

Sprint 3 - Tasca S3.01. Manipulació de taules

Nivell 1

- Exercici 1

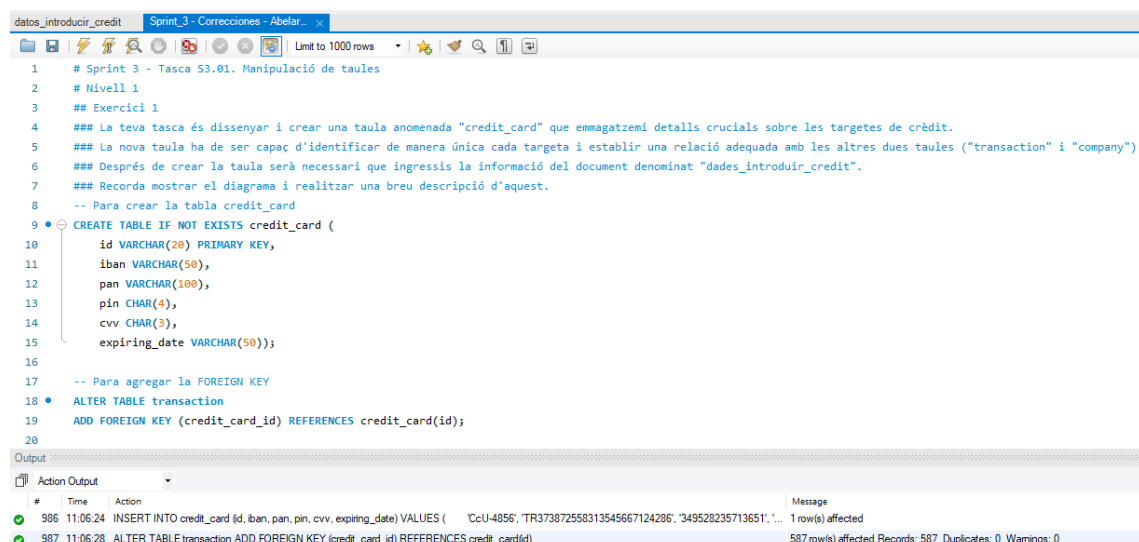
La teva tasca és dissenyar i crear una taula anomenada "credit_card" que emmagatzemi detalls crucials sobre les targetes de crèdit. La nova taula ha de ser capaç d'identificar de manera única cada targeta i establir una relació adequada amb les altres dues taules ("transaction" i "company"). Després 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.

La primera vez que cree la tabla lo hice con valores arbitrarios que luego tuve que modificar para que coincidan con los me piden los siguientes ejercicios. Así que tenía la opción de modificarlos desde el principio o continuar como los había planteado por primera vez. Decidí dejarlos como la primera vez para que el desarrollo del Sprint fuera más orgánico.

Después de crear la tabla agregué la Foreign Key para que pueda interactuar con la tabla de hechos transaction.

```
CREATE TABLE IF NOT EXISTS credit_card (  
  id VARCHAR(20) PRIMARY KEY,  
  iban VARCHAR(50),  
  pan VARCHAR(100),  
  pin CHAR(4),  
  cvv CHAR(3),  
  expiring_date VARCHAR(50));
```

```
ALTER TABLE transaction  
ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
```

