


Sprint 3

Manipulacion de Tablas



Nivell 1

- 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.



The screenshot shows a database management tool interface. On the left, the 'Navigator' pane displays the 'transactions' schema with tables 'company', 'credit_card', and 'transaction'. The 'credit_card' table is selected, showing its columns: id, iban, pan, pin, cvv, and expiring_date. The main editor shows the SQL code to create the 'credit_card' table with the following structure:

```
CREATE TABLE transactions.credit_card
(
  id VARCHAR (15) PRIMARY KEY,
  iban VARCHAR (34) NOT NULL,
  pan VARCHAR (19) NOT NULL,
  pin CHAR (4) NOT NULL,
  cvv CHAR (3) NOT NULL,
  expiring_date VARCHAR (10) NOT NULL,
  CONSTRAINT chk_pin_format CHECK (pin REGEXP "^[0-9]{4}$"),
  CONSTRAINT chk_cvv_format CHECK (cvv REGEXP "^[0-9]{3}$")
);
```

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

#	Time	Action	Message
7	10:27:42	DROP TABLE 'transactions'.'credit_card'	0 row(s) affected
8	10:27:46	CREATE TABLE transactions.credit_card (id VARCHAR (15) PRIMARY KEY, iban VAR...	0 row(s) affected

Se crea una tabla llamada credit_card en el esquema transactions.

Se le agregan los campos requeridos (id, iban, pan, pin, cvv, expiring_date) como columnas y se contrastan sus características en el archivo "dades_introduir_credit", para ver que de que se componen. (Se puede comenzar con VARCHAR 255 y luego adaptar a los datos, sin embargo, he optado por primero observar los datos y sus características y luego crear la tabla acorde a sus valores)

Para poder ligar la nueva tabla con las otras existentes a través de su PrimaryKey (credit_card.id), se explora que columna de las otras tablas hace referencia a esta. La tabla transaction se relaciona con su ForeignKey (credit_card_id) con la nueva tabla. Para que se pueda relacionar esta nueva tabla de credit_card con la existente a través de su PrimaryKey, debe tener las mismas características que la ya existente ForeignKey

de transaction. (VARCHAR(15)).

Se tiene en cuenta las características específicas de cada columna, que valores pueden tener, cuantos dígitos serán permitidos, por si serán exclusivamente números o también pueden contener letras, por si son obligatorios u opcionales.

En los campos pin y cvv se le ha aplicado un check constraint para garantizar de que solo contengan números del 0-9, además de que cumplan con la longitud obligatoria de 4 y 3.

Se especifica la PrimaryKey, el id de esta tabla, esto garantiza la unicidad de este valor además de que implica la función NOT NULL.

En las otras columnas se agrega la función NOT NULL, para que los registros sean obligatorios y no puedan dejarse vacíos.

Posteriormente se le introducen todos los datos contenidos en el archivo “dades_introducir_credit”

The screenshot displays a database management interface with three main panels. The left panel, titled 'SCHEMAS', shows a tree view of the database structure with 'credit_card' selected. The middle panel, 'Query 1', shows a SQL script with 15 'INSERT INTO' statements for the 'credit_card' table. The right panel, 'Output', shows the execution results of the queries.

Table: credit_card

Columns:	data type
id	char(8) PK
iban	varchar(34)
pan	char(19)
pin	char(4)
cvv	char(3)
expiring_date	varchar(10)

Query 1: datos_introducir_credit*

```
1
2  -- Insertamos datos de credit_card
3  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-2938', 'TR3019503122
4  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-2945', 'D02685476374
5  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-2952', 'B645IVQL5271
6  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-2959', 'CR7242477244
7  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-2966', 'B672LKTQ1562
8  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-2973', 'PT8780622813
9  • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-2980', 'DE3924188188
10 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-2987', 'GE8968143483
11 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-2994', 'BH6271442836
12 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-3001', 'CY4908742665
13 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-3008', 'LU5072166936
14 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-3015', 'PS1193982162
15 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (      'CcU-3022', 'GT9169516285
```

Output:

#	Time	Action	Message
551	10:44:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4...	1 row(s) affected
552	10:44:10	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4...	1 row(s) affected

