



MUJERES IT

PROYECTO FINAL
SQL



Lilian Virginia Lombardi López



PRIMERA ENTREGA

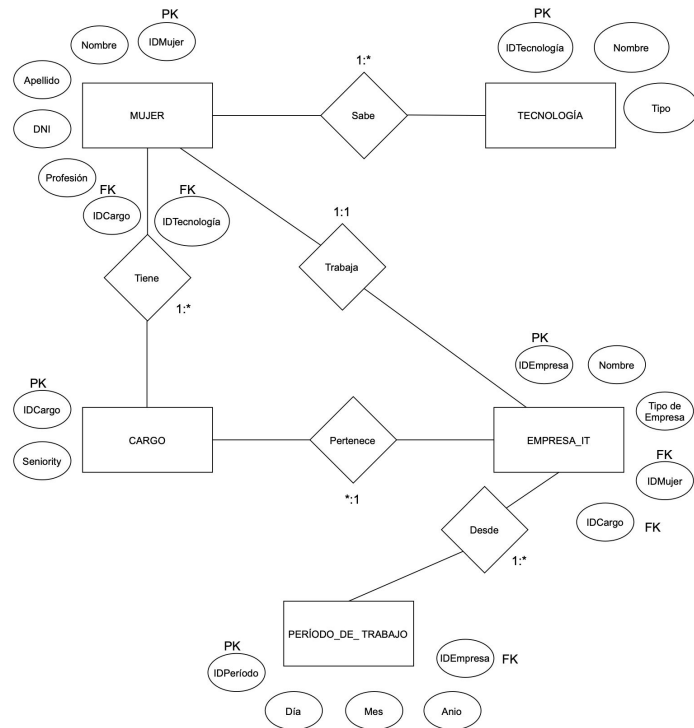
PROYECTO FINAL
SQL



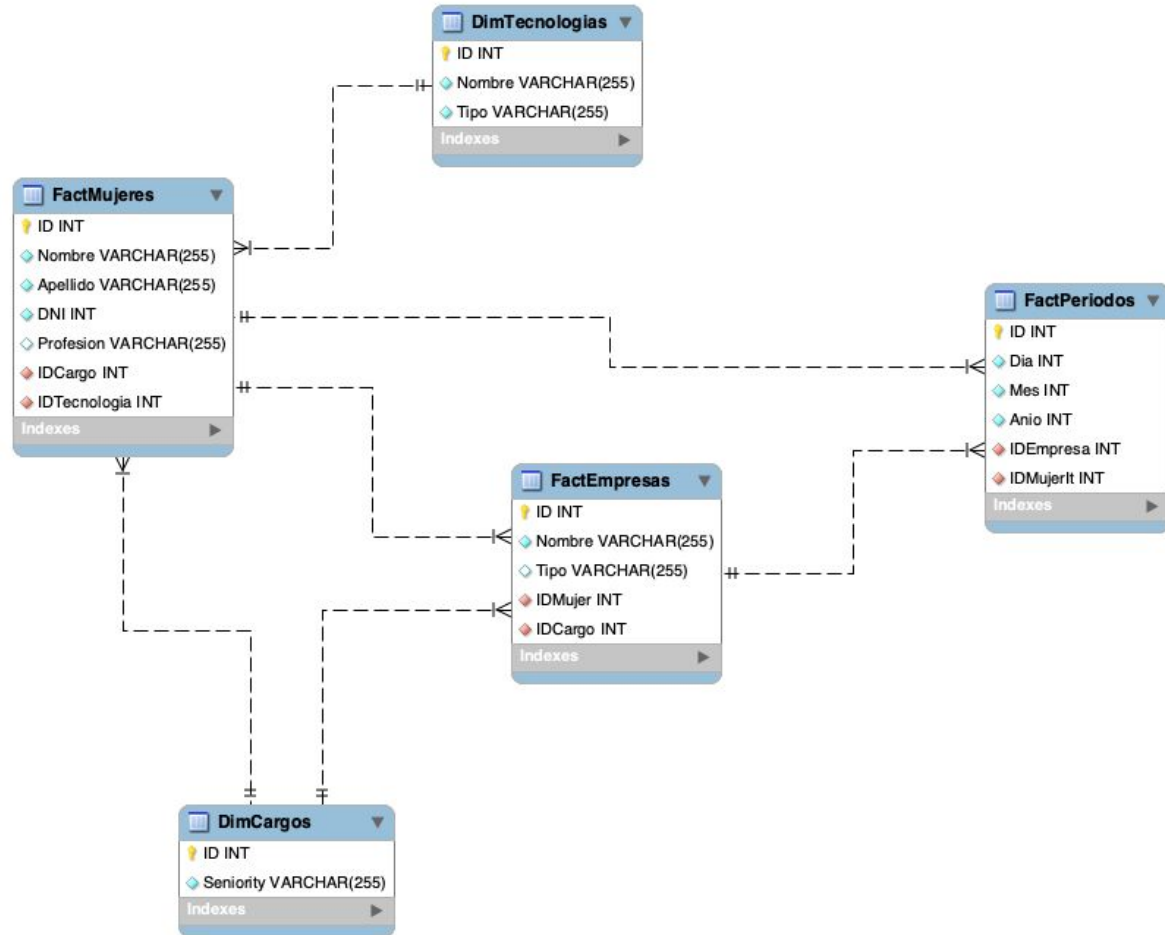
Nombre del Proyecto: Mujeres IT, Mujeres en IT