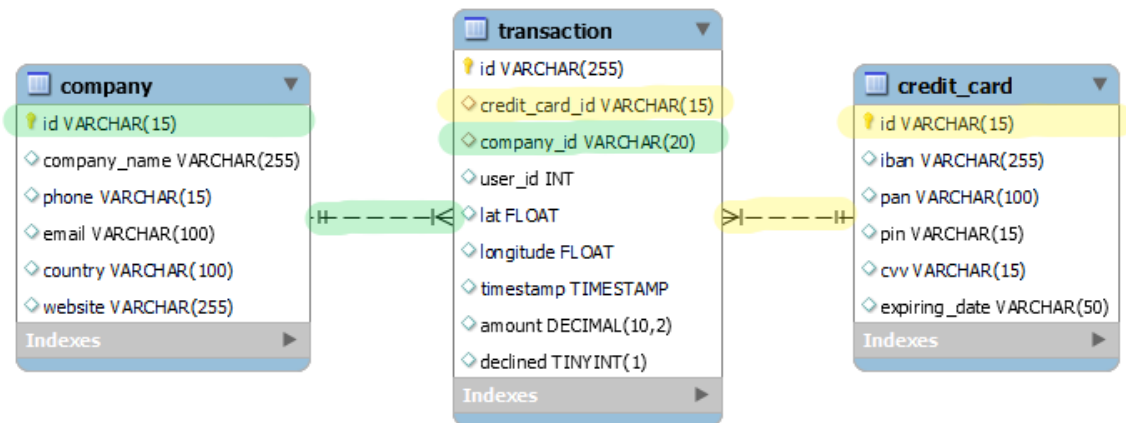


The screenshot shows a SQL IDE window titled "datos_introducir_credit" with a tab "Sprint_3 - Correcciones - Abelar". The SQL editor contains the following code:

```
1 # Sprint 3 - Tasca S3.01. Manipulació de taules  
2 # Nivell 1  
3 ## Exercici 1  
4 ### La teva tasca és dissenyar i crear una taula anomenada "credit_card" que emmagatzemi detalls crucials sobre les targetes de crèdit.  
5 ### 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").  
6 ### Després de crear la taula serà necessari que ingressis la informació del document denominat "dades_introduir_credit".  
7 ### Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.  
8 -- Para crear la tabla credit_card  
9 • CREATE TABLE IF NOT EXISTS credit_card (  
10   id VARCHAR(20) PRIMARY KEY,  
11   iban VARCHAR(50),  
12   pan VARCHAR(100),  
13   pin CHAR(4),  
14   cvv CHAR(3),  
15   expiring_date VARCHAR(50));  
16  
17 -- Para agregar la FOREIGN KEY  
18 • ALTER TABLE transaction  
19   ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);  
20
```

The Output window shows the following results:

#	Time	Action	Message
986	11:06:24	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-4856', 'TR373872568313545667124286', '349528235713651', '...', '1 row(s) affected	
987	11:06:28	ALTER TABLE transaction ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0



Acá podemos ver en el diagrama que creamos la tabla `credit_card`, y que la FOREIGN KEY fue agregada de forma satisfactoria.

Ahora, con la tabla creada, los datos cargados y la FOREIGN KEY implementada correctamente, podemos interactuar con la tabla `credit_card` con gran facilidad.

- 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.

En esta ocasión lo primero que hice fue un "SELECT * FROM credit_card WHERE id = "CcU-2938";", para ver los datos que habían asignados al id CcU-2938. Luego con UPDATE cambie el iban usando el id como condicional en el WHERE.

The screenshot shows a database interface with a query editor and a results grid. The query is:

```
-- Para ver los datos que habían asignados al id CcU-2938
SELECT *
FROM credit_card
WHERE id = "CcU-2938";
```

The results grid shows one row with the following data:

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

The output section shows the action output:

```
SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 1000
```

The message indicates: 1 row(s) returned.

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

The screenshot shows a database interface with a query editor and a results grid. The query is:

```
## Ejercicio 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.
-- Para actualizar el iban de "TR301950312213576817638661" que es erroneo, a "R323456312213576817699999" que es el correcto.
UPDATE credit_card
SET iban = "R323456312213576817699999"
WHERE id = "CcU-2938";
```

The output section shows the action output:

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

The message indicates: 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

Para verificar los cambios hechos

The screenshot shows a database interface with a query editor and a results grid. The query is:

```
-- Para verificar los datos asignados al id CcU-2938
SELECT *
FROM credit_card
WHERE id = "CcU-2938";
```

The results grid shows one row with the following data:

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

The output section shows the action output:

```
SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 1000
```

The message indicates: 1 row(s) returned.

