

SPRINT 3- NIVEL 1

Ejercicio 1.1

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.

-- Primero creamos la tabla credit card en el schema transactions con la siguiente informacion (id, iban, pan, pin, cvv, expiring_date)

```
CREATE DATABASE IF NOT EXISTS transactions;
```

```
USE transactions;
```

```
Create table if not exists credit_card (
```

```
id VARCHAR(15) PRIMARY KEY,
```

```
iban VARCHAR(50),
```

```
pan VARCHAR(50),
```

```
pin VARCHAR(4),
```

```
cvv VARCHAR(3),
```

```
expiring_date VARCHAR(20));
```

The screenshot shows a database management tool interface. On the left, the 'Navigator' pane displays the 'SCHEMAS' section with a tree view showing 'transactions' and its sub-items: 'company', 'credit_card', 'transaction', 'Views', 'Stored Procedures', and 'Functions'. The 'Information' pane below it shows 'No object selected'. The main editor pane displays the following SQL script:

```
1  /* SPRINT 3 - NIVEL 1
2  Ejercicio 1.1
3  Tu tarea es diseñar y crear una tabla llamada "credit_card" que almacene detalles cruciales sobre las tarjetas de crédito.
4  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").
5  Después de crear la tabla será necesario que ingreses la información del documento denominado "datos_introducir_credit".
6  Recuerda mostrar el diagrama y realizar una breve descripción del mismo. */
7
8  -- Primero creamos la tabla credit card en el schema transactions con la siguiente informacion (id, iban, pan, pin, cvv, expiring_date)
9
10 CREATE DATABASE IF NOT EXISTS transactions;
11 USE transactions;
12 Create table if not exists credit_card (
13 id VARCHAR(15) PRIMARY KEY,
14 iban VARCHAR(50),
15 pan VARCHAR(50),
16 pin VARCHAR(4),
17 cvv VARCHAR(3),
18 expiring_date VARCHAR(20));
19
```

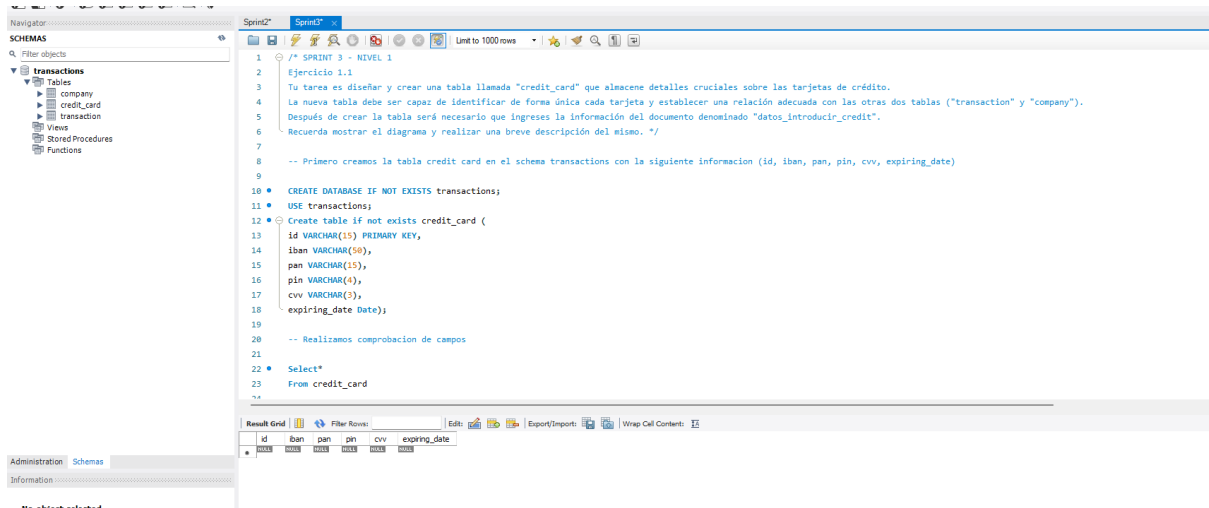
The 'Output' pane at the bottom shows the execution results:

Action	Message	Duration / Fetch
60 12:51:46 USE transactions	0 rows(s) affected	0.000 sec
61 12:51:46 Create table if not exists credit_card (id VARCHAR(15) PRIMARY KEY, iban VARCHAR(50), pan VARCHAR(50), pin VARCHAR(4), cvv VARCHAR(3), expiring_date VARCHAR(20))	0 rows(s) affected	0.016 sec

-- Realizamos comprobacion de campos

Select*

From credit_card



-- Insertamos la informacion del documento denominado "datos_introducir_credit" * (código completo en archivo sql)

```
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CCU-2938',  
'TR301950312213576817638661', '5424465566813633', '3257', '984', '10/30/22');
```

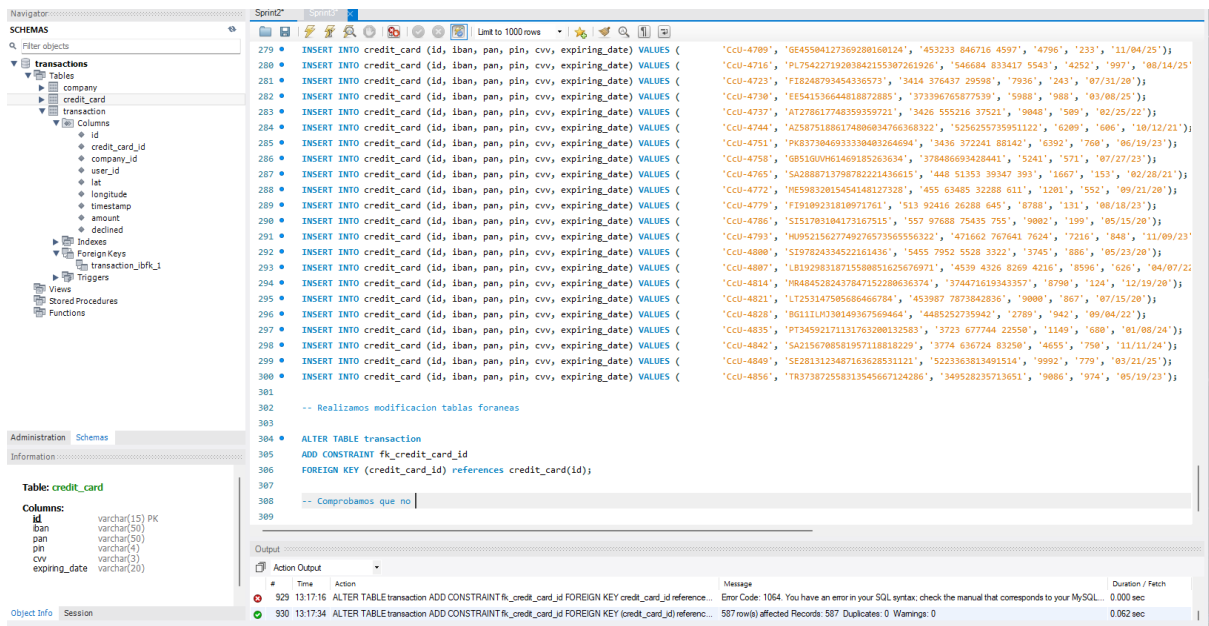
.....

-- Realizamos modificación de claves foráneas

ALTER TABLE transaction

ADD CONSTRAINT fk_credit_card_id

FOREIGN KEY (credit_card_id) references credit_card(id);

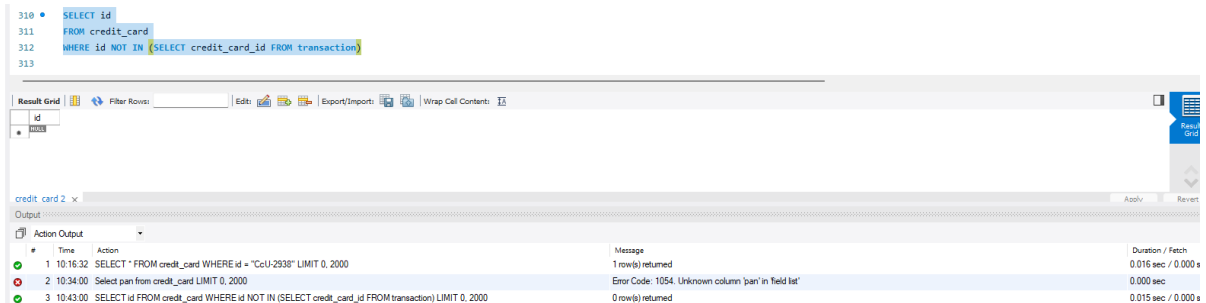


-- Comprobamos que no haya id's/credit card id que no estén en ambas tablas

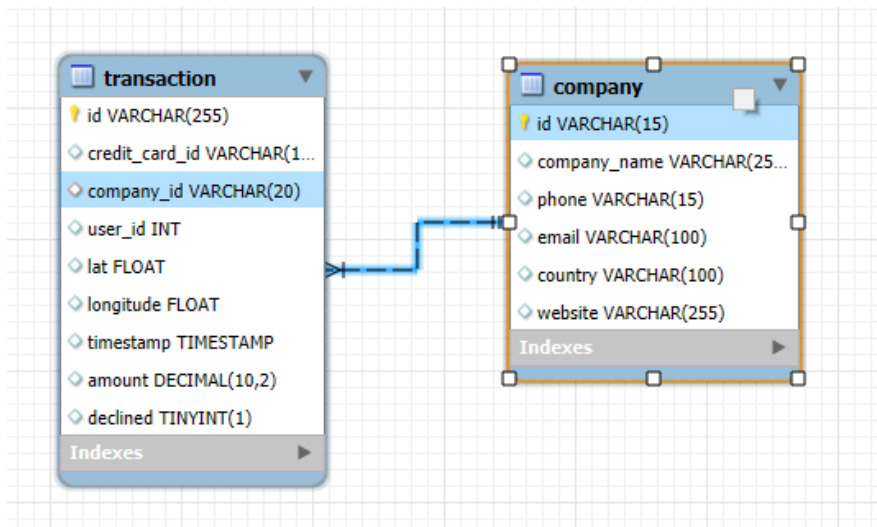
SELECT id

FROM credit_card

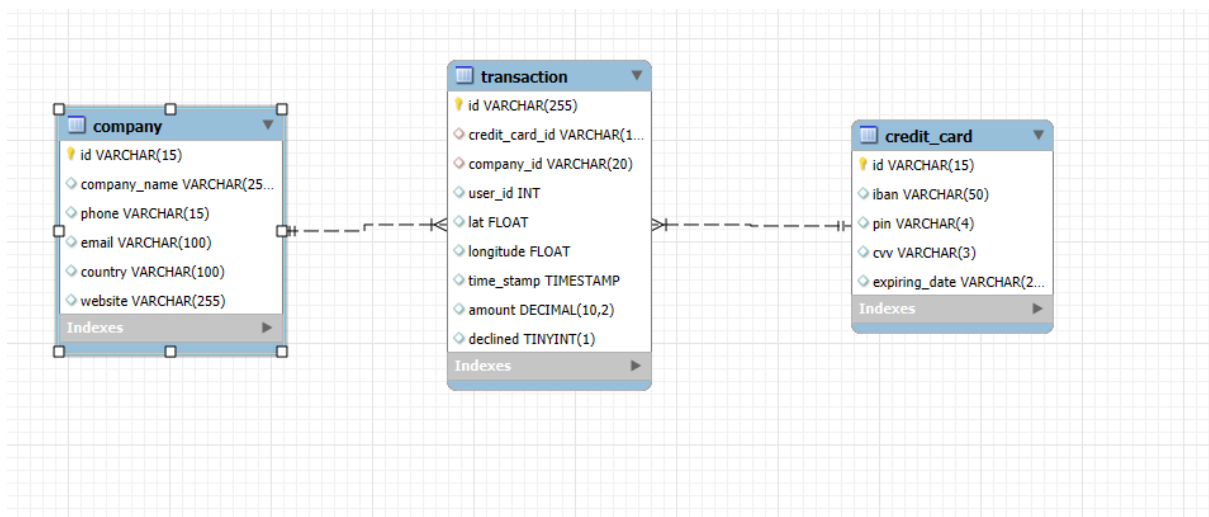
WHERE id NOT IN (SELECT credit_card_id FROM transaction)



Una vez realizado estos cambios la estructura de la database ha cambiado desde el Sprint 2, de esto:



A esto:



En este diagrama se puede observar que la base de datos esta compuesta por 3 tablas diversas, relacionadas entre si

Asi mismo, podemos ver que cada tabla tiene una Primary key, que a su vez están relacionadas con las claves foráneas en la tabla transaction:

Company_id es una clave foránea y está relacionada con company.id

Credit_card_id es una clave foránea y esta relacionada directamente con credit.card_id

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.

-- Visualizamos valor actual.

```
SELECT *
```

```
FROM credit_card
```

```
WHERE id = "CcU-2938"
```

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane shows a tree view of the database structure, including tables like 'company', 'credit_card', and 'transaction'. The 'credit_card' table is selected, and its columns are listed: id, iban, pan, pin, cvv, and expiring_date. The main pane displays a SQL script with the following queries:

```
ALTER TABLE transaction
ADD CONSTRAINT fk_credit_card_id
FOREIGN KEY (credit_card_id) references credit_card(id);

-- Comprobamos que no haya id's/credit card id que no esten en ambas tablas

SELECT id
FROM credit_card
WHERE id NOT IN (SELECT credit_card_id FROM transaction)

/* 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. */

-- Visualizamos valor actual.

SELECT *
FROM credit_card
WHERE id = "CcU-2938"

-- Realizamos el cambio
```

The 'Result Grid' shows the results of the query, with columns: id, iban, pan, pin, cvv, expiring_date. The results are:

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22

The 'Output' pane shows the execution of the query, with a message: 'Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL...'. The 'Action Output' pane shows the results of the query, with a message: '1 row(s) returned'.

-- Realizamos el cambio

```
UPDATE credit_card
```

```
SET iban = "R323456312213576817699999"
```

```
WHERE id = "CcU-2938";
```

The screenshot shows the same database management tool interface. The SQL script now includes the update query:

```
UPDATE credit_card
SET iban = 'R323456312213576817699999'
WHERE id = 'CcU-2938';
```

The 'Output' pane shows the execution of the query, with a message: 'Error Code: 1146. Table transactions.credit_card doesn't exist'. The 'Action Output' pane shows the results of the query, with a message: '1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0'.

-- Comprobamos cambio

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

The screenshot shows a database management interface with a left sidebar containing a tree view of database objects (transaction, credit_card, user, etc.). The main area displays a SQL script with the following content:

```
-- Visualizamos valor actual...  
SELECT *  
FROM credit_card  
WHERE id = "CcU-2938"  
  
-- Realizamos el cambio  
UPDATE credit_card  
SET iban = 'R323456312213576817699999'  
WHERE id = 'CcU-2938';  
  
-- Comprobamos cambio  
SELECT *  
FROM credit_card  
WHERE id = "CcU-2938"
```

Below the script, the 'Result Grid' shows the output of the SELECT queries. The first result set contains one row with the following data:

id	iban	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	5424465566813633	3257 984	10/20/22

The second result set is empty. The 'Action Output' section at the bottom shows the execution status of the queries:

#	Time	Action	Message	Duration / Fetch
937	13:35:47	SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
938	13:35:51	SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

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
lato	829.999
longitud	-117.999
amunt	111.11
declined	0

-- Primero comprobaremos las columnas en la tabla transaction para ver que no haya discrepancias y realizaremos las queries correspondientes para ver que esta transacción ha sido registrada anteriormente:

Sprint2* Sprint3* transactions.transaction >									
Info Columns Indexes Triggers Foreign keys Partitions Grants DDL									
Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra	Comments	
amount	decimal(10,2)		YES			select,insert,update,references			
company_id	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references			
credit_card_id	varchar(15)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references			
declined	tinyint(1)		YES			select,insert,update,references			
id	varchar(255)		NO	utf8mb4	utf8mb4_0900_...	select,insert,update,references			
lat	float		YES			select,insert,update,references			
longitude	float		YES			select,insert,update,references			
timestamp	timestamp		YES			select,insert,update,references			
user_id	int		YES			select,insert,update,references			

Select *

From transaction

Where id = "108B1D1D-5B23-A76C-55EF-C568E49A99DD"

Select *

From transaction

Where company_id = "b-9999"

Select*

From company

Where id = "b-9999"

Select*

From transaction

Where user_id = "9999"

Select *

From credit_card

Where id = "CcU-9999"

Tras realizar esta query y ver la estructura de la tabla , podemos observar que hay discrepancias en el nombre de las columnas y falta informacion para poder añadir, como el timestamp.

Por otro lado, se puede ver que esta compañía no esta registrada tampoco en la tabla de company, por lo que tendremos que crear también un registro con los datos de la compañía y con los datos de credit card, ya que las tablas company, credit card y transaction tiene relación entre ellas (Clave primaria y foránea)

```
394
395 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, time_stamp, amount, declined ) VALUES ('10881D1D-5623-A76C-55EF-C568E49A990D', 'CCU-9999', 'b-9999', '9999', '829.999', '-117.999',
396
397 --
398
399
400
```

Output

#	Time	Action	Message	Duration / Fetch
8	13:15:31	Select * From credit_card Where id = "CCU-9999" LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
9	13:25:45	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, time_stamp, amount, declined) VALUES ('10881D1D-5...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (transactions.transaction, CONSTRAINT transaction_j...	0.000 sec

En un entorno de negocio, esta solicitud de añadir información a la base de datos se rechazaría debido y se solicitaría la información adicional para garantizar la integridad de la información.

A efectos de este ejercicio, supondremos que nos han facilitado la información de la compañía b-9999 y se ha facilitado la información , corrigiendo las tablas y obteniendo el Timestamp a día de hoy

-- Conseguir Timestamp

SELECT NOW();

SELECT CURRENT_TIMESTAMP();

-- 2025-03-05 13:57:27


```

365 -- Conseguir Timestamp
366
367 SELECT NOW();
368 • SELECT CURRENT_TIMESTAMP();
369
370 -- 2025-03-05 13:57:27
371
372 • INSERT INTO transactions (id, credit_card_id, company_id, user_id, lat, longitude, expiring_date) VALUES (
373

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [f1](#)

CURRENT_TIMESTAMP()
2025-03-05 13:57:27

Result 11 | Result 12 x

Output

Action Output

#	Time	Action	Message
944	13:57:27	SELECT NOW() LIMIT 0, 1000	1 row(s) returned
945	13:57:27	SELECT CURRENT_TIMESTAMP() LIMIT 0, 1000	1 row(s) returned

-- Mejoramos la nomenclatura de las tablas para facilitar la introducción de datos timestamp to time_stamp

Alter table transaction

Rename column timestamp to time_stamp;

-- Añadimos la compañía - asumimos que nos han facilitado todos los datos de la compañía b-9999

```

INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-
9999', 'Nueva compañía', '00 87 56 52 33', 'juanito.nuevacompañia@ohno.net', 'Germany',
'https://otranuevacompañia/site');

```

-- y comprobamos que se ha añadido

Select *

From transaction

Where company_id = "b-9999"

```

388 * Alter table transaction
389 Rename column timestamp to time_stamp;
390
391 -- Añadimos la compañía - asumimos que nos han facilitado todos los datos de la compañía b-9999
392
393 * INSERT INTO company (id, company_name, phone, email, country, website) VALUES ( 'b-9999', 'Nueva compañía', '00 87 56 52 33', 'juanito.nuevacompañia@ohmo.net', 'Germany', 'https://otranuevacompañia/site');
394
395 -- y comprobamos que se ha añadido
396
397 * Select *
398 From transaction
399 Where company_id = "b-9999"
400

```

id	company_name	phone	email	country	website
b-9999	Nueva compañía	00 87 56 52 33	juanito.nuevacompañia@ohmo.net	Germany	https://otranuevacompañia/site

company 4 x

Output

#	Time	Action	Message	Duration / Fetch
10	13.26.21	INSERT INTO company (id, company_name, phone, email, country, website) VALUES ('b-9999', 'Nueva compañía', '00 87 56 52 33', 'jua...	1 row(s) affected	0.000 sec
11	13.26.36	Select * From company Where id = 'b-9999' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

-- Añadimos la credit card id - asumimos que nos han facilitado todos los datos de la credit "CcU-9999"

```
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-9999', 'ES999820559993545766124999', '999528235713999', '9999', '999', '05/02/29')
```

-- y comprobamos que se ha añadido

```
Select *
```

```
From credit_card
```

```
Where id = "CcU-9999"
```

```

401 -- Añadimos la credit card id - asumimos que nos han facilitado todos los datos de la credit "CcU-9999"
402 * INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-9999', 'ES999820559993545766124999', '999528235713999', '9999', '999', '05/02/29')
403
404
405 -- y comprobamos que se ha añadido
406
407 * Select *
408 From credit_card
409 Where id = "CcU-9999"
410

```

id	iban	pan	pin	cvv	expiring_date
CcU-9999	ES999820559993545766124999	999528235713999	9999	999	05/02/29

credit_card 5 x

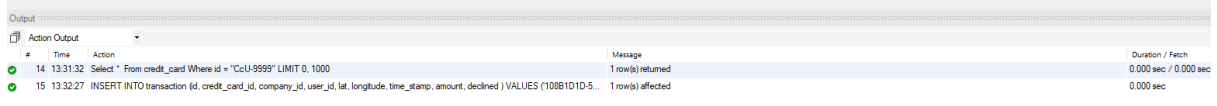
Output

#	Time	Action	Message	Duration / Fetch
13	13.31.12	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-9999', 'ES999820559993545766124999', '9995282357...	1 row(s) affected	0.000 sec
14	13.31.32	Select * From credit_card Where id = "CcU-9999" LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

-- Ahora podemos añadir la línea a transacción

```
INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, time_stamp,
amount, declined ) VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999',
'9999', '829.999', '-117.999', '2025-03-05 13:57:27', '111.11', '0')
```

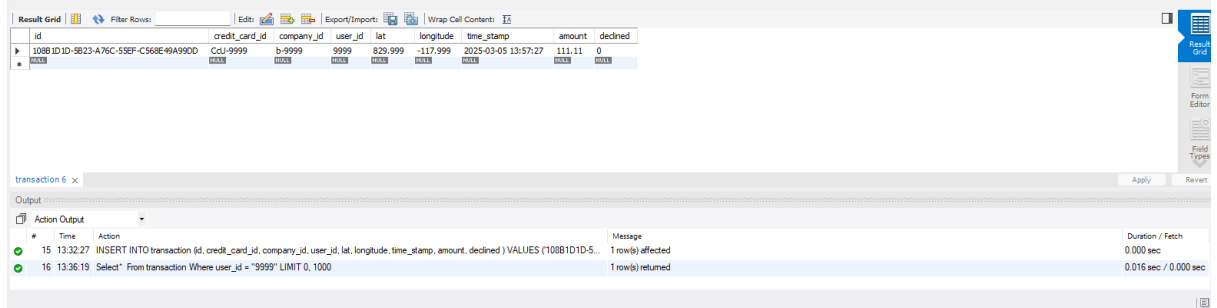
```
411 -- Ahora podemos añadir la línea a transaction
412
413 INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, time_stamp, amount, declined ) VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999',
414
415 --
416
417
418
```



-- comprobamos que se ha añadido

```
Select*
From transaction
Where user_id = "9999"
```

```
413 INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, time_stamp, amount, declined ) VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999',
414
415 -- comprobamos que se ha añadido
416
```



Ejercicio 1.4

Desde cursos humanos te solicitan eliminar la columna "pan" de la tabla credit_card.
Recuerda mostrar el cambio realizado.*/

-- comprobar que existe la columna pan

```
Select pan from credit_card
```

```

425 -- comprobar que existe la columna pan
426
427 Select pan from credit_card
428

```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

pan
54244655668 13633
514242821948828
4556 453 55 5287
372461377349375
448566 886747 7265
544 58654 54343 384
402400 7149545969
3763747687 76666
344283273252993
511722 924833 2244
4485744464433884
3784 662233 17389
5164 1379 4842 2951
3429 279566 77631
3768 451556 48766
455676 6437463635
4024007123722
3484 621767 21237
3467 732741 26810
3464 789562 23352
4539 322 74 2377
5266 3346 1135 1687
3488 754223 46253
448 55418 98863 789
549547146395283
341834822877471
527646 533375 6577
4716 443 46 4368
4146 3441 9766 7148

credit_card 1 x

Output

#	Time	Action	Message	Duration / Fetch
1	11:57:48	Select pan from om credit_card LIMIT 0, 2000	Error Code: 1146. Table 'transactions.om' doesn't exist	0.000 sec
2	11:57:58	Select pan from credit_card LIMIT 0, 2000	276 row(s) returned	0.000 sec / 0.000 sec

-- Eliminamos la columna pan de credit card

```

426
427 Select pan from credit_card
428
429 -- Eliminamos la columna pan de credit card
430
431 ALTER TABLE credit_card
432 DROP COLUMN pan
433
434 -- comprobar que existe la columna pan
435
436 select pan from credit_card
437

```

Output

#	Time	Action	Message	Duration / Fetch
2	11:57:58	Select pan from credit_card LIMIT 0, 2000	276 row(s) returned	0.000 sec / 0.000 sec
3	12:00:24	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
4	12:00:59	Select pan from credit_card LIMIT 0, 2000	Error Code: 1054. Unknown column 'pan' in field list	0.000 sec

ALTER TABLE credit_card

DROP COLUMN pan

