

**UNIVERSIDAD TECNOLÓGICA  
NACIONAL FACULTAD  
REGIONAL RESISTENCIA**



**Base de Datos**

**Entrega Final  
Grupo 13**

**Integrantes:**

**Espindola Agustín  
Judkevich Natasha Nicole  
Maldonado Leandro Arian  
Leguizamon Sofia Violeta  
Ramirez Roshdestwensky Alexis**

# Índice

➤ <b>Consignas Etapa 1</b> .....	<b>2</b>
Sistema gestor de bases de datos:.....	2
Análisis del escenario y modelado:.....	2
Esquema relacional:.....	2
Esquema físico:.....	2
Recomendaciones sobre los esquemas.....	3
Base de Datos:.....	3
➤ <b>Consignas Etapa 2</b> .....	<b>3</b>
➤ <b>Etapa 1</b> .....	<b>4</b>
Sistemas Gestor de Datos.....	4
En este trabajo utilizaremos MySQL como SGBD.....	4
Desventaja de los SGBD (Personal Calificado).....	5
Dominio.....	5
Entidad.....	5
➤ <b>Análisis y Modelado</b> .....	<b>6</b>
<b>Supuestos Semánticos:</b> .....	<b>6</b>
➤ <b>Scripts MySQL</b> .....	<b>7</b>
➤ <b>DICCIONARIO DE DATOS</b> .....	<b>16</b>
➤ <b>Etapa 2</b> .....	<b>21</b>
<b>INSERCIÓN, MODIFICACIÓN, BORRADO DE DATOS</b> .....	<b>21</b>
Cinco consultas distintas para inserción de datos (distintos a los ya cargados en forma masiva).....	21
Cinco consultas distintas para modificación de datos.....	23
Cinco consultas sql distintas para borrado de filas.....	24
<b>Consultas SELECT</b> .....	<b>26</b>

## ➤Consignas Etapa 1

Sistema gestor de bases de datos:

- En base a la documentación oficial del SGBD utilizado, detallar las características principales.
- Elegir cinco conceptos vistos en la unidad 1 sobre los SGBD y explicar cómo los implementa el SGBD utilizado.

Análisis del escenario y modelado:

Deberán realizarse las siguientes consignas:

- Leer el escenario en general en forma individual.
- Analizar el escenario en grupo. Deberán registrarse los ítems que no resulten claros respecto a los requerimientos planteados en el escenario y que requieran alguna consulta con el "usuario".
- Confeccionar el *diagrama entidad - relación* en base a la notación aprendida en el cursado, representando toda la semántica posible y aclarando en un apartado qué requerimientos no se pueden representar y todo lo que se considere necesario para entender correctamente el DER.

Esquema relacional:

Tomando como entrada el DER y las consideraciones hechas al mismo, se deberá realizar el esquema de la base de datos en el modelo relacional. Este esquema puede ser gráfico o bien un esquema relacional textual.

- Convertir las entidades y relaciones del DER a esquemas de relación, según la metodología practicada en clases.
- Tener en cuenta:
  - Nombres dados a las tablas/relaciones.
  - Dominio de los atributos.
  - Atributos de claves primarias y claves foráneas.
  - Los distintos tipos de restricciones (dominio, integridad, participación, etc.).
  - Manejos de integridad referencial.
- El resultado de esta actividad es un archivo que será parte de la documentación del TPI.

Esquema físico:

A partir de la actividad anterior, obtener el script sql para implementar las tablas y relaciones resultantes del modelo relacional en el SGBD instalado.

- El resultado de esta etapa debe ser un script sql que deberá ejecutarse en el SGBD.

#### Recomendaciones sobre los esquemas

- Podrá utilizarse en estas instancias algún software o herramienta CASE para facilitar el modelado (p.e. MySQL Workbench).
- En el esquema físico tener en cuenta los nombres de tablas y campos, por ejemplo, el domicilio de un cliente no conviene llamarlo “Domicilio Cliente” sino “DomCli” (o similar). También prestar especial atención a los tipos de datos y sus características, justificando adecuadamente cada elección.

#### Base de Datos:

Una vez creada la base de datos con todos los objetos planteados en el esquema físico, realizar las siguientes consignas:

- Cargar datos en las tablas para que todas tengan al menos una fila. Al menos dos tablas deben tener más de 10 filas.
- Utilizando alguna herramienta para generación de datos, importar masivamente filas a las tablas indicadas. Tener en cuenta que se intenta trabajar con volúmenes importantes similares a un sistema real, por lo que deberá asegurarse que al menos una tabla cuente con más de 100.000 registros.

## ➤Consignas Etapa 2

Las consultas deberán ejecutarse sobre la base de datos cargada con datos suficientes para que las mismas muestren resultados pertinentes. Si los requerimientos solicitados implican replantear el esquema y hacer alguna modificación, o bien hacer alguna modificación en los datos, dejar constancia de esto.

Responder mediante sentencias sql las siguientes consignas: INSERCIÓN, MODIFICACIÓN, BORRADO DE DATOS

Una vez cargadas las tablas con datos, realizar a criterio del grupo:

- cinco consultas distintas para inserción de datos (distintos a los ya cargados en forma masiva),
- cinco consultas distintas para modificación de datos,
- cinco consultas sql distintas para borrado de filas.

Las consultas pueden ser ejecutadas sobre una misma tabla o distintas, pero todas deben cumplir condiciones medianamente complejas para ejecutarse (por ej. buscar valores en otra tabla relacionada mediante una subconsulta).

#### Consultas SELECT

- 1) Listado de docentes que viven en una provincia distinta de aquella en la que trabajan.
- 2) Listado de docentes que poseen títulos de posgrado y no realizan tareas de investigación.
- 3) Informar promedio de edad de los docentes que poseen más de 10 años de antecedentes como docentes.
- 4) Listar DNI y nombre de los docentes que presentaron más de un cargo docente en las declaraciones juradas de los últimos 3 años
- 5) Listado de docentes cuya carga horaria superan las 20 horas semanales, en función de la última declaración jurada presentada.
- 6) Apellido y nombre de aquellos docentes que poseen la máxima cantidad de cargos docentes actualmente. (La cantidad de cargos surge de sumar todos los cargos docentes que se ejercen - suma de cargos docentes de la última declaración jurada -. Una vez que se sabe la cantidad de cargos por docente se puede averiguar cuál es la máxima cantidad y seguidamente los docentes que tienen esa máxima cantidad). No nos interesa las horas.
- 7) Listado de docentes solteros/as (sin esposa/o e/o hijos a cargo en la obra social).
- 8) Cantidad de docentes cuyos hijos a cargo son todos menores de 10 años.
- 9) Informar aquellos docentes que posean alguna persona del grupo familiar a cargo en la obra social que no es beneficiario del seguro de vida obligatorio.
- 10) Informar Cantidad de individuos asegurados por provincia.

## ➤Etapa 1

### Sistemas Gestor de Datos

En este trabajo utilizaremos MySQL como SGBD.

MySQL es un Sistema de Gestión de Bases de Datos (SGBD) relacional de código abierto que se utiliza para almacenar y gestionar grandes cantidades de datos de manera eficiente. Como SGBD, MySQL proporciona un entorno para crear, modificar y administrar bases de datos relacionales, donde los datos se organizan en tablas con filas y columnas.

Una de las características principales de MySQL es su capacidad para ejecutar consultas complejas utilizando el lenguaje estándar de consultas SQL (Structured Query Language). SQL permite a los usuarios realizar una variedad de operaciones en la base de datos, como insertar, actualizar, eliminar y consultar datos.

Bloqueo

MySQL utiliza diferentes tipos de bloqueos para controlar el acceso concurrente a los datos:

1. Bloqueos de Lectura (Read Locks): Permiten que varias conexiones puedan leer los mismos datos simultáneamente, pero evitan que otras conexiones realicen modificaciones en esos datos hasta que se libere el bloqueo de lectura.
2. Bloqueos de Escritura (Write Locks): Permiten que solo una conexión a la vez pueda modificar los datos protegidos por el bloqueo de escritura. Los bloqueos de escritura evitan que otras conexiones realicen tanto lecturas como escrituras en los datos bloqueados.

#### Desventaja de los SGBD (Personal Calificado)

Diseñar una base de datos eficiente y escalable requiere un conocimiento profundo de los principios de diseño de bases de datos relacionales, así como una comprensión de las necesidades específicas del proyecto. Un DBA (Administrador de Base de Datos) calificado puede diseñar esquemas de bases de datos que optimicen el rendimiento y la eficiencia de la aplicación.

#### Dominio

El término “Dominio” en el contexto de MySQL se refiere a un conjunto de valores que pueden aparecer en una columna y la cantidad de veces que aparecen. En otras palabras, es el conjunto de valores únicos dentro de una columna específica de una tabla.

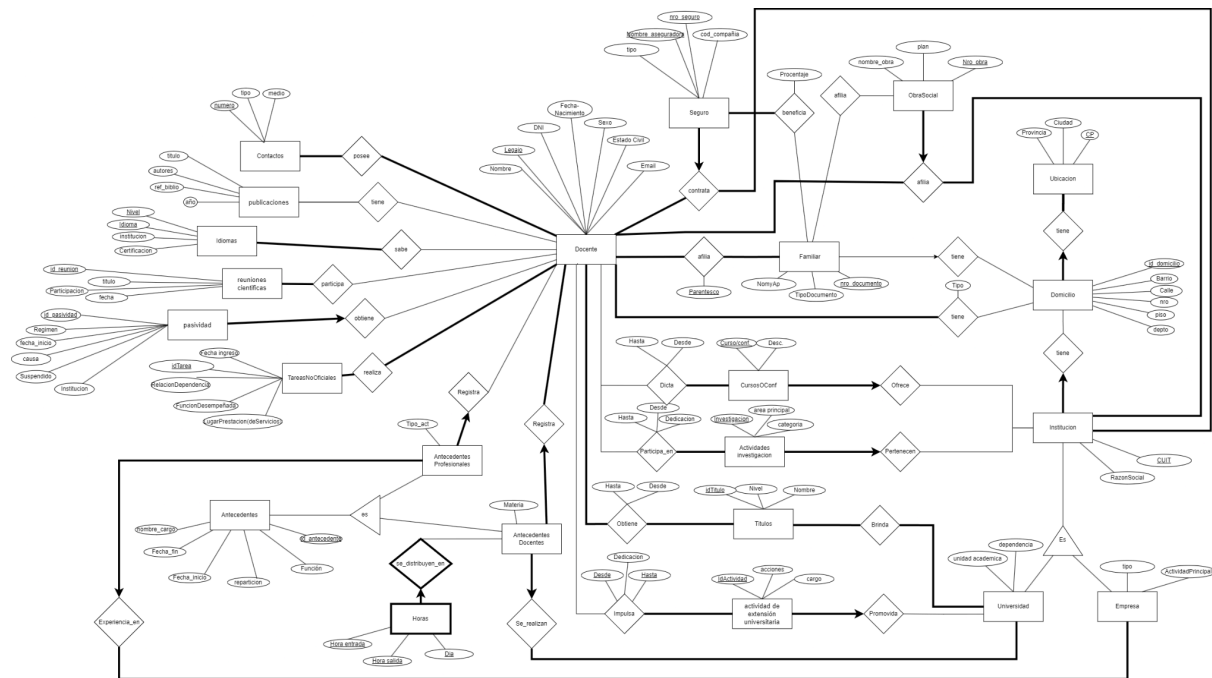
#### Entidad

En MySQL, el concepto de "entidad" se relaciona directamente con el modelo de datos relacional y se refiere a una tabla individual dentro de una base de datos y cada fila en una tabla representa una instancia única de la entidad que está modelando. En otras palabras, una entidad es una representación de un conjunto específico de objetos o conceptos del mundo real dentro del contexto de la base de datos.

#### Ventajas SGBD(Integridad y seguridad de datos)

Seguridad: MySQL ofrece una serie de características de seguridad, incluyendo autenticación basada en contraseñas, encriptación de datos en reposo y en tránsito, y control de acceso a nivel de usuario y privilegio. Esto ayuda a proteger los datos sensibles contra accesos no autorizados y garantiza la integridad de la información.

## ➤ Análisis y Modelado



## Supuestos Semánticos:

**SS1:Obra social y seguro pueden tener un grupo familiar distinto, por lo tanto el docente podrá tener más de un grupo familiar.**

**SS2: un mismo contacto puede ser de varios, pueden ser dos docentes que estén casados y compartan algún número, lo mismo con domicilio.**

**SS3: idtitulo identifica un título para cada universidad.**

**SS4:Por ley un trabajador en relación de dependencia debe tener al menos una obra social.**

**SS5:En los antecedentes se guardan hasta el trabajo actual al momento de rellenar los formularios.**

**SS6:No se necesitan vincular publicaciones y reuniones con alguna institución.**

**SS7:El docente puede tener más de una obra social.**

**SS8:Las dependencias se plantean como atributo de universidad.**

**SS9:En antecedentes se encuentran los cargos.**

**SS10: Por obra social es posible no tener asociado el domicilio de algún familiar.**

**SS11: Una de las dos relaciones de domicilio con docente o familiar, existe.**

**SS12: Una de las 3 relaciones de domicilio con docente, familiar o institución existe.**

**SS13: En docente\_tiene\_domicilio, tipo puede ser 'laboral','personal' o 'otro'.**

**SS14: En Contactos, tipo puede ser 'laboral','personal','otro', y medio puede ser 'telefono','celular' o 'mail'.**

## ➤ Scripts MySQL

```
CREATE TABLE Docente (  
    legajo int PRIMARY KEY,  
    DNI int,  
    nombre VARCHAR(100) NOT NULL,  
    estado_civil VARCHAR(50),  
    email VARCHAR(100),  
    sexo ENUM('M', 'F'),  
    fecha_nacimiento DATE  
);
```

```
CREATE TABLE contactos (  
    numero BIGINT PRIMARY KEY,  
    tipo VARCHAR(50),  
    medio VARCHAR(100)  
);
```

```
CREATE TABLE DocentePoseeContactos (  
    legajo INT,  
    numero BIGINT,  
    PRIMARY KEY (legajo, numero),  
    FOREIGN KEY (legajo) REFERENCES docente(legajo),  
    FOREIGN KEY (numero) REFERENCES contactos(numero)  
);
```

```
CREATE TABLE Publicaciones (  
    titulo VARCHAR(255) PRIMARY KEY,  
    año INT,  
    ref_biblio VARCHAR(255)  
);
```



```

CREATE TABLE DocenteTienePublicaciones (
    legajo INT,
    titulo VARCHAR(255),
    PRIMARY KEY (legajo, titulo),
    FOREIGN KEY (legajo) REFERENCES docente(legajo),
    FOREIGN KEY (titulo) REFERENCES publicaciones(titulo)
);

CREATE TABLE Idiomas (
    id_idioma int primary key,
    nivel VARCHAR(3),
    idioma VARCHAR(20),
    institucion VARCHAR(50),
    certificacion varchar(50)
);

CREATE TABLE DocenteSabeIdiomas (
    legajo INT,
    id_idioma int,
    PRIMARY KEY (legajo,id_idioma),
    FOREIGN KEY (id_idioma) REFERENCES idiomas(id_idioma),
    FOREIGN KEY (legajo) REFERENCES docente(legajo)
);

CREATE TABLE ReunionesCientificas (
    id_reunion INT PRIMARY KEY,
    titulo VARCHAR(100),
    fecha DATE,
    participacion VARCHAR(30)
);

CREATE TABLE DocenteParticipaReuniones (
    legajo INT,
    id_reunion INT,
    PRIMARY KEY (legajo, id_reunion),
    FOREIGN KEY (legajo) REFERENCES docente(legajo),

