**Universidad Nacional de Colombia**

**Facultad de Ingeniería** 

**Curso:** Bases de datos Grupo 3

**Estudiantes**:

[Jonathan Ferney Casallas Moreno](mailto:jcasallasm@unal.edu.co), 1030523518

[Camilo Alejandro Raba Gomez](mailto:craba@unal.edu.co), 1001285732

[Oscar Iván García Montañez](mailto:osigarciamo@unal.edu.co), 1032443732

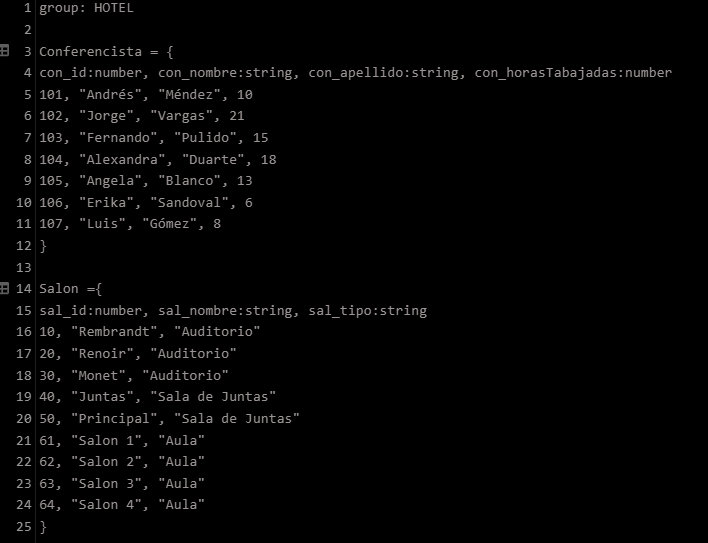
**TALLER AR**

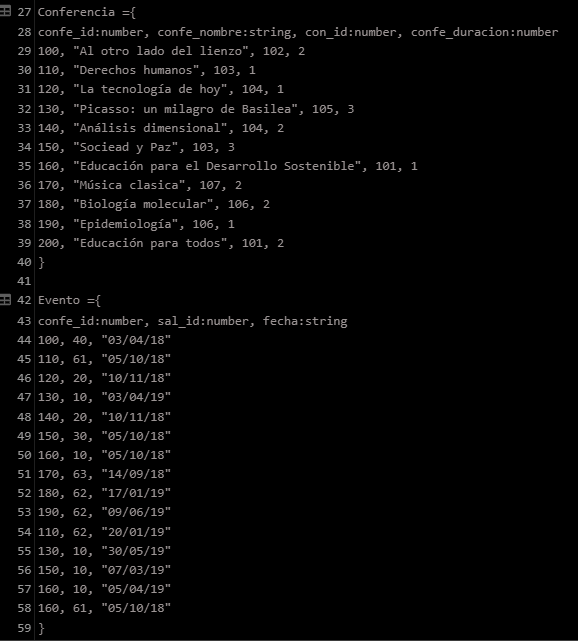
**1.** (20 ptos) En ReLax seleccionar la base de datos de ejemplo Silberschatz - UniversityDB. Entender las tablas y sus llaves foráneas. Realizar las siguientes consultas:

1. Listar todas las secciones de los cursos con su día y hora de inicio
2. Listar todos los estudiantes que tomaron el curso 'Database System Concepts', en qué año y cuál fue su nota.
3. Realizar 5 consultas más que sean interesantes (mínimo dos tablas relacionadas).