- Exercici 3

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

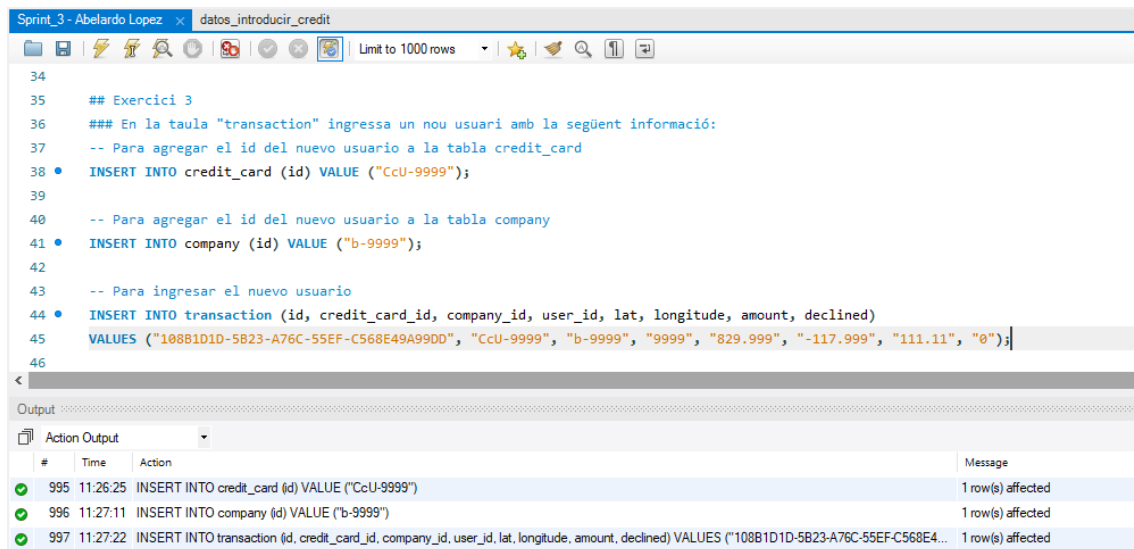
En este ejercicio estuve largo rato tratando de ingresar el nuevo usuario a la tabla transacion pero no podía porque no se puede modificar una CHILD ROW con un FOREIGN KEY CONSTRAINT. Así que primero agregué el credit_card_id, y el company_id a sus respectivas tablas, para de esa manera poder ingresar todos los valores del nuevo usuario a la tabla transaction.

```
INSERT INTO credit_card (id) VALUE ("CcU-9999");
```

```
INSERT INTO company (id) VALUE ("b-9999");
```

```
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");
```



The screenshot shows a SQL IDE window titled "Sprint_3 - Abelardo Lopez" with a file named "datos_introducir_cred". The editor contains three SQL statements with comments in Spanish. The output pane shows the execution results for each statement, indicating that each statement affected 1 row.

```
34  
35  ## Exercici 3  
36  ### En la taula "transaction" ingressa un nou usuari amb la següent informació:  
37  -- Para agregar el id del nuevo usuario a la tabla credit_card  
38  • INSERT INTO credit_card (id) VALUE ("CcU-9999");  
39  
40  -- Para agregar el id del nuevo usuario a la tabla company  
41  • INSERT INTO company (id) VALUE ("b-9999");  
42  
43  -- Para ingresar el nuevo usuario  
44  • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)  
45  VALUES ("108B1D1D-5B23-A76C-55EF-C568E49A99DD", "CcU-9999", "b-9999", "9999", "829.999", "-117.999", "111.11", "0");  
46
```

Output

#	Time	Action	Message
995	11:26:25	INSERT INTO credit_card (id) VALUE ("CcU-9999")	1 row(s) affected
996	11:27:11	INSERT INTO company (id) VALUE ("b-9999")	1 row(s) affected
997	11:27:22	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ("108B1D1D-5B23-A76C-55EF-C568E4...	1 row(s) affected

- Exercici 4

Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card. Recorda mostrar el canvi realitzat.

Para este ejercicio lo que hice fue alterar la tabla credit_card para así poder hacer un DROP de la columna pan. Luego hice un SELECT * FROM credit_card; para ver el resultado sin la columna pan.

