

Ejercicio 1

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

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

Detalle(nroTicket, idProducto, cantidad, preciounitario)

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

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

π nombre, apellido, DNI, telefono, direccion(σ DNI > 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

π nombre, apellido, DNI, telefono, direccion(σ DNI > 22222222 and total < 100000 (Cliente |x| Factura))

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

π nombre, apellido, DNI, telefono, direccion(σ fecha >= "01/01/2020"(Cliente |x| Factura))

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

compras2020 <- π nombre, apellido, DNI, telefono, direccion(σ fecha.año >= "01/01/2020" and fecha <= "31/12/2020" (Cliente |x| Factura))

π nombre, apellido, DNI, telefono, direccion (Cliente |x| Factura) - compras2020

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

comprasAntes <- π nombre, apellido, DNI, telefono, direccion(σ fecha.año <= "01/01/2020" (Cliente |x| Factura))

comprasDespues <- π nombre, apellido, DNI, telefono, direccion(σ fecha >= "31/12/2020" (Cliente |x| Factura))

compras <- (comprasAntes \cup comprasDespues)

π nombre, apellido, DNI, telefono, direccion (Cliente - compras)

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

productosVendidos <- π nombre, descripcion, precio, stock (Producto |x| Detalle)

π nombre, descripcion, precio, stock (Producto - productosVendidos)

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

productoX <- π nombre, apellido, DNI, telefono, direccion (σ nombreP = "ProductoX" and fecha.año >= "01/01/2020" and fecha <= "31/12/2020" (Cliente |x| Factura |x| Producto |x| Detalle))

π nombre, apellido, DNI, telefono, direccion (Producto - productoX)

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'.

productoA <- (σ nombreP = "Producto A"(Cliente |x| Factura |x| Producto |x| Detalle))

productoB <- (σ nombreP = "Producto B"(Cliente |x| Factura |x| Producto |x| Detalle))

π nombre, apellido, DNI, telefono, direccion (productoA - productoB)

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

π nroTicket, total, fecha, hora, DNI (σ nombreP = "Producto C" (Cliente |x| Factura |x| Producto |x| Detalle))

10. Agregar un producto con id de producto 1000, nombre "Producto Z", descripción "mi producto", precio \$10000 y stock 1000. Se supone que el idProducto 1000 no existe.

Producto \Leftarrow Producto $\cup \{(1000, \text{"Producto Z"}, \text{"mi producto"}, 10000, 1000)\}$

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(fk), 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.

π DNI, nombre, apellido, dirección, email, fecha_nacimiento (σ (apellido = "Garcia" or fecha_nacimiento < "01/01/2005") and 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 <- π nroEscenario, nombre_escenario, ubicación, cubierto, m2, descripción (σ fecha >= "01/01/2019" and fecha < "31/12/2019" (Escenario |x| Recital))

π nombre_escenario, ubicacion, descripcion (Recital - recitales2019)

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

π nombre_escenario, ubicacion, descripcion (σ genero_musical = "rock and roll" or (fecha >= "01/01/2020" and fecha <= "31/12/2020")) (Escenario |x| Recital |x| Banda))

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

π nombreBanda, genero_musical, año_creacion (σ cubierto = true and (fecha >= "01/01/2019" and fecha <= "31/12/2019")) (Escenario |x| Recital |x| Banda))

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

π DNI, nombre, apellido, dirección y email (σ (fecha_nacimiento >= "01/01/2000" and <= "31/12/2005") and genero_musical = "pop" and (fecha_nacimiento >= "01/01/2020" and <= "31/12/2020")) (Escenario |x| Recital |x| Banda |x| Integrante))

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'.