**2.** (40 puntos) Un hotel dispone de diferentes tipos de salones para llevar a cabo eventos. Un evento corresponde con la realización de una conferencia en un salón específico en una fecha específica. Las conferencias tienen un id, nombre, duración (en horas), y son realizadas por solo un conferencista, pero un conferencista puede realizar muchas conferencias. Un conferencista tiene id, nombre, apellido y horas trabajadas:

**1.1** Crear el script en ReLAX:

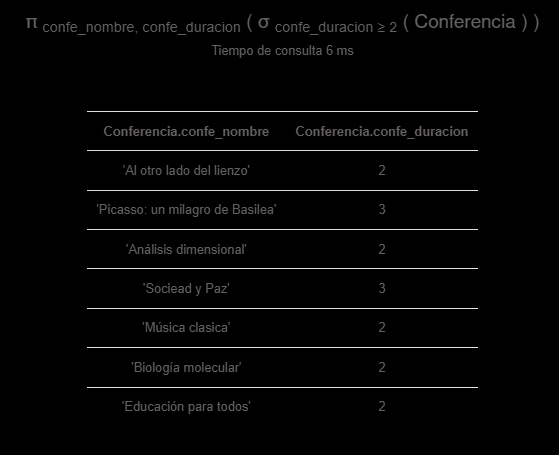




**1.2** Escribir la expresión en Álgebra Relacional de cada una de las siguientes consultas en ReLAX y ejecutarlas (para cada una de las consultas guardar el conjunto resultante):

1. nombre de las conferencias y su duración, solamente las que tienen duración de dos (2) horas o más:

*πconfe\_nombre, confe\_duracion(σconfe\_duracion≥2(Conferencia))*



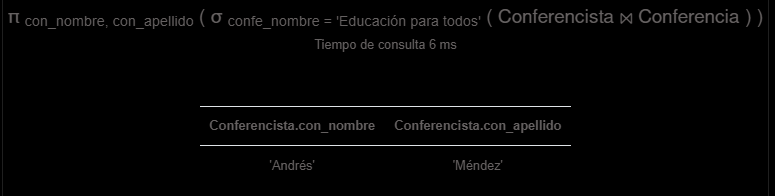
1. nombre de la conferencia con su respectivo conferencista (nombre y apellido):

*πconfe\_nombre,con\_nombre,con\_apellido(Conferencista⨝Conferencia)*

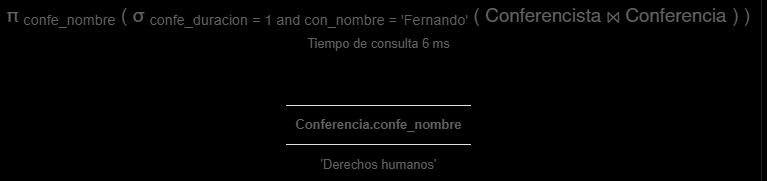


1. nombre y apellido del conferencista de la conferencia “Educación para todos”:

*πcon\_nombre,con\_apellido(σconfe\_nombre='Educación para todos'(Conferencista⨝Conferencia))*

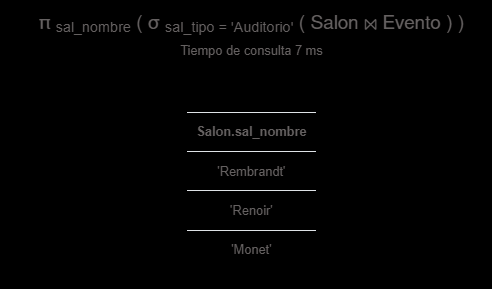


1. nombre de las conferencias con duración de 1 hora del conferencista “Fernando”:

*πconfe\_nombre(σconfe\_duracion=1and con\_nombre='Fernando'(Conferencista⨝Conferencia))*

1. nombre de los auditorios donde se han dictado conferencias:

*πsal\_nombre(σsal\_tipo='Auditorio'(Salon⨝Evento))*



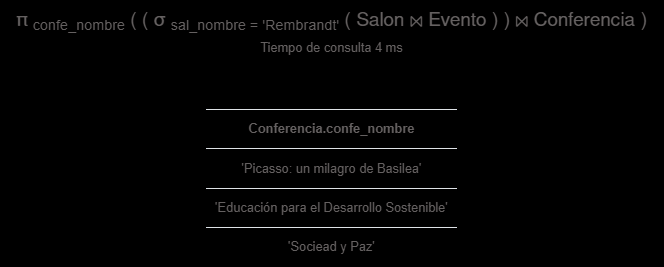
1. nombre y tipo de los salones donde se llevan a cabo conferencias:

