

PRÁCTICO N° 4: LENGUAJE SQL

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

PRIMERA PARTE: EJERCICIOS VARIADOS

EJERCICIO 1:

A continuación se muestran las tablas que forman una base de datos.

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

Ciu(Ciudad) = {ciudad_id, ciudad_nom}

Cree tablas teniendo en cuenta las restricciones de integridad referencial

A.

Clave foránea - Tabla PERS	Nulos	Eliminar	Modificar
ciudad_idc	No	Restringir	Propagar

B.

Clave foránea - Tabla PERS	Nulos	Eliminar	Modificar
ciudad_idc	No	Nulificar	Propagar

1. Inserte ciudades con los siguientes datos:

< 1,'Galilea' >
< 2,'Betsaida' >
< 3,'Patmos' >
< 4,'Jerusalén' >

2. Inserte tuplas en la tabla Personas con los siguientes datos:

< 1,'Pedro Cortes',1,16 >
< 2,'Santiago Mas',2, 25>
< 3,'Juan Basti',5, 50 >
< 4,'Andrea Hernández',1 >
< 5,'Ana Perez',1,30 >

3. Elimine la ciudad con identificador igual a 1.

SEGUNDA PARTE: A CONTINUACIÓN SE PRESENTAN CASOS SOBRE LOS CUALES DEBERÁN IMPLEMENTARSE LAS CONSULTAS PLANTEADAS

Caso 1: Cursos de Verano - Dataset ID: ce3db2b90c57e7627d4ada85e2b800eb

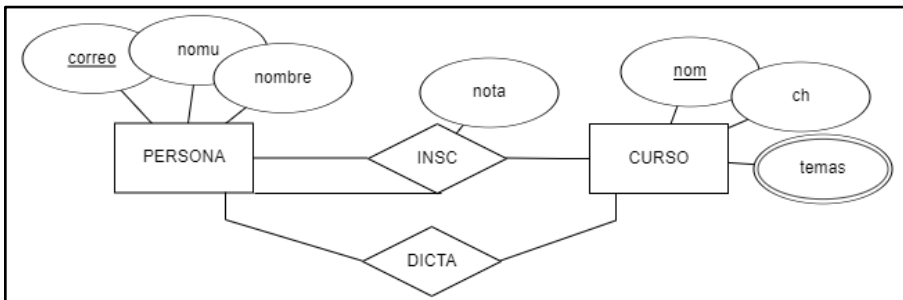
PERS (Personas) = { correo, nomu (nombre usuario), nombre }

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

INSC (Inscribe) = { correo, nom, correod, nota }

DICTA = { correo, nom }

TEMAS={ nom, tema }



1. Cree las tablas correspondientes a la Base de Datos Cursos de VERANO

2. Inserte nuevas Personas con los siguientes datos:

< anagarcia56@gmail.com, AnaG56, Ana Garcia >
< franciscopeña52@yahoo.com.ar, FranPeña865, Gonzalo Ramos >
< pedroibañez@yahoo.com.ar, PedroIbañez223, Pedro Ibañez >
< gisellevasquez201@yahoo.com.ar, Giselle201, Giselle Vasquez >
< giselleV146@gmail.com, GVasquez24, Giselle Vasquez >

3. Inserte el nuevo curso con los siguientes datos:

< Python I, 29 >
< Python II, 28 >
< Kotlin I, 45 >
< Kotlin II, 42 >

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

< Python I, Estructuras de control >
< Python I, Aplicacion >
< Python II, Estructuras de datos >
< Python II, Estructuras de control >
< Kotlin I, Estructuras de datos >
< Kotlin I, Aplicacion >

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

< estefaniaperalta@gmail.com, Python I >
< estefaniaperalta@gmail.com, Python II >

<gonzaloramos1978@yahoo.com.ar, Kotlin I>

<gonzaloramos1978@yahoo.com.ar, Kotlin II>

6. Inserte la nueva tupla para INSCRIPCION con los siguientes datos:

<anagarcia32@gmail.com, Python I, estefaniaperalta@gmail.com, 7>

<franciscopeña52@yahoo.com.ar, Python I, estefaniaperalta@gmail.com, 9>

<anagarcia32@gmail.com, Python II, pedroibañez@yahoo.com.ar, 8>

<anagarcia56@gmail.com, Python II, estefaniaperalta@gmail.com, 6>

<pedroibañez@yahoo.com.ar, Python I, estefaniaperalta@gmail.com, 9>

<rubengonzales104@gmail.com, Kotlin I, gonzaloramos1978@yahoo.com.ar, 8>

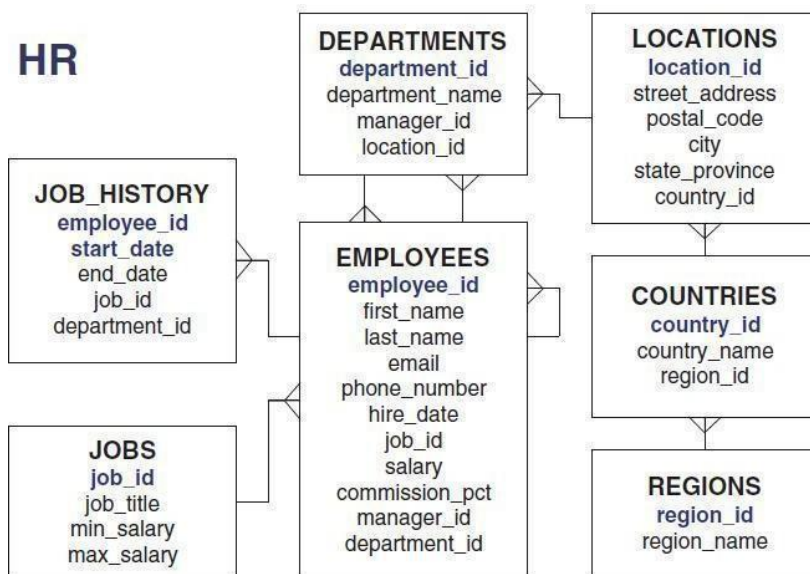
<rubengonzales104@gmail.com, Kotlin II, gonzaloramos1978@yahoo.com.ar, 6>

7. Actualice la carga horaria del curso Ruby por 60.
8. Elimine el curso Ruby I.
9. Correo y nombre de todas las personas.
10. Cantidad de cursos.
11. Cantidad de docentes.
12. Nota máxima obtenida en el curso "Python I".
13. Nombre de los cursos ordenados por nombre.
14. Nombre del curso que tiene una carga horaria superior a la de todos los cursos que dicta "pedroibañez@yahoo.com.ar".
15. Personas, docentes o alumnos(todos sus datos) que se llama Rosa
16. Cursos que tienen una carga horaria superior a la del curso "Kotlin I", ordenados descendentemente por cantidad de horas.
17. Cursos (todos los datos) cuya carga horaria sea superior a las 40 horas reloj.
18. Cursos (todos los datos) cuya carga horaria se encuentre entre 40 y 45 horas reloj.
19. Docentes (correo y nombre) que dictan cursos.
20. Listado de los cursos (nombre) junto a los datos del docente que los dicta.
21. Obtenga el curso (todos los datos) junto a los datos de los alumnos inscriptos. Se deben incluir todos los cursos registrados más allá que no tengan alumnos inscriptos.. (Usar OUTER JOIN)
22. Docentes (todos los datos) que dictan los cursos "Python I".
23. Docentes (todos los datos) que dictan los cursos "Python II".
24. Listado de docentes (correo) que dictan el curso "Python I" y/o "Python II".
25. Docentes (correo) que dictan los cursos "Python I" y "Python II".
26. Docentes (todos los datos) que cursaron algún curso de verano.
27. Alumnos (todos los datos) que se inscribieron en el curso "Kotlin I".
28. Alumnos (todos los datos) que se inscribieron en el curso "Kotlin II".
29. Listado de alumnos (correo) que se inscribieron tanto en el curso "Kotlin I" como "Kotlin II".
30. Alumnos (todos los datos) que aprobaron el curso "Python I" y "Python II".
31. Alumnos (Correo) que se inscribieron en más de un curso de verano.
32. Docentes (correo) que dictan más de un curso.
33. Docentes (todos los datos) que dictan más de un curso cuya carga horaria sea inferior a 30 horas reloj.
34. Alumnos (correo) que cursaron los mismos cursos.
35. Pares de Alumnos (todos los datos) que cursaron los mismos cursos.
36. Pares de Alumnos que cursaron los mismos cursos con distinto profesor.

37. Alumnos (todos los datos) que se inscribieron en todos los cursos de verano.
38. Alumnos (todos los datos) que se inscribieron en todos los cursos que dicta el profesor con correo "pedroibañez@yahoo.com.ar"
39. Nombre/s de los cursos que tienen la mayor carga horaria.
40. 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.
41. Muestre los datos contenidos en la vista, ordenados según el nombre.
42. Especifique la Vista (Alumnos_python1) que contenga los alumnos que se inscribieron en el curso "PYTHON I" correo, nombre de usuario y nombre).
43. Muestre los datos contenidos en la vista creada en el punto anterior, cuyo correo sea una cuenta de Gmail

Caso 2 - ESQUEMA HR (RECURSOS HUMANOS) Script

A continuación se muestra el esquema HR de Recursos Humanos.



Employees = {employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id}

Jobs = {job_id, job_title, min_salary, max_salary}

Job_History = {employee_id, start_date, end_date, job_id, department_id}

Departments = {departament_id, departament_name, manager_id, location_id}

Locations= {location_id, street_address, postal_code, city, state_province, country_id}

Countries= {country_id, country_name, region_id}

Regions= {region_id, region_name}

1. Obtenga el nombre completo (*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. Obtenga el nombre completo (*first_name* y *last_name*) y el salario (*salary*) de todos los empleados ordenado alfabéticamente por apellido (*last_name*).
3. Obtenga 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. Obtenga la cantidad de empleados que trabajan en el departamento 100.

5. Obtenga todos los datos de los departamentos con identificador 10 ó 70 (*department_id*).
6. Obtenga el apellido (*last_name*) de los empleados junto al nombre del departamento (*department_name*) donde trabajan.
7. Obtenga el apellido (*last_name*) de los empleados de los departamentos de Finanzas (Finance) y/o Transporte (Shipping).
8. Obtenga sin repetir los tipos de trabajos (*jobs*) que realizan en los departamentos los empleados. Liste el identificador del departamento y el nombre del trabajo.
9. Obtenga 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.
10. Obtenga el nombre de los departamentos que tienen al menos 3 empleados.
11. Obtenga el apellido 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.
12. Obtenga el salario máximo de los empleados del departamento 110. Ordene el listado en forma ascendente por salario.
13. Obtenga el nombre del/de los empleados que tienen el sueldo máximo del departamento 110. Ordene el listado en forma ascendente por salario
14. Obtenga todos los datos de los empleados que no sean supervisores (o directores). NOTA: En la tabla departments el atributo manager_id tiene la identificación de los supervisores de cada dpto.

Caso 3– OBRAS CIVILES

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}

1. Muestre, a través de una consulta, los materiales (descripción) pedidos el día 06/06/2020.
2. Muestre para cada obra (indicando descripción) todos los materiales solicitados (descripción). Deben informarse todas las obras, más allá que aún no tenga materiales pedidos.
3. Muestre la cantidad total de bolsas de cal que han sido pedidas a la ferretería MR S.A.
4. Muestre la cantidad total de obras que han pedido materiales a la ferretería MR S.A.
5. Muestre, para cada material pedido a alguna ferretería, el código de material, código de obra y la cantidad total pedida (independientemente de la ferretería).
6. Muestre la descripción de materiales pedidos para alguna obra en una cantidad promedio mayor a 320 unidades.
7. Muestre el nombre del material menos pedido (en cantidad total).
8. Muestre la descripción de las obras que no han utilizado pintura.
9. Muestre el nombre de las obras abastecidas totalmente por la ferretería MR S.A.
10. Muestre el nombre de los materiales que han sido pedidos para todas las obras realizadas.
11. Actualice el teléfono de la Ferretería San Ignacio por el número 4312548.
12. Elimine el Material con descripción Cemento Avellaneda.
13. 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 Santa Lucia.

14. 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).
 - Actualice la cantidad de materiales pedidos de las obras incrementándolas en 100.

Caso 4: Películas

Se almacena en una base de datos información referente a las personas y las películas en las que actúan y/o dirigen como así también lo referido a sus seguidores.

Persona= {nom, nombre, fechanac (fecha de nacimiento)} **Película** = {título, estreno (fecha de estreno), lema}

Dirige = {nom, título}

Actúa= {nom, título}

Sigue= {seguido (Persona), seguidor (Persona)}

- Personas (nombre) que han actuado en más de una película estrenada en el año 1990.
- Películas (título y lema) en las que han actuado solamente personas que nacieron antes del 1970.
- Personas (todos los datos) que han actuado en todas las películas dirigidas por Carrie-Anne Moss.
- Obtener el título y fecha de estreno de las películas dirigidas por Keanu Reeves.
- Personas (todos los datos) que han actuado y/o dirigido en las mismas Películas en las que actuó Keanu Reeves.
- Personas (nombre) que han actuado en las películas The Matrix y The Matrix Revolutions.
- Persona/s (todos los datos) que ha/n dirigido más películas.
- Nombre de la persona junto a la cantidad de películas que ha dirigido.
- Personas (todos sus datos) que han participado actuando y dirigiendo la misma película.
- Título de la película junto a la cantidad de personas que participaron actuando y/o dirigiendo.
(consultar)

Caso 5: Empresa de comida rápida

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}

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

EJERCICIOS PROPUESTOS

Caso 6 - RECARGAS Y VIAJES DE LA TARJETA SUBE

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. Personas (todos los datos) que solamente han tomado la línea 26A de la provincia de San Juan.
 2. Empresas (todos los datos) en las que viajaron personas oriundas de la provincia de San Juan y San Luis.
 3. Personas (dni y nombre) que viajaron más de una vez en una fecha, para una línea y empresa determinada.
 4. Personas que viajaron más de una vez en la línea 17 La positiva, el día 02/03/16.
 5. Obtener todos los datos de las personas que viajaron en todas las empresas de la provincia de San Juan (sin importar la línea).
 6. Obtener el importe total que gastó Juan Pérez cuyo dni es 36800700 en lo que va del año 2018.
 7. 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 lo que va del año 2018.

Caso 7 – 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 correspondiente a la empresa La Positiva.
6. Mostrar el nombre de las calles por las que circulan 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