Gustavo <- (σ nombre = "Gustavo Cerati" (Integrante |x| Recital |x| Banda))

Carlos <- (σ nombre = "Carlos Gardel" (Integrante |x| Recital |x| Banda))

π DNI, nombre, apellido, email (Gustavo - Carlos)

7. Modificar el año de creación de la banda de nombre 'Ratones Paranoicos' a: 1983.
Banda (codigoB, nombreBanda, genero_musical, año_creacion)

Ratones <- (σ nombreBanda = "Ratones Paranoicos" (Banda))

δ año_creacion <- 1983 (Ratones)

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

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

π nombreBanda, fecha, hora, nombre_escenario, ubicacion (σ fecha = "04/12/2019" (Escenario |x| Recital |x| Banda))

Ejercicio 3

Agencia (RAZON_SOCIAL, dirección, telef, e-mail)

Ciudad (CODIGOPOSTAL, nombreCiudad, añoCreación)

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

Viaje (FECHA, HORA, DNI, cpOrigen (Fk), cpDestino (Fk), razon_social(Fk), descripcion)

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

1. Eliminar el cliente con DNI:25326992.

Cliente <- Cliente - (σ DNI = 25326992 (Cliente))

2. Listar datos personales de clientes que solo realizaron viajes locales. Se consideran viajes locales aquellos que tienen la misma ciudad como origen y destino.

π DNI, nombre, apellido, teléfono, dirección (σ cpOrigen = cpDestino (Cliente |x| Viaje))

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

antes <- (σ DNI = 22222222 and fecha <= "01/01/2020" (Agencia |x| Viaje |x| ClienteDni))

despues <- (σ DNI = 22222222 and fecha >= "01/06/2020" (Agencia |x| Viaje |x| ClienteDni))

π RAZON_SOCIAL, dirección, telef, e-mail (antes \cup despues)

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

viajes2019 <- (σ fecha <= "01/01/2019" and fecha <= "01/12/2019" (Agencia |x| Viaje))

viajes2020 <- (σ fecha <= "01/01/2020" and fecha <= "01/12/2020" (Agencia |x| Viaje))

π RAZON_SOCIAL, dirección, telef, e-mail (viajes2019 - viajes2020)

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

Viaje <- Viaje \cup {(1234, "Durañona 513", 2223123412, "mktagency@gmail.com")}

6. Listar datos personales de clientes que viajaron con destino a la ciudad de 'Lincoln' pero no realizaron viajes con origen en 'La Plata'.

viajesLincoln <- (σ cpDestino = "Lincoln" (Cliente |x| Viaje))

viajesLaPlata <- (σ cpDestino = "La Plata" (Cliente |x| Viaje))

π DNI, nombre, apellido, teléfono, dirección (viajesLincoln - viajesLaPlata)

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

π nombre, apellido, direccion, telef (Agencia |x| Viaje |x| ClienteDni)) % π RAZON_SOCIAL (Agencia)

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

viajes2020 <- (σ fecha <= "01/01/2020" and fecha <= "01/12/2020 (Viaje |x| Ciudad)

π CODIGOPOSTAL, nombreCiudad, añoCreación (Viaje |x| Ciudad - viajes2020)

9. Reportar información de agencias que realizaron viajes durante 2019 o que tengan dirección igual a 'General Pinto 1234'.

π RAZON_SOCIAL, dirección, telef, e-mail (σ direccion = "General Pinto 1234" or (fecha >= 01/01/2019 and fecha <= 31/12/2019) (Agencia |x| Viaje))

10. Actualizar el teléfono del cliente con DNI 2789655 al siguiente número de teléfono: 221-4400345.

clienteDni <- (σ DNI = 2789655 (Cliente)

δ telef <- 221-4400345 (clienteDni)

Ejercicio 4

Equipo (codigoE, nombreE, descripcionE)

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

Laguna (nroLaguna, nombreL, ubicación, extension, descripción)

TorneoPesca (codTorneo, fecha, hora, nroLaguna(fk), descripcion)

Inscripcion (codTorneo(fk),codigoE(fk), asistio, gano) // asistio y gano son true/false

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.