*πsal\_nombre,sal\_tipo(Salon⨝Evento)*



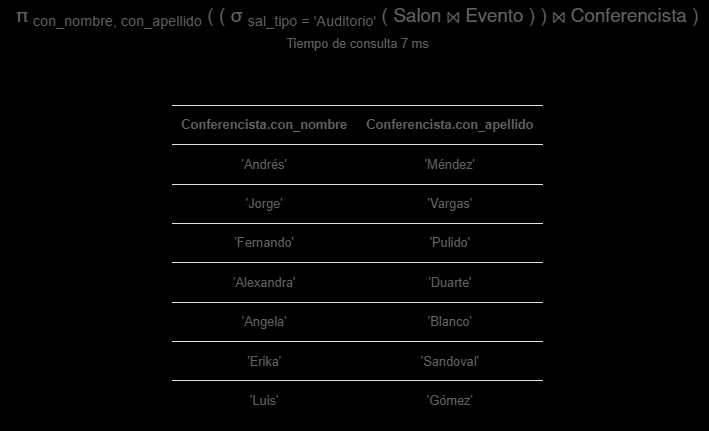
1. nombre de las conferencias que se han realizado en el salón “Rembrandt”:

*πconfe\_nombre((σsal\_nombre='Rembrandt'(Salon⨝Evento))⨝Conferencia)*



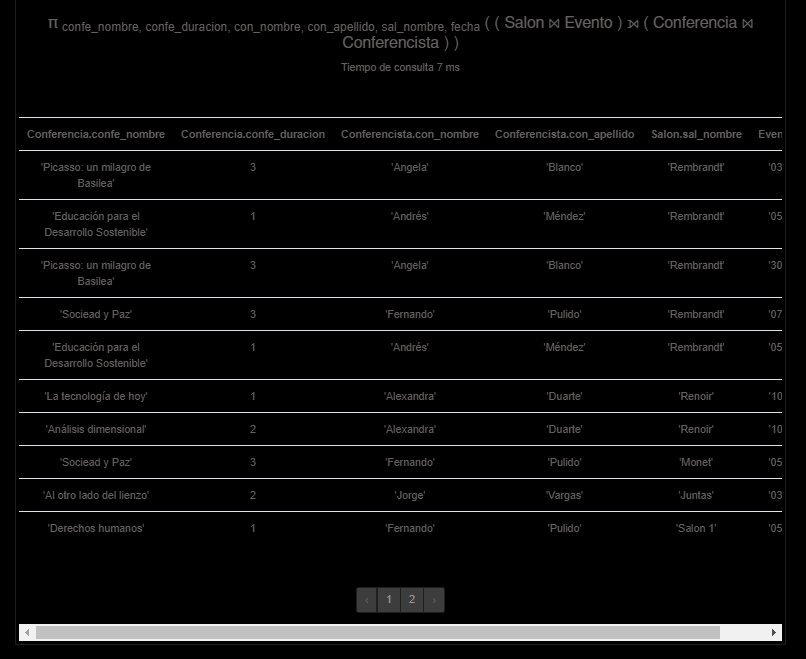
1. nombre y apellido de los conferencistas que han tenido conferencias en salones de tipo Auditorio:

*πcon\_nombre,con\_apellido((σsal\_tipo='Auditorio'(Salon⨝Evento))⨝Conferencista)*



1. nombre de la conferencia, duración, nombre y apellido del conferencista, nombre del salón y fecha por cada evento realizado:

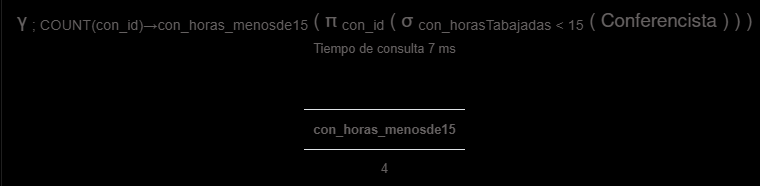
*πconfe\_nombre,confe\_duracion,con\_nombre,con\_apellido,sal\_nombre,fecha((Salon⨝Evento)⟕(Conferencia⨝Conferencista))*