ALTER TABLE credit_card DROP COLUMN pan;

The screenshot shows a database management tool interface. The top panel displays SQL code for Exercise 4. The middle panel shows the 'Result Grid' with a table of credit card data. The bottom panel shows the 'Output' log with two successful execution messages.

```
47 ## Exercici 4
48 ### Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card.
49 ### Recorda mostrar el canvi realitzat.
50 -- Para hacer el DROP de la columna pan
51 • ALTER TABLE credit_card DROP COLUMN pan;
52
53 -- Para ver la tabla credit_card sin la columna "pan"
54 • SELECT *
55   FROM credit_card;
56
```

id	iban	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	9080	887	08/24/23
CcU-2952	BG451VQL52710525608255	4598	438	06/29/21
CcU-2959	CR7242477244335841535	3583	667	02/24/23
CcU-2966	BG72LKTQ15627628377363	4900	130	10/29/24
CcU-2973	PT87806228135092429456346	8760	887	01/30/25
CcU-2980	DE39241881883086277136	5075	596	07/24/22
CcU-2987	GE89681434837748781813	2298	797	10/31/23
CcU-2994	BH62714428368066765294	7545	595	02/28/22

credit_card 5 x

Output

#	Time	Action	Message
998	11:32:30	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
999	11:33:01	SELECT * FROM credit_card LIMIT 0, 1000	276 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.

En este ejercicio primero fue ver los registros asignados con un SELECT filtrado por el id. Luego usé la función DELETE para eliminar ese registro de la base de datos. Y finalmente, corroboré que todo se hizo correctamente volviendo a correr el SELECT.

DELETE FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";

The screenshot shows a database management interface with a SQL editor and an output window. The SQL editor contains the following code:

```
57 # Nivell 2
58 ## Exercici 1
59 ### Elimina de la taula transaction el registre amb ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de dades.
60 -- Para eliminar el registro de l tabla transaction
61 • DELETE FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
62
63 • SELECT *
64 FROM transaction
65 WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
66
```

The output window shows the results of the SQL execution:

#	Time	Action	Message
✓ 1000	11:36:43	SELECT * FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	1 row(s) returned
✓ 1001	11:36:56	DELETE FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02"	1 row(s) affected
✓ 1002	11:37:58	SELECT * FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	0 row(s) returned

- 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.

Para este ejercicio lo primero que hice fue crear la Query con los datos que me está pidiendo el enunciado.

Como los datos están en dos tablas distintas usé un JOIN para poder acceder a la información que se me solicita.

Agrupé por la columna company_id de la tabla transaction porque de esa manera me daría un promedio de cada compañía, y no un promedio de toda la tabla transaction en general.

Para finalizar cree la vista con su nombre correspondiente. Y ordené los datos de mayor a menor promedio de compra para mostrar la vista.

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

The screenshot shows a SQL IDE window titled "datos_introducir_credi" and "Sprint_3 - Correcciones - Abelar...". The editor contains the following SQL code:

```
66 ## Exercici 2
67 ### La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives.
68 ### S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions.
69 ### Serà necessària que creïs una vista anomenada VistaMarketing que contingui la següent informació:
70 ### Nom de la companyia. Telèfon de contacte. País de residència. Mitjana de compra realitzat per cada companyia.
71 ### Presenta la vista creada, ordenant les dades de major a menor mitjana de compra.
72 • CREATE VIEW VistaMarketing AS
73 SELECT company_name, phone, country, AVG(amount) AS average
74 FROM company AS c
75 JOIN transaction AS t ON c.id = t.company_id
76 GROUP BY company_id;
77
78 -- Para ver la vista creada
79 • SELECT *
80 FROM VistaMarketing
81 ORDER BY average DESC;
82
```

Below the editor, the "Result Grid" shows the output of the query. It displays a table with 4 columns: company_name, phone, country, and average. The results are ordered by average in descending order.