DESCRIPCIÓN: El objetivo de este Proyecto es crear una base de datos para almacenar la información de las Mujeres que trabajan en IT, ya sea el cargo al que pertenecen, como a la empresa y las fechas de ingreso a la misma entre otra información, con la idea de hacer uso de los datos para su posterior análisis.

Primer Modelo Entidad - Relación:



Modelo ER:



Tablas Proyecto:

DimCargos: Esta es una tabla dimensional de mi modelo dimensional

Columna	Tipo de Dato	Clave	Características
ID	INT	PK	UNIQUE NOT NULL AUTO_INCREMENT
SENIORITY	VARCHAR(255)		NOT NULL

DimTecnologias: Esta es una tabla dimensional de mi modelo dimensional

Columna	Tipo de Dato	Clave	Características
ID	INT	PK	NOT NULL AUTO_INCREMENT
NOMBRE	VARCHAR(255)		NOT NULL
TIPO	VARCHAR(255)		NOT NULL

FactMujeres: Esta es una tabla de hechos de mi modelo dimensional

Columna	Tipo de Dato	Clave	Características
ID	INT	PK	NOT NULL AUTO_INCREMENT
NOMBRE	VARCHAR(255)		NOT NULL
APELLIDO	VARCHAR(255)		NOT NULL
DNI	INT		NOT NULL
PROFESION	VARCHAR(255)		NOT NULL
IDCARGO	INT	FK	NOT NULL
IDTECNOLOGIA	INT	FK	NOT NULL

FactEmpresas: Esta es una tabla de hechos de mi modelo dimensional

Columna	Tipo de Dato	Clave	Características
ID	INT	PK	UNIQUE NOT NULL
NOMBRE	VARCHAR(255)		NOT NULL
TIPO	VARCHAR(255)		
IDMUJER	INT	FK	NOT NULL
IDCARGO	INT	FK	NOT NULL

FactPeriodos: Esta es una tabla de hechos de mi modelo dimensional

Columna	Tipo de Dato	Clave	Características
ID	INT	PK	NOT NULL AUTO_INCREMENT
DIA	INT		NOT NULL
MES	INT		NOT NULL
ANIO	INT		NOT NULL
IDEMPRESA	INT	FK	NOT NULL
IDMUJERIT	INT	FK	NOT NULL



SEGUNDA ENTREGA

PROYECTO FINAL
SQL



Listado de Vistas:

- **Primera Vista:** vista_profesion_mujeresit.

OBJETIVO: Mostrar los nombres y apellidos de las Mujeres IT, su profesión y el tipo de de seniority que tiene cada una de ellas.

DESCRIPCIÓN: La idea principal de esta vista es que la persona que vaya a utilizar la base de datos sepa quien es la Mujer IT, su profesión y el tipo de seniority que tiene la misma. En este caso es muy importante conocer el seniority de cada una de ellas, para saber qué experiencia tienen en su profesión. Es debido a eso que se crea esta vista.

TABLAS QUE COMPONEN A LA VISTA: Las tablas que componen la vista, vista_profesion_mujeresit, son: FactMujeres y DimCargos.

EJEMPLO VISTA, vista_profesion_mujeresit:

Result Grid					
Filter Rows: <input type="text" value="Search"/>					
	nombre	apellido	profesion	seniority	
▶	Virginia	Lombardi	iOS Developer	Trainee	
▶	Lorena	Alvarez	Java Developer	Junior	
▶	Siri	Altez	iOS Developer	Junior	
▶	Alicia	Gonzalez	Python Developer	Junior Advanced	
▶	Martina	Bernardi	Android Developer	Semi Senior	
▶	Lucia	Perez	Systems Engineering	Senior	
▶	Isabel	Vayra	Web UI Developer	Senior Advanced	
▶	Micaela	Nelson	Web UI Developer	Senior Level One	
▶	Mariana	Orman	iOS Developer	Senior Level Two	
▶	Natalia	Bonjour	C++ Developer	Senior Level Three	
▶	Susana	Mori	C Developer	Senior Level Four	

- **Segunda Vista:** vista_ingreso_mujeresit_en_empresait.

OBJETIVO: Mostrar el nombre, el apellido de la Mujer IT. Y también incluir información relevante como el día, el mes y el año de ingreso de la Mujer IT a la Empresa que integra. Ordenado de forma DESC por el año.

DESCRIPCIÓN: Debido a que es importante tener una idea de cuándo es que ingresó la Mujer IT a la Empresa correspondiente es que se crea esta vista. Con el fin de registrar el nombre y apellido de la misma, la fecha de ingreso y por último la empresa en la que trabaja actualmente.

TABLAS QUE COMPONEN A LA VISTA: Las tablas que componen la vista, vista_ingreso_mujeresit_en_empresait, son: FactMujeres, FactPeriodos y FactEmpresas.

EJEMPLO VISTA, vista_ingreso_mujeresit_en_empresait:

Result Grid   Filter Rows: <input type="text" value="Search"/> Export:							
	Nombre_Mujer_IT	Apellido	Dia	Mes	Anio	Empresa	
▶	Natalia	Bonjour	18	1	2021	SparkDigital	
▶	Mariana	Orman	11	7	2021	Sabre	
▶	Alicia	Gonzalez	21	6	2021	Overactive	
▶	Lorena	Alvarez	23	12	2020	d-Local	
▶	Lucia	Perez	20	2	2019	TCS	
▶	Virginia	Lombardi	10	5	2016	Globant	
▶	Micaela	Nelson	12	8	2015	Nimacloud	
▶	Susana	Mori	1	4	2014	Mercado Libre	
▶	Isabel	Vayra	29	3	2013	Tienda Inglesa	
▶	Martina	Bernardi	31	10	2010	Wabbi	


- **Tercera Vista:** vista__mujeresit__trabajan__empresa.

OBJETIVO: Mostrar el nombre y el apellido de la Mujer IT. Y también incluir información relevante como su profesión, empresa en la cual trabaja y el tipo de empresa.

DESCRIPCIÓN: Considerando que hoy en día hay muchos tipos de empresas en las que una Mujer IT puede trabajar, es que se crea esta vista. Para poder visualizar en qué tipo de empresa y en cual empresa trabaja cada una.

TABLAS QUE COMPONEN A LA VISTA: Las tablas que componen la vista, vista__mujeresit__trabajan__empresa, son: FactMujeres y FactEmpresas.

EJEMPLO VISTA, vista_mujeresit_trabajan_empresa:

Result Grid   Filter Rows: <input type="text" value="Search"/> Export: 						
	Nombre	Apellido	Profesion	Empresa	Tipo_Empresa	
▶	Virginia	Lombardi	iOS Developer	Globant	Global Company	
●	Lucia	Perez	Systems Engineering	TCS	Global Company	
	Mariana	Orman	iOS Developer	Sabre	Global Company	
●	Lorena	Alvarez	Java Developer	d-Local	Global Company	
	Alicia	Gonzalez	Python Developer	Overactive	Global Company	
●	Martina	Bernardi	Android Developer	Wabbi	Start Up	
	Isabel	Vayra	Web UI Developer	Tienda Inglesa	Global Company	
●	Micaela	Nelson	Web UI Developer	Nimacloud	Start Up	
	Susana	Mori	C Developer	Mercado Libre	Start Up	
●	Natalia	Bonjour	C++ Developer	SparkDigital	Start Up	


- **Cuarta Vista:** vista_mujeresit_saben_tecnologias.

OBJETIVO: Mostrar el nombre, el apellido de la Mujer IT. Y también incluir información relevante como la tecnología que manejan y el tipo de dicha tecnología.

DESCRIPCIÓN: Se crea a partir de la necesidad de saber cuántas Mujeres IT saben determinada tecnología y el tipo de la misma. Esto sirve para saber cuántas Mujeres IT conocen la misma tecnología también.

TABLAS QUE COMPONEN A LA VISTA: Las tablas que componen la vista, vista_mujeresit_saben_tecnologias, son: FactMujeres y DimTecnologias.

EJEMPLO VISTA, vista_mujeresit_saben_tecnologias:

Result Grid   Filter Rows: <input type="text" value="Search"/>					
	Nombre	Apellido	Tencologia	Tipo_Tecnologia	
▶	Virginia	Lombardi	iOS	Mobile	
▶	Mariana	Orman	iOS	Mobile	
▶	Siri	Altez	iOS	Mobile	
▶	Isabel	Vayra	React	Web UI	
▶	Lucia	Perez	MongoDB	Web UI	
▶	Micaela	Nelson	VUE JS	Web UI	
▶	Lorena	Alvarez	Java	Programming	
▶	Natalia	Bonjour	C++	Programming	
▶	Alicia	Gonzalez	Python	Programming	
▶	Susana	Mori	C	Programming	

- **Quinta Vista:** vista_profesion_tecnologia_iOS_de_mujeresit.

OBJETIVO: Mostrar el nombre, el apellido de la Mujer IT, filtrando por las Mujeres IT que saben sobre la tecnología iOS.

DESCRIPCIÓN: Esta vista se crea para tener conocimiento de cuantas Mujeres IT hay en la base de datos que sepan la tecnología iOS. Incluyendo nombre apellido de las mismas, así como la profesión y la tecnología que manejan que en este caso va a ser iOS.

TABLAS QUE COMPONEN A LA VISTA: Las tablas que componen la vista, vista_profesion_tecnologia_iOS_de_mujeresit, son: FactMujeres y DimTecnologias.

EJEMPLO VISTA, vista_profesion_tecnologia_iOS_de_mujeresit:

Result Grid				
Filter Rows:				
	Nombre	Apellido	Profesion	Tecnologia
▶	Virginia	Lombardi	iOS Developer	iOS
	Mariana	Orman	iOS Developer	iOS
	Siri	Altez	iOS Developer	iOS

Script SQL Creación de Funciones:

Con el objetivo de procesar y manipular datos de forma procedural y eficiente, se presentan las siguientes funciones para la Base de Datos MujeresIT.

- **Función Conteo_Mujeres_en_IT_entre_periodos:**

OBJETIVO: Contar la cantidad de Mujeres IT que ingresaron en determinado período elegido por el usuario.

DESCRIPCIÓN: Primero se crea una vista llamada: **vista_traer_fecha**. Esta vista lo que hace es traer la fecha en formato por ejemplo: 2012-01-23 (es decir año-mes-día). Para realizar una consulta en la función Conteo_Mujeres_en_IT_entre_periodos, se debe ingresar los parámetros FechaIni (del tipo DATE) y FechaFin (del tipo DATE). El resultado que arroja la función es del tipo INT y corresponde a la cantidad de Mujeres IT que entraron en ese período de fechas.

TABLAS QUE INTERVIENEN EN LA FUNCIÓN: La tabla que interviene en la función es: FactPeriodos.

EJEMPLO FUNCIÓN Conteo_Mujeres_en_IT_entre_periodos:

	MujeresIT. 'Conteo_Mujeres_en_IT_entre_periodos' ('2016-1-12', '2021-12-1')
▶	6

- **Función Conteo_Mujeres_en_IT_en_tecnologia_tipo:**

OBJETIVO: Contar la cantidad de Mujeres IT que manejan X tipo de tecnología.

DESCRIPCIÓN: Para realizar una consulta en la función Conteo_Mujeres_en_IT_en_tecnologia_tipo, se debe ingresar el parámetro Tipo (de tipo VARCHAR(255)). El resultado que arroja la función es del tipo INT y corresponde a la cantidad de Mujeres IT que manejan ese tipo de tecnología dentro de la base de datos.

TABLAS QUE INTERVIENEN EN LA FUNCIÓN: Las tablas que intervienen en dicha función son DimTecnologias y FactMujeres.

EJEMPLO DE Conteo_Mujeres_en_IT_en_tecnologia_tipo:

Call stored function MujeresIT.Conto_Mujeres_en_IT_en_tecnol...

Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

Tipo VARCHAR(255)

Cancel Execute

MujeresIT.Conto_Mujeres_en_IT_en_tecnologia_tipo('Programming')	
▶	4

Script SQL creación de Procedimientos Almacenados:

- **PROCEDIMIENTO ALMACENADO Eliminar y Agregar Valores a DimCargos:**

OBJETIVO: Eliminar y agregar valores a la tabla DimCargos.

DESCRIPCIÓN: Para utilizar el procedimiento almacenado, se debe indicar en el primer parámetro el idCargo que se quiere borrar o agregar y en el segundo parámetro el seniority que se desea borrar o agregar, como por ejemplo: Trainee, Software Engineer, etc. El resultado que arrojará el procedimiento será o bien la agregación de un seniority o la eliminación de uno ya existente en la tabla DimCargos.

TABLAS QUE INTERVIENEN EN EL PROCEDIMIENTO: La tabla que interviene en este procedimiento es DimCargos.

EJEMPLO PROCEDIMIENTO ALMACENADO Eliminar y Agregar Valores a DimCargos:

Call stored procedure MujeresIT.Eliminar y Agregar Valores a Dim...

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

idcargo	<input type="text" value="35"/>	[IN]	INT
seniority	<input type="text" value="chitect Designer"/>	[IN]	VARCHAR(255)

- **PROCEDIMIENTO ALMACENADO** Ordenamiento sobre Tabla FactMujeres:

OBJETIVO: Ordenar la tabla FactMujeres.

DESCRIPCIÓN: Para utilizar el procedimiento almacenado, se debe indicar en el primer parámetro que es el campo por el cual se quiere ordenar la tabla FactMujeres. En el segundo parámetro se debe ingresar el tipo de ordenamiento que se quiere hacer, cómo esta realizado con un ENUM eso le da la pauta al usuario que esté utilizando la base de datos si lo quiere hacer ASC o DESC. Este procedimiento además tiene una particularidad, que es que tiene dos validaciones por si el usuario no ingresa nada en alguno de los campos descriptos anteriormente.

TABLAS QUE INTERVIENEN EN EL PROCEDIMIENTO: La tabla que interviene en este procedimiento es FactMujeres.

EJEMPLO PROCEDIMIENTO ALMACENADO Ordenamiento sobre Tabla FactMujeres:

Call stored procedure MujeresIT.Ordenamiento sobre Tabla Fact...

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

campo	<input type="text" value="Nombre"/>	[IN] VARCHAR(255)
tipo_ordenamiento	<input type="text" value="DESC"/>	[IN] ENUM('ASC', 'DESC')

Cancel Execute

Listado de Triggers:

- TRIGGER Aft_Ins_MujerIt:

OBJETIVO: Crear un trigger después de la inserción de datos.

DESCRIPCIÓN: Se crea primero que nada una tabla llamada **adiciones_mujeres_general** la cual va a guardar la información necesaria que se genere al disparar el trigger, como se puede observar en la imagen.

```
CREATE TABLE MujeresIT.adiciones_mujeres_general(  
  ID_adicion INT AUTO_INCREMENT PRIMARY KEY,  
  Fecha DATE NOT NULL,  
  Hora TIME NOT NULL,  
  Usuario_ID VARCHAR(255) NOT NULL,  
  ID INT,  
  Nombre VARCHAR(255),  
  Apellido VARCHAR(255),  
  Profesion VARCHAR(255),  
  Tipo_operacion VARCHAR(255));
```

Una vez realizada la tabla se procede a escribir el código del trigger. El trigger denominado Aft_Ins_MujerIt, lo que hace es que después de la inserción en la tabla FactMujeres se va a aplicar por cada fila insertar en la tabla adiciones_mujeres_general (creada en Tables_For_Triggers) los datos pasados: Fecha, Hora Usuario_ID, ID, Nombre, Apellido, Profesion y Tipo_operacion. Y en VALUES le pasamos los valores que vamos a insertar en dicha tabla. Se realiza una inserción para probar el funcionamiento de este trigger.

TABLAS QUE INTERVIENEN: La tabla que interviene es FactMujeres.

BENEFICIOS: Esto es beneficioso para por ejemplo cuando un usuario X agrega datos a la tabla FactMujeres, para saber qué datos relevantes se agregaron y la información del usuario así como la fecha y la hora en que se realizaron estos cambios.

EJEMPLO DEL TRIGGER Aft_Ins_MujerIt:

[illegible]

- **TRIGGER Bef_Del_MujerIt:**

OBJETIVO: Crear un trigger antes de la eliminación de datos.

DESCRIPCIÓN: Se crea primero que nada una tabla llamada **eliminaciones_mujeres_general** la cual va a guardar la información necesaria que se genere al disparar el trigger, como se puede observar en la imagen.

```
● ○ CREATE TABLE MujeresIT.eliminaciones_mujeres_general(  
    ID_eliminacion INT AUTO_INCREMENT PRIMARY KEY,  
    Fecha DATE NOT NULL,  
    Hora TIME NOT NULL,  
    Usuario_ID VARCHAR(255) NOT NULL,  
    ID INT,  
    Nombre VARCHAR(255),  
    Apellido VARCHAR(255),  
    Profesion VARCHAR(255),  
    Tipo_operacion VARCHAR(255));
```

Una vez realizada la tabla se procede a escribir el código del trigger. El trigger denominado Bef_Del_MujerIt, lo que hace es que antes de la eliminación en la tabla FactMujeres se va a aplicar por cada fila insertar en la tabla eliminaciones__mujeres__general (creada en Tables__For__Triggers) los datos pasados: Fecha, Hora Usuario_ID, ID, Nombre, Apellido, Profesion y Tipo_operacion. Y en VALUES le pasamos los valores que vamos a insertar en dicha tabla. Se realiza una eliminación para probar el funcionamiento de este trigger.

TABLAS QUE INTERVIENEN: La tabla que interviene es FactMujeres.

BENEFICIOS: Esto es beneficioso para por ejemplo cuando un usuario X elimina datos de la tabla FactMujeres, para saber qué datos relevantes se eliminaron y la información del usuario así como la fecha y la hora en que se realizaron estos cambios.