Escribir una consulta para conocer:

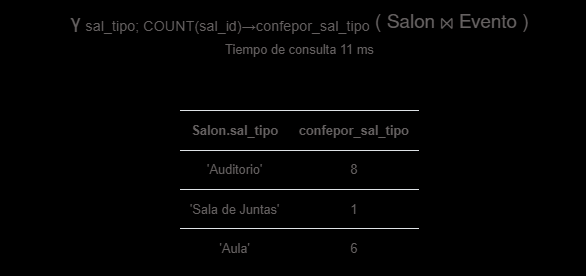
1. ¿Cuántos conferencistas tienen menos de 15 horas trabajadas?

*γ count(con\_id) -> con\_horas\_menosde15(πcon\_id(σcon\_horasTabajadas<15(Conferencista)))*



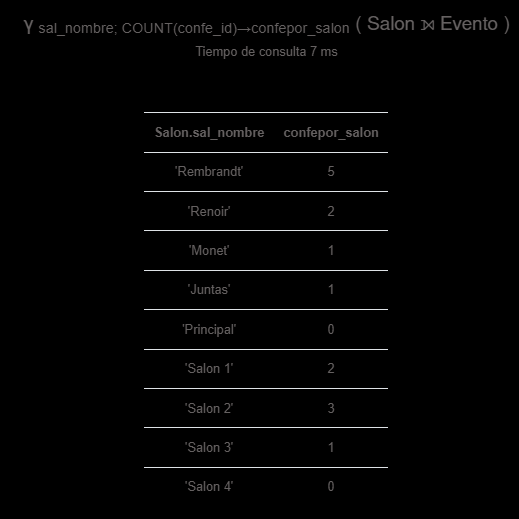
1. ¿Cuántas conferencias se han llevado a cabo por tipo de salón?

*γ sal\_tipo; count(sal\_id)→confepor\_sal\_tipo (Salon⨝Evento)*



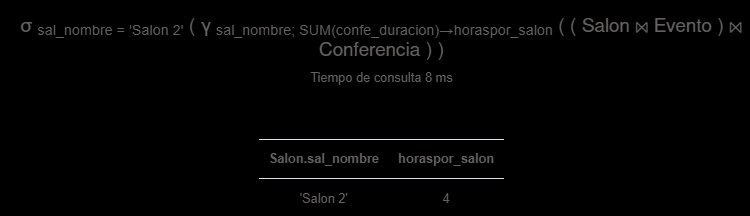
1. ¿Cuántas conferencias se han llevado a cabo por salón? Si existen salones donde no se hayan llevado a cabo eventos, estos salones deben estar reflejados en la consulta con valor de 0 en el conteo.

*γ sal\_nombre; count(confe\_id) → confepor\_salon (Salon⟕Evento)*



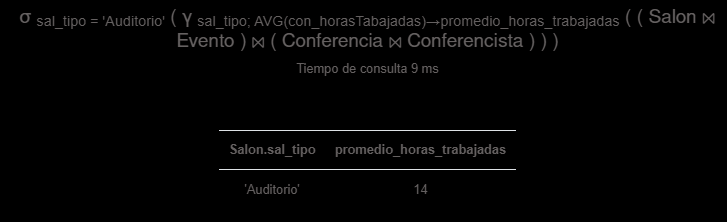
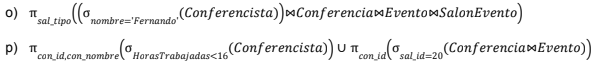
1. ¿Cuál es la ocupación (horas usadas) del salón “Salon 2”?

*σsal\_nombre='Salon 2'(γ sal\_nombre; sum(confe\_duracion) → horaspor\_salon ((Salon⨝Evento)⨝Conferencia))*



1. ¿Cuál es el promedio de las horas trabajadas de los conferencistas con exposiciones en“Auditorios”?

