



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. Tipos de Restricciones	3
1.1. Primary Key	3
1.1.1. Descripción	3
1.1.2. Sustento	3
1.1.3. Script para crear la restricción	4
1.1.4. Script de prueba	5
1.2. Unique	5
1.2.1. Descripción	5
1.2.2. Sustento	5
1.2.3. Script para crear la restricción	6
1.2.4. Script de prueba	7
1.3. Foreign key	7
1.3.1. Descripción	7
1.3.2. Sustento	7
1.3.3. Script para crear la restricción	8
1.3.4. Script de prueba	9
1.4. NULL / NOT NULL	12
1.4.1. Descripción	12
1.4.2. Sustento	12
1.4.3. Script para crear la restricción	13
1.4.4. Script de prueba	13
1.5. Check	14
1.5.1. Descripción	14
1.5.2. Sustento	14

1.5.3.	Script para crear la restricción	14
1.5.4.	Script de prueba	15
2.	Deasrrollo del Video	16
3.	Conclusiones	17
4.	Recomendaciones	17

SEGUNDA PRÁCTICA DE SQL I - IMPLEMENTACIÓN

21 de febrero de 2021

1. Tipos de Restricciones

1.1. Primary Key

1.1.1. Descripción

Esta restricción es una integridad de entidad(fila) y esta identifica cada registro o fila como única. Se crea un índice para mejorar el rendimiento. Los valores nulos no son permitidos. Es, decir los iddato no se deben repetir y tampoco deben ser nulos

1.1.2. Sustento

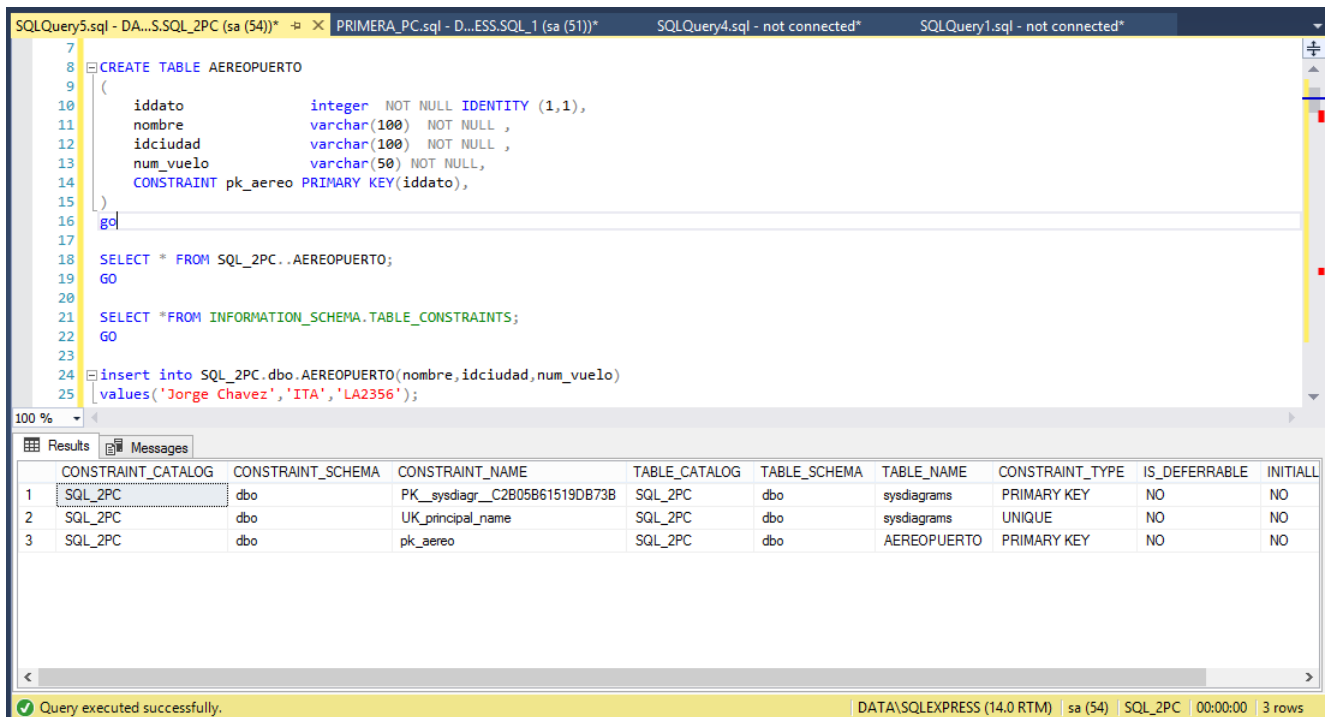
El iddato no puede ser negativo, esto es general. Por politicas de la empresa todos los vuelos se deben registrar, es por ello que no puede ser nulo.

1.1.3. Script para crear la restricción

```
8 CREATE TABLE AEREOPUERTO
9 (
10     iddato          integer NOT NULL IDENTITY (1,1),
11     nombre          varchar(100) NOT NULL ,
12     idciudad        varchar(100) NOT NULL ,
13     num_vuelo       varchar(50) NOT NULL,
14     CONSTRAINT pk_aereo PRIMARY KEY(iddato),
15 )
16 go
17
18 SELECT * FROM SQL_2PC..AEREOPUERTO;
19 GO
20
21 SELECT *FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS;
22 GO
23
```

Figura 1: Scrip de la restricción

1.1.4. Script de prueba



The screenshot shows a SQL Server Enterprise Manager window with a script editor and a results pane. The script editor contains the following SQL code:

```
7
8 CREATE TABLE AEREOPUERTO
9 (
10     iddato          integer NOT NULL IDENTITY (1,1),
11     nombre          varchar(100) NOT NULL ,
12     idciudad        varchar(100) NOT NULL ,
13     num_vuelo       varchar(50) NOT NULL,
14     CONSTRAINT pk_aereo PRIMARY KEY(iddato),
15 )
16
17
18 SELECT * FROM SQL_2PC..AEREOPUERTO;
19 GO
20
21 SELECT *FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS;
22 GO
23
24 insert into SQL_2PC.dbo.AEREOPUERTO(nombre,idciudad,num_vuelo)
25 values('Jorge Chavez','ITA','LA2356');
```

The results pane shows the output of the third query, displaying a table with 10 columns: CONSTRAINT_CATALOG, CONSTRAINT_SCHEMA, CONSTRAINT_NAME, TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, CONSTRAINT_TYPE, IS_DEFERRABLE, and INITIAL.

	CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE	IS_DEFERRABLE	INITIAL
1	SQL_2PC	dbo	PK_sysdiagr__C2B05B61519DB73B	SQL_2PC	dbo	sysdiagrams	PRIMARY KEY	NO	NO
2	SQL_2PC	dbo	UK_principal_name	SQL_2PC	dbo	sysdiagrams	UNIQUE	NO	NO
3	SQL_2PC	dbo	pk_aereo	SQL_2PC	dbo	AEREOPUERTO	PRIMARY KEY	NO	NO

The status bar at the bottom indicates: Query executed successfully. DATA\SQLEXPRESS (14.0 RTM) | sa (54) | SQL_2PC | 00:00:00 | 3 rows

Figura 2: Script de prueba

1.2. Unique

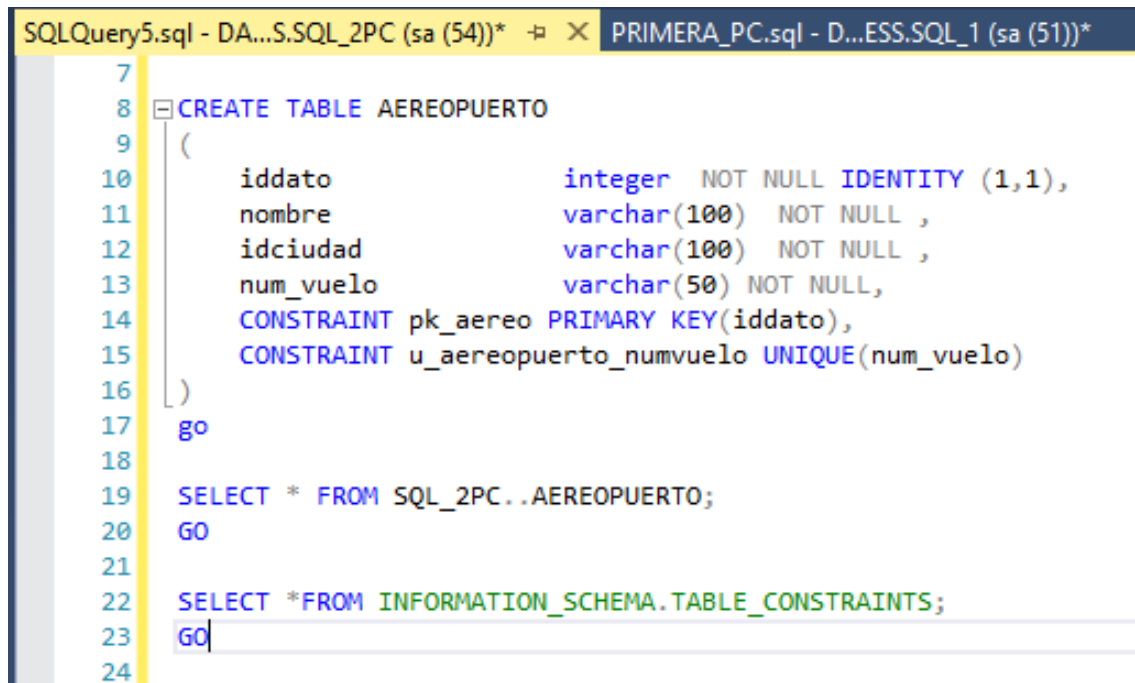
1.2.1. Descripción

Esta restricción es una integridad de entidad(fila) y previene la duplicación de las llaves alternas, y se asegura que un índice se crea para que mejore el rendimiento. Se permiten valores nulos Es decir estos datos no se pueden repetir, es decir en la empresa, no se debe repetir el nombre delaereoperto y el lugar de destino

1.2.2. Sustento

Segun la empresa el numero de vuelo debe ser unico este no se puede repetir y además no puede ser nulo

1.2.3. Script para crear la restricción



```
SQLQuery5.sql - DA...S.SQL_2PC (sa (54))* X PRIMERA_PC.sql - D...ESS.SQL_1 (sa (51))*
7
8 CREATE TABLE AEREOPUERTO
9 (
10     iddato          integer NOT NULL IDENTITY (1,1),
11     nombre          varchar(100) NOT NULL ,
12     idciudad        varchar(100) NOT NULL ,
13     num_vuelo       varchar(50) NOT NULL,
14     CONSTRAINT pk_aereo PRIMARY KEY(iddato),
15     CONSTRAINT u_aereopuerto_numvuelo UNIQUE(num_vuelo)
16 )
17 go
18
19 SELECT * FROM SQL_2PC..AEREOPUERTO;
20 GO
21
22 SELECT *FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS;
23 GO
24
```

Figura 3: Scrip de la restricción

1.2.4. Script de prueba

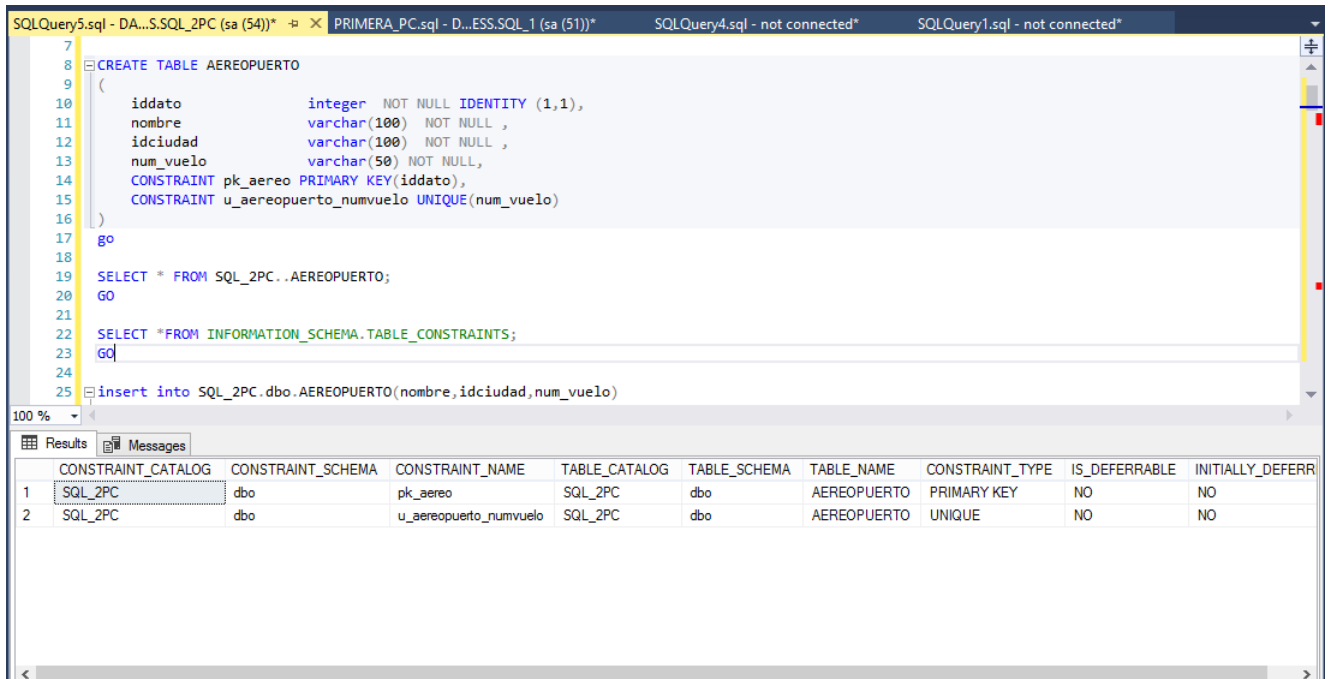


Figura 4: Script de prueba

1.3. Foreign key

1.3.1. Descripción

Define la columna o combinación de columnas de una tabla secundaria, cuyos valores dependen de la llave primaria de una tabla primaria

1.3.2. Sustento

En este caso estamos definiendo el foreign key con la tabla Aereopuerto y la Tabla Base

1.3.3. Script para crear la restricción

```
--
48  /*Tabla AEREOPUERTO*/
49
50  CREATE TABLE AEREOPUERTO
51  (
52      iddato          integer NOT NULL IDENTITY (001,1),
53      nombre          varchar(100) NOT NULL ,
54      idciudad        varchar(100) NOT NULL ,
55      num_vuelo       varchar(50) NOT NULL,
56      CONSTRAINT pk_aereo PRIMARY KEY(iddato),
57      CONSTRAINT u_aereopuerto_numvuelo UNIQUE(num_vuelo)
58  )
59  go
60
61  insert into SQL_2PC.dbo.AEREOPUERTO(nombre,idciudad,num_vuelo)
62  values('Augusto Severo','MEX','LA3569');
63  GO
64
65  insert into SQL_2PC.dbo.AEREOPUERTO(nombre,idciudad,num_vuelo)
66  values('Jorge Cahvez','ITA','LA2719');
67  GO
68
69  insert into SQL_2PC.dbo.AEREOPUERTO(nombre,idciudad,num_vuelo)
70  values('Jorge Cahvez','PAR','LE16657');
71  GO
72
73  SELECT * FROM SQL_2PC..AEREOPUERTO;
74  GO
75
--
```

Figura 5: TABLA AEREOPUERTO

```

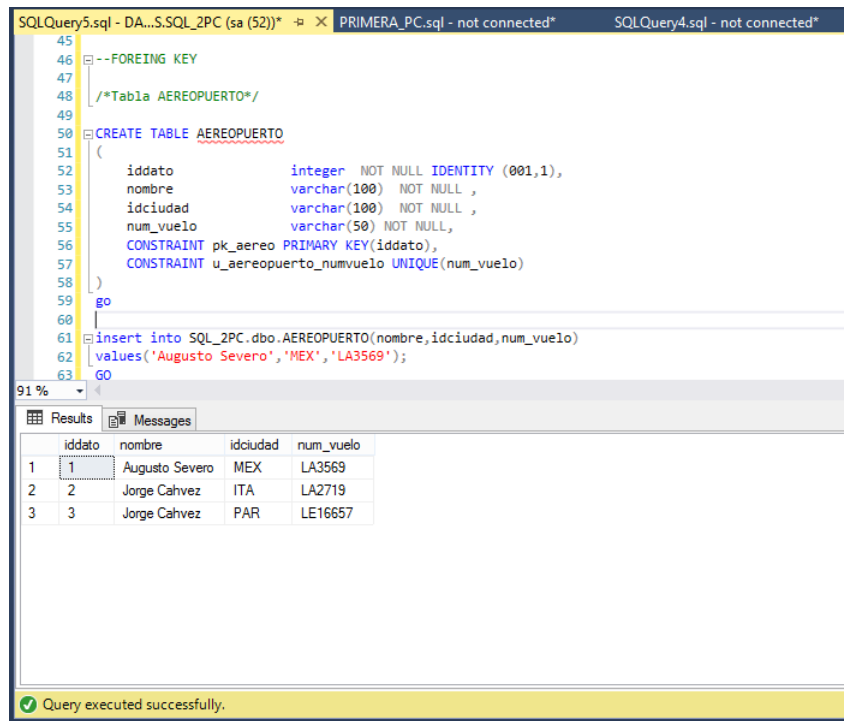
77  /*Tabla BASE*/
78
79  CREATE TABLE BASE
80  (
81      idbase          varchar(100) NOT NULL ,
82      nombase         varchar(50)  NOT NULL ,
83      iddato          integer  NULL ,
84      idmatricula     integer  NOT NULL ,
85      CONSTRAINT PK_BASE PRIMARY KEY (idbase ),
86      CONSTRAINT FK_BASE_iddato
87          FOREIGN KEY (iddato)
88          REFERENCES AEREOPUERTO(iddato)
89  )
90  go
91
92  insert into SQL_2PC.dbo.BASE(idbase,nombase,iddato,idmatricula)
93  values('dvd','vtg675',1,23);
94  GO
95
96
97  SELECT * FROM SQL_2PC.dbo.BASE;
98  go
99
100 insert into SQL_2PC.dbo.BASE(idbase,nombase,iddato,idmatricula)
101 values('dvd','vtg675',4,23);
102 GO
103

