

Nivell 1 – Exercici 1

Creo la taula "credit_card" amb una FOREIGN KEY NOT NULL i assignant el tipus de dades a cada camp que correspon al tipus i llarg dels dades proporcionades a l'arxiu SQL.

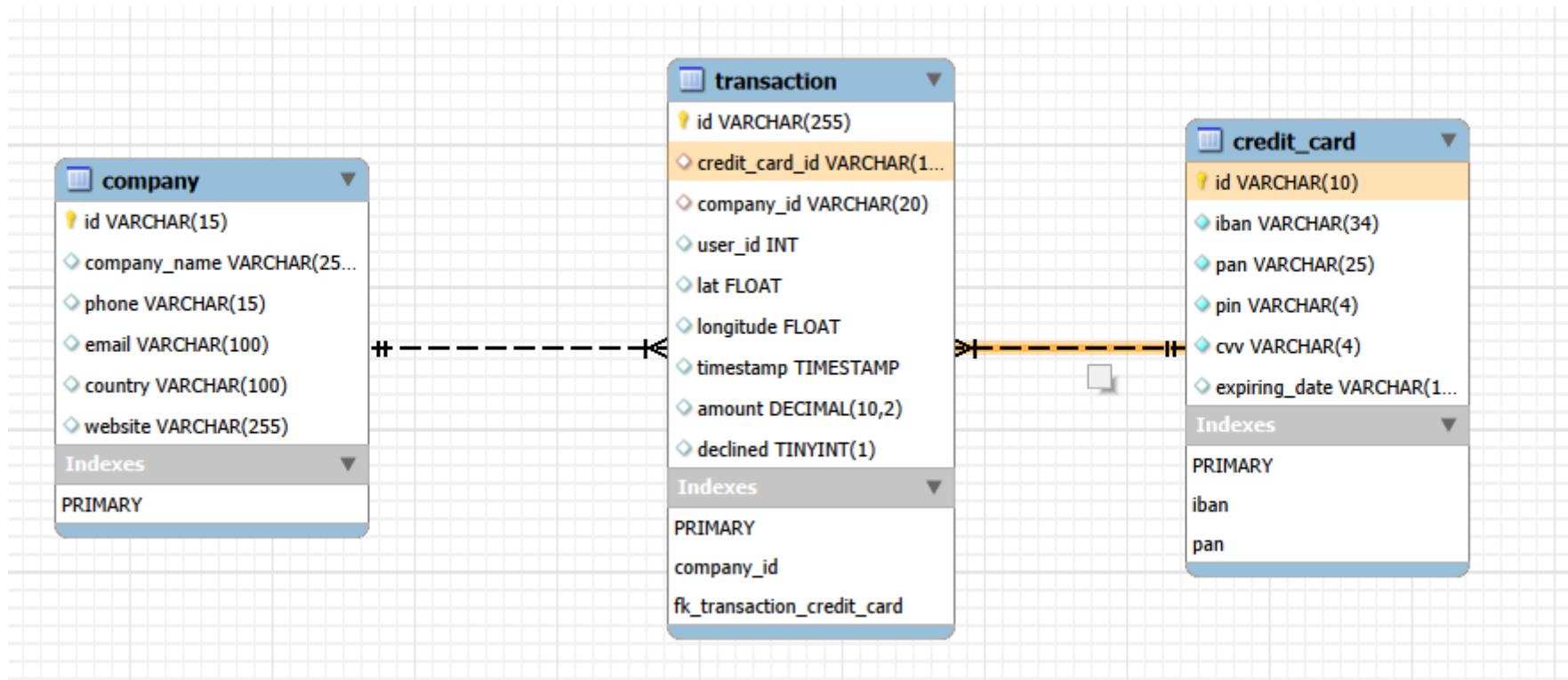
```
8 • CREATE TABLE IF NOT EXISTS credit_card (
9     id VARCHAR(10) NOT NULL PRIMARY KEY,
10    iban VARCHAR(34) UNIQUE,
11    pan VARCHAR(25) UNIQUE,
12    pin VARCHAR(4),
13    cvv VARCHAR(4),
14    expiring_date VARCHAR(10)
15 );
16
17
```

Output		
#	Time	Action
1	21:19:02	CREATE TABLE IF NOT EXISTS credit_card (id VARCHAR(10) NOT NULL PRIMARY KEY, iban VARCHAR(34) UNI... 0 row(s) affected

Creo la FOREIGN KEY a la taula "transaction" per vincular-la a la taula "credit_card" recentment creada.

```
41 • ALTER TABLE transaction
42   ADD CONSTRAINT fk_transaction_credit_card
43   FOREIGN KEY (credit_card_id)
44   REFERENCES credit_card(id)
45   ON DELETE RESTRICT
46   ON UPDATE RESTRICT;
```

Output		
#	Time	Action
1	20:31:12	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_credit_card FOREIGN KEY (credit_card_id) REFERENCES credit_card(id) ON DELETE RESTRICT ON UPDATE RESTRICT 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0



La relació cardinal es (1:N) entre “credit_card” i “transaction” i també entre “company” i “transaction” en ambdós casos a través de claus externes amb restriccions específiques per garantir la coherència i la fiabilitat de les dades entre les taules.

Introduixo les dades a la taula "credit_card"

```
5020 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9578', 'XX991539646456110567870254', '8999808823061411', '2872', '772', '07/29/27');
5021 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9579', 'XX296393091587170202131236', '9690060468678689', '8379', '134', '12/25/25');
5022 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9580', 'XX781258889851950806677358', '5541182364498931', '9273', '737', '03/27/29');
5023 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9581', 'XX915670516405388124398147', '2624305470167630', '4336', '926', '06/29/25');
```

Output :

