



CEPS - UNI
CENTRO DE EXTENSÓN CULTURAL Y PROYECCIÓN
SOCIAL

SQL I - IMPLEMENTACIÓN DE BASE DE DATOS

Docente:

Eric Gustavo Coronel Castillo

Integrantes:

Calvo Palomino, Sharon Antuanet

Castro Adrianzen, Alejandro

Velarde Troya, Paul

Índice

1. Datos de la Empresa	2
2. Presentación del Caso	3
3. Diseño de Base de Datos	3
3.1. Modelo Lógico	4
3.2. Modelo Fisico	5
4. Implementación de la Base de Datos	6
5. Deasrrollo del Video	10
6. Conclusiones	10
7. Recomendaciones	11

PRIMERA PRÁCTICA DE SQL I - IMPLEMENTACIÓN

14 de febrero de 2021

1. Datos de la Empresa

LATAM es una aerolínea de capitales peruanos y chilenos (51 % peruanos, 49 % chilenos debido a la legislación peruana), con RUC 20100103657-LATAM AIRLINES GROUP S.A. SUCURSAL PERU, filial del grupo LATAM Airlines, además, formó parte de alianza con Oneworld. LATAM Perú es conformada por tres grupos comerciales sociales con sede en Perú. Teniendo una participación importante en el mercado aéreo peruano, compite con las aerolíneas Star Perú, Sky Airline Perú, etc.

Su centro de operaciones es el Aeropuerto Internacional Jorge Chávez (LIM), de la ciudad de Lima. Ofrece 16 destinos en el mercado local, operando en las regiones de la costa, sierra y selva peruanas a través de más de 100 vuelos diarios.

2. Presentación del Caso

El área que estamos detallando es la Gerencia de Operaciones, donde el área, en conjunto con otras áreas, canaliza y valida todo el proceso para que un avión pueda despegar a cada vuelo. Se encarga de asignar las tomas de los aviones, tripulación de cabina, tripulación de mando, hora de despegue y aterrizaje, aeropuerto de destino, entre otros. Todo el corazón de la operación aeronáutica se gestiona de aquí.

3. Diseño de Base de Datos

A continuación presentaremos el Diseño de base de datos que se usa en la empresa Latam, específicamente en el área de Operaciones.

El flujo necesario para que un vuelo salga es el siguiente:

El Aeropuerto puede tener varias bases (una base por aerolínea). La base se va a encargar de seleccionar a la tripulación de cabina, tripulación de mando (Pilotos) así como la matrícula del avión que llevará cada vuelo. Además, se va a decidir si es que el vuelo hará varios "tramos" en el mismo día de acuerdo a la legislación aeronáutica actual vigente y las regulaciones aeroportuarias.

Cada avión tendrá designado un número de vuelo, el cual varía según la aerolínea, base, tipo de vuelo y destino.

Asimismo, cada vuelo tendrá una tripulación asignada para atender emergencias, brindar servicios de comida y bebida, entre otros. La cantidad mínima de tripulación por vuelo depende de la cantidad de asientos (ocupados o no) de cada avión. Debe haber un tripulante de cabina por cada 50 asientos.

Por el lado del aeropuerto, la codificación del mismo se basa en la ciudad y el código IATA (International Air Transport Association). La capacidad del aeropuerto se determina también de acuerdo a la cantidad de pistas de aterrizaje y la cantidad de vuelos asignados.