```

Figura 6: TABLA BASE

1.3.4. Script de prueba

En este ejemplo vemos que el iddato puede tomar 3 valores



```
45
46 --FOREIGN KEY
47
48 /*Tabla AEREOPUERTO*/
49
50 CREATE TABLE AEREOPUERTO
51 (
52     iddato          integer NOT NULL IDENTITY (001,1),
53     nombre          varchar(100) NOT NULL ,
54     idciudad        varchar(100) NOT NULL ,
55     num_vuelo       varchar(50) NOT NULL ,
56     CONSTRAINT pk_aereo PRIMARY KEY(iddato),
57     CONSTRAINT u_aereopuerto_numvuelo UNIQUE(num_vuelo)
58 )
59 GO
60
61 insert into SQL_2PC.dbo.AEREOPUERTO(nombre,idciudad,num_vuelo)
62 values('Augusto Severo','MEX','LA3569');
63 GO
```

	iddato	nombre	idciudad	num_vuelo
1	1	Augusto Severo	MEX	LA3569
2	2	Jorge Cahvez	ITA	LA2719
3	3	Jorge Cahvez	PAR	LE16657

91 %

Query executed successfully.

Figura 7: TABLA AEREOPUERTO

En la tabla base, veremos que cuando incluimos la restriccion de Foreign Key, solo puede tomar dichos valores ya que si toma otros sale error, como veremos a continuaciòn:

```
77 /*Tabla BASE*/
78
79 CREATE TABLE BASE
80 (
81     idbase          varchar(100) NOT NULL ,
82     nombase         varchar(50)  NOT NULL ,
83     iddato          integer      NULL ,
84     idmatricula     integer      NOT NULL ,
85     CONSTRAINT PK_BASE PRIMARY KEY (idbase ),
86     CONSTRAINT FK_BASE_iddato
87         FOREIGN KEY (iddato)
88         REFERENCES AEREOPUERTO(iddato)
89 )
90 go
91
92 insert into SQL_2PC.dbo.BASE(idbase,nombase,iddato,idmatricula)
93 values('LIM','vtg675',3,23);
94 GO
95
96
```

91 %

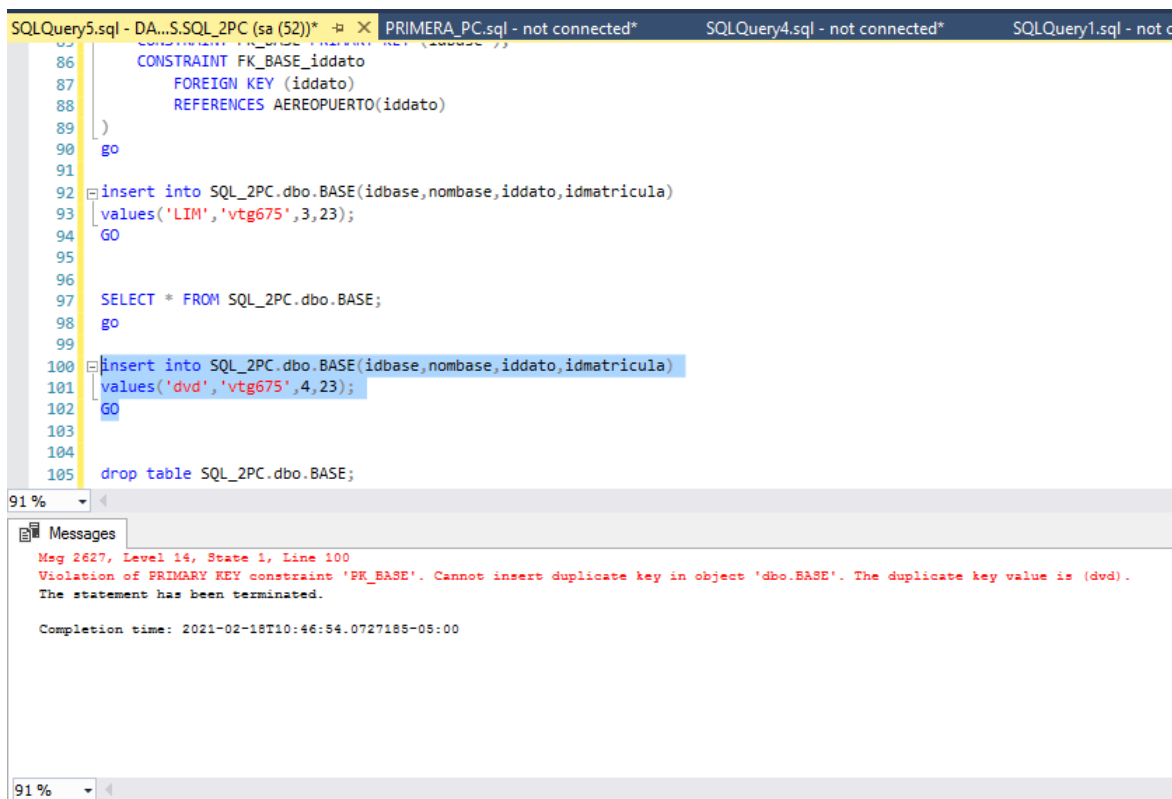
Results Messages

	idbase	nombase	iddato	idmatricula
1	dvd	vtg675	1	23
2	LIM	vtg675	3	23

Query executed successfully.