```

431 ALTER TABLE credit_card
432 DROP COLUMN pan
433
434 /* SPRINT 3 - NIVEL 2
435
436 Ejercicio 2.1
437 Elimina de la tabla transacciones el resultado con ID 27632015, 6094 1850, 8456, 88709603893 de la base de datos?

```

Output

#	Time	Action	Message	Duration / Fetch
1	11:57:48	Select pan from om credit_card LIMIT 0, 2000	Error Code: 1146. Table 'transactions.om' doesn't exist	0.000 sec
2	11:57:58	Select pan from credit_card LIMIT 0, 2000	276 row(s) returned	0.000 sec / 0.000 s
3	12:00:24	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec

-- comprobar que existe la columna pan

Select pan from credit_card

426	
427	<code>Select pan from credit_card</code>
428	
429	<code>-- Eliminamos la columna pan de credit card</code>
430	
431	<code>ALTER TABLE credit_card</code>
432	<code>DROP COLUMN pan</code>
433	
434	<code>-- comprobar que existe la columna pan</code>
435	
436	<code>select pan from credit_card</code>
437	

#	Time	Action	Message	Duration / Fetch
2	11:57:58	Select pan from credit_card LIMIT 0, 2000	276 row(s) returned	0.000 sec / 0.000 sec
3	12:00:24	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
4	12:00:59	Select pan from credit_card LIMIT 0, 2000	Error Code: 1054. Unknown column 'pan' in 'field list'	0.000 sec

SPRINT 3- NIVEL 2

Ejercicio 2.1

Elimina de la tabla transacción el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos*/

-- Comprobamos el registro

Select *

FROM transaction

WHERE ID = "02C6201E-D90A-1859-B4EE-88D2986D3B02"

421	<code>/* SPRINT 3 - NIVEL 2</code>
422	
423	<code>Ejercicio 2.1 Elimina de la tabla transacción el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos*/</code>
424	
425	<code>-- Comprobamos el registro</code>
426	
427	<code>Select *</code>
428	<code>FROM transaction</code>
429	<code>WHERE ID = "02C6201E-D90A-1859-B4EE-88D2986D3B02"</code>
430	

id	credit_card_id	company_id	user_id	lat	longitude	time_stamp	amount	declined
02C6201E-D90A-1859-B4EE-88D2986D3B02	Ccl-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.92	0

#	Time	Action	Message	Duration / Fetch
18	13:41:57	Select FROM transaction WHERE ID = "02C6201E-D90A-1859-B4EE-88D2986D3B02"	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syn...	0.000 sec
19	13:42:11	Select * FROM transaction WHERE ID = "02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec

-- Eliminamos el registro - podemos utilizar DELETE o DELETE ROW

DELETE FROM transaction

WHERE ID = "02C6201E-D90A-1859-B4EE-88D2986D3B02"

-- y comprobamos que se ha eliminado

```
Select *
```

```
FROM transaction
```

```
WHERE ID = "02C6201E-D90A-1859-B4EE-88D2986D3B02"
```

```
433 DELETE FROM transaction
434 WHERE ID = "02C6201E-D90A-1859-B4EE-88D2986D3B02" ;
435
436 -- y Comprobamos que se ha eliminado
437
438 Select *
439 FROM transaction
440 WHERE ID = "02C6201E-D90A-1859-B4EE-88D2986D3B02"
441
442 - En este caso no hace falta eliminar datos de otras tablas, ya que hay otras dependencias
```

id	credit_card_id	company_id	user_id	lat	longitude	time_stamp	amount	declined
22	134730	0	1	40.683	-89.268	1546308000	1000	0

EJERCICIO 2.2

La sección de marketing desea tener acceso a información específica para realizar análisis y estrategias efectivas.

Se ha solicitado crear una vista que proporcione detalles clave sobre las compañías y sus transacciones.

Será necesaria que crees una vista llamada VistaMarketing que contenga la siguiente información:

Nombre de la compañía. Teléfono de contacto. País de residencia. Media de compra realizado por cada compañía. Presenta la vista creada, ordenando los datos de mayor a menor promedio de compra.

```
Create view VistaMarketing as
```

```
Select company_name as Nombre_compañia, company.phone as Telefono_contacto,
company.country as Pais_residencia , AVG(amount) AS Media_compra
```

```
From Transaction
```

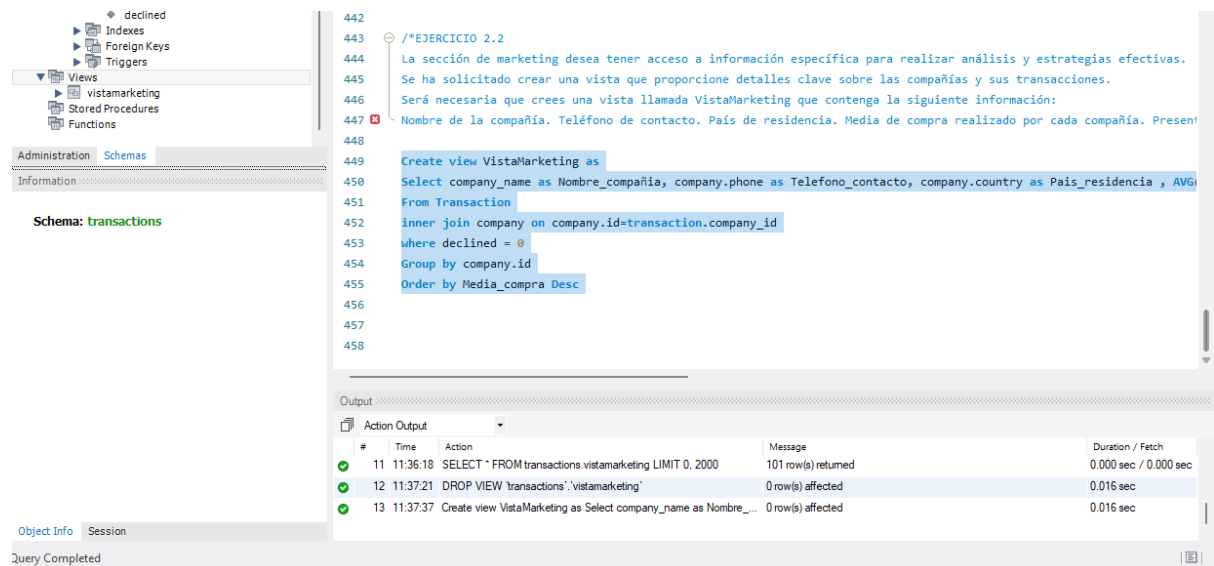
```
inner join company on company.id=transaction.company_id
```

