DELETE FROM articulos
WHERE nombre='impresora';
```

```
SELECT * FROM articulos;
```

```
DELETE FROM articulos
WHERE codigo <> 4;
```

```
SELECT * FROM articulos;
```

```
/** EJERCICIO 15
```

*****/

```
DROP TABLE IF EXISTS agenda;
```

```
CREATE TABLE agenda (  
    apellido VARCHAR(30),  
    nombre VARCHAR(20),  
    domicilio VARCHAR(30),  
    telefono VARCHAR(11)  
);  
  
INSERT INTO agenda (apellido,nombre,domicilio,telefono)  
VALUES ('Acosta','Alberto','Colon 123','4234567'),  
('Juarez','Juan','Avellaneda 135','4458787'),  
('Lopez','Maria','Urquiza 333','4545454'),  
('Lopez','Jose','Urquiza 333','4545454'),  
('Suarez','Susana','Gral. Paz 1234','4123456');
```

```
SELECT * FROM agenda;
```

```
UPDATE agenda SET nombre='Juan Jose'  
WHERE nombre='Juan';  
SELECT * FROM agenda;
```

```
UPDATE agenda SET telefono='4445566'  
WHERE telefono='4545454';  
SELECT * FROM agenda;
```

```
UPDATE agenda SET nombre='Juan Jose'  
WHERE nombre='Juan';  
SELECT * FROM agenda;
```

```
/** EJERCICIO 16
```

*****/

```
DROP TABLE IF EXISTS libros;
```

```
CREATE TABLE libros (  
    titulo VARCHAR(30),  
    autor VARCHAR(20),  
    editorial VARCHAR(15),  
    precio FLOAT  
);  
  
INSERT INTO libros (titulo, autor, editorial, precio)  
VALUES ('El aleph','Borges','Emece',25.00),  
('Martin Fierro','Jose Hernandez','Planeta',35.50),  
('Aprenda PHP','Mario Molina','Emece',45.50),  
('Cervantes y el quijote','Borges','Emece',25),  
('Matematica estas ahi','Paenza','Siglo XXI',15);
```

```
SELECT * FROM libros;
```

```
UPDATE libros SET autor='Adrian Paenza'  
WHERE autor='Paenza';  
SELECT * FROM libros;
```

```
UPDATE libros SET autor='Adrian Paenza'
```

```
WHERE autor='Paenza';
SELECT * FROM libros;

UPDATE libros SET precio=27
WHERE autor='Mario Molina';
SELECT * FROM libros;

UPDATE libros SET editorial='Emece S.A.'
WHERE editorial='Emece';
SELECT * FROM libros;

/** EJERCICIO 17
*****/

DROP TABLE IF EXISTS medicamentos;

CREATE TABLE medicamentos (
    codigo INTEGER NOT NULL,
    nombre VARCHAR(20) NOT NULL,
    laboratorio VARCHAR(20),
    precio FLOAT,
    cantidad INTEGER NOT NULL
);
SELECT table_name,column_name,udt_name,character_maximum_length,is_nullable
FROM information_schema.columns
WHERE table_name = 'medicamentos';

INSERT INTO medicamentos (codigo,nombre,laboratorio,precio,cantidad)
VALUES (1,'Sertal gotas',NULL,NULL,100),
(2,'Sertal compuesto',NULL,8.90,150),
(3,'Buscapina','Roche',NULL,200);

SELECT * FROM medicamentos;

INSERT INTO medicamentos (codigo,nombre, laboratorio,precio,cantidad)
VALUES (4,'Bayaspirina','',0,150),
(0,','', 'Bayer',15.60,0);

SELECT * FROM medicamentos;

INSERT INTO medicamentos (codigo,nombre,laboratorio,precio,cantidad)
VALUES (NULL,'Amoxidal jarabe','Bayer',25,120); -- Should fail

SELECT * FROM medicamentos
WHERE laboratorio is NULL;

SELECT * FROM medicamentos
WHERE laboratorio='';

SELECT * FROM medicamentos
WHERE precio is NULL;

SELECT * FROM medicamentos
WHERE precio=0;

SELECT * FROM medicamentos
```

```
WHERE laboratorio <> '';

SELECT * FROM medicamentos
WHERE laboratorio is NOT NULL;

SELECT * FROM medicamentos
WHERE precio <> 0;

SELECT * FROM medicamentos
WHERE precio is NOT NULL;

/** EJERCICIO 18
*****/

DROP TABLE IF EXISTS peliculas;

CREATE TABLE peliculas (
    codigo INT NOT NULL,
    titulo VARCHAR(40) NOT NULL,
    actor VARCHAR(20),
    duracion INT
);

SELECT table_name,column_name,udt_name,character_maximum_length,is_nullable
FROM information_schema.columns
WHERE table_name = 'peliculas';

INSERT INTO peliculas (codigo,titulo,actor,duracion)
VALUES (1,'Mision imposible','Tom Cruise',120),
(2,'Harry Potter y la piedra fi losofal',NULL,180),
(3,'Harry Potter y la camara secreta','Daniel R.',NULL),
(4,'Mision imposible 2','',150),
(5,'Mujer bonita','R. Gere-J. Roberts',0);

SELECT * FROM peliculas;

INSERT INTO peliculas (codigo,titulo,actor,duracion)
VALUES (NULL,'Mujer bonita','R. Gere-J. Roberts',190);

SELECT * FROM peliculas
WHERE actor is NULL;

UPDATE peliculas SET duracion=120
WHERE duracion is NULL;

UPDATE peliculas SET actor='DESConocido'
WHERE actor='';

SELECT * FROM peliculas;

DELETE FROM peliculas
WHERE titulo='';

SELECT * FROM peliculas;

/** EJERCICIO 19
*****/
```

```
DROP TABLE IF EXISTS libros;

CREATE TABLE libros (
    codigo INT NOT NULL,
    titulo VARCHAR(40) NOT NULL,
    autor VARCHAR(20),
    editorial VARCHAR(15),
    PRIMARY KEY(codigo)
);

INSERT INTO libros (codigo,titulo,autor,editorial)
VALUES (1,'El aleph','Borges','Emece'),
(2,'Martin Fierro','Jose Hernandez','Planeta'),
(3,'Aprenda PHP','Mario Molina','Nuevo Siglo');

INSERT INTO libros (codigo,titulo,autor,editorial)
VALUES (2,'Alicia en el pais de las maravillas','Lewis Carroll','Planeta');

INSERT INTO libros (codigo,titulo,autor,editorial)
VALUES (NULL,'Alicia en el pais de las maravillas','Lewis Carroll','Planeta');

UPDATE libros SET codigo=1
WHERE titulo='Martin Fierro';

/** EJERCICIO 20
*****/

DROP TABLE IF EXISTS alumnos;

CREATE TABLE alumnos (
    legajo VARCHAR(4) NOT NULL,
    documento VARCHAR(8),
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
    PRIMARY KEY(documento),
    PRIMARY KEY(legajo)
);

DROP TABLE IF EXISTS alumnos;

CREATE TABLE alumnos (
    legajo VARCHAR(4) NOT NULL,
    documento VARCHAR(8),
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
    PRIMARY KEY(documento)
);

SELECT table_name,column_name,udt_name,character_maximum_length,is_nullable
FROM information_schema.columns
WHERE table_name = 'alumnos';

INSERT INTO alumnos (legajo,documento,nombre,domicilio)
VALUES ('A233','22345345','Perez Mariana','Colon 234'),
('A567','23545345','Morales Marcos','Avellaneda 348');
```



```
INSERT INTO alumnos (legajo,documento,nombre,domicilio)
VALUES ('A642','23545345','Gonzalez Analia','Caseros 444');

INSERT INTO alumnos (legajo,documento,nombre,domicilio)
VALUES ('A685',NULL,'Miranda Carmen','Uspallata 999');

/** EJERCICIO 21
*****/

DROP TABLE IF EXISTS medicamentos;

CREATE TABLE medicamentos (
    codigo SERIAL,
    nombre VARCHAR(20),
    laboratorio VARCHAR(20),
    precio FLOAT,
    cantidad INTEGER,
    PRIMARY KEY (codigo)
);

SELECT table_name,column_name,udt_name,character_maximum_length,is_nullable
FROM information_schema.columns
WHERE table_name = 'medicamentos';

INSERT INTO medicamentos (nombre, laboratorio,precio,cantidad)
VALUES ('Sertal','Roche',5.2,100),
('Buscapina','Roche',4.10,200),
('Amoxidal 500','Bayer',15.60,100);

SELECT * FROM medicamentos;

/** EJERCICIO 22
*****/

DROP TABLE IF EXISTS peliculas;

CREATE TABLE peliculas (
    codigo SERIAL,
    titulo VARCHAR(40),
    actor VARCHAR(20),
    duracion INT,
    PRIMARY KEY (codigo)
);

SELECT table_name,column_name,udt_name,character_maximum_length,is_nullable
FROM information_schema.columns
WHERE table_name = 'peliculas';

INSERT INTO peliculas (titulo,actor,duracion)
VALUES ('Mision imposible','Tom Cruise',120),
('Harry Potter y la piedra filosofal','xxx',180),
('Harry Potter y la camara secreta','xxx',190),
('Mision imposible 2','Tom Cruise',120),
('La vida es bella','zzz',220);

SELECT * FROM peliculas;

UPDATE peliculas SET actor='Daniel R.' WHERE codigo=3;
```

```
DELETE FROM peliculas WHERE titulo='La vida es bella';
```

```
DELETE FROM peliculas WHERE duracion=120;
```

```
SELECT * FROM peliculas;
```

```
INSERT INTO peliculas (titulo,actor,duracion)
VALUES ('Mujer bonita','Richard Gere',120);
```

```
SELECT * FROM peliculas;
```

```
/** EJERCICIO 23
```

```
*****/
```

```
DROP TABLE IF EXISTS medicamentos;
```

```
CREATE TABLE medicamentos (
    codigo SERIAL,
    nombre VARCHAR(20),
    laboratorio VARCHAR(20),
    precio FLOAT,
    cantidad INTEGER,
    PRIMARY KEY (codigo)
);
```

```
INSERT INTO medicamentos (nombre, laboratorio,precio,cantidad)
VALUES ('Sertal','Roche',5.2,100),
('Buscapina','Roche',4.10,200),
('Amoxidal 500','Bayer',15.60,100);
```

```
DELETE FROM medicamentos;
```

```
INSERT INTO medicamentos (nombre, laboratorio,precio,cantidad)
VALUES ('Sertal','Roche',5.2,100),
('Amoxidal 500','Bayer',15.60,100);
```

```
SELECT * FROM medicamentos;
```

```
TRUNCATE TABLE medicamentos RESTART IDENTITY;
```

```
INSERT INTO medicamentos (nombre, laboratorio,precio,cantidad)
VALUES ('Buscapina','Roche',4.10,200);
```

```
SELECT * FROM medicamentos;
```

```
/** EJERCICIO 24
```

```
*****/
```

```
DROP TABLE IF EXISTS autos;
```

```
CREATE TABLE autos (
    patente CHAR(6),
    marca VARCHAR(20),
    modelo CHAR(4),
    precio FLOAT,
```

```
PRIMARY KEY (patente)
);

INSERT INTO autos
VALUES ('ACD123','Fiat 128','1970',15000),
('ACG234','Renault 11','1990',40000),
('BCD333','Peugeot 505','1990',80000),
('GCD123','Renault Clio','1990',70000),
('BCC333','Renault Megane','1998',95000),
('BVF543','Fiat 128','1975',20000);

SELECT * FROM autos
WHERE modelo='1990';

DROP TABLE autos;

DROP TABLE IF EXISTS autos;

CREATE TABLE autos (
patente CHARACTER(6),
marca CHARACTER varying(20),
modelo CHARACTER(4),
precio FLOAT,
PRIMARY KEY (patente)
);

INSERT INTO autos
VALUES ('ACD123','Fiat 128','1970',15000);

SELECT * FROM autos;

/** EJERCICIO 25
*****/

DROP TABLE IF EXISTS clientes;

CREATE TABLE clientes (
documento CHAR(8),
apellido VARCHAR(20),
nombre VARCHAR(20),
domicilio VARCHAR(30),
telefono VARCHAR (11)
);

INSERT INTO clientes
VALUES ('2233344','Perez','Juan','Sarmiento 980','4342345'),
('2333344','Perez','Ana','Colon 234'),
('2433344','Garcia','Luis','Avellaneda 1454','4558877'),
('2533344','Juarez','Ana','Urquiza 444','4789900');

SELECT * FROM clientes
WHERE apellido='Perez';

/** EJERCICIO 26
*****/
```

```
DROP TABLE IF EXISTS cuentas;
```

```
CREATE TABLE cuentas (  
    numero INT NOT NULL,  
    documento CHAR(8) NOT NULL,  
    nombre VARCHAR(30),  
    saldo DECIMAL(15,2),  
    PRIMARY KEY (numero)  
);
```

```
INSERT INTO cuentas(numero,documento,nombre,saldo)  
VALUES ('1234','25666777','Pedro Perez',500000.60),  
('2234','27888999','Juan Lopez',-250000),  
('3344','27888999','Juan Lopez',4000.50),  
('3346','32111222','Susana Molina',1000);
```

```
SELECT * FROM cuentas  
WHERE saldo < 4000;
```

```
SELECT numero,saldo FROM cuentas  
WHERE nombre='Juan Lopez';
```

```
SELECT * FROM cuentas  
WHERE saldo < 0;
```

```
SELECT * FROM cuentas  
WHERE numero <>3000;
```

```
/** EJERCICIO 27
```

```
*****/
```

```
DROP TABLE IF EXISTS empleados;
```

```
CREATE TABLE empleados (  
    nombre VARCHAR(30),  
    documento CHAR(8),  
    sexo CHAR(1),  
    domicilio VARCHAR(30),  
    sueldobasico DECIMAL(7,2),  
    cantidadhijos SMALLINT  
);
```