3.1. Modelo Lógico

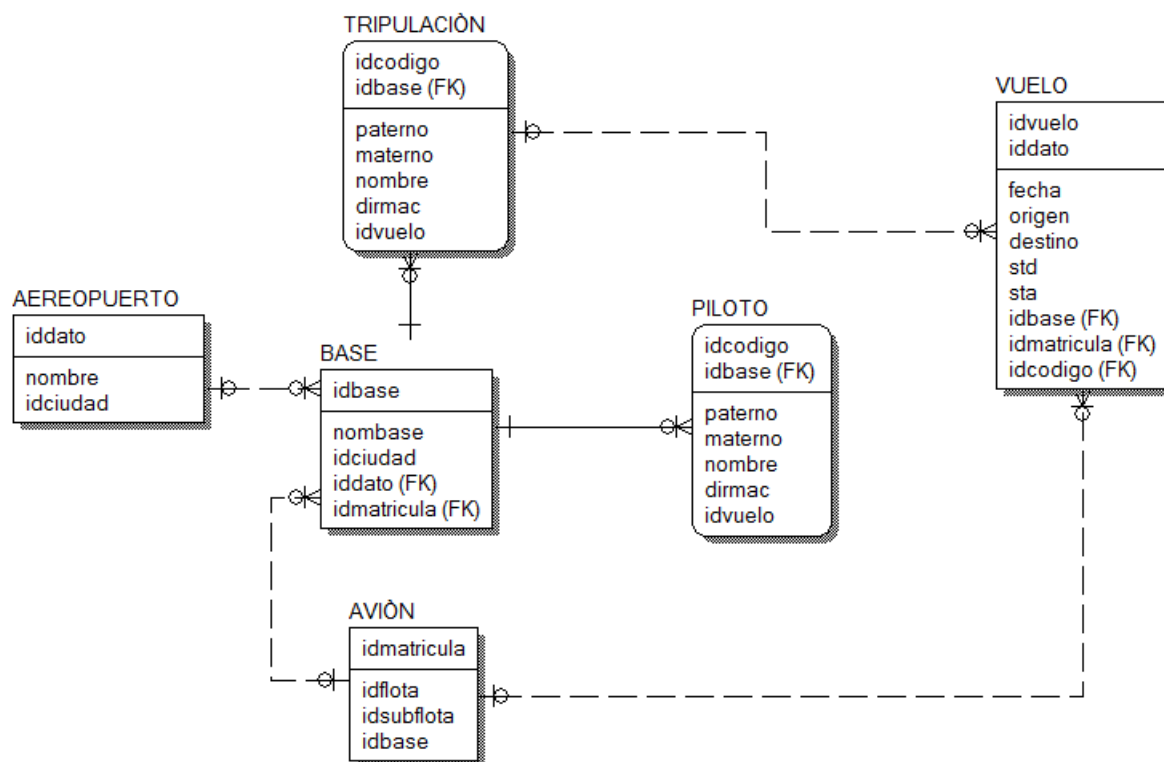


Figura 1: Modelo Lógico

3.2. Modelo Fisico

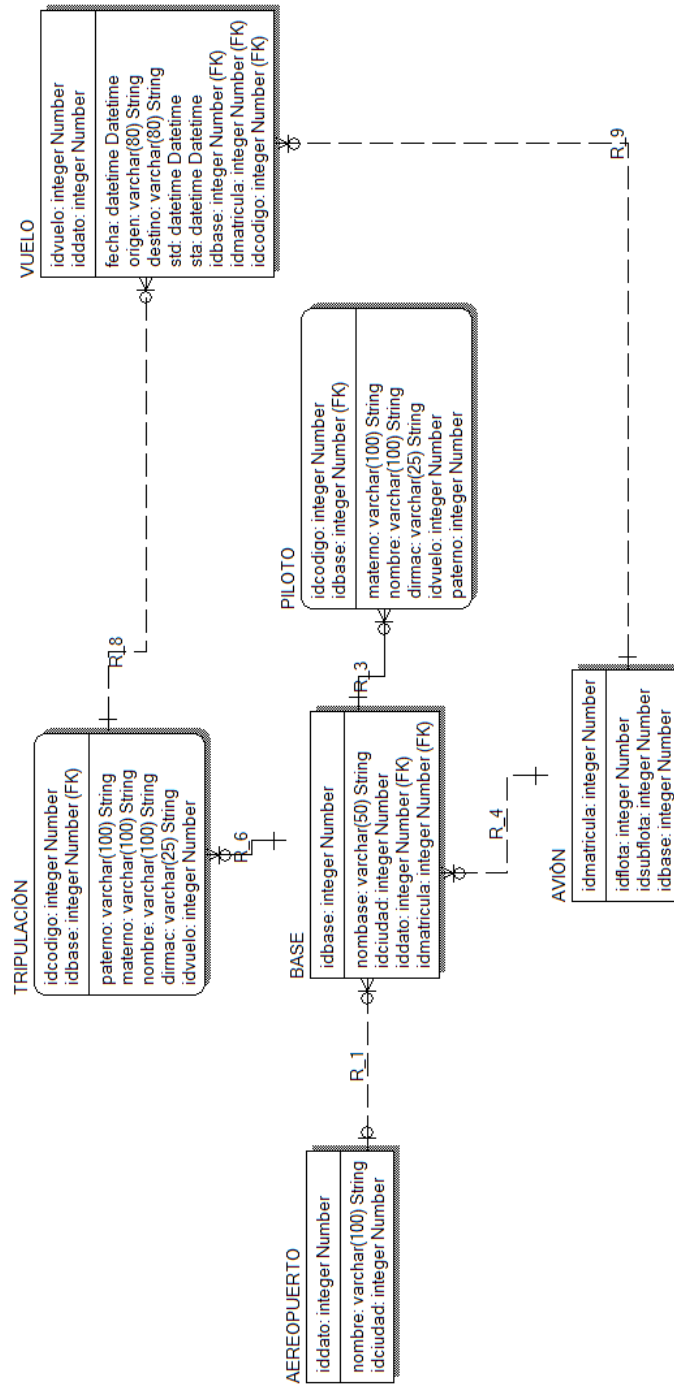
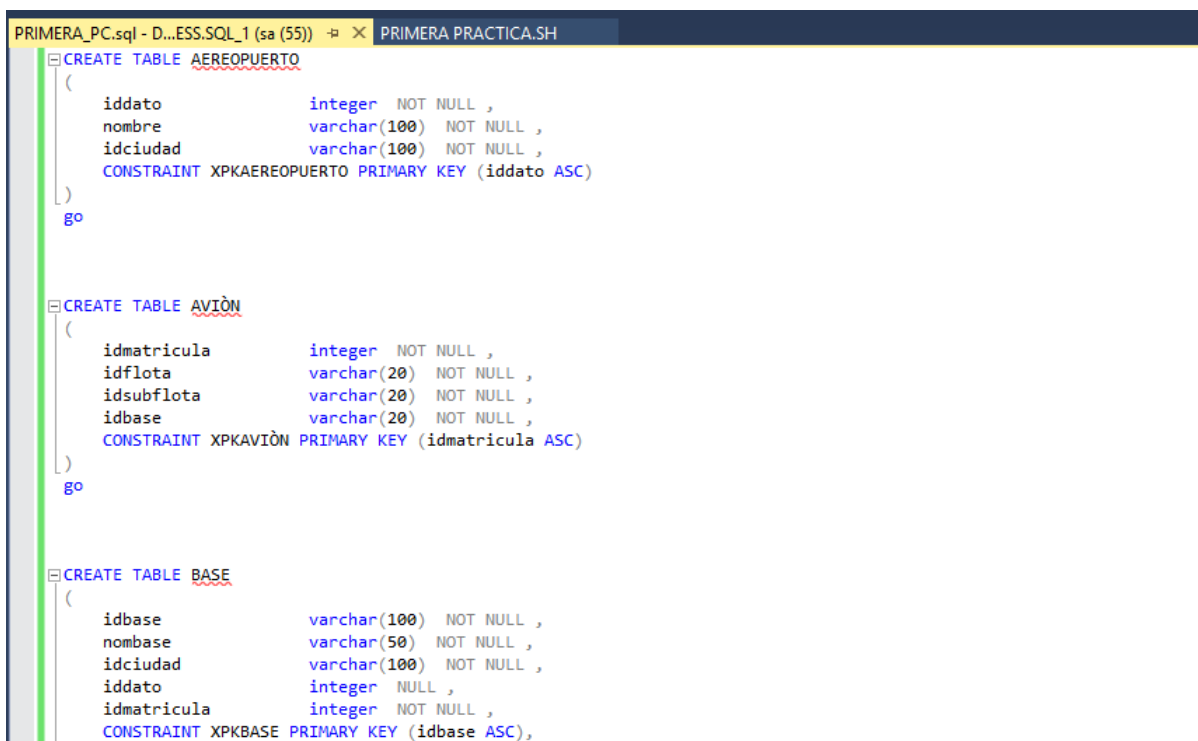


Figura 2: Modelo Fisico

4. Implementación de la Base de Datos

El primero paso será insertar el código del diseño de base de datos, mostrado anteriormente, de la empresa LATAM, para luego poder insertar los respectivos datos en dicha base.



```
PRIMERA_PC.sql - D...ESS.SQL_1 (sa (55)) -> X PRIMERA PRACTICA.SH
CREATE TABLE AEREOPUERTO
(
    iddato          integer NOT NULL ,
    nombre          varchar(100) NOT NULL ,
    idciudad        varchar(100) NOT NULL ,
    CONSTRAINT XPKAEREOPUERTO PRIMARY KEY (iddato ASC)
)
go

CREATE TABLE AVIÒN
(
    idmatricula     integer NOT NULL ,
    idflota         varchar(20) NOT NULL ,
    idsubflota      varchar(20) NOT NULL ,
    idbase          varchar(20) NOT NULL ,
    CONSTRAINT XPKAVIÒN PRIMARY KEY (idmatricula ASC)
)
go

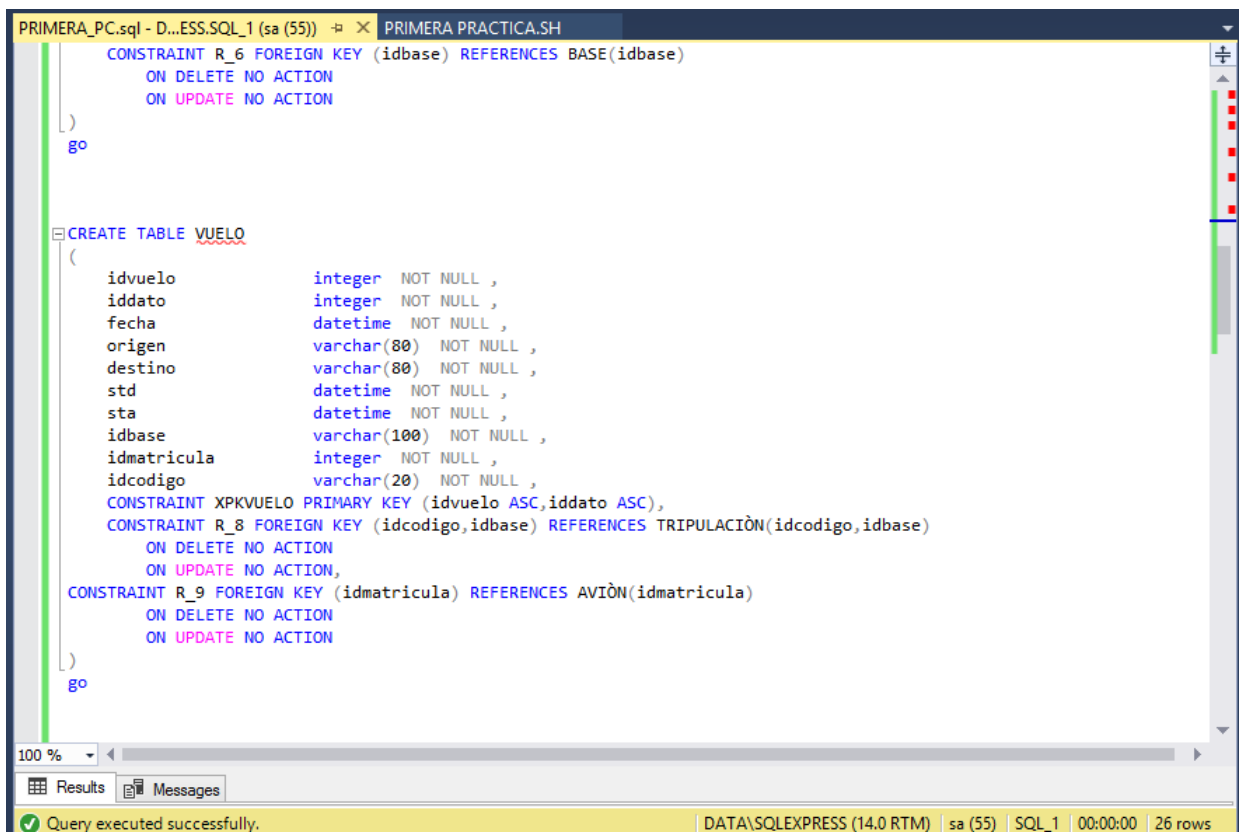
CREATE TABLE BASE
(
    idbase          varchar(100) NOT NULL ,
    nombase        varchar(50) NOT NULL ,
    idciudad        varchar(100) NOT NULL ,
    iddato          integer NULL ,
    idmatricula     integer NOT NULL ,
    CONSTRAINT XPKBASE PRIMARY KEY (idbase ASC),
```