Luego verifico si se han introducido los datos en la tabla de credit_card

23
24
25 • **SELECT ***
26 **FROM credit_card;**
27
28
29
30

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

	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

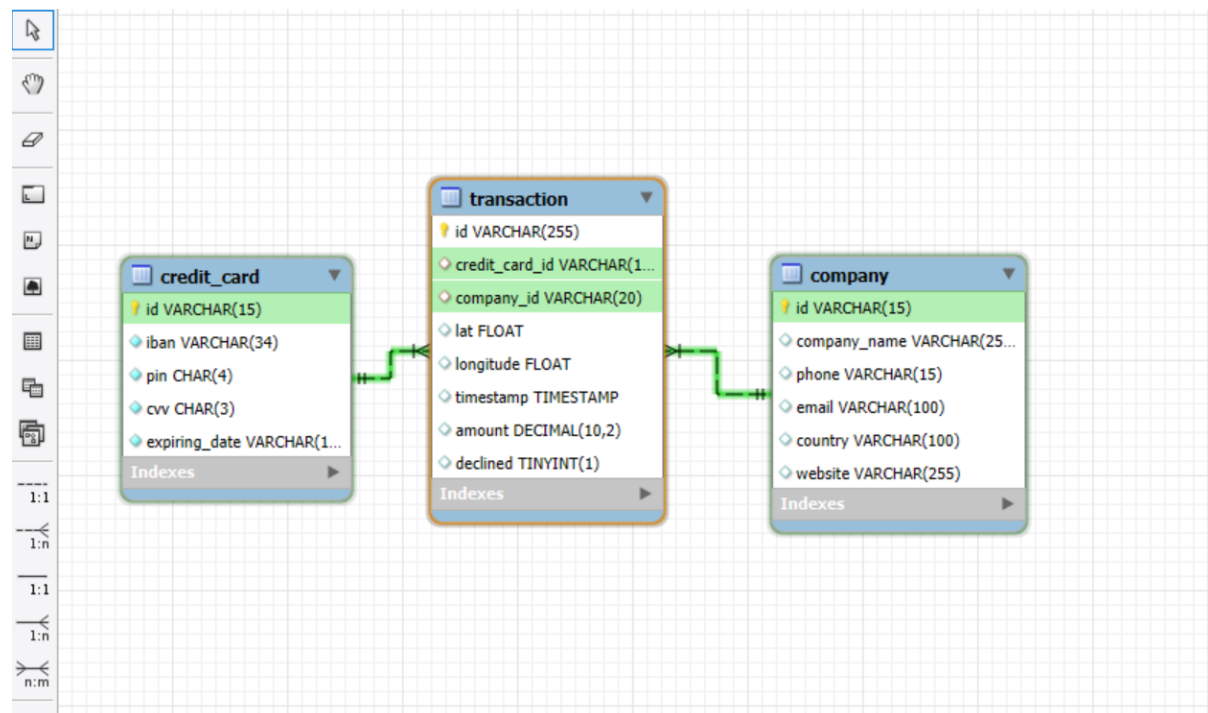
credit_card 2 x Apply

Output

Action Output

#	Time	Action	Message
✓ 8	10:27:46	CREATE TABLE transactions.credit_card (id VARCHAR(15) PRIMARY KEY, iban ...	0 row(s) affected
✓ 9	10:45:37	SELECT * FROM credit_card LIMIT 0, 50000	275 row(s) returned

El diagrama resultante es:



- 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.

Se cambia el registro del iban del usuario con el id CcU-2938, al que inicialmente se le habían asignado estos valores:

31
32 • `SELECT *`
33 `FROM credit_card`
34 `WHERE id = "CcU-2938";`
35
36
37
38

Result Grid

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

credit_card 3 x Apply

Output

Action Output

#	Time	Action	Message
✓ 9	10:45:37	SELECT * FROM credit_card LIMIT 0, 50000	275 row(s) returned
✓ 10	10:50:23	SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 50000	1 row(s) returned

33
34
35
36 • `UPDATE credit_card SET iban = "R323456312213576817699999" WHERE id = "CcU-2938";`
37 • `SELECT *`
38 `FROM credit_card`
39 `WHERE id = "CcU-2938";`
40

Result Grid

	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22
*	HULL	HULL	HULL	HULL	HULL	HULL

credit_card 4 x Apply

Output

Action Output

#	Time	Action	Message
✓ 11	10:52:43	UPDATE credit_card SET iban = "R323456312213576817699999" WHERE id = "...	1 row(s) affected Rows matched: 1 Changed: 1 Wa
✓ 12	10:52:59	SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 50000	1 row(s) returned

Con el comando update, se cambia el número de cuenta de este usuario.

Primero se elige la tabla en la que se quiere realizar el cambio, luego el campo que se quiere "colocar" (set) que en este caso es el iban y se inserta el valor que debe tener, después del =. Además, se tiene que especificar para quien se realiza este cambio, el usuario con el id CcU-2938. Siempre es recomendable usar la Primary-Key, para filtrar el registro en el que se debe efectuar el cambio, para que no se cambien más valores si estos están repetidos, a no ser que sea el propósito.

- Exercici 3

En la taula "transaction" ingresa 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

```

50 USE company;
51 INSERT INTO company (id)
52 VALUES ("b-9999");
53 USE transactions;
54 INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
55 VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", "9999", "829.999", "-117.999",
56 111.11, 0);
57 SELECT *
58 FROM transaction
59 WHERE company_id = "b-9999";
60

```

The screenshot shows a database management interface. At the top, there's a toolbar with options like 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. Below this is a 'Result Grid' showing the data inserted into the 'transaction' table. The grid has columns for 'id', 'credit_card_id', 'company_id', 'user_id', 'lat', 'longitude', 'timestamp', 'amount', and 'declined'. The first row contains the values: '108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', 'NULL', '111.11', and '0'. Below the grid, there's a tab labeled 'transaction 5'. At the bottom, there's an 'Output' section showing the execution of SQL commands and their results. The commands are: 'USE company;', 'INSERT INTO company (id) VALUES ("b-9999");', 'USE transactions;', 'INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", "9999", "829.999", "-117.999", 111.11, 0);', and 'SELECT * FROM transaction WHERE company_id = "b-9999" LIMIT 0, 50000'. The results show that 1 row(s) were affected for the insert and 1 row(s) were returned for the select.

#	Time	Action	Message
17	10:59:13	INSERT INTO company (id) VALUES ("b-9999")	1 row(s) affected
18	10:59:28	INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, a...	1 row(s) affected
19	11:01:15	SELECT * FROM transaction WHERE company_id = "b-9999" LIMIT 0, 50000	1 row(s) returned

Para poder efectuar esta agregación a la tabla de transacciones, primeramente, hay

que agregarle a la tabla de company el company_id, ya que esta tabla está conectada con su Primary Key (los ids de las companys) a la tabla de transacciones.

Posteriormente se agregan los otros datos en la tabla de transacciones.

Podemos verificar que en la tabla de transacciones se han agregado los campos requeridos. El timestamp se ha quedado nulo ya que el dato no existe.