π DNI, nombre, apellido, email (σ ciudad = "La Plata" and (fecha >= 01/01/2019 and fecha <= 31/12/2019) (Integrante |x| Equipo |x| Inscripcion |x| TorneoPesca))

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

antes <- (σ fecha <= 01/01/2019 (Equipo |x| Inscripcion |x| TorneoPesca))

despues <- (σ fecha >= 31/12/2019 (Equipo |x| Inscripcion |x| TorneoPesca))

inscriptoNo2019 <- (antes \cup despues)

π nombreE, descripcionE (Equipo |x| Inscripcion |x| TorneoPesca |x| - InscriptoNo2019)

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

```
torneo2019 (σ <- fecha >= "01/01/2019" and fecha <= "31/12/2019" (Laguna |x| TorneoPesca))
torneo2020 (σ <- fecha >= "01/01/2020" and fecha <= "31/12/2020" (Laguna |x| TorneoPesca))
```

π nombreL, ubicación, extension, descripción (torneo2019 - torneo2020)

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.

```
π nombreE, descripcionE (σ nombreL = "laguna x" and (fecha >= 01/01/2019 and fecha <= 31/12/2019) and gano = true (Equipo |x| Inscripcion |x| TorneoPesca |x| Laguna))
```

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

```
π nombreE, descripcionE (Equipo |x| Inscripcion |x| TorneoPesca) % π nro_laguna (Laguna)
```

6. Eliminar el equipo con código 10000.

Borrar Integrantes -> Inscripcion -> Equipo9

```
Integrante <- Integrante - (σ codigoE = 10000(Integrante |x| Equipo)
Inscripcion <- Inscripcion - (σ codigoE = 10000(Equipo |x| Inscripcion)
Equipo <- Equipo - (σ codigoE = 10000(Equipo))
```

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

```
lagunasConTorneo <- π nroLaguna, nombreL, ubicación, extension, descripción (Laguna |x| TorneoPesca)
```

```
π nombreL, ubicación, extension, descripción(Laguna - lagunasConTorneo)
```

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

```
equipos2019 <- (σ fecha >= "01/01/2019" and fecha <= "31/12/2019 (Equipo |x| Inscripto |x| TorneoPesca)
equipos2020 <- (σ fecha >= "01/01/2020" and fecha <= "31/12/2020 (Equipo |x| Inscripto |x| TorneoPesca)
```

```
π nombreE, descripcionE (equipos2019 - equipos2020)
```

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'.

```
π DNI, nombre, apellido, ciudad, email (σ asistio = true or gano = true and nombreL = "Laguna Brava" (Integrante |x| Equipo |x| Inscripto |x| TorneoPesca |x| Laguna)
```

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(FK), dni(FK), 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.

π Club.nombre, Club.anioFundacion (σ Club.codigoClub = Estadio.codigoClub and Ciudad.codigoCiudad = Club.codigoCiudad and Ciudad.nombre = "La Plata" (Ciudad x Club x Estadio))

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

River <- π codigoClub (σ nombre = "River Plate" (Club))

Boca <- π codigoClub (σ nombre = "Boca Juniors" (Club))

jugadoresRiver <- π dni, nombre, apellido, edad, codigoCiudad(FK) ((River |x| (σ hasta > "26/10/2024" (Club |x|)) Jugador))

jugadoresBoca <- π dni, nombre, apellido, edad, codigoCiudad(FK) ((Boca |x| (σ hasta > "26/10/2024" (Club |x|)) Jugador))

π dni, nombre, apellido, edad, codigoCiudad(FK) (jugadoresRiver \cap jugadoresBoca)

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

Sapardo <- π dni (σ nombre = "Marcelo Gallardo" (Jugador))

CodigosDeClubos <- π codigoClub (ClubJugador |x| Sapardo)

π Club.nombre, Club.anioFundacion, Ciudad.nombre (CodigosDeClubos |x| (σ Club.codigoCiudad = Ciudad.codigoCiudad (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.