```

```
        FOREIGN KEY (id_reunion) REFERENCES
ReunionesCientificas(id_reunion)
);
```

```
CREATE TABLE PasividadObtenida (
    id_pasividad INT PRIMARY KEY,
    regimen VARCHAR(50),
    fecha_inicio DATE,
    suspendido BOOLEAN,
    causa VARCHAR(255),
    legajo INT not null,
    FOREIGN KEY (legajo) REFERENCES Docente(legajo)
);
```

```
CREATE TABLE Ubicacion(
    CP int PRIMARY KEY,
    Provincia varchar(50),
    ciudad varchar(50)
);
```

```
CREATE TABLE Domicilio (
    id_domicilio INT PRIMARY KEY,
    barrio VARCHAR(100),
    calle VARCHAR(100),
    numero INT,
    piso INT,
    depto VARCHAR(50),
    cp INT not null,
    FOREIGN KEY (CP) REFERENCES Ubicacion(CP)
);
```

```
CREATE TABLE Institucion (
    CUIT BIGINT PRIMARY KEY,
    razon_social VARCHAR(100),
    id_domicilio INT,
    FOREIGN KEY (id_domicilio) REFERENCES Domicilio(id_domicilio)
);
```

```

CREATE TABLE Empresa(
    cuit BIGINT PRIMARY KEY,
    ActividadPrincipal VARCHAR(100),
    tipo VARCHAR(10),
    FOREIGN KEY (cuit) REFERENCES Institucion(cuit)
);

```

```

CREATE TABLE Universidad(
    cuit BIGINT PRIMARY KEY,
    dependencia VARCHAR(100),
    unidad_academica VARCHAR(50),
    FOREIGN KEY (cuit) REFERENCES Institucion(cuit)
);

```

```

CREATE TABLE TareasNoOficiales (
    id_tarea INT PRIMARY KEY,
    fecha_inicio DATE,
    RelacionDependencia VARCHAR(50),
    FuncionDesempeñada VARCHAR(50),
    institucion bigint,
    FOREIGN KEY (institucion) REFERENCES Institucion(CUIT)
);

```

```

CREATE TABLE DocenteRealizaTareaNoOficiales(
    legajo INT,
    id_tarea INT,
    PRIMARY KEY (legajo, id_tarea),
    FOREIGN KEY ( legajo) REFERENCES Docente(legajo),
    FOREIGN KEY (id_tarea) REFERENCES
TareasNoOficiales(id_Tarea)
);

```

```

CREATE TABLE Antecedentes (
    id_antecedentes INT PRIMARY KEY,
    id_docentes INT not null,

```

```

    nombre_cargo VARCHAR(200),
    funcion VARCHAR(255),
    reparticion INT,
    fecha_inicio DATE ,
    fecha_fin DATE,
    FOREIGN KEY (id_docentes) REFERENCES Docente(Legajo)
);

CREATE TABLE Antecedentes_Docentes(
    id_antecedentes INT PRIMARY KEY,
    materia VARCHAR(200),
    id_corresponde BIGINT NOT NULL,
    FOREIGN KEY (id_antecedentes) REFERENCES
Antecedentes(id_antecedentes),
    FOREIGN KEY (id_corresponde) REFERENCES Universidad(CUIT)
);

CREATE TABLE Antecedentes_Profesionales (
    id_antecedentes INT PRIMARY KEY,
    tipo_act VARCHAR(200),
    id_corresponde BIGINT NOT NULL,
    FOREIGN KEY (id_antecedentes) REFERENCES
Antecedentes(id_antecedentes),
    FOREIGN KEY (id_corresponde) REFERENCES Empresa(CUIT)
);

CREATE TABLE Horas (
    id_hora int primary key,
    hora_entrada TIME,
    hora_salida TIME,
    dia TINYINT,
    id_antecedentes INT NOT NULL,
    FOREIGN KEY (id_antecedentes) REFERENCES
Antecedentes_Docentes (id_antecedentes) ON DELETE CASCADE
);

CREATE TABLE Seguro (
    nro_seguro INT PRIMARY KEY,

```

```

    nombre_aseguradora VARCHAR(100),
    cod_compania INT,
    tipo VARCHAR(50),
    Legajo INT NOT NULL,
    CUIT BIGINT NOT NULL,
    FOREIGN KEY (legajo) REFERENCES Docente(Legajo),
    FOREIGN KEY (CUIT) REFERENCES Institucion(CUIT)
);

```

```

CREATE TABLE Obra_Social (
    Nro_Obra INT PRIMARY KEY,
    Nombre_obra VARCHAR (30),
    Plan VARCHAR (60),
    Legajo INT NOT NULL,
    CUIT BIGINT NOT NULL,
    FOREIGN KEY (Legajo) REFERENCES Docente(Legajo),
    FOREIGN KEY (CUIT) REFERENCES Institucion(CUIT)
);

```

```

CREATE TABLE Familiar (
    nro_documento INT PRIMARY KEY,
    NombreCompleto VARCHAR(100),
    tipo_documento VARCHAR(50),
    id_Domicilio INT,
    fecha_nacimiento DATE,
    FOREIGN KEY (id_domicilio) REFERENCES
Domicilio(id_domicilio)
);

```

```

CREATE TABLE DocenteAfiliaFamiliar(
    nro_documento INT,
    parentesco VARCHAR(50),
    Legajo INT,
    PRIMARY KEY(nro_documento,legajo),
    FOREIGN KEY (Legajo) REFERENCES Docente (Legajo),
    FOREIGN KEY (nro_documento) REFERENCES Familiar
(nro_documento)

```

);

```
CREATE TABLE SeguroBeneficiaFamiliar(  
    nro_seguro INT,  
    nro_documento INT,  
    porcentaje REAL,  
    PRIMARY KEY(nro_seguro,nro_documento),  
    FOREIGN KEY (nro_seguro) REFERENCES Seguro(nro_seguro),  
    FOREIGN KEY (nro_documento) REFERENCES  
Familiar(nro_documento)  
);
```

```
CREATE TABLE ObraAfiliaFamiliar(  
    nro_obra INT,  
    nro_documento INT,  
    PRIMARY KEY(nro_obra,nro_documento),  
    FOREIGN KEY (nro_obra) REFERENCES obra_social(nro_obra),  
    FOREIGN KEY (nro_documento) REFERENCES  
Familiar(nro_documento)  
);
```

```
CREATE TABLE DocenteTieneDomicilio (  
    legajo INT,  
    id_domicilio INT,  
    tipo VARCHAR(50),  
    Primary key(legajo,id_domicilio),  
    FOREIGN KEY (legajo) REFERENCES Docente(Legajo),  
    FOREIGN KEY (id_domicilio) REFERENCES  
Domicilio(id_domicilio)  
);
```

```
CREATE TABLE CursoOconf (  
    curso_Conf VARCHAR(50) PRIMARY KEY,  
    descripcion VARCHAR(100),  
    cuit BIGINT NOT NULL,  
    FOREIGN KEY (cuit) REFERENCES Institucion(CUIT)  
);
```

```

CREATE TABLE DocenteDictaConferencia(
    curso_Conf VARCHAR(50),
    legajo INT,
    desde DATE,
    hasta DATE,
    primary key (curso_Conf,legajo),
    foreign key (curso_conf) references
Curso0Conf(curso_conf),
    foreign key (legajo) references Docente(legajo)
);

```

```

CREATE TABLE ActividadesInvestigacion (
    investigacion INT PRIMARY KEY,
    area_principal VARCHAR(100),
    categoria VARCHAR(50),
    institucion BIGINT NOT NULL,
    FOREIGN KEY (institucion) REFERENCES Institucion(CUIT)
);

```

```

CREATE TABLE DocenteParticipaInvestigacion(
    Legajo INT,
    investigacion INT,
    desde DATE,
    hasta DATE,
    dedicación int,
    PRIMARY KEY(Legajo,investigacion),
    FOREIGN KEY (Legajo) REFERENCES Docente(Legajo),
    FOREIGN KEY (investigacion) REFERENCES
ActividadesInvestigacion(Investigacion)
);