company_name	phone	country	average
Eget Ipsum Ltd	03 67 44 56 72	United States	473.075000
Non Magna LLC	06 71 73 13 17	United Kingdom	468.345000
Sed Id Limited	07 28 18 18 13	United States	461.210000
Justo Eu Arcu Ltd	08 42 56 71 52	Italy	443.635000
Eget Tincidunt Dui Institute	05 35 93 32 44	Netherlands	442.520000
Viverra Donec Foundation	03 33 12 32 73	United Kingdom	442.280000
Vestibulum Lorem PC	02 02 87 33 40	Belgium	434.060000
Aliquet Diam Limited	02 76 61 47 46	United States	425.640000
Maecenas Malesuada Fringilla Inc.	09 38 53 76 61	Netherlands	408.620000
Non Ante LLP	08 89 47 65 08	Sweden	407.790000
Egestas Nunc Sed Limited	06 01 02 70 47	Italy	406.110000
Nunc Sit Incorporated	07 28 42 63 63	Norway	405.355000
Magna A Neque Industries	04 14 44 64 62	Australia	396.315000
Amet Luctus Vulputate Foundation	03 18 54 24 19	Canada	390.325000
Aliquam PC	01 45 73 52 16	Germany	385.265000
Neque Tellus Incorporated	04 43 18 34 19	Ireland	364.005000
Placerat LLP	05 43 67 24 41	Netherlands	357.080000
Elit Ebum Laoreet Associates	07 69 74 17 45	Canada	351.840000
Fusce Corp.	08 14 97 58 85	United States	350.125000
Sapien Nunc Pulvinar LLP	08 37 12 58 11	New Zealand	349.655000
Mauris Institute	05 29 60 36 87	Sweden	346.875000
Tincidunt Orci Limited	01 78 18 81 44	Norway	326.860000
Netus Et Malesuada Ltd	02 55 43 68 46	Netherlands	325.335000
Dictum Eu Corp.	03 04 73 67 31	Canada	318.645000

The "Output" section at the bottom shows the execution details:

```
# Time Action Message
999 11:08:56 SELECT * FROM VistaMarketing ORDER BY average DESC LIMIT 0, 1000 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".

Este ejercicio es muy sencillo porque puedo interactuar con la vista creada como si fuera una tabla, simplemente tengo que usar su nombre en el FROM.

Entonces para este ejercicio solo tendré que filtrar por el país Alemania.

```
SELECT *  
FROM VistaMarketing  
WHERE country = "Germany";
```

The screenshot shows a database interface with a SQL editor and a results grid. The SQL query is as follows:

```
82  
83  ## Exercici 3  
84  ### Filtra la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"  
85  • SELECT *  
86  FROM VistaMarketing  
87  WHERE country = "Germany";  
88
```

The results grid displays the following data:

company_name	phone	country	average
Ac Fermentum Incorporated	06 85 56 52 33	Germany	206.465000
Convallis In Incorporated	06 66 57 29 50	Germany	156.730000
Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.025238
Augue Foundation	06 88 43 15 63	Germany	240.800000
Ac Industries	09 34 65 40 60	Germany	289.645000
Auctor Mauris Corp.	05 62 87 14 41	Germany	184.310000
Aliquam PC	01 45 73 52 16	Germany	385.265000
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000

The output section shows the following message:

```
1286 11:39:44 SELECT * FROM VistaMarketing WHERE country = "Germany" LIMIT 0, 1000 8 row(s) returned
```


Nivell 3

- Exercici 1

La setmana vinent tindrà 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:

Para este ejercicio empecé por revisar el diagrama que me dan como modelo a seguir para ver lo que tendría que ir haciendo para llegar a ese resultado.

Entonces primero cargué la estructura de datos de la nueva tabla user, y los datos correspondientes a la tabla.

Segundo, modifiqué la tabla credit_card para que tenga las columnas que le corresponden, y también su adecuado formato en cada columna.

Tercero, modifiqué la tabla company para hacer un DROP de la columna website.

Y para finalizar, renombre la columna email a personal_email en la tabla user, y también renombre la tabla user a data_user como se ve en el diagrama.

-- Modificar la tabla credit_card

```
ALTER TABLE credit_card
```

```
ADD COLUMN fecha_actual DATE NULL DEFAULT NULL AFTER expiring_date,
```

```
CHANGE COLUMN pin pin VARCHAR(4) NULL DEFAULT NULL,
```

```
CHANGE COLUMN cvv cvv INT NULL DEFAULT NULL,
```

```
CHANGE COLUMN expiring_date expiring_date VARCHAR(10) NULL DEFAULT NULL;
```

-- Modificar la tabla company

