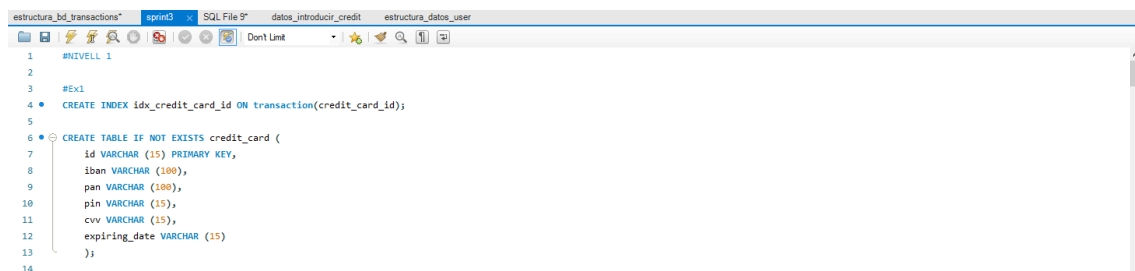


NIVELL 1

Ex1

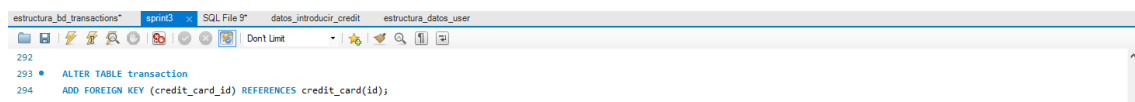
Primero crearemos la tabla *credit_card*, con la información de las tarjetas de crédito. Asignaremos a todas las columnas el formato **VARCHAR** ya que sin tener mas información es el formato que de momento nos servira para todo tipo de datos. Definiremos el campo *id* como **PRIMARY KEY**. Se creara también un *index* en la tabla *transaction* para hacer busquedas mas eficientes en dicha tabla cuando se relaciones *transaction.credit_card_id = credit_card.id*.



```
1 #NIVELL 1
2
3 #Ex1
4 CREATE INDEX idx_credit_card_id ON transaction(credit_card_id);
5
6 CREATE TABLE IF NOT EXISTS credit_card (
7     id VARCHAR (15) PRIMARY KEY,
8     iban VARCHAR (100),
9     pan VARCHAR (100),
10    pin VARCHAR (15),
11    cvv VARCHAR (15),
12    expiring_date VARCHAR (15)
13 );
14
```

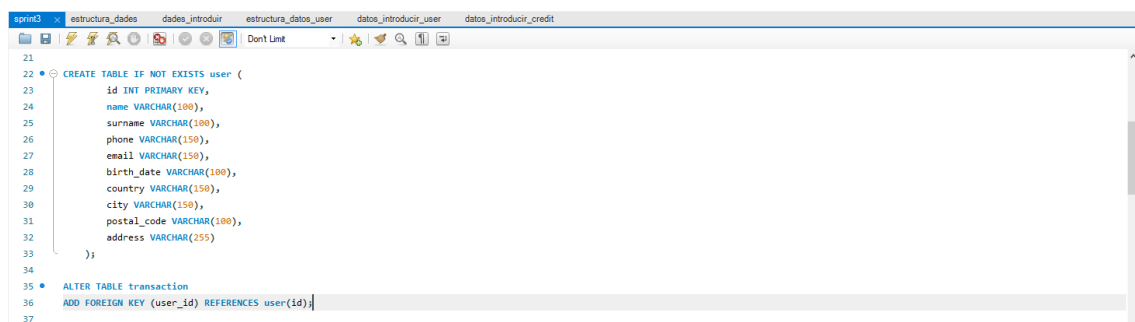
Con la tabla creada ejecutaremos el archivo proporcionado para llenar la misma de datos.

A continuación crearemos la **FOREIGN KEY** en la tabla *transaction* relacionando *transaction.credit_card_id = credit_card.id*. Lo haremos así porque tiene mas sentido que la **CONSTRAINT** se compruebe al hacer inputs en la tabla de transacciones, de este modo no se puede agregar una *row* nueva a la tabla *transaction* si el *credit_card_id* no existe previamente en la tabla *credit_card*. Esto tiene mas sentido que lo contrario, que seria que una transaccion con una *credit_card_id* tenga que existir antes de añadir una nueva tarjeta de credito a la base de datos. Esto lo haremos mediante un **ALTER TABLE – ADD FOREIGN KEY**.



```
292
293 ALTER TABLE transaction
294 ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
295
```

Del mismo modo a la hora de crear la tabla de users eliminaremos la linea que crea una **FOREIGN KEY** relacionando *user.id = transaction.user_id*. Por la misma razón que se ha argumentado anteriormente y crearemos esta relación en la tabla *transaction* mediante **ALTER TABLE – ADD FOREