

Práctico Nº4: Lenguaje SQL

Resuelva los ejercicios planteados usando el lenguaje SQL. Utilizar el SGBDR PostgreSQL, para ello instalar la herramienta Pgadmin.

EJERCICIO 1: A continuación, se muestran las tablas que forman una base de datos.

PERS (Personas) = {**id**, **nom**, **idC** (identificador de ciudad), **edad**}

CIU(Ciudad) = {**ciudadId**, **ciudadNom**}

1. Crear las tablas teniendo en cuenta las siguientes especificaciones para la clave foránea: identificador de ciudad.

a.

Reglas de Integridad	Nulos	Eliminar	Modificar
Pers.idC	No	Restringir	Propagar

b.

Reglas de Integridad	Nulos	Eliminar	Modificar
Pers.idC	No	Nulificar	Restringir

2. Insertar ciudades con los siguientes datos:

< 1,'Estambul' >

< 2,'Roma' >

< 3,'Barcelona' >

< 4,'Praga' >

3. Insertar tuplas en la tabla Personas con los siguientes datos:

< 1,'Juan Manuel Ariza',1,16 >

< 2,'Santiago Manrique',2, 25>

< 3,'Luciano Perez',5, 50 >

< 4,'Andrea Hernández',1 >

< 5,'Ana Perez',1,30 >

4. Ejecutar las siguientes consultas:

- a. Muestre el nombre de las ciudades ordenadas alfabéticamente.
- b. Muestre las personas (todos los datos) mayores de edad.
- c. Muestre la cantidad de personas.
- d. Muestre la cantidad de ciudades.
- e. Eliminar la ciudad con identificador igual a 1.
- f. Muestre la cantidad de ciudades donde viven personas.

EJERCICIO 2: A continuación, se detallan diferentes casos sobre los cuales realizar las consultas planteadas.

Caso 1: Cursos de Verano Script

https://docs.google.com/document/d/1wlvcpTlfNND1_0gNhbmW9BV-jmuy_I_T/edit?usp=sharing&oid=116529533474267264229&rtpof=true&sd=true

PERS (Personas) = {Correo, NomU (nombre usuario), Nom}

CURSO (Cursos) = {Nom, ch (carga horaria)}

INSC (Inscribe) = {Correo, Nom, correod, nota}

DICTA = {Correo, Nom}

TEMAS= {Nom, tema}

ACTUALIZACIÓN

A. Inserte nuevas Personas con los siguientes datos:

1. < kf@gmail.com, kf, Katerin Falcon >
1. < mgh@gmail.com, mgh, Rosa González >
2. < rgh@gmail.com, rh, Rosa López >

B. Inserte el nuevo curso con los siguientes datos:

3. < Ruby, 40 >

C. Inserte los temas para el curso con los siguientes datos:

4. < Ruby, Estructura de Datos >
1. < Ruby, Estructura de Control >

D. Inserte la nueva tupla para DICTA con los siguientes datos:

5. < kf@gmail.com, Ruby >

E. Modifique la carga horaria del curso Ruby con el valor 60.

6. Elimine el curso Ruby.

CONSULTAS

1. Correo y nombre de todas las personas.
2. Nombre de los cursos.
3. Nombre del curso que tiene una carga horaria superior a la de todos los cursos que dicta "pedroibañez@yahoo.com.ar".
4. Datos (todos) de las personas (docentes o alumnos) que poseen como nombre Rosa (solamente o alguno de ellos).
5. Nombres de los cursos que tienen una carga horaria superior a la del curso "Kotlin I".
6. Cursos (todos los datos) cuya carga horaria sea superior a las 40 horas reloj.
7. Cursos (todos los datos) cuya carga horaria se encuentre entre 40 y 45 horas reloj.
8. Cursos (nombre renombrar como nombre curso) junto a los temas que se imparten.
9. Docentes (correo y nombre) que dictan cursos.
10. Cursos (nombre) junto a los datos del docente que los dicta.

11. Cursos (todos los datos) junto a los datos de los alumnos inscriptos en ellos. Se deben incluir todos los cursos registrados más allá que no tengan alumnos inscriptos.
12. Docentes (todos los datos) que dictan los cursos “Python I”.
13. Docentes (todos los datos) que dictan los cursos “Python II”.
14. Docentes (correo y nombre) que dictan el curso “Python I” y/o “Python II”.
15. Docentes (correo) que dictan los cursos “Python I” y “Python II”.
16. Docentes (todos los datos) que cursaron algún curso de verano.
17. Alumnos (todos los datos) que se inscribieron en el curso “Kotlin I”.
18. Alumnos (todos los datos) que se inscribieron en el curso “Kotlin II”.
19. Alumnos (correo) que se inscribieron tanto en el curso “Kotlin I” como “Kotlin II”.
20. Alumnos (todos los datos) que aprobaron el curso “Python I” y “Python II”.
21. Alumnos (Correo) que se inscribieron en más de un curso de verano.
22. Docentes (todos los datos) que dictan más de un curso cuya carga horaria sea inferior a 30 horas reloj.
23. Alumnos (correo) que cursaron los mismos cursos.
24. Alumnos que cursaron los mismos cursos con distinto profesor.
25. Alumnos (todos los datos) que se inscribieron en todos los cursos de verano.
26. Alumnos (todos los datos) que se inscribieron en todos los cursos que dicta el profesor con correo “pedroibañez@yahoo.com.ar”
27. Nombre del/los curso/s que tienen la mayor carga horaria.

TABLAS VIRTUALES/VISTAS

28. Especifique la Vista “Cursoscortos” que tenga los siguientes atributos nombre, carga horaria. Los cursos cortos son aquellos cuya carga horaria es inferior a las 40 horas.
29. Muestre los datos contenidos en la vista, ordenados según el nombre.
30. Inserte el curso “DBA PostgreSQL” con una carga horaria de 50 horas, a través de la vista.
31. Elimine la vista, y vuelva a crearla pero agregando la especificación “WITH CHECK OPTION”.
32. Inserte el curso “DBA Oracle” con una carga horaria de 55 horas, a través de la vista.
33. Especifique la Vista “Alumnos_python1” que tenga los siguientes atributos correo, nombre de usuario, nombre y representen a los alumnos que se inscribieron en el curso “PYTHON I”.
34. Muestre los datos contenidos en la vista creada en el punto anterior, cuyo correo sea una cuenta de Gmail.
35. Especifique la Vista “Alumnos_python2” que tenga los siguientes atributos nombre de usuario, nombre y representen a los alumnos que se inscribieron en el curso “PYTHON II”.
36. Muestre los datos contenidos en la vista.
37. Inserte un nuevo alumno con los siguientes datos: < orm@gmail.com, or, Orlando Martin >

GESTIÓN DE USUARIOS

38. Cree un usuario (alumno) con contraseña “alumno1”.
39. Cambie su contraseña, por “alumno”.
40. Concédale el permiso de SELECT e INSERT sobre la tabla CURSO.
41. Visualice los permisos del usuario “alumno”: `SELECT table_name, privilege_type FROM information_schema.table_privileges WHERE grantee = 'alumno';`

42. Accediendo con ese usuario (debe generar una nueva instancia que reference al mismo servidor, pero con el usuario “alumno”), ejecute un SELECT sobre la tabla CURSO y luego, sobre la tabla DICTA. Analice las respuestas.
43. Elimine el permiso SELECT sobre la tabla DICTA.
44. Visualice nuevamente los permisos del usuario “alumno”.

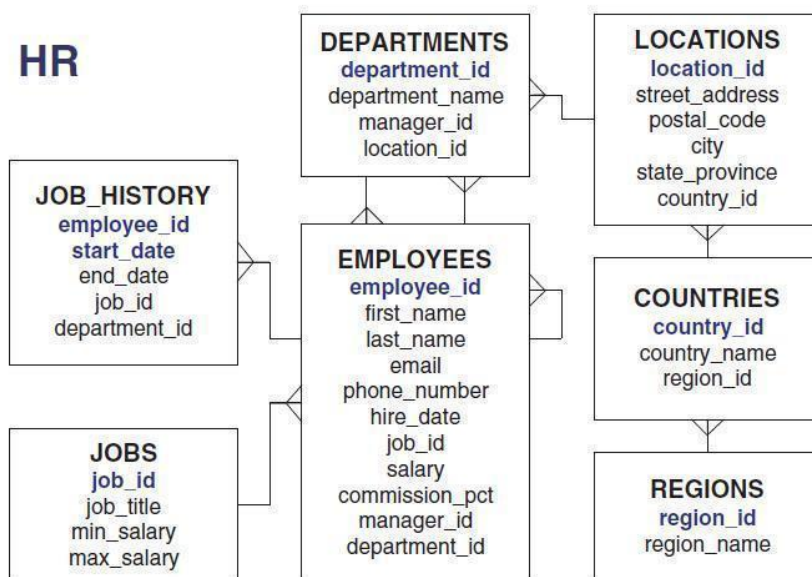
CONSULTAS AL CATÁLOGO

45. Muestre los datos de las bases de datos creadas (tabla pg_database del catálogo)
46. Muestre las tablas de la base de datos actual.
47. Muestre las columnas e índices de una tabla.
48. Muestra los usuarios conectados.
49. Muestre el tamaño que ocupa la tabla DICTA.

Caso 2 - ESQUEMA HR (RECURSOS HUMANOS)

Script: <https://drive.google.com/open?id=1NTEa6iW1UCWQtW55IZ3v52ulxbQ6egv>

A continuación, se muestra el esquema HR de Recursos Humanos.



1. Muestre el nombre (*first_name* y *last_name*) y el salario (*salary*) de todos los empleados. Cambie el nombre de la columna *first_name* a **Nombre** y *last_name* a **Apellido** y *salary* a **Salario**.
2. Muestre el nombre (*first_name* y *last_name*) y el salario (*salary*) de todos los empleados ordenado alfabéticamente por apellido (*last_name*).
3. Muestre el apellido (*last_name*) y la Comisión (*commission_pct*) que perciben los empleados cuya comisión sea mayor a 0.25. Mostrarlos ordenados en forma descendente por *last_name*.
4. Muestre la cantidad de empleados que trabajan en el departamento 100.
5. Muestre todos los datos de los departamentos con identificador 10 ó 70 (*department_id*).
6. Muestre el nombre (*last_name*) de los empleados junto al nombre del departamento (*department_name*) donde trabajan.

- Muestre el nombre (*last_name*) de los empleados de los departamentos de Finanzas (Finance) y/o Transporte (Shipping).
- Muestre sin repetir los tipos de trabajos (*jobs*) que realizan en los departamentos los empleados. Liste el identificador del departamento y el nombre del trabajo.
- Muestre los departamentos en los que los empleados realizan trabajos de Contabilidad (Stock Clerk) y Asistente de Administración (Shipping Clerk). Liste el identificador del departamento y el nombre del trabajo.
- Muestre el nombre de los departamentos que tienen al menos 3 empleados.
- Muestre el nombre de los empleados (*last_name*) y el salario (*salary*) de aquellos empleados que tienen un salario mayor que el salario promedio. Ordene el listado en forma descendente por salario.
- Muestre el salario máximo de los empleados del departamento 110. Ordene el listado en forma ascendente por salario.
- Muestre el nombre del/de los empleados que tienen el sueldo máximo del departamento 110. Ordene el listado en forma ascendente por salario
- Muestre los empleados (*first_name* y *last_name*) que no sean supervisores (o directores). NOTA: En la tabla departments el atributo *manager_id* tiene la identificación de los supervisores de cada dpto.
- Especifique la Vista EmpFinan que contenga todos los empleados del departamento 100 con los atributos *first_name* como nombre, *last_name* como apellido y *department_name* como nom_dpto.
- Realice una consulta que muestre la vista completa.
- Gestión de usuarios y privilegios: Crear los siguientes usuarios y otorgarles privilegios.

<i>Usuario</i>	<i>Privilegios</i>
HRDep	Permisos de SELECT, INSERT, UPDATE, ALTER sobre objetos del esquema.
HrUser	Permisos de SELECT sobre objetos del esquema.

Caso 3– OBRAS CIVILES

Script: <https://docs.google.com/document/d/1GQr83FXsIZ0Ggsr9Egq0ENAJj0ZCMxCVwSLFKX1j4HA/edit?usp=sharing>

El gobierno de la provincia ha programado la construcción de varias obras civiles, para administrar la información relativa a esta temática cuenta con una base de datos formada por las siguientes tablas:

Obras = {Co, Desc, Direc, Zona, EmpCon}

Materiales = {Cm, Desc, Precio}

Ferreterías = {Cuit, Nom, Direc, Zona, Tel}

Pedidos = {Co, Cm, Cuit, Fecha, Cant}

- Genere las tablas correspondientes, especificando claves primarias y foráneas.
- Inserte las tuplas en cada tabla creada.
- Muestre los materiales (descripción) pedidos el día 06/06/2010.
- Muestre para cada obra (indicando descripción) todos los materiales solicitados (descripción). Deben informarse todas las obras, más allá que aún no tenga materiales pedidos.
- Muestre la cantidad total de bolsas de cal que han sido pedidas a la ferretería MR S.A.
- Muestre la cantidad total de obras que han pedido materiales a la ferretería MR S.A.
- Muestre, para cada material pedido a alguna ferretería, el código de material, código de obra y la cantidad total pedida (independientemente de la ferretería).

8. Muestre la descripción de materiales pedidos para alguna obra en una cantidad promedio mayor a 320 unidades.
9. Muestre el nombre del material menos pedido (en cantidad total).
10. Muestre la descripción de las obras que no han utilizado pintura.
11. Muestre el nombre de las obras abastecidas totalmente por la ferretería MR S.A.
12. Muestre el nombre de los materiales que han sido pedidos para todas las obras realizadas.
13. Actualice el teléfono de la Ferretería San Ignacio por el número 4312548.
14. Elimine el Material con descripción Cemento Avellaneda.
15. Especifique la Vista “**ObrasCuyoNorte**” que contenga Co (código de la obra), Direc (dirección de la obra) y EmpCon (empresa constructora) de las obras ubicadas en la zona Cuyo Norte.
16. Especifique la Vista “**ObrasMat**” que contenga Obra (código de la obra), Empresa (empresa constructora), Material (descripción del material) y CantMat (cantidad de materiales pedidos).
 - . Muestre los datos contenidos en la vista, ordenados según obra (descendente) y material (ascendente).
 - a. Actualice la cantidad de materiales pedidos de las obras incrementándolas en 100.

Caso 4: Empresa de comida rápida

Script: https://docs.google.com/document/d/1Ki_TK6BS6Tvv0pHWL4DFHSW0_SJWT5d_j_WNGybaUu0/edit?usp=sharing

Se almacena en una base de datos, información referente a los productos que comercializan distintas sucursales de una empresa de comidas rápidas.

Sucursal = {num, direcc, tel, provincia}

Producto = {cod, num, nombre, precio, stock}

Ingrediente = {cod (código del producto), num, nombre, kcal (cantidad de calorías)}

Factura = {num (número de factura), fecha, calif(calificación otorgada)}

Detalle = {numf (número de factura), cod (código del producto), num (número de la sucursal), precio, cantidad}

1. Sucursales (todos los datos) que vendieron todos los productos que se comercializan en San Luis.
2. Provincias donde se venden más de un producto.
3. Sucursales (todos los datos) que venden tanto Pizza napolitana como Pizza napolitana especial.
4. Productos (nombre y stock) comercializados en las provincias de Cuyo (San Juan, Mendoza y San Luis).
5. Provincias que no venden Pizza muzzarella.
6. Sucursales (número y dirección) que no venden el producto ‘Pizza con jamón’.
7. Nombre de productos vendidos por todas las sucursales de San Juan.
8. Para cada sucursal de la provincia de San Juan, obtener número e importe total facturado.
9. Provincias donde se venden más de un producto.
10. Provincia/s que más facturó durante el 2020.

Caso 5: COVID 19

Script: https://docs.google.com/document/d/1B_ESPWxtIC6yS60mP920-zlUIX1knuXQV6QESZIGaLQ/edit?usp=sharing

Se almacena en una base de datos, información referente a las personas y las aplicaciones de las vacunas COVID19 en nuestro país.

Persona= {dni, pais, nombre, sexo, fechanac (*fecha de nacimiento*), provincia (*provincia donde reside*)}

Centro = {nombre, dirección, provincia}

Vacuna= {nombre, fechac (*fecha de creación*), cantidad (*cantidad de dosis*), laboratorio}

Aplica = {dni, pais, nombre (*nombre del centro*), nombrev (*nombre de la vacuna*), fecha, dnip (*dni del profesional*), paisp (*país del profesional*)}

1. Personas argentinas (dni, país y nombre) que se han aplicado más de una dosis.
2. Centros de salud (todos los datos) en los cuales se han aplicado la vacuna Sinopharm y/o Sputnik.
3. Personas (dni) que se han aplicado la vacuna Sinopharm pero no la Sputnik
4. Centros de salud (nombre y provincia) que han vacunado tanto a personas oriundas de San Juan como oriundas de Mendoza durante el 2020.
5. Profesionales (todos los datos) que han vacunado únicamente a personas argentinas.
6. Vacunas (todos los datos) que se han aplicado en todas las provincias de Cuyo (San Juan, Mendoza y San Luis).
7. Centros de salud (nombre) que han vacunado a más de 100 personas (aplicaciones).
8. Profesional (todos los datos) que realizó más vacunaciones (aplicaciones) durante el año 2020.

Caso 6: Inspectores

Script: <https://docs.google.com/document/d/1WNW3XLUry4Wm7YK5FGadXHeedAvIm8QybzlPJT-zrB0/edit?usp=sharing>

Se almacena en una base de datos, información referente a los inspectores y los comercios que tienen asignados en nuestra provincia.

Inspector= {cuil, nombre, tel, dep (*nombre del departamento donde vive*)}

Comercio = {cuit, dirección, tel, cuil (*cuil del inspector a cargo*), nomd (*nombre del departamento donde se ubica*)}

Controles = {cuil (*cuil del inspector*), cuit, fecha (*fecha en que se realizó la inspección*)}

1. Inspectores (todos los datos) que han inspeccionado comercios de Chimbas.
2. Inspectores (cuil) que han inspeccionado comercios en Rivadavia, pero no en Capital.
3. Listado que incluya los comercios (cuit, dirección y departamento) ubicados en los departamentos Santa Lucía, Rawson y Capital. Deben incluirse aquellos comercios que hayan sido controlados en alguna ocasión.
4. Comercios (todos los datos) que fueron controlados en más de una oportunidad por el inspector con dni 30400500.
5. Comercios (todos los datos) que fueron controlados en más de una oportunidad por el mismo inspector.

6. Comercios que no fueron controlados durante el 2022.
7. Inspectores (cuit y nombre) que han controlado todos los comercios de Capital.
8. Inspectores (todos los datos) que han controlado tanto comercios de Chimbas como de Capital.
9. Comercio (todos los datos) que fue más controlado durante el 2021.

EJERCICIOS PROPUESTOS:

Caso7- RECARGAS Y VIAJES DE LA TARJETA SUBE

Tomar datos del DataSet ID: 683552b8a83b3d9776ffc8cc4e4ad7b3

Tar (Tarjeta) = {num, monto, dni, nombre, prov}

LIN (Líneas) = {codlinea, cuit}

EMP (Empresas) = {cuit, nombre, direc, prov}

V (Viajes) = {num, fecha, hora, codlinea, empresa, importe}

Restricciones:

- **V** (Viajes): En esta tabla se almacenará qué viajes se registraron para un número de tarjeta en particular. Por supuesto, una persona puede viajar en la misma línea y empresa, pero viaja en una línea y empresa concreta para una fecha-hora.
1. Genere las tablas correspondientes, especificando claves primarias y foráneas.
 2. Inserte las tuplas en cada tabla creada.
 3. Personas (todos los datos) que solamente han tomado la línea 26A de la provincia de San Juan.
 4. Empresas (todos los datos) en las que viajaron personas oriundas de la provincia de San Juan y San Luis.
 5. Personas (dni y nombre) que viajaron más de una vez en una fecha, para una línea y empresa determinada.
 6. Personas (dni) que viajaron más de una vez en la línea 17 La positiva, el día 02/03/16.
 7. Obtener todos los datos de las personas que viajaron en todas las empresas de la provincia de San Juan (sin importar la línea).
 8. Obtener el importe total que gastó Juan Pérez cuyo dni es 36800700 en lo que del año 20018.
 9. Genera la Vista VIAJES_LINEA que contenga, para cada empresa (cuit y nombre) y para cada línea (código de línea) la cantidad de viajes que se hicieron en el año 2018.

Caso 8– CARTELERA DE ESPECTÁCULOS

Tomar datos del DataSet ID: fb79485591e887805a240134376ef48f

Pers (Persona) = {DNI, Tel, email}

Teatro = {Codt, Nom, Direc, Tel, Capac, prov}

Esp (Espectáculo) = {Cod, Nom, autor, tipo (concierto, danza, teatro, etc)}

Tiene = {Codt, Cod, Fechac, Fechaf, precio}

Compra = {DNI, Codt, Cod, fecha, precio}

1. Personas (dni y email) que han comprado entradas a espectáculos en la provincia de San Juan.
2. Obtener los espectáculos (conciertos) que estarán en cuyo (provincias de San Juan, Mendoza, San Luis y La Rioja).
3. Espectáculos (nombre y autor) que estarán en el teatro “La Quintanilla” (Mendoza) a partir de abril del presente año.

4. Teatros (Nombre y Dirección) que tienen únicamente espectáculos cuyo precio supera los \$500.
5. Espectáculos (todos los datos) que han estado en todos los teatros cuya capacidad no supera las 300 personas.
6. Personas (todos los datos) que han comprado más de una entrada para el mismo espectáculo (sin importar el teatro).
7. Personas (email) que han comprado más de una entrada al espectáculo “Un pianista brillante” para el “Teatro del Bicentenario”.
8. Personas (dni) que asistieron a todas las presentaciones de espectáculos, o sea, si un espectáculo se presentó en dos teatros, debe haber asistido a ambas presentaciones.
9. Personas (dni, nombre y apellido) que asistieron a todas las presentaciones en teatros de cuyo de espectáculos, o sea, si un espectáculo se presentó en dos teatros de la zona cuyo, debe haber asistido a ambas presentaciones.
10. Personas (todos los datos) que asistieron a todos los conciertos (es un tipo de espectáculo) presentados en teatros de cuyo, es decir, si un concierto se presentó en dos teatros de la zona cuyo, debe haber asistido a ambas presentaciones.

Caso 9– EMPRESAS DE COLECTIVOS

La provincia cuenta con una base de datos que posee información acerca de las distintas empresas de ómnibus que existen. Las tablas que la forman se plantean a continuación:

EMP (*Empresas de Omnibus*) = {Cuit, Nom, Dir, Tel}

CHOF (*Choferes*) = {Dni, Nom, Edad, Dom, celular, Patente, Turno (*M: mañana, T: tarde, N: nocturno*)}