*σsal\_tipo='Auditorio'(γ sal\_tipo; avg(con\_horasTabajadas)→promedio\_horas\_trabajadas((Salon⨝Evento)⨝(Conferencia⨝Conferencista)))*

**1.3** De las siguientes expresiones en álgebra relacional, muestre el proceso para hallar el resultado de la expresión (paso a paso) y diga en palabras que representa.

1. Paso a paso:

**Primero.** Se seleccionan los datos del conferencista cuyo nombre es Fernando de la tabla ‘Conferencista’:

*σcon\_nombre='Fernando'(Conferencista)*

**Segundo.** Unimos el resultado de la consulta del primer paso y la tabla ‘Conferencia’ usando el natural join. Las tablas se unen en las filas donde se da la coincidencia ‘Conferencista.con\_id = Conferencia.con\_id’.El resultado de la consulta nos arroja los datos del conferencista Fernando junto con los datos de las conferencias de las cuales se encarga el mismo:

*σcon\_nombre='Fernando'(Conferencista)⨝Conferencia*

**Tercero.** Unimos el resultado de la consulta del segundo paso y la tabla ‘Evento’ usando el natural join. Las tablas se unen en las filas donde se da la coincidencia ‘Conferencia.confe\_id=Evento.confe\_id’. El resultado de esta consulta nos arroja los datos de la consulta anterior junto al id del salón y la fecha en donde se realizaron las conferencias a cargo del conferencista Fernando:

*σcon\_nombre='Fernando'(Conferencista)⨝Conferencia⨝Evento*

**Cuarto.** Unimos el resultado de la consulta del tercer paso y la tabla ‘Salon’ usando el natural join. Las tablas se unen en las filas donde se da la coincidencia ‘Evento.sal\_id = Salon.sal\_id’. El resultado de esta consulta nos arroja los datos de la consulta anterior junto al nombre y el tipo del salón en donde se realizaron las conferencias a cargo del conferencista Fernando:

*σcon\_nombre='Fernando'(Conferencista)⨝Conferencia⨝Evento⨝Salon*

**Quinto.** Finalmente proyectamos sal\_tipo para saber el tipo de salón en los cuales se llevaron a cabo las conferencias a cargo del conferencista Fernando:

*πsal\_tipo((σcon\_nombre='Fernando'(Conferencista))⨝Conferencia⨝Evento⨝Salon)*



1. Paso a paso:

**Primero.** Se seleccionan los datos de los conferencistas que hayan trabajado menos de 16 horas:

*σcon\_horasTabajadas<16(Conferencista)*

**Segundo.** Se proyectan el Id y el nombre de los conferencistas que nos resultaron de la consulta anterior. El resultado de esta consulta son el Id y el nombre de los conferencistas que hayan trabajado menos de 16 horas:

*πcon\_id,con\_nombre(σcon\_horasTabajadas<16(Conferencista))*

**Tercero.** Por otro lado unimos las tablas ‘Conferencia’ y ‘Evento’ usando natural join donde se da la coincidencia ‘Conferencia.confe\_id = Evento.confe\_id. El resultado de esta consulta nos arroja los datos de las conferencias que fueron presentadas así como las datos de la presentación de las conferencias:

*Conferencia⨝Evento*

**Cuarto.** Luego seleccionamos de la consulta anterior los datos de las conferencias presentadas en el salón cuyo id es 20:

*σsal\_id=20(Conferencia⨝Evento)*

**Quinto.** Proyectamos la columna con\_id para obtener el Id de los conferencistas. El resultado de esta consulta es el Id de los conferencistas que se presentaron el salón cuyo Id es igual a 20:

πcon\_id(σsal\_id=20(Conferencia⨝Evento))

**Sexto.** Unimos los resultados obtenidos en el segundo y quinto paso mediante una unión. Esta unión no es posible puesto que en la consulta del segundo paso obtenemos dos columnas (con\_id,con\_nombre) y en el quinto paso obtenemos sólo una columna (con\_id):

