

CONSULTAS BÁSICAS:

1. Nombre de docentes del departamento de Ingeniería de Sistemas e Industrial que han tomado cursos en los últimos 3 años.

Algebra Relacional:

`Docentes que hayan cursado un programa en los ultimos 3

anos` $\leftarrow \Pi$ *apellido_doc AS Apellidos,nombre_doc AS Nombre* (

σ *year(fecha_inicio)>year(curdate())-3*(((*DOCENTE|x|PROGRAMA_DOCENTE*)|*x|PROGRAMA*)))

SQL:

CREATE OR REPLACE VIEW `Docentes que hayan cursado un programa en los ultimos 3 anos` AS

SELECT DISTINCT APELLIDO_DOC AS Apellidos, NOMBRE_DOC AS Nombre

FROM DOCENTE NATURAL JOIN PROGRAMA_DOCENTE NATURAL JOIN PROGRAMA

WHERE YEAR(FECHA_INICIO) > YEAR(CURDATE())-3

ORDER BY Apellidos;

2. Nombre de docentes adscritos a un programa y que se encuentran en el extranjero.

Algebra Relacional:

`Docentes que tengan incrito un programa en el

extranjero` $\leftarrow \Pi$ *apellido_doc AS Apellidos,nombre_doc AS Nombre,tema AS Programa,nombre_pais AS Pais* (

σ *cod_pais <> 'COL'*(((*DOCENTE|x|PROGRAMA_DOCENTE*)|*x|PROGRAMA*)|*x|INSTITUCION*)|*x|CIUDAD*)|*x|PAIS*))

SQL:

CREATE OR REPLACE VIEW `Docentes que tengan incrito un programa en el extranjero` AS

SELECT DISTINCT APELLIDO_DOC AS Apellidos, NOMBRE_DOC AS Nombre, TEMA AS Programa, NOMBRE_PAIS AS Pais

FROM DOCENTE NATURAL JOIN PROGRAMA_DOCENTE NATURAL JOIN PROGRAMA NATURAL JOIN INSTITUCION NATURAL JOIN CIUDAD NATURAL JOIN PAIS

WHERE COD_PAIS <> 'COL'

ORDER BY APELLIDO_DOC, TEMA;

3. Nombre de docentes que cursaron o cursan el programa de Aalborg.

Algebra Relacional:

`Docentes que tengan inscrito el programa Aalborg` $\leftarrow \Pi$ *nombre_doc AS Nombre,apellido_doc AS Apellidos* (

σ *tema = 'Aalborg'*(((*DOCENTE|x|PROGRAMA_DOCENTE*)|*x|PROGRAMA*)))

SQL:

CREATE OR REPLACE VIEW `Docentes que tengan inscrito el programa Aalborg` AS

SELECT DISTINCT NOMBRE_DOC AS Nombre, APELLIDO_DOC AS Apellidos

FROM DOCENTE NATURAL JOIN PROGRAMA_DOCENTE NATURAL JOIN PROGRAMA

WHERE TEMA = 'Aalborg';

- Correo de los docentes de planta que han cursado programas de enfoque pedagógico y de tipo extensión.

Algebra Relacional:

`Correo docentes con programa de

extension/pedagogia` $\leftarrow \Pi$ correo_inst AS "Correo institucional" (

$\sigma_{nombre_enfoue = 'Pedagogia' \text{ AND } nombre_tipo_prog = 'Extension' \text{ AND } nombre_tipo_doc = 'Planta'}$ (((((DOCENTE
|x|PROGRAMA_DOCENTE)|x|PROGRAMA)|x|TIPO_DOCENTE)|x|ENFOQUE)|x|TIPO_PROGRAMA)))

SQL:

**CREATE OR REPLACE VIEW `Correo docentes con programa de extension/pedagogia`
AS**

SELECT DISTINCT Correo_inst **AS** "Correo institucional"
FROM DOCENTE **NATURAL JOIN** PROGRAMA_DOCENTE **NATURAL**
JOIN PROGRAMA **NATURAL JOIN** TIPO_DOCENTE **NATURAL**
JOIN ENFOQUE **NATURAL JOIN** TIPO_PROGRAMA
WHERE Nombre_enfoque = 'Pedagogia' **AND**
Nombre_tipo_prog = 'Extension' **AND** Nombre_tipo_doc = 'Planta';

- Nombre de los profesores que tienen inscrito un programa pero no lo cursan actualmente.