Action Output			Message	Duration / Fetch
#	Time	Action		
4983	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9564', 'XX641179722562813208059896', '3101008088023097', '1822', '746', '03/26/29')	1 row(s) affe...	0.000 sec
4984	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9565', 'XX64260602192605745163758', '5522454509376387', '3053', '879', '04/29/27')	1 row(s) affe...	0.015 sec
4985	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9566', 'XX327046332926811278055630', '6161849983263025', '6889', '278', '06/27/25')	1 row(s) affe...	0.000 sec
4986	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9567', 'XX865381783872538603280064', '0745947204130257', '4955', '190', '03/28/25')	1 row(s) affe...	0.000 sec
4987	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9568', 'XX386538547822381431747659', '2296680186377148', '8326', '522', '05/27/26')	1 row(s) affe...	0.000 sec
4988	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9569', 'XX305272728964047897361563', '0734570924276022', '6083', '424', '06/27/25')	1 row(s) affe...	0.000 sec
4989	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9570', 'XX633942010258286181516573', '7233675965634598', '5619', '285', '03/27/28')	1 row(s) affe...	0.016 sec
4990	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9571', 'XX484915382437141996579468', '1265066247957405', '6585', '104', '11/27/28')	1 row(s) affe...	0.000 sec
4991	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9572', 'XX405448186269730388404508', '8966622840225053', '1995', '220', '07/27/29')	1 row(s) affe...	0.000 sec
4992	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9573', 'XX965338931053088901924906', '9099040179637171', '2277', '224', '01/25/28')	1 row(s) affe...	0.015 sec
4993	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9574', 'XX662761758361432686520775', '3171178331318656', '5084', '706', '11/28/27')	1 row(s) affe...	0.000 sec
4994	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9575', 'XX499881216073626571196473', '8086033007377786', '7256', '660', '10/25/29')	1 row(s) affe...	0.000 sec
4995	20:54:00	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9576', 'XX52971093058771264172007', '4554226978064107', '4884', '724', '06/28/29')	1 row(s) affe...	0.000 sec
4996	20:54:01	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9577', 'XX158914078594808633147086', '8110437271856107', '5864', '772', '10/29/26')	1 row(s) affe...	0.016 sec
4997	20:54:01	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9578', 'XX991539646456110567870254', '8999808823061411', '2872', '772', '07/29/27')	1 row(s) affe...	0.000 sec
4998	20:54:01	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9579', 'XX296393091587170202131236', '9690060468678689', '8379', '134', '12/25/25')	1 row(s) affe...	0.000 sec
4999	20:54:01	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9580', 'XX781258889851950806677358', '5541182364498931', '9273', '737', '03/27/29')	1 row(s) affe...	0.000 sec
5000	20:54:01	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcS-9581', 'XX915670516405388124398147', '2624305470167630', '4336', '926', '06/29/25')	1 row(s) affe...	0.000 sec

Nivell 1 – Exercici 2

Només s'utilitza la taula "credit_card", de manera que no cal fer referència a la taula a l'instrucció WHERE.

```
5030 • UPDATE credit_card
5031   SET iban = 'TR323456312213576817699999'
5032   WHERE id = 'CcU-2938';
5033
5034 • SELECT *
5035   FROM credit_card
5036   WHERE id = 'CcU-2938';
5037
```

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

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR323456312213576817699999	5424465566813633	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL	NULL

credit_card 2 x Apply Revert

Output

#	Time	Action	Message	Duration / Fetch
1	21:08:02	UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE id = 'CcU-2938'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.015 sec
2	21:09:41	SELECT * FROM credit_card WHERE id = 'CcU-2938'	1 row(s) returned	0.000 sec / 0.000 sec

Result Grid
Form Editor

Nivell 1 – Exercici 3

Primer, he de crear el nou registre de targeta de crèdit, ja que no existeix a la taula de “credit_card” i que la FOREIGN KEY de la taula “company” és una FOREIGN KEY a la taula “transaction”. Aquest registre és necessari per fer la modificació sol·licitada a la taula “transaction” i mantenir la coherència de les dades entre les taules.

Vaig introduir un valor NULL per a tots els camps per als quals no tinc dades i també per tal que es puguin identificar aquests valors que falten. Una bona pràctica seria demanar a la persona que sol·licita aquesta modificació de dades que proporcioni tots aquests valors per garantir la integritat de les dades de la base de dades.

```
047 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-9999', NULL , NULL , NULL, NULL, NULL);
048 • SELECT *
049   FROM credit_card
050 WHERE id = 'CcU-9999';
051
```

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

- SQL Editor:** Contains the executed SQL code from line 047 to 051.
- Result Grid:** Shows the result of the SELECT query. It has a header row with columns: id, iban, pan, pin, cvv, and expiring_date. Below it, there are two data rows:
 - The first row corresponds to the inserted record with ID 'CcU-9999' and all other fields set to NULL.
 - The second row is a blank row with all fields set to NULL.
- Action Output:** Displays the log of actions taken:
 - Action 1: INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-9999', NULL , NULL , NULL, NULL, NULL); - Message: 1 row(s) affected - Duration / Fetch: 0.000 sec / 0.000 sec
 - Action 2: SELECT * FROM credit_card WHERE id = 'CcU-9999'; - Message: 1 row(s) returned - Duration / Fetch: 0.000 sec / 0.000 sec

Només s'utilitza la taula "credit_card", de manera que no cal fer referència a la taula a l'instrucció WHERE.

Segon, i pel mateix motiu que per a la taula "credit_card", he de crear la nova companya a la taula "company".

```
052 • INSERT INTO company (id, company_name, phone, email, country, website) VALUES ('b-9999', NULL , NULL , NULL, NULL, NULL);
053 • SELECT *
054   FROM company
055 WHERE id = 'b-9999';
056
```

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window containing the provided SQL statements. Below it is a 'Result Grid' window displaying the data from the 'company' table. The grid has columns: id, company_name, phone, email, country, and website. One row is visible with id 'b-9999' and all other fields as NULL. To the right of the result grid is a vertical toolbar with icons for Result Grid, Form Editor, and other options. Below the result grid is an 'Output' window showing the log of actions taken: an 'INSERT' statement was executed at 21:29:03, affecting 1 row(s), and a 'SELECT *' statement was executed at 21:29:20, returning 1 row(s). The log also includes 'Duration / Fetch' times: 0.015 sec for the insert and 0.000 sec / 0.000 sec for the select.

Només s'utilitza la taula "company", de manera que no cal fer referència a la taula a l'instrucció WHERE.