```
where declined = 0
```

```
Group by company.id
```

Order by Media_compra Desc

** Se han eliminado de esta lista las operaciones declinadas

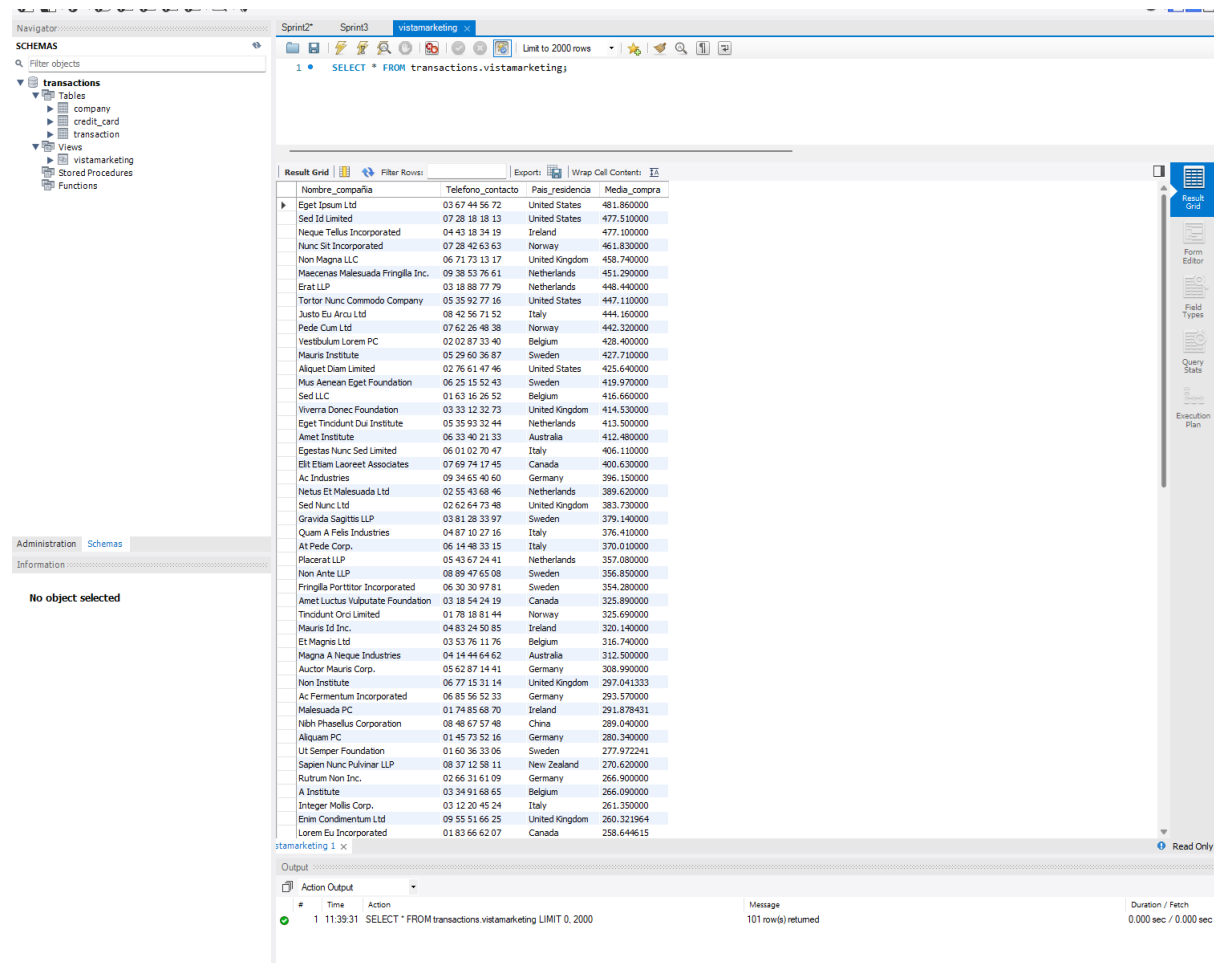


Schema: transactions

```
442
443 /*EJERCICIO 2.2
444 La sección de marketing desea tener acceso a información específica para realizar análisis y estrategias efectivas.
445 Se ha solicitado crear una vista que proporcione detalles clave sobre las compañías y sus transacciones.
446 Será necesaria que crees una vista llamada VistaMarketing que contenga la siguiente información:
447 Nombre de la compañía. Teléfono de contacto. País de residencia. Media de compra realizado por cada compañía. Presente
448
449 Create view VistaMarketing as
450 Select company_name as Nombre_compañía, company.phone as Telefono_contacto, company.country as Pais_residencia , AVG
451 From Transaction
452 inner join company on company.id=transaction.company_id
453 where declined = 0
454 Group by company.id
455 Order by Media_compra Desc
456
457
458
```

Output

#	Time	Action	Message	Duration / Fetch
11	11:36:18	SELECT * FROM transactions.vistamarketing LIMIT 0, 2000	101 row(s) returned	0.000 sec / 0.000 sec
12	11:37:21	DROP VIEW transactions.vistamarketing	0 row(s) affected	0.016 sec
13	11:37:37	Create view VistaMarketing as Select company_name as Nombre_...	0 row(s) affected	0.016 sec



Result Grid

Nombre_compañía	Telefono_contacto	Pais_residencia	Media_compra
Eget Ipsum Ltd	03 67 44 56 72	United States	481.860000
Sed Id Limited	07 28 18 18 13	United States	477.510000
Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.100000
Nunc Sit Incorporated	07 28 42 63 63	Norway	461.830000
Non Magna LLC	06 71 73 13 17	United Kingdom	458.740000
Masconas Malesuada Fringilla Inc.	09 38 53 78 61	Netherlands	451.290000
Erat LLP	03 18 88 77 79	Netherlands	448.440000
Tortor Nunc Commodo Company	05 35 92 77 16	United States	447.110000
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	444.160000
Pede Cum Ltd	07 62 26 48 38	Norway	442.320000
Vestibulum Lorem PC	02 02 87 33 40	Belgium	428.400000
Magna Institute	05 29 60 36 87	Sweden	427.710000
Aliquet Diam Limited	02 76 61 47 46	United States	425.640000
Mus Aenean Eget Foundation	06 25 15 52 43	Sweden	419.970000
Sed LLC	01 63 16 26 52	Belgium	416.660000
Viverra Donec Foundation	03 33 12 32 73	United Kingdom	414.530000
Eget Tincidunt Oui Institute	05 35 93 32 44	Netherlands	413.500000
Amet Institute	06 33 40 21 33	Australia	412.480000
Egetas Nunc Sed Limited	06 01 02 70 47	Italy	406.110000
Elit Elam Laoreet Associates	07 69 74 17 45	Canada	400.630000
Ac Industries	09 34 65 40 60	Germany	396.150000
Netus Et Malesuada Ltd	02 55 43 68 46	Netherlands	389.620000
Sed Nunc Ltd	02 62 64 73 48	United Kingdom	383.730000
Gravida Sagittis LLP	03 81 28 33 97	Sweden	379.140000
Quam A Feis Industries	04 87 10 27 16	Italy	376.410000
Alt Pede Corp.	06 14 48 33 15	Italy	370.010000
Placerat LLP	05 43 67 24 41	Netherlands	357.080000
Non Ante LLP	08 89 47 65 08	Sweden	356.850000
Fringilla Porttitor Incorporated	06 30 30 97 81	Sweden	354.280000
Amet Luctus Vulputate Foundation	03 18 54 24 19	Canada	325.890000
Tincidunt Orci Limited	01 78 18 81 44	Norway	325.690000
Mauris Id Inc.	04 83 24 50 85	Ireland	320.140000
Et Magna Ltd	03 53 76 11 76	Belgium	316.740000
Magna A Neque Industries	04 14 44 64 62	Australia	312.500000
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.990000
Non Institute	06 77 15 31 14	United Kingdom	297.041333
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.570000
Malesuada PC	01 74 85 68 70	Ireland	291.878431
Nibh Phasellus Corporation	08 48 67 57 48	China	289.040000
Aliquam PC	01 45 73 52 16	Germany	280.340000
Ut Semper Foundation	01 60 36 33 06	Sweden	277.972241
Sapient Nunc Pulvinar LLP	08 37 12 58 11	New Zealand	270.620000
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000
A Institute	03 34 91 68 65	Belgium	266.090000
Integer Mollis Corp.	03 12 20 45 24	Italy	261.350000
Ervin Conditimentum Ltd	09 55 51 66 25	United Kingdom	260.321964
Lorem Eu Incorporated	01 83 66 62 07	Canada	258.644615

EJERCICIO 2.3

Filtrar la vista VistaMarketing para mostrar sólo las compañías que tienen su país de residencia en "Germany"

-- utilizando la función de filtro (Filter rows) en MySQL Workbench en la view vistamarketing */

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the 'transactions' database with tables (company, credit_card, transaction) and views (vistamarketing). The 'vistamarketing' view is selected, showing its fields: Nombre_compañia, Telefono_contacto, Pais_residencia, and Media_compra. The main editor shows a SQL query: `SELECT * FROM transactions.vistamarketing;`. Below the query, the 'Result Grid' displays the filtered data for 'Germany'.

Nombre_compañia	Telefono_contacto	Pais_residencia	Media_compra
Ac Industries	09 34 65 40 60	Germany	396.150000
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.990000
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.570000
Aliquam PC	01 45 73 52 16	Germany	280.340000
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000
Nunc Interdum Incorporated	05 18 15 48 13	Germany	242.947692
Nueva compañía	00 87 56 52 33	Germany	111.110000
Convallis In Incorporated	06 66 57 29 50	Germany	60.990000
Augue Foundation	06 88 43 15 63	Germany	15.050000

Utilizando función select

The screenshot shows the MySQL Workbench interface. The main editor displays a SQL query: `SELECT * FROM vistamarketing Where Pais_residencia ="Germany"`. Below the query, the 'Result Grid' displays the filtered data for 'Germany'.

Nombre_compañia	Telefono_contacto	Pais_residencia	Media_compra
Ac Industries	09 34 65 40 60	Germany	396.150000
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.990000
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.570000
Aliquam PC	01 45 73 52 16	Germany	280.340000
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000
Nunc Interdum Incorporated	05 18 15 48 13	Germany	242.947692
Nueva compañía	00 87 56 52 33	Germany	111.110000
Convallis In Incorporated	06 66 57 29 50	Germany	60.990000
Augue Foundation	06 88 43 15 63	Germany	15.050000

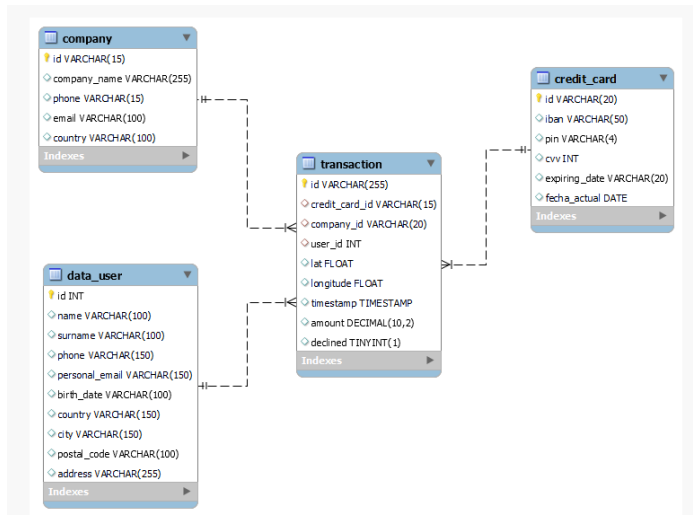
NIVEL3

EJERCICIO 3.1

La próxima semana tendrás una nueva reunión con los gerentes de marketing.

Un compañero de tu equipo realizó modificaciones en la base de datos, pero no recuerda cómo las realizó.

Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama



Revisamos la estructura actual de la base de datos:

Tal y como se ha visto en el Nivel 1 de este sprint , vemos que no existe la tabla data_user en nuestra base de datos, por la que añadiremos a continuación tomando como referencia los campos facilitados por nuestro compañero

Creamos la tabla user (Se ha utilizado el archivo estructura datos_user pero se ha simplificado el código inicial para facilitar la creación de la tabla (e.g. - NO INDEX y no FOREIGN KEY