- **TRIGGER Bef_Del_Tecnologia:**

OBJETIVO: Crear un trigger antes de la eliminación de datos de la tabla.

DESCRIPCIÓN: Se crea primero que nada una tabla llamada **eliminaciones_tecnologias_general** la cual va a guardar la información necesaria que se genere al disparar el trigger, como se puede observar en la imagen.

```
● ⊖ CREATE TABLE MujeresIT.eliminaciones_tecnologias_general(  
    ID_eliminacion INT AUTO_INCREMENT PRIMARY KEY,  
    Fecha DATE NOT NULL,  
    Hora TIME NOT NULL,  
    Usuario_ID VARCHAR(255) NOT NULL,  
    ID INT,  
    Nombre VARCHAR(255),  
    Tipo VARCHAR(255),  
    Tipo_operacion VARCHAR(255));
```

Una vez realizada la tabla se procede a escribir el código del trigger. El trigger denominado `Bef_Del_Tecnologia`, lo que hace es que antes de la eliminación en la tabla `DimTecnologias` se va a aplicar por cada fila insertar en la tabla `eliminaciones_tecnologias_general` (creada en `Tables_For_Triggers`) los datos pasados: `Fecha Hora Usuario_ID`, `ID`, `Nombre`, `Tipo` y `Tipo_operacion`. Y en `VALUES` le pasamos los valores que vamos a insertar en dicha tabla. Se realiza una eliminación para probar el funcionamiento de este trigger. La particularidad que tiene este trigger es que para que no tire un error se debe utilizar el siguiente trozo de código: `SET FOREIGN_KEY_CHECKS=0;`. Es más que nada para que no haya problemas con las FK, y se pueda realizar la eliminación sin problemas.

TABLAS QUE INTERVIENEN: La tabla que interviene es `DimTecnologias`.

BENEFICIOS: Encuentro que este trigger es muy beneficioso para cuando un usuario X elimina datos de la tabla `DimTecnologias`, porque esto deja registrado quien, cuándo y qué datos fueron eliminados con respecto a la misma.

- **TRIGGER Aft_Ins_Tecnologia:**

OBJETIVO: Crear un trigger después de la inserción de datos de la tabla.

DESCRIPCIÓN: Se crea primero que nada una tabla llamada **agregaciones_tecnologias_general** la cual va a guardar la información necesaria que se genere al disparar el trigger, como se puede observar en la imagen.

```
CREATE TABLE MujeresIT.agregaciones_tecnologias_general(  
    ID_eliminacion INT AUTO_INCREMENT PRIMARY KEY,  
    Fecha DATE NOT NULL,  
    Hora TIME NOT NULL,  
    Usuario_ID VARCHAR(255) NOT NULL,  
    ID INT,  
    Nombre VARCHAR(255),  
    Tipo VARCHAR(255),  
    Tipo_operacion VARCHAR(255));
```

Una vez realizada la tabla se procede a escribir el código del trigger. El trigger denominado Aft_Ins_Tecnologia, lo que hace es que antes de la inserción de datos en la tabla FactMujeres se va a aplicar por cada fila insertar en la tabla agregaciones_tecnologias_general (creada en Tables_For_Triggers) los datos pasados: Fecha, Hora Usuario_ID, ID, Nombre, Tipo y Tipo_operacion. Y en VALUES le pasamos los valores que vamos a insertar en dicha tabla. Se realiza una inserción para probar el funcionamiento de este trigger.

TABLAS QUE INTERVIENEN: La tabla que interviene es DimTecnologias.

BENEFICIOS: Esto es beneficioso para por ejemplo cuando un usuario X agrega datos a la tabla DimTecnologias, para saber qué datos relevantes se agregaron y la información del usuario así como la fecha y la hora en que se realizaron estos cambios.

EJEMPLO DEL TRIGGER Aft_Ins_Tecnologia:

[illegible]

ARCHIVO .SQL:

LINK REPO:

https://github.com/LVLVirginiaLombardi/MujeresIT_ProyectoFinal

Se envió invite al repo como colaborador.