

# SQL - Manipulación de tablas

## Tabla de contenido

<b>SQL - Manipulación de tablas .....</b>	<b>1</b>
<b>Nivel 1 .....</b>	<b>2</b>
- Ejercicio 1 .....	2
- Ejercicio 2 .....	5
- Ejercicio 3 .....	6
<b>Nivel 2 .....</b>	<b>9</b>
Ejercicio 1 .....	9
Ejercicio 2.....	9
Ejercicio 3.....	9
<b>Nivel 3 .....</b>	<b>9</b>
Ejercicio 1 .....	9
Ejercicio 2.....	9

# Nivel 1

## - Ejercicio 1

Tu tarea es diseñar y crear una tabla llamada "credit\_card" que almacene detalles cruciales sobre las tarjetas de crédito. La nueva tabla debe ser capaz de identificar de forma única cada tarjeta y establecer una relación adecuada con las otras dos tablas ("transaction" y "company"). Después de crear la tabla será necesario que ingreses la información del documento denominado "datos\_introducir\_credit". Recuerda mostrar el diagrama y realizar una breve descripción del mismo.

Diseñamos y creamos una tabla llamada credit\_card. Para determinar los campos observamos en recursos los campos que luego tendremos que insertar, ya que el diseño tiene que permitir que éstos se puedan ingresen. Así vemos que es conveniente utilizar tipos de datos de texto varchar tanto para la fecha como para datos estrictamente numéricos como pin pan o cvv. Esto se debe a que no realizaremos operaciones con estos números y si empiezan con "0" pueden traernos algún problema al momento de la inserción. En cuanto a la fecha, también, el tipo "date" corresponde a DD-MM-YYYY, y las fechas que tenemos que insertar tienen el formato MM-DD-YY. Por tanto, armamos una estructura de texto varchar y luego de la inserción ya veremos si tenemos que modificarlo.

Al crear la tabla asignamos el campo id como PK para identificar de forma única cada tarjeta.

```
create table if not exists credit_card (  
    id varchar(15) primary key,  
    iban varchar(150),  
    pan varchar(150),  
    pin varchar(100),  
    cvv varchar(100),  
    expiring_date varchar(100)  
);
```

Además, luego de crear la tabla, establecemos como FK de la tabla transactions el campo credit\_card\_id, referenciando al campo id de la tabla credit\_card que acabamos de crear.

```
-- creamos la FK en la tabla de hechos transaction  
alter table transaction  
add foreign key (credit_card_id) references credit_card(id);
```

También creamos un índice para el campo id de la tabla credit\_card para agilizar las búsquedas y filtros de la tabla.

```
-- creamos el índice  
create index idx_credit_card_id on transaction(credit_card_id);
```

TASCA SPRINT 3

De esta manera establecemos las relaciones adecuadas con las otras tablas existentes “transaction” y “company”

```
28 • show tables;
29 • desc credit_card;
30 • desc transaction;
31
```

100% 11:29

Result Grid Filter Rows: Search Export:

	Field	Type	Null	Key	Default	Extra
▶	id	varchar(15)	NO	PRI	NULL	
	iban	varchar(150)	YES		NULL	
	pan	varchar(150)	YES		NULL	
	pin	varchar(100)	YES		NULL	
	cvv	varchar(100)	YES		NULL	
	expiring_date	varchar(100)	YES		NULL	

Finalmente insertamos los datos de "datos\_introducir\_credit".

datos\_introducir\_user (1) manipulando tablas datos\_introducir\_credit t\_transaction\* t\_credit\_card\* Garcia\_Scarafia-Sprint\_3\*