LIN (*Líneas*) = {CodL, Descr, Cuit}

UNI (*Unidades*) = {Patente, Modelo, CodM, CodL}

MARCAS = {CodM, Nom}

CALLES = {CodC, Nom, Orient}

RECO (*Recorrido*) = {CodL, CodC, Orden, Cant_cuad (*cantidad de cuadras*)}

1. Crear las tablas correspondientes, especificando clave primaria y foránea.
2. Inserte las tuplas necesarias en cada relación creada para luego poder realizar las consultas que se le solicitan.
3. Mostrar todos los datos de las unidades que son manejadas por un solo chofer.
4. Mostrar el nombre de la empresa que posee mayor cantidad de unidades modelo 2009.
5. Mostrar el personal (dni y nombre) correspondiente a la empresa La Positiva.
6. Mostrar el nombre de las calles por las que circula alguna de las líneas de cada empresa de transporte, es decir, deben estar presentes todas las empresas.
7. Mostrar el nombre de las calles por las que circula más de una línea de colectivos.
8. Mostrar el nombre de las calles por las que circula una línea más de una vez en su recorrido.
9. Mostrar la descripción de la línea que tiene el recorrido más extenso.
10. Mostrar todos los datos de las empresas que poseen unidades de todas las marcas.
11. Mostrar el recorrido de la línea X (descripción de la línea) a partir de la calle Mitre.
12. Mostrar el nombre de las empresas que sólo poseen unidades marca Mercedes Benz.
13. Mostrar el nombre y orientación de las calles por las que ninguna línea pasa en su recorrido.
14. Sobre el esquema planteado enuncie las consultas en SQL para actualizarla:

- a. Actualice el celular del Chofer con “dni = 18923145” por el número 0264 4312548.
 - b. Actualice el nombre de la calle con código “calle10” por Hipólito Irigoyen.
 - c. Elimine la unidad (colectivo) con patente ABC 123 por darse de baja.
15. Sobre el esquema planteado especifique las siguientes vistas:
- a. Una Vista de “TurnoMañ” con los atributos siguientes: Dni, nombre del chofer y patente de la unidad que maneja.

Caso 10-INFORMACIÓN ESTELAR DE LA VÍA LÁCTEA

Tomar datos del DataSet ID: c21dc0ebdb674a8ae0a5cf0d9daaa787

Sean las siguientes relaciones de una base de datos que gestiona la información estelar de la vía láctea.

Pla (Planeta) = {**nomp**, **fechadesc** (fecha de descubrimiento), **cantal** (cantidad de años luz del sol)}

As (Astrónomo) = {**nomas**, **paisnac**, **fecha-nac**, **uni** (universidad)}

D(Descubrió) = {**nomp** (planeta), **nomas** (astrónomo)}

S (Satélite) = {**cods**, **nom** (nombre), **nomas** (nombre del astrónomo que lo bautizó), **fechadesc** (fecha de descubrimiento), **planeta** (nombre del planeta)}

Nota: Un planeta es un cuerpo celeste que gira alrededor del Sol. Un satélite es un cuerpo celeste más pequeño que gira alrededor de un planeta.

Restricciones:

- **D** (Descubrió): En esta relación (tabla) se almacena para cada planeta el/los astrónomos que lo descubrieron.
- **S** (satélite): En esta relación (tabla) se almacena el/los satélites pertenecientes a cada planeta y el astrónomo que lo bautizó.

1. Planetas (nombre, cantidad de años luz) descubiertos únicamente por la Universidad de Texas (es decir, todos los descubridores pertenecen a esa universidad).
2. Astrónomos (nombre y universidad) que han descubierto planetas y/o satélites durante el período 1950-2016.
3. Planetas (nombre, cantidad de años luz) junto al/a los Astrónomo/s (nombre) que lo descubrió o descubrieron.
4. Astrónomos (nombre y país de nacimiento) de la universidad de Antioquía que descubrieron tanto planetas como satélites.
5. Planetas (nombre y fecha de descubrimiento) que tienen más de un satélite.
6. Planetas (nombre y fecha de descubrimiento) que no tienen satélites.
7. Genere la vista Planetas_16 que contenga, nombre del planeta junto al nombre de sus satélites, de todos los planetas descubiertos en el año 2016.
8. ¿La vista es actualizable? Justifique.