Ara puc insertar la transacció sol·licitada. He posat NULL per a l'hora de la transacció pels mateixos motius que els esmentats anteriorment.

```
057 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('10881D1D-5B23-A76C-55E1-  
058 • SELECT *  
059 FROM transaction  
060 WHERE company_id = 'b-9999';
```

The screenshot shows the MySQL Workbench interface with the 'transaction' table selected. The table has the following columns: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, and declined. A single row is present in the table:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	
10881D1D-5B23-A76C-55E1-C568E49A99DD	NULL	CcU-9999	b-9999	9999	829.999	-117.999	NULL	111.11	0

Below the table, the 'Output' tab displays the execution history:

#	Time	Action	Message	Duration / Fetch
1	21:35:33	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined) VALUES ('10881D1D-5B23-A76C-55E1-C568E49A99DD', NULL, 'CcU-9999', 'b-9999', 9999, 829.999, -117.999, 111.11, 0)	1 row(s) affected	0.016 sec
2	21:35:37	SELECT * FROM transaction WHERE company_id = 'b-9999'	1 row(s) returned	0.000 sec / 0.000 sec

Només s'utilitza la taula "transaction", de manera que no cal fer referència a la taula a l'instrucció WHERE.

Nivell 1 – Exercici 4

```
068 • ALTER TABLE credit_card
069   DROP COLUMN pan;
070
071 • SELECT *
072   FROM credit_card;
```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons for filtering, editing, and exporting data. Below the toolbar is a result grid displaying the contents of the 'credit_card' table. The table has columns: id, iban, pin, cvv, and expiring_date. There are 8 rows of data. On the right side of the interface, there is a sidebar with three tabs: 'Result Grid' (selected), 'Form Editor', and 'Revert'. Below the result grid, there is an 'Output' section which displays the execution history of the two queries. The first query, 'ALTER TABLE credit_card DROP COLUMN pan;', was run at 21:40:59 and affected 0 rows. The second query, 'SELECT * FROM credit_card', was run at 21:41:11 and returned 5001 rows. The duration for both operations is listed as 0.000 sec / 0.016 sec.

id	iban	pin	cvv	expiring_date
CcS-4857	XX4857591835292505850771	1819	467	09/27/25
CcS-4858	XX8581768137002436094025	3964	817	12/28/28
CcS-4859	XX7826930491423553609370	4983	277	11/26/26
CcS-4860	XX5559590368835304645299	6876	661	07/27/27
CcS-4861	XX2035182877195191627307	5710	398	04/25/26
CcS-4862	XX4774721462463645409758	4042	174	11/27/26
CcS-4863	XX1476829664245046207111	5969	449	12/27/29

credit_card 8

Output

#	Time	Action	Message	Duration / Fetch
1	21:40:59	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.188 sec
2	21:41:11	SELECT * FROM credit_card	5001 row(s) returned	0.000 sec / 0.016 sec

Nivell 2 – Exercici 1

Aquí teniu la transacció objectiu .

```
080 •  SELECT *
081   FROM transaction
082 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
083
```

The screenshot shows a database query results window. At the top, there is a code editor with the following SQL query:

```
080 •  SELECT *
081   FROM transaction
082 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
083
```

Below the code editor is a result grid table with the following data:

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	CcS-5019	b-2370	438	41.5972	12.2218	2016-12-21 20:07:18	155.63	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

On the right side of the window, there is a sidebar with icons for "Result Grid" and "Form Editor". Below the table, the output pane shows the query and its execution details:

transaction 21 x

Output

Action Output

#	Time	Action
1	22:13:02	SELECT * FROM transaction WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD'

Message: 1 row(s) returned

Duration / Fetch: 0.000 sec / 0.000 sec

Només s'utilitza la taula "transaction", de manera que no cal fer referència a la taula a l'instrucció WHERE.

Aquí teniu l'estat de la taula amb la transacció suprimida segons la consulta executada a la fila 2 de la sortida:

```
080 •  SELECT *
081   FROM transaction
082 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
083
```

The screenshot shows a database management interface with a Result Grid and an Action Output history.

Result Grid: A table with columns: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined. The table contains one row with all values set to NULL.

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

Action Output: A history of database actions.

#	Time	Action	Message	Duration / Fetch
1	22:13:02	SELECT * FROM transaction WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD'	1 row(s) returned	0.000 sec / 0.000 sec
2	22:14:42	DELETE FROM transaction WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD'	1 row(s) affected	0.000 sec
3	22:15:05	SELECT * FROM transaction WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD'	0 row(s) returned	0.000 sec / 0.000 sec

Nivell 2 – Exercici 2

Amb la primera consulta, creo la vista. Amb la segona consulta, l'anomeno per obtenir l'informe. L'estrucció de "VistaMarketing" es pot veure a la part inferior esquerra de la captura de pantalla.

The screenshot shows the MySQL Workbench interface. On the left, the database schema is visible, including the 'transactions' schema which contains the 'vistamarketing' view. The central pane displays the SQL code for creating the view:

```
95 • CREATE VIEW VistaMarketing AS
96   SELECT c.id AS id_companya, c.company_name AS companya, c.phone AS telefon, c.country AS pais, ROUND(AVG(t.amount),2) AS media_vendes
97   FROM company AS c
98   JOIN transaction AS t
99   ON c.id = t.company_id
100  WHERE t.declined = 0
101  GROUP BY id_companya, companya, telefon, pais
102  ORDER BY media_vendes DESC;
103
104 • SELECT *
105   FROM VistaMarketing;
```

Below the code, the 'Result Grid' shows the data returned by the query:

	id_companya	companya	telefon	pais	media_vendes
▶	b-2222	Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.91
	b-2282	Pretium Neque Corp.	07 77 48 55 28	Australia	275.58
	b-2422	Uma Convallis Associates	06 01 24 77 04	United States	273.57
	b-2538	At Associates	09 56 61 10 65	New Zealand	272.74
	b-2498	Metus Vitae Associates	08 25 44 40 66	Australia	270.05
	b-2570	Aliquet Diam Limited	02 76 61 47 46	United States	269.29
	b-2470	Nec Luctus LLC	02 14 71 75 73	Norway	268.60
	b-2382	Neque Telus Incorporated	04 43 18 34 19	Ireland	267.56
	b-2514	Gras Consulting	07 50 10 85 63	Belgium	267.38
	b-2274	Sed LLC	01 63 16 26 52	Belgium	266.61
	b-2398	Eget Ipsum Ltd	03 67 44 56 72	United States	266.27
	b-2394	Tortor Nunc Commodo Com...	05 35 92 77 16	United States	266.26
	b-2334	Amet Institute	06 33 40 21 33	Australia	265.54

The bottom pane shows the 'Output' window with the log of actions:

#	Time	Action	Message	Duration / Fetch
1	11:25:13	CREATE VIEW VistaMarketing AS SELECT c.id AS id_companya, c.company_name AS companya, c.phone AS telefon, c.country AS pais, ROUND(AVG(t.amount),2) AS media_vendes	0 row(s) affected	0.000 sec
2	11:25:17	SELECT * FROM VistaMarketing	101 row(s) returned	0.640 sec / 0.000 sec

Nivell 2 – Exercici 3

En aquest cas, simplement filtro la vista al país desitjat. Si l'equip de màrqueting necessita aquest informe de manera regular/freqüent, generaria aquest informe en particular com una vista específica o actualitzaria el de VistaMarketing perquè només retorna les empreses alemanyes.