```

```

CREATE TABLE Titulos (
    idtitulo INT PRIMARY KEY,
    nivel VARCHAR(50),
    nombre VARCHAR(100)
);
CREATE TABLE DocenteObtieneTitulo(

```

```

        idTitulo INT,
        Legajo INT,
        Desde DATE,
        Hasta DATE,
        PRIMARY KEY(idTitulo,Legajo),
        FOREIGN KEY(idTitulo) REFERENCES Titulos(idtitulo),
        FOREIGN KEY(Legajo) REFERENCES Docente(Legajo)
    );
CREATE TABLE UniversidadBrindaTitulo(
    idTitulo INT,
    Universidad BIGINT,
    PRIMARY KEY(idTitulo,Universidad),
    FOREIGN KEY(idTitulo) REFERENCES Titulos(idtitulo),
    FOREIGN KEY(Universidad) REFERENCES Universidad(CUIT)
);
CREATE TABLE ActividadesExtensionUniversitaria (
    idActividad INT PRIMARY KEY,
    acciones VARCHAR(255),
    cargo VARCHAR(50),
    CUIT BIGINT,
    FOREIGN KEY (CUIT) REFERENCES Universidad(CUIT)
);

CREATE TABLE docenteImpulsaActividadExt (
desde DATE,
hasta DATE ,
id_actividad_ext INT,
Legajo INT,
dedicación int,
PRIMARY KEY( id_actividad_ext,Legajo ),
FOREIGN KEY (Legajo) REFERENCES Docente (Legajo),
FOREIGN KEY (id_actividad_ext) REFERENCES
ActividadesExtensionUniversitaria (idActividad)
);

```



#	Time	Action	Message
1	08:08:32	CREATE TABLE Docente ( legajo int PRIMARY KEY, DNI int, nombre VARCHAR(100) NOT NULL, estado_civil VARCHAR(50), em...	0 row(s) affected
2	08:08:32	CREATE TABLE contactos ( numero INT PRIMARY KEY, tipo VARCHAR(50), medio VARCHAR(50) )	0 row(s) affected
3	08:08:32	CREATE TABLE DocentePoseeContactos ( legajo INT, numero INT, PRIMARY KEY (legajo, numero), FOREIGN KEY (legajo) REFER...	0 row(s) affected
4	08:08:33	CREATE TABLE Publicaciones ( titulo VARCHAR(255) PRIMARY KEY, año INT, ref_biblio VARCHAR(255) )	0 row(s) affected
5	08:08:33	CREATE TABLE DocenteTienePublicaciones ( legajo INT, titulo VARCHAR(255), PRIMARY KEY (legajo, titulo), FOREIGN KEY (legajo) R...	0 row(s) affected
6	08:08:33	CREATE TABLE Idiomas ( id_idioma int primary key, nivel VARCHAR(3), idioma VARCHAR(20), institucion VARCHAR(50), certificacio...	0 row(s) affected
7	08:08:33	CREATE TABLE DocenteSabIdiomas ( legajo INT, id_idioma int, PRIMARY KEY (legajo, id_idioma), FOREIGN KEY (id_idioma) REFER...	0 row(s) affected
8	08:08:33	CREATE TABLE ReunionesCientificas ( id_reunion INT PRIMARY KEY, titulo VARCHAR(100), fecha DATE, participacion VARCHAR(3...	0 row(s) affected
9	08:08:33	CREATE TABLE DocenteParticipaReuniones ( legajo INT, id_reunion INT, PRIMARY KEY (legajo, id_reunion), FOREIGN KEY (legajo) ...	0 row(s) affected
10	08:08:33	CREATE TABLE PasividadObtenida ( id_pasividad INT PRIMARY KEY, regimen VARCHAR(50), fecha_inicio DATE, suspendido BOOL...	0 row(s) affected
11	08:08:33	CREATE TABLE Ubicacion( CP int PRIMARY KEY, Provincia varchar(50), ciudad varchar(50) )	0 row(s) affected
12	08:08:33	CREATE TABLE Domicilio ( id_domicilio INT PRIMARY KEY, barrio VARCHAR(100), calle VARCHAR(100), numero INT, piso INT, ...	0 row(s) affected
13	08:08:33	CREATE TABLE Institucion ( CUIT INT PRIMARY KEY, razon_social VARCHAR(100), id_domicilio INT, FOREIGN KEY (id_domicilio) REF...	0 row(s) affected
14	08:08:33	CREATE TABLE Empresa( cuit INT PRIMARY KEY, ActividadPrincipal VARCHAR(100), tipo VARCHAR(10), FOREIGN KEY (cuit) REF...	0 row(s) affected
15	08:08:33	CREATE TABLE Universidad( cuit INT PRIMARY KEY, dependencia VARCHAR(100), unidad_academica VARCHAR(50), FOREIGN K...	0 row(s) affected
16	08:08:33	CREATE TABLE TareasNoOficiales ( id_tarea INT PRIMARY KEY, fecha_inicio DATE, RelacionDependencia VARCHAR(50), Funcion...	0 row(s) affected
17	08:08:33	CREATE TABLE DocenteRealizaTareaNoOficiales( legajo INT, id_tarea INT, PRIMARY KEY (legajo, id_tarea), FOREIGN KEY (legajo) ...	0 row(s) affected
18	08:08:33	CREATE TABLE Antecedentes ( id_antecedentes INT PRIMARY KEY, id_docentes INT not null, nombre_cargo VARCHAR(200), funci...	0 row(s) affected
19	08:08:33	CREATE TABLE Antecedentes_Docentes( id_antecedentes INT PRIMARY KEY, materia VARCHAR(200), id_corresponde INT NOT NULL...	0 row(s) affected
20	08:08:34	CREATE TABLE Antecedentes_Profesionales ( id_antecedentes INT PRIMARY KEY, tipo_act VARCHAR(200), id_corresponde INT NOT...	0 row(s) affected
21	08:08:34	CREATE TABLE Horas ( id_hora int primary key, hora_entrada TIME, hora_salida TIME, dia TINYINT, id_antecedentes INT NOT NU...	0 row(s) affected
22	08:08:34	CREATE TABLE Seguro ( nro_seguro INT PRIMARY KEY, nombre_aseguradora VARCHAR(100), cod_compania INT, tipo VARCHAR(...	0 row(s) affected
23	08:08:34	CREATE TABLE Obra_Social ( Nro_Obra INT PRIMARY KEY, Nombre_obra VARCHAR (30), Plan VARCHAR (60), Legajo INT NOT N...	0 row(s) affected
24	08:08:34	CREATE TABLE Familiar ( nro_documento INT PRIMARY KEY, NombreCompleto VARCHAR(100), tipo_documento VARCHAR(50), id...	0 row(s) affected
25	08:08:34	CREATE TABLE DocenteAfiliaFamiliar( nro_documento INT, parentesco VARCHAR(50), Legajo INT, PRIMARY KEY(nro_documento, L...	0 row(s) affected
26	08:08:34	CREATE TABLE SeguroBeneficiaFamiliar( nro_seguro INT, nro_documento INT, porcentaje REAL, PRIMARY KEY(nro_seguro, nro_doc...	0 row(s) affected
27	08:08:34	CREATE TABLE ObraAfiliaFamiliar( nro_obra INT, nro_documento INT, PRIMARY KEY(nro_obra, nro_documento), FOREIGN KEY (nro...	0 row(s) affected
28	08:08:34	CREATE TABLE DocenteTieneDomicilio ( legajo INT, id_domicilio INT, tipo VARCHAR(50), Primary key(legajo, id_domicilio), FOREIG...	0 row(s) affected
29	08:08:34	CREATE TABLE CursoOconf ( curso_Conf VARCHAR(50) PRIMARY KEY, descripcion VARCHAR(100), cuit INT NOT NULL, FOREIG...	0 row(s) affected
30	08:08:34	CREATE TABLE DocenteDictaConferencia( curso_Conf VARCHAR(50), legajo INT, desde DATE, hasta DATE, primary key (curso_Conf, J...	0 row(s) affected
31	08:08:34	CREATE TABLE ActividadesInvestigacion ( investigacion INT PRIMARY KEY, area_principal VARCHAR(100), categoria VARCHAR(50), ...	0 row(s) affected
32	08:08:34	CREATE TABLE DocenteParticipaInvestigacion( Legajo INT, investigacion INT, desde DATE, hasta DATE, dedicación int, PRIM...	0 row(s) affected
33	08:08:34	CREATE TABLE Títulos ( idTitulo INT PRIMARY KEY, nivel VARCHAR(50), nombre VARCHAR(100) )	0 row(s) affected
34	08:08:34	CREATE TABLE DocenteObtieneTitulo( idTitulo INT, Legajo INT, Desde DATE, Hasta DATE, PRIMARY KEY(idTitulo, Legajo), FO...	0 row(s) affected
35	08:08:34	CREATE TABLE UniversidadBrindaTitulo( idTitulo INT, Universidad INT, PRIMARY KEY(idTitulo, Universidad), FOREIGN KEY(idTitulo) ...	0 row(s) affected
36	08:08:34	CREATE TABLE ActividadesExtensionUniversitaria ( idActividad INT PRIMARY KEY, acciones VARCHAR(255), cargo VARCHAR(50), ...	0 row(s) affected
37	08:08:35	CREATE TABLE docenteImpulsaActividadExt ( desde DATE, hasta DATE , id_actividad_ext INT, Legajo INT, dedicación int, PRIMARY KEY( ...	0 row(s) affected