Figura 8: TABLA BASE

A continuaci3n vemos que existe error debido a que no existe el valor 4 en iddato



The screenshot shows a SQL Server Enterprise Manager interface. At the top, there are several tabs: 'SQLQuery5.sql - DA...S.SQL_2PC (sa (52))', 'PRIMERA_PC.sql - not connected', 'SQLQuery4.sql - not connected', and 'SQLQuery1.sql - not c'. The active window is 'SQLQuery5.sql'. It contains a SQL script with the following lines:

```
86 CONSTRAINT FK_BASE_iddato
87 FOREIGN KEY (iddato)
88 REFERENCES AEREOPUERTO(iddato)
89 )
90 go
91
92 insert into SQL_2PC.dbo.BASE(idbase,nombase,iddato,idmatricula)
93 values('LIM','vtg675',3,23);
94 GO
95
96 SELECT * FROM SQL_2PC.dbo.BASE;
97 go
98
99
100 insert into SQL_2PC.dbo.BASE(idbase,nombase,iddato,idmatricula)
101 values('dvd','vtg675',4,23);
102 GO
103
104
105 drop table SQL_2PC.dbo.BASE;
```

Below the script, there is a 'Messages' pane showing an error message:

```
Msg 2627, Level 14, State 1, Line 100
Violation of PRIMARY KEY constraint 'FK_BASE'. Cannot insert duplicate key in object 'dbo.BASE'. The duplicate key value is (dvd).
The statement has been terminated.

Completion time: 2021-02-18T10:46:54.0727185-05:00
```

Figura 9: Error

1.4. NULL / NOT NULL

1.4.1. Descripción

Especifica el valor que se mantendrá para la columna cuando un valor no se ha ingresado explícitamente en una sentencia INSERT. Es el valor por defecto que toma una columna cuando no se especifica un valor, el valor por defecto es NULL

1.4.2. Sustento

Si las columnas de nuestra tabla no aceptan Null, se produce error

1.4.3. Script para crear la restricción

```
SQLQuery5.sql - DA...S.SQL_2PC (sa (52))* X PRIMERA_PC.sql - not connected*
117  /*NOT NULL/NULL*/
118
119
120  /*Tabla AEREOPUERTO*/
121
122  CREATE TABLE AEREOPUERTO
123  (
124      iddato          integer NOT NULL IDENTITY (500,1),
125      nombre          varchar(100) NOT NULL DEFAULT 'None',
126      idciudad        varchar(100) NULL ,
127      num_vuelo       varchar(50) NOT NULL DEFAULT 'LA1234',
128      CONSTRAINT pk_aereo PRIMARY KEY(iddato),
129      CONSTRAINT u_aereopuerto_numvuelo UNIQUE(num_vuelo)
130  )
131  go
132
133  insert into SQL_2PC.dbo.AEREOPUERTO(nombre)
134  values('Jorge Cahvez');
135  go
136
137  SELECT * FROM SQL_2PC.dbo.AEREOPUERTO;
138  go
139
```

Figura 10: Scrip de la restricciòn

1.4.4. Script de prueba

```
SQLQuery5.sql - DA...S.SQL_2PC (sa (52))* X PRIMERA_PC.sql - not connected*
117  /*NOT NULL/NULL*/
118
119
120  /*Tabla AEREOPUERTO*/
121
122  CREATE TABLE AEREOPUERTO
123  (
124      iddato          integer NOT NULL IDENTITY (500,1),
125      nombre          varchar(100) NOT NULL DEFAULT 'None',
126      idciudad        varchar(100) NULL ,
127      num_vuelo       varchar(50) NOT NULL DEFAULT 'LA1234',
128      CONSTRAINT pk_aereo PRIMARY KEY(iddato),
129      CONSTRAINT u_aereopuerto_numvuelo UNIQUE(num_vuelo)
130  )
131  go
132
133  insert into SQL_2PC.dbo.AEREOPUERTO(nombre)
134  values('Jorge Cahvez');
135  go
136
137  SELECT * FROM SQL_2PC.dbo.AEREOPUERTO;
138  go
139
```

Figura 11: Scrip de la restricciòn

1.5. Check

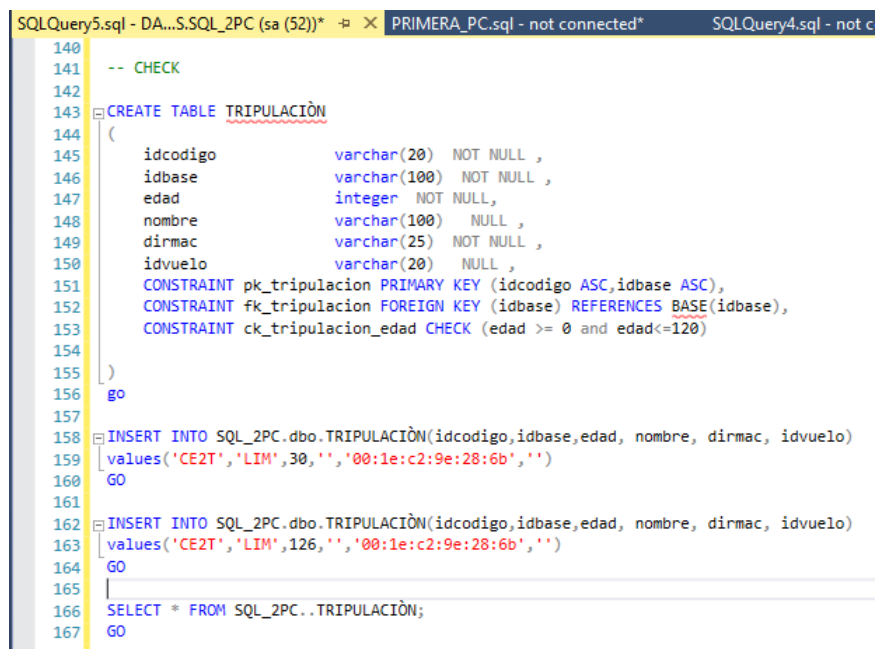
1.5.1. Descripción

Especifica los valores de los datos que son aceptables en la columna

1.5.2. Sustento

Es decir los valores que se insertan en una columna deben cumplir una condición

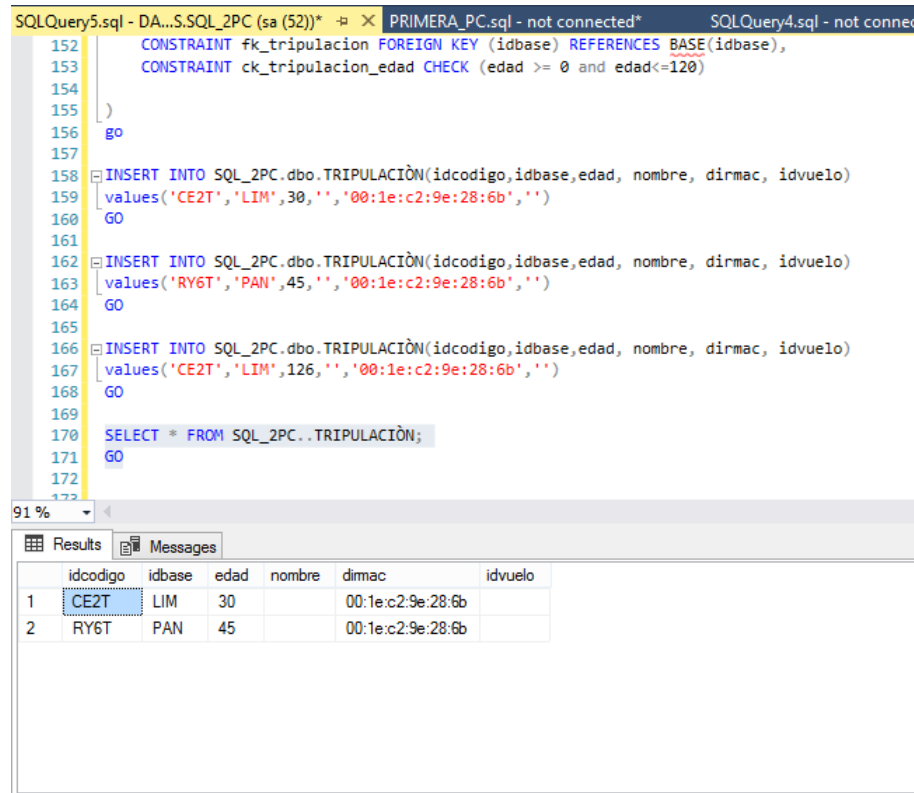
1.5.3. Script para crear la restricción



```
SQLQuery5.sql - DA...S.SQL_2PC (sa (52))*  PRIMERA_PC.sql - not connected*  SQLQuery4.sql - not c
140
141 -- CHECK
142
143 CREATE TABLE TRIPULACIÒN
144 (
145     idcodigo          varchar(20) NOT NULL ,
146     idbase            varchar(100) NOT NULL ,
147     edad              integer NOT NULL,
148     nombre            varchar(100) NULL ,
149     dirmac            varchar(25) NOT NULL ,
150     idvuelo           varchar(20) NULL ,
151     CONSTRAINT pk_tripulacion PRIMARY KEY (idcodigo ASC,idbase ASC),
152     CONSTRAINT fk_tripulacion FOREIGN KEY (idbase) REFERENCES BASE(idbase),
153     CONSTRAINT ck_tripulacion_edad CHECK (edad >= 0 and edad<=120)
154 )
155
156 go
157
158 INSERT INTO SQL_2PC.dbo.TRIPULACIÒN(idcodigo,idbase,edad, nombre, dirmac, idvuelo)
159 values('CE2T','LIM',30,'','00:1e:c2:9e:28:6b','')
160 GO
161
162 INSERT INTO SQL_2PC.dbo.TRIPULACIÒN(idcodigo,idbase,edad, nombre, dirmac, idvuelo)
163 values('CE2T','LIM',126,'','00:1e:c2:9e:28:6b','')
164 GO
165
166 SELECT * FROM SQL_2PC..TRIPULACIÒN;
167 GO
168
```

Figura 12: Scrip de la restricciòn

1.5.4. Script de prueba



```
SQLQuery5.sql - DA...S.SQL_2PC (sa (52))*  PRIMER_A_PC.sql - not connected*  SQLQuery4.sql - not connec
152     CONSTRAINT fk_tripulacion FOREIGN KEY (ibase) REFERENCES BASE(ibase),
153     CONSTRAINT ck_tripulacion_edad CHECK (edad >= 0 and edad<=120)
154
155 )
156 GO
157
158 INSERT INTO SQL_2PC.dbo.TRIPULACION(idcodigo,ibase,edad, nombre, dirmac, idvuelo)
159 values('CE2T','LIM',30,'','00:1e:c2:9e:28:6b','')
160 GO
161
162 INSERT INTO SQL_2PC.dbo.TRIPULACION(idcodigo,ibase,edad, nombre, dirmac, idvuelo)
163 values('RY6T','PAN',45,'','00:1e:c2:9e:28:6b','')
164 GO
165
166 INSERT INTO SQL_2PC.dbo.TRIPULACION(idcodigo,ibase,edad, nombre, dirmac, idvuelo)
167 values('CE2T','LIM',126,'','00:1e:c2:9e:28:6b','')
168 GO
169
170 SELECT * FROM SQL_2PC..TRIPULACION;
171 GO
172
173
```

91 %

Results Messages

	idcodigo	ibase	edad	nombre	dirmac	idvuelo
1	CE2T	LIM	30		00:1e:c2:9e:28:6b	
2	RY6T	PAN	45		00:1e:c2:9e:28:6b	

Figura 13: Scrip de la restricciòn

Cuando introducimos un valor diferente al rango, notamos que se produce un error, como lo veremos a continuaciòn:


```
SQLQuery5.sql - DA...S.SQL_2PC (sa (52)) *  PRIMER_A_PC.sql - not connected*  SQLQuery4.sql - not connected*  SQLQuery1.sql - not connected*
152     CONSTRAINT fk_tripulacion FOREIGN KEY (idbase) REFERENCES BASE(idbase),
153     CONSTRAINT ck_tripulacion_edad CHECK (edad >= 0 and edad<=120)
154
155 )
156 GO
157
158 INSERT INTO SQL_2PC.dbo.TRIPULACIÓN(idcodigo,idbase,edad, nombre, dirmac, idvuelo)
159 values('CE2T','LIM',30,'','00:1e:c2:9e:28:6b','')
160 GO
161
162 INSERT INTO SQL_2PC.dbo.TRIPULACIÓN(idcodigo,idbase,edad, nombre, dirmac, idvuelo)
163 values('RY6T','PAN',45,'','00:1e:c2:9e:28:6b','')
164 GO
165
166 INSERT INTO SQL_2PC.dbo.TRIPULACIÓN(idcodigo,idbase,edad, nombre, dirmac, idvuelo)
167 values('CE2T','LIM',126,'','00:1e:c2:9e:28:6b','')
168 GO
169
170 SELECT * FROM SQL_2PC..TRIPULACIÓN;
171 GO
172
173
91 %
Messages
Msg 2627, Level 14, State 1, Line 166
Violation of PRIMARY KEY constraint 'pk_tripulacion'. Cannot insert duplicate key in object 'dbo.TRIPULACIÓN'. The duplicate key value is (CE2T, LIM).
The statement has been terminated.
Completion time: 2021-02-18T12:22:36.3536550-05:00
91 %
```

Figura 14: Error

2. Deasrrollo del Video

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

<https://youtu.be/LcHZfLml5Sc>

3. 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.

4. 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 esten interrelacionadas con el giro del negocio de esta manera se podra evitar en un futuro cambiar su estructura.