Només s'utilitza la vista "VistaMarketing", de manera que no cal fer referència a la vista a l'instrucció WHERE.

The screenshot shows a database interface with a code editor at the top and a result grid below it. The code editor contains the following SQL query:

```
5116 • SELECT *
5117   FROM VistaMarketing
5118 WHERE pais = 'Germany';
5119
```

The result grid displays a table with four columns: companya, telefon, pais, and media_vendes. The data is as follows:

companya	telefon	pais	media_vendes
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.91
Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.32
Convallis In Incorporated	06 66 57 29 50	Germany	257.69
Ac Industries	09 34 65 40 60	Germany	255.17
Rubrum Non Inc.	02 66 31 61 09	Germany	255.14
Auctor Mauris Corp.	05 62 87 14 41	Germany	254.68
Augue Foundation	06 88 43 15 63	Germany	253.56
Aliquam PC	01 45 73 52 16	Germany	252.96

On the right side of the interface, there is a sidebar with icons for Result Grid, Form Editor, and Field Types. Below the result grid, there is an output section titled "Action Output" showing the execution history of the query:

#	Time	Action	Message	Duration / Fetch
1	22:33:32	CREATE VIEW VistaMarketing AS SELECT c.company_name AS companya, c.phone AS telefon, c.country AS pais, ROUND(AVG(t.amount),2) AS m...	0 row(s) affected	0.016 sec
2	22:34:51	SELECT * FROM VistaMarketing	101 row(s) returned	0.657 sec / 0.000 sec
3	22:37:16	SELECT * FROM VistaMarketing WHERE pais = 'Germany'	8 row(s) returned	0.047 sec / 0.000 sec

Nivell 3 – Exercici 1

Les taules "credit_card", "company" i "data_user" tenen una relació d'un a molts amb la taula "transaction". Per tant, primer revisaré les taules "credit_card" i "company" i crearé la de "data_user" abans de treballar amb la taula "transaction".

Primer, el company ha modificat la taula "credit_card" en comparació amb com la vaig configurar a l'Exercici 1, Nivell 1:

- El camp "id" s'ha ampliat de VARCHAR(10) a VARCHAR(20).
- El camp "iban" s'ha ampliat de VARCHAR(34) a VARCHAR(50)
- El tipus de dades del camp "cvv" s'ha modificat de VARCHAR a INT
- La longitud del camp "expiring_date" s'ha modificat de VARCHAR(10) a VARCHAR(20), suposo que ja que la captura de pantalla està truncada i, per tant, no es pot llegir.
- El camp "fecha_actual" s'ha afegit amb un tipus de dades DATE. Suposo que això era per operar la conversió del camp anterior a un que es pogués utilitzar a partir de la comparació des d'una perspectiva de tipus de dades de data. Això podria ser clarament una acció útil. Tanmateix, comentaria que aquest camp utilitzava l'idioma espanyol, cosa que no és coherent amb la resta de la base de dades que utilitza l'idioma anglès. Per tant, aquest nom es beneficiaria de ser modificat per un d'anglès com ara "data_real".

Captura de pantalla abans del canvi:

The screenshot shows the MySQL Workbench interface. In the top-left pane, the 'credit_card' table structure is displayed with columns: id (varchar(10) PK), iban (varchar(34)), pin (varchar(4)), cvv (varchar(4)), and expiring_date (varchar(10)). Below the table structure, a query editor window is open. The code in the editor is as follows:

```
Table: credit_card
Columns:
id      varchar(10) PK
iban    varchar(34)
pin     varchar(4)
cvv     varchar(4)
expiring_date  varchar(10)

5118 WHERE pais = 'Germany';
5119
5120
5121
5122
5123
5124 -- **** Nivell 3 ****
5125
5126
5127 -- **** Exercici 1 ****
5128
5129 • USE transactions;
5130 • ALTER TABLE credit_card
5131 CHANGE COLUMN id id VARCHAR(20);
```

In the bottom pane, there is an 'Output' tab showing the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
1	11:50:42	USE transactions	0 row(s) affected	0.000 sec
2	11:50:54	ALTER TABLE credit_card CHANGE COLUMN id id VARCHAR(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.032 sec

Captura de pantalla després del canvi:

The screenshot shows the MySQL Workbench interface. In the top left, the connection is set to 'world'. The 'Schemas' tab is selected. Below it, the 'Table: credit_card' section shows the table structure with columns: id (PK), iban, pin, cvv, expiring_date, and fecha_actual. The 'Columns' table lists these columns with their current types. The main area displays a series of SQL statements (5128 to 5146) being executed. The statements include changing the type of 'iban' to VARCHAR(50), changing 'cvv' to INT, changing 'expiring_date' to VARCHAR(20), adding a new column 'fecha_actual' of type DATE NULL, and finally updating the 'fecha_actual' column to match the format of 'expiring_date' using the STR_TO_DATE function. The 'Output' pane at the bottom shows the results of each statement, indicating 0 rows affected for most, except for the update which affected 5001 rows.

```
5128 • USE transactions;
5129 • ALTER TABLE credit_card
5130 • CHANGE COLUMN id id VARCHAR(20);
5131
5132
5133 • ALTER TABLE credit_card
5134 • CHANGE COLUMN iban iban VARCHAR(50);
5135
5136 • ALTER TABLE credit_card
5137 • CHANGE COLUMN cvv cvv INT;
5138
5139 • ALTER TABLE credit_card
5140 • CHANGE COLUMN expiring_date expiring_date VARCHAR(20);
5141
5142 • ALTER TABLE credit_card
5143 • ADD COLUMN fecha_actual DATE NULL;
5144
5145 • UPDATE credit_card
5146 SET fecha_actual = STR_TO_DATE(expiring_date, '%m/%d/%y');
```