Figura 3: Implementación de Base de Datos

```
PRIMERA_PC.sql - D:\ESS.SQL_1 (sa (55)) x PRIMERA PRACTICA.SH
CREATE TABLE BASE
(
    idbase          varchar(100) NOT NULL ,
    nombase         varchar(50)  NOT NULL ,
    idciudad        varchar(100) NOT NULL ,
    iddato          integer      NULL ,
    idmatricula     integer      NOT NULL ,
    CONSTRAINT XPKBASE PRIMARY KEY (idbase ASC),
    CONSTRAINT R_1 FOREIGN KEY (iddato) REFERENCES AEREOPUERTO(iddato)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT R_4 FOREIGN KEY (idmatricula) REFERENCES AVIÒN(idmatricula)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
go

CREATE TABLE TRIPULACIÒN
(
    idcodigo        varchar(20)  NOT NULL ,
    idbase          varchar(100) NOT NULL ,
    paterno         varchar(100) NOT NULL ,
    materno         varchar(100) NOT NULL ,
    nombre          varchar(100) NOT NULL ,
    dirmac          varchar(25)  NOT NULL ,
    idvuelo         varchar(20)  NOT NULL ,
    CONSTRAINT XPKTRIPULACIÒN PRIMARY KEY (idcodigo ASC,idbase ASC),
    CONSTRAINT R_6 FOREIGN KEY (idbase) REFERENCES BASE(idbase)
```

