

Sprint3. Sara Gutierrez Amigo

- Exercici 1

La teva tasca és dissenyar i crear una taula anomenada "credit_card" que emmagatzemi detalls crucials sobre les targetes de crèdit. La nova taula ha de ser capaç d'identificar de manera única cada targeta i establir una relació adequada amb les altres dues taules ("transaction" i "company"). Després de crear la taula serà necessari que ingressis la informació del document denominat "dades_introduir_credit". Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.

```
1 • CREATE DATABASE Sprint4;
```

Sabemos mediante la información que nos han dado en “Dades_introduir_credit”, que las columnas que tenemos que crear en esta tabla nueva son las de: id, iban, pan, pin, cvv y expiring_date. Por tanto, el primer paso será crear y dar nombre a la tabla y estos conceptos, indicando que es en el Schema Transactions en el que vamos a estar trabando:

```
2 #Ejercicio 1
3 • USE transactions;
4 • CREATE TABLE credit_card(id VARCHAR (100),
5     iban VARCHAR (100),
6     pan VARCHAR (100),
7     pin VARCHAR (100),
8     cvv VARCHAR (100),
9     expiring_date VARCHAR (100));
```

Para no complicarnos y empezar a escoger diferentes tipos de datos (numéricos, texto o fecha) hemos asignado que todos los datos sean de tipo VARCHAR porque es el más comúnmente utilizado y que contenga 100 caracteres.

CORRECCIÓN: Si hubiéramos querido ser más específicos podíamos haber dicho que el tipo de dato en *Expiring_date* sea *DATE*, el *CVV* sea *CHAR(3)* porque vemos en el archivo de inserción de datos que se trata de 3 números fijos. *Pin* que sea *CHAR(4)* porque también se trata de 4 números fijos. Es decir, tenemos que ser precisos en el tipo de dato para evitar problemas de formato.

Una vez creados los campos copiamos y pegamos el archivo de “Dades_introduir_credit” y lo ejecutamos. Con el comando *INSERT INTO* estamos asignando los valores que queremos a cada uno de los campos.

Sprint3. Sara Gutierrez Amigo

```
2 • Use transactions;
3 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2938', '
4 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2945', '
5 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2952', '
6 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2959', '
7 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2966', '
8 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2973', '
9 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2980', '
10 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2987', '
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2994', '
12 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3001', '
13 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3008', '
14 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3015', '
15 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3022', '
16 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3029', '
17 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-3036', '
```

Y la tabla que nos queda es esta:

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	08/24/23
CcU-2952	BG45IVQL52710525608255	4556 453 55 5287	4598	438	06/29/21
CcU-2959	CR7242477244335841535	372461377349375	3583	667	02/24/23
CcU-2966	BG72LKTQ15627628377363	448566 886747 7265	4900	130	10/29/24
CcU-2973	PT87806228135092429456346	544 58654 54343 384	8760	887	01/30/25
CcU-2980	DE39241881883086277136	402400 7145845969	5075	596	07/24/22
CcU-2987	GE89681434837748781813	3763 747687 76666	2298	797	10/31/23
CcU-2994	BH62714428368066765294	344283273252593	7545	595	02/28/22

874 13:08:46 select * FROM credit_card 275 row(s) returned 0.000 sec / 0.000 sec

Tenemos ahora que establecer lo que va a ser nuestra Primary Key. Sabemos que esa Primary Key no se puede repetir y tampoco puede tener un valor nulo. Para hacer eso ejecutamos la siguiente función con el comando ALTER TABLE que nos sirve para modificar valores que ya han sido establecidos o creados con anterioridad:

```
11 • ALTER TABLE credit_card
12   MODIFY COLUMN id VARCHAR(100) NOT NULL UNIQUE;
```

Y luego le indicamos que sea nuestra Primary Key:

```
14 • ALTER TABLE credit_card
15   ADD PRIMARY KEY (id);
```

CORRECCIÓN: En lugar de establecer la Primary Key paso por paso después de crear la tabla la podríamos haber creado directamente en este primer paso. Cuando establecemos el tipo de dato al crear ese campo, le decimos que sea nuestra Primary Key.

Sprint3. Sara Gutierrez Amigo

Las características de las Primary Keys y las Foreign Keys de las diferentes tablas, para que se puedan establecer las relaciones entre ellas de manera correcta, tienen que tener las mismas características. Nos aseguramos modificando de la tabla de Transaction, la foreign Key de la siguiente manera:

```
17 • ALTER TABLE transaction
18     MODIFY COLUMN credit_card_id VARCHAR(100);
```

CORRECCIÓN: Lo correcto hubiera sido mirar primero que tipo de dato tenemos en la tabla de transacciones, porque es la de hechos, y poner el mismo tipo de dato para nuestra primary key de la tabla de credit_card que estamos creando. No modificar un dato de la tabla de hechos.