#	Time	Action	Message	Duration / Fetch
3	11:52:06	ALTER TABLE credit_card CHANGE COLUMN iban iban VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
4	11:53:50	ALTER TABLE credit_card CHANGE COLUMN cvv cvv INT	5001 row(s) affected Records: 5001 Duplicates: 0 Warnings: 0	0.203 sec
5	11:54:41	ALTER TABLE credit_card CHANGE COLUMN expiring_date expiring_date VARCHAR(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
6	12:16:45	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec

Calia desactivar una seguretat de MySQL relacionada amb l'ús de l'UPDATE sense aplicar cap condició WHERE per dur a terme aquesta transformació de dades a la columna "fecha_actual". Vaig desactivar la seguretat per processar les 5000 files i després la vaig reactivar.

This screenshot shows the same MySQL Workbench interface. The 'Output' pane at the bottom displays the execution of an UPDATE statement. The statement updates the 'credit_card' table by setting 'fecha_actual' to the value of 'expiring_date' using the STR_TO_DATE function. The log shows one row affected, which matches the number of rows matched (5001). The duration of the operation is 0.360 seconds.

```
5145 • UPDATE credit_card
5146 SET fecha_actual = STR_TO_DATE(expiring_date, '%m/%d/%y');
```

#	Time	Action	Message	Duration / Fetch
1	12:25:55	UPDATE credit_card SET fecha_actual = STR_TO_DATE(expiring_date, "%m/%d/%y")	5000 row(s) affected Rows matched: 5001 Changed: 5000 Warnings: 0	0.360 sec

En segon lloc, el company va modificar la taula "company" en comparació amb com estava configurada a l'Sprint 2. El camp " website" es va eliminar.

Captura de pantalla abans del canvi:

The screenshot shows the MySQL Workbench interface. On the left, there's a tree view with 'Information Schema' expanded, showing 'Tables' and 'Schemas'. Under 'Tables', 'company' is selected. The main pane displays the table structure:

```
Table: company
Columns:
id          varchar(15) PK
company_name varchar(255)
phone        varchar(15)
email        varchar(100)
country      varchar(100)
website      varchar(255)
```

Below the table structure, the SQL editor shows the following code:

```
5151 -- modificacions a la taula "company"
5152
5153 • ALTER TABLE company
5154   DROP COLUMN website;
5155
```

Captura de pantalla després del canvi:

The screenshot shows the MySQL Workbench interface after the 'website' column was dropped. The table structure remains the same as before:

```
Table: company
Columns:
id          varchar(15) PK
company_name varchar(255)
phone        varchar(15)
email        varchar(100)
country      varchar(100)
```

However, the SQL editor now shows the executed command:

```
5153 • ALTER TABLE company
5154   DROP COLUMN website;
5155
```

In the 'Output' pane, the log entry for the executed command is shown:

#	Time	Action
1	12:30:42	ALTER TABLE company DROP COLUMN website

Details for the log entry:

- Message: 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
- Duration / Fetch: 0.047 sec

En tercer lloc, cal crear la taula "data_user" i importar les dades relacionades. Primer l'importo com a "usuari" segons l'estructura de taula i les dades de la taula proporcionades per evitar més complicacions.

The screenshot shows the MySQL Workbench interface. On the left, the database structure is visible under the 'Transactions' schema, including tables like 'company', 'credit_card', 'transaction', and 'user'. The 'user' table is selected. In the main pane, the SQL editor contains the following code:

```

5145 ADD COLUMN fecha_actual DATE NULL;
5146
5147 • UPDATE credit_card
5148 SET fecha_actual = STR_TO_DATE(expiring_date, '%m/%d/%y');
5149
5150
5151 -- modificacions a la taula "company"
5152
5153 • ALTER TABLE company
5154 DROP COLUMN website;
5155
5156 -- importacio de la taula "user"
5157
5158 • CREATE TABLE IF NOT EXISTS user (
5159     id CHAR(10) PRIMARY KEY,
5160     name VARCHAR(100),
5161     surname VARCHAR(100),
5162     phone VARCHAR(150),
5163     email VARCHAR(150),
5164     birth_date VARCHAR(100),
5165     country VARCHAR(150),
5166     city VARCHAR(150),
5167     postal_code VARCHAR(100),
5168     address VARCHAR(255)
5169 );
5170

```

The 'Output' tab shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	12:45:15	CREATE TABLE IF NOT EXISTS user (id CHAR(10) PRIMARY KEY, name VARCHAR(100), surname VARCHAR(100), phone VARCHAR(150), email VARCHAR(150), birth_date VARCHAR(100), country VARCHAR(150), city VARCHAR(150), postal_code VARCHAR(100), address VARCHAR(255))	0 row(s) affected	0.031 sec

The screenshot shows the MySQL Workbench interface with the 'user' table selected. The 'Data' tab displays the imported data:

country, city, postal_code, address	VALUES ("4939", "Nzultp", "Lylyewdk", "+94-736-5751", "nzultp.lylyewdk@example.com", "Mar 5, 1999", "United States", "Philadelphia", "19101", "477 Lylyewdk St");
L0167 •	VALUES ("4945", "Pimdet", "Zvagjirk", "+91-727-8072", "pimdet.zvagjirk@example.com", "Jul 1, 1969", "Canada", "Toronto", "M5A 1A1", "45 Zvagjirk St");
L0168 •	VALUES ("4947", "Upyafn", "Tfiiwpnz", "+95-878-8855", "upyafn.tfiiwpnz@example.com", "Aug 1, 1985", "United States", "Los Angeles", "90001", "360 Tfiiwpnz St");
L0169 •	VALUES ("4950", "Akxgbk", "Fxtrblub", "+96-670-3225", "akxgbk.fxtrblub@example.com", "Jan 3, 1984", "Canada", "Vancouver", "V5K 0A1", "19 Fxtrblub St");
L0170 •	VALUES ("4952", "Fufhmv", "Sbgptpx", "+61-136-1383", "fufhmv.sbgptpx@example.com", "Jul 22, 1960", "Canada", "Montreal", "H1A 0A1", "570 Sbgptpx St");
L0171 •	VALUES ("4961", "Phwfif", "Myxbvzkf", "+34-594-1874", "phwfif.myxbvzkf@example.com", "Nov 2, 1972", "United States", "New York", "10001", "790 Myxbvzkf St");
L0172 •	VALUES ("4977", "Gjldyg", "Egqqecse", "+62-591-5059", "gjldyg.egqqecse@example.com", "Aug 16, 1986", "United States", "Chicago", "60601", "537 Egqqecse St");
L0173 •	VALUES ("4980", "Asgaxi", "Btardzti", "+39-155-6819", "asgaxi.btardzti@example.com", "Jul 24, 1999", "Canada", "Winnipeg", "R2C 0A1", "394 Btardzti St");
L0174 •	VALUES ("4983", "Eremdc", "Orekppbr", "+59-545-4710", "eremd.c.orekppbr@example.com", "Apr 5, 1953", "Canada", "Winnipeg", "R2C 0A1", "635 Orekppbr St");
L0175 •	VALUES ("4999", "Omjnoy", "Rmuqvxgw", "+32-354-7682", "omjnoy.rmuqvxgw@example.com", "Aug 26, 1996", "United States", "San Diego", "92101", "437 Rmuqvxgw St");
L0176 •)	

The 'Output' tab shows the execution results for each insert statement:

#	Time	Action	Message	Duration / Fetch
4997	12:46:51	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("4961", "Phwfif", "Myxbvzkf", "+34-594-1874", "phwfif.myxbvzkf@example.com", "Nov 2, 1972", "United States", "New York", "10001", "790 Myxbvzkf St") 1 row(s) affected	0.016 sec
4998	12:46:51	VALUES ("4977", "Gjldyg", "Egqqecse", "+62-591-5059", "gjldyg.egqqecse@example.com", "Aug 16, 1986", "United States", "Chicago", "60601", "537 Egqqecse St") 1 row(s) affected	0.000 sec
4999	12:46:51	VALUES ("4980", "Asgaxi", "Btardzti", "+39-155-6819", "asgaxi.btardzti@example.com", "Jul 24, 1999", "Canada", "Winnipeg", "R2C 0A1", "394 Btardzti St") 1 row(s) affected	0.000 sec
5000	12:46:51	VALUES ("4983", "Eremdc", "Orekppbr", "+59-545-4710", "eremd.c.orekppbr@example.com", "Apr 5, 1953", "Canada", "Winnipeg", "R2C 0A1", "635 Orekppbr St") 1 row(s) affected	0.016 sec
5001	12:46:51)	"4999", "Omjnoy", "Rmuqvxgw", "+32-354-7682", "omjnoy.rmuqvxgw@example.com", "Aug 26, 1996", "United States", "San Diego", "92101", "437 Rmuqvxgw St") 1 row(s) affected	0.000 sec

A continuació, cal fer les modificacions següents:

- El camp "id" s'ha modificat de CHAR(10) inclòs al fitxer d'estructura de la taula a un tipus de dades INT

- El camp "email" s'ha rebatejat com a "personal_email"

- La taula s'ha rebatejat com a "data_user" en comptes del nom "user" inclòs a l'estructura de la taula. Això s'ha d'haver fet després d'importar els valors de la taula; altrament, s'hauria hagut de modificar tota la importació de la consulta de dades fent referència al nom correcte de la taula.

Captura de pantalla abans del canvi:

Table: user

Columns:

- id** char(10) PK
- name varchar(100)
- surname varchar(100)
- phone varchar(150)
- email** varchar(150)
- birth_date varchar(100)
- country varchar(150)
- city varchar(150)
- postal_code varchar(100)
- address varchar(255)

L0173
L0174 • ALTER TABLE user
L0175 CHANGE COLUMN id id INT;
L0176
L0177 • ALTER TABLE user
L0178 RENAME COLUMN email TO personal_email;
L0179
L0180 • ALTER TABLE user
L0181 RENAME TO data_user;
L0182

Output

Action Output

#	Time	Action

Message

Duration / Fetch

Captura de pantalla després del canvi:

Table: data_user

Columns:

- id** int PK
- 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)

L0173
L0174 • ALTER TABLE user
L0175 CHANGE COLUMN id id INT;
L0176
L0177 • ALTER TABLE user
L0178 RENAME COLUMN email TO personal_email;
L0179
L0180 • ALTER TABLE user
L0181 RENAME TO data_user;
L0182

Output

Action Output

#	Time	Action
1	12:51:38	ALTER TABLE user CHANGE COLUMN id id INT
2	12:51:39	ALTER TABLE user RENAME COLUMN email TO personal_email
3	12:51:39	ALTER TABLE user RENAME TO data_user

Message

5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

0 row(s) affected

Duration / Fetch

0.172 sec

0.016 sec

0.015 sec

FR Français (France)

Finalment, cal fer les modificacions següents a la taula "transaction":

- El "credit_card_id" s'ha modificat de VARCHAR(15) a VARCHAR(20), suposo que ja que la captura de pantalla està truncada i, per tant, no es pot llegir. Això és per tal que coincideixi amb la longitud del camp "id" a la taula "credit_card", ja que "credit_card_id" és una FOREIGN KEY a la taula "transaction".

- Cal crear una FOREIGN KEY en aquesta taula al camp "user_id" enllaçant-la amb la taula "data_user" al seu camp "id". Prèviament, i per tal de permetre la creació de la FOREIGN KEY, he d'inserir el registre de l'usuari "9999" de l'exercici 3, Nivell 1 per tal que tots els registres coincideixin tant a la taula "transaction" com a la taula "data_user".

- Una bona pràctica hauria estat ajustar el camp "company_id" a VARCHAR(15) en comptes de (20) per replicar el tipus de dades del camp "id" a la taula "company", ja que "company_id" és una FOREIGN KEY a la taula "transaction".