Algebra Relacional:

`Docentes con programas inscritos que aún no han

visto` $\leftarrow \Pi$ nombre_doc AS Nombre, apellido_doc AS Apellidos (

$\sigma_{fecha_inicio > curdate()}$ (((DOCENTE|x|PROGRAMA_DOCENTE)|x|PROGRAMA)))

SQL:

**CREATE OR REPLACE VIEW `Docentes con programas inscritos que aun no han visto`
AS**

SELECT DISTINCT NOMBRE_DOC **AS** Nombre, APELLIDO_DOC **AS**
Apellido
FROM DOCENTE **NATURAL JOIN** PROGRAMA_DOCENTE **NATURAL**
JOIN PROGRAMA
WHERE FECHA_INICIO > CURDATE();

- Nombre de docentes de planta del departamento de Ingeniería de Sistemas e Industrial que se encuentran adscritos a un programa en Colombia.

Algebra Relacional:

`Docentes de Sistemas e Industrial en un programa`

$\leftarrow \Pi$ nombre_doc AS Nombre, apellido_doc AS Apellido (

$\sigma_{nombre_dep = 'Sistemas e industrial' \text{ AND } cod_pais = 'COL' \text{ AND } nombre_tipo_doc = 'Planta'}$ (((((((DOCENTE
|x|PROGRAMA_DOCENTE)|x|PROGRAMA)|x|INSTITUCION)|x|CIUDAD)|x|PAIS)
|x|TIPO_DOCENTE)|x|DEPARTAMENTO)))

SQL:

```
CREATE OR REPLACE VIEW `Docentes de Sistemas e Industrial en un programa` AS
SELECT DISTINCT NOMBRE_DOC AS Nombre, APELLIDO_DOC AS
Apellido
FROM DOCENTE NATURAL JOIN PROGRAMA_DOCENTE NATURAL
JOIN PROGRAMA NATURAL JOIN INSTITUCION NATURAL JOIN
CIUDAD NATURAL JOIN PAIS NATURAL JOIN TIPO_DOCENTE
NATURAL JOIN DEPARTAMENTO
WHERE Nombre_dep = 'Sistemas e Industrial' AND COD_PAIS = 'COL'
AND Nombre_tipo_doc = 'Planta';
```

7. Nombre de los docentes ocasionales que han visto un programa ofrecido por la Universidad de los Andes, programas con un costo menor a 1200000 o ambas.

Algebra Relacional:

$\pi_{\text{Nombre_doc}}(\sigma_{\text{Nombre_tipo_doc} = \text{'Ocasional'} \text{ AND } \text{Nombre_instit} = \text{'Universidad de los andes'}}(\text{INSTITUCION} \bowtie \text{PROGRAMA} \bowtie \text{PROGRAMA_DOCENTE} \bowtie \text{DOCENTE} \bowtie \text{TIPO_DOCENTE})) \cap$
 $(\pi_{\text{Nombre_doc}}(\sigma_{\text{Costo} < 1200000 \text{ AND } \text{Nombre_tipo_doc} = \text{'Ocasional'}}(\text{INSTITUCION} \bowtie \text{PROGRAMA} \bowtie \text{PROGRAMA_DOCENTE} \bowtie \text{DOCENTE} \bowtie \text{TIPO_DOCENTE})))$

SQL:

```
CREATE OR REPLACE VIEW `Docentes de Sistemas e Industrial en un programa` AS
SELECT NOMBRE_DOC
FROM INSTITUCION NATURAL JOIN PROGRAMA NATURAL JOIN
PROGRAMA_DOCENTE NATURAL JOIN DOCENTE NATURAL
JOIN TIPO_DOCENTE
WHERE NOMBRE_TIPO_DOC = 'Ocasional' AND NOMBRE_INSTIT =
'Universidad de los Andes'
UNION
(SELECT NOMBRE_DOC
FROM INSTITUCION NATURAL JOIN PROGRAMA NATURAL JOIN
PROGRAMA_DOCENTE NATURAL JOIN DOCENTE NATURAL
JOIN TIPO_DOCENTE
WHERE COSTO < 1200000 AND NOMBRE_TIPO_DOC = 'Ocasional');
```

8. Nombre de todos los docentes que han visto programas en Google y Harvard.

Algebra Relacional:

$\pi_{\text{Nombre_doc}}(\sigma_{\text{Nombre_instit} = \text{'GOOGLE'}}(\text{INSTITUCION} \bowtie \text{PROGRAMA} \bowtie \text{PROGRAMA_DOCENTE} \bowtie \text{DOCENTE})) \cap$
 $(\pi_{\text{Nombre_doc}}(\sigma_{\text{Nombre_instit} = \text{'Harvard'}}(\text{INSTITUCION} \bowtie \text{PROGRAMA} \bowtie \text{PROGRAMA_DOCENTE} \bowtie \text{DOCENTE})))$

SQL:

```
(SELECT DISTINCT NOMBRE_DOC
FROM DOCENTE NATURAL JOIN PROGRAMA_DOCENTE NATURAL
      JOIN PROGRAMA NATURAL JOIN INSTITUCION
WHERE NOMBRE_INSTIT = 'GOOGLE')
INTERSECT
(SELECT DISTINCT NOMBRE_DOC
FROM DOCENTE NATURAL JOIN PROGRAMA_DOCENTE NATURAL
      JOIN PROGRAMA NATURAL JOIN INSTITUCION)
WHERE NOMBRE_INSTIT = 'Harvard');
```

Traducción para MySQL:

```
CREATE OR REPLACE VIEW DOC_GOO AS
SELECT DISTINCT NOMBRE_DOC AS Nombre_doc
FROM DOCENTE NATURAL JOIN PROGRAMA_DOCENTE NATURAL
      JOIN PROGRAMA NATURAL JOIN INSTITUCION
WHERE NOMBRE_INSTIT = 'GOOGLE';
```

```
CREATE OR REPLACE VIEW DOC_HARV AS
SELECT DISTINCT NOMBRE_DOC AS Nombre_doc
FROM DOCENTE NATURAL JOIN PROGRAMA_DOCENTE NATURAL
      JOIN PROGRAMA NATURAL JOIN INSTITUCION
WHERE NOMBRE_INSTIT = 'Harvard';
```

```
CREATE OR REPLACE VIEW `Docentes que han visto programas
      Google/Harvard` AS
SELECT DISTINCT DOC_GOO.Nombre_doc
FROM DOC_GOO INNER JOIN DOC_HARV
USING(Nombre_doc);
```

9. Nombre de todos los docentes que hayan visto en todas las instituciones de Colombia.

Algebra Relacional:

`Docentes, en todas las instituciones de Colombia` <--

$(\Pi_{nombre_doc} \sigma_{nombre_doc \neq Nombre_doc} ((DOCENTE \bowtie |x| PROGRAMA_DOCENTE) \bowtie |x| PROGRAMA) \bowtie |x| INSTITUCION) \div$
 $\Pi_{nombre_instit} (\sigma_{cod_pais = 'COL'} (INSTITUCION))$

SQL:

```
CREATE OR REPLACE VIEW ALL_DOC_1 AS
SELECT DISTINCT NOMBRE_DOC , NOMBRE_INSTIT
FROM DOCENTE NATURAL JOIN PROGRAMA_DOCENTE NATURAL
      JOIN PROGRAMA NATURAL JOIN INSTITUCION;
```

```
CREATE OR REPLACE VIEW ALL_DOC_2 AS
SELECT DISTINCT NOMBRE_INSTIT
```

```
FROM INSTITUCION
WHERE COD_PAIS = 'COL';
```

```
CREATE OR REPLACE VIEW `Docentes, en todas las instituciones de
Colombia` AS
SELECT ALL_DOC_1.Nombre_doc
FROM ALL_DOC_1
WHERE ALL_DOC_1.NOMBRE_INSTIT IN(
    SELECT ALL_DOC_2.NOMBRE_INSTIT
    FROM ALL_DOC_2)
GROUP BY ALL_DOC_1.Nombre_doc
HAVING COUNT(*)=(
    SELECT COUNT(*)
    FROM ALL_DOC_2);
```

10. Nombre de las ciudades de colombia que no ofrecen programas.

Algebra Relacional:

`Ciudades de colombia que no ofrecen programas` \leftarrow

$(\Pi_{\text{nombre_ciudad AS Ciudad}}(\sigma_{\text{cod_pais = 'COL'}}(\text{PAIS} \bowtie \text{CIUDAD}))) -$