- Exercici 4

Des de recursos humans et solliciten eliminar la columna "pan" de la taula credit_card. Recordra mostrar el canvi realitzat.

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane shows a tree view with 'transactions' > 'credit_card' selected. The 'Columns' list for 'credit_card' includes 'id', 'iban', 'pin', 'cvv', and 'expiring_date'. The 'Table: credit_card' section shows the column definitions: 'id' (varchar(15) PK), 'iban' (varchar(34)), 'pin' (char(4)), 'cvv' (char(3)), and 'expiring_date' (varchar(10)).

The main editor shows the following SQL commands:

```
62
63
64 • Alter table credit_card drop pan;
65 • SELECT *
66 FROM credit_card;
67
68
69
```

The 'Result Grid' shows the following data:

	id	iban	pin	cvv	expiring_date
▶	CcU-2938	R3234563122135768176999999	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

The 'Output' pane shows the 'Action Output' table:

#	Time	Action	Message
20	11:03:57	CREATE TABLE transactions.credit_card (id VARCHAR (15) PRIMARY KEY, iban ...	Error Code: 1050. Table 'credit_card' already exist
21	11:04:00	Alter table credit_card drop pan	0 row(s) affected Records: 0 Duplicates: 0 Warn
22	11:04:08	SELECT * FROM credit_card LIMIT 0, 50000	275 row(s) returned



Nivell 2

Exercici 1

Elimina de la taula transaction el registre amb ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de dades.

```
72
73 • delete from transaction where id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
74 • SELECT *
75   FROM transaction where id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
76
77
78
79
```

Result Grid

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 7 x Apply

Output

Action Output

#	Time	Action	Message
✓ 22	11:04:08	SELECT * FROM credit_card LIMIT 0, 50000	275 row(s) returned
✓ 23	11:19:01	delete from transaction where id = "02C6201E-D90A-1859-B4EE-88D2986D3B02"	0 row(s) affected
✓ 24	11:19:05	SELECT * FROM transaction where id = "02C6201E-D90A-1859-B4EE-88D2986D3B02"	0 row(s) returned

Aquí eliminamos el registro y luego verificamos que este eliminado.

Exercici 2

La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives. S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions. Serà necessària que creïs una vista anomenada VistaMarketing que contingui la següent informació: Nom de la companyia. Telèfon de contacte. País de residència. Mitjana de compra realitzat per cada companyia. Presenta la vista creada, ordenant les dades de major a menor mitjana de compra.

SCHEMAS

Filter objects

- Triggers
- transaction
- Views
 - vistamarketing
 - nombrecompañia
 - telefono
 - pais
 - compramedia
- Stored Procedures
- Functions
- world

Administration Schemas

Information: View: **vistamarketing**

Columns:

- nombrecompañia varchar(25)
- telefono varchar(15)
- pais varchar(10)
- compramedia decimal(10,2)

```

85 SELECT company_name as nombrecompañia, phone as telefono, country as pais,
86 round(avg(amount),2) as compramedia
87 FROM company
88 JOIN transaction ON company_id = company.id
89 WHERE declined = 0
90 GROUP BY nombrecompañia, telefono, pais
91 ORDER BY compramedia DESC;
92

```

nombrecompañia	telefono	pais	compramedia
Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
Sed Id Limited	07 28 18 18 13	United States	477.51
Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10
Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
Non Magna LLC	06 71 73 13 17	United Kingdom	458.74

Result 9 x

Output

Action Output

#	Time	Action	Message
26	11:22:28	SELECT company_name as nombrecompañia, phone as telefono, country as pais, ...	Error Code: 1052. Column 'id' in on clause is ambi
27	11:22:38	SELECT company_name as nombrecompañia, phone as telefono, country as pais, ...	101 row(s) returned
28	11:24:04	Apply changes to vistamarketing	Changes applied

Primero se crea la tabla con los resultados que se quieren ver y luego se crea una Vista a la que podemos acceder con SELECT.

SCHEMAS

Filter objects

- transaction
- user
- Views
 - informetecnico
 - vistamarketing
 - nombrecompañia
 - telefono
 - pais
 - compramedia
- Stored Procedures
- Functions
- world

Administration Schemas

Information: Table: **transaction**

Columns:

- id varchar(255) PK
- credit_card_id varchar(15)
- company_id varchar(20)
- user_id int
- lat float
- longitude float
- timestamp timestamp
- amount decimal(10,2)

```

92
93
94 SELECT *
95 FROM vistamarketing;
96
97
98
99

```

nombrecompañia	telefono	pais	compramedia
Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
Sed Id Limited	07 28 18 18 13	United States	477.51
Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10
Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
Non Magna LLC	06 71 73 13 17	United Kingdom	458.74

vistamarketing 13 x

Output

Action Output

#	Time	Action	Message
309	11:56:41	SELECT transaction.id as Idtransaccion, name as NombreUsuaría, sumame as Apel...	586 row(s) returned
310	11:57:29	Apply changes to Informe Tecnico	Changes applied
311	12:36:13	SELECT * FROM vistamarketing LIMIT 0, 50000	101 row(s) returned

Exercici 3

Filtra la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"

```
97
98
99 • SELECT *
100 FROM Vistamarketing
101 WHERE pais IN ("Germany");
102
103
104
```

nombrecompañia	telefono	pais	compramedia
Ac Industries	09 34 65 40 60	Germany	396.15
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.99
Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.57
Aliquam PC	01 45 73 52 16	Germany	280.34
Rutrum Non Inc.	02 66 31 61 09	Germany	266.90