```
INSERT INTO empleados (nombre,documento,sexo,domicilio,sueldobasico,cantidadhijos)  
VALUES ('Juan Perez','22333444','m','Sarmiento 123',500,2),  
('Ana Acosta','24555666','f','Colon 134',850,0),  
('Bartolome Barrios','27888999','m','Urquiza 479',10000.80,4),  
('Susana Molina','29000555','f','Salta 876',800.888,3),  
('Marta Juarez','32444555','f','Sucre 1086',5000000,2);
```

```
SELECT * FROM empleados  
WHERE sueldobasico <>900;
```

```
SELECT * FROM empleados  
WHERE cantidadhijos > 0;
```

```

/** EJERCICIO 28
*****/

DROP TABLE IF EXISTS alumnos;

CREATE TABLE alumnos (
    apellido VARCHAR(30),
    nombre VARCHAR(30),
    documento CHAR(8),
    domicilio VARCHAR(30),
    fechaingreso DATE,
    fechanacimiento DATE
);
SET DATESTYLE to 'European';

INSERT INTO alumnos
VALUES ('Gonzalez','Ana','22222222','Colon 123','10-08-1990','15/02/1972'),
('Juarez','Bernardo','25555555','Sucre 456','03-03-1991','15/02/1972'),
('Perez','Laura','26666666','Bulnes 345','03-03-91',NULL),
('Lopez','Carlos','27777777','Sarmiento 1254','03-15-1990',NULL);

SELECT * FROM alumnos WHERE fechaingreso < '1-1-91';
SELECT * FROM alumnos WHERE fechanacimiento is NULL;

/** EJERCICIO 29
*****/

DROP TABLE IF EXISTS visitantes;

CREATE TABLE visitantes (
    nombre VARCHAR(30),
    edad SMALLINT,
    sexo CHAR(1) DEFAULT 'f',
    domicilio VARCHAR(30),
    ciudad VARCHAR(20) DEFAULT 'Cordoba',
    telefono VARCHAR(11),
    mail VARCHAR(30) DEFAULT 'no tiene',
    montocompra DECIMAL (6,2)
);

SELECT *
FROM information_schema.columns
WHERE table_name = 'visitantes';

INSERT INTO visitantes (nombre, domicilio, montocompra)
VALUES ('Susana Molina','Colon 123',59.80),
('Marcos Torres',29,'Carlos Paz','marcostorres@hotmail.com');
SELECT * FROM visitantes;

INSERT INTO visitantes
VALUES ('Marcelo Morales',38,DEFAULT,DEFAULT,DEFAULT,'4255232',
'marcelomorales@hotmail.com',DEFAULT);

INSERT INTO visitantes DEFAULT VALUES;

```

```
SELECT * FROM visitantes;
```

```
/** EJERCICIO 30
```

```
*****/
```

```
DROP TABLE IF EXISTS prestamos;
```

```
CREATE TABLE prestamos (
    titulo VARCHAR(40) NOT NULL,
    documento CHAR(8) NOT NULL,
    fechaprestamo DATE NOT NULL,
    fechadevolucion DATE,
    devuelto CHAR(1) DEFAULT 'n'
);
```

```
INSERT INTO prestamos (titulo,documento,fechaprestamo,fechadevolucion)
VALUES ('Manual de 1 grado','23456789','2006-12-15','2006-12-18'),
('Alicia en el pais de las maravillas','23456789','2006-12-16'),
('El aleph','22543987','2006-12-16','2006-08-19'),
('Manual de geografia 5 grado','25555666','2006-12-18','s');
```

```
SELECT *FROM prestamos;
```

```
INSERT INTO prestamos
VALUES ('Manual de historia','32555666','2006-10-25',DEFAULT,DEFAULT);
```

```
SELECT *FROM prestamos;
```

```
INSERT INTO prestamos DEFAULT VALUES;
```

```
/** EJERCICIO 31
```

```
*****/
```

```
DROP TABLE IF EXISTS articulos;
```

```
CREATE TABLE articulos (
    codigo SERIAL,
    nombre VARCHAR(20),
    descripcion VARCHAR(30),
    precio DECIMAL(9,2),
    cantidad SMALLINT DEFAULT 0,
    PRIMARY KEY (codigo)
);
```

```
INSERT INTO articulos (nombre, descripcion, precio,cantidad)
VALUES ('impresora','Epson Stylus C45',400.80,20),
('impresora','Epson Stylus C85',500),
('monitor','Samsung 14',800),
('teclado','ingles Biswal',100,50);
```

```
UPDATE articulos SET precio=precio+(precio*0.15);
```

```
SELECT * FROM articulos;
```

```
SELECT nombre||','||descripcion
FROM articulos;
```

```
UPDATE articulos SET cantidad=cantidad-5
WHERE nombre='teclado';
SELECT * FROM articulos;

/** EJERCICIO 32
*****/

DROP TABLE IF EXISTS libros;

CREATE TABLE libros (
    codigo SERIAL,
    titulo VARCHAR(40) NOT NULL,
    autor VARCHAR(20) DEFAULT 'DESConocido',
    editorial VARCHAR(20),
    precio DECIMAL(6,2),
    cantidad SMALLINT DEFAULT 0,
    PRIMARY KEY (codigo)
);
INSERT INTO libros (titulo,autor,editorial,precio)
VALUES ('El aleph','Borges','Emece',25),
('Java en 10 MINutos','Mario Molina','Siglo XXI',50.40,100),
('Alicia en el pais de las maravillas','Lewis Carroll','Emece',15,50);

SELECT titulo, autor,editorial,precio,cantidad,precio*cantidad AS monto_total
FROM libros;

SELECT titulo,autor,precio,
precio*0.1 AS descuento,
precio-(precio*0.1) AS precio_final
FROM libros
WHERE editorial='Emece';

SELECT titulo||'-'||autor AS titulo_y_autor
FROM libros;

/** EJERCICIO 33
*****/

DROP TABLE IF EXISTS clientes;

CREATE TABLE clientes (
    documento CHAR(8),
    apellido VARCHAR(20),
    nombre VARCHAR(20),
    domicilio VARCHAR(30),
    telefono VARCHAR (11)
);
INSERT INTO clientes(documento,apellido,nombre,domicilio,telefono)
VALUES ('2233344','Perez','Juan','Sarmiento 980','4342345'),
('2333344','Perez','Ana','Colon 234','2345123'),
('2433344','Garcia','Luis','Avellaneda 1454','4558877'),
('2533344','Juarez','Ana','Urquiza 444','4789900');

SELECT documento,upper(apellido),upper(nombre),domicilio,telefono
```

```
FROM clientes;

SELECT documento,apellido,substr(nombre,1,1),domicilio,telefono
FROM clientes;

/** EJERCICIO 34
*****/

DROP TABLE IF EXISTS clientes;

CREATE TABLE clientes (
    codigo SERIAL,
    nombre VARCHAR(30) NOT NULL,
    domicilio VARCHAR(30),
    ciudad VARCHAR(20),
    provincia VARCHAR (20),
    credito DECIMAL(9,2),
    PRIMARY KEY(codigo)
);

INSERT INTO clientes(nombre,domicilio,ciudad,provincia,credito)
VALUES ('Lopez Marcos','Colon 111','Cordoba','Cordoba',1900.56),
('Perez Ana','San Martin 222','Cruz del Eje','Cordoba',450.33),
('Garcia Juan','Rivadavia 333','Villa del Rosario','Cordoba',190),
('Olmos Luis','Sarmiento 444','Rosario','Santa Fe',670.22),
('Pereyra Lucas','San Martin 555','Cruz del Eje','Cordoba',500.55);

SELECT * FROM clientes;

SELECT codigo,nombre,domicilio,ciudad,provincia,ceil(credito)
FROM clientes;

/** EJERCICIO 35
*****/

DROP TABLE IF EXISTS alumnos;

CREATE TABLE alumnos (
    apellido VARCHAR(30),
    nombre VARCHAR(30),
    documento CHAR(8),
    domicilio VARCHAR(30),
    fechaingreso DATE,
    fechanacimiento DATE
);

SET DATESTYLE to 'European';

INSERT INTO alumnos
VALUES ('Gonzalez','Ana','22222222','Colon 123','10-08-1990','15/02/1972'),
('Juarez','Bernardo','25555555','Sucre 456','03-03-1991','15/02/1972'),
('Perez','Laura','26666666','Bulnes 345','03-03-91',NULL);

SELECT * FROM alumnos
WHERE fechanacimiento is NULL;
```



```
SELECT apellido,nombre,EXTRACT(year FROM fechanacimiento)
FROM alumnos;
```

```
/** EJERCICIO 36
```

```
*****/
```

```
DROP TABLE IF EXISTS visitas;
```

```
CREATE TABLE visitas (
```

```
numero SERIAL,
nombre VARCHAR(30) DEFAULT 'Anonimo',
mail VARCHAR(50),
pais VARCHAR (20),
fecha TIMESTAMP,
PRIMARY KEY(numero)
```

```
);
```

```
INSERT INTO visitas (nombre,mail,pais,fecha)
```

```
VALUES ('Ana Maria Lopez','AnaMaria@hotmail.com','Argentina','2006-10-10 10:10'),
('Gustavo Gonzalez','GustavoGGonzalez@hotmail.com','Chile','2006-10-10 21:30'),
('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-10-11 15:45'),
('Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-10-12 08:15'),
('Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-09-12 20:45'),
('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-09-12 16:20'),
('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-09-15 16:25');
```

```
SELECT * FROM visitas
ORDER BY fecha DESC;
```

```
SELECT nombre,pais,EXTRACT(month FROM fecha)
FROM visitas
ORDER BY pais,EXTRACT(month FROM fecha) DESC;
```

```
SELECT nombre,mail,
EXTRACT(month FROM fecha) AS mes,
EXTRACT(day FROM fecha) AS dia,
EXTRACT(hour FROM fecha) AS hora
FROM visitas
ORDER BY 3,4,5;
```

```
SELECT mail, pais
FROM visitas
WHERE EXTRACT(month FROM fecha)=10
ORDER BY 2;
```

```
/** EJERCICIO 37
```

```
*****/
```

```
DROP TABLE IF EXISTS medicamentos;
```

```
CREATE TABLE medicamentos (
```

```
codigo SERIAL,
nombre VARCHAR(20),
laboratorio VARCHAR(20),
precio DECIMAL(5,2),
cantidad SMALLINT,
```

```

PRIMARY KEY(codigo)
);
INSERT INTO medicamentos (nombre,laboratorio,precio,cantidad)
VALUES ('Sertal','Roche',5.2,100),
('Buscapina','Roche',4.10,200),
('Amoxidal 500','Bayer',15.60,100),
('Paracetamol 500','Bago',1.90,200),
('Bayaspirina','Bayer',2.10,150);

INSERT INTO medicamentos (nombre,laboratorio,precio,cantidad)
VALUES ('Amoxidal jarabe','Bayer',5.10,250);

SELECT codigo,nombre
FROM medicamentos
WHERE laboratorio='Roche' AND precio < 5;

SELECT * FROM medicamentos
WHERE laboratorio='Roche' OR precio < 5;

SELECT * FROM medicamentos
WHERE NOT laboratorio='Bayer' AND cantidad=100;

SELECT * FROM medicamentos
WHERE laboratorio='Bayer' AND NOT cantidad=100;

DELETE FROM medicamentos
WHERE laboratorio='Bayer' AND precio > 10;

UPDATE medicamentos SET cantidad=200
WHERE laboratorio='Roche' AND precio > 5;

DELETE FROM medicamentos
WHERE laboratorio='Bayer' OR precio < 3;

/** EJERCICIO 38
*****/

DROP TABLE IF EXISTS peliculas;

CREATE TABLE peliculas (
codigo SERIAL,
titulo VARCHAR(40) NOT NULL,
actor VARCHAR(20),
duracion SMALLINT,
PRIMARY KEY (codigo)
);
INSERT INTO peliculas (titulo,actor,duracion)
VALUES ('Mision imposible','Tom Cruise',120),
('Harry Potter y la piedra filosofal','Daniel R.',180),
('Harry Potter y la camara secreta','Daniel R.',190),
('Mision imposible 2','Tom Cruise',120),
('Mujer bonita','Richard Gere',120),
('Tootsie','D. Hoffman',90),
('Un oso rojo','Julio Chavez',100),
('Elsa y Fred','China Zorrilla',110);

```

```
SELECT * FROM peliculas
WHERE actor='Tom Cruise' OR actor='Richard Gere';

SELECT * FROM peliculas
WHERE actor='Tom Cruise' AND duracion < 100;

UPDATE peliculas SET duracion=200
WHERE actor='Daniel R.' AND duracion=180;

DELETE FROM peliculas
WHERE NOT actor='Tom Cruise' AND duracion <>100;

/** EJERCICIO 39
*****/

DROP TABLE IF EXISTS peliculas;

CREATE TABLE peliculas (
    codigo SERIAL,
    titulo VARCHAR(40) NOT NULL,
    actor VARCHAR(20),
    duracion SMALLINT,
    PRIMARY KEY (codigo)
);

INSERT INTO peliculas(titulo,actor,duracion)
VALUES ('Mision imposible','Tom Cruise',120),
('Harry Potter y la piedra filosofal','Daniel R.',NULL),
('Harry Potter y la camara secreta','Daniel R.',190),
('Mision imposible 2','Tom Cruise',120),
('Mujer bonita',NULL,120),
('Tootsie','D. Hoffman',90),
('Un oso rojo');

SELECT * FROM peliculas
WHERE actor is NULL;

UPDATE peliculas SET duracion=0
WHERE duracion is NULL;

DELETE FROM peliculas
WHERE actor is NULL AND duracion=0;

SELECT * FROM peliculas;

/** EJERCICIO 40
*****/

DROP TABLE IF EXISTS visitas;

CREATE TABLE visitas (
    numero SERIAL,
    nombre VARCHAR(30) DEFAULT 'Anonimo',
    mail VARCHAR(50),
    pais VARCHAR (20),
    fechayhora TIMESTAMP,
```

```

PRIMARY KEY(numero)
);
INSERT INTO visitas (nombre,mail,pais,fechayhora)
VALUES ('Ana Maria Lopez','AnaMaria@hotmail.com','Argentina','2006-10-10 10:10'),
('Gustavo Gonzalez','GustavoGGonzalez@gotmail.com','Chile','2006-10-10 21:30'),
('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-10-11 15:45'),
('Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-10-12 08:15'),
('Fabiola Martinez','MartinezFabiola@hotmail.com','Mexico','2006-09-12 20:45'),
('Juancito','JuanJosePerez@gmail.com','Argentina','2006-09-12 16:20'),
('Juancito','JuanJosePerez@hotmail.com','Argentina','2006-09-15 16:25'),
('Federico1','federicogarcia@xaxamail.com','Argentina');

```

```

SELECT * FROM visitas
WHERE fechayhora BETWEEN '2006-09-12' AND '2006-10-11';

```

```

SELECT * FROM visitas
WHERE numero BETWEEN 2 AND 5;

```

```

/** EJERCICIO 41

```

```

*****/

```

```

DROP TABLE IF EXISTS autos;

```

```

CREATE TABLE autos (
    patente CHAR(6),
    marca VARCHAR(20),
    modelo CHAR(4),
    precio DECIMAL(8,2),
    PRIMARY KEY(patente)
);

```

```

INSERT INTO autos
VALUES ('ACD123','Fiat 128','1970',15000),
('ACG234','Renault 11','1980',40000),
('BCD333','Peugeot 505','1990',80000),
('GCD123','Renault Clio','1995',70000),
('BCC333','Renault Megane','1998',95000),
('BVF543','Fiat 128','1975',20000);

```

```

SELECT * FROM autos
WHERE modelo BETWEEN '1970' AND '1990'
ORDER BY modelo;

```

```

SELECT * FROM autos
WHERE precio BETWEEN 50000 AND 100000;

```

```

/** EJERCICIO 42

```

```

*****/

```

```

DROP TABLE IF EXISTS medicamentos;

```

```

CREATE TABLE medicamentos (
    codigo SERIAL,
    nombre VARCHAR(20),
    laboratorio VARCHAR(20),
    precio DECIMAL(6,2),

```

```

cantidad SMALLINT,
fechavencimiento DATE NOT NULL,
PRIMARY KEY(codigo)
);

INSERT INTO medicamentos(nombre,laboratorio,precio,cantidad,fechavencimiento)
VALUES ('Sertal','Roche',5.2,1,'2005-02-01'),
('Buscapina','Roche',4.10,3,'2006-03-01'),
('Amoxidal 500','Bayer',15.60,100,'2007-05-01'),
('Paracetamol 500','Bago',1.90,20,'2008-02-01'),
('Bayaspirina','Bayer',2.10,150,'2009-12-01'),
('Amoxidal jarabe','Bayer',5.10,250,'2010-10-01');

SELECT nombre,precio FROM medicamentos
WHERE laboratorio in ('Bayer','Bago');

SELECT * FROM medicamentos
WHERE cantidad BETWEEN 1 AND 5;

SELECT * FROM medicamentos
WHERE cantidad in (1,2,3,4,5);

/** EJERCICIO 43
*****/

DROP TABLE IF EXISTS empleados;

CREATE TABLE empleados (
nombre VARCHAR(30),
documento CHAR(8),
domicilio VARCHAR(30),
fechaingreso DATE,
seccion VARCHAR(20),
sueldo DECIMAL(6,2),
PRIMARY KEY(documento)
);

INSERT INTO empleados
VALUES ('Juan Perez','22333444','Colon 123','1990-10-08','Gerencia',900.50),
('Ana Acosta','23444555','Caseros 987','1995-12-18','Secretaria',590.30),
('Lucas Duarte','25666777','Sucre 235','2005-05-15','Sistemas',790),
('Pamela Gonzalez','26777888','Sarmiento 873','1999-02-12','Secretaria',550),
('Marcos Juarez','30000111','Rivadavia 801','2002-09-22','Contaduria',630.70),
('Yolanda Perez','35111222','Colon 180','1990-10-08','AdMINistracion',400),
('Rodolfo Perez','35555888','Coronel Olmedo 588','1990-05-28','Sistemas',800);

SELECT * FROM empleados
WHERE nombre like '%Perez%';

SELECT * FROM empleados
WHERE domicilio like 'Co%8%';

SELECT nombre,sueldo FROM empleados
WHERE sueldo NOT like '%.00';

SELECT * FROM empleados
WHERE fechaingreso like '%1990%';

```

```
/** EJERCICIO 44
***** /

DROP TABLE IF EXISTS medicamentos;

CREATE TABLE medicamentos (
    codigo SERIAL,
    nombre VARCHAR(20),
    laboratorio VARCHAR(20),
    precio DECIMAL(6,2),
    cantidad SMALLINT,
    fechavencimiento DATE NOT NULL,
    numerolote INT DEFAULT NULL,
    PRIMARY KEY(codigo)
);

INSERT INTO medicamentos(nombre,laboratorio,precio,cantidad,fechavencimiento,numerolote)
VALUES ('Sertal','Roche',5.2,1,'2005-02-01',NULL),
('Buscapina','Roche',4.10,3,'2006-03-01',NULL),
('Amoxidal 500','Bayer',15.60,100,'2007-05-01',NULL),
('Paracetamol 500','Bago',1.90,20,'2008-02-01',NULL),
('Bayaspirina',NULL,2.10,NULL,'2009-12-01',NULL),
('Amoxidal jarabe','Bayer',NULL,250,'2009-12-15',NULL);

SELECT COUNT(*)
FROM medicamentos;

SELECT COUNT(laboratorio)
FROM medicamentos;

SELECT COUNT(precio) AS conprecio,
COUNT(cantidad) AS concantidad
FROM medicamentos;

SELECT COUNT(precio)
FROM medicamentos
WHERE laboratorio like 'B%';

SELECT COUNT(numeralote)
FROM medicamentos;

/** EJERCICIO 45
***** /

DROP TABLE IF EXISTS empleados;

CREATE TABLE empleados (
    nombre VARCHAR(30),
    documento CHAR(8),
    domicilio VARCHAR(30),
    seccion VARCHAR(20),
    sueldo DECIMAL(6,2),
    cantidadhijos SMALLINT,
    PRIMARY KEY(documento)
```

```
);
```

```
INSERT INTO empleados
VALUES ('Juan Perez','22333444','Colon 123','Gerencia',5000,2),
('Ana Acosta','23444555','Caseros 987','Secretaria',2000,0),
('Lucas Duarte','25666777','Sucre 235','Sistemas',4000,1),
('Pamela Gonzalez','26777888','Sarmiento 873','Secretaria',2200,3),
('Marcos Juarez','30000111','Rivadavia 801','Contaduria',3000,0),
('Yolanda Perez','35111222','Colon 180','AdMINistracion',3200,1),
('Rodolfo Perez','35555888','Coronel Olmedo 588','Sistemas',4000,3),
('Martina Rodriguez','30141414','Sarmiento 1234','AdMINistracion',3800,4),
('Andres Costa','28444555',DEFAULT,'Secretaria',NULL,NULL);
```

```
SELECT COUNT(*)
FROM empleados;
```

```
SELECT COUNT(sueldo)
FROM empleados
WHERE seccion='Secretaria';
```

```
SELECT MAX(sueldo) AS Mayorsueldo,
MIN(sueldo) AS Menorsueldo
FROM empleados;
```

```
SELECT MAX(cantidadhijos)
FROM empleados
WHERE nombre like '%Perez%';
```

```
SELECT AVG(sueldo)
FROM empleados;
```

```
SELECT AVG(sueldo)
FROM empleados
WHERE seccion='Secretaria';
```

```
SELECT AVG(cantidadhijos)
FROM empleados
WHERE seccion='Sistemas';
```

```
/** EJERCICIO 46
```

```
*****/
```

```
DROP TABLE IF EXISTS visitantes;
```

```
CREATE TABLE visitantes (
nombre VARCHAR(30),
edad SMALLINT,
sexo CHAR(1) DEFAULT 'f',
domicilio VARCHAR(30),
ciudad VARCHAR(20) DEFAULT 'Cordoba',
telefono VARCHAR(11),
mail VARCHAR(30) DEFAULT 'no tiene',
montocompra DECIMAL (6,2)
);
```

```
INSERT INTO visitantes
VALUES ('Susana Molina',35,DEFAULT,'Colon 123',DEFAULT,NULL,NULL,59.80),
```

```
( 'Marcos Torres',29,'m',DEFAULT,'Carlos Paz',DEFAULT,'marcostorres@hotmail.com',
150.50),
( 'Mariana Juarez',45,DEFAULT,DEFAULT,'Carlos Paz',NULL,DEFAULT,23.90),
( 'Fabian Perez',36,'m','4556677','fabianperez@xaxamail.com'),
( 'Alejandra Gonzalez','La Falda',280.50),
( 'Gaston Perez',29,'m','Carlos Paz','gastonperez1@gmail.com',95.40),
( 'Liliana Torres',40,DEFAULT,'Sarmiento 876',DEFAULT,DEFAULT,DEFAULT,85),
( 'Gabriela Duarte',21,NULL,NULL,'Rio Tercero',DEFAULT,'gabrielaltorres@hotmail.com',
321.50);
```

```
SELECT ciudad, COUNT(*)
FROM visitantes
GROUP BY ciudad;
```

```
SELECT ciudad, COUNT(telefono)
FROM visitantes
GROUP BY ciudad;
```

```
SELECT sexo, SUM(montocompra)
FROM visitantes
GROUP BY sexo;
```

```
SELECT sexo,ciudad,
MAX(montocompra) AS mayor,
MIN(montocompra) AS menor FROM visitantes
GROUP BY sexo,ciudad;
```

```
SELECT ciudad,
AVG(montocompra) AS promediodecompras
FROM visitantes
GROUP BY ciudad;
```

```
SELECT ciudad,
COUNT(*) AS cantidadconmail
FROM visitantes
WHERE mail is NOT NULL AND mail <> 'no tiene'
GROUP BY ciudad;
```

```
/** EJERCICIO 47
```

```
*****/
```

```
DROP TABLE IF EXISTS empleados;
```

```
CREATE TABLE empleados (
nombre VARCHAR(30),
documento CHAR(8),
domicilio VARCHAR(30),
seccion VARCHAR(20),
sueldo DECIMAL(6,2),
cantidadhijos SMALLINT,
fechaingreso DATE,
PRIMARY KEY(documento)
);
```

```
INSERT INTO empleados
VALUES ('Juan Perez','22333444','Colon 123','Gerencia',5000,2,'1980-05-10'),
```



```
( 'Ana Acosta', '23444555', 'Caseros 987', 'Secretaria', 2000, 0, '1980-10-12' ),
( 'Lucas Duarte', '25666777', 'Sucre 235', 'Sistemas', 4000, 1, '1985-05-25' ),
( 'Pamela Gonzalez', '26777888', 'Sarmiento 873', 'Secretaria', 2200, 3, '1990-06-25' ),
( 'Marcos Juarez', '30000111', 'Rivadavia 801', 'Contaduria', 3000, 0, '1996-05-01' ),
( 'Yolanda Perez', '35111222', 'Colon 180', 'AdMINistracion', 3200, 1, '1996-05-01' ),
( 'Rodolfo Perez', '35555888', 'Coronel Olmedo 588', 'Sistemas', 4000, 3, '1996-05-01' ),
( 'Martina Rodriguez', '30141414', 'Sarmiento 1234', 'AdMINistracion', 3800, 4, '2000-09-01'
),
( 'Andres Costa', '28444555', DEFAULT, 'Secretaria', NULL, NULL, NULL );
```

```
SELECT seccion, COUNT(*)
FROM empleados
GROUP BY seccion;
```

```
SELECT seccion, AVG(cantidadhijos) AS promediodehijos
FROM empleados
GROUP BY seccion;
```

```
SELECT EXTRACT(year FROM fechaingreso), COUNT(*)
FROM empleados
GROUP BY EXTRACT(year FROM fechaingreso);
```

```
SELECT seccion, AVG(sueldo) AS promediodesueldo
FROM empleados
WHERE cantidadhijos > 0 AND cantidadhijos is NOT NULL
GROUP BY seccion;
```

```
/** EJERCICIO 48
```

```
*****/
```

```
DROP TABLE IF EXISTS clientes;
CREATE TABLE clientes (
    codigo SERIAL,
    nombre VARCHAR(30) NOT NULL,
    domicilio VARCHAR(30),
    ciudad VARCHAR(20),
    provincia VARCHAR (20),
    telefono VARCHAR(11),
    PRIMARY KEY(codigo)
);
```

```
INSERT INTO clientes(nombre,domicilio,ciudad,provincia,telefono)
VALUES ( 'Lopez Marcos', 'Colon 111', 'Cordoba', 'Cordoba', 'NULL' ),
( 'Perez Ana', 'San Martin 222', 'Cruz del Eje', 'Cordoba', '4578585' ),
( 'Garcia Juan', 'Rivadavia 333', 'Villa del Rosario', 'Cordoba', '4578445' ),
( 'Perez Luis', 'Sarmiento 444', 'Rosario', 'Santa Fe', NULL ),
( 'Pereyra Lucas', 'San Martin 555', 'Cruz del Eje', 'Cordoba', '4253685' ),
( 'Gomez Ines', 'San Martin 666', 'Santa Fe', 'Santa Fe', '0345252525' ),
( 'Torres Fabiola', 'Alem 777', 'Villa del Rosario', 'Cordoba', '4554455' ),
( 'Lopez Carlos', NULL, 'Cruz del Eje', 'Cordoba', NULL ),
( 'Ramos Betina', 'San Martin 999', 'Cordoba', 'Cordoba', '4223366' ),
( 'Lopez Lucas', 'San Martin 1010', 'Posadas', 'Misiones', '0457858745' );
```

```
SELECT ciudad, provincia,
COUNT(*) AS cantidad
FROM clientes
```

```

GROUP BY ciudad,provincia;

SELECT ciudad, provincia,
COUNT(*) AS cantidad
FROM clientes
GROUP BY ciudad,provincia
HAVING COUNT(*) > 1;

/** EJERCICIO 49
*****/

DROP TABLE IF EXISTS visitantes;

CREATE TABLE visitantes (
nombre VARCHAR(30),
edad SMALLINT,
sexo CHAR(1),
domicilio VARCHAR(30),
ciudad VARCHAR(20),
telefono VARCHAR(11),
montocompra DECIMAL(6,2) NOT NULL
);
INSERT INTO visitantes
VALUES ('Susana Molina',28,'f',NULL,'Cordoba',NULL,45.50),
('Marcela Mercado',36,'f','Avellaneda 345','Cordoba','4545454',22.40),
('Alberto Garcia',35,'m','Gral. Paz 123','Alta Gracia','03547123456',25),
('Teresa Garcia',33,'f',DEFAULT,'Alta Gracia','03547123456',120),
('Roberto Perez',45,'m','Urquiza 335','Cordoba','4123456',33.20),
('Marina Torres',22,'f','Colon 222','Villa Dolores','03544112233',95),
('Julietta Gomez',24,'f','San Martin 333','Alta Gracia',NULL,53.50),
('Roxana Lopez',20,'f','NULL','Alta Gracia',NULL,240),
('Liliana Garcia',50,'f','Paso 999','Cordoba','4588778',48),
('Juan Torres',43,'m','Sarmiento 876','Cordoba',NULL,15.30);

SELECT ciudad,sexo,
SUM(montocompra) AS Total
FROM visitantes
GROUP BY ciudad,sexo
HAVING SUM(montocompra) > 50;

SELECT ciudad,sexo, COUNT(*) AS cantidad,
SUM(montocompra) AS total,
AVG(montocompra) AS promedio
FROM visitantes
GROUP BY ciudad,sexo
HAVING AVG(montocompra) > 30
ORDER BY total;

/** EJERCICIO 50
*****/

DROP TABLE IF EXISTS clientes;

CREATE TABLE clientes (
codigo SERIAL,

```

```

nombre VARCHAR(30) NOT NULL,
domicilio VARCHAR(30),
ciudad VARCHAR(20),
provincia VARCHAR (20),
PRIMARY KEY(codigo)
);

INSERT INTO clientes(nombre,domicilio,ciudad,provincia)
VALUES ('Lopez Marcos','Colon 111','Cordoba','Cordoba'),
('Perez Ana','San Martin 222','Cruz del Eje','Cordoba'),
('Garcia Juan','Rivadavia 333','Villa del Rosario','Cordoba'),
('Perez Luis','Sarmiento 444','Rosario','Santa Fe'),
('Pereyra Lucas','San Martin 555','Cruz del Eje','Cordoba'),
('Gomez Ines','San Martin 666','Santa Fe','Santa Fe'),
('Torres Fabiola','Alem 777','Villa del Rosario','Cordoba'),
('Lopez Carlos',NULL,'Cruz del Eje','Cordoba'),
('Ramos Betina','San Martin 999','Cordoba','Cordoba'),
('Lopez Lucas','San Martin 1010','Posadas','Misiones');

SELECT distinct provincia FROM clientes;

SELECT COUNT(distinct provincia) AS cantidad
FROM clientes;

SELECT distinct ciudad FROM clientes;

SELECT COUNT(distinct ciudad) FROM clientes;

SELECT distinct ciudad FROM clientes
WHERE provincia='Cordoba';

SELECT provincia,COUNT(distinct ciudad)
FROM clientes
GROUP BY provincia;

/** EJERCICIO 51
*****

DROP TABLE IF EXISTS inmuebles;

CREATE TABLE inmuebles (
documento VARCHAR(8) NOT NULL,
apellido VARCHAR(30),
nombre VARCHAR(30),
domicilio VARCHAR(20),
barrio VARCHAR(20),
ciudad VARCHAR(20),
tipo CHAR(1),
superficie DECIMAL (8,2)
);

INSERT INTO inmuebles
VALUES ('11000000','Perez','Alberto','San Martin 800','Centro','Cordoba','e',100),
('11000000','Perez','Alberto','Sarmiento 245','Gral. Paz','Cordoba','e',200),
('12222222','Lopez','Maria','San Martin 202','Centro','Cordoba','e',250),
('13333333','Garcia','Carlos','Paso 1234','Alberdi','Cordoba','b',200),

```

```
( '13333333', 'Garcia', 'Carlos', 'Guemes 876', 'Alberdi', 'Cordoba', 'b', 300 ),
( '14444444', 'Perez', 'Mariana', 'Caseros 456', 'Flores', 'Cordoba', 'b', 200 ),
( '15555555', 'Lopez', 'Luis', 'San Martin 321', 'Centro', 'Carlos Paz', 'e', 500 ),
( '15555555', 'Lopez', 'Luis', 'Lopez y Planes 853', 'Flores', 'Carlos Paz', 'e', 350 ),
( '16666666', 'Perez', 'Alberto', 'Sucre 1877', 'Flores', 'Cordoba', 'e', 150 );
```

```
SELECT distinct apellido FROM inmuebles;
```

```
SELECT distinct documento FROM inmuebles;
```

```
SELECT COUNT(distinct documento)
FROM inmuebles
WHERE ciudad='Cordoba';
```

```
SELECT COUNT(ciudad)
FROM inmuebles
WHERE domicilio like 'San Martin %';
```

```
SELECT distinct apellido,nombre
FROM inmuebles;
```

```
SELECT documento,COUNT(distinct barrio) AS cantidad
FROM inmuebles
GROUP BY documento;
```

```
/** EJERCICIO 52
```

```
*****/
```

```
DROP TABLE IF EXISTS consultas;
```

```
CREATE TABLE consultas (
    fechayhora TIMESTAMP NOT NULL,
    medico VARCHAR(30) NOT NULL,
    documento CHAR(8) NOT NULL,
    paciente VARCHAR(30),
    obrasocial VARCHAR(30),
    PRIMARY KEY(fechayhora,medico)
);
```

```
INSERT INTO consultas
VALUES ( '2006/11/05 8:00', 'Lopez', '12222222', 'Acosta Betina', 'PAMI');
```

```
INSERT INTO consultas
VALUES ( '2006/11/05 8:30', 'Lopez', '23333333', 'Fuentes Carlos', 'PAMI');
```

```
INSERT INTO consultas
VALUES ( '2006/11/05 8:00', 'Perez', '34444444', 'Garcia Marisa', 'IPAM');
```

```
INSERT INTO consultas
VALUES ( '2006/11/05 8:00', 'Duarte', '45555555', 'Pereyra Luis', 'PAMI');
```

```
INSERT INTO consultas
VALUES ( '2006/11/05 8:00', 'Perez', '23333333', 'Fuentes Carlos', 'PAMI');
```

```
/** EJERCICIO 53
```

```
*****/
```

```
DROP TABLE IF EXISTS inscritos;
```

```
CREATE TABLE inscritos (  
    documento CHAR(8) NOT NULL,  
    nombre VARCHAR(30),  
    deporte VARCHAR(15) NOT NULL,  
    anio SMALLINT,  
    matricula CHAR(1),  
    PRIMARY KEY(documento,deporte,anio)  
);
```

```
INSERT INTO inscritos  
VALUES ('12222222','Juan Perez','tenis','2005','s'),  
       ('23333333','Marta Garcia','tenis','2005','s'),  
       ('34444444','Luis Perez','tenis','2005','n');
```

```
INSERT INTO inscritos  
VALUES ('12222222','Juan Perez','futbol','2005','s');
```

```
INSERT INTO inscritos  
VALUES ('12222222','Juan Perez','natacion','2005','s');
```

```
INSERT INTO inscritos  
VALUES ('12222222','Juan Perez','basquet','2005','n');
```

```
INSERT INTO inscritos  
VALUES ('12222222','Juan Perez','tenis','2006','s');
```

```
INSERT INTO inscritos  
VALUES ('12222222','Juan Perez','tenis','2007','s');
```

```
INSERT INTO inscritos  
VALUES ('12222222','Juan Perez','tenis','2005','s');
```

```
UPDATE inscritos SET deporte='tenis'  
WHERE documento='12222222' AND deporte='futbol' AND anio='2005';
```

```
/** EJERCICIO 54
```

```
*****/
```

```
DROP TABLE IF EXISTS empleados;
```

```
CREATE TABLE empleados (  
    documento VARCHAR(8),  
    nombre VARCHAR(30),  
    fechanacimiento DATE,  
    cantidadhijos SMALLINT,  
    seccion VARCHAR(20),  
    sueldo DECIMAL(6,2)  
);
```

```
ALTER TABLE empleados  
ADD CONSTRAINT CK_empelados_sueldo_positivo  
CHECK (sueldo > 0);
```

```
INSERT INTO empleados
```

```
VALUES ('22222222','Alberto Lopez','1965/10/05',1,'Sistemas',1000),
('33333333','Beatriz Garcia','1972/08/15',2,'AdMINistracion',3000),
('34444444','Carlos Caseres','1980/10/05',0,'Contadur&iacute;a',6000);
```

```
ALTER TABLE empleados
ADD CONSTRAINT CK_empleados_sueldo_maximo
CHECK (sueldo <>5000);
```

```
DELETE FROM empleados WHERE sueldo=6000;
```

```
ALTER TABLE empleados
ADD CONSTRAINT CK_empleados_sueldo_maximo
CHECK (sueldo <>5000);
```

```
ALTER TABLE empleados
ADD CONSTRAINT CK_fechanacimiento_actual
CHECK (fechanacimiento < CURRENT_DATE);
```

```
ALTER TABLE empleados
ADD CONSTRAINT CK_cantidadhijos_valores
CHECK (cantidadhijos BETWEEN 0 AND 15);
```

```
SELECT *
FROM information_schema.table_constraints
WHERE table_name = 'empleados';
```

```
INSERT INTO empleados
VALUES ('24444444','Carlos Fuentes','1980/02/05',2,'AdMINistracion',-1500);
```

```
INSERT INTO empleados
VALUES ('25555555','Daniel Garcia','2007/05/05',2,'Sistemas',1550);
```

```
UPDATE empleados SET cantidadhijos=21 WHERE documento='22222222';
```

```
/** EJERCICIO 55
```

```
*****/
```

```
DROP TABLE IF EXISTS vehiculos;
```

```
CREATE TABLE vehiculos (
numero SERIAL,
patente CHAR(6),
tipo CHAR(4),
fechahoraentrada TIMESTAMP,
fechahorasalida TIMESTAMP,
PRIMARY KEY(numero)
);
```

```
INSERT INTO vehiculos (patente,tipo,fechahoraentrada,fechahorasalida)
VALUES ('AIC124','auto','2007/01/17 8:05','2007/01/17 12:30'),
('CAA258','auto','2007/01/17 8:10',NULL),
('DSE367','moto','2007/01/17 8:30','2007/01/17 18:00');
```

```
ALTER TABLE vehiculos
ADD CONSTRAINT CK_vehiculos_fechahoraentrada_actual
CHECK (fechahoraentrada <>CURRENT_DATE);
```

```
ALTER TABLE vehiculos
ADD CONSTRAINT CK_vehiculos_fechahoraentradasalida
CHECK (fechahoraentrada <>fechahorasalida);

INSERT INTO vehiculos (patente,tipo,fechahoraentrada,fechahorasalida)
VALUES ('ABC123','auto','2007/05/05 10:10',NULL);

UPDATE vehiculos SET fechahorasalida='2007/01/17 7:30'
WHERE patente='CAA258';

SELECT *
FROM information_schema.table_constraints
WHERE table_name = 'vehiculos';

SELECT * FROM vehiculos;

/** EJERCICIO 56
*****/

DROP TABLE IF EXISTS empleados;

CREATE TABLE empleados (
    documento VARCHAR(8) NOT NULL,
    nombre VARCHAR(30),
    seccion VARCHAR(20)
);

INSERT INTO empleados
VALUES ('22222222','Alberto Lopez','Sistemas'),
('23333333','Beatriz Garcia','AdMINistracion'),
('23333333','Carlos Fuentes','AdMINistracion');

ALTER TABLE empleados
ADD CONSTRAINT PK_empleados_documento
PRIMARY KEY(documento);

DELETE FROM empleados
WHERE nombre='Carlos Fuentes';

ALTER TABLE empleados
ADD CONSTRAINT PK_empleados_documento
PRIMARY KEY(documento);

UPDATE empleados SET documento='22222222'
WHERE documento='23333333';

ALTER TABLE empleados
ADD CONSTRAINT PK_empleados_nombre
PRIMARY KEY(nombre);

INSERT INTO empleados VALUES (NULL,'Marcelo Juarez','Sistemas');

SELECT * FROM empleados;

SELECT *
FROM information_schema.table_constraints
```

```
WHERE table_name = 'empleados';
```

```
/** EJERCICIO 57
```

```
*****/
```

```
DROP TABLE IF EXISTS remis;
```

```
CREATE TABLE remis (  
    numero SERIAL,  
    patente CHAR(6),  
    marca VARCHAR(15),  
    modelo CHAR(4)  
);
```

```
INSERT INTO remis (patente,marca,modelo)VALUES ('ABC123','Renault 12','1990');
```

```
INSERT INTO remis (patente,marca,modelo)VALUES ('DEF456','Fiat Duna','1995');
```

```
ALTER TABLE remis  
    ADD CONSTRAINT PK_remis_patente  
    PRIMARY KEY(patente);
```

```
ALTER TABLE remis  
    ADD CONSTRAINT PK_remis_numero  
    PRIMARY KEY(numero);
```

```
SELECT *  
    FROM information_schema.table_constraints  
    WHERE table_name = 'remis';
```

```
/** EJERCICIO 58
```

```
*****/
```

```
DROP TABLE IF EXISTS remis;
```

```
CREATE TABLE remis (  
    numero SERIAL,  
    patente CHAR(6),  
    marca VARCHAR(15),  
    modelo CHAR(4)  
);
```

```
INSERT INTO remis(patente,marca,modelo)  
VALUES ('ABC123','Renault clio','1990'),  
    ('DEF456','Peugeot 504','1995'),  
    ('DEF456','Fiat Duna','1998'),  
    ('GHI789','Fiat Duna','1995'),  
    (NULL,'Fiat Duna','1995');
```

```
ALTER TABLE remis  
    ADD CONSTRAINT UQ_remis_patente  
    UNIQUE(patente);
```

```
DELETE FROM remis WHERE numero=3;
```



```
ALTER TABLE remis
  ADD CONSTRAINT UQ_remis_patente
  UNIQUE(patente);

INSERT INTO remis (patente,marca,modelo)VALUES ('ABC123','Renault 11','1995');

INSERT INTO remis(patente,marca,modelo) VALUES (NULL,'Renault 11','1995');

SELECT *
  FROM information_schema.table_constraints
  WHERE table_name = 'remis';

/** EJERCICIO 59
*****/

DROP TABLE IF EXISTS vehiculos;

CREATE TABLE vehiculos (
  patente CHAR(6) NOT NULL,
  tipo CHAR(1),
  horaallegada TIMESTAMP NOT NULL,
  horasalida TIMESTAMP
);

ALTER TABLE vehiculos
  ADD CONSTRAINT PK_vehiculos_patentellegada
  PRIMARY KEY(patente,horaallegada);

INSERT INTO vehiculos
  VALUES ('SDR456','a','2005/10/10 10:10',NULL),
  ('SDR456','m','2005/10/10 10:10',NULL),
  ('SDR456','m','2005/10/10 12:10',NULL),
  ('SDR111','m','2005/10/10 10:10',NULL);

SELECT *
  FROM information_schema.table_constraints
  WHERE table_name = 'vehiculos';

ALTER TABLE vehiculos
  DROP CONSTRAINT PK_vehiculos_patentellegada;

SELECT *
  FROM information_schema.table_constraints
  WHERE table_name = 'vehiculos';

/** EJERCICIO 60
*****/

DROP TABLE IF EXISTS agenda;

CREATE TABLE agenda (
  apellido VARCHAR(30),
  nombre VARCHAR(20) NOT NULL,
  domicilio VARCHAR(30),
  telefono VARCHAR(11),
```

```

mail VARCHAR(30)
);

INSERT INTO agenda
VALUES ('Perez','Juan','Sarmiento 345','4334455','juancito@gmail.com'),
('Garcia','Ana','Urquiza 367','4226677','anamariagarcia@hotmail.com'),
('Lopez','Juan','Avellaneda 900',NULL,'juancitoLopez@gmail.com'),
('Juarez','Mariana','Sucre 123','0525657687','marianaJuarez2@gmail.com'),
('Molinari','Lucia','Peru 1254','4590987','molinarilucia@hotmail.com'),
('Ferreyra','Patricia','Colon 1534','4585858',NULL),
('Perez','Susana','San Martin 333',NULL,NULL),
('Perez','Luis','Urquiza 444','0354545256','perezluisalberto@hotmail.com'),
('Lopez','Maria','Salta 314',NULL,'lopezmariayo@gmail.com');

CREATE INDEX I_agenda_apellido ON agenda(apellido);

CREATE UNIQUE INDEX I_agenda_mail ON agenda(mail);

DROP INDEX I_agenda_apellido;

DROP INDEX I_agenda_mail;

/** EJERCICIO 61
*****/

DROP TABLE IF EXISTS agenda;

CREATE TABLE agenda (
    apellido VARCHAR(30),
    nombre VARCHAR(20) NOT NULL,
    domicilio VARCHAR(30),
    telefono VARCHAR(11),
    mail VARCHAR(30)
);

INSERT INTO agenda(apellido,nombre,domicilio,telefono,mail)
VALUES ('Martinez','Pablo','Colon 233','236712','mpablo@gmail.com'),
('Rodriguez','Luis','Dean Funes 345','451111','rluis@yahoo.com.ar'),
('Lopez','Ana','Peron 344','349696','alopez@hotmail.com'),
('Cosario','Carla','Leones 334','349595','ccorsario@gmail.com'),
('Revol','Carlos','Lima 333','6712343','crevol@gmail.com');

SELECT * FROM agenda LIMIT 3;

SELECT * FROM agenda LIMIT 3 OFFSET 2;

SELECT * FROM agenda ORDER BY apellido LIMIT 4 OFFSET 2;

/** EJERCICIO 62
*****/

DROP TABLE IF EXISTS clientes;

CREATE TABLE clientes (
    codigo SERIAL,

```

```

nombre VARCHAR(30),
domicilio VARCHAR(30),
ciudad VARCHAR(20),
codigoprovincia SMALLINT NOT NULL,
PRIMARY KEY(codigo)
);
DROP TABLE IF EXISTS provincias;

CREATE TABLE provincias (
codigo SERIAL,
nombre VARCHAR(20),
PRIMARY KEY (codigo)
);

INSERT INTO provincias (nombre)
VALUES ('Cordoba'),
('Santa Fe'),
('Corrientes');

INSERT INTO clientes(nombre,domicilio,ciudad,codigoprovincia)
VALUES ('Lopez Marcos','Colon 111','Córdoba',1),
('Perez Ana','San Martin 222','Cruz del Eje',1),
('Garcia Juan','Rivadavia 333','Villa Maria',1),
('Perez Luis','Sarmiento 444','Rosario',2),
('Pereyra Lucas','San Martin 555','Cruz del Eje',1),
('Gomez Ines','San Martin 666','Santa Fe',2),
('Torres Fabiola','Alem 777','Ibera',3);

SELECT c.nombre,domicilio,ciudad,p.nombre
FROM clientes AS c
JOIN provincias AS p
ON c.codigoprovincia=p.codigo;

SELECT c.nombre,domicilio,ciudad,p.nombre
FROM clientes AS c
JOIN provincias AS p
ON c.codigoprovincia=p.codigo
ORDER BY p.nombre;

SELECT c.nombre,domicilio,ciudad
FROM clientes AS c
JOIN provincias AS p
ON c.codigoprovincia=p.codigo
WHERE p.nombre='Santa Fe';

```

/** EJERCICIO 63

*****/

```

DROP TABLE IF EXISTS inscritos;

```

```

CREATE TABLE inscritos (
nombre VARCHAR(30),
documento CHAR(8),
deporte VARCHAR(15),
matricula CHAR(1),
PRIMARY KEY(documento,deporte)

```

```
);  
DROP TABLE IF EXISTS inasistencias;  
  
CREATE TABLE inasistencias (  
    documento CHAR(8),  
    deporte VARCHAR(15),  
    fecha DATE  
);  
  
INSERT INTO inscritos  
VALUES ('Juan Perez','22222222','tenis','s'),  
('Maria Lopez','23333333','tenis','s'),  
('Agustin Juarez','24444444','tenis','n'),  
('Marta Garcia','25555555','natacion','s'),  
('Juan Perez','22222222','natacion','s'),  
('Maria Lopez','23333333','natacion','n');  
  
INSERT INTO inasistencias  
VALUES ('22222222','tenis','2006-12-01'),  
('22222222','tenis','2006-12-08'),  
('23333333','tenis','2006-12-01'),  
('24444444','tenis','2006-12-08'),  
('22222222','natacion','2006-12-02'),  
('23333333','natacion','2006-12-02');  
  
SELECT nombre,insc.deporte,ina.fecha  
FROM inscritos AS insc  
JOIN inasistencias AS ina  
ON insc.documento=ina.documento AND insc.deporte=ina.deporte  
ORDER BY nombre, insc.deporte;  
  
SELECT nombre,insc.deporte, ina.fecha  
FROM inscritos AS insc  
JOIN inasistencias AS ina  
ON insc.documento=ina.documento AND insc.deporte=ina.deporte  
WHERE insc.documento='22222222';  
  
SELECT nombre,insc.deporte, ina.fecha  
FROM inscritos AS insc  
JOIN inasistencias AS ina  
ON insc.documento=ina.documento AND insc.deporte=ina.deporte  
WHERE insc.matricula='s';  
  
/** EJERCICIO 64  
*****/  
  
DROP TABLE IF EXISTS clientes;  
  
CREATE TABLE clientes (  
    codigo SERIAL,  
    nombre VARCHAR(30),  
    domicilio VARCHAR(30),  
    ciudad VARCHAR(20),  
    codigoprovincia SMALLINT NOT NULL,  
    PRIMARY KEY(codigo)  
);
```

```
DROP TABLE IF EXISTS provincias;
```

```
CREATE TABLE provincias (  
    codigo SERIAL,  
    nombre VARCHAR(20),  
    PRIMARY KEY (codigo)  
);
```

```
INSERT INTO provincias (nombre)  
VALUES ('Cordoba'),  
('Santa Fe'),  
('Corrientes');
```

```
INSERT INTO clientes(nombre,domicilio,ciudad,codigoprovincia)  
VALUES ('Lopez Marcos','Colon 111','Córdoba',1),  
('Perez Ana','San Martin 222','Cruz del Eje',1),  
('Garcia Juan','Rivadavia 333','Villa Maria',1),  
('Perez Luis','Sarmiento 444','Rosario',2),  
('Gomez Ines','San Martin 666','Santa Fe',2),  
('Torres Fabiola','Alem 777','La Plata',4),  
('Garcia Luis','Sucre 475','Santa Rosa',5);
```

```
SELECT c.nombre,domicilio,ciudad, p.nombre  
FROM clientes AS c  
LEFT JOIN provincias AS p  
ON codigoprovincia = p.codigo;
```

```
SELECT c.nombre,domicilio,ciudad, p.nombre  
FROM provincias AS p  
LEFT JOIN clientes AS c  
ON codigoprovincia = p.codigo;
```

```
SELECT c.nombre,domicilio,ciudad, p.nombre  
FROM clientes AS c  
LEFT JOIN provincias AS p  
ON codigoprovincia = p.codigo  
WHERE p.codigo is NOT NULL;
```

```
SELECT c.nombre,domicilio,ciudad, p.nombre  
FROM clientes AS c  
LEFT JOIN provincias AS p  
ON codigoprovincia = p.codigo  
WHERE p.codigo is NULL  
ORDER BY c.nombre;
```

```
SELECT c.nombre,domicilio,ciudad, p.nombre  
FROM clientes AS c  
LEFT JOIN provincias AS p  
ON codigoprovincia = p.codigo  
WHERE p.nombre='Cordoba';
```

```
/** EJERCICIO 65
```

```
*****/
```

```
DROP TABLE IF EXISTS clientes;
```

```

CREATE TABLE clientes (
    codigo SERIAL,
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
    ciudad VARCHAR(20),
    codigoprovincia SMALLINT NOT NULL,
    PRIMARY KEY(codigo)
);

DROP TABLE IF EXISTS provincias;

CREATE TABLE provincias (
    codigo SERIAL,
    nombre VARCHAR(20),
    PRIMARY KEY (codigo)
);

INSERT INTO provincias (nombre)
VALUES ('Cordoba'),
       ('Santa Fe'),
       ('Corrientes');

INSERT INTO clientes(nombre,domicilio,ciudad,codigoprovincia)
VALUES ('Lopez Marcos','Colon 111','Córdoba',1),
       ('Perez Ana','San Martin 222','Cruz del Eje',1),
       ('Garcia Juan','Rivadavia 333','Villa Maria',1),
       ('Perez Luis','Sarmiento 444','Rosario',2),
       ('Gomez Ines','San Martin 666','Santa Fe',2),
       ('Torres Fabiola','Alem 777','La Plata',4),
       ('Garcia Luis','Sucre 475','Santa Rosa',5);

SELECT c.nombre,domicilio,ciudad, p.nombre
FROM provincias AS p
RIGHT JOIN clientes AS c
ON codigoprovincia = p.codigo;

SELECT c.nombre,domicilio,ciudad, p.nombre
FROM clientes AS c
LEFT JOIN provincias AS p
ON codigoprovincia = p.codigo;

SELECT c.nombre,domicilio,ciudad, p.nombre
FROM provincias AS p
RIGHT JOIN clientes AS c
ON codigoprovincia = p.codigo
WHERE p.codigo is NOT NULL;

SELECT c.nombre,domicilio,ciudad, p.nombre
FROM provincias AS p
RIGHT JOIN clientes AS c
ON codigoprovincia = p.codigo
WHERE p.codigo is NULL
ORDER BY ciudad;

```

/** EJERCICIO 66

*****/

```
DROP TABLE IF EXISTS deportes;
```

```
CREATE TABLE deportes (  
    codigo SERIAL,  
    nombre VARCHAR(30),  
    profesor VARCHAR(30),  
    PRIMARY KEY (codigo)  
);
```

```
DROP TABLE IF EXISTS inscritos;
```

```
CREATE TABLE inscritos (  
    documento CHAR(8),  
    codigodeporte SMALLINT NOT NULL,  
    matricula CHAR(1)  
);
```

```
INSERT INTO deportes(nombre,profesor)  
VALUES ('tenis','Marcelo Roca'),  
('natacion','Marta Torres'),  
('basquet','Luis Garcia'),  
('futbol','Marcelo Roca');
```

```
INSERT INTO inscritos  
VALUES ('22222222',3,'s'),  
('23333333',3,'s'),  
('24444444',3,'n'),  
('22222222',2,'s'),  
('23333333',2,'s'),  
('22222222',4,'n'),  
('22222222',5,'n');
```

```
SELECT documento,d.nombre,matricula  
FROM inscritos AS i  
JOIN deportes AS d  
ON codigodeporte=codigo;
```

```
SELECT documento,d.nombre,matricula  
FROM inscritos AS i  
LEFT JOIN deportes AS d  
ON codigodeporte=codigo;
```

```
SELECT documento,d.nombre,matricula  
FROM deportes AS d  
RIGHT JOIN inscritos AS i  
ON codigodeporte=codigo;
```

```
SELECT nombre  
FROM deportes AS d  
LEFT JOIN inscritos AS i  
ON codigodeporte=codigo  
WHERE codigodeporte is NULL;
```

```
SELECT documento  
FROM inscritos AS i  
LEFT JOIN deportes AS d
```

```
ON codigodeporte=codigo
WHERE codigo is NULL;
```

```
SELECT documento,nombre,profesor,matricula
FROM inscritos AS i
FULL JOIN deportes AS d
ON codigodeporte=codigo;
```

```
/** EJERCICIO 67
```

```
***** /
```

```
DROP TABLE IF EXISTS mujeres;
```

```
CREATE TABLE mujeres (
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
    edad INT
);
```

```
DROP TABLE IF EXISTS varones;
```

```
CREATE TABLE varones (
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
    edad INT
);
```

```
INSERT INTO mujeres
VALUES ('Maria Lopez','Colon 123',45),
('Liliana Garcia','Sucre 456',35),
('Susana Lopez','Avellaneda 98',41);
```

```
INSERT INTO varones
VALUES ('Juan Torres','Sarmiento 755',44),
('Marcelo Oliva','San Martin 874',56),
('Federico Pereyra','Colon 234',38),
('Juan Garcia','Peru 333',50);
```

```
SELECT m.nombre,m.edad,v.nombre,v.edad
FROM mujeres AS m
CROSS JOIN varones AS v;
```

```
SELECT m.nombre,m.edad,v.nombre,v.edad
FROM mujeres AS m
CROSS JOIN varones AS v
WHERE m.edad > 40 AND v.edad > 40;
```

```
SELECT m.nombre,m.edad,v.nombre,v.edad
FROM mujeres AS m
CROSS JOIN varones AS v
WHERE m.edad-v.edad BETWEEN -10 AND 10;
```

```
/** EJERCICIO 68
```

```
***** /
```

```
DROP TABLE IF EXISTS guardias;
```



```

CREATE TABLE guardias (
    documento CHAR(8),
    nombre VARCHAR(30),
    sexo CHAR(1),
    domicilio VARCHAR(30),
    PRIMARY KEY (documento)
);

DROP TABLE IF EXISTS tareas;

CREATE TABLE tareas (
    codigo SERIAL,
    domicilio VARCHAR(30),
    descripcion VARCHAR(30),
    horario CHAR(2),
    PRIMARY KEY (codigo)
);

INSERT INTO guardias
VALUES ('22333444','Juan Perez','m','Colon 123'),
      ('24333444','Alberto Torres','m','San Martin 567'),
      ('25333444','Luis Ferreyra','m','Chacabuco 235'),
      ('23333444','Lorena Viale','f','Sarmiento 988'),
      ('26333444','Irma Gonzalez','f','Mariano Moreno 111');

INSERT INTO tareas(domicilio,descripcion,horario)
VALUES ('Colon 1111','vigilancia exterior','AM'),
      ('Urquiza 234','vigilancia exterior','PM'),
      ('Peru 345','vigilancia interior','AM'),
      ('Avellaneda 890','vigilancia interior','PM');

SELECT nombre,t.domicilio,descripcion
FROM guardias
CROSS JOIN tareas AS t;

SELECT nombre,t.domicilio,descripcion
FROM guardias AS g
CROSS JOIN tareas AS t
WHERE (g.sexo='f' AND t.descripcion='vigilancia interior') OR (g.sexo='m' AND t.
descripcion='vigilancia exterior');

/** EJERCICIO 69
*****/

DROP TABLE IF EXISTS clientes;

CREATE TABLE clientes (
    nombre VARCHAR(30),
    sexo CHAR(1),
    edad INT,
    domicilio VARCHAR(30)
);

INSERT INTO clientes
VALUES ('Maria Lopez','f',45,'Colon 123'),
      ('Liliana Garcia','f',35,'Sucre 456'),

```

```
( 'Susana Lopez', 'f', 41, 'Avellaneda 98' ),
( 'Juan Torres', 'm', 44, 'Sarmiento 755' ),
( 'Marcelo Oliva', 'm', 56, 'San Martin 874' ),
( 'Federico Pereyra', 'm', 38, 'Colon 234' ),
( 'Juan Garcia', 'm', 50, 'Peru 333' );
```

```
SELECT cm.nombre, cm.edad, cv.nombre, cv.edad
FROM clientes AS cm
CROSS JOIN clientes cv
WHERE cm.sexo='f' AND cv.sexo='m';
```

```
SELECT cm.nombre, cm.edad, cv.nombre, cv.edad
FROM clientes AS cm
JOIN clientes cv
ON cm.nombre <> cv.nombre
WHERE cm.sexo='f' AND cv.sexo='m';
```

```
SELECT cm.nombre, cm.edad, cv.nombre, cv.edad
FROM clientes AS cm
CROSS JOIN clientes cv
WHERE cm.sexo='f' AND cv.sexo='m' AND cm.edad-cv.edad BETWEEN -5 AND 5;
```

```
/** EJERCICIO 70
*****/
```

```
DROP TABLE IF EXISTS equipos;
```

```
CREATE TABLE equipos (
nombre VARCHAR(30),
barrio VARCHAR(20),
domicilio VARCHAR(30),
entrenador VARCHAR(30)
);
```

```
INSERT INTO equipos
VALUES ( 'Los tigres', 'Gral. Paz', 'Sarmiento 234', 'Juan Lopez' ),
( 'Los leones', 'Centro', 'Colon 123', 'Gustavo Fuentes' ),
( 'Campeones', 'Pueyrredon', 'Guemes 346', 'Carlos Moreno' ),
( 'Cebollitas', 'Alberdi', 'Colon 1234', 'Luis Duarte' );
```

```
SELECT e1.nombre, e2.nombre, e1.barrio AS sede
FROM equipos AS e1
CROSS JOIN equipos AS e2
WHERE e1.nombre <> e2.nombre;
```

```
SELECT e1.nombre, e2.nombre, e1.barrio AS sede
FROM equipos AS e1
JOIN equipos AS e2
ON e1.nombre <> e2.nombre;
```

```
SELECT e1.nombre, e2.nombre, e1.barrio AS sede
FROM equipos AS e1
CROSS JOIN equipos AS e2
WHERE e1.nombre > e2.nombre;
```

```
/** EJERCICIO 71
***** /

DROP TABLE IF EXISTS visitantes;

CREATE TABLE visitantes (
    nombre VARCHAR(30),
    edad SMALLINT,
    sexo CHAR(1) DEFAULT 'f',
    domicilio VARCHAR(30),
    codigociudad SMALLINT NOT NULL,
    mail VARCHAR(30),
    montocompra DECIMAL (6,2)
);

DROP TABLE IF EXISTS ciudades;

CREATE TABLE ciudades (
    codigo SERIAL,
    nombre VARCHAR(20),
    PRIMARY KEY(codigo)
);

INSERT INTO ciudades(nombre)
VALUES ('Cordoba'),
('Carlos Paz'),
('La Falda'),
('Cruz del Eje');

INSERT INTO visitantes
VALUES ('Susana Molina', 35, 'f', 'Colon 123', 1, NULL, 59.80),
('Marcos Torres', 29, 'm', 'Sucre 56', 1, 'marcostorres@hotmail.com', 150.50),
('Mariana Juarez', 45, 'f', 'San Martin 111', 2, NULL, 23.90),
('Fabian Perez', 36, 'm', 'Avellaneda 213', 3, 'fabianperez@xaxamail.com', 0),
('Alejandra Garcia', 28, 'f', NULL, 2, NULL, 280.50),
('Gaston Perez', 29, 'm', NULL, 5, 'gastonperez1@gmail.com', 95.40),
('Mariana Juarez', 33, 'f', NULL, 2, NULL, 90);

SELECT c.nombre,
COUNT(*) AS cantidad
FROM ciudades AS c
JOIN visitantes AS v
ON codigociudad=c.codigo
GROUP BY c.nombre;

SELECT c.nombre, sexo,
AVG(montocompra) AS promediodecompra
FROM ciudades AS c
JOIN visitantes AS v
ON codigociudad=c.codigo
GROUP BY c.nombre, sexo;

SELECT c.nombre,
COUNT(mail) AS tienenmail
FROM ciudades AS c
JOIN visitantes AS v
ON codigociudad=c.codigo
```

```

GROUP BY c.nombre;

SELECT c.nombre,
       MAX(montocompra)
FROM visitantes AS v
JOIN ciudades AS c
ON codigociudad=c.codigo
GROUP BY c.nombre;

/** EJERCICIO 72
*****/

DROP TABLE IF EXISTS socios;

CREATE TABLE socios (
    documento CHAR(8) NOT NULL,
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
    PRIMARY KEY(documento)
);

DROP TABLE IF EXISTS deportes;

CREATE TABLE deportes (
    codigo SERIAL,
    nombre VARCHAR(20),
    profesor VARCHAR(15),
    PRIMARY KEY(codigo)
);

DROP TABLE IF EXISTS inscritos;

CREATE TABLE inscritos (
    documento CHAR(8) NOT NULL,
    codigodeporte SMALLINT NOT NULL,
    anio CHAR(4),
    matricula CHAR(1),--'s'=paga, 'n'=impaga
    PRIMARY KEY(documento,codigodeporte,anio)
);

INSERT INTO socios
VALUES ('22222222','Ana Acosta','Avellaneda 111'),
      ('23333333','Betina Bustos','Bulnes 222'),
      ('24444444','Carlos Castro','Caseros 333'),
      ('25555555','Daniel Duarte','Dinamarca 44');

INSERT INTO deportes(nombre,profesor)
VALUES ('basquet','Juan Juarez'),
      ('futbol','Pedro Perez'),
      ('natacion','Marina Morales'),
      ('tenis','Marina Morales');

INSERT INTO inscritos
VALUES ('22222222',3,'2006','s'),
      ('23333333',3,'2006','s'),
      ('24444444',3,'2006','n'),
      ('22222222',3,'2005','s'),
      ('22222222',3,'2007','n'),
      ('24444444',1,'2006','s'),

```

```
( '24444444',2,'2006','s'),
( '26666666',0,'2006','s');
```

```
SELECT s.nombre,d.nombre,anio
FROM deportes AS d
RIGHT JOIN inscritos AS i
ON codigodeporte=d.codigo
LEFT JOIN socios AS s
ON i.documento=s.documento;
```

```
SELECT s.nombre,d.nombre,anio,matricula
FROM deportes AS d
FULL JOIN inscritos AS i
ON codigodeporte=d.codigo
FULL JOIN socios AS s
ON s.documento=i.documento;
```

```
SELECT s.nombre,d.nombre,anio,matricula
FROM deportes AS d
JOIN inscritos AS i
ON codigodeporte=d.codigo
JOIN socios AS s
ON s.documento=i.documento
WHERE s.documento='22222222';
```

```
/** EJERCICIO 73
```

```
*****/
```

```
DROP TABLE IF EXISTS clientes;
```

```
CREATE TABLE clientes (
    codigo SERIAL,
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
    ciudad VARCHAR(20),
    codigoprovincia SMALLINT,
    PRIMARY KEY(codigo)
);
```

```
DROP TABLE IF EXISTS provincias;
```

```
CREATE TABLE provincias (
    codigo SERIAL,
    nombre VARCHAR(20)
);
```

```
ALTER TABLE clientes
ADD CONSTRAINT FK_clientes_codigoprovincia
FOREIGN KEY (codigoprovincia)
REFERENCES provincias(codigo);
```

```
ALTER TABLE provincias
ADD CONSTRAINT PK_provincias_codigo
PRIMARY KEY (codigo);
```

```
INSERT INTO provincias VALUES (1,'Cordoba');
INSERT INTO provincias VALUES (2,'Santa Fe');
```

```
INSERT INTO provincias VALUES (3,'Misiones');
INSERT INTO provincias VALUES (4,'Rio Negro');

INSERT INTO clientes(nombre,domicilio,ciudad,codigoprovincia)
VALUES ('Perez Juan','San Martin 123','Carlos Paz',1),
('Moreno Marcos','Colon 234','Rosario',2),
('Acosta Ana','Avellaneda 333','Posadas',3),
('Luisa Lopez','Juarez 555','La Plata',6);

ALTER TABLE clientes
ADD CONSTRAINT FK_clientes_codigoprovincia
FOREIGN KEY (codigoprovincia)
REFERENCES provincias(codigo);

DELETE FROM clientes WHERE codigoprovincia=6;

ALTER TABLE clientes
ADD CONSTRAINT FK_clientes_codigoprovincia
FOREIGN KEY (codigoprovincia)
REFERENCES provincias(codigo);

INSERT INTO clientes(nombre,domicilio,ciudad,codigoprovincia)
VALUES ('Garcia Marcos','Colon 877','Lules',9);

DELETE FROM provincias WHERE codigo=3;

DELETE FROM provincias WHERE codigo=4;

UPDATE provincias SET codigo=7 WHERE codigo=1;

/** EJERCICIO 74
*****/

DROP TABLE IF EXISTS clientes;

CREATE TABLE clientes (
codigo INT,
nombre VARCHAR(30),
domicilio VARCHAR(30),
ciudad VARCHAR(20),
referenciadopor INT,
PRIMARY KEY(codigo)
);

INSERT INTO clientes
VALUES (50,'Juan Perez','Sucre 123','Cordoba',NULL),
(90,'Marta Juarez','Colon 345','Carlos Paz',NULL),
(110,'Fabian Torres','San Martin 987','Cordoba',50),
(125,'Susana Garcia','Colon 122','Carlos Paz',90),
(140,'Ana Herrero','Colon 890','Carlos Paz',9);

ALTER TABLE clientes
ADD CONSTRAINT FK_clientes_referenciadopor FOREIGN KEY (referenciadopor)
REFERENCES clientes (codigo);

UPDATE clientes SET referenciadopor=90 WHERE referenciadopor=9;
```

```
ALTER TABLE clientes
  ADD CONSTRAINT FK_clientes_referenciadopor FOREIGN KEY (referenciadopor)
  REFERENCES clientes (codigo);

INSERT INTO clientes VALUES (150,'Karina Gomez','Caseros 444','Cruz del Eje',8);

UPDATE clientes SET codigo=180 WHERE codigo=90;

DELETE FROM clientes WHERE nombre='Marta Juarez';

UPDATE clientes SET codigo=180 WHERE codigo=125;

DELETE FROM clientes WHERE codigo=110;

/** EJERCICIO 75
*****/

DROP TABLE IF EXISTS clientes;