*πcon\_id,con\_nombre(σcon\_horasTabajadas<16(Conferencista))∪πcon\_id(σsal\_id=20(Conferencia⨝Evento))*

Una forma para hacer que la unión sea posible sería modificando el tercer paso, haciendo un natural join también con la tabla ‘Conferencista’, el cuarto paso se puede dejar tal cual como está y en el quinto paso modificamos la proyección añadiendo con\_nombre:

*Conferencia⨝Evento⨝Conferencista* **\*corrección 3er paso**

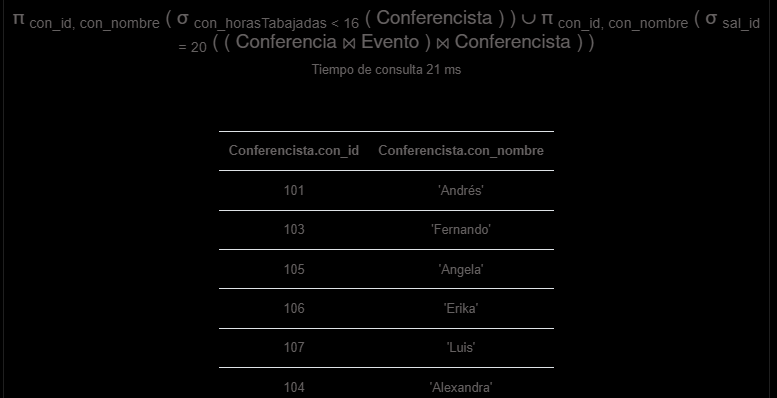
*σsal\_id=20(Conferencia⨝Evento⨝Conferencista)* ***\*4to paso***

*πcon\_id,con\_nombre(σsal\_id=20(Conferencia⨝Evento⨝Conferencista))* ***\*corrección 5to paso***

Haciendo esas correcciones el número de columnas y el tipo de datos coinciden, por tanto la unión es posible:

*πcon\_id,con\_nombre(σcon\_horasTabajadas<16(Conferencista)) ∪ πcon\_id,con\_nombre(σsal\_id=20(Conferencia⨝Evento⨝Conferencista))*

El resultado de la consulta anterior son el Id y nombre de los conferencistas que trabajaron menos de 16 horas y/o dictaron conferencias en el salón cuyo id es igual a 20.

**3.** (30 puntos) Considere la siguiente base de datos relacional: cedula\_jefe en la relación empleado es llave foránea a la relación empleado.

* empleado(cedula, nombre, calle, ciudad, cedula\_jefe)
* empresa(nit\_empresa, nombre, ciudad)
* trabaja(cedula, nit\_empresa, sueldo, cargo)

Obtener una expresión en álgebra relacional de cada una de las siguientes consultas:

1. nombre de todos los empleados y nombre de su jefe

*π nombre, jef\_nombre (empleado ⨝ empleado.cedula\_jefe=jefe.cedula\_jefe)*

1. nombre de todos los empleados que viven en la misma ciudad y en la misma calle que sus jefes

*π nombre ((σ empleado.calle=jefe.jef\_calle ^ empleado.ciudad=jefe.jef\_ciudad) (empleado ⨝ empleado.cedula\_jefe=jefe.cedula\_jefe))*

1. nombre de los empleados que viven en la misma ciudad que la compañía para la que trabajan

*π nombre (σ empleado.ciudad=empresa.ciudad (empleado ⨝ trabaja.nit\_empresa = empresa.nit\_empresa)*)

1. nombres y ciudades de residencia de todos los empleados que trabajan para el “Banco Importante”

π nombre, ciudad (σ trabaja.nit\_empresa=empresa.nit\_empresa ^ empresa.nombre="Banco Importante" (empleado ⨝ trabaja.cedula=empleado.cedula ⨝ trabaja.nit\_empresa=empresa.nit\_empresa))