EdadMenos25 <- (σ edad < 25 (Jugador))

CodigosClubosJunin <- π codigoClub (σ Club.codigoCiudad = Ciudad.codigoCiudad and Ciudad.nombre = "Junin" (Club x Ciudad))

π dni, nombre, apellido (EdadMenos25 |x| ClubJugador) |x| CodigosClubosJunin)

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

EdadMayor25 <- π dni (σ edad > 25 (Jugador))
Chivilcoy <- π codigoCiudad (σ nombre = "Chivilcoy" (Ciudad))

π nombre (EdadMayor25 \times ClubJugador \times Club \times Chivilcoy)

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

π nombre, apellido (Jugador \times ClubJugador) % (π dni (ClubJugador))

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

JugadoresLaPlata <- π dni (σ Jugador.codigoCiudad = Ciudad.codigoCiudad and Ciudad.nombre = "La Plata" (Jugador \times Ciudad))
ClubesJugadoresLaPlata <- π codigoClub, nombre, anioFundacion, codigoCiudad(FK) (JugadoresLaPlata \times ClubJugador \times Club)

π nombre (Club - ClubesJugadoresLaPlata)

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

RosarioCentral <- π codigoClub (σ nombre = "Rosario Central" (Club))

π dni, nombre, apellido (Jugador \times ClubJugador \times RosarioCentral)

9. Eliminar al jugador cuyo dni es: 24242424.

ClubJugador - ClubJugador (σ dni = 24242424 (ClubJugador))
Jugador - Jugador (σ dni = 24242424 (Jugador))

Ejercicio 6

Proyecto = (codProyecto, nombreP, 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(fk), DNI(fk), 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

π nombreP, descripcion, fechaInicioP, fechaFinP (σ fechaFinP > fechaFinEstimada (Proyecto))

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

Responsables <- π DNI, nombre, apellido, telefono, direccion, fechaIngreso(σ Empleado.DNI = Proyecto.DNIResponsable (Empleado x Proyecto))

π DNI, nombre, apellido, telefono, direccion, fechaIngreso (Empleado - Responsables)

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

Front <- π equipoFrontEnd (σ nombreP = "Proyecto X" (Proyecto))

Back <- π equipoBackEnd (σ nombreP = "Proyecto X" (Proyecto))

EquiposFront <- π DNI, nombre, apellido, telefono, direccion, fechaIngreso (Front |x| Empleado_Equipo |x| Empleado)

EquiposBack <- π DNI, nombre, apellido, telefono, direccion, fechaIngreso (Back |x| Empleado_Equipo |x| Empleado)

π DNI, nombre, apellido, telefono, direccion, fechaIngreso (EquiposFront \cup EquiposBack)

4. Listar nombre de equipo y datos personales de líderes de equipos que no tengan empleados asignados y trabajen con tecnología 'Java'.
nose como resolverlo

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

δ nombre <- Fabio (σ DNI = 40568965 (Empleado))

δ apellido <- Torrejon (σ DNI = 40568965 (Empleado))

δ direccion <- 8 y 44 (σ DNI = 40568965 (Empleado))

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

responsables <- π Empleado.DNI, Empleado.nombre, Empleado.apellido, Empleado.telefono, Empleado.direccion(σ Proyecto.DNIResponsable = Empleado.DNI (Empleado x Proyecto))

lideres <- π Empleado.DNI, Empleado.nombre, Empleado.apellido, Empleado.telefono, Empleado.direccion(σ Equipo.DNILider = Empleado.DNI (Empleado x Equipo))

π DNI, nombre, apellido, teléfono y dirección (responsables - lideres)

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

```
back <-  $\pi$  Equipo.nombreE, Equipo.descripcionTecnologias ( $\sigma$  Equipo.codEquipo = Proyecto.equipoBackend(Equipo x Proyecto))
front <-  $\pi$  Equipo.nombreE, Equipo.descripcionTecnologias ( $\sigma$  Equipo.codEquipo = Proyecto.equipoFrontend(Equipo x Proyecto))
```

```
 $\pi$  nombreE, descripcionTecnologias(back  $\cap$  front)
```

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

```
Proyecto = (codProyecto, nombreP, 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(fk), DNI(fk), fechaInicio, fechaFin, descripcionRol)
```

```
 $\pi$  Proyecto.nombreP, Proyecto.descripcion, Proyecto.descripcion, Empleado.DNI, Empleado.apellido ( $\sigma$  Proyecto.DNIResponsable = Empleado.DNI and (Proyecto.fechaFinEstimada >= "01/01/2019 and Proyecto.fechaFinEstimada <= "31/12/2019") (Proyecto 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.