```
CREATE TABLE IF NOT EXISTS data_user (
```

```
    id INT PRIMARY KEY,  
    name VARCHAR(100),  
    surname VARCHAR(100),  
    phone VARCHAR(150),  
    personal_email VARCHAR(150),  
    birth_date VARCHAR(100),  
    country VARCHAR(150),  
    city VARCHAR(150),  
    postal_code VARCHAR(100),  
    address VARCHAR(255)
```

);

```
497
498 CREATE TABLE IF NOT EXISTS data_user (
499     id INT PRIMARY KEY,
500     name VARCHAR(100),
501     surname VARCHAR(100),
502     phone VARCHAR(150),
503     personal_email VARCHAR(150),
504     birth_date VARCHAR(100),
505     country VARCHAR(150),
506     city VARCHAR(150),
507     postal_code VARCHAR(100),
508     address VARCHAR(255),
509     FOREIGN KEY(id) REFERENCES transaction(user_id)
510 );
511
512
513
514
515
```

Output

#	Time	Action	Message	Duration / Fetch
7	12:11:17	DROP TABLE transactions.`user`	0 row(s) affected	0.016 sec
8	12:11:59	CREATE INDEX idx_user_id ON transaction(user_id)	Error Code: 1061. Duplicate key name 'idx_user_id'	0.000 sec
9	12:12:08	CREATE TABLE IF NOT EXISTS data_user (id INT PRIMARY KEY, name VARCHAR(100), surman...	0 row(s) affected	0.031 sec

Añadimos la información a la tabla utilizando la información del archivo datos_introducir_user. Se han realizado los siguientes cambios en las columnas en el dataset:

User a data_user

Email a personal_email

datos_introducir_user (1)

Find & Replace: Q|-user, Replace: data_user

```
1 SET foreign_key_checks = 0;
2
3 -- Insertamos datos de data_user
4 INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
5 INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
6 INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
7 INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
8 INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
9 INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
10 INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
```

datos_introducir_user (1)

Find & Replace: Q|-email, Replace: personal_email

```
1 SET foreign_key_checks = 0;
2
3 -- Insertamos datos de data_user
4 INSERT INTO data_user (id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address) VALUES (
5 INSERT INTO data_user (id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address) VALUES (
6 INSERT INTO data_user (id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address) VALUES (
7 INSERT INTO data_user (id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address) VALUES (
```

```
INSERT INTO data_user (id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address) VALUES (
    "1", "Zeus", "Gamble", "1-282-581-0551",
    "interdum.enim@protonmail.edu", "Nov 17, 1985",
    "United States", "Lowell", "73544", "348-7818 Sagittis St.");
```

```
INSERT INTO data_user (id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address) VALUES ( "2", "Garrett", "Mcconnell", "(718) 257-2412", "integer.vitae.nibh@protonmail.org", "Aug 23, 1992", "United States", "Des Moines", "59464", "903 Sit Ave");
```

The screenshot shows a database management interface with a left sidebar containing 'Administration', 'Schemas', and 'Information'. The main area displays a list of SQL queries (lines 516 to 556) that insert data into the 'data_user' table. The queries are numbered and include comments like '-- Insertamos datos de data_user'. The output pane at the bottom shows two successful insertions, each affecting 1 row in 0.000 seconds.

-- Añadimos Usuario 9999 a la tabla data_user, que ya habíamos observado que no existía en la base de datos

```
INSERT INTO data_user (id, name, surname, phone, personal_email, birth_date, country, city, postal_code, address) VALUES ( "9999", "Carlos", "carlos", "(999) 563-2771", "upsie@outlook.couk", "Jul 29, 1991", "Germany", "Auburn", "5999", "489-9999 Ave");
```

-- y comprobamos que la informacion se ha subido a la tabla

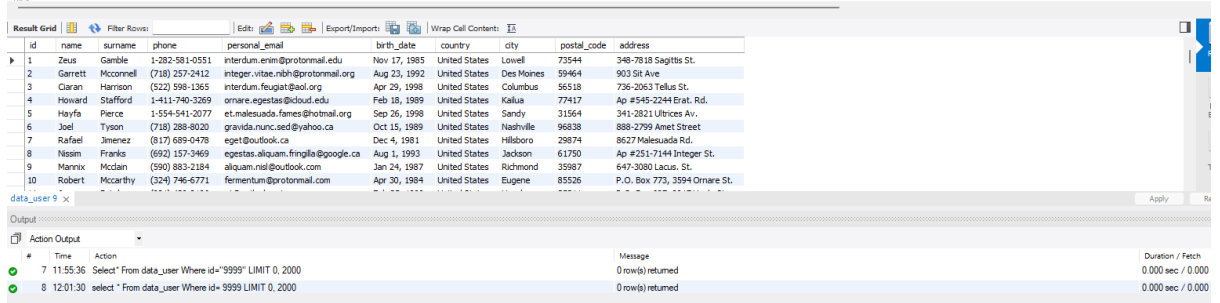
Select *

From data_user


```

798 -- A continuacion tenemos que relacionar esta tabla con la tabla de transaction, en este caso user_id en la tabla de transaction sera una clave foranea que se relacionará con la clave primaria ID en la tabla d
799
800 ALTER TABLE transaction
801 ADD CONSTRAINT fk_user_id
802 FOREIGN KEY (user_id) references data_user(id)
803 SET FOREIGN_KEY_CHECKS = 1
804
805 -- Comprobamos que la informacion es correcta
806
807 Select*
808 From data_user
809
810
811 -- Ahora la informacion esta alineada con los requisitos de nuestro compañero de Marketing, procedemos a realizar varios cambios en las tablas de la base de datos:
812
813 -- En la tabla credit_card crearemos una nueva columna fecha_actual y modificaremos el tipo de datos en la columna cvv de Varchar(3) a INT
814
815

```



-- Ahora que la tabla data_user y su información esta alineada con los requisitos de nuestro compañero de Marketing, procedemos a realizar varios cambios adicionales en las otras tablas de la base de datos:

-- En la tabla credit_card crearemos una nueva columna fecha_actual y modificaremos el tipo de datos en la columna cvv de Varchar(3) a INT