Captura de pantalla abans del canvi:

The screenshot shows the MySQL Workbench interface with the 'transaction' table selected in the schema browser. The table structure is displayed, showing columns: id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, and declined. The 'credit_card_id' column is defined as VARCHAR(20). Below the table structure, the SQL editor contains the following code:

```
-- modificacions de la taula "transaction"
ALTER TABLE transaction
CHANGE COLUMN credit_card_id credit_card_id VARCHAR(20);

INSERT INTO data_user (id) VALUES ('9999');

ALTER TABLE transaction
ADD CONSTRAINT fk_transaction_data_user
FOREIGN KEY (user_id)
REFERENCES data_user(id)
ON DELETE RESTRICT
ON UPDATE RESTRICT;
```

The code includes comments indicating the purpose of each step: changing the column type, inserting a specific value into the 'data_user' table, and adding a foreign key constraint.

Captura de pantalla després del canvi:

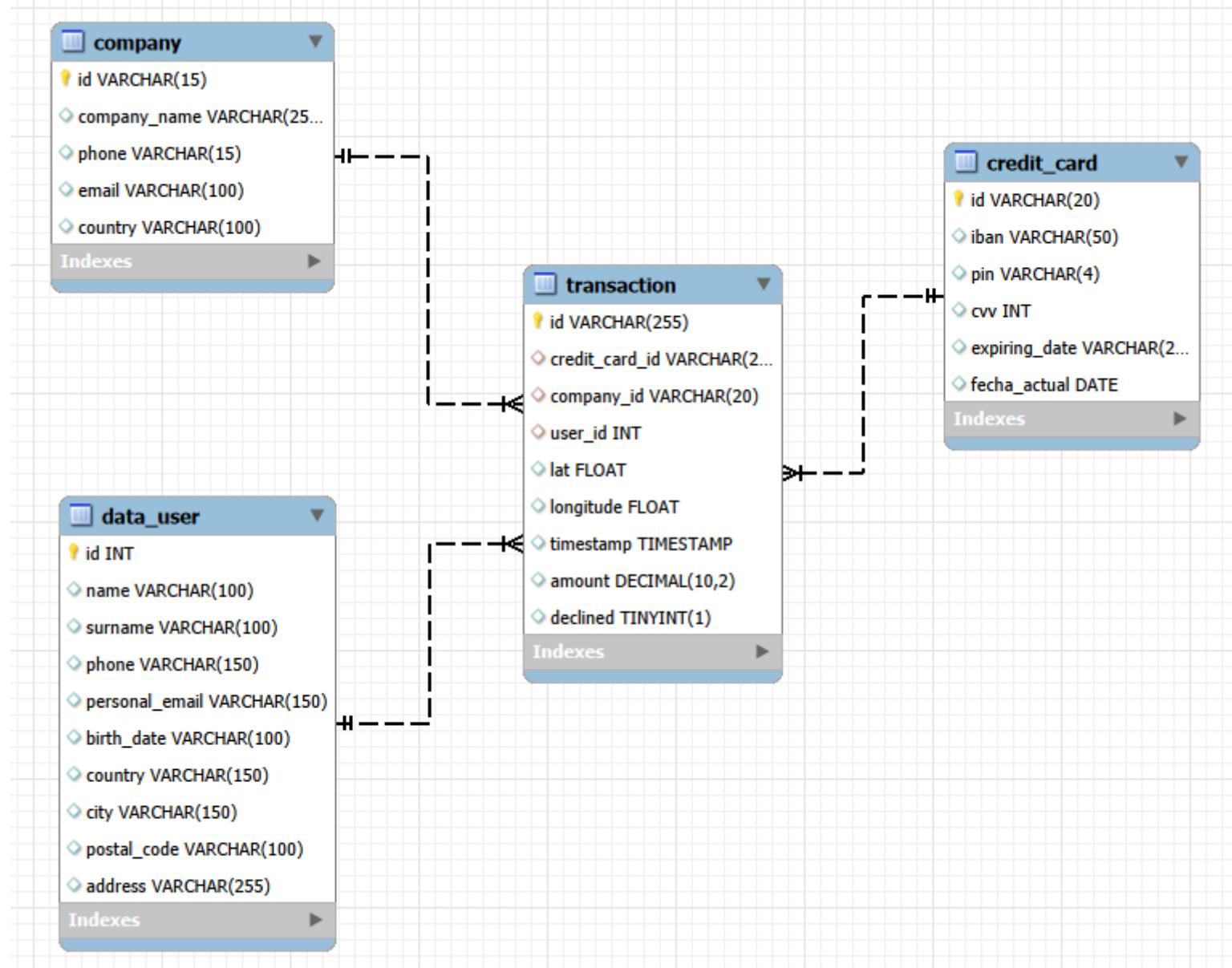
The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is "data_user".
- Table:** transaction
- Columns:**
 - id**: varchar(255) PK
 - credit_card_id**: varchar(20)
 - company_id**: varchar(20)
 - user_id**: int
 - lat**: float
 - longitude**: float
 - timestamp**: timestamp
 - amount**: decimal(10,2)
 - declined**: tinyint(1)
- Related Tables:**
 - Target credit_card (**credit_card_id** → id)
 - On Update RESTRICT
 - On Delete RESTRICT
 - Target data_user (**user_id** → id)
 - On Update RESTRICT
 - On Delete RESTRICT
 - Target company (**company_id** → id)
 - On Update RESTRICT
 - On Delete RESTRICT
- Script History:**

```
-- 182    -- modificacions de la taula "transaction"
183
184 • ALTER TABLE transaction
185     CHANGE COLUMN credit_card_id credit_card_id VARCHAR(20);
186
187 • INSERT INTO data_user (id) VALUES ('9999');
188
189 • ALTER TABLE transaction
190     ADD CONSTRAINT fk_transaction_data_user
191     FOREIGN KEY (user_id)
192     REFERENCES data_user(id)
193     ON DELETE RESTRICT
194     ON UPDATE RESTRICT;
195
196
197
```
- Action Output:**

#	Time	Action	Message	Duration / Fetch
1	13:45:56	INSERT INTO data_user (id) VALUES ('9999')	1 row(s) affected	0.000 sec
2	13:46:05	ALTER TABLE transaction CHANGE COLUMN credit_card_id credit_card_id VARCHAR(20)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
3	13:46:08	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_data_user FOREIGN KEY (user_id) REFERENCES data_user(id) ON DELETE RES...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	3.813 sec

Estat final que es ho mateix que el diagrama del company de feina:



Nivell 3 – Exercici 2

Creo la vista amb les columnes sol·licitades. Afegeixo la rebutjades, ja que la menció "transacció realitzada" no especifica si s'ha completat o no. A més, tal com indica el nom de la vista, l'informe podria utilitzar-se per investigar problemes dels sistemes i el motiu pel qual certes transaccions s'haurien rebutjat. En qualsevol cas, caldria precisar aquest objectiu de la vista/informe amb la persona que ho demana per definir millor la informació que s'hi afegirà.

