NOTA:Dadas las siguientes relaciones, resolver utilizando Álgebra Relacional las consultas planteadas.

operaciones:

* σ selección
* π proyección
* υ union
* ∩ intersección
* x producto cartesiano
* |x| producto natural
* - diferencia
* ÷ división
* ρ renombre
* ← asignación
* δ actualización

símbolos útiles:

* ∧
* ∨

Ejercicio 1:

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

Factura​​(​nroTicket​, total, fecha, hora,idCliente(Fk))

Detalle​(​nroTicket, idProducto​, cantidad, preciounitario)

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

1. Listar nombre, apellido, DNI, teléfono y dirección de clientes con DNI superior a 22222222.

πnombre,apellido,DNI,teléfono,dirección(σDNI>22222222(Cliente))

2. Listar nombre, apellido, DNI, teléfono y dirección de clientes con DNI superior a 22222222 y que tengan facturas cuyo total no supere los $100000.

πnombre,apellido,DNI,teléfono,direccion((σDNI>22222222(Cliente) |x| πidCliente(σtotal<=$100000 (Factura)))

¿Está bien la solución? creo que termino con muchos clientes repetidos en caso de que hayan realizado mas de una compra que cumpla la condición.

3. Listar nombre, apellido, DNI, teléfono y dirección de clientes que realizaron compras durante 2020.

πnombre,apellido,DNI,teléfono,direccion(Cliente |x| πidCliente(σfecha>=1/1/2020 & fecha<1/1/2021 (Factura)))

Lo hice como en la teoría, que había un ejemplo muy parecido pero...

¿Está bien la solución? creo que termino con muchos clientes repetidos en caso de que hayan realizado mas de una compra que cumpla la condición.

Por ejemplo: cliente con id 98 tiene 2 facturas del 2020,al hacer la selección obtenemos la tabla:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nroTicket | total | fecha | hora | idCliente(fk) |
| ​​1 | 200 | 12/03/2020 | 12:00 | 98 |
| ​​2 | 600 | 10/04/2020 | 12:00 | 98 |

luego hacemos la proyección de idCliente y obtenemos:

|  |
| --- |
| idCliente(fk) |
| 98 |
| 98 |

Si entonces hacemos el producto natural con los clientes, quedaria asi:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| idCliente(fk) | nombre | apellido | DNI | teléfono | dirección |
| 98 | Juan | Ponci | 33333333 | 42675595 | 123 e/ 23 y 23 |
| 98 | Juan | Ponci | 33333333 | 42675595 | 123 e/ 23 y 23 |

Esto no estaría mal como resultado

πnombre,apellido,DNI,teléfono,dirección(

Cliente - (Cliente - (Cliente |x| πidCliente(σfecha>=1/1/2020 & fecha<1/1/2021 (Factura)))))

Pienso que así se pueden eliminar los renglones repetidos ¿Esta bien?

No se exige que no haya repetidos.

4. Listar nombre, apellido, DNI, teléfono y dirección de clientes que no realizaron compras durante 2020.

πnombre,apellido,DNI,teléfono,direccion(Cliente |x| πidCliente(σfecha<1/1/2020 | fecha>=1/1/2021 (Factura)))

Luego de corregir los anteriores corregir esto en consecuencia.

5. Listar nombre, apellido, DNI, teléfono y dirección de clientes que solo tengan compras durante 2020.

πnombre,apellido,DNI,teléfono,direccion((Cliente |x| πidCliente(σfecha>=1/1/2020 & fecha<1/1/2021 (Factura)))

- (Cliente |x| πidCliente(σfecha<1/1/2020 & fecha>=1/1/2021 (Factura))))

Luego de corregir los anteriores corregir esto en consecuencia.

6. Listar nombre, descripción, precio y stock de productos no vendidos.

πnombreP,descripción,precio,stock(Productos - (πidProducto(Detalle) |x| Productos))

Luego de corregir los anteriores corregir esto en consecuencia.

7. Listar nombre, apellido, DNI, teléfono y dirección de clientes que no compraron el producto con nombre ‘ProductoX’ durante 2020.

ClientesQueSi ← πidCliente,nombre,apellido,DNI,teléfono,dirección

(σ(fecha>=1/1/2020 | fecha<1/1/2021) & nombreP=”Productox” (Cliente |x| Factura |x| Detalle |x| Producto))

πnombre,apellido,DNI,teléfono,dirección(Clientes - ClientesQueSi)

Luego de corregir los anteriores corregir esto en consecuencia.

8. Listar nombre, apellido, DNI, teléfono y dirección de clientes que compraron el producto con nombre ‘Producto A’’ y no compraron el producto con nombre ‘Producto B’.

ClientesA ← πidCliente,nombre,apellido,DNI,teléfono,dirección

(σnombreP=”Producto A” (Cliente |x| Factura |x| Detalle |x| Producto))

ClientesB ← πidCliente,nombre,apellido,DNI,teléfono,dirección

(σnombreP=”Producto B” (Cliente |x| Factura |x| Detalle |x| Producto))

πnombre,apellido,DNI,teléfono,dirección(ClientesA ∩ (Clientes - ClientesB))

9. Listar nroTicket, total, fecha, hora y DNI del cliente, de aquellas facturas donde se haya comprado el producto ‘Producto C’.

FacturasC ← π​nroTicket​,total,fecha,hora,idCliente(σnombreP=”Producto C” (Factura |x| Detalle |x| Producto))

πnroTicket, total, fecha, hora,DNI(Cliente |x| FacturasC)

10. Agregar un producto con id de producto 1000, descripción “mi producto”, precio $10000, nombreP “producto Z” y stock 1000. Se supone que el idProducto 1000 no existe.

Producto ← Producto υ {(1000,”mi producto”$10000,”Producto Z”,1000)}

Ejercicio 2:

Banda​(​codigoB​,nombreBanda​,​ genero\_musical, año\_creacion)

Integrante ​(​DNI​,nombre,apellido,dirección,email,fecha\_nacimiento,codigoB(fk))

Escenario​(​nroEscenario​,nombre\_escenario,ubicación,cubierto,m2,descripción)

Recital​(​fecha,hora,nroEscenario,​codigoB(fk))

1. Listar datos personales de integrantes con apellido ‘Garcia’ o fecha de nacimiento anterior a 2005 que toquen en bandas de rock and roll.

πDNI​,nombre,apellido,dirección,email,fecha\_nacimiento

(σapellido=”García” or (fecha\_nacimiento<1/1/2005 & genero\_musical=”rock and roll”) (Banda |x| Integrante))

2. Listar nombre de escenario, ubicación y descripción de escenarios que no tuvieron recitales durante 2019.

Recitales2019 ← (σfecha>=1/1/2019 & fecha<1/1/2020(Recital))

Escenarios2019 ← (Escenario |x| πnroEscenario(Recitales2019))

πnombre\_escenario,ubicación,descripción(Escenario - Escenarios2019)

3. Listar nombre de escenario, ubicación y descripción de escenarios que tuvieron recitales con género musical rock and roll o tuvieron recitales durante 2020.

BandasRaR ← (σgenero\_musical=”rock and roll”(Banda))

RecitalesRaR ← (Recital |x| πcodigoB(BandasRaR))

Recitales2020 ← (σfecha>=1/1/2019 & fecha<1/1/2020(Recital))

Recitales2020oRaR ← (Recitales2020 υ RecitalesRaR)

Escenarios2020oRaR ← (Escenario |x| πnroEscenario(Recitales2020yRaR))

πnombre\_escenario,ubicación,descripción(Escenario - Escenarios2020yRaR)

4. Listar nombre, género musical y año de creación de bandas que hayan realizado recitales en escenarios cubiertos durante 2019. // cubierto es true, false según corresponda.

RecitalesC2019 ← (σ(fecha>=1/1/2019 & fecha<1/1/2020) & cubierto(Recital))

πnombre,genero\_musical,año\_creacion(Bandas |x| σcodigoB(RecitalesC2019))

5. Listar DNI, nombre, apellido, dirección y email de integrantes nacidos entre 2000 y 2005 y que toquen en bandas con género pop que hayan tenido recitales durante 2020.

IntegrantesND2000H2005 ← σfecha\_nacimiento>=1/1/2000 & fecha\_nacimiento<1/1/2006(Integrantes)

BandasPop ← σgenero\_musical=”pop”(Banda)

Recitales2020 ← (σfecha>=1/1/2020 & fecha<1/1/2021(Recital))

BandasPopCR2020 ← (BandasPop |x| πcodigoB(Recitales2020))

IntegrantesBPopCR2020yND2000H2005 ←

(IntegrantesND2000H2005 |x| πcodigoB(BandasPopCR2020))

πDNI​,nombre,apellido,dirección,email(IntegrantesBPCR2020yND2000H2005)

6. Listar DNI, nombre, apellido, email de integrantes que hayan tocado en el escenario con nombre ‘Gustavo Cerati’ y no hayan tocado en el escenario con nombre ‘Carlos Gardel’.

GC ← (σnombre=”Gustavo Cerati”(Integrante))

IntegrantesGC ← ((Integrante |x| πcodigoB(GC)) - GC)

CG ← (σnombre=”Carlos Gardel”(Integrante))

IntegrantesCG ← ((Integrante |x| πcodigoB(CG)) - CG)

IntegrantesNCG ← Integrante - IntegrantesCG

IntegrantesCGCyNCG ← (IntegrantesGC ∩ IntegrantesNCG)

πDNI​,nombre,apellido,email(IntegrantesGCyNCG)

7. Modificar el año de creación de la banda de nombre ‘Ratones Paranoicos’ a: 1983.

RP ← (σnombreBanda=”Ratones Paranoicos”(Banda))

δ año\_creacion←1938(RP)

8. Reportar nombre, género musical y año de creación de bandas que hayan realizado recitales durante 2019, y además hayan tocado durante 2020.

Recitales2019 ← (σfecha>=1/1/2019 & fecha<1/1/2020(Recital))

Recitales2020 ← (σfecha>=1/1/2020 & fecha<1/1/2021(Recital))

Bandas2019 ← (πcodigoB(Recital2019) |x| Banda)

Bandas2020 ← (πcodigoB(Recital2020) |x| Banda)

Banda2019y2020 ← (Recitales2019 ∩ Recitales2020)

πnombreBanda,genero\_musical,año\_creacion(Banda2019y2020)

9. Listar el cronograma de recitales del dia 04/12/2019. Se deberá listar: nombre de la banda que ejecutará el recital, fecha, hora, y el nombre y ubicación del escenario correspondiente.

RecitalesEnFecha ← (σfecha=4/12/2019(Recital))

πnombreBanda,fecha,hora,nombre\_escenario,ubicacion(Escenario |x| RecitalesEnFecha |x| Bandas)

Ejercicio 3:

Agencia ​(​RAZON\_SOCIAL​, dirección, telef, e-mail)

Ciudad ​(​CODIGOPOSTAL​, nombreCiudad, añoCreación)

Cliente ​(​DNI​, nombre, apellido, teléfono,dirección)

Viaje​(FECHA,HORA,DNI​(fk),cpOrigen(Fk),cpDestino(Fk),razon\_social(Fk),descripción)

//cpOrigen y cpDestino corresponden a la ciudades origen y destino del viaje

1. Eliminar el cliente con DNI:25326992.

Cliente ← (Cliente - σDNI=25326992(Cliente))

2. Listar datos personales de clientes que solo realizaron viajes locales. (En cada viaje realizado coincide la localidad origen con el destino, cpOrigen y cpDestino).

ViajesLocales ← (Viaje - σcpOrigen=cpDestino(Viaje))

πDNI(ViajesLocales) |x| Cliente

3. Listar información de agencias que no tengan viajes para el cliente DNI:22222222 durante el primer semestre de 2020.

πRAZON\_SOCIAL,dirección,telef,e-mail(Agencia |x| Viaje |x| σDNI=22222222(Cliente))

4. Listar información de agencias que realizaron viajes durante 2019 y no realizaron viajes durante 2020.

Viajes2019 ← (σFECHA>=1/1/2019 & FECHA<1/1/2020(Viaje))

Viajes2020 ← (σFECHA>=1/1/2020 & FECHA<1/1/2021(Viaje))

ViajesNo2020 ← (Viaje - Viajes2020)

Viajes2019yNo2020 ← (Viaje ∩ Viajes2020)

πRAZON\_SOCIAL(Viajes2019yNo2020) |x| Agencia

5. Agregar una agencia de viajes con los datos que desee.

Agencia ← Agencia υ {(“razon social 1”,“118 e/ 12 y 13”,85264426,”empMsM@gmail.com”)}

6. Listar datos personales de clientes que viajaron con ciudad destino ‘Lincoln’ pero no realizaron viajes con origen ‘La Plata’.

ViajesALincoln ← (σcpDestino=”Lincoln”(Viaje))

ViajesDeLaPlata ← (σcpOrigen=”La Plata”(Viaje))

ClientesViajaronALincoln ← (Cliente |x| πDNI(ViajesALincoln))

ClientesViajaronDeLAPlata ← (Cliente |x| πDNI(ViajesDeLaPlata))

ClientesViajaronALincoln - ClientesViajaronDeLAPlata

7. Listar nombre, apellido, dirección y teléfono de clientes que viajaron con todas las agencias.

π​RAZON\_SOCIAL,nombre,apellido,dirección,teléfono(Cliente |x| Viaje) ÷ π​RAZON\_SOCIAL(Agencia)

8. Listar código postal, nombre Ciudad y año creación de ciudades que no recibieron viajes durante 2020.

Viajes2020 ← (σFECHA>=1/1/2020 & FECHA<1/1/2021(Viaje))

π​​CODIGOPOSTAL​,nombreCiudad,añoCreación(σcpDestino=CODIGOPOSTAL(π​​cpDestino(Viajes2020) x Ciudad))

9. Reportar información de agencias que realizaron viajes durante 2019 o que tengan dirección igual a ‘General Pinto’.

Viajes2019 ← (σFECHA>=1/1/2019 & FECHA<1/1/2020(Viaje))

AgenciasRV2019 ← (πRAZON\_SOCIAL(Viajes2019) |x| Agencia)

AgenciasDeGP ← (σdireccion=”General Pinto”(Agencia))

(AgenciasRV2019 υ AgenciasDeGP)

10. Actualizar el teléfono del cliente con DNI: 2789655 a: 221-4400345.

ClienteObjetivo ← (σDNI=2789655”(Cliente))

Cliente ← (Cliente - ClienteObjetivo)

δ telef←221-4400345(ClienteObjetivo)

Cliente ← (Cliente υ ClienteObjetivo)

Luego de corregir los anteriores corregir esto en consecuencia.

Ejercicio 4:

Equipo​(​codigoE​, nombreE​,​ descripcionE)

Integrante ​(​DNI​, nombre, apellido, ciudad, email, telefono, codigoE(fk))

Laguna​(​nroLaguna​, nombreL, ubicación, extensión, descripción)

TorneoPesca​(​codTorneo​, fecha, hora,nroLaguna(fk)​,​ descripción)

Inscripción​(codTorneo,codigoE,​asistió, ganó​)

​// asistió y ganó son true o false según corresponda.

1. Listar DNI, nombre, apellido y email de integrantes que sean de la ciudad ‘La Plata’ y estén inscriptos en torneos que se disputaron durante 2019.

Torneos2019 ← (σFECHA>=1/1/2019 & FECHA<1/1/2020(TorneoPesca))

π​DNI​,nombre,apellido,ciudad,email(Torneos2019 |x| Inscripción |x| Equipo |x| Integrante)

2. Reportar nombre y descripción de equipos que solo se hayan inscripto en torneos de 2019.

Torneos2019 ← (σFECHA>=1/1/2019 & FECHA<1/1/2020(TorneoPesca))

Equipos2019 ← (πcodigoE(πcodTorneo(Torneos2019) |x| Inscripción) |x| Equipos)

TorneosNo2019 ←(TorneoPesca - Torneo2019)

EquiposNo2019 ← (πcodigoE(πcodTorneo(TorneosNo2019) |x| Inscripción) |x| Equipos)

π​nombreE,descripcionE(Equipos2019 - EquiposNo2019)

3. Listar nombre, ubicación, extensión y descripción de lagunas que hayan tenido torneos durante 2019 y no hayan tenido torneos durante 2020.

Torneos2019 ← (σFECHA>=1/1/2019 & FECHA<1/1/2020(TorneoPesca))

Torneos2020 ← (σFECHA>=1/1/2020 & FECHA<1/1/2021(TorneoPesca))

Lagunas2019 ← (πnroLaguna(Torneos2019) |x| Laguna)

Lagunas2020 ← (πnroLaguna(Torneos2020) |x| Laguna)

π​nombreE,descripcionE(Lagunas2019 - Lagunas2019)

4. Listar para la laguna con nombre ‘laguna x’, nombre y descripción de equipos ganadores de torneos que se disputaron durante 2019 en la mencionada laguna.

CodLagunaX ← (σnombreL=”laguna x”(Laguna))

Torneo2019EL ← (σFECHA>=1/1/2019 & FECHA<1/1/2020(TorneoPesca) |x| CodLagunaX)

π​nombreE,descripcionE(πcodigoE(πcodTorneo(Torneo2019EL) |x| σgano(Inscripción)) |x| Equipos)

5. Reportar nombre, y descripción de equipos que tengan inscripciones en todas las lagunas.

π​nombreE,descripcionE,nroLaguna(Torneo |x| Inscripción |x| Equipo) ÷ π​nroLaguna(Laguna)

6. Eliminar el equipo con código: 10000.

Equipo ← (Equipo - σcodigoE=10000(Equipo))

Luego de corregir los siguientes corregir esto en consecuencia.

7. Listar nombreL, ubicación, extensión y descripción de lagunas que no tuvieron torneos.

πnombreL,ubicación,extensión,descripción(Laguna - (πnroLaguna(Torneo) |x| Laguna))

8. Reportar nombre, y descripción de equipos que tengan inscripciones a torneos a disputarse durante 2019, pero no tienen inscripciones a torneos de 2020.

CodTorneo2019 ← (πcodTorneo(σFECHA>=1/1/2019 & FECHA<1/1/2020(TorneoPesca)))

CodTorneo2020 ← (πcodTorneo(σFECHA>=1/1/2020 & FECHA<1/1/2021(TorneoPesca)))

Equipo2019 ← (πcodigoE(CodTorneo2019 |x| Inscripción) |x| Equipos)

Equipo2020 ← (πcodigoE(CodTorneo2020 |x| Inscripción) |x| Equipos)

π​nombreE,descripcionE(Equipo2019 - Equipo2020)

9. Listar DNI, nombre, apellido, ciudad y email de integrantes que asistieron o ganaron algún torneo que se disputó en la laguna con nombre: ‘Laguna Brava’.

CodLagunaBrava ← (σnombreL=”laguna Brava”(Laguna))

CodTorneoEL ← (πcodTorneo(TorneoPesca |x| CodLagunaBrava))

CodEquipoEL ← (πcodigoE(πcodigoE(CodTorneoEL |x| σgano or asistió(Inscripción)) |x| Equipos))

π​DNI​, nombre, apellido, ciudad, email(CodEquipoEL |x| Integrante)

Ejercicio 5:

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

Ciudad(codigoCiudad,nombre)

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

Jugador(dni,nombre,apellido,edad,codigoCiudad(FK))

ClubJugador(codigoClub,dni,desde,hasta)

1. Reportar nombre y año de fundación de clubes de la ciudad de La Plata, además del nombre y dirección del estadio del mismo.

πnombre,anioFundacion(Estadio |x| Club |x| σnombre=”La Plata”(Ciudad))

2. Listar datos personales de jugadores actuales del club River Plate que hayan jugado en el club Boca Juniors.

πnombre,anioFundacion((πcodioClub(σnombre=”River Plate”(Club))) |x| ClubJugador |x| Jugador)

3. Listar información de todos los clubes donde se desempeñó el jugador: Marcelo Gallardo. Indicar nombre, año de fundación y localidad del club.

πnombre,anioFundacion,ciudad.nombre

(πdni(σnombre=”Marcelo Gallardo”(Jugador)) |x| ClubJugador |x| Club |x| Ciudad)

4. Reportar dni, nombre y apellido de aquellos jugadores que no tengan más de 25 años y jueguen en algún club de la ciudad de Junín.

CodCiudadJ ← (πcodigoCiudad(σnombre=”Junín”(Ciudad)))

CodClubJ ← (πcodigoClub(Club |x| CodCiudadJ))

JugadorNM25A ← (σedad<=25(Jugador))

πdni,nombre,apellido(CodClubJ |x| σhasta=nil(ClubJugador) |x| σedad<=25(Jugador))

5. Mostrar el nombre de los clubes que tengan jugadores de la ciudad de Chivilcoy mayores de 25 años.

CodCiudadC ← (πcodigoCiudad(σnombre=”Chivilcoy”(Ciudad)))

CodJugadorDeCM25A ← (πdni(σedad<=25(Jugador) |x| CodCiudadC))

πnombre(Club |x| σhasta=nil(ClubJugador) |x| CodJugadorDeCM25A)

6. Reportar el nombre y apellido de aquellos jugadores que hayan jugado en todos los clubes.

π​jugador.nombre,apellido,club.nombre(Club |x| ClubJugador |x| Jugador) ÷ π​club.nombre(Club)

7. Listar nombre de los clubes que no hayan tenido ni tengan jugadores de la ciudad de La Plata.

CodCiudadLP ← (πcodigoCiudad(σnombre=”La Plata”(Ciudad)))

CodJugadorDeLP ← (πdni(Jugador) |x| CodCiudadLP))

ClubConJugDeLP ← πClub.odigoClub,Club.nombre,Club.anioFundacion,Club.codigoCiudad

(Club |x| ClubJugador |x| CodJugadorDeLP)

πCub.nombre(Club - ClubConJugDeLP)

8. Mostrar dni, nombre y apellido de aquellos jugadores que jugaron o juegan en el club: Club Atlético Rosario Central.

CodClubARC ← (πcodigoClub(σnombre=”Atlético Rosario Central”(Ciudad)))

πJugador.nombre,Jugador.nombre,Jugador.apellido(Jugador |x| ClubJugador |x| CodClubARC)

9. Eliminar al jugador cuyo dni es: 24242424.

ClubJugador ← (ClubJugador - (ClubJugador |x| πdni(σdni=24242424(Jugador))))

Jugador ← (Jugador - σdni=24242424(Jugador))

Ejercicio 6:

Proyecto​(​codProyecto​,nombrP,descripcion,fechaInicioP,fechaFinP

,fechaFinEstimada,DNIResponsable(fk),equipoBackend(fk),equipoFrontend(fk))

//DNIResponsable corresponde a un empleado, equipoBackend y equipoFrontend corresponden a un equipo

Equipo​(​codEquipo​, nombreE, descripcionTecnologias,DNILider(fk))

​//DNILider corresponde a un empleado

Empleado​(​DNI​,nombre, apellido, telefono, direccion, fechaIngreso)

Empleado\_Equipo​(​codEquipo,DNI,fechaInicio​,fechaFin,descripcionRol)

1. Listar nombre, descripción, fecha de inicio y fecha de fin de proyectos ya finalizados que no fueron terminados antes de la fecha de fin estimada.