```

797 -- A continuacion tenemos que relacionar esta tabla con la tabla de transaction, en este caso user_id en la tabla de transaction sera una clave foranea que se relacionará con la clave
798
799 ALTER TABLE transaction
800 ADD CONSTRAINT fk_user_id
801 FOREIGN KEY (user_id) references data_user(id)
802
803 -- Ahora la informacion esta alineada con los requisitos de nuestro compañero de Marketing, procedemos a realizar varios cambios en las tablas de la base de datos:
804
805 -- En la tabla credit_card crearemos una nueva columna fecha_actual y modificaremos el tipo de datos en la columna cvv de Varchar(3) a INT
806
807 -- Creamos nueva columna
808
809 ALTER TABLE credit_card
810 ADD Column fecha_actual Date;
811
812 -- Modificamos columna
813
814 ALTER TABLE credit_card
815 Modify column cvv INT
816

```



-- Creamos nueva columna

```
ALTER TABLE credit_card
```

```
ADD Column fecha_actual Date;
```

-- Modificamos características de la columna CVV , de Varchar a INT

ALTER TABLE credit_card

Modify column cvv INT

```
796 -- A continuación tenemos que relacionar esta tabla con la tabla de transaction, en este caso user_id en la tabla de transaction será una clave foránea que se relacionará con la clave
797 ALTER TABLE transaction
798 ADD CONSTRAINT fk_user_id
799 FOREIGN KEY (user_id) references data_user(id)
800
801 -- Ahora la información está alineada con los requisitos de nuestro compañero de Marketing, procedemos a realizar varios cambios en las tablas de la base de datos:
802
803 -- En la tabla credit_card crearemos una nueva columna fecha_actual y modificaremos el tipo de datos en la columna cvv de Varchar(3) a INT
804
805 -- Creamos nueva columna
806
807 ALTER TABLE credit_card
808 ADD Column fecha_actual Date;
809
810 -- Modificamos columna
811
812 ALTER TABLE credit_card
813 Modify column cvv INT;
```

Output

#	Time	Action	Message	Duration / Fetch
586	12:40:32	ALTER TABLE transaction ADD CONSTRAINT fk_user_id FOREIGN KEY (user_id) references data_user(id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
587	12:49:09	ALTER TABLE credit_card ADD Column fecha_actual Date	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
588	12:50:19	ALTER TABLE credit_card Modify column cvv INT	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0	0.047 sec

-- Revisamos que todo correcto

Select *

From credit_card

-- Eliminamos la columna website de la tabla company y después comprobamos que se haya eliminado

Select * website

From company

Alter table company

Drop column website

Select * website

From company

```
817 Select *
818 From credit_card
819
820 -- Eliminamos la columna website de la tabla company despues de comprobar
821
822 Select website
823 From company
824
825 Alter
```

Result Grid

website
https://instagram.com/site
https://whatsapp.com/group/9
https://pinterest.com/sub/cars
https://cnn.com/user/110
https://netfix.com/settings
https://nytimes.com/user/110
https://cnn.com/one
https://netfix.com/sub/cars
https://ebay.com/sub
https://pinterest.com/sub/cars
https://naver.com/site
https://instagram.com/group/9
https://google.com/one
https://reddit.com/fr
https://yahoo.com/sub
https://netfix.com/sub
https://reddit.com/user/110
https://baidu.com/group/9
https://instagram.com/fr
https://yahoo.com/fr
https://wikipedia.org/en-us
https://baidu.com/sub/cars
https://whatsapp.com/site
https://reddit.com/fr
https://ebay.com/settings
https://netfix.com/group/9
https://cnn.com/group/9
https://google.com/fr

company 7 x

Output

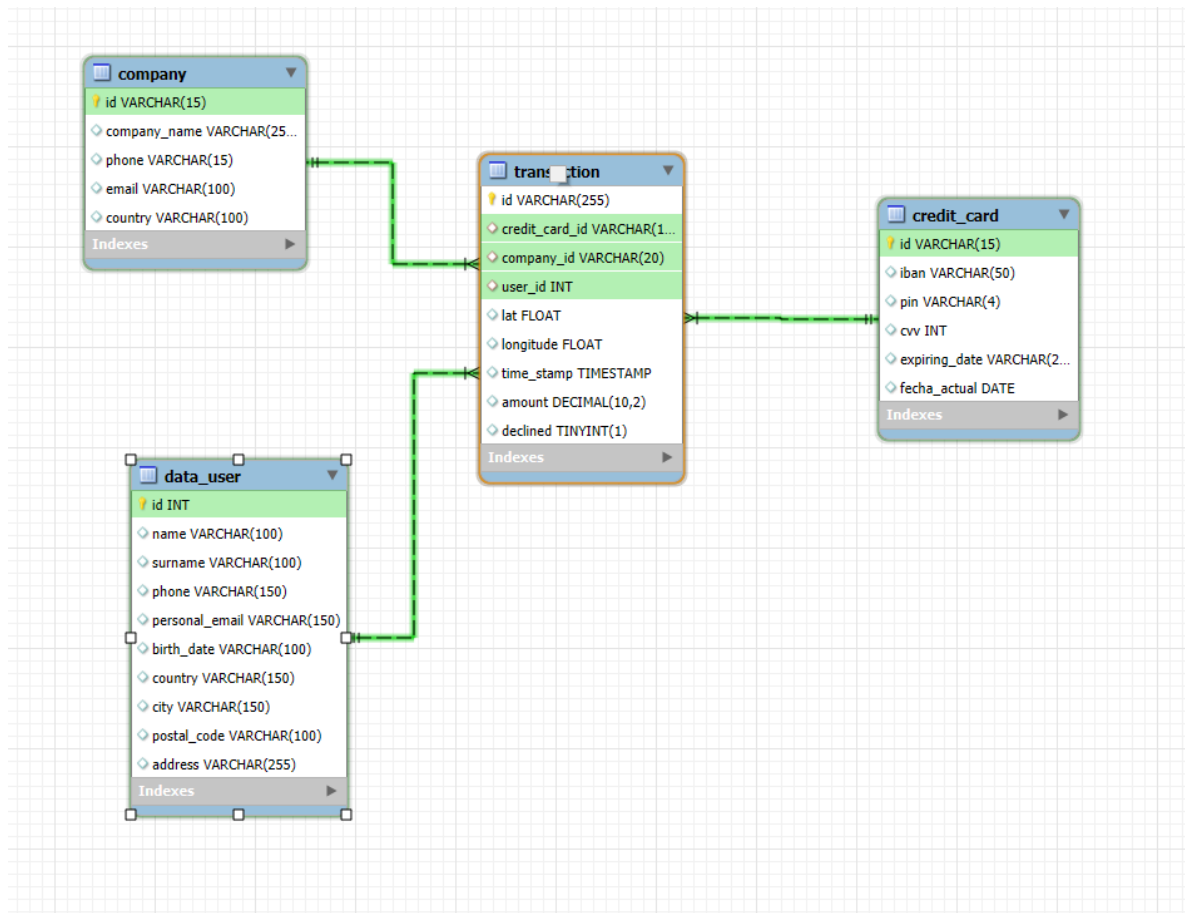
#	Time	Action	Message	Duration / Fetch
598	12:59:50	UPDATE credit_card SET fecha_actual = curdate() Where Id >= 0	Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KE...	0.000 sec
599	13:00:00	UPDATE credit_card SET fecha_actual = curdate() Where Id >= 0	Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a KE...	0.000 sec
600	13:01:41	Select website From company LIMIT 0, 2000	101 row(s) returned	0.000 sec / 0.000 sec

```
820 -- Eliminamos la columna website de la tabla company y despues comprobamos que se haya eliminado
821
822 Select website
823 From company
824
825 Alter table company
826 Drop column website
827
828 Select website
829 From company
830
831
832
833
834
835
836
837
```

Output

#	Time	Action	Message	Duration / Fetch
600	13:01:41	Select website From company LIMIT 0, 2000	101 row(s) returned	0.000 sec / 0.000 sec
601	13:02:43	Alter table company Drop column website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
602	13:03:10	Select website From company LIMIT 0, 2000	Error Code: 1054. Unknown column 'website' in field list	0.000 sec

Una vez realizados todos estos cambios podemos ver como la estructura es la misma que la de nuestro compañero de Marketing



Ejercicio 3 2

La empresa también te solicita crear una vista llamada "InformeTecnico" que contenga la siguiente información:

- ID de la transacción
- Nombre del usuario/a
- Apellido del usuario/a
- IBAN de la tarjeta de crédito usada.
- Nombre de la compañía de la transacción realizada.

Asegúrate de incluir información relevante de ambas tablas y utiliza alias para cambiar de nombre columnas según sea necesario*/

Create view InformeTecnico as

Select transaction.id as ID, data_user.name as Nombre, data_user.surname as Apellido, credit_card.iban as IBAN, company_name as Nombre_compañía

From Transaction

inner join company on company.id=transaction.company_id

inner join data_user on data_user.id=transaction.user_id

inner join credit_card on credit_card.id=transaction.credit_card_id