COMO SE NOS HUBIERA QUEDADO LA TABLA si la hubiéramos creado considerando las CORRECCIONES pertinentes:

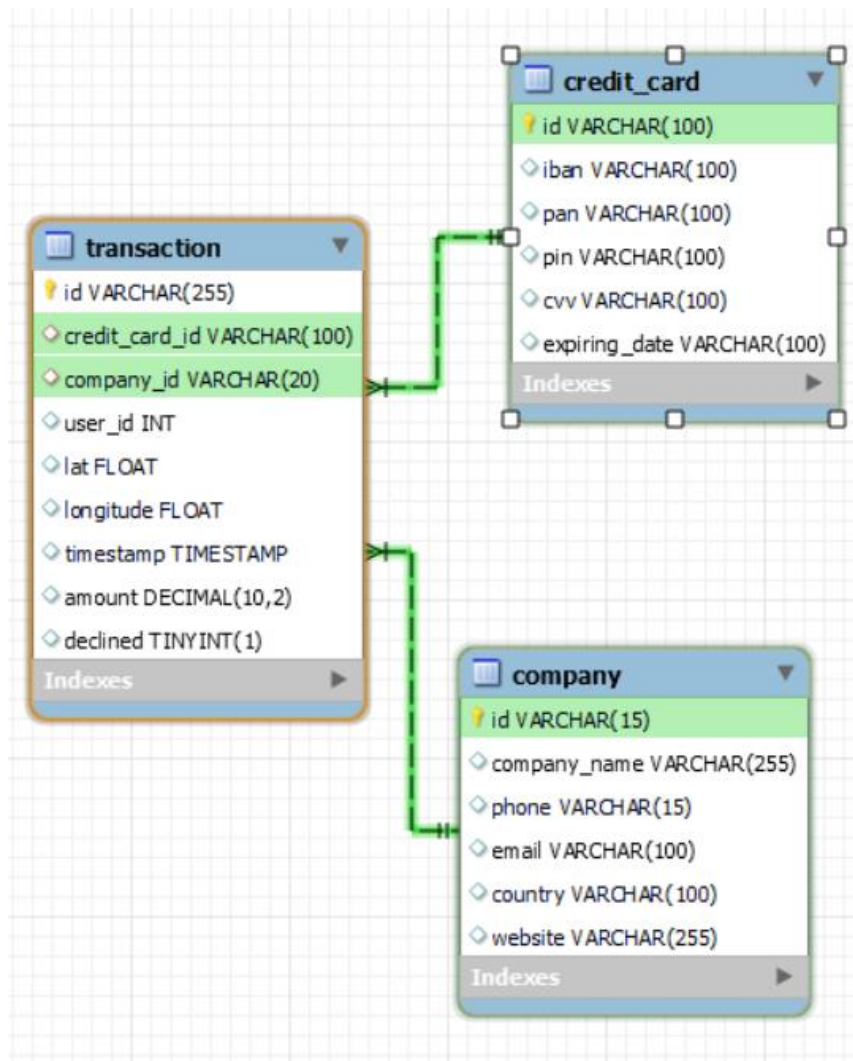
```
5 • CREATE TABLE credit_card(id VARCHAR (100) PRIMARY KEY,
6     iban VARCHAR (100),
7     pan VARCHAR (50),
8     pin CHAR (4),
9     cvv CHAR (3),
10    expiring_date DATE);
```

Ahora solo nos queda establecer la relación entre estas dos tablas de 1 a muchos con el siguiente query:

```
20 • ALTER TABLE transaction
21     ADD CONSTRAINT fk_transaction_credit_card
22     FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
```

El modelo que se nos crea con estas indicaciones dadas es el siguiente:

Sprint3. Sara Gutierrez Amigo



- Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: R323456312213576817699999. Recorda mostrar que el canvi es va realitzar.

Comprobamos primero que el dato que queremos modificar sea diferente a lo que vamos a lo que queremos cambiar con la siguiente query:

```
1 • SELECT * FROM credit_card
2 WHERE id = "CcU-2938";
```

Sprint3. Sara Gutierrez Amigo

*NOTA: No me acordé de hacer una captura al resultado anterior a cambiarlo, pero la comprobación era correcta, había un número de IBAN diferente.