πnombrP,descripción,fechaInicioP,fechaFinP(σfechaFinP<>nil && fechaFin >= fechaFinEstimada(Proyecto))

2. Listar DNI, nombre, apellido, teléfono, dirección y fecha de ingreso de empleados que no hayan sido responsables de proyectos.

Empleado - πDNI,nombre,apellido,telefono,direccion,fechaIngreso

( σEmpleado.DNI=Proyecto.DNIResponsable(Proyecto x Empleado))

3. Listar DNI, nombre, apellido, teléfono y dirección de todos los empleados que trabajan en el proyecto con nombre ‘Proyecto X’. No es necesario informar responsable y líderes.

ProyectoX ← (σnombreP=”Proyecto X”(Proyecto))

EquipoXB ←(σProyectoX.equipoBackend=Equipo.codEquipo(πequipoBackend(ProyectoX) x Equipo))

EquipoXF ←(σProyectoX.equipoFrontend=Equipo.codEquipo(πequipoFrontend(ProyectoX) x Equipo))

EquipoX ← (EquipoXB υ EquipoXF)

πDNI,nombre,apellido,telefono,direccion,direccion(EquipoX |x| Empleado\_Equipo |x| Empleado)

4. Listar nombre de equipo y datos personales de líderes de equipos que no tengan empleados asignados y trabajen con tecnología ‘Java’.

Empleado - EmpleadoLider- EmpleadoEquipo

¿Cómo me doy cuenta que un líder no tiene empleados asignados?

5. Modificar nombre, apellido y dirección del empleado con DNI: 40568965 con los datos que desee.

EmpleadoObjetivo ← (σDNI=40568965”(Empleado))

δ nombre←”Pablo”,apellido←”Escobar”(EmpleadoObjetivo)

6. Listar DNI, nombre, apellido, teléfono y dirección de empleados que son responsables de proyectos pero no han sido líderes de equipo.

EmpleadoResponsable ←

(πDNI,nombre,apellido,telefono,direccion,direccion(σEmpleado.DNI=Proyecto.DNIResponsable(Proyecto x Empleado)))

EmpleadoLider ←

(πDNI,nombre,apellido,telefono,direccion,direccion(σEmpleado.DNI=Equipo.DNILider(Equipo x Empleado)))

πDNI,nombre,apellido,telefono,direccion,direccion(EmpleadoResponsable - EmpleadoLider)

7. Listar nombre de equipo y descripción de tecnologías de equipos que hayan sido asignados como equipos frontend y backend.

EquipoB ←(σProyecto.equipoBackend=Equipo.codEquipo(πequipoBackend(Proyecto) x Equipo))

EquipoF ←(σProyecto.equipoFrontend=Equipo.codEquipo(πequipoFrontend(Proyecto) x Equipo))

EquipoByF ← (EquipoB ∩ EquipoF)

πnombreE,descripcionTecnologias(EquipoByF)

8. Listar nombre, descripción, fecha de inicio, nombre y apellido de responsables de proyectos a finalizar durante 2019.

ProyectoEstPara2019 ← (σfechaFinEstimada>=1/1/2019 && fechaFinEstimada<1/1/2020(Proyecto))

πProyecto.nombrP,Proyecto.descripción,Proyecto.fechaInicioP,,Empleado.nombre,Empleado.apellido

(σEmpleado.DNI=Proyecto.DNIResponsable(ProyectoEstPara2019 x Empleado)))

9. Listar nombre de equipo, descripción de tecnología y la información personal del líder, de equipos que no estén asignados a ningún proyecto aun.