Figura 4: Implementaciòn de Base de Datos

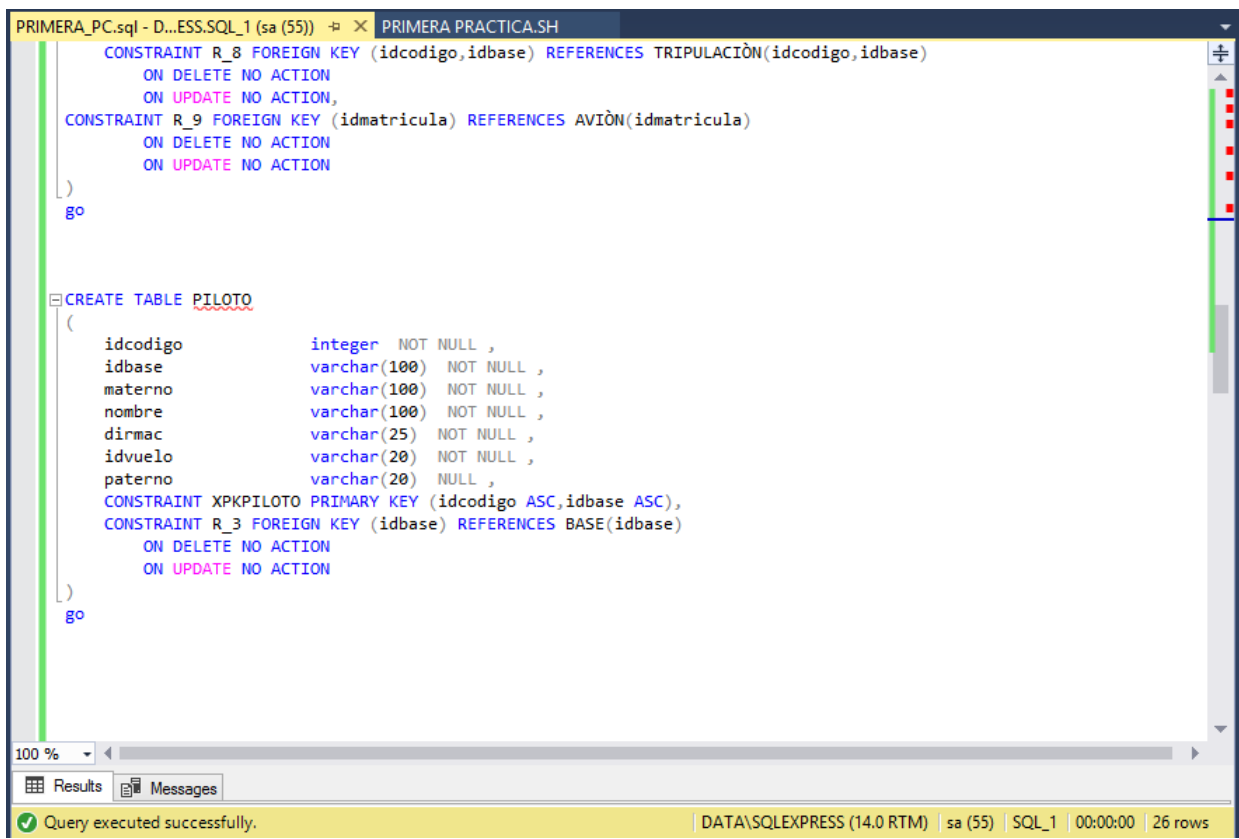


```
PRIMERA_PC.sql - D:\...ESS.SQL_1 (sa (55))  PRIMERA PRACTICA.SH
CONSTRAINT R_6 FOREIGN KEY (idbase) REFERENCES BASE(idbase)
ON DELETE NO ACTION
ON UPDATE NO ACTION
)
go

CREATE TABLE VUELO
(
    idvuelo          integer NOT NULL ,
    iddato           integer NOT NULL ,
    fecha            datetime NOT NULL ,
    origen           varchar(80) NOT NULL ,
    destino          varchar(80) NOT NULL ,
    std              datetime NOT NULL ,
    sta              datetime NOT NULL ,
    idbase           varchar(100) NOT NULL ,
    idmatricula      integer NOT NULL ,
    idcodigo         varchar(20) NOT NULL ,
    CONSTRAINT XPKVUELO PRIMARY KEY (idvuelo ASC,iddato ASC),
    CONSTRAINT R_8 FOREIGN KEY (idcodigo,idbase) REFERENCES TRIPULACIÓN(idcodigo,idbase)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT R_9 FOREIGN KEY (idmatricula) REFERENCES AVIÓN(idmatricula)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
)
go

100 %
Results Messages
Query executed successfully. DATA\SQLEXPRESS (14.0 RTM) sa (55) SQL_1 00:00:00 26 rows
```

Figura 5: Implementación de Base de Datos



```
PRIMERA_PC.sql - D:\ESS.SQL_1 (sa (55)) - PRIMERA PRACTICA.SH
    CONSTRAINT R_8 FOREIGN KEY (idcodigo,idbase) REFERENCES TRIPULACIÒN(idcodigo,idbase)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT R_9 FOREIGN KEY (idmatricula) REFERENCES AVIÒN(idmatricula)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
go

CREATE TABLE PILOTO
(
    idcodigo          integer NOT NULL ,
    idbase             varchar(100) NOT NULL ,
    materno            varchar(100) NOT NULL ,
    nombre             varchar(100) NOT NULL ,
    dirmac             varchar(25) NOT NULL ,
    idvuelo            varchar(20) NOT NULL ,
    paterno            varchar(20) NULL ,
    CONSTRAINT XPKPILOTO PRIMARY KEY (idcodigo ASC,idbase ASC),
    CONSTRAINT R_3 FOREIGN KEY (idbase) REFERENCES BASE(idbase)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
)
go

100 %
Results Messages
Query executed successfully. DATA\SQLEXPRESS (14.0 RTM) sa (55) SQL_1 00:00:00 26 rows
```

Figura 6: Implementaci3n de Base de Datos

Luego de relizar el c3digo de la base de datos, insertamos los datos en dicha base; y esto se puede lograr con la funci3n "INSERT INTO"

Como lo veremos a continuaci3n:

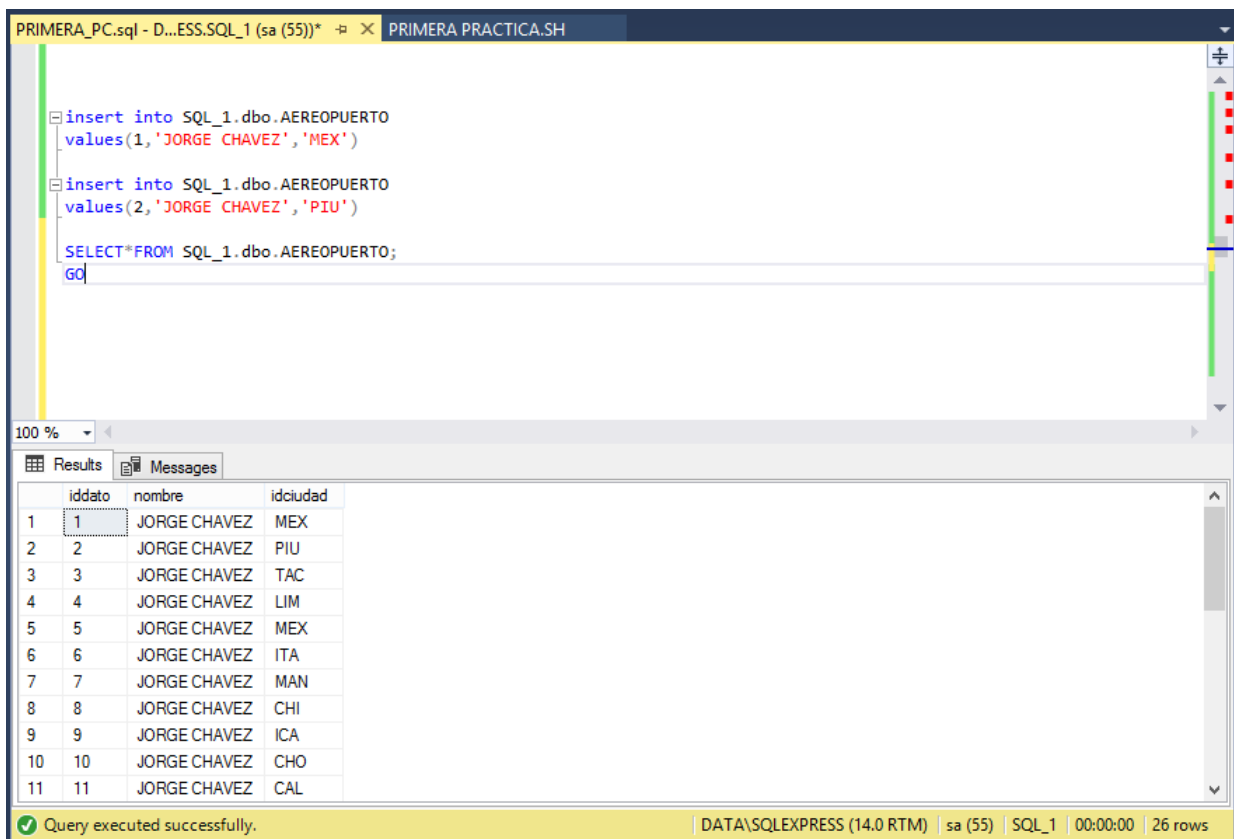


Figura 7: Implementaciòn de Base de Datos

5. Deasrrollo del Video

A continuaciòn mostraremos el link del video, en el cual explicamos el procedimineto utilizado para la presente pràctica calificada

<https://www.youtube.com/watch?v=oi3SG86tYnEfeature=youtu.be>

6. Conclusiones

- Podemos concluir que por el intermedio de la siguiente estructura podemos llevar un control mas optimizado del proceso de operaciones en latam.

- Llevando una adecuada estructura de la información, permitirá tomar mejores decisiones, ya que tendremos la información a la mano.
- Siempre mantener dentro de la base un lenguaje sencillo y claro para poder administrarla en el futuro.

7. Recomendaciones

- Cada cierto tiempo debemos dar mantenimiento a la estructura de datos para poder optimizarla.
- Debemos tratar en lo posible de que todas las áreas involucradas en la operatividad de la empresa tengan una estructura de datos.
- Tratar que siempre las bases estén interrelacionadas con el giro del negocio de esta manera se podrá evitar en un futuro cambiar su estructura.