```
ALTER TABLE company
```

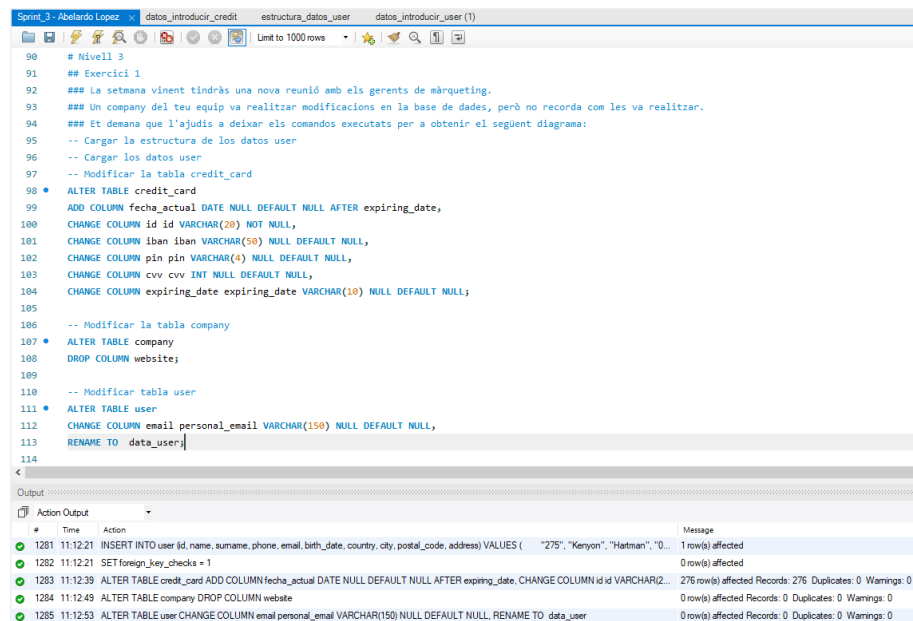
```
DROP COLUMN website;
```

-- Modificar tabla user

```
ALTER TABLE user
```

```
CHANGE COLUMN email personal_email VARCHAR(150) NULL DEFAULT NULL,
```

```
RENAME TO data_user;
```



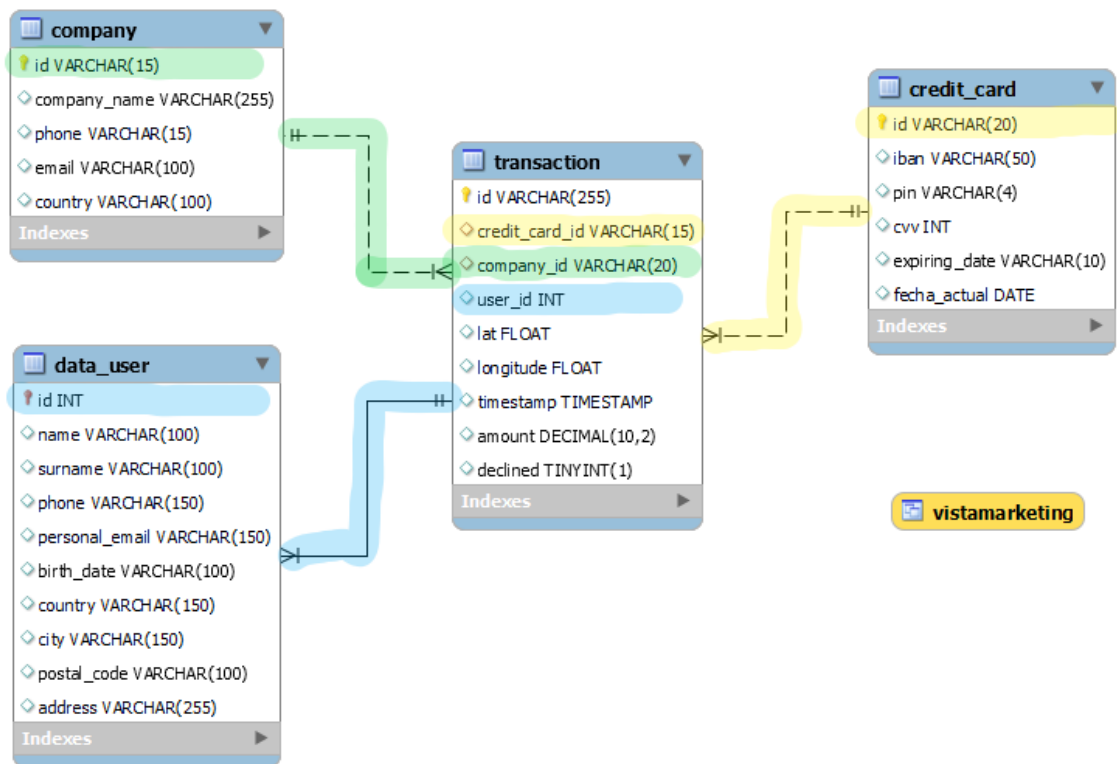
The screenshot shows a SQL IDE with a script editor and an output window. The script editor contains the following SQL commands:

```
90 # Nivell 3
91 ## Exercici 1
92 ### La setmana vinent tindrà una nova reunió amb els gerents de màrqueting.
93 ### Un company del teu equip va realitzar modificacions en la base de dades, però no recorda com les va realitzar.
94 ### Et demana que l'ajudis a deixar els comandos executats per a obtenir el següent diagrama:
95 -- Cargar la estructura de los datos user
96 -- Cargar los datos user
97 -- Modificar la tabla credit_card
98 * ALTER TABLE credit_card
99   ADD COLUMN fecha_actual DATE NULL DEFAULT NULL AFTER expiring_date,
100   CHANGE COLUMN id id VARCHAR(20) NOT NULL,
101   CHANGE COLUMN iban iban VARCHAR(50) NULL DEFAULT NULL,
102   CHANGE COLUMN pin pin VARCHAR(4) NULL DEFAULT NULL,
103   CHANGE COLUMN cvv cvv INT NULL DEFAULT NULL,
104   CHANGE COLUMN expiring_date expiring_date VARCHAR(10) NULL DEFAULT NULL;
105
106 -- Modificar la tabla company
107 * ALTER TABLE company
108   DROP COLUMN website;
109
110 -- Modificar tabla user
111 * ALTER TABLE user
112   CHANGE COLUMN email personal_email VARCHAR(150) NULL DEFAULT NULL,
113   RENAME TO data_user;
114
```

The output window shows the results of the execution:

#	Time	Action	Message
1281	11:12:21	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("275", "Keryon", "Hartman", "0...	1 row(s) affected
1282	11:12:21	SET foreign_key_checks = 1	0 row(s) affected
1283	11:12:39	ALTER TABLE credit_card ADD COLUMN fecha_actual DATE NULL DEFAULT NULL AFTER expiring_date, CHANGE COLUMN id id VARCHAR(20) NOT NULL, CHANGE COLUMN iban iban VARCHAR(50) NULL DEFAULT NULL, CHANGE COLUMN pin pin VARCHAR(4) NULL DEFAULT NULL, CHANGE COLUMN cvv cvv INT NULL DEFAULT NULL, CHANGE COLUMN expiring_date expiring_date VARCHAR(10) NULL DEFAULT NULL;	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
1284	11:12:49	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
1285	11:12:53	ALTER TABLE user CHANGE COLUMN email personal_email VARCHAR(150) NULL DEFAULT NULL, RENAME TO data_user	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

-- Diagrama



- 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.

Para este ejercicio la vista necesita información de practicante todas las tablas, entonces tuve que concatenar múltiples JOINS desde las tablas de dimensiones a la tabla transacción que es la tabla de hechos donde todas concurren.

Teniendo lo anterior en mente procedí con ir estructurando la Query que me permitiera acceder a toda la información solicitada, y dándole el alias transaction_id a la columna id porque simplemente id es muy ambiguo, y cree la vista.

Para finalizar simplemente mostré la vista ordenada por el id de transaction.

-- Para crear la vista solicitada

```
CREATE VIEW InformeTecnico AS
```

```
SELECT t.id AS transaction_id, du.name, du.surname, cc.iban, c.company_name
```