Limit to 1000 rows

```

1
2 -- Insertamos datos de credit_card
3 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2938', 'TR301950312213576817638'
4 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2945', 'D0268547637485374752165'
5 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2952', 'BG45IVQL52710525608255',
6 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2959', 'CR7242477244335841535',
7 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2966', 'BG72LKTQ15627628377363',
8 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2973', 'PT878062281350924294563'
9 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2980', 'DE39241881883086277136',
10 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2987', 'GE89681434837748781813',
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2994', 'BH62714428368066765294',
12 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3001', 'CY490874266547745812668'
13 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3008', 'LU507216693616119230',
14 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3015', 'PS119398216295715968342'
15 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3022', 'GT916951628505569774231'
16 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3029', 'AZ623174139824414181237'
17 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3036', 'A7303360020258428658430'

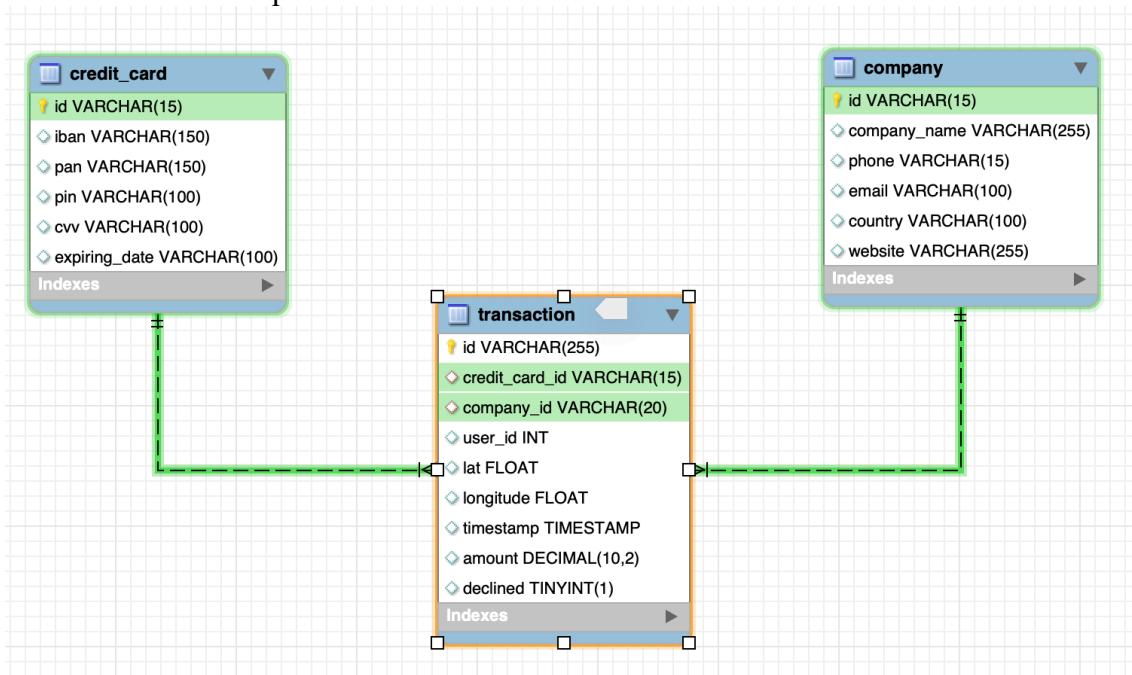
```

Action Output

	Time	Action	Response	Duration / Fets
✓ 273	17:47:56	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (...)	1 row(s) affected	0.026 sec
✓ 274	17:47:56	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (...)	1 row(s) affected	0.026 sec
✓ 275	17:47:56	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (...)	1 row(s) affected	0.026 sec
✓ 276	17:47:57	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (...)	1 row(s) affected	0.026 sec
✓ 277	17:47:57	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (...)	1 row(s) affected	0.027 sec
✓ 278	17:47:57	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (...)	1 row(s) affected	0.026 sec
✓ 279	17:47:57	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (...)	1 row(s) affected	0.026 sec

TASCA SPRINT 3

Visualización del esquema:



Observamos que la nueva tabla `credit_card` se relaciona con su PK `id` con la FK de la tabla `transaction` `credit_card_id`. La relación es de uno a muchos de la tabla de dimensiones a la tabla de hechos. Significa que una tarjeta de crédito puede estar en una o muchas transacciones, mientras que cada transacción puede tener una única tarjeta de crédito.

## - Ejercicio 2

El departamento de Recursos Humanos ha identificado un error en el número de cuenta del usuario con ID CcU-2938. La información que debe mostrarse para este registro es: R323456312213576817699999. Recuerda mostrar que el cambio se realizó.

Exploramos el registro que se busca cambiar:

```
37
38 • select iban from credit_card where id = 'CcU-2938';
39
40 • update credit_card set iban = 'R323456312213576817699999' where id = 'CcU-2938';
41
```

100% 26:38

Result Grid Filter Rows: Search Export:

iban
TR301950312213576817638661

Lo modificamos y mostramos el resultado:

```
37
38 • select iban from credit_card where id = 'CcU-2938';
39
40 • update credit_card set iban = 'R323456312213576817699999' where id = 'CcU-2938';
41
```

100% 45:38

Result Grid Filter Rows: Search Export:

iban
R323456312213576817699999

Para modificar datos dentro de una tabla usamos el comando:

UPDATE 'tabla' SET 'campo\_a\_modificar' = 'nuevoRegistro' WHERE ...