```
back <-  $\pi$  Equipo.codEquipo, Equipo.nombreE, Equipo.descripcionTecnologias, Equipo.DNILider ( $\sigma$  Equipo.codEquipo = Proyecto.equipoBackend(Equipo x Proyecto))
front <-  $\pi$  Equipo.codEquipo, Equipo.nombreE, Equipo.descripcionTecnologias, Equipo.DNILider ( $\sigma$  Equipo.codEquipo = Proyecto.equipoFrontend(Equipo x Proyecto))
```

```
equiposAsignados <- (back  $\cup$  front)
```

```
 $\pi$  codEquipo, nombreE, descripcionTecnologias, DNILider (Equipo - equiposAsignados)
```

Ejercicio 7

```
Vehiculo = (patente, modelo, marca, peso, km)
Camion = (patente(fk), largo, max_toneladas, cant_ruedas, tiene_acoplado)
Auto = (patente(fk), es_electrico, tipo_motor)
Service = (fecha, patente(fk), km_service, observaciones, monto)
Parte = (cod_parte, nombre, precio_parte)
Service_Parte = ([fecha, patente](fk), cod_parte(fk), precio, cantidad)
```

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

π patente, largo, max_toneladas, cant_ruedas, tiene_acoplado, modelo, marca, peso, km (σ cant_ruedas ≥ 8 and cant_ruedas ≤ 12 and km_service < 10000 (Vehiculo |x| Camion |x| Service))

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”.

AutosAceite $\leftarrow \pi$ patente (σ observaciones = “cambio de aceite” and km_service < 130000 (Auto |x| Service))

AutosGral $\leftarrow \pi$ patente (σ observaciones = “inspeccion general” and nombre = “filtro de combustible” (Auto |x| Service |x| Service_Parte |x| Parte))

π patente (AutosAceite \cup AutosGral)

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

PatentesCamiones $\leftarrow \pi$ patente (σ km > 350000 (Vehiculo |x| Camion))

Service_Parte \leftarrow Service_Parte - (Service_Parte |x| PatentesCamiones)

Service \leftarrow Service - (Service |x| PatentesCamiones)

Vehiculo \leftarrow Vehiculo - (Vehiculo |x| PatentesCamiones)

Camion \leftarrow Camion - (Camion |x| PatentesCamiones)

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

Services2019 $\leftarrow (\sigma$ fecha \geq “01/01/2019” and fecha \leq “31/12/2019 (Service))

π nombre, precio (Parte |x| Service_parte) % π fecha, patente (Services2019)

5. Listar todos los autos que sean eléctricos. Mostrar información de patente, modelo, marca y peso.

π patente, modelo, marca, peso (σ es_electrico = true (Vehiculo |x| Auto))

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

Parte \leftarrow Parte $\cup \{(\text{“Aleron”, } 3400)\}$

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

Service_Parte \leftarrow Service_Parte - (σ patente = “AAA564” (Service_parte))

Service \leftarrow Service - (σ patente = “AAA564” (Service))

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

δ precio_parte \leftarrow precio_parte * 1.015(Parte)

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

Services2019 <- (σ fecha >= "01/01/2019" and fecha <= "31/12/2019 (Service))

π patente, modelo, marca, peso, km (Vehiculo \times Service2019)

Ejercicio 8

Barberia = (codBarberia, razon_social, direccion, telefono)

Cliente = (nroCliente, DNI, nombreApellidoC, direccionC, fechaNacimiento, celular)

Barbero = (codEmpleado, DNIB, nombreApellidoB, direccionB, telefonoContacto, mail)