## ➤ DICCIONARIO DE DATOS

Nombre campo	Descripción	Entidad/es a la que pertenece	funcion	dominio	acepta nulos
legajo	Legajo del docente correspondiente	Docente, DocentePoseeContactos, DocenteTienePublicaciones, DocenteSabeIdiomas, DocenteParticipaReuniones, PasividadObtenida, DocenteRealizaTareaNoOficiales, DocenteAfiliaFamiliar, DocenteTieneDomicilio, Seguro, Obra_Social,	PK, FK	INT	NO
id_docente	Legajo referenciando	DocenteParticipaInvestigacion, DocenteObtieneTitulo,	FK	INT	NO

	al docente	CursoDictaDocente,docenteImpulsaActividadExt			
DNI	DNI del docente	Docente		INT	SÍ
nombre	Nombre del docente	Docente		VARCHAR(100)	NO
estado_civil	Estado civil del docente	Docente		VARCHAR(50)	SÍ
email	Email del docente	Docente		VARCHAR(100)	SÍ
sexo	Sexo del docente	Docente		ENUM('M', 'F')	SÍ
fecha_nacimiento	Fecha de nacimiento del docente	Docente, familiar		DATE	SÍ
numero	Número de contacto	contactos, DocentePoseeContactos	PK, FK	INT	NO
tipo	Tipo de contacto	contactos		VARCHAR(50)	SÍ
medio	Medio de contacto	contactos		VARCHAR(50)	SÍ
titulo	Título de la publicación	Publicaciones, DocenteTienePublicaciones	PK, FK	VARCHAR(255)	NO
año	Año de la publicación	Publicaciones, DocenteTienePublicaciones	PK, FK	INT	NO
ref_biblio	Referencia bibliográfica	Publicaciones, DocenteTienePublicaciones	PK, FK	VARCHAR(255)	NO
nivel	Nivel de idioma	Idiomas, DocenteSabeIdiomas	PK, FK	VARCHAR(50)	NO
idioma	Idioma	Idiomas, DocenteSabeIdiomas	PK, FK	VARCHAR(50)	NO
institucion	Institución donde se aprendió	Idiomas, ActividadesInvestigacion		VARCHAR(50), INT	SÍ, NO
certificacion	Certificación obtenida	Idiomas		VARCHAR(50)	SÍ

id_reunion	ID de la reunión científica	ReunionesCientificas, DocenteParticipaReuniones	PK, FK	INT	NO
fecha	Fecha de la reunión	ReunionesCientificas		DATE	SÍ
participacion	Participación en la reunión	ReunionesCientificas		VARCHAR(30)	SÍ
id_pasividad	ID de la pasividad	PasividadObtenida	PK	INT	NO
regimen	Régimen de pasividad	PasividadObtenida		VARCHAR(50)	SÍ
fecha_inicio	Fecha de inicio	PasividadObtenida, TareasNoOficiales, Antecedentes		DATE	SÍ, NO
suspendido	Si la pasividad está suspendida	PasividadObtenida		BOOLEAN	SÍ
causa	Causa de la pasividad	PasividadObtenida		VARCHAR(255)	SÍ
id_tarea	ID de la tarea no oficial	TareasNoOficiales, DocenteRealizaTareaNoOficiales	PK, FK	INT	NO
RelacionDependencia	Relación de dependencia	TareasNoOficiales		VARCHAR(50)	SÍ
FuncionDesempeñada	Función desempeñada	TareasNoOficiales		VARCHAR(50)	SÍ
id_antecedentes	ID de los antecedentes	Antecedentes, Antecedentes_Docentes, Antecedentes_Profesionales, Horas	PK, FK	INT	NO
id_docentes	ID del docente	Antecedentes	FK	INT	NO
nombre_cargo	Nombre del cargo	Antecedentes		VARCHAR(200)	NO
funcion	Función desempeñada	Antecedentes		VARCHAR(255)	SÍ
reparticion	Repartición	Antecedentes		INT	SÍ
fecha_fin	Fecha de fin	Antecedentes		DATE	SÍ
materia	Materia	Antecedentes_Docentes		VARCHAR(	SÍ

				200)	
id_corresponde	ID de la universidad	Antecedentes_Docentes, Antecedentes_Profesionales	FK	INT	NO
tipo_act	Tipo de actividad	Antecedentes_Profesionales		VARCHAR(200)	SÍ
hora_entrada	Hora de entrada	Horas	PK	TIME	NO
hora_salida	Hora de salida	Horas	PK	TIME	NO
dia	Día	Horas	PK	TINYINT	NO
nro_seguro	Número del seguro	Seguro, SeguroBeneficiaFamiliar	PK, FK	INT	NO
nombre_aseguradora	Nombre de la aseguradora	Seguro		VARCHAR(100)	SÍ
cod_compania	Código de la compañía	Seguro		INT	SÍ
tipo	Tipo de seguro	Seguro		VARCHAR(50), INT	SÍ
tipo	tipo de empresa	Empresa		VARCHAR(10)	Si
CUIT	CUIT de la institución	Seguro, Obra_Social, Universidad, Institucion, Empresa	FK/PK	BIGINT	NO
nro_obra	Número de obra social	Obra_Social, ObraAfiliaFamiliar	PK, FK	INT	NO
Nombre_obra	Nombre de la obra social	Obra_Social		VARCHAR(30)	SÍ
Plan	Plan de la obra social	Obra_Social		VARCHAR(60)	SÍ
nro_documento	Número de documento del familiar	Familiar, DocenteAfiliaFamiliar, SeguroBeneficiaFamiliar, ObraAfiliaFamiliar	PK, FK	INT	NO
NombreCompleto	Nombre completo del familiar	Familiar		VARCHAR(100)	SÍ
tipo_documento	Tipo de	Familiar		VARCHAR(	SÍ

	documento del familiar			50)	
id_domicilio	ID del domicilio	Familiar, Domicilio, Institucion	FK/PK	INT	NO
parentesco	Parentesco	Familiar, DocenteAfiliaFamiliar	PK	VARCHAR(50)	NO
porcentaje	Porcentaje de beneficio	SeguroBeneficiaFamiliar		REAL	SÍ
CP	Código postal	Ubicacion, Domicilio	PK, FK	INT	NO
Provincia	Provincia	Ubicacion		VARCHAR(50)	SÍ
Localidad	Localidad	Ubicacion		VARCHAR(50)	SÍ
tipo	Tipo de domicilio	Domicilio		VARCHAR(50)	SÍ
barrio	Barrio	Domicilio		VARCHAR(100)	SÍ
calle	Calle	Domicilio		VARCHAR(100)	SÍ
numero	Número de calle	Domicilio		INT	SÍ
piso	Piso	Domicilio		INT	SÍ
depto	Departamento	Domicilio		VARCHAR(50)	SÍ
cp	Código postal	Domicilio	FK	VARCHAR(10)	NO
razon_social	Razón social	Institucion		VARCHAR(100)	SÍ
CursoConf	Configuración del curso	CursoOfreceInst	PK	VARCHAR(50)	NO
desc	Descripción	CursoOfreceInst, CursoDictaDocente		VARCHAR(100)	SÍ
curso_Conf	Curso configurado	CursoDictaDocente	PK	VARCHAR(50)	NO
desde	Desde cuándo	CursoDictaDocente,		DATE	SÍ