CREATE TABLE clientes (
  codigo SERIAL,
  nombre VARCHAR(30),
  domicilio VARCHAR(30),
  ciudad VARCHAR(20),
  codigoprovincia SMALLINT,
  PRIMARY KEY(codigo)
);

DROP TABLE IF EXISTS provincias;

CREATE TABLE provincias (
  codigo SERIAL,
  nombre VARCHAR(20),
  PRIMARY KEY(codigo)
);

INSERT INTO provincias
VALUES (1,'Cordoba'),
      (2,'Santa Fe'),
      (3,'Misiones'),
      (4,'Rio Negro');

INSERT INTO clientes(nombre,domicilio,ciudad,codigoprovincia)
VALUES ('Perez Juan','San Martin 123','Carlos Paz',1),
      ('Moreno Marcos','Colon 234','Rosario',2),
      ('Acosta Ana','Avellaneda 333','Posadas',3);

ALTER TABLE clientes
  ADD CONSTRAINT FK_clientes_codigoprovincia
  FOREIGN KEY (codigoprovincia)
  REFERENCES provincias(codigo)
  ON UPDATE casCade
  ON DELETE no action;

DELETE FROM provincias WHERE codigo=3;
```

```
UPDATE provincias SET codigo=9 WHERE codigo=3;
```

```
SELECT *FROM provincias;
```

```
SELECT *FROM clientes;
```

```
ALTER TABLE clientes
  ADD CONSTRAINT FK_clientes_codigoprovincia
  FOREIGN KEY (codigoprovincia)
  REFERENCES provincias(codigo)
  ON UPDATE casCade
  ON DELETE casCade;
```

```
DROP TABLE provincias;
```

```
/** EJERCICIO 76
```

```
*****/
```

```
DROP TABLE IF EXISTS proveedores;
```

```
CREATE TABLE proveedores (
  codigo SERIAL,
  nombre VARCHAR (30),
  domicilio VARCHAR(30),
  PRIMARY KEY(codigo)
);
```

```
DROP TABLE IF EXISTS clientes;
```

```
CREATE TABLE clientes (
  codigo SERIAL,
  nombre VARCHAR (30),
  domicilio VARCHAR(30),
  PRIMARY KEY(codigo)
);
```

```
DROP TABLE IF EXISTS empleados;
```

```
CREATE TABLE empleados (
  documento CHAR(8) NOT NULL,
  nombre VARCHAR(20),
  apellido VARCHAR(20),
  domicilio VARCHAR(30),
  PRIMARY KEY(documento)
);
```

```
INSERT INTO proveedores(nombre,domicilio)
VALUES ('Bebida cola','Colon 123'),
('Carnes Unica','Caseros 222'),
('Lacteos Blanca','San Martin 987'),
('Supermercado Lopez','Avellaneda 34'),
('Almacen Anita','Colon 987'),
('Garcia Juan','Sucre 345'),
('23333333','Federico','Lopez','Colon 987'),
('28888888','Ana','Marquez','Sucre 333'),
('30111111','Luis','Perez','Caseros 956');
```



```

SELECT nombre, domicilio FROM proveedores
UNION
SELECT nombre, domicilio FROM clientes
UNION
SELECT (apellido||' '||nombre), domicilio FROM empleados;

SELECT nombre, domicilio, 'proveedor' AS categoria FROM proveedores
UNION
SELECT nombre, domicilio, 'cliente' FROM clientes
UNION
SELECT (apellido||' '||nombre), domicilio , 'empleado' FROM empleados
ORDER BY categoria;

/** EJERCICIO 77
*****/

DROP TABLE IF EXISTS alumnos;

CREATE TABLE alumnos (
    documento CHAR(8),
    nombre VARCHAR(30),
    nota DECIMAL(4,2),
    PRIMARY KEY(documento)
);

INSERT INTO alumnos
VALUES ('30111111','Ana Algarbe',5.1),
('30222222','Bernardo Bustamante',3.2),
('30333333','Carolina Conte',4.5),
('30444444','Diana DoMINGuez',9.7),
('30555555','Fabian Fuentes',8.5),
('30666666','Gaston Gonzalez',9.70);

SELECT alumnos.*
FROM alumnos
WHERE nota=
(SELECT MAX(nota) FROM alumnos);

SELECT titulo,autor, precio
FROM libros
WHERE precio=
(SELECT titulo, MAX(precio) FROM libros);

SELECT alumnos.*,
(SELECT AVG(nota) FROM alumnos)-nota AS diferencia
FROM alumnos
WHERE nota <
(SELECT AVG(nota) FROM alumnos);

UPDATE alumnos SET nota=4
WHERE nota=
(SELECT MIN(nota) FROM alumnos);

DELETE FROM alumnos
WHERE nota <

```

```
(SELECT AVG(nota) FROM alumnos);

/** EJERCICIO 78
*****/

DROP TABLE IF EXISTS ciudades;

CREATE TABLE ciudades (
    codigo SERIAL,
    nombre VARCHAR(20),
    PRIMARY KEY (codigo)
);

DROP TABLE IF EXISTS clientes;

CREATE TABLE clientes (
    codigo SERIAL,
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
    codigociudad SMALLINT NOT NULL,
    PRIMARY KEY(codigo)
);

INSERT INTO ciudades (nombre)
VALUES ('Cordoba'),
('Cruz del Eje'),
('Carlos Paz'),
('La Falda'),
('Villa Maria');

INSERT INTO clientes(nombre,domicilio,codigociudad)
VALUES ('Lopez Marcos','Colon 111',1),
('Lopez Hector','San Martin 222',1),
('Perez Ana','San Martin 333',2),
('Garcia Juan','Rivadavia 444',3),
('Perez Luis','Sarmiento 555',3),
('Gomez Ines','San Martin 666',4),
('Torres Fabiola','Alem 777',5),
('Garcia Luis','Sucre 888',5);

SELECT nombre
FROM ciudades
WHERE codigo in
(SELECT codigociudad
FROM clientes
WHERE domicilio like 'San Martin %');

SELECT distinct ci.nombre
FROM ciudades AS ci
JOIN clientes AS cl
ON codigociudad=ci.codigo
WHERE domicilio like 'San Martin%';

SELECT nombre
FROM ciudades
WHERE codigo NOT in
```

```
(SELECT codigociudad
FROM clientes
WHERE nombre like 'G%');

SELECT codigociudad
FROM clientes
WHERE nombre like 'G%';

/** EJERCICIO 79
***** /

DROP TABLE IF EXISTS socios;

CREATE TABLE socios (
    numero SERIAL,
    documento CHAR(8),
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
    PRIMARY KEY (numero)
);

DROP TABLE IF EXISTS inscritos;

CREATE TABLE inscritos (
    numerosocio INT NOT NULL,
    deporte VARCHAR(20) NOT NULL,
    cuotas SMALLINT,
    PRIMARY KEY(numerosocio,deporte)
);

INSERT INTO socios(documento,nombre,domicilio)
VALUES ('23333333','Alberto Paredes','Colon 111'),
('24444444','Carlos Conte','Sarmiento 755'),
('25555555','Fabian Fuentes','Caseros 987'),
('26666666','Hector Lopez','Sucre 344');

INSERT INTO inscritos
VALUES (1,'tenis',1),
(1,'basquet',2),
(1,'natacion',1),
(2,'tenis',9),
(2,'natacion',1),
(2,'basquet',DEFAULT),
(2,'futbol',2),
(3,'tenis',8),
(3,'basquet',9),
(3,'natacion',0),
(4,'basquet',10);

SELECT numero,nombre,deporte
FROM socios AS s
JOIN inscritos AS i
ON numerosocio=numero;

SELECT nombre
FROM socios
```

```

JOIN inscritos AS i
ON numero=numerosocio
WHERE deporte='natacion' AND numero= any
(SELECT numerosocio
FROM inscritos AS i
WHERE deporte='tenis');

```

```

SELECT deporte
FROM inscritos AS i
WHERE numerosocio=1 AND deporte= any
(SELECT deporte
FROM inscritos AS i
WHERE numerosocio=2);

```

```

SELECT i1.deporte
FROM inscritos AS i1
JOIN inscritos AS i2
ON i1.deporte=i2.deporte
WHERE i1.numerosocio=1 AND i2.numerosocio=2;

```

```

SELECT deporte
FROM inscritos AS i
WHERE numerosocio=2 AND cuotas > any
(SELECT cuotas
FROM inscritos
WHERE numerosocio=1);

```

```

SELECT deporte
FROM inscritos AS i
WHERE numerosocio=2 AND cuotas > all
(SELECT cuotas
FROM inscritos
WHERE numerosocio=1);

```

```

DELETE FROM inscritos
WHERE numerosocio=any
(SELECT numerosocio
FROM inscritos
WHERE cuotas=0);

```

```

/** EJERCICIO 80

```

```

*****/

```

```

DROP TABLE IF EXISTS socios;

```

```

CREATE TABLE socios (
numero SERIAL,
documento CHAR(8),
nombre VARCHAR(30),
domicilio VARCHAR(30),
PRIMARY KEY (numero)
);

```

```

DROP TABLE IF EXISTS inscritos;

```

```

CREATE TABLE inscritos (
numerosocio INT NOT NULL,

```

```

deporte VARCHAR(20) NOT NULL,
cuotas SMALLINT,
PRIMARY KEY(numerosocio,deporte)
);

INSERT INTO socios(documento,nombre,domicilio)
VALUES ('23333333','Alberto Paredes','Colon 111'),
('24444444','Carlos Conte','Sarmiento 755'),
('25555555','Fabian Fuentes','Caseros 987'),
('26666666','Hector Lopez','Sucre 344');

INSERT INTO inscritos
VALUES (1,'tenis',1),
(1,'basquet',2),
(1,'natacion',1),
(2,'tenis',9),
(2,'natacion',1),
(2,'basquet',DEFAULT),
(2,'futbol',2),
(3,'tenis',8),
(3,'basquet',9),
(3,'natacion',0),
(4,'basquet',10);

SELECT nombre,domicilio,
(SELECT COUNT(*)
FROM inscritos AS i
WHERE s.numero=i.numerosocio) AS deportes
FROM socios AS s;

SELECT nombre,
(SELECT (COUNT(*)*10)
FROM inscritos AS i
WHERE s.numero=i.numerosocio) AS total,
(SELECT SUM(i.cuotas)
FROM inscritos AS i
WHERE s.numero=i.numerosocio) AS pagas
FROM socios AS s;

SELECT nombre,
COUNT(i.deporte)*10 AS total,
SUM(i.cuotas) AS pagas
FROM socios AS s
JOIN inscritos AS i
ON numero=numerosocio
GROUP BY nombre;

/** EJERCICIO 81
*****/

DROP TABLE IF EXISTS socios;

CREATE TABLE socios (
numero SERIAL,
documento CHAR(8),
nombre VARCHAR(30),

```

```

domicilio VARCHAR(30),
PRIMARY KEY (numero)
);

DROP TABLE IF EXISTS inscritos;

CREATE TABLE inscritos (
    numerosocio INT NOT NULL,
    deporte VARCHAR(20) NOT NULL,
    cuotas SMALLINT,
    PRIMARY KEY(numerosocio,deporte)
);

INSERT INTO socios(documento,nombre,domicilio)
VALUES ('23333333','Alberto Paredes','Colon 111'),
('24444444','Carlos Conte','Sarmiento 755'),
('25555555','Fabian Fuentes','Caseros 987'),
('26666666','Hector Lopez','Sucre 344');
```

```

INSERT INTO inscritos
VALUES (1,'tenis',1),
(1,'basquet',2),
(1,'natacion',1),
(2,'tenis',9),
(2,'natacion',1),
(2,'basquet',DEFAULT),
(2,'futbol',2),
(3,'tenis',8),
(3,'basquet',9),
(3,'natacion',0),
(4,'basquet',10);
```

```

SELECT nombre
FROM socios AS s
WHERE exists
(SELECT *FROM inscritos AS i
    WHERE s.numero=i.numerosocio
    AND i.deporte='natacion');
```

```

SELECT nombre
FROM socios AS s
WHERE NOT exists
(SELECT *FROM inscritos AS i
    WHERE s.numero=i.numerosocio
    AND i.deporte='natacion');
```

```

SELECT s.*
FROM socios AS s
WHERE exists
(SELECT *FROM inscritos AS i
    WHERE s.numero=i.numerosocio
    AND i.cuotas=10);
```

```

/** EJERCICIO 82
```

```

***** /
```

```
DROP TABLE IF EXISTS deportes;
```

```
CREATE TABLE deportes (
    nombre VARCHAR(15),
    profesor VARCHAR(30),
    dia VARCHAR(10),
    cuota DECIMAL(5,2)
);
```

```
INSERT INTO deportes
VALUES ('tenis','Ana Lopez','lunes',20),
      ('natacion','Ana Lopez','martes',15),
      ('futbol','Carlos Fuentes','miercoles',10),
      ('basquet','Gaston Garcia','jueves',15),
      ('padle','Juan Huerta','lunes',15),
      ('handball','Juan Huerta','martes',10);
```

```
SELECT distinct d1.profesor      FROM deportes AS d1
WHERE d1.profesor in
(SELECT d2.profesor      FROM deportes AS d2
 WHERE d1.nombre <> d2.nombre);
```

```
SELECT distinct d1.profesor      FROM deportes AS d1
JOIN deportes AS d2
ON d1.profesor=d2.profesor WHERE d1.nombre <> d2.nombre;
```

```
SELECT nombre
FROM deportes
WHERE nombre <> 'natacion' AND dia =
(SELECT dia
 FROM deportes
 WHERE nombre='natacion');
```

```
SELECT d1.nombre
FROM deportes AS d1
JOIN deportes AS d2
ON d1.dia=d2.dia
WHERE d2.nombre='natacion' AND d1.nombre <> d2.nombre;
```

```
/** EJERCICIO 83
```

```
*****/
```

```
DROP TABLE IF EXISTS socios;
```

```
CREATE TABLE socios (
    documento CHAR(8) NOT NULL,
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
    PRIMARY KEY(documento)
);
```