$\Pi_{\text{nombre_ciudad AS Ciudad}}(\sigma_{\text{cod_pais = 'COL'}}(((\text{PAIS} \bowtie \text{CIUDAD}) \bowtie \text{INSTITUCION}) \bowtie \text{PROGRAMA))))$

SQL:

```
(SELECT DISTINCT Nombre_ciudad AS Ciudad
FROM PAIS NATURAL JOIN CIUDAD
WHERE cod_pais = 'COL')
MINUS
(SELECT DISTINCT Nombre_ciudad AS Ciudad
FROM PAIS NATURAL JOIN CIUDAD NATURAL JOIN INSTITUCION
NATURAL JOIN PROGRAMA
WHERE cod_pais = 'COL');
```

Traducción para MySQL:

```
CREATE OR REPLACE VIEW CIUDADES_COL AS
SELECT DISTINCT Nombre_ciudad AS Ciudad
FROM PAIS NATURAL JOIN CIUDAD
WHERE cod_pais = 'COL';
```

```
CREATE OR REPLACE VIEW CIUDADES_COL_PROG AS
SELECT DISTINCT Nombre_ciudad AS Ciudad
FROM PAIS NATURAL JOIN CIUDAD NATURAL JOIN INSTITUCION
NATURAL JOIN PROGRAMA
WHERE cod_pais = 'COL';
```

```
CREATE OR REPLACE VIEW `Ciudades de colombia que no ofrecen
programas` AS
```

```

SELECT CIUDADES_COL.Ciudad
FROM CIUDADES_COL LEFT JOIN CIUDADES_COL_PROG ON
      CIUDADES_COL.Ciudad = CIUDADES_COL_PROG.Ciudad
WHERE CIUDADES_COL_PROG.Ciudad IS NULL;

```

FUNCIONES AGREGADAS(2 HAVING):

1. Cantidad de dinero invertido en programas por departamento. Donde el dinero invertido sea mayor a 5200000

Algebra Relacional:

`Cantidad de dinero invertido por dep. Mayor a 5.2M` $\leftarrow \sigma_{Dinero_invertido > 5200000} ($
nombre_dep \bowtie $\rho_{sum(costo) as dinero_invertido} ($
DEPARTAMENTO \bowtie $\rho_{DOCENTE} \bowtie$ $\rho_{PROGRAMA_DOCENTE} \bowtie$ $\rho_{PROGRAMA}))$

SQL:

```

CREATE OR REPLACE VIEW `Cantidad de dinero invertido por dep.
Mayor a 5.2M` AS
SELECT NOMBRE_DEP, SUM(COSTO) AS DINERO_INVERTIDO
FROM DEPARTAMENTO NATURAL JOIN DOCENTE NATURAL JOIN
      PROGRAMA_DOCENTE NATURAL JOIN PROGRAMA
GROUP BY NOMBRE_DEP
HAVING DINERO_INVERTIDO > 5200000;

```

2. Cantidad de programas acreditados por país.

Algebra Relacional:

`Cantidad de programas acreditados por país` $\leftarrow \rho_{nombre_pais} \bowtie \rho_{count(nombre_pais)} ($
 $\sigma_{acreditacion = 1} ($
PROGRAMA \bowtie $\rho_{INSTITUCION} \bowtie$ $\rho_{CIUDAD} \bowtie$ $\rho_{PAIS}))$

SQL:

```

CREATE OR REPLACE VIEW `Cantidad de programas acreditados por
pais` AS
SELECT NOMBRE_PAIS, COUNT(*) AS Cantidad
FROM PROGRAMA NATURAL JOIN INSTITUCION NATURAL JOIN
      CIUDAD NATURAL JOIN PAIS
WHERE ACREDITACION = 1
GROUP BY NOMBRE_PAIS;

```

3. Promedio de dinero invertido para profesores ocasionales.

Algebra Relacional:

`Promedio invertido por Docentes Ocasionales` $\leftarrow \rho_{avg(costo) as promedio} ($
 $\sigma_{Nombre_tipo_doc = 'Ocasional'} ($
TIPO_DOCENTE \bowtie $\rho_{DOCENTE} \bowtie$ $\rho_{PROGRAMA_DOCENTE} \bowtie$ $\rho_{PROGRAMA}))$

SQL:

```

CREATE OR REPLACE VIEW `Promedio invertido por Docentes
Ocasionales` AS
SELECT ROUND(AVG(COSTO), 2) AS PROMEDIO
FROM TIPO_DOCENTE NATURAL JOIN DOCENTE NATURAL JOIN
PROGRAMA_DOCENTE NATURAL JOIN PROGRAMA
WHERE NOMBRE_TIPO_DOC = 'Ocasional';