		DocenteParticipaInvestigacion, DocenteObtieneTitulo			
hasta	Hasta cuándo	CursoDictaDocente, DocenteParticipaInvestigacion, DocenteObtieneTitulo		DATE	SÍ
investigacion	Investigación	ActividadesInvestigacion, DocenteParticipaInvestigacion	PK, FK	INT	NO
area_principal	Área principal	ActividadesInvestigacion		VARCHAR( 100)	SÍ
categoria	Categoría	ActividadesInvestigacion		VARCHAR( 50)	SÍ
dedicacion	Dedicación	ActividadesInvestigacion, ActividadesExtensionUniversitaria		VARCHAR( 50)	SÍ
	ID del título	Titulos, DocenteObtieneTitulo, UniversidadBrindaTitulo	PK, FK	INT	NO
nombre	Nombre del título	Titulos		VARCHAR( 100)	SÍ
idActividad	ID de la actividad de extensión	ActividadesExtensionUniversitaria	PK	INT	NO
acciones	Acciones de la extensión	ActividadesExtensionUniversitaria		VARCHAR( 255)	SÍ
cargo	Cargo en la extensión	ActividadesExtensionUniversitaria		VARCHAR( 50)	SÍ
ActividadPrinci pal	Actividad principal de la empresa	Empresa		VARCHAR( 100)	SÍ
tipo	Tipo de empresa	Empresa		INT	SÍ
dependencia	Dependencia de la universidad	Universidad		VARCHAR( 100)	SÍ
unidad_academ ica	Unidad academica a la que hace referencia	Universidad		VARCHAR( 50)	SÍ

## ➤ Etapa 2

### INSERCIÓN, MODIFICACIÓN, BORRADO DE DATOS

#### Cinco consultas distintas para inserción de datos (distintos a los ya cargados en forma masiva)

♦ El profesor con legajo 12234 participa en todas las investigaciones de la facultad resistencia que sean de inteligencia artificial

```
insert into docenteparticipainvestigacion
(Legajo, investigacion, desde, hasta, dedicación)
select 12234, i.investigacion, CURRENT_DATE(), NULL, 4
from actividadesinvestigacion i
where area_principal='Ciencia de datos'
and i.institucion in (select CUIT from institucion where
razon_social='Universidad Tecnológica Nacional Frre')
```

```
1 • insert into docenteparticipainvestigacion (Legajo, investigacion, desde, hasta, dedicación)
2   select 12234, i.investigacion, CURRENT_DATE(), NULL, 4
3   from actividadesinvestigacion i
4   where area_principal='Ciencia de datos'
5   and i.institucion in (select CUIT from institucion where razon_social='Universidad Tecnológica Nacional Frre')
```

Output			
Action Output			
#	Time	Action	Message
1	15:31:43	insert into docenteparticipainvestigacion (Legajo, investigacion, desde, hasta, dedicación) sel...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0

♦ Cargar todas las universidades que estan como institucion pero no cargadas en la tabla universidad

```
INSERT INTO universidad (CUIT, dependencia, unidad_academica)
SELECT I.CUIT, NULL, NULL FROM institucion I
WHERE I.razon_social LIKE 'Universidad%'
AND I.CUIT NOT IN(SELECT U.CUIT FROM universidad U )
```

```
1 • INSERT INTO universidad (CUIT, dependencia, unidad_academica)
2   SELECT I.CUIT, NULL, NULL FROM institucion I
3   WHERE I.razon_social LIKE 'Universidad%'
4   AND I.CUIT NOT IN(SELECT U.CUIT FROM universidad U )
```

Output			
Action Output			
#	Time	Action	Message
1	15:34:25	INSERT INTO universidad (CUIT, dependencia, unidad_academica) SELECT I.CUIT, NULL, NULL FROM institucion I WHERE I.razon_social...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0

♦ Cargar todos los profesores que tengan doctorado en ciencia de datos para inteligencia artificial

```

INSERT INTO Antecedentes (id_antecedentes, id_docentes,
nombre_cargo, funcion, reparticion, fecha_inicio, fecha_fin)
    select (select count(*) from antecedentes) + ROW_NUMBER()
OVER (ORDER BY (SELECT 1)) ,
        d.legajo, 'profesor', 'adjunto temporal inteligencia
artificial', NULL, CURRENT_DATE(), NULL
    from (
        -- todos los profesores que trabajen en la facultad
regional resistencia y tengan titulo doctorado en ciencia de
datos

        Select distinct d.legajo
        from docente d
        join antecedentes a on a.id_docentes=d.legajo
        join antecedentes_Docentes ad on
a.id_antecedentes=ad.id_antecedentes
        join docenteobtienetitulo dt on dt.legajo=d.legajo
        join titulos t on dt.idtitulo=t.idtitulo
        where ad.id_Corresponde=30123456780 and
t.nombre="Doctorado en ciencias de datos") d

```

```

INSERT INTO Antecedentes (id_antecedentes, id_docentes, nombre_cargo, funcion, reparticion, fecha_inicio, fecha_fin)
select (select count(*) from antecedentes) + ROW_NUMBER() OVER (ORDER BY (SELECT 1)) ,
    d.legajo, 'profesor', 'adjunto temporal inteligencia artificial', NULL, CURRENT_DATE(), NULL
from (
    -- todos los profesores que trabajen en la facultad regional resistencia y tengan titulo doctorado en ciencia de
    Select distinct d.legajo
    from docente d
    join antecedentes a on a.id_docentes=d.legajo
    join antecedentes_Docentes ad on a.id_antecedentes=ad.id_antecedentes
    join docenteobtienetitulo dt on dt.legajo=d.legajo
    join titulos t on dt.idtitulo=t.idtitulo
    where ad.id_Corresponde=30123456780 and t.nombre="Doctorado en ciencias de datos") d

```

```

INSERT INTO Antecedentes_Docentes (id_antecedentes, materia,
id_corresponde)
SELECT
    d.id_antecedentes,
    'Inteligencia artificial',
    30123456780
FROM
-- hago esto para tener la cantidad de filas a ingresar

```



```

(Select id_antecedentes
      from docente d
      join antecedentes a on a.id_docentes=d.legajo
      where a.funcion='adjunto temporal inteligencia
artificial' ) d

INSERT INTO Antecedentes_Docentes (id_antecedentes, materia, id_corresponde)
SELECT
      d.id_antecedentes,
      'Inteligencia artificial',
      30123456780
FROM
-- hago esto para tener la cantidad de filas a ingresar
(Select id_antecedentes
      from docente d
      join antecedentes a on a.id_docentes=d.legajo
      where a.funcion='adjunto temporal inteligencia artificial' ) d

```

Action Message  
 :24 INSERT INTO Antecedentes\_Docentes (id\_antecedentes, materia, id\_corresponde) SELECT d.id\_antecedentes, 'Inteligencia artifi... 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0

◆Cargar los antecedentes de profesores que cargaron su lugar de trabajo pero no detallaron que trabajan

◆Se realiza la carga primero en antecedentes profesionales y después en antecedentes

```

insert into antecedentes_profesionales
select (select count(*) from antecedentes) + ROW_NUMBER() OVER
(ORDER BY (SELECT 1)) ,e.actividadPrincipal, e.cuit from
docente d
join docentetienedomicilio dd on d.legajo=dd.legajo
join docenteobtienetitulo dt on dt.legajo=d.legajo
join titulos t on dt.idtitulo=t.idtitulo
join institucion i on i.id_domicilio=dd.id_domicilio
join empresa e on i.cuit=e.cuit
where dd.tipo='trabajo'