```
DROP TABLE IF EXISTS deportes;
```

```
CREATE TABLE deportes (
    codigo SERIAL,
    nombre VARCHAR(20),
```

```

profesor VARCHAR(15),
PRIMARY KEY(codigo)
);

DROP TABLE IF EXISTS inscritos;

CREATE TABLE inscritos (
    documento CHAR(8) NOT NULL,
    codigodeporte SMALLINT NOT NULL,
    anio CHAR(4),
    matricula CHAR(1),--'s'=paga, 'n'=impaga
    PRIMARY KEY(documento,codigodeporte,anio)
);

INSERT INTO socios
VALUES ('22222222','Ana Acosta','Avellaneda 111'),
      ('23333333','Betina Bustos','Bulnes 222'),
      ('24444444','Carlos Castro','Caseros 333'),
      ('25555555','Daniel Duarte','Dinamarca 44');

INSERT INTO deportes(nombre,profesor)
VALUES ('basquet','Juan Juarez'),
      ('futbol','Pedro Perez'),
      ('natacion','Marina Morales'),
      ('tenis','Marina Morales');

INSERT INTO inscritos
VALUES ('22222222',3,'2006','s'),
      ('23333333',3,'2006','s'),
      ('24444444',3,'2006','n'),
      ('22222222',3,'2005','s'),
      ('22222222',3,'2007','n'),
      ('24444444',1,'2006','s'),
      ('24444444',2,'2006','s');

SELECT i.documento,i.codigodeporte,d.nombre AS deporte, anio, matricula, d.profesor
FROM deportes AS d
JOIN inscritos AS i
ON d.codigo=i.codigodeporte;

SELECT s.nombre,td.deporte,td.profesor,td.anio,td.matricula
FROM socios AS s
JOIN (SELECT i.documento,i.codigodeporte,d.nombre AS deporte, anio, matricula, d.profesor
FROM deportes AS d
JOIN inscritos AS i
ON d.codigo=i.codigodeporte) AS td
ON td.documento=s.documento;

/** EJERCICIO 84
*****/

DROP TABLE IF EXISTS socios;

CREATE TABLE socios (
    numero SERIAL,

```



```

documento CHAR(8),
nombre VARCHAR(30),
domicilio VARCHAR(30),
PRIMARY KEY (numero)
);

DROP TABLE IF EXISTS inscritos;

CREATE TABLE inscritos (
    numerosocio INT NOT NULL,
    deporte VARCHAR(20) NOT NULL,
    matricula CHAR(1), -- 'n' o 's'
    PRIMARY KEY(numerosocio,deporte)
);

INSERT INTO socios(documento,nombre,domicilio)
VALUES ('23333333','Alberto Paredes','Colon 111'),
('24444444','Carlos Conte','Sarmiento 755'),
('25555555','Fabian Fuentes','Caseros 987'),
('26666666','Hector Lopez','Sucre 344');
```

```

INSERT INTO inscritos
VALUES (1,'tenis','s'),
(1,'basquet','s'),
(1,'natacion','s'),
(2,'tenis','s'),
(2,'natacion','s'),
(2,'basquet','n'),
(2,'futbol','n'),
(3,'tenis','s'),
(3,'basquet','s'),
(3,'natacion','n'),
(4,'basquet','n');
```

```

UPDATE inscritos SET matricula='s'
WHERE numerosocio=
(SELECT numero
FROM socios
WHERE documento='25555555');
```

```

DELETE FROM inscritos
WHERE numerosocio in
(SELECT numero
FROM socios AS s
JOIN inscritos
ON numerosocio=numero
WHERE matricula='n');
```

```

/** EJERCICIO 85
```

```

*****/
```

```

DROP TABLE IF EXISTS clientes;
```

```

CREATE TABLE clientes (
    codigo SERIAL,
    nombre VARCHAR(30),
    domicilio VARCHAR(30),
```

```

PRIMARY KEY(codigo)
);
DROP TABLE IF EXISTS facturas;

CREATE TABLE facturas (
    numero INT NOT NULL,
    fecha DATE,
    codigocliente INT NOT NULL,
    total DECIMAL(6,2),
    PRIMARY KEY(numero)
);
INSERT INTO clientes(nombre,domicilio)
VALUES ('Juan Lopez','Colon 123'),
('Luis Torres','Sucre 987'),
('Ana Garcia','Sarmiento 576'),
('Susana Molina','San Martin 555');
INSERT INTO facturas
VALUES (1200,'2007-01-15',1,300),
(1201,'2007-01-15',2,550),
(1202,'2007-01-15',3,150),
(1300,'2007-01-20',1,350),
(1310,'2007-01-22',3,100);
DROP TABLE IF EXISTS clientespref;

CREATE TABLE clientespref (
    nombre VARCHAR(30),
    domicilio VARCHAR(30)
);

INSERT INTO clientespref
SELECT nombre,domicilio
FROM clientes
WHERE codigo in
(SELECT codigocliente
FROM clientes AS c
JOIN facturas AS f
ON codigocliente=codigo
GROUP BY codigocliente
HAVING SUM(total) > 500);

SELECT *FROM clientespref;

```

/** EJERCICIO 86

*****/

```

DROP TABLE IF EXISTS socios;

```

```

CREATE TABLE socios (
    documento CHAR(8) NOT NULL,
    nombre VARCHAR(40),
    domicilio VARCHAR(30),
    PRIMARY KEY (documento)
);

```

```

DROP TABLE IF EXISTS profesores;

```

```
CREATE TABLE profesores (
    documento CHAR(8) NOT NULL,
    nombre VARCHAR(40),
    domicilio VARCHAR(30),
    PRIMARY KEY (documento)
);
DROP TABLE IF EXISTS cursos;
```

```
CREATE TABLE cursos (
    numero SERIAL,
    deporte VARCHAR(20),
    dia VARCHAR(15),
    documentoprofesor CHAR(8),
    PRIMARY KEY (numero)
);
DROP TABLE IF EXISTS inscritos;
```

```
CREATE TABLE inscritos (
    documentosocio CHAR(8) NOT NULL,
    numero SMALLINT NOT NULL,
    matricula CHAR(1),
    PRIMARY KEY (documentosocio,numero)
);
```

```
INSERT INTO socios
VALUES ('30000000','Fabian Fuentes','Caseros 987'),
      ('31111111','Gaston Garcia','Guemes 65'),
      ('32222222','Hector Huerta','Sucre 534'),
      ('33333333','Ines Irala','Bulnes 345');
```

```
INSERT INTO profesores
VALUES ('22222222','Ana Acosta','Avellaneda 231'),
      ('23333333','Carlos Caseres','Colon 245'),
      ('24444444','Daniel Duarte','Sarmiento 987'),
      ('25555555','Esteban Lopez','Sucre 1204');
```

```
INSERT INTO cursos(deporte,dia,documentoprofesor)
VALUES ('tenis','lunes','22222222'),
      ('tenis','martes','22222222'),
      ('natacion','miercoles','22222222'),
      ('natacion','jueves','23333333'),
      ('natacion','viernes','23333333'),
      ('futbol','sabado','24444444'),
      ('futbol','lunes','24444444'),
      ('basquet','martes','24444444');
```

```
INSERT INTO inscritos
VALUES ('30000000',1,'s'),
      ('30000000',3,'n'),
      ('30000000',6,NULL),
      ('31111111',1,'s'),
      ('31111111',4,'s'),
      ('32222222',8,'s');
```

```
CREATE view vista_club as
SELECT s.nombre AS socio,s.documento AS docsocio,s.domicilio AS domsocio,c.deporte,dia,
p.nombre AS profesor, matricula
```

```

FROM socios AS s
FULL JOIN inscritos AS i
ON s.documento=i.documentosocio
FULL JOIN cursos AS c
ON i.numero=c.numero
FULL JOIN profesores AS p
ON c.documentoprofesor=p.documento;

```

```
SELECT *FROM vista_club;
```

```

SELECT deporte,dia,COUNT(socio) AS cantidad
FROM vista_club
WHERE deporte is NOT NULL
GROUP BY deporte,dia
ORDER BY cantidad;

```

```

SELECT deporte,dia FROM vista_club
WHERE socio is NULL AND deporte is NOT NULL;

```

```

SELECT socio FROM vista_club
WHERE deporte is NULL AND socio is NOT NULL;

```

```
SELECT profesor FROM vista_club WHERE deporte is NULL AND profesor is NOT NULL;
```

```
SELECT socio, docsocio FROM vista_club WHERE deporte is NOT NULL AND matricula <> 's';
```

```

SELECT distinct profesor,dia
FROM vista_club WHERE profesor is NOT NULL;

```

```

SELECT distinct profesor,dia
FROM vista_club WHERE profesor is NOT NULL
ORDER BY dia;

```

```

SELECT socio FROM vista_club
WHERE deporte='tenis' AND dia='lunes';

```

```
/** EJERCICIO 87
```

```
*****/
```

```

CREATE SEQUENCE sec_legajoempleados
START WITH 100
MINVALUE 1
MAXVALUE 999
INCREMENT BY 2;

```

```
DROP TABLE IF EXISTS empleados;
```

```

CREATE TABLE empleados (
legajo bigint DEFAULT NEXTVAL('sec_legajoempleados'),
documento CHAR(8) NOT NULL,
nombre VARCHAR(30) NOT NULL,
PRIMARY KEY(legajo)
);

```

```

INSERT INTO empleados(documento,nombre)
VALUES ('22333444','Ana Acosta'),

```

```
( '23444555', 'Betina Bustamante'),
( '24555666', 'Carlos Caseros'),
( '25666777', 'Diana DoMINGuez'),
( '26777888', 'Estela Esper');
```

```
SELECT * FROM empleados;
```

```
SELECT * FROM sec_legajoempleados;
```

```
DROP SEQUENCE sec_legajoempleados;
```

```
/** EJERCICIO 88
```

```
*****/
```

```
CREATE SEQUENCE sec_codigoaulainscritos
START WITH 1
MINVALUE 1
MAXVALUE 2
CYCLE;
```

```
DROP TABLE IF EXISTS inscritos;
```

```
CREATE TABLE inscritos (
    documento CHAR(8) NOT NULL,
    nombre VARCHAR(30) NOT NULL,
    codigocurso INT DEFAULT NEXTVAL('sec_codigoaulainscritos'),
    PRIMARY KEY(documento)
);
INSERT INTO inscritos(documento,nombre)
VALUES ( '20000000', 'Rodriguez Pablo'),
( '30000000', 'Mercado Ana'),
( '40000000', 'Morello Luis'),
( '50000000', 'Prado Juan'),
( '60000000', 'Solis Maria');
```

```
SELECT * FROM inscritos WHERE codigocurso=1;
```

```
SELECT * FROM inscritos WHERE codigocurso=2;
```

```
/** EJERCICIO 89
```

```
*****/
```

```
DROP TABLE IF EXISTS medicamentos;
```

```
CREATE TABLE medicamentos (
    codigo SERIAL,
    nombre VARCHAR(20),
    laboratorio VARCHAR(20),
    precio DECIMAL(5,2),
    cantidad SMALLINT,
    PRIMARY KEY(codigo)
);
```

```
INSERT INTO medicamentos (nombre,laboratorio,precio,cantidad)
VALUES ( 'Sertal', 'Roche', 5.2, 100),
( 'Buscapina', 'Roche', 4.10, 200),
```

```

('Amoxidal 500','Bayer',15.60,100),
('Paracetamol 500','Bago',1.90,200),
('Bayaspirina','Bayer',2.10,150),
('Amoxidal jarabe','Bayer',5.10,250);

```

```

CREATE OR REPLACE FUNCTION preciopromedio() RETURNS DECIMAL
as
'SELECT AVG(precio) FROM medicamentos;'
LANGUAGE SQL;

```

```
SELECT preciopromedio();
```

```
SELECT * FROM medicamentos;
```

```
SELECT * FROM medicamentos WHERE precio > preciopromedio();
```

```
/** EJERCICIO 90
```

```
*****/
```

```
DROP TABLE IF EXISTS medicamentos;
```

```

CREATE TABLE medicamentos (
codigo SERIAL,
nombre VARCHAR(20),
laboratorio VARCHAR(20),
precio DECIMAL(5,2),
cantidad SMALLINT,
PRIMARY KEY(codigo)
);

```

```

INSERT INTO medicamentos (nombre,laboratorio,precio,cantidad)
VALUES ('Sertal','Roche',5.2,100),
('Buscapina','Roche',4.10,200),
('Amoxidal 500','Bayer',15.60,100),
('Paracetamol 500','Bago',1.90,200),
('Bayaspirina','Bayer',2.10,150),
('Amoxidal jarabe','Bayer',5.10,250);

```

```

CREATE OR REPLACE FUNCTION borrarmedicamento(INTEGER) RETURNS void
as
'DELETE FROM medicamentos WHERE codigo=$1;'
LANGUAGE SQL;

```

```
SELECT borrarmedicamento(1);
```

```
SELECT * FROM medicamentos;
```

```
/** EJERCICIO 91
```

```
*****/
```

```
DROP TABLE IF EXISTS medicamentos;
```

```

CREATE TABLE medicamentos (
codigo SERIAL,
nombre VARCHAR(20),

```

```
laboratorio VARCHAR(20),
precio DECIMAL(5,2),
cantidad SMALLINT,
PRIMARY KEY(codigo)
);

INSERT INTO medicamentos (nombre,laboratorio,precio,cantidad)
VALUES ('Sertal','Roche',5.2,100),
('Buscapina','Roche',4.10,200),
('Amoxidal 500','Bayer',15.60,100),
('Paracetamol 500','Bago',1.90,200),
('Bayaspirina','Bayer',2.10,150);

INSERT INTO medicamentos (nombre,laboratorio,precio,cantidad)
VALUES ('Amoxidal jarabe','Bayer',5.10,250);

CREATE OR REPLACE FUNCTION medicamentomasCaro() RETURNS medicamentos
as
'SELECT * FROM medicamentos WHERE precio=(SELECT MAX(precio) FROM medicamentos);'
LANGUAGE SQL;

SELECT medicamentomasCaro();
```