

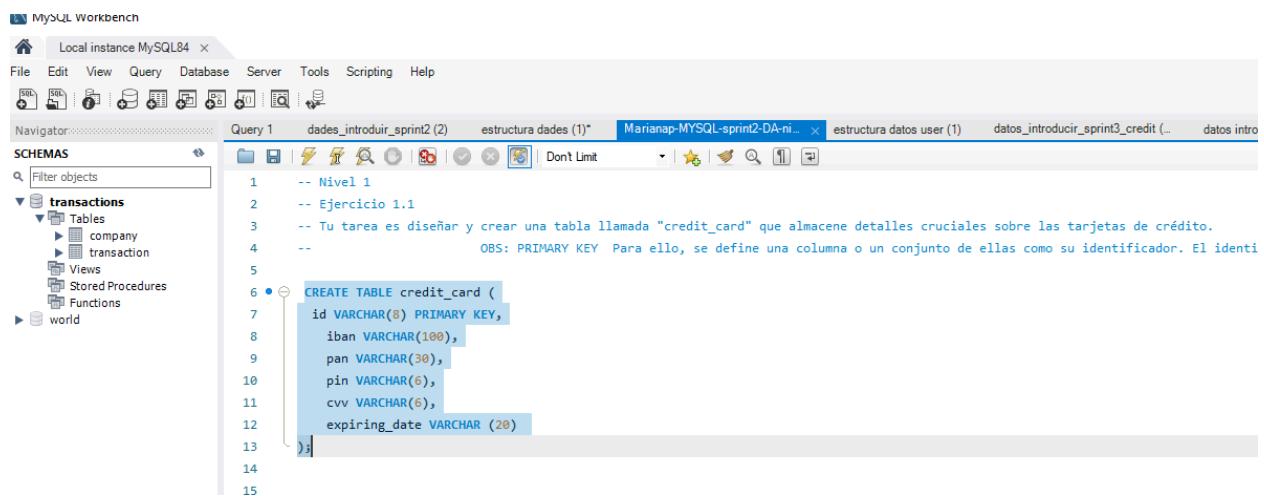
Data Analytics - SQL

Sprint 3 - Mariana P.

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.



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** Local instance MySQL84, File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (transactions), Tables (company, transaction), Views, Stored Procedures, Functions, world.
- Query Editor:** Title: dades_introduir_sprint2 (2) - estructura dades (1)*. Content:

```
1 -- 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 --
5
6 • CREATE TABLE credit_card (
7     id VARCHAR(8) PRIMARY KEY,
8     iban VARCHAR(100),
9     pan VARCHAR(30),
10    pin VARCHAR(6),
11    cvv VARCHAR(6),
12    expiring_date VARCHAR (20)
13 )
14
15
```
- Toolbars:** Standard MySQL Workbench toolbar.

-- COMENTARIOS: aquí tuve que modificar los valores de fechas que no estaban como VARCHAR y por eso no me dejaba ejecutar la tabla.

- 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").

Schemas

```

15
16 -- La nueva tabla debe ser capaz de identificar de forma única cada tarjeta y establecer una relación adecuada con las otras dos.
17 -- Después de crear la tabla será necesario que ingreses la información del documento denominado "datos_introducir_credit".
18
19
20 • SET SQL_SAFE_UPDATES = 0;
21
22
23 • ALTER TABLE transaction
24 ADD CONSTRAINT fk_cc_transaction
25 FOREIGN KEY (credit_card_id)
26 REFERENCES credit_card(id);
27
28

```

MySQL Workbench

Local instance MySQL84 (trans...) MySQL Model EER Diagram

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

transaction

Tables

- company
- credit_card
- transaction

Columns

- id
- credit_card_id
- company_id
- user_id
- lat
- longitude
- timestamp
- amount
- declined
- avg_sales

Indexes

Foreign Keys

Triggers

user

Views

Stored Procedures

Functions

world

Administration Schemas

Information

Table: transaction

Columns:

	Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
1	<input checked="" type="checkbox"/> id	VARCHAR(100)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL					
2	<input checked="" type="checkbox"/> credit_card_id	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
3	<input checked="" type="checkbox"/> company_id	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
4	<input checked="" type="checkbox"/> user_id	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
5	<input checked="" type="checkbox"/> lat	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
6	<input checked="" type="checkbox"/> longitude	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
7	<input checked="" type="checkbox"/> timestamp	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
8	<input checked="" type="checkbox"/> amount	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
9	<input checked="" type="checkbox"/> declined	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
10	<input checked="" type="checkbox"/> avg_sales	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name: Data Type: Charset/Collation: Default: Comments: Storage: Virtual Stored Primary Key Not Null Unique Binary Unsigned Zero Fill Auto Increment Generated

Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	11:10:52	CREATE TABLE credit_card (id VARCHAR(8) PRIMARY KEY, ib...	Error Code: 1050. Table 'credit_card' already exists	0.000 sec
6	11:20:42	Apply changes to transaction		
7	11:22:11	ALTER TABLE transactions.transaction CHANGE COLUMN id as i...	Error Code: 1064. You have an error in your SQL syntax; check the ...	0.000 sec
8	11:27:15	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.000 sec
9	11:28:56	UPDATE credit_card SET expiring_date = STR_TO_DATE(expirin...	Error Code: 1054. Unknown column 'expiring_date' in field list'	0.000 sec
10	11:29:40	UPDATE credit_card SET expiring_date = "%m/%d/%y"	Error Code: 1054. Unknown column 'expiring_date' in field list'	0.000 sec
11	11:30:21	ALTER TABLE transactions ADD CONSTRAINT cc.transaction FO...	Error Code: 1064. You have an error in your SQL syntax; check the ...	0.000 sec
12	11:30:39	ALTER TABLE transactions ADD CONSTRAINT cc.transaction FO...	Error Code: 1064. You have an error in your SQL syntax; check the ...	0.000 sec
13	11:31:12	ALTER TABLE transactions.transaction ADD CONSTRAINT fk_cc...	Error Code: 1166. Table 'transactions' has no primary key and...	0.000 sec

Table: transaction

Columns:

- id** varchar(100) PK
- credit_card_id** varchar(100)

Foreign Keys:

Foreign Key Name	Referenced Table	Column	Referenced Column	Foreign Key Options
fk_cc_transaction	'transactions'.`credit_card`	<input type="checkbox"/> id <input type="checkbox"/> credit_card_id <input checked="" type="checkbox"/> company_id <input type="checkbox"/> user_id <input type="checkbox"/> lat <input type="checkbox"/> longitude <input type="checkbox"/> timestamp <input type="checkbox"/> amount <input type="checkbox"/> declined <input type="checkbox"/> avg_sales	<input type="checkbox"/> id	On Update: RESTRICT On Delete: RESTRICT
fk_c_transaction	'transactions'.`company`	<input type="checkbox"/> id <input type="checkbox"/> credit_card_id <input type="checkbox"/> company_id <input type="checkbox"/> user_id <input type="checkbox"/> lat <input type="checkbox"/> longitude <input type="checkbox"/> timestamp <input type="checkbox"/> amount <input type="checkbox"/> declined <input type="checkbox"/> avg_sales	<input type="checkbox"/> id	<input type="checkbox"/> Skip in SQL generation

Triggers:

Partitioning:

Options:

Apply Revert

COMENTARIOS:

Primary Key = **id** en la tabla **transaction**

Foreign Key 1= **fk_cc_transaction**

Foreign Key 2 = **fk_c_transaction**

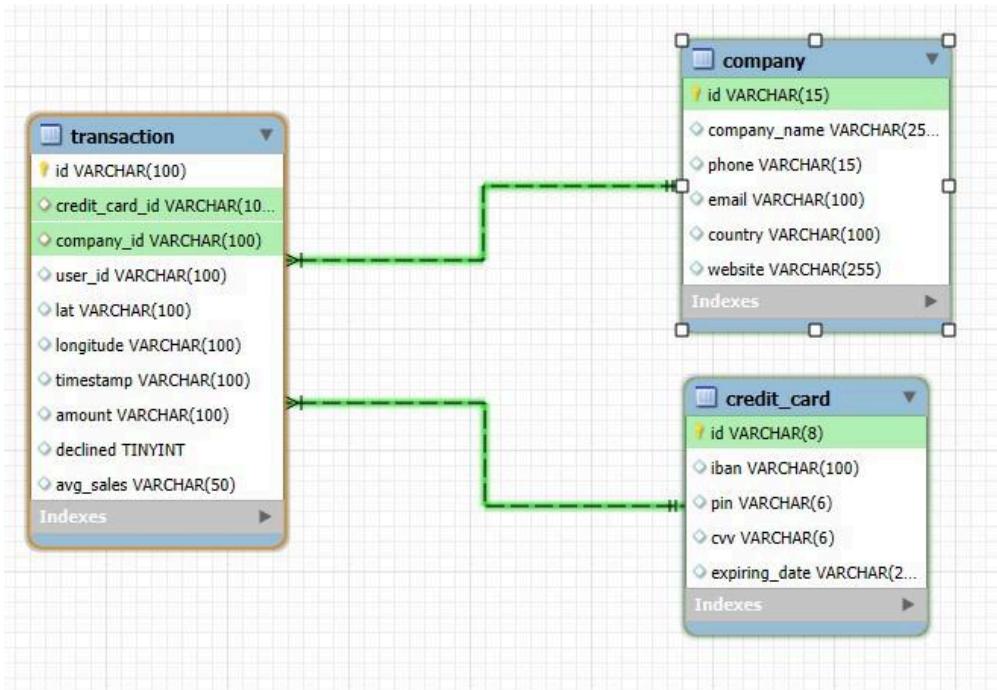
- Después de crear la tabla será necesario que ingreses la información del documento denominado "datos_introducir_credit".

SELECT *
FROM credit_card

Result Grid

	id	iban	pan	pin	cvv	expiring_date
Ccs-4857	XX485791835292505850771	2314242385113924	1819	467	09/27/25	
Ccs-4858	XX8581768137002436094025	6582720299715533	3964	817	12/28/28	
Ccs-4859	XX7826930491423553609374	8861684536289642	4983	277	11/26/26	
Ccs-4860	XX555959036838350464529	2481155151498459	6876	661	07/27/27	
Ccs-4861	XX2035182877195191627307	1308930301149557	5710	399	04/25/26	
Ccs-4862	XX4774721462463645409758	6715617009807829	4042	174	11/27/26	
Ccs-4863	XX1476829664245046207111	3140879819451394	5969	449	12/27/29	
Ccs-4864	XX8380298893385731196159	5793672133649114	8481	139	02/28/26	
Ccs-4865	XX7085078596101025280599	5101552687251312	7847	903	11/25/28	
Ccs-4866	XX4792859188206596406839	8080768801072613	9271	961	02/28/25	

- Recuerda mostrar el diagrama y realizar una breve descripción del mismo.



Ejercicio 1.2

El departamento de Recursos Humanos ha identificado un error en el número de cuenta asociado a su tarjeta de crédito con ID CcU-2938. La información que debe mostrarse para este registro es: TR323456312213576817699999. Recuerda mostrar que el cambio se realizó.

-- COMENTARIOS: aqui tengo 2 opciones principales: reemplazar el iban directamente o drop y luego insert el nuevo registro TR:

-- es importante confirmar si este valor es único o si hay otras filas con el mismo valor.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema with tables like 'credit_card' and 'transaction'. The main pane shows the 'credit_card' table structure with columns: id, iban, pan, pin, cvv, and expiring_date. An update query is being typed into the SQL editor:

```
update credit_card
set iban = 'TR323456312213576817699999'
where id = 'CcU-2938';
```

The results grid below shows one row of data:

id	iban
CcU-2938	TR323456312213576817699999
*	HULL

Ejercicio 1.3

En la tabla "transaction" ingresa una nueva transacción 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

-- COMENTARIOS:

- el ejercicio no ofrece datos en la columna timestamp*
- tuve que actualizar las palabras 'user', 'amount' y 'longitud'

```
94    -- Nivel 1
95    -- Ejercicio 3
96    -- En la tabla "transaction" ingresa una nueva transacción con la siguiente información:
97
98 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
99     VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0')
100
101
102 • select id, credit_card_id, company_id, user_id, lat, longitude, amount, declined
103     from transaction
104 ;
```

Ejercicio 1.4

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

```
106      -- Nivel 1
107      -- Ejercicio 1.4
108      -- Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit_card. Recuerda mostrar el cambio realizado.
109
110 • ALTER TABLE credit_card
111     DROP COLUMN pan;
112
113 • select pan
114     from credit_card
115
116 ✘ select *
117     from credit_card;
118
119
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

	id	iban	pin	cvv	expiring_date
▶	CcS-4857	XX4857591835292505850771	1819	467	09/27/25

Nivel 2

Ejercicio 2.1

Elimina de la tabla transacción el registro con
ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de datos.