```

```

insert into antecedentes_profesionales
select (select count(*) from antecedentes) + ROW_NUMBER() OVER (ORDER BY (SELECT 1)) ,
e.actividadPrincipal, e.cuit
from docente d
join docentetienedomicilio dd on d.legajo=dd.legajo
join docenteobtienetitulo dt on dt.legajo=d.legajo
join titulos t on dt.idtitulo=t.idtitulo
join institucion i on i.id_domicilio=dd.id_domicilio
join empresa e on i.cuit=e.cuit
where dd.tipo='trabajo'

```

put  
 3:17 insert into antecedentes\_profesionales select (select count(\*) from antecedentes) + ROW\_NUMBER() OVER (ORDER BY (SELECT 1)) ,e.a... 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Si hubiera habido una carga, realizaremos la siguiente consulta:

```

INSERT INTO Antecedentes (id_antecedentes, id_docentes,
nombre_cargo, funcion, reparticion, fecha_inicio, fecha_fin)
select (select count(*) from antecedentes) + ROW_NUMBER() OVER
(ORDER BY (SELECT 1))
,d.legajo, 'Desconocido', 'Desconocido', NULL, CURRENT_DATE(), NULL
from docente d
join docentetienedomicilio dd on d.legajo=dd.legajo
join docenteobtienetitulo dt on dt.legajo=d.legajo
join titulos t on dt.idtitulo=t.idtitulo
join institucion i on i.id_domicilio=dd.id_domicilio
where dd.tipo='trabajo'

```

## Cinco consultas distintas para modificación de datos

◆ Actualizar la participación de un docente en específico que dejó de participar en una investigación

```

UPDATE  docenteparticipainvestigacion
set hasta =CURRENT_DATE()
WHERE legajo= (SELECT legajo from docente where nombre='Juan
Perez')

```

```
UPDATE docenteparticipainvestigacion
set hasta =CURRENT_DATE()
WHERE legajo= (SELECT legajo from docente where nombre='Juan Perez')
```

Output	Action	Message
3:42	UPDATE docenteparticipainvestigacion set hasta =CURRENT_DATE() WHERE legajo= (SELECT legajo from docente where nombre='Jua...	2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0

◆Se actualiza la razón social de las instituciones “UTN” a “Universidad Tecnológica Nacional”

```
UPDATE institucion
set razon_social =REPLACE(razon_social, 'UTN', 'Universidad
Tecnologica nacional')
where razon_social LIKE ('UTN%')
UPDATE institucion
set razon_social =REPLACE(razon_social, 'UTN', 'Universidad Tecnologica nacional')
where razon_social LIKE ('UTN%')
```

Output	Action	Message
20:22	UPDATE institucion set razon_social =REPLACE(razon_social, 'UTN', 'Universidad Tecnologica nacional') where razon_social LIKE('UTN%')	1 row(s) affected Rows matched: 4 Changed: 1 Warnings: 0

◆Actualizar los docentes que tengan una pareja cargada como familiar como estado civil=casado

```
UPDATE docente d
set d.estado_civil='Casado'
where d.legajo in ( select legajo from docenteAfiliaFamiliar
where parentesco LIKE 'Espos%')
UPDATE docente d
set d.estado_civil='Casado'
where d.legajo in ( select legajo from docenteAfiliaFamiliar where parentesco LIKE 'Espos%')
```

Output	Action	Message
3:53	UPDATE docente d set d.estado_civil='Casado' where d.legajo in ( select legajo from docenteAfiliaFamiliar where parentesco LIKE 'Espos%')	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

◆Actualizar la obra social de los seguros asociados a la UTN

```
UPDATE seguro
SET nombre_aseguradora='INSSEP'
where CUIT in (select I.CUIT from Institucion I
where I.Razon_Social LIKE 'UTN FRRe')
```

```

• UPDATE seguro
  SET nombre_aseguradora='INSSEP'
  where CUIT in (select I.CUIT from Institucion I
  where I.Razon_Social LIKE 'Universidad Tecnologica Nacional%')

```

---

Time	Action	Message
17:35:35	UPDATE seguro SET nombre_aseguradora='INSSEP' where CUIT in (select I.CUIT from Institucion I where I.Razon_Social LIKE 'Universidad Tecnolo...	15 row(s) affected Rows matched: 15 Changed: 15 Warnings: 0

◆Se mueven las horas de analisis matematico de la facultad de resistencia una hora

```

UPDATE Horas
set hora_Entrada=DATE_SUB(hora_entrada,INTERVAL 1 HOUR),
hora_salida=DATE_SUB(hora_salida,INTERVAL 1 HOUR)
where id_Antecedentes in (
select ad.id_antecedentes from antecedentes_docentes ad
join universidad u on u.CUIT=ad.id_Corresponde
where ad.materia='Analisis matematico' and
u.unidad_academica='Facultad regional resistencia')

```

```

UPDATE Horas
set hora_Entrada=DATE_SUB(hora_entrada,INTERVAL 1 HOUR),
hora_salida=DATE_SUB(hora_salida,INTERVAL 1 HOUR)
where id_Antecedentes in (
select id_Antecedentes from antecedentes_docentes ad
join institucion I on I.CUIT=ad.id_Corresponde
where ad.materia='Inteligencia Artificial' and I.razon_social LIKE '%Frre%')

```

---

Time	Action	Message
05:53	UPDATE Horas set hora_Entrada=DATE_SUB(hora_entrada,INTERVAL 1 HOUR); hora_salida=DATE_SUB(hora_salida,INTERVAL 1 HOUR) where i...	2 row(s) affected Rows matched: 2 Changed: 2 Warnings: 0

## Cinco consultas sql distintas para borrado de filas.

◆Borrar todos los contactos que sean del tipo correo porque es información ya que ya se encuentra en la tabla de docente (relacionado con la generación masiva)  
Estas consultas sirven para limpiar los datos de la carga masiva pues varios son repetidos o no tienen sentido con el resto del esquema

```
DELETE FROM contactos WHERE tipo='Email'
```

◆ Borrar todos los familiar que no tienen docente asociado

```
DELETE FROM familiar
WHERE nro_documento NOT IN(SELECT nro_documento FROM
docenteafiliafamiliar)
```

◆ Borrar todos los domicilios que no tengan una entidad asociada

```
DELETE FROM Domicilio
WHERE id_domicilio NOT IN (
SELECT id_domicilio FROM Institucion
UNION
SELECT id_domicilio FROM docentetienedomicilio
UNION
SELECT id_domicilio FROM familiar)
```

◆ Borrar todos los datos de un docente

```
DELETE FROM PasividadObtenida WHERE legajo = 1;
DELETE FROM DocentePoseeContactos WHERE legajo = 1;
DELETE FROM DocenteTienePublicaciones WHERE legajo = 1;
DELETE FROM DocenteSabeldiomas WHERE legajo = 1;
DELETE FROM DocenteParticipaReuniones WHERE legajo = 1;
DELETE FROM DocenteRealizaTareaNoOficiales WHERE legajo = 1;
DELETE FROM Seguro WHERE legajo = 1;
DELETE FROM Obra_Social WHERE legajo = 1;
DELETE FROM DocenteAfiliaFamiliar WHERE legajo = 1;
DELETE FROM DocenteTieneDomicilio WHERE legajo = 1;
DELETE FROM DocenteDictaConferencia WHERE legajo = 1;
DELETE FROM DocenteParticipaInvestigacion WHERE legajo = 1;
DELETE FROM DocenteObtieneTitulo WHERE legajo = 1;
DELETE FROM docenteImpulsaActividadExt WHERE legajo = 1;
DELETE FROM Docente WHERE legajo = 1;
```