```

4. Relación porcentual entre el costo promedio de programas de Colombia vs el costo promedio de programas de EU.

Algebra Relacional:

```

COSTO_PROMEDIO_COL ←  $\rho_{avg(costo) as Promedio (\sigma_{cod\_pais = 'COL' (PROGRAMA | x | INSTITUCION | x | CIUDAD | x | PAIS))}$ 
COSTO_PROMEDIO_USA ←  $\rho_{avg(costo) as Promedio (\sigma_{cod\_pais = 'USA' (PROGRAMA | x | INSTITUCION | x | CIUDAD | x | PAIS))}$ 

```

`Relacion porcentual entre el costo promedio de COL y USA` $\leftarrow \Pi_{COSTO_PROMEDIO_COL.Promedio, COSTO_PROMEDIO_USA.Promedio} (\sigma_{cod_pais = 'COL'} (COSTO_PROMEDIO_COL \times COSTO_PROMEDIO_USA))$

SQL:

```

CREATE OR REPLACE VIEW COSTO_PROMEDIO_COL AS
SELECT AVG(COSTO) AS Promedio
FROM PROGRAMA NATURAL JOIN INSTITUCION NATURAL JOIN
CIUDAD NATURAL JOIN PAIS
WHERE cod_pais = 'COL';

```

```

CREATE OR REPLACE VIEW COSTO_PROMEDIO_EU AS
SELECT AVG(COSTO) AS Promedio
FROM PROGRAMA NATURAL JOIN INSTITUCION NATURAL JOIN
CIUDAD NATURAL JOIN PAIS
WHERE cod_pais = 'USA';

```

```

CREATE OR REPLACE VIEW `Relacion porcentual entre el costo promedio
de COL y USA` AS
SELECT
ROUND(COSTO_PROMEDIO_COL.Promedio/COSTO_PROMEDIO_EU.Promedio*
100, 2) AS "Relacion Porcentual"

FROM COSTO_PROMEDIO_COL, COSTO_PROMEDIO_EU;

```

5. Cantidad de programas que se estén cursando actualmente por institución, para instituciones con más de 2 programas.

Algebra Relacional:

```

`Cantidad de Programas por Institucion, con cantidad > 2`  $\leftarrow \sigma_{Cantidad \geq 2} (nombre\_instit as institucion \rho_{count(institucion) as cantidad} (\sigma_{year(fecha\_finalizacion) \geq 2018} ($ 

```

INSTITUCION|x|PROGRAMA)))

SQL:

CREATE OR REPLACE VIEW `Cantidad de Programas por Institucion,
con cantidad > 2` AS

SELECT NOMBRE_INSTIT AS Institucion, COUNT(*) AS Cantidad
FROM INSTITUCION NATURAL JOIN PROGRAMA
WHERE YEAR(FECHA_FINALIZACION) >= 2018
GROUP BY NOMBRE_INSTIT
HAVING Cantidad >= 2;

FUNCIONES ANIDADAS (EXISTS):

1. Nombre de países que no ofrecen cursos de educación docente.

SQL:

CREATE OR REPLACE VIEW PAIS_NO_CURSOS AS
SELECT NOMBRE_PAIS
FROM PAIS
WHERE NOMBRE_PAIS NOT IN (
SELECT NOMBRE_PAIS
FROM ((PAIS NATURAL JOIN CIUDAD) NATURAL JOIN
INSTITUCION));

2. Nombre de docentes que no están adscritos a un programa.

SQL:

CREATE OR REPLACE VIEW DOCENTE_NO_DOCENTE AS
SELECT NOMBRE_DOC
FROM DOCENTE
WHERE NOMBRE_DOC NOT IN (
SELECT NOMBRE_DOC
FROM DOCENTE NATURAL JOIN PROGRAMA_DOCENTE
NATURAL JOIN PROGRAMA);

3. Departamentos con docentes vinculados.

SQL:

CREATE OR REPLACE VIEW `Departamentos sin docentes` AS
SELECT DISTINCT NOMBRE_DEP
FROM DEPARTAMENTO NATURAL JOIN DOCENTE
WHERE EXISTS (
SELECT NOMBRE_DEP FROM DEPARTAMENTO);