24 #Ejercicio 2

25 • `UPDATE credit_card SET iban = "R323456312213576817699999" WHERE Id = "CcU-2938";`

Y comprobamos:

29 • `SELECT * FROM credit_card WHERE id = "CcU-2938";`

	id	iban	pin	cvv	expiring_date
►	CcU-2938	R323456312213576817699999	3257	984	10/30/22

Cuando queremos actualizar un valor en concreto utilizamos el comando UPDATE. Indicamos primero mediante SET el campo que queremos modificar y el valor nuevo que queremos asignar y luego le decimos para que dato exactamente queremos que se modifique, en nuestro caso le decimos que sea para el ID Ccu 2938.

- Exercici 3

En la taula "transaction" ingressa un nou usuari amb la següent informació:

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

36 • `INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)`

37 `VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", 9999, 829.999, -117.999, 111.11, 0);`

Sprint3. Sara Gutierrez Amigo

Utilizamos los comandos INSERT INTO para seleccionar primero la tabla y los campos que queremos insertar nuevos, después añadimos cada uno de los valores que queremos añadir. Al hacer este paso primero nos salió el siguiente error:

```
"Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails
(`transactions`.`transaction`, CONSTRAINT `fk_transaction_credit_card` FOREIGN KEY
(`credit_card_id`) REFERENCES `credit_card` (`id`))"
```

Y lo mismo con la tabla de Company:

```
"Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails
(`transactions`.`transaction`, CONSTRAINT `fk_transaction_credit_card` FOREIGN KEY
(`credit_card_id`) REFERENCES `credit_card` (`id`))"
```

Explicación:

La tabla de hechos Transaction, como hemos explicado antes está relacionada de muchos a 1 con la tabla Company y Credit_card. Si añadimos una foreign key a la tabla transaction tiene que estar incluida sí o sí en las dos tablas de dimensiones con las que está relacionada. Las Primary Keys de las tablas Credit_card y Company tienen que existir sí o sí para que puedan aparecer en la tabla de hechos.

La solución:

Creamos primero esos dos valores en cada una de las tablas de dimensiones:

```
29 #También tenemos que crear este nuevo valor de ID en el resto de las tablas por las que está relacionado:
30 • INSERT INTO company(id)
31   VALUES ("b-9999");
32
33 • INSERT INTO credit_card(id)
34   VALUES ("Ccu-9999");
35
36 • INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
37   VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD", "Ccu-9999", "b-9999", 9999, 829.999, -117.999, 111.11, 0);
```

Una vez tenemos creados esos dos valores en las tablas de dimensión podemos ejecutar el query con el nuevo usuario en la tabla Transaction.

```
1 • SELECT * FROM transaction
2   WHERE user_id = 9999;
```

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	108B1D1D-5B23-A76C-55EF-C568E49A99DD	Ccu-9999	b-9999	9999	829.999	-117.999	HULL	111.11	0

Con la comprobación vemos que está todo correcto.

- Exercici 4

Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_*card. Recordar mostrar el canvi realitzat.

Sprint3. Sara Gutierrez Amigo

39 #Ejercicio 4



40 • ALTER TABLE credit_card DROP COLUMN pan;

41




42 • SELECT * FROM credit_card;

880 13:23:35 select * FROM credit_card 276 row(s) returned 0.000 sec / 0.000 sec



Result Grid

  Filter Rows:

Edit:

Export/Import:

Wrap C

	id	iban	pin	cvv	expiring_date
▶	CcU-2938	R323456312213576817699999	3257	984	10/30/22
	CcU-2945	DO26854763748537475216568689	9080	887	08/24/23
	CcU-2952	BG45IVQL52710525608255	4598	438	06/29/21
	CcU-2959	CR7242477244335841535	3583	667	02/24/23
	CcU-2966	BG72LKTQ15627628377363	4900	130	10/29/24
	CcU-2973	PT87806228135092429456346	8760	887	01/30/25
	CcU-2980	DE39241881883086277136	5075	596	07/24/22
	CcU-2987	GE89681434837748781813	2298	797	10/31/23
	CcU-2994	BH62714428368066765294	7545	595	02/28/22
	CcU-3001	CY49087426654774581266832110	9562	867	09/16/22
	CcU-3008	LU507216693616119230	1856	740	04/05/25

El comando para modificar una tabla, como ya hemos visto es ALTER TABLE, en este caso lo que tenemos que hacer es eliminar una de las columnas que estaban creadas. Lo hacemos mediante el comando DROP COLUMN. En la comprobación vemos como la columna Pan ya no aparece.

*NOTA: En el ejercicio 1 hay una captura de pantalla de la tabla original donde se ve que estaba la columna PAN.