1. nombre de todos los empleados que no trabajan para el “Banco Mundial”

π nombre (empleado - (π nombre (σ empresa.nombre="Banco Mundial" (empleado ⨝ trabaja.nit\_empresa=empresa.nit\_empresa))))

1. nombre y ciudad de residencia de todos los empleados que trabajan para el “Banco Importante” y ganan más de $20.000.000 anuales

π nombre, ciudad (σ trabaja.sueldo>20000000 ^ trabaja.nit\_empresa=empresa.nit\_empresa ^ empresa.nombre="Banco Importante" (empleado ⨝ trabaja.cedula=empleado.cedula ⨝ trabaja.nit\_empresa=empresa.nit\_empresa))

1. nombre de los empleados que ganan más que cualquier empleado del “Banco Pequeño”

π nombre (σ trabaja.sueldo>(σ nit\_empresa="Banco Pequeño" (ρ nit\_empresa trabaja)) (empleado ⨝ trabaja.cedula=empleado.cedula))

1. Suponga que hay oficinas de las compañías en varias ciudades y en una ciudad pueden haber varias compañías. Encontrar todas las compañías instaladas en cada ciudad en la que está instalado el “Banco DiaDia”

π nombre (σ trabaja.nit\_empresa=empresa.nit\_empresa ^ empresa.nombre="Banco DiaDia" (trabaja ⨝ empresa.nit\_empresa=trabaja.nit\_empresa))

1. nombre de la compañía con mayor número de empleados

a. π nombre (σ count(cedula) = (select max(count(cedula)) from trabaja group by nit\_empresa) (trabaja ⨝ empresa.nit\_empresa=trabaja.nit\_empresa))

1. nombre de la compañía con la nómina más reducida

π nombre (σ count(cedula) = (select min(count(cedula)) from trabaja group by nit\_empresa) (trabaja ⨝ empresa.nit\_empresa=trabaja.nit\_empresa))

1. nombre de las compañías cuyos empleados ganen un sueldo más elevado, que el sueldo medio del “Banco Importante”

π nombre (σ trabaja.sueldo > (select avg(trabaja.sueldo) from trabaja ⨝ empresa.nit\_empresa=trabaja.nit\_empresa ^ empresa.nombre="Banco Importante") (trabaja ⨝ empresa.nit\_empresa=trabaja.nit\_empresa))

Obtener una expresión en álgebra relacional de cada una de las siguientes peticiones:

1. Modificar la base de datos de manera que “Juan” viva ahora en “Bogotá”

UPDATE empleado

SET ciudad = 'Bogotá'

WHERE nombre = 'Juan'

1. Aumentar el sueldo de todos los jefes en un 2%

UPDATE trabaja

SET sueldo = sueldo \* 1.02

WHERE cedula IN (SELECT cedula\_jefe FROM empleado)

1. Aumentar el sueldo de los empleados que no son jefes en 10% a menos que el sueldo resultante sea mayor que $1'000.000. En este caso dar solo un aumento del 3%

UPDATE trabaja

SET sueldo = CASE

WHEN sueldo \* 1.1 <= 1000000 THEN sueldo \* 1.1

ELSE sueldo \* 1.03

END

WHERE cedula NOT IN (SELECT cedula\_jefe FROM empleado)

1. Borrar todas las tuplas de la relación trabajo de los empleados del “Banco Pequeño”

DELETE FROM trabaja

WHERE nit\_empresa = 'Banco Pequeño'

1. Un empleado cambia de lugar de trabajo. El empleado X deja de trabajar en una compañía y pasa a otra.

UPDATE trabaja

SET nit\_empresa = 'nuevo\_nit\_empresa'

WHERE cedula = 'cedula\_empleado\_X' AND nit\_empresa = 'antiguo\_nit\_empresa'

**4.** (10 ptos) Se tiene la relación ventas(bar, bebida, precio). Encontrar los bares que vendan dos (2) bebidas diferentes al mismo precio. Solución en álgebra relacional. Probar en ReLAX (crear datos).