CodEquipoAsF ← (πequipoFrontend(Proyecto))

CodEquipoAsB ← (πequipoBackend(Proyecto))

CodEquipoAs ← (CodEquiposAsF υ CodEquiposAsB)

EuipoNoAs ← (Equipo - (CodEquipoAs |x| Equipo))

πEquipo.nombreE,Equipo.descripcionTecnologias,Empleado.DNI​,nombre,Empleado.apellido,Empleado.telefono,Empleado.direccion,Empleado.fechaIngreso

(σEmpleado.DNI=EquipoNoAs.DNILider((EquipoNoAs x Empleado))

Ejercicio​ 7:

Vehiculo​(​patente​, modelo, marca, peso, km)

Camion(​patente​, largo, max\_toneladas, cant\_ruedas, tiene\_acoplado)

Auto(​patente​, es\_electrico, tipo\_motor)

Service(​fecha, patente​, km\_service, observaciones, monto)

Parte(​cod\_parte​, nombre, precio\_parte)

Service\_Parte​(​fecha, patente, cod\_parte​,precio)

1. Listar todos los datos de aquellos camiones que tengan entre 8 y 12 ruedas, y que hayan realizado algún service antes de los 10000 km.

Servicem10000 ← (σkm\_service<10000(Service))

Camiond8a12R ← (σcant\_ruedas<=12 & cant\_ruedas>=8(Camión))

πVehiculo.patente,Vehiculo.modelo,Vehiculo.marca,Vehiculo.peso,Vehiculo.km,Camion8a12R.largo,Camion8a12R.max\_toneladas,

Camion8a12R.cant\_ruedas,Camion8a12R.tiene\_acoplado((Vehiculo |x| Camiond8a12R) |x| Servicem10000)

2. Listar los autos que hayan realizado el service “cambio de aceite” antes de los 13.000 km o hayan realizado el service “inspección general” que incluya la parte “filtro de combustible”.

ServiceConcreto1 ← (σkm\_service<13000 & observaciones=”cambio de aceite” (Service))

ServiceConcreto2 ← πService.fecha,Service.patente​,Service.km\_service,Service.observaciones,Service.monto

((σobservaciones=”inspección general”(Service) |x| Servicio\_Parte |x| σnombre=”filtro de combustible”(Parte)))

ServiceConcreto1o2 ← (ServiceConcreto1 υ ServiceConcreto2)

πpatente(ServiceConcreto1o2) |x| Auto

3. Dar de baja todos los camiones con más de 350.000 km.

CamionConcreto ← (πpatente(σkm>350000(Vehiculo)) |x| Camion)

4. Listar el nombre y precio de aquellas partes que figuren en todos los service realizados durante 2019.

πParte.nombre,Parte.precio\_parte,Service\_Parte.fecha(Parte |x| Service\_Parte) ÷ πService\_Parte.fecha(Service)

5. Listar todos los autos cuyo tipo de motor sea eléctrico. Mostrar información de patente, modelo, marca y peso.

πpatente,Vehiculo.modelo,Vehiculo.marca,Vehiculo.precio(σesElectrico=true(Auto) |x| Vehiculo)

6. Dar de alta una parte, cuyo nombre sea “Aleron” y precio $3400.

Parte ← Parte υ {(99999, “Aleron”, 3400)}

7. Dar de baja todos los services que se realizaron al auto con patente ‘AAA564’.

ServiceConcreto ← σpatente=”AAA564”(Service)

Service\_ParteConcreto ← σpatente=”AAA564”(Service\_Parte)

Service ← **(**Service - ServiceConcreto)

Service\_Parte ← **(**Service\_Parte - Service\_ParteConcreto)

8. Modificar el precio de las partes incrementando un 15 % dicho valor.

δ precio\_parte←precio\_parte+precio\_parte\*0,15(Parte)

Luego de corregir los anteriores corregir esto en consecuencia.

9.Listar todos los vehículos que hayan tenido services durante el 2019.

Vehiculo |x| πpatente(σfecha>=1/1/2019 & fecha<1/1/2020(Service))