Atencion = (codEmpleado(fk), fecha, hora, codBarberia(fk), nroCliente(fk), descTratamiento, valor)

1. Listar DNI, nombre completo, dirección, teléfono de contacto y email de barberos que tengan atenciones con valor superior a 5000.

π DNIB, nombreApellidoB, direccionB, telefonoContacto, mail (σ valor > 5000 (Barbero \times Atencion))

2. Listar DNI, nombre y apellido, dirección, fecha de nacimiento y celular de clientes que tengan atenciones en la barbería con razón social 'Corta barba' y que también se hayan atendido en la barbería con razón social 'Barberia Barbara'.

CortaBarba <- (σ razon_social = "Corta barba" (Cliente \times Atencion \times Barberia))

Barbara <- (σ razon_social = "Barberia Barbara" (Cliente \times Atencion \times Barberia))

π DNI, nombreApellidoC, direccionC, fechaNacimiento, celular (CortaBarba \cap Barbara)

3. Eliminar el cliente con DNI 22222222.

ClienteDNI <- π nroCliente (σ DNI = 22222222 (Cliente))

Atencion <- Atencion - (Atencion \times ClienteDNI)

Cliente <- Cliente - (σ DNI = 22222222 (Cliente))

4. Listar los clientes más jóvenes que el cliente con nombre y apellido 'Juan Perez'

MenoresAJuan <- (σ Cli.fechaNacimiento < Cliente.fechaNacimiento (σ nombreApellidoC = "Juan Perez" (Cliente)) \times ρ Cli(Cliente))

π nroCliente, DNI, nombreApellidoC, direccionC, fechaNacimiento, celular (MenoresAJuan)

5. Listar los clientes que han tenido atenciones con todos los barberos que han trabajado en la barbería con razón social 'Corta Barba'.

BarberosBarba <- (σ razon social = "Corta Barba" (Barbero \times Atencion \times Barberia))

π nroCliente, DNI, nombreApellidoC, direccionC, fechaNacimiento, celular (Cliente $|x|$ Atencion) % π nroCliente (BarberosBarba)

6. Listar DNI y nombre completo de los barberos que solamente tengan atenciones a partir de 2024.

AtencionesAntes2024 <- (σ fecha < 31/12/2023 (Atencion))
AtencionesDespues2024 <- (σ fecha >= 01/01/2024 (Atencion))
Atenciones2024 <- (AtencionesDespues2024 - AtencionesAntes2024)

π DNIB, nombreApellidoB (Barbero $|x|$ Atenciones2024)

7. Modificar la dirección de la barbería con razón social 'Pelo & Barba' con su nueva dirección: '13 n°1234 La Plata'.

δ direccion <- "13 n°1234 La Plata" (Barberia)

8. Listar los datos de las atenciones realizadas por las barberías durante el mes de Septiembre de 2024, indicando por cada atención la razón social de la barbería, el nombre completo del empleado que realizó la atención, el nombre completo del cliente que recibió la atención, la fecha y hora, la descripción de los tratamientos aplicados y el valor de la atención.

π razon_social, nombreApellidoB, nombreApellidoC, fecha, hora, descTratamiento, valor (σ fecha >= "01/09/2024" and fecha <= "31/09/2024" (Barberia $|x|$ Atencion $|x|$ Cliente $|x|$ Barbero))

Ejercicio 9

Club = (IdClub, nombre, ciudad)

Complejo = (IdComplejo, nombre, IdClub(fk))

Cancha = (IdCancha, nombre, IdComplejo(fk))

Entrenador = (IdEntrenador, nombreEntrenador, fechaNacimiento, direccion)

Entrenamiento = (IdEntrenamiento, fecha, IdEntrenador(fk), IdCancha(fk))

1. Listar nombre, fecha de nacimiento y dirección de entrenadores que hayan entrenado en la canchas denominadas "Cancha 1" y "Cancha 2" del complejo con nombre "Norte" del club "Deportivo La Plata".