```
FROM transaction AS t
```

```
JOIN data_user AS du ON du.id = t.user_id
```

```
JOIN credit_card AS cc ON cc.id = t.credit_card_id
```

```
JOIN company AS c ON c.id = t.company_id;
```

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
-- 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.
-- Para crear la vista solicitada
CREATE VIEW InformeTecnico AS
SELECT t.id AS transaction_id, du.name, du.surname, cc.iban, c.company_name
FROM transaction AS t
JOIN data_user AS du ON du.id = t.user_id
JOIN credit_card AS cc ON cc.id = t.credit_card_id
JOIN company AS c ON c.id = t.company_id;

-- Para ver el resultado de la vista creada
SELECT *
FROM InformeTecnico
ORDER BY transaction_id DESC;
```

The results pane shows the output of the query, displaying a table with 5 columns: transaction_id, name, surname, iban, and company_name. The table contains 100 rows of data, sorted by transaction_id in descending order.

transaction_id	name	surname	iban	company_name
FE96CE47-8D59-381C-4E18-E3CA3D44E8FF	Kerlyon	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
PD89051B-AE8D-77DC-E450-88083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
PD2E8957-414B-8EEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-80D92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
FBD7E3D6-846B-F5BC-9C49-6A487601D0C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT053WCF58868200575771634583813	Arcu LLP
FAAD39FC-1A17-E141-43D3-359A5BA7C83B	Hedwig	Gilbert	GE90157928843338134463	Lorem Eu Incorporated
FA053936-75D8-85FA-490D-98624E1B920A	Hedwig	Gilbert	GT02497653655330848247645975	Non Justo Corp.
F85A7D75-2778-9D75-D776-3F41A82DE88	Sarah	Beck	VG1468087984174645729577	Ut Semper Foundation
F84D0C08-CCB5-2444-1B4E-5966289FBA88	Jasper	Landry	VG1468087984174645729577	Ut Semper Foundation
F5ACD74B-4275-5AA1-2414-6EF417636898	Nora	Reeves	MD1234119525145401270486	Nunc Interdum Incorporated
F56FC4A4-0039-9F64-7376-85632B91121B	Lynn	Riddle	CR7242477244335841535	Ut Semper Foundation
F53B3C81-3379-E0BF-5AB9-6F4CC2C5479C	Sonya	Mckee	EE541536644818872885	Arcu LLP
F4BCAE41-388E-EA8D-9C24-466F7CEB9F9A	Chester	Haynes	CY94263537405015481188625576	Malesuada PC
F2B3E45-2E6D-E891-9D05-33DBACE58DE4	Heather	Burks	SM6022751049715477062682363	Malesuada PC
F28E106B-5418-4667-9514-2E2A823ECC65	Hedwig	Gilbert	RO76DAF06583348580208155	Pede Ultrices Ltd
F233E386-5A74-6383-8111-09FC9BA38011	Nero	Mills	HU9521562774927657356556322	Arcu LLP
F22B8361-E3CD-BC41-DD6A-A4694F175CD8	Alko	Chaney	MD5723087436783068347555	Non Institute
F1A598A2-66C5-50A9-F1CE-FB1D69866C39	Craig	Shepherd	R323456312213576817699999	Lorem Eu Incorporated
EFEECC2E-2A69-AB33-0599-E82AA689E383	Alka	Kirney	R323456312213576817699999	Lorem Eu Incorporated
EF9E08A9-457D-D7C1-41E4-059B31C35CEC	Cassan...	Ferguson	DO68192976973138848171352176	Ut Semper Foundation
E6AAE515-66B6-8449-5926-5D40397988A9	Ocean	Nelson	AE640696354928782425103	Et Magnis Ltd

The output pane shows the execution of the query, displaying the number of rows affected and the message.

#	Time	Action	Message
1284	11:21:13	CREATE VIEW InformeTecnico AS SELECT t.id AS transaction_id, du.name, du.surname, cc.iban, c.company_name FROM transaction AS t JOIN da...	0 row(s) affected
1285	11:21:20	SELECT * FROM InformeTecnico ORDER BY transaction_id DESC LIMIT 0, 1000	586 row(s) returned