**Empleados** (ide, enombre, fecha\_nacimiento, salario)

CP: ide

CNN: enombre, salario

**TrabajaEn** (ide, idd, desde, hasta)

CF: idd-> Departamentos (idd)

ide-> Empleados(ide)

**Departamentos** (idd, nombre, presupuesto, idadm)

CP: idd

CF: idadm -> Empleados(ide)

CNN: nombre, presupuesto

1. Liste nombre y salario de cada empleado que trabaje o haya trabajado en I+D y en el departamento de inovación

**SQL:**

SELECT nombre, salario FROM Empleados e

NATURAL JOIN TrabajaEn t

NATURAL JOIN Departamentos d

WHERE d.nombre=”Innovación”

INTERSECT SELECT nombre, salario FROM Empleados e

NATURAL JOIN TrabajaEn t

NATURAL JOIN Departamentos d

WHERE d.nombre=”I+D” ;

**ÁLGEBRA RELACIONAL:**

innov <- (σ nombre = “Innovación” (Departamentos) ⋈ TrabajaEn ⋈ Empleados)

id <- (σ nombre = “I+D” (Departamentos) ⋈ TrabajaEn ⋈ Empleados)

π nombre, salario (innov ∩ id)

1. Encuentre los idadm de los gerentes que administren solo presupuestos mayores a 10000000

**SQL:**

SELECT DISTINCT idadm FROM Departamentos d

WHERE NOT EXISTS

( SELECT DISTINCT idadm FROM Departamentos d

WHERE d.presupuesto<100000

**ÁLGEBRA RELACIONAL:**

π idadm (Departamentos) - π idadm(σ presupuesto<100000 (Departamentos))

aclaración: los gerentes que administren departamentos cuyos presupuestos sean mayores a 100000

1. Encuentre los nombres de los gerentes que administran los mayores presupuestos. Si un gerente administra más de un departamento, se deben sumar sus presupuestos

**SQL:**

SELECT e.enombre

FROM Empleados e

NATURAL JOIN TrabajaEn t

NATURAL JOIN Departamentos d

WHERE e.ide=d.idadm AND MAX(SUM(d.presupuestos))

**ÁLGEBRA RELACIONAL:**

no se puede

1. Busque empleados que han trabajado en **todos** los departamentos y liste nombres y salarios

**SQL:**

SELECT e.enombre, e.salario

FROM Empleados e

INNER JOIN

**ÁLGEBRA RELACIONAL:**

π nombre, salario, ide (Empleados) / π ide (TrabajaEn ⋈ Departamentos)

1. Liste los nombres y edades de los administradores que controlan más de 15000000 de presupuesto en total.

**SQL:**

SELECT e.enombre, DATEDIFF(YEAR, e.fecha\_nacimiento, GETDATE()) as edad

FROM Departamentos d

NATURAL JOIN TrabajaEn t

NATURAL JOIN Empleados e

WHERE d.idadm=e.ide

GROUP BY idadm

HAVING SUM(d.presupuesto) > 15000000

**ÁLGEBRA RELACIONAL:**

No se puede realizar en álgebra relacional

1. Listar nombres de los gerentes que solo administran presupuestos mayores a 10000000 pero al menos 1 menor a 15000000

**SQL:**

SELECT e.nombre

FROM Departamento d

NATURAL JOIN TrabajaEn t

NATURAL JOIN Empleados e

WHERE d.presupuesto<15000000

GROUP BY d.idadm

HAVING SUM(d.presupuesto) > 1000000

**ÁLGEBRA RELACIONAL:**

1. Listar nombres de empleados que empezaron en ventas en el primer semestre de 2020

**SQL:**

SELECT enombre

FROM Empleados e

NATURAL JOIN TrabajaEn t

WHERE t.desde BETWEEN “2020-01-01” AND “2020-05-31”

**ÁLGEBRA RELACIONAL:**

1. Listar de los departamentos sin administrador, nombre, presupuesto y salario promedio de los empleados actuales

**SQL:**

SELECT dnombre, d.presupuesto

AVG(e.salario) AS promedio

FROM Departamentos d

NATURAL JOIN TrabajaEn t

NATURAL JOIN Empleados e

WHERE d.idadm IS NULL

GROUP BY dnombre

**ÁLGEBRA RELACIONAL:**

No es posible resolverlo por algebra relacional, ya que no se pueden hacer operaciones aritméticas

* Un empleado puede trabajar o haber trabajado en varios departamentos
* **hasta** es null para los empleados actuales
* un departamento puede o no tener administrador y tener varios empleados y un empleado puede administrar varios departamentos