```
844 Create view InformeTecnico as
845 Select transaction.id as ID, data_user.name as Nombre, data_user.surname as Apellido, credit_card.iban as IBAN, company_name as Nombre_compañia
846 From Transaction
847 inner join company on company.id=transaction.company_id
848 inner join data_user on data_user.id=transaction.user_id
849 inner join credit_card on credit_card.id=transaction.credit_card_id
850
851 /*Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción. */
852
853 -- IMPORTANTE Se ha ordenado en orden descendente utilizando la interfaz grafica de la vista en funcion de la variable ID de transaccion, por lo que no se ha añadido un ORDER BY EN EL CODIGO
854
855
856
857
```

Output

#	Time	Action	Message
634	14:07:07	Create view InformeTecnico as Select transaction.id as ID, data_user.name as Nombre, data_user.surname as Apellido, transaction.iban as IBAN, company_name as Nombre_compañia Fro...	Error Code: 1054. Unknown column 'transaction.iban' in field list'
635	14:07:27	Create view InformeTecnico as Select transaction id as ID, data_user.name as Nombre, data_user.surname as Apellido, credit_card.iban as IBAN, company_name as Nombre_compañia Fro...	0 row(s) affected
636	14:07:41	SELECT * FROM transactions.informetecnico LIMIT 0, 2000	586 row(s) returned

Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.

Sprint2 Sprint3 vistamarketing informetecnico

Limit to 2000 rows

1 • SELECT * FROM transactions.informetecnico;

ID	Nombre	Apellido	IBAN	Nombre_compañia
FE96CE47-8D59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2D86-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1F-8D41-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955...	Nunc Interdum Incorporated
FD89D518-AE8D-77DC-E450-88083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2B957-414B-E8EC-E9AD-59AA7ABA6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-28DC-9E49-8D092A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
FBD7E0D6-A6A8-F5BC-QCA9-EA4B8760100C	Hedwig	Gilbert	MU4132333444534342541344788...	Mauris Id Inc.
FAC76A8D-8448-69AA-E892-426C2F12621C	Slade	Poole	MT053WCF5886820057577163458...	Arcu LLP
FAAD3FFC-1A17-E141-43D3-359A5BA7CB38	Hedwig	Gilbert	GE90157928843338134463	Lorem Eu Incorporated
FA053936-75D8-89FA-490D-98624E1B920A	Hedwig	Gilbert	GT0249765365330848247645975	Non Justo Corp.
F85A7D75-27F8-9D75-0776-3F41A828CE88	Sarah	Beck	VG1468087984174645729577	Ut Semper Foundation
F843D0C8-C8B5-2444-1B4E-5966289FBA88	Jasper	Landry	VG1468087984174645729577	Ut Semper Foundation
FSACD74B-4275-5AA1-2414-6EF417636B98	Nora	Reeves	MD1234119525145401270486	Nunc Interdum Incorporated
F56FCA4A-0039-9F64-7376-85632B91121B	Lynn	Riddle	CR7242477244335841535	Ut Semper Foundation
F5583CE1-3379-E0BF-5AB9-6F4CC2C5479C	Sonya	McKee	EE541536644818872885	Arcu LLP
F4BCAE41-388E-EA8D-9C24-466F7CEB9F9A	Chester	Haynes	CY94263537405015481188625576	Malesuada PC
F2B3645-2E6D-E891-9D05-33D8ACE58DE4	Heather	Burks	SM6022751049715477062682363	Malesuada PC
F28E106B-5418-4667-9514-2E2A823ECC65	Hedwig	Gilbert	RO76D4F06583348580208155	Pede Ultrices Ltd
F233E385-5A74-63B3-8111-09FC9BA38011	Nero	Mills	HU95215627749276573565556322	Arcu LLP
F22B8361-E3CD-BC41-0D6A-A4694F175CD8	Alko	Chaney	MD5723087436783068347555	Non Institute
F1A598A2-96C5-50A9-F1CE-FB1D69866C39	Craig	Shepherd	R323456312213576817699999	Lorem Eu Incorporated
EFEECC2E-2A69-AB33-Q599-E82AA689E3B3	Alka	Kinney	R323456312213576817699999	Lorem Eu Incorporated
BF9E0849-457D-D7C1-41E4-059831C35CEC	Cassa...	Ferguson	DO68192976973138848171352176	Ut Semper Foundation
EAAE515-6686-B449-5926-54D397988AD9	Ocean	Nelson	AE640696354928782425103	Et Magnis Ltd
EE8D5A8-CEC8-9387-524C-A911BB473C1F	Ainsley	Herrera	BH62714428368066765294	Enim Conditum Ltd
EE2D8157-D63E-6A91-A363-9CF688530AD3	Jade	Porter	GI98JML122444875373267	Malesuada PC
EDC458E3-4E51-2D30-5CD6-838D10959BA5	Hedwig	Gilbert	RO76D4F06583348580208155	Pede Ultrices Ltd
ED9CCFC6-33C8-D6A9-731F-876ZDF1659E	Sonya	Hobbs	BH62714428368066765294	Enim Conditum Ltd
ED1CFB7D-E626-CE54-57A1-A03677885E1F	Clark	Olson	MT89SLEX1421609154562585469...	Cras Consulting
ECA28D28-2B62-921A-A912-E68C2806C27E	Hedwig	Gilbert	PL80213758374176596516617081	At Pede Corp.
EC8D77B4-5570-7DC6-3DE9-E46A861A7479	Clark	Hewitt	MD1234119525145401270486	Enim Conditum Ltd
EC243849-3D95-A9B3-29B4-6F8AA1A5B999	Ocean	Nelson	AL36723216264979973318818691	Non Institute
EB46D6A6-0851-FA55-1292-D804B2684794	Lynn	Riddle	CR7242477244335841535	Ut Semper Foundation
EB1426E1-B4EC-3423-4FE9-6FB2151A3D85	Cleo	George	GE36372634666440824664	Nunc Interdum Incorporated
EAE19DC1-C847-6D79-673D-00E7696AC336	Raymond	Onell	CR29344270026455556549	Nunc Interdum Incorporated
EAD9435C-77C0-3D84-D9A9-BF535EDABE7A	Ocean	Nelson	MKS23334446478878011	Non Institute
EAD89828-65AA-E817-F946-E7365972D94E	Stone	Robbins	S135069513568830361	Malesuada PC
EAC77318-72E4-51E3-AB5A-9BCBBA39C8F2	Hedwig	Gilbert	CH8995351081824762557	Non Ante LLP
EA69483B-09A2-6170-73C4-65AFE5B71F4F	Brennan	Wynn	CR7242477244335841535	Ut Semper Foundation
EA41A50E-27D6-3612-788C-AD736D4EF523	Chase	Ellis	CR7242477244335841535	Ut Semper Foundation
EA2C3281-C9C1-A387-44F8-729FB451C76	Kenyon	Hartman	R323456312213576817699999	Ac Fermentum Incorporated
EA06C211-6A86-C9A8-7CA3-C84D266D4B62	Burke	Graham	GE38261236401561281317	Arcu LLP
E95B26BA-2466-6ECB-81D4-57CB6225AB78	Porter	Francis	SE2813123487163628531121	Arcu LLP
EBA862CE-AC8B-A89-620C-DDD7FEB16AA2	Gisela	Johnston	BH62714428368066765294	Enim Conditum Ltd
E8954683-8BA4-B66B-C641-569542558DC	Elton	Roberson	ME89166252329582146461	Malesuada PC

informetecnico 1 x

Output

#	Time	Action	Message
634	14:07:07	Create view InformeTecnico as Select transaction.id as ID, data_user.name as Nombre, data_user.surname as Apellido, transaction.iban as IBAN, company_name as Nombre_compañia Fro...	Error Code: 1054. Unknown column 'transaction.iban'
635	14:07:27	Create view InformeTecnico as Select transaction id as ID, data_user.name as Nombre, data_user.surname as Apellido, credit_card.iban as IBAN, company_name as Nombre_compañia Fro...	0 row(s) affected
636	14:07:41	SELECT * FROM transactions.informetecnico LIMIT 0, 2000	586 row(s) returned

-- IMPORTANTE Se ha ordenado en orden descendente utilizando la interfaz grafica de la vista en función de la variable ID de transacción, por lo que no se ha añadido un ORDER BY EN EL CODIGO