```
135
136      -- Nivel 2
137      -- Ejercicio 1
138      -- Elimina de la tabla transacción el registro con ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de datos.
139
140 • DELETE FROM transaction
141 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
142
143
144 • SELECT id
145     FROM transaction
146 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
147
148
149 -- Nivel 2
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

	id
•	NULL

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 realizada por cada compañía.

**Presenta la vista creada, ordenando los datos de mayor a menor promedio de compra

The screenshot shows the MySQL Workbench interface. The Navigator pane on the left displays the database schema, including the 'transaction' table under the 'transaction' schema, and a 'vistamarketing' view under the 'world' schema. The main window contains a query editor with the following SQL code:

```
-- Nivel 2
-- Ejercicio 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.

-- 1 crear una vista llamada VistaMarketing que llevará datos de las dos tablas: company y transaction

CREATE VIEW VistaMarketing AS (
    SELECT c.company_name, c.phone, c.country, AVG(t.amount) AS avg_sales
    FROM company c
    JOIN transaction t
    ON c.id = t.company_id
    WHERE t.amount > 0
    GROUP BY c.company_name, c.phone, c.country
    ORDER BY avg_sales DESC);
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** transaction, user, vistamarketing
- Views:** vistamarketing (with columns company_name, phone, country, avg_sales)
- Query 1:** SELECT * FROM transactions.vistamarketing;
- Result Grid:** Displays data from the vistamarketing view.

company_name	phone	country	avg_sales
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.867160
Pretium Neque Corp.	07 77 48 55 28	Australia	276.158330
Urna Convallis Associates	06 01 24 77 04	United States	274.235011
At Associates	09 56 61 10 65	New Zealand	272.214870
Metus Vitae Associates	08 25 44 40 66	Australia	270.080965
Aliquet Diam Limited	02 76 61 47 46	United States	269.599181
Nec Luctus LLC	02 14 71 75 73	Norway	268.604837
Neque Tellus Incorporated	04 43 18 34 19	Ireland	267.850372
Tortor Nunc Commodo Company	05 35 92 77 16	United States	267.836085
Cras Consulting	07 50 10 85 63	Belgium	267.439409
Fringilla Tristique Ltd	03 67 44 56 77	United States	267.060541

Ejercicio 2.3

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

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** transaction, user, vistamarketing
- Views:** vistamarketing (with columns company_name, phone, country, avg_sales)
- Query 1:** (Annotated code)

```

170    DROP COLUMN avg_sales;
171
172
173    -- Nivel 2
174    -- Ejercicio 3
175    -- Filtra la vista VistaMarketing para mostrar sólo las compañías que tienen su país de residencia en "Germany"
176
177 •    SELECT company_name, country
178      FROM vistamarketing
179     WHERE country = 'Germany';

```
- Result Grid:** Displays data from the filtered vistamarketing view.

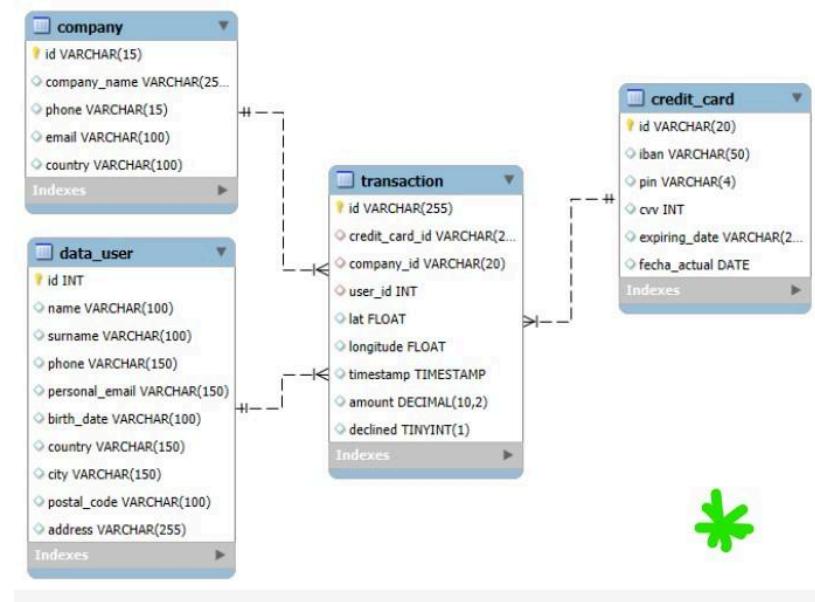
company_name	country
Ac Fermentum Incorporated	Germany
Nunc Interdum Incorporated	Germany
Convallis In Incorporated	Germany
Ac Industries	Germany
Rutrum Non Inc.	Germany
Auctor Mauris Corp.	Germany
Augue Foundation	Germany
Aliquam PC	Germany

Nivel 3

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:

Diagrama Modelo



-- ** Recordatorio: En esta actividad, es necesario que describas el "paso a paso" de las tareas realizadas.

-- ** Es importante realizar descripciones sencillas, simples y fáciles de comprender. Para realizar esta actividad deberás trabajar con los archivos denominados "estructura_datos_user" y "datos_introducir_user"

-- ** Recuerda seguir trabajando sobre el modelo y las tablas con las que ya has trabajado hasta ahora.

PASO A PASO

1. Entender la relación entre todas las tablasa través de las 'Primaray Key' y 'Foreign Key'. Osea, entender cual tabla tiene la FK y cual columna es la FK de esa tabla.