Vistamarketing 10 x Read Or

Output

#	Time	Action	Message
✓ 27	11:22:38	SELECT company_name as nombrecompañia, phone as telefono, country as pais, ...	101 row(s) returned
✓ 28	11:24:04	Apply changes to vistamarketing	Changes applied
✓ 29	11:25:18	SELECT * FROM Vistamarketing WHERE pais IN ("Germany") LIMIT 0, 50000	8 row(s) returned

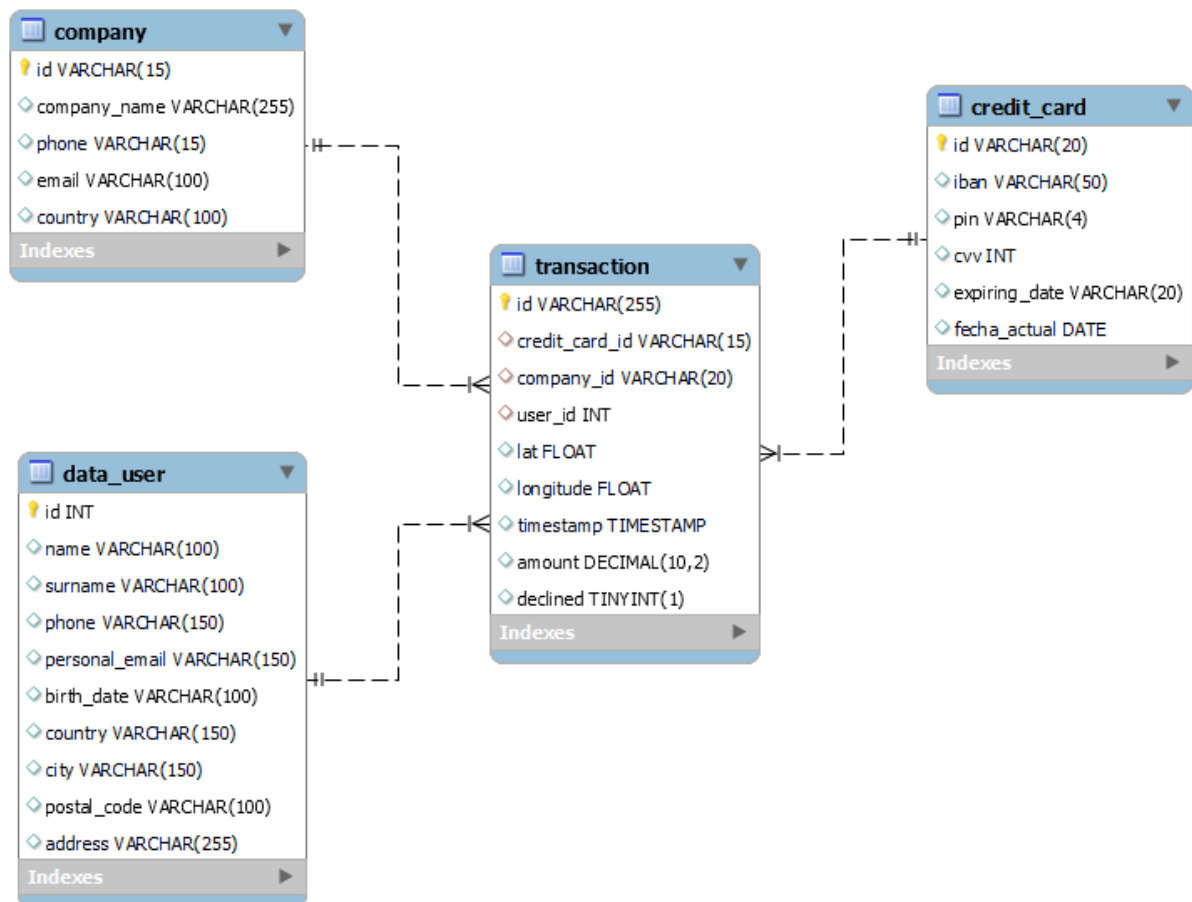
Desde la tabla creada anteriormente en vistamarketing, se filtran los resultados a través de la condición "Germany".



Nivell 3

Exercici 1

La setmana vinent tindràs una nova reunió amb els gerents de màrqueting. Un company del teu equip va realitzar modificacions en la base de dades, però no recorda com les va realitzar. Et demana que l'ajudis a deixar els comandos executats per a obtenir el següent diagrama:



(El diagrama que quiero obtener)

Primero agrego la tabla user con los registros proporcionados.

The screenshot shows a database management interface. On the left, the 'SCHEMAS' pane displays the 'user' table structure with columns: id, name, surname, phone, email, birth_date, country, city, postal_code, and address. The 'Columns' pane shows the data types for each column. The main area displays the 'Result Grid' with the following data:

	id	name	surname	phone	email	birth_date	country	city	postal_o
1	Zeus	Gamble		1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell	73544
2	Garrett	Mcconnell		(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59464
3	Claran	Harrison		(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbus	56518
4	Howard	Stafford		1-411-740-3269	ornare.egestas@icloud.edu	Feb 18, 1989	United States	Kailua	77417

Below the result grid, the 'Action Output' pane shows the following actions:

#	Time	Action	Message
306	11:29:53	INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, post...	1 row(s) affected
307	11:29:53	SET foreign_key_checks = 1	0 row(s) affected
308	11:30:47	SELECT * FROM user LIMIT 0, 50000	275 row(s) returned