◆ Borrar una Institución que no esté asociada a una “Empresa” o una “Universidad”

```
DELETE FROM Institucion
WHERE CUIT NOT IN (
    SELECT CUIT FROM Empresa
)
AND CUIT NOT IN (
    SELECT CUIT FROM Universidad
);
```

```
DELETE FROM Domicilio WHERE id_domicilio NOT IN (
    SELECT id_domicilio FROM Institucion
);
```

## Consultas SELECT

1) Listado de docentes que viven en una provincia distinta de aquella en la que trabajan.

```
SELECT distinct D.*
FROM Docente D
INNER JOIN DocenteTieneDomicilio DD on D.legajo=DD.Legajo
INNER JOIN Domicilio dom1 on dom1.id_domicilio =
DD.id_domicilio
INNER JOIN Antecedentes AA on AA.id_docentes = D.legajo -- and
AA.fecha_fin IS NULL
INNER JOIN Antecedentes_Docentes A on A.id_antecedentes =
AA.id_antecedentes
INNER JOIN Institucion I on I.CUIT = A.id_corresponde
INNER JOIN domicilio dom2 on dom2.id_domicilio= I.id_domicilio
Where dom1.cp <> dom2.cp
```

2) Listado de docentes que poseen títulos de posgrado y no realizan tareas de investigación.

```
SELECT DISTINCT D.nombre, D.legajo
FROM Docente D
INNER JOIN DocenteObtieneTitulo DOT on DOT.Legajo=D.Legajo
INNER JOIN Titulos T on T.idtitulo=DOT.idtitulo
WHERE T.nivel='Posgrado' and D.legajo not in(select DPI.legajo
from DocenteParticipaInvestigacion DPI)
```

3) Informar promedio de edad de los docentes que poseen más de 10 años de antecedentes como docentes.

```
Select AVG(edad)
From (SELECT YEAR(CURRENT_DATE())- YEAR(D.fecha_nacimiento)
edad
FROM Docente D
INNER JOIN Antecedentes A on A.ID_dOCENTES = D.legajo
INNER JOIN Antecedentes_Docentes AA on AA.id_antecedentes =
A.id_antecedentes
where (YEAR(CURRENT_DATE()) - YEAR(A.Fecha_inicio))>=10
and A.FECHA_FIN IS NULL OR (YEAR(A.FECHA_FIN )-
YEAR(A.Fecha_inicio))>10) t )
```

4) Listar DNI y nombre de los docentes que presentaron más de un cargo docente en las declaraciones juradas de los últimos 3 años

```
SELECT D.DNI, D.nombre
FROM (SELECT D.DNI,D.nombre,
YEAR(AA.Fecha_inicio),COUNT(DISTINCT AA.Fecha_inicio) AS
can_cargos
Docente D
INNER JOIN Antecedentes A on A.id_docentes = D.legajo
INNER JOIN Antecedentes_Docentes AA on AA.id_antecedentes =
A.id_antecedentes
WHERE A.fecha_fin IS NULL OR
(YEAR(CURRENT_DATE())-YEAR(A.fecha_fin))<=3
GROUP BY D.DNI,D.nombre, YEAR(AA.Fecha_inicio)
WHERE can_cargos>1
```

5) Listado de docentes cuya carga horaria supera las 20 horas semanales, en función de la última declaración jurada presentada.

```
SELECT D.legajo
FROM Docente D
INNER JOIN Antecedentes A on A.id_docentes = D.legajo
INNER JOIN Antecedentes AA on AA.id_antecedentes =
A.id_antecedentes
INNER JOIN Horas H0 ON AA.id_antecedentes = H0.id_antecedentes
Group by D.legajo
Having SUM(H0.hora_salida - H0.hora_entrada) > 20
```

6) Apellido y nombre de aquellos docentes que poseen la máxima cantidad de cargos docentes actualmente. (La cantidad de cargos surge de sumar todos los cargos docentes que se ejercen - suma de cargos docentes de la última declaración jurada -. Una vez que se sabe la cantidad de cargos por docente se puede averiguar cuál es la máxima cantidad y seguidamente los docentes que tienen esa máxima cantidad). No nos interesa las horas.

```
select nombre
from Docente
where legajo IN (
        SELECT D.legajo
```

```

        from Docente D
        INNER JOIN Antecedentes A on A.id_docentes = D.legajo
        INNER JOIN Antecedentes_Docentes AA on AA.id_antecedentes
= A.id_antecedentes
        group by D.legajo
        having count(AA.materia) = (
            select max(cargos) from (SELECT
D.legajo,count(AA.materia) cargos
            from Docente D
            INNER JOIN Antecedentes A on A.id_docentes =
D.legajo
            INNER JOIN Antecedentes_Docentes AA on
AA.id_antecedentes = A.id_antecedentes
            group by D.legajo) temp ))

```

7) Listado de docentes solteros/as (sin esposa/o e/o hijos a cargo en la obra social).

```

SELECT DISTINCT D.*
FROM Docente d
INNER JOIN DocenteAfiliaFamiliar daf ON d.legajo = daf.Legajo
INNER JOIN Familiar f ON daf.nro_documento = f.nro_documento
INNER JOIN Obra_Social os ON d.legajo = os.Legajo
WHERE d.estado_civil = 'Soltero' or d.estado_civil IS NULL
or d.legajo NOT in(
    SELECT daf2.Legajo
    FROM DocenteAfiliaFamiliar daf2
    INNER JOIN Familiar f2 ON daf2.nro_documento =
f2.nro_documento
    WHERE daf2.Legajo = d.legajo
    AND (daf2.parentesco IN ('Esposo', 'Esposa', 'Hijo',
'Hija')));

```

8) Cantidad de docentes cuyos hijos a cargo son todos menores de 10 años.

```

SELECT COUNT(DISTINCT d.legajo) AS cantidad_docentes
FROM Docente d
INNER JOIN DocenteAfiliaFamiliar daf ON d.legajo = daf.Legajo
INNER JOIN Familiar f ON daf.nro_documento = f.nro_documento
WHERE daf.parentesco IN ('Hijo', 'Hija')

```



```

    AND f.fecha_nacimiento > DATE_SUB(CURDATE(), INTERVAL 10
YEAR)
    AND NOT EXISTS (
        SELECT 1
        FROM DocenteAfiliaFamiliar daf2
        INNER JOIN Familiar f2 ON daf2.nro_documento =
f2.nro_documento
        WHERE daf2.Legajo = d.legajo
            AND daf2.parentesco IN ('Hijo', 'Hija')
            AND f2.fecha_nacimiento <= DATE_SUB(CURDATE(), INTERVAL
10 YEAR)
    );

```

9) Informar aquellos docentes que posean alguna persona del grupo familiar a cargo en la obra social que no es beneficiario del seguro de vida obligatorio.

```

SELECT DISTINCT d.legajo, d.nombre, d.email
FROM Docente d
INNER JOIN Obra_Social os ON d.legajo = os.Legajo
INNER JOIN DocenteAfiliaFamiliar daf ON d.legajo = daf.Legajo
INNER JOIN Familiar f ON daf.nro_documento = f.nro_documento
inner join obraafiliafamiliar oaf on oaf.nro_obra=os.nro_obra
inner JOIN SeguroBeneficiaFamiliar sbf ON f.nro_documento =
sbf.nro_documento
inner JOIN Seguro s ON sbf.nro_seguro = s.nro_seguro AND
s.tipo = 'Obligatorio'
WHERE s.nombre_aseguradora!= os.nombre_obra
and s.legajo=d.legajo

```

10) Informar Cantidad de individuos asegurados por provincia.

```

SELECT u.Provincia, COUNT(DISTINCT s.nro_seguro) AS
cantidad_asegurados
FROM Seguro s
INNER JOIN Docente d ON s.Legajo = d.legajo
INNER JOIN DocenteTieneDomicilio dtd ON d.legajo = dtd.legajo

```

```
INNER JOIN Domicilio dom ON dtd.id_domicilio =  
dom.id_domicilio  
INNER JOIN Ubicacion u ON dom.cp = u.CP  
GROUP BY u.Provincia;
```