## - Ejercicio 3

En la tabla "transaction" ingresa un nuevo usuario con la siguiente información:

Id	108B1D1D-5B23-A76C-55EF-C568E49A99DD
credit_card_id	CcU-9999
company_id	b-9999
user_id	9999
lat	829.999
longitud	-117.999
amunt	111.11
declined	0

Ante una inserción de datos en la tabla transaction, verificamos qué campos tiene esta tabla y observamos que es necesario agregar el registro del campo timestamp. Para ello, entre la longitud y el amount, agregamos `current_date()`.

```
56 -- verifico los campos de la tabla transaction para ver los tipos de datos a insertar, agrego el campo timestamp que faltaba.
57 desc transaction;
58 insert into transaction(id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)
59 values('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, current_date(), 111.11, 0);
```

Además, verificamos que la compañía que queremos ingresar no está en la tabla compañías.

```
0 -- confirmo que la compañía no existe en la tabla compañía, verifico los campos de la tabla y agrego el id
1 select * from company
2 where id = 'b-9999';
3 desc company;
```

TASCA SPRINT 3

Agregamos el id a compañía y confirmamos que el registro ya esté en la tabla compañía

```
61 • select * from company
62   where id = 'b-9999';
63 • desc company;
64 • insert into company(id) values ('b-9999');
65 -- visualizo la transacción insertada
```

100% 20:60

Result Grid Filter Rows: Search Edit: Export/Im

	id	company_name	phone	email	country	website
▶	b-9999	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	NULL	NULL	NULL	NULL

Hacemos lo propio con el id en la tabla credit card:

```
63 -- verificación del cumplimiento de las restricciones de clave externa (FK) en credit_card (id)
64 • select id from credit_card where id = 'CcU-9999';
65 • insert into credit_card(id) values ('CcU-9999');
66
67 -- ingreso el nuevo usuario solicitado
```

100% 29:64

Result Grid Filter Rows: Search Edit: Export/Import:

	id
▶	CcU-9999
	NULL

Finalmente ingresamos los datos

```
66
67 -- ingreso el nuevo usuario solicitado
68 • insert into transaction(id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)
69   values('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, current_date(), 111.11, 0);
```

100% 24:67

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 20	01:09:31	insert into transaction(id, credit_c...	1 row(s) affected	0.043 sec

Y verificamos que hayan ingresado correctamente:

```
70 -- visualizo la transacción insertada
71 • select * from transaction
72   order by timestamp desc
73   limit 5;
```

100% 26:70

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

	id	credit_card_id	company_id	use...	lat	longitude	timestamp	amount	...
▶	108B1D1D-5...	CcU-9999	b-9999	9999	829.999	-117.999	2024-06-03...	111.11	0
	69D90229-A...	CcU-2959	b-2362	92	4.07537	129.723	2022-03-16...	329.03	0
	FD89D51B-A...	CcU-3960	b-2494	250	8.01217	-64.6715	2022-03-16...	200.72	0
	CA515AD3-5...	CcU-4709	b-2278	206	80.0508	-167.602	2022-03-15...	25.78	0
	DC5ABC81-6...	CcU-3799	b-2618	261	-6.94376	-56.6999	2022-03-15...	101.32	0
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 14

Action Output

	Time	Action	Response
✓ 21	01:10:55	select * from transaction order by...	5 row(s) returned

## - Ejercicio 4

Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit\_card. Recuerda mostrar el cambio realizado.

Visualizo los campos existentes en la tabla credit\_card

```
81
82 * SELECT pan FROM credit_card;
83 * alter table credit_card drop column pan;
84 * show columns from credit_card;
85
86 -- drop table credit_card
87 -- corro codigo sin foreign key
```

100% 30:84

Result Grid Filter Rows: Search Export:

	Field	Type	Null	Key	Default	Extra	
▶	id	varchar(15)	NO	PRI	NULL		
	iban	varchar(150)	YES		NULL		
	pan	varchar(150)	YES		NULL		
	pin	varchar(100)	YES		NULL		
	cvv	varchar(100)	YES		NULL		
	expiring_date	varchar(100)	YES		NULL		

Result 16

Elimino el campo 'pan' y vuelvo a verificar las tablas para confirmar el cambio.

```
82 * SELECT pan FROM credit_card;
83 * alter table credit_card drop column pan;
84 * show columns from credit_card;
```

100% 20:84

Result Grid Filter Rows: Search Export:

	Field	Type	Null	Key	Default	Extra	
▶	id	varchar(15)	NO	PRI	NULL		
	iban	varchar(150)	YES		NULL		
	pin	varchar(100)	YES		NULL		
	cvv	varchar(100)	YES		NULL		
	expiring_date	varchar(100)	YES		NULL		



## Nivel 2

Ejercicio 1

Ejercicio 2

Ejercicio 3

## Nivel 3

Ejercicio 1

Ejercicio 2