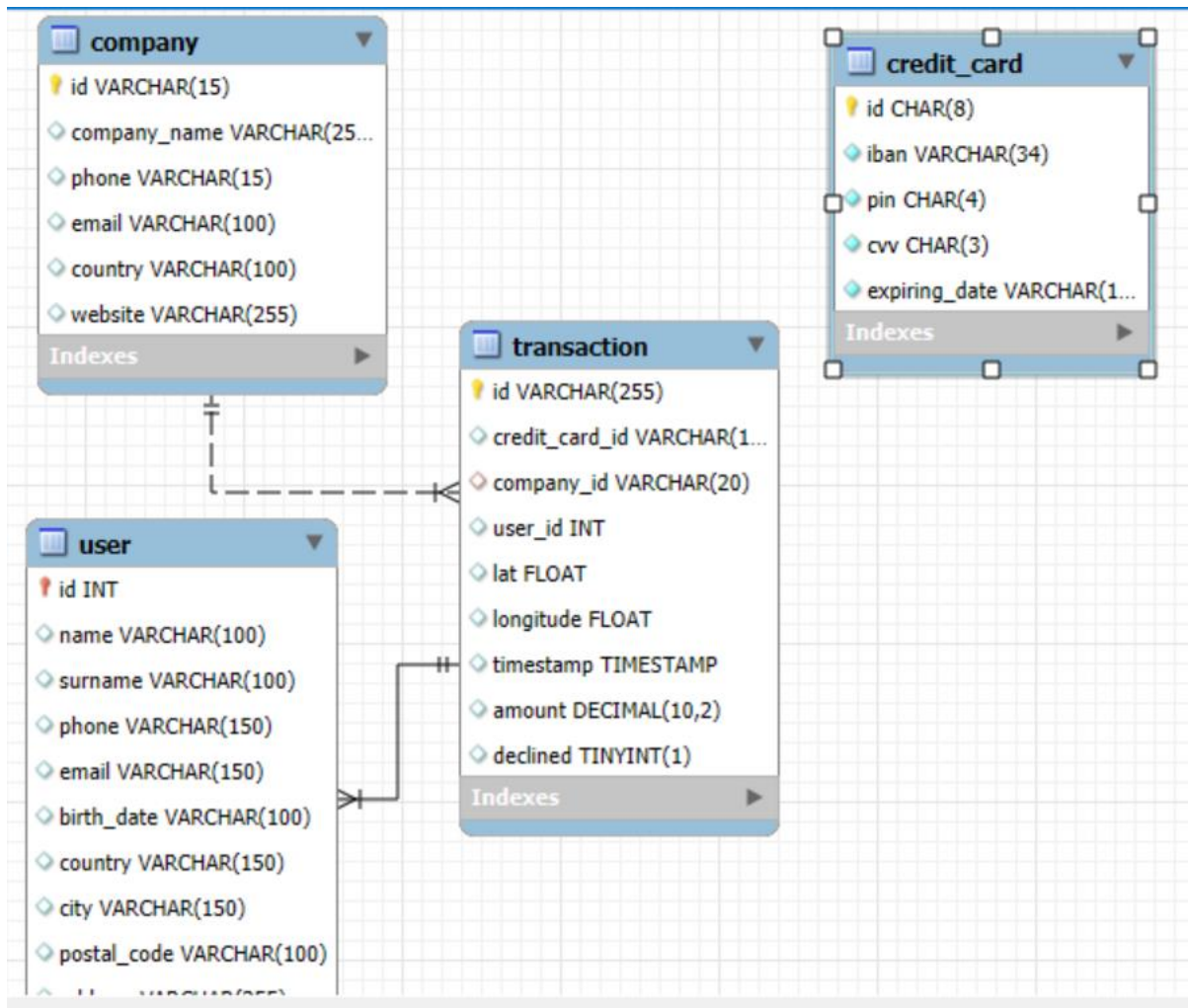
Luego visualizo el esquema a través de Reverse Engineer.

Entro a Database, Reverse Engineer y elijo la tabla en la que quiero administrar las conexiones:

The screenshot shows the 'Reverse Engineer Database' dialog box. The 'Select Schemas' section is active, displaying a list of schemas to choose from:

- ☐ biblioteca
- ☐ hospitales
- ☐ olympics
- ☐ sakila
- ☐ tienda_online
- ☒ transactions
- ☐ world

Llego a este diagrama:



Realizo cambios en la tabla de transaction:

Aqui se administran las conexiones de las Foreign Keys de la tabla de hechos transaction.

transaction - Table

Table Name: Schema: **transactions**

Foreign Key Name	Referenced Table	Column	Referenced Column
transaction_ibfk_1	'transactions', 'company'		
transaction_ibfk_2	'transactions', 'credit_card'		
transaction_ibfk_3	'transactions', 'data_user'		

Foreign Key Options

On Update:

On Delete:

☐ Skip in SQL generation

Foreign Key Comment:

Columns Indexes **Foreign Keys** Triggers Partitioning Options Inserts Privileges

Luego realizo cambios en la tabla de user:

Hay que cambiar el nombre de la tabla y campos. Se pueden editar directamente para que coincida con los nombres del ejercicio. (nombre data_user y personal_email)

data_user - Table

Table Name: Schema: **transactions**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
phone	VARCHAR(150)	<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
personal_email	VARCHAR(150)	<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
birth_date	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
country	VARCHAR(150)	<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

Columns | Indexes | Foreign Keys | Triggers | Partitioning | Options | Inserts | Privileges

Realizo cambios en la tabla de credit_card:

credit_card - Table

Table Name: Schema: **transactions**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
pin	VARCHAR(4)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
cvv	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
expiring_date	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fecha_actual	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: Data Type:

Charset/Collation:

Default:

Comments:

Storage: ☐ Virtual ☐ Stored

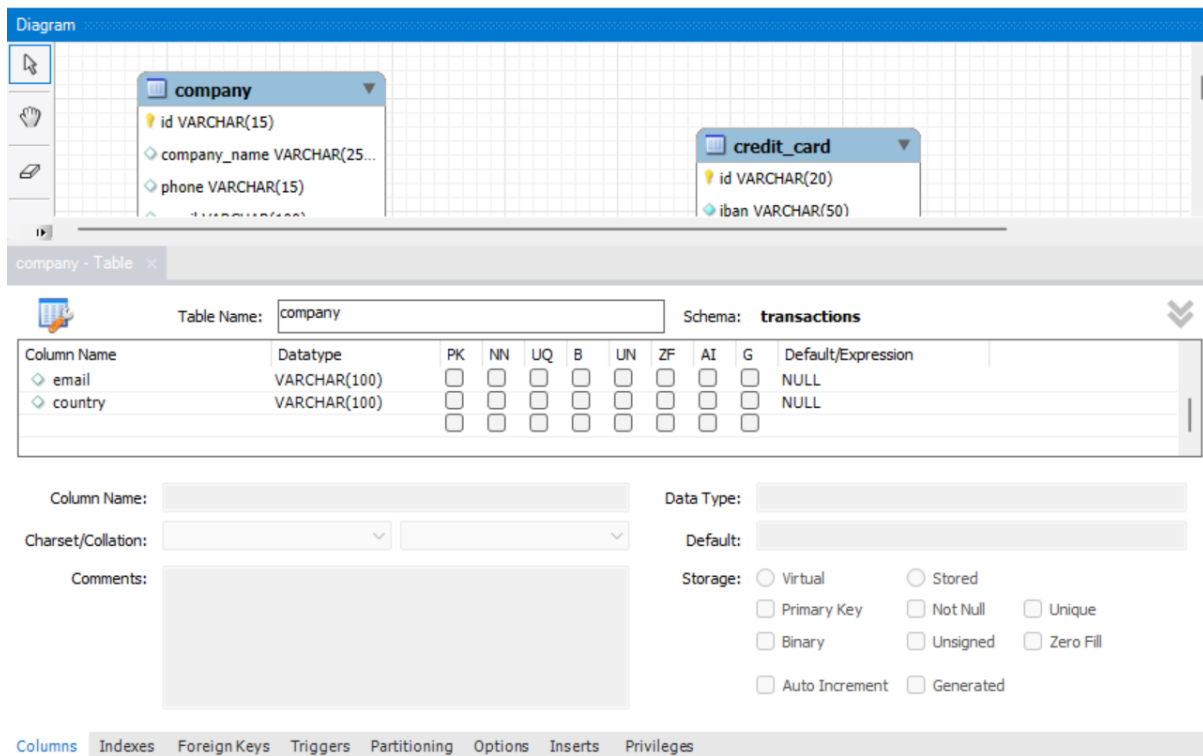
☐ Primary Key ☒ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

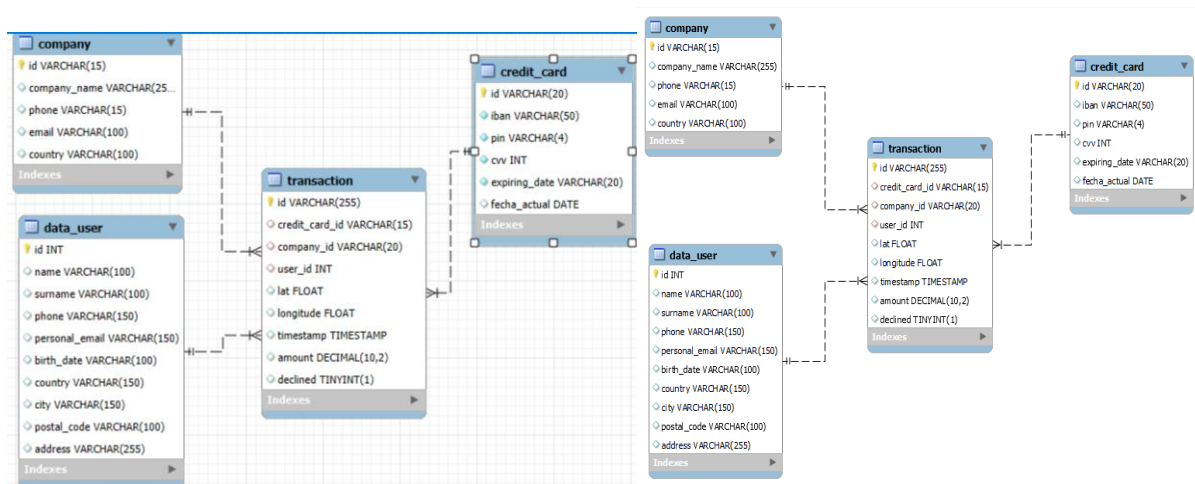
Cambio los Data_types en algunas columnas y agrego una columna de DATE.