Seguint la mateixa lògica i per proporcionar dades tècniques exhaustives, he afegit punts de dades que podrien proporcionar informació sobre problemes del sistema, com ara el temps de la transacció i els respectius països del client i l'empresa, per fer possible la identificació de qualsevol discrepància.

Finalment, he afegit tots els ID de CLAU PRIMÀRIA per facilitar la identificació de cada element de la base de dades i la investigació posterior de qualsevol problema que es pogués identificar.

Abans de la creació:

The screenshot shows the MySQL Workbench interface with the 'transactions' schema selected. In the left pane, under 'Views', there is a single entry for 'vistamarketing'. The main pane displays the SQL code for creating the 'InformeTecnico' view:

```
194
195
196 -- **** Exercici 2 ****
197
198
199 • CREATE VIEW InformeTecnico AS
200   SELECT t.id AS "ID de la transacció", d.name AS "Nom de l'usuari/ària", d.surname AS "Cognom de l'usuari/ària", d.country AS "País de l'usuari/ària", cc.iban
201     FROM transaction AS t
202   JOIN company AS c
203     ON c.id = t.company_id
204   JOIN data_user AS d
205     ON d.id = t.user_id
206   JOIN credit_card AS cc
207     ON cc.id = t.credit_card_id
208   ORDER BY "ID de la transacció" DESC;
209
210
211 • SELECT *
212   FROM InformeTecnico
```

The code includes several comments and descriptive column names in Spanish. The 'Output' tab at the bottom is visible but empty.

Creació i consulta de la vista:

Filter objects

- test
- transactions**
 - Tables
 - Views
 - informetecnico**
- vistamarketing
- Stored Procedures
- Functions

```

199 • CREATE VIEW InformeTecnico AS
200   SELECT t.id AS "ID de la transacció", d.name AS "Nom de l'usuari/ària", d.surname AS "Cognom de l'usuari/ària", d.country AS "País de l'usuari/ària", cc.id AS "ID targeta crèdit"
201   FROM transaction AS t
202   JOIN company AS c
203   ON c.id = t.company_id
204   JOIN data_user AS d
205   ON d.id = t.user_id
206   JOIN credit_card AS cc
207   ON cc.id = t.credit_card_id
208   ORDER BY "ID de la transacció" DESC;
209
210
211 • SELECT *
212   FROM InformeTecnico;

```

Administration Schemas

Information

View: informetecnico

Columns:

ID de la transacció	Nom de l'usuari/ària	Cognom de l'usuari/ària	País de l'usuari/ària	ID targeta crèdit	IBAN de la targeta de crèdit usada	Data de caducitat de la targeta de crèdit	ID companya	Nom de la companyia de la transacció realitzada	País de la companya
00138D3B-206D-4C03-94B7-63A2676EB984	Bnyr	Astuw	Italy	Cc5-4899	XX6559192879846049524868	02/26/27	b-2222	Ac Fermentum Incorporated	Germany
0013C1B5-3B84-4DEC-8154-E283FECASE9	Oois	Magurk	Spain	Cc5-5070	XX3986101402612563256121	01/30/29	b-2222	Ac Fermentum Incorporated	Germany
00201A11-2E62-4AC4-941D-198FC30877F0	Minerva	Wilkins	United Kingdom	CcU-j512	PT85885236815643142117247	05/02/21	b-2222	Ac Fermentum Incorporated	Germany
00235618-A05C-4D49-90CB-83A9403D8923	Qlgmbh	Tdpgcty	Sweden	Cc5-8137	XX344726545995864622888095	01/25/29	b-2222	Ac Fermentum Incorporated	Germany
005A5A7B-1F1A-4B6C-9815-1625A78C9C38	Qgihl	Wglchsgn	Portugal	Cc5-8998	XX56781948177169984505259	08/26/25	b-2222	Ac Fermentum Incorporated	Germany
00687139-48B2-4FFA-BE73-B20376F04AB4	Dxwigl	Hwcrw	Germany	Cc5-4870	XX644366380416773213949	01/25/25	b-2222	Ac Fermentum Incorporated	Germany
0074F4D-32F1-4827-8758-5896314623A	Sfbajg	Lvahwsn	Portugal	Cc5-8081	XX50534192304296926602260	10/28/25	b-2222	Ac Fermentum Incorporated	Germany
00AA89CD-3906-4DC8-8A1D-138E73D9C90A9	Miyvych	Lehzesm	United Kingdom	Cc5-6797	XX351920436838165027042306	07/25/27	b-2222	Ac Fermentum Incorporated	Germany
00BE0904-6920-47D8-AE88-325E22698290	Fpymtvae	Kvuvnta	Portugal	Cc5-4983	XX6536484667648913457320	04/28/28	b-2222	Ac Fermentum Incorporated	Germany
00DA0383-E048-4577-8ED1-3C56C258FF2F	Zqofuz	Murumfwe	Germany	Cc5-9223	XX485221033677161621716060	10/27/28	b-2222	Ac Fermentum Incorporated	Germany
00DD11DE-ED01-4B8D-93A0-174D183A59D9	Wpdeav	Rkcojark	France	Cc5-7681	XX968498943366183971975457	12/27/25	b-2222	Ac Fermentum Incorporated	Germany

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result Grid Form Editor Field Types

Informtecnico 5 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	11:56:50	CREATE VIEW InformeTecnico AS SELECT t.id AS "ID de la transacció", d.name AS "Nom de l'usuari/ària", d.surname AS "Cognom de l'usuari/ària", d.country AS "País de l'usuari/ària", cc.id AS "ID targeta crèdit"	0 row(s) affected	0.000 sec
2	11:56:55	SELECT * FROM InformeTecnico	100000 row(s) returned	0.000 sec / 1.140 sec

Read Only