DeportivoLaPlata <- π idClub (σ nombre = "Deportivo La Plata" (Club))

Norte <- π idComplejo (σ nombre = "Norte" (Complejo))

EntrenoCancha1 <- (σ nombre = "Cancha 1" (Entrenador $|x|$ Entrenamiento Cancha $|x|$ DeportivoLaPlata $|x|$ Norte))

EntrenoCancha2 <- (σ nombre = "Cancha 2" (Entrenador $|x|$ Entrenamiento Cancha $|x|$ DeportivoLaPlata $|x|$ Norte))

π nombreEntrenador, fechaNacimiento, direccion (EntrenoCancha1 \cap EntrenoCancha2)

2. Listar nombre y ciudad de todos los clubes en los que entrena el entrenador "Marcos Perez".

$\text{MarcosPerez} \leftarrow \pi \text{ IdEntrenador } (\sigma \text{ nombreEntrenador} = \text{"Marcos Perez"} (\text{Entrenador}))$

$\pi \text{ Club.nombre, ciudad } (\sigma \text{ Complejo.idComplejo} = \text{Cancha.IdComplejo and Complejo.idClub} = \text{Club.IdClub } (\text{Cancha } |x| \text{ Entrenamiento } |x| \text{ MarcosPerez } x \text{ Complejo } x \text{ Club}))$

3. Eliminar los entrenamientos del entrenador 'Juan Perez'.

$\text{Entrenamiento} \leftarrow \text{Entrenamiento} - \pi \text{ IdEntrenamiento, fecha, IdEntrenador, IdCancha } (\sigma \text{ nombreEntrenador} = \text{"Juan Perez"} (\text{Entrenador } |x| \text{ Entrenamiento}))$

4. Listar los nombres de los clubes que se ubican en la misma ciudad que el club con nombre "Crucero del Sur".

$\text{clubesMismaCiudad} \leftarrow (\sigma \text{ Club.ciudad} = \text{cl.ciudad} (\sigma \text{ nombre} = \text{"Crucero del Sur"} (\text{Club})) x \rho \text{ cl } (\text{Club}))$

$\pi \text{ nombre } (\text{clubesMismaCiudad})$

5. Listar nombre y fecha de nacimiento de los entrenadores que hayan realizado en una misma fecha entrenamientos en todas las canchas del complejo con nombre "Centro" del club "Centro Fomento LH".

6. Listar nombre, fecha de nacimiento y dirección de aquellos entrenadores que entrenan clubes de la ciudad "La Plata" pero que no entrenan clubes de la ciudad "Berisso"

$\text{ClubesLaPlata} \leftarrow \pi \text{ idClub } (\sigma \text{ ciudad} = \text{"La Plata"} (\text{Club}))$

$\text{ClubesBerisso} \leftarrow \pi \text{ idClub } (\sigma \text{ ciudad} = \text{"Berisso"} (\text{Club}))$

$\text{EntrenadoresLaPlata} \leftarrow (\sigma \text{ Cancha.IdComplejo} = \text{Complejo.IdComplejo } (\text{Entrenador } |x| \text{ Entrenamiento } |x| \text{ Cancha } x \text{ Complejo } |x| \text{ ClubesLaPlata}))$

$\text{EntrenadoresBerisso} \leftarrow (\sigma \text{ Cancha.IdComplejo} = \text{Complejo.IdComplejo } (\text{Entrenador } |x| \text{ Entrenamiento } |x| \text{ Cancha } x \text{ Complejo } |x| \text{ ClubesBerisso}))$

$\pi \text{ nombreEntrenador, fechaNacimiento, direccion } (\text{EntrenadoresLaPlata} - \text{EntrenadoresBerisso})$

7. Listar la información de las canchas que disponen los clubes de la ciudad "La Plata". Por cada resultado, se debe informar el nombre del club, el nombre del complejo y el nombre de la cancha.

$\pi \text{ Club.nombre, Complejo.nombre, Cancha.nombre } (\sigma \text{ Club.IdClub} = \text{Complejo.IdClub and Complejo.IdComplejo} = \text{Cancha.IdComplejo } (\text{Cancha } x \text{ Complejo } x \text{ Club}))$