Acomodo los datos en la tabla Company:

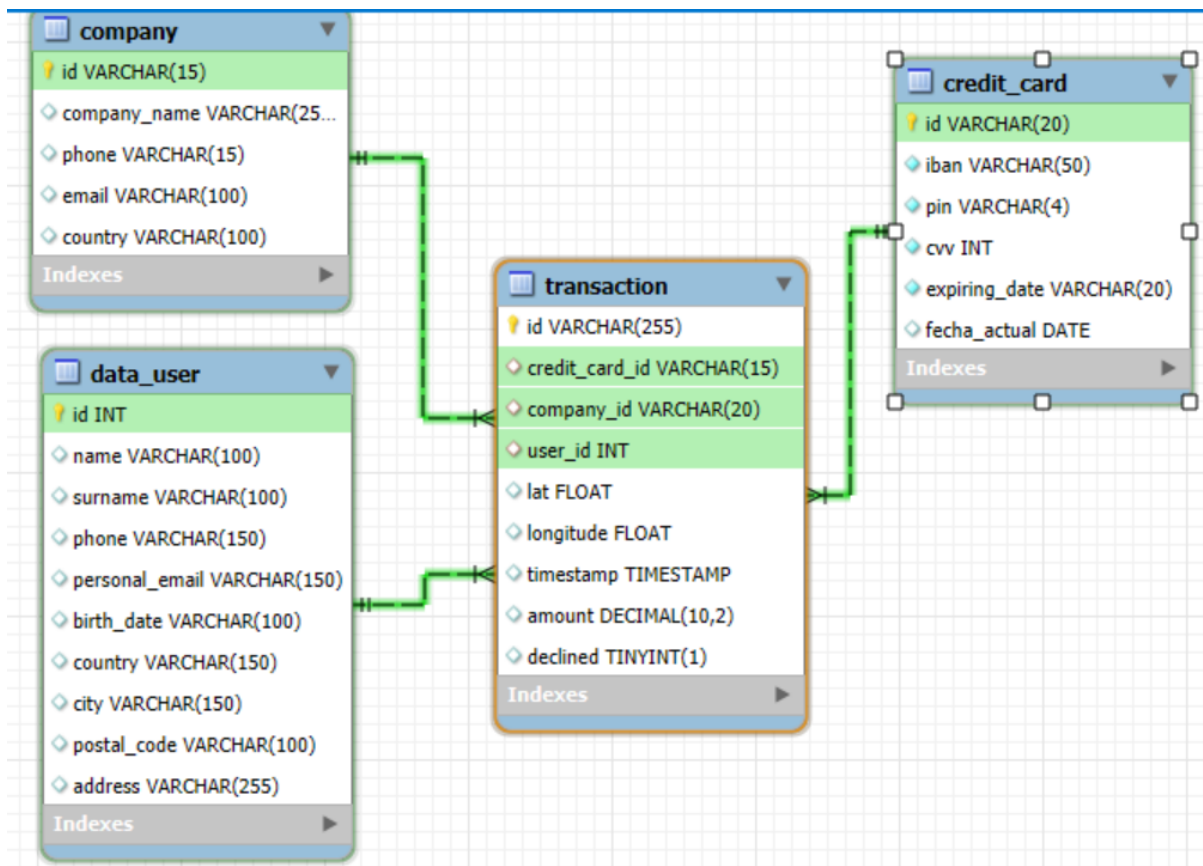


Borro la columna website.

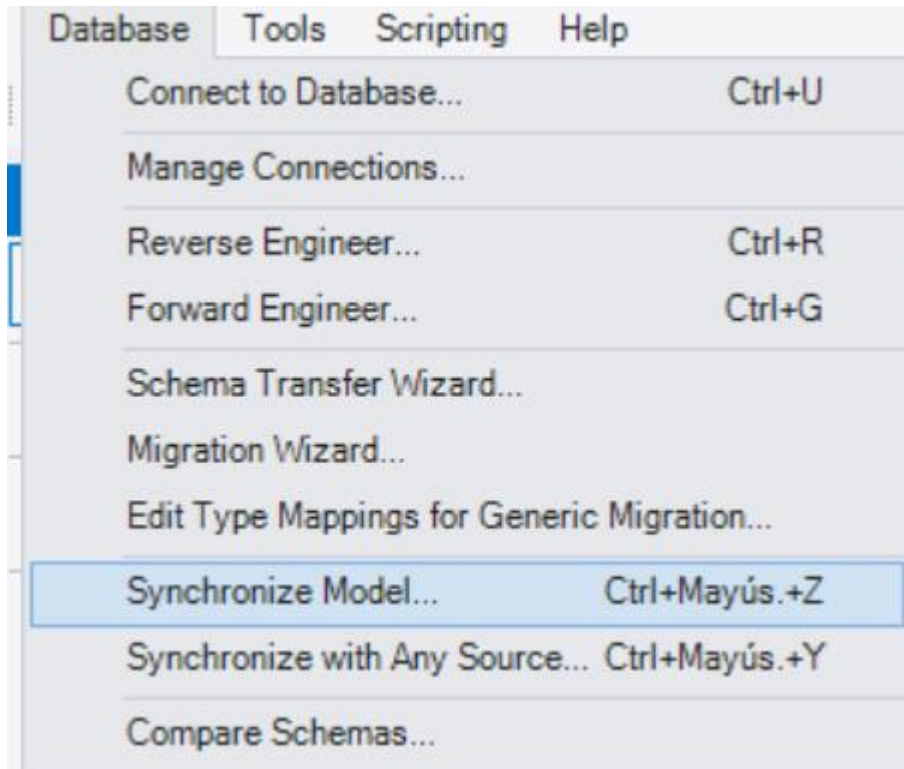
Obtengo este diagrama (Al lado la referencia):



El modelo demuestra estas relaciones con las PrimaryKeys:



Luego sincronizo el modelo de Reverse Engineer con el esquema



Me señala en que tablas se están realizando cambios:

Model and Database Differences

Double click arrows in the list to choose whether to ignore changes, update the model with database changes or vice-versa. You can also apply an action to multiple selected rows.