```
-- En este Schema todas las tablas tienes una columna llamada 'ID' pero eso no significa que
todas tienan relación entre ellas.
-- Es visible que la tabla 'Transaction' tienes las siguientes relaciones:
-- (PRIMARY) transaction -> company = company_id (FOREIGN) / company.id (PRIMARY)
-- (PRIMARY) transaction -> credit_card = credit_card_id (FOREIGN)/ credit_card.id
(PRIMARY)
-- (PRIMARY) transaction -> user = user_id (FOREIGN) / user.id (PRIMARY)
```

-- 1.Comparamos los tipos de valores entre las dos tablas:

```
SHOW columns FROM transaction;
```

```
SHOW columns FROM user;
```

-- 2. Actualizamos la tabla 'user' desde INT a VARCHAR

```
ALTER TABLE user
```

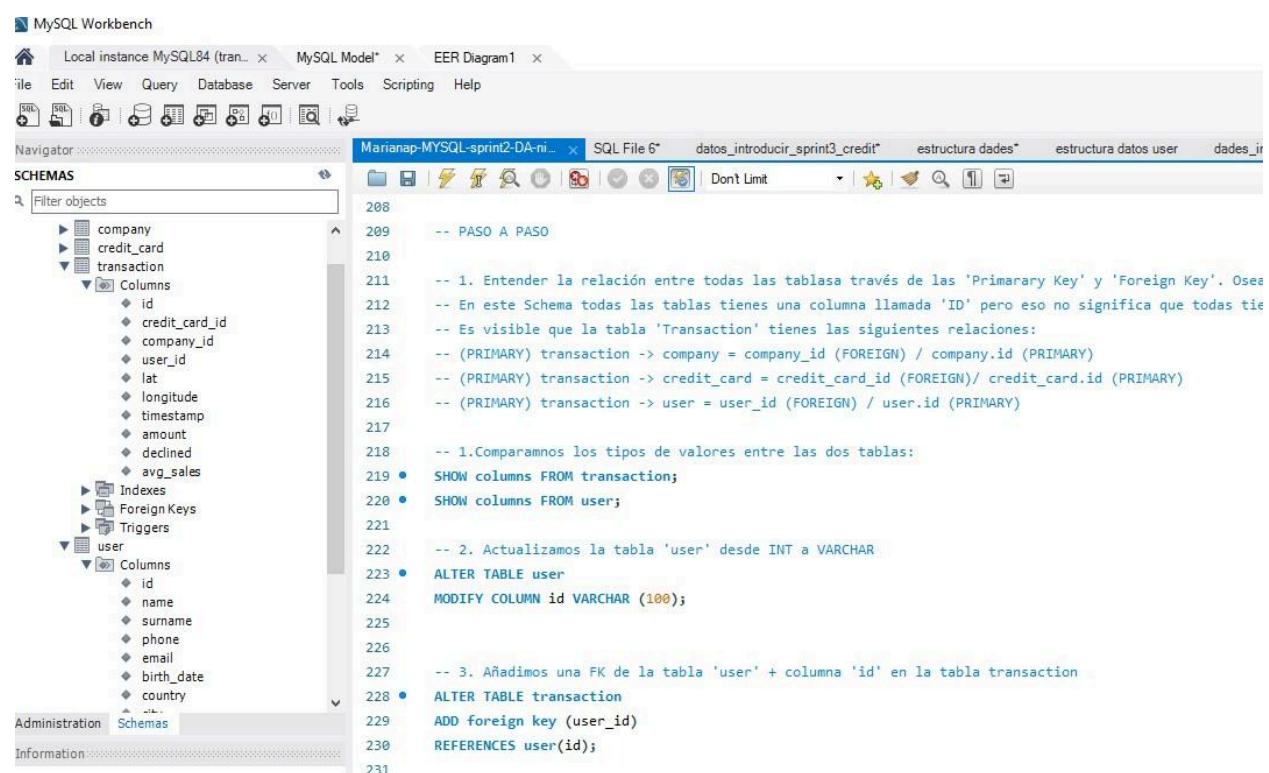
```
MODIFY COLUMN id VARCHAR (100);
```

-- 3. Añadimos una FK de la tabla 'user' + columna 'id' en la tabla transaction

```
ALTER TABLE transaction
```

```
ADD foreign key (user_id)
```

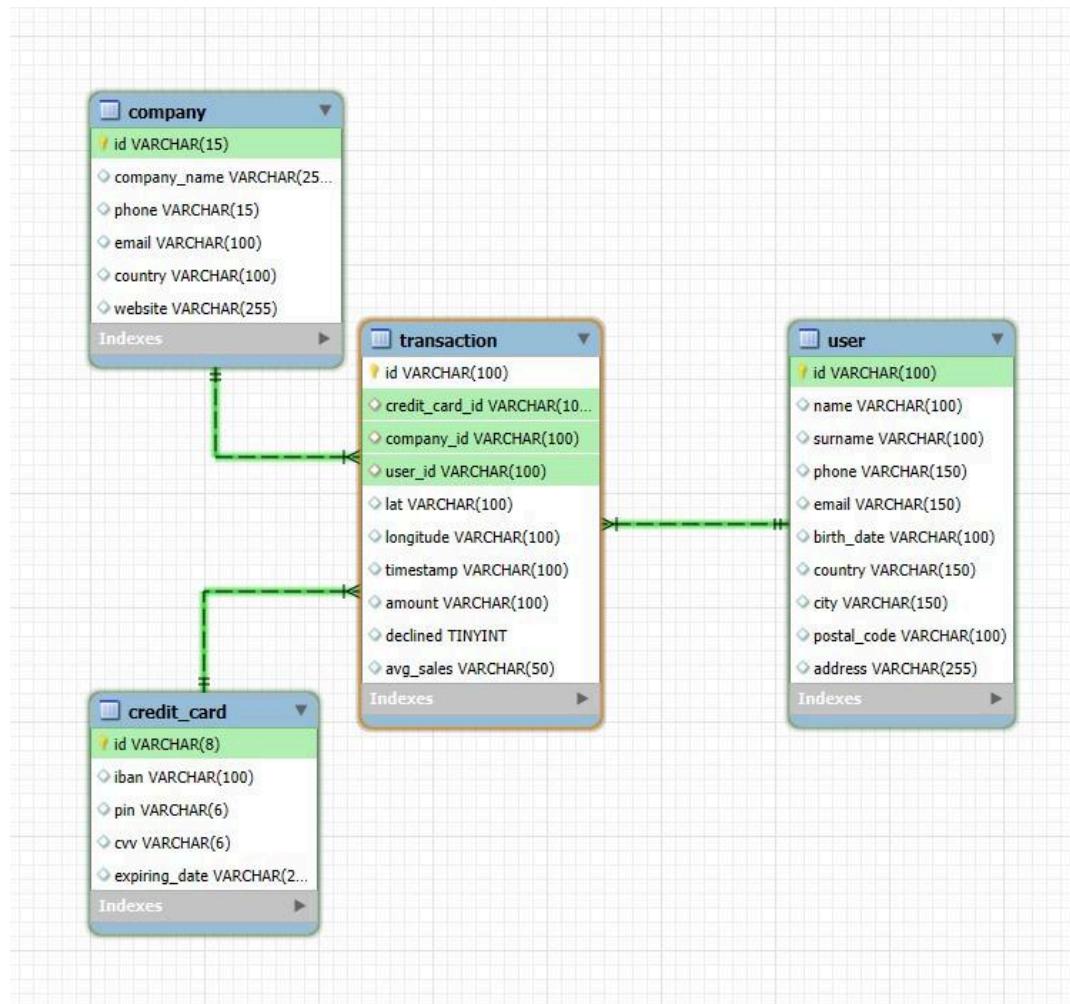
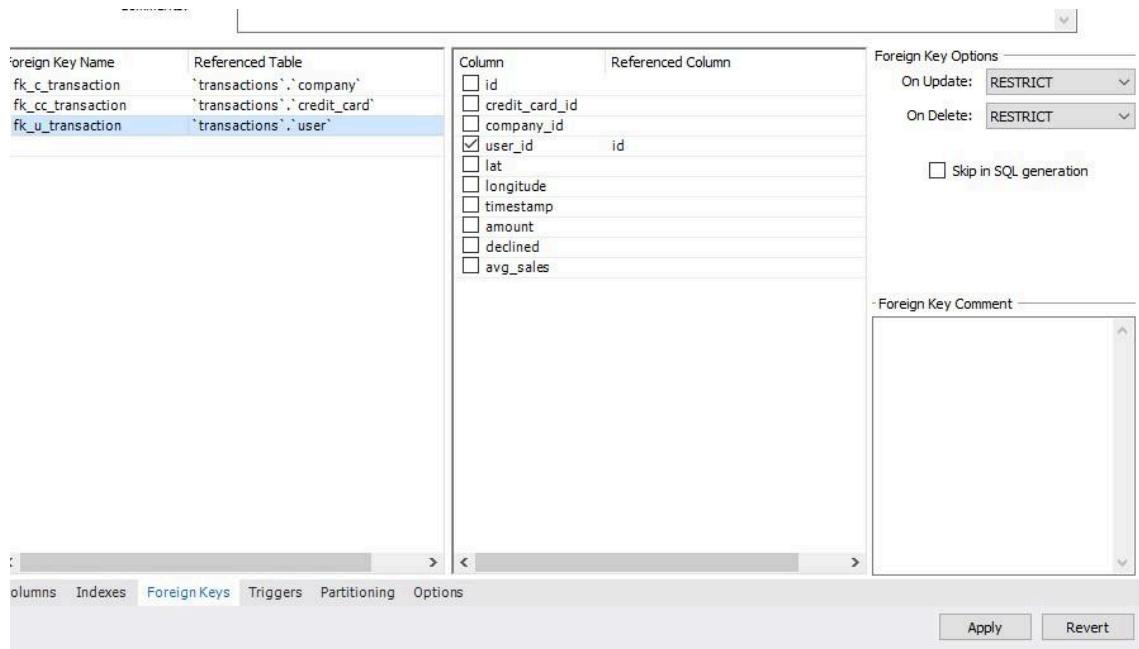
```
REFERENCES user(id);
```



The screenshot shows the MySQL Workbench interface with the SQL editor open. The code in the editor is as follows:

```

208
209 -- PASO A PASO
210
211 -- 1. Entender la relación entre todas las tablasa través de las 'Primaray Key' y 'Foreign Key'. Osea
212 -- En este Schema todas las tablas tienes una columna llamada 'ID' pero eso no significa que todas tie
213 -- Es visible que la tabla 'Transaction' tienes las siguientes relaciones:
214 -- (PRIMARY) transaction -> company = company_id (FOREIGN) / company.id (PRIMARY)
215 -- (PRIMARY) transaction -> credit_card = credit_card_id (FOREIGN)/ credit_card.id (PRIMARY)
216 -- (PRIMARY) transaction -> user = user_id (FOREIGN) / user.id (PRIMARY)
217
218 -- 1.Comparamos los tipos de valores entre las dos tablas:
219 • SHOW columns FROM transaction;
220 • SHOW columns FROM user;
221
222 -- 2. Actualizamos la tabla 'user' desde INT a VARCHAR
223 • ALTER TABLE user
224 MODIFY COLUMN id VARCHAR (100);
225
226
227 -- 3. Añadimos una FK de la tabla 'user' + columna 'id' en la tabla transaction
228 • ALTER TABLE transaction
229 ADD foreign key (user_id)
230 REFERENCES user(id);
231
```



Ejercicio 3.2

La empresa también le pide 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.

Asegúrese de incluir información relevante de las tablas que conocerá y utilice alias para cambiar de nombre columnas según sea necesario.

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

```
238 -- Nivel 3
239 -- Ejercicio 2
240 -- La empresa también le pide crear una vista llamada "InformeTecnico" que contenga la siguiente información:
241 -- ID de la transacción
242 -- Nombre del usuario/a
243 -- Apellido del usuario/a
244 -- IBAN de la tarjeta de crédito usada.
245 -- Nombre de la compañía de la transacción realizada.
246
247
248 -- **Asegúrese de incluir información relevante de las tablas que conocerá y utilice alias para cambiar de nombre columnas según sea necesario.
249 -- **Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.
250
251 • CREATE VIEW InformeTecnico AS
252     SELECT t.id AS id,
253         u.name, u.surname, cc.iban, c.company_name
254     FROM transaction t
255     JOIN user u
256     ON t.user_id = u.id
257     JOIN credit_card cc
258     ON t.credit_card_id = cc.id
259     JOIN company c
260     ON t.company_id = c.id
261     ORDER BY t.id DESC;
262
263
264 • SELECT *
265     FROM InformeTecnico
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
```

id	name	surname	iban	company_name
FFFFD1D6-9495-47CE-B54A-7D8BE1CC274B	Bmrgli	Tprvmmrc	XX79-814451211289182490...	Turpis Company
FFFFCF76D-ECF0-4985-A2D0-82AB75998FC	Dfied	Vlajqjd	XX63625170164789203667...	Amet Nulla Donec Corporation
FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	FaoVqfy	XX1262771433042236314537...	Nunc Interdum Incorporated
FFFFB270D-F53A-4D5D-9666-E5307C53C84	Gzgpa	Urzjulh	XX395114267082019952567...	Viverra Donec Foundation
FFFF9E3CE-234E-408C-A8EF-F9CAD577224A	Yshmg	Zpspleed	XX8845462156537570367941	Convallis In Incorporated

EXTRA *consulta personal (no considerar)

CONSULTA:

https://www.w3schools.com/sql/sql_ref_mysql.asp
<https://www.youtube.com/watch?v=q5wFWfsS-4I>
<https://www.youtube.com/watch?v=KUEdpNf284U>
https://www.youtube.com/watch?v=9pfkpdF_N40
https://www.youtube.com/watch?v=9pfkpdF_N40
<https://10web.io/blog/mysql-error-1452>
https://www.w3schools.com/sql/sql_view.asp
<https://www.youtube.com/watch?v=BXUzPi7YhLA>
https://www.w3schools.com/mysql/mysql_join_cross.asp

EJEMPLOS DE QUERY para identificar PK y FK:

-- EJEMPLO de query para identificar -----

SELECT

```
    TABLE_NAME,  
    COLUMN_NAME,  
    CONSTRAINT_NAME,  
    REFERENCED_TABLE_NAME,  
    REFERENCED_COLUMN_NAME  
FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE  
WHERE REFERENCED_TABLE_NAME IS NOT NULL;
```

-- EJEMPLO DE QUERY PARA CONECTAR TODAS LAS TABLAS

SELECT

```
    t1.columnA,  
    t2.columnB,  
    t3.columnC,  
    t4.columnD  
FROM table1 AS t1  
JOIN table2 AS t2  
    ON t1.columnX = t2.columnY  
JOIN table3 AS t3  
    ON t1.columnM = t3.columnN  
JOIN table4 AS t4  
    ON t1.columnP = t4.columnQ;
```

```
-- finding primary key -----
SELECT
    TABLE_NAME,
    COLUMN_NAME
FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE
WHERE CONSTRAINT_NAME = 'PRIMARY'
    AND TABLE_SCHEMA = 'your_database_name';
```

```
-- finding foreign keys -----
SELECT
    TABLE_NAME,
    COLUMN_NAME,
    CONSTRAINT_NAME,
    REFERENCED_TABLE_NAME,
    REFERENCED_COLUMN_NAME
FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE
WHERE REFERENCED_TABLE_NAME IS NOT NULL
    AND TABLE_SCHEMA = 'your_database_name';
```