Model	Update	Source
▼ transactions	↔	transactions
company	→	company
credit_card	→	credit_card
transaction	→	transaction
data_user	→	user
informetecninco	↔	informetecninco
vistamarketing	↔	vistamarketing

Y me muestra el código que aplicara para realizar las consultas:

Preview Database Changes to be Applied

```

12  ALTER TABLE `transactions`.`company`
13  DROP COLUMN `website`;
14
15  ALTER TABLE `transactions`.`credit_card`
16  ADD COLUMN `fecha_actual` DATE NULL DEFAULT NULL AFTER `expiring_date`,
17  CHANGE COLUMN `id` `id` VARCHAR(20) NOT NULL ,
18  CHANGE COLUMN `iban` `iban` VARCHAR(50) NOT NULL ,
19  CHANGE COLUMN `pin` `pin` VARCHAR(4) NOT NULL ,
20  CHANGE COLUMN `cvv` `cvv` INT(11) NOT NULL ,
21  CHANGE COLUMN `expiring_date` `expiring_date` VARCHAR(20) NOT NULL ;
22
23  ALTER TABLE `transactions`.`transaction`
24  ADD INDEX `transaction_ibfk_2_idx` (`credit_card_id` ASC) VISIBLE,
25  ADD INDEX `transaction_ibfk_3_idx` (`user_id` ASC) VISIBLE;
26  ;
27
28  ALTER TABLE `transactions`.`user`
29  CHANGE COLUMN `email` `personal_email` VARCHAR(150) NULL DEFAULT NULL , RENAME TO `transactions`.`data_user` ;
30
31  ALTER TABLE `transactions`.`transaction`
32  ADD CONSTRAINT `transaction_ibfk_2`
33  FOREIGN KEY (`credit_card_id`)
34  REFERENCES `transactions`.`credit_card` (`id`)
35  ON DELETE NO ACTION
36  ON UPDATE NO ACTION,
37  ADD CONSTRAINT `transaction_ibfk_3`
38  FOREIGN KEY (`user_id`)
39  REFERENCES `transactions`.`data_user` (`id`)
40  ON DELETE NO ACTION
41  ON UPDATE NO ACTION;
42

```

The screenshot shows a database management interface with a left sidebar containing a tree view of schemas and tables. The main area displays a SQL script for creating a table named `transactions.credit_card`. The script includes column definitions for `id`, `iban`, `pan`, `pin`, `cvv`, and `expiring_date`, along with two check constraints: `chk_pin_format` and `chk_cvv_format`. Below the script, the 'Output' tab shows the execution results, including a message indicating that the table 'credit_card' already exists.

Table: credit_card

Columns:

id	varchar(20)
iban	PK
pin	varchar(50)
cvv	varchar(4)
expiring_date	int
fecha_actual	varchar(20)
	date

SQL Script:

```
CREATE TABLE transactions.credit_card
(id VARCHAR (15) PRIMARY KEY,
iban VARCHAR (34) NOT NULL,
pan VARCHAR (19) NOT NULL,
pin CHAR (4) NOT NULL,
cvv CHAR (3) NOT NULL,
expiring_date VARCHAR (10) NOT NULL,
CONSTRAINT chk_pin_format CHECK (pin REGEXP "^[0-9]{4}$"),
CONSTRAINT chk_cvv_format CHECK (cvv REGEXP "^[0-9]{3}$")
);
```

Output:

#	Time	Action	Message
329	13:53:05	Alter table credit_card drop pan	Error Code: 1091. Can't DROP 'pan'; check that column/k
330	13:53:09	SELECT * FROM credit_card LIMIT 0, 50000	275 row(s) returned
331	13:54:49	CREATE TABLE transactions.credit_card (id VARCHAR (15) PRIMARY KEY, ib...	Error Code: 1050. Table 'credit_card' already exists

Aquí puedo ver que se han aplicado los cambios a la tabla existente de `credit_card` teniendo en cuenta los datatypes indicados en este ejercicio.

Exercici 2

L'empresa també et sol·licita crear una vista anomenada "InformeTecnico" que contingui la següent informació:

- ID de la transacció
- Nom de l'usuari/ària
- Cognom de l'usuari/ària
- IBAN de la targeta de crèdit usada.
- Nom de la companyia de la transacció realitzada.
- Assegura't d'incloure informació rellevant de totes dues taules i utilitza àlies per a canviar de nom columnes segons sigui necessari.

Mostra els resultats de la vista, ordena els resultats de manera descendent en funció de la variable ID de transaction.

The screenshot shows a SQL IDE interface. At the top, a toolbar includes icons for file operations, search, and execution, along with a 'Limit to 50000 rows' dropdown. Below the toolbar, a SQL query is entered in a text area:

```
132 FROM transaction
133 JOIN company ON company_id = company.id
134 JOIN data_user ON user_id = data_user.id
135 JOIN credit_card ON credit_card_id = credit_card.id
136 ORDER BY Idtransaccion DESC;
137 • SELECT*
138 FROM informetecnico;
139
```

Below the query editor, the 'Result Grid' displays the results of the query. The grid has five columns: Idtransaccion, NombreUsuaría, Apellido, iban, and nombrecompañía. The first four rows of data are visible:

Idtransaccion	NombreUsuaría	Apellido	iban	nombrecompañía
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC

Below the result grid, the 'Output' pane shows the 'Action Output' for the query execution. It displays three rows of log messages:

#	Time	Action	Message
323	13:44:39	SELECT* FROM informetecnico LIMIT 0, 50000	586 row(s) returned
324	13:45:32	SELECT* FROM vistamarketing LIMIT 0, 50000	101 row(s) returned
325	13:46:55	SELECT* FROM informetecnico LIMIT 0, 50000	586 row(s) returned

Aquí se visualizan todos los datos que deben incluirse en el InformeTecnico, además de que se le asignan nombres más explicativos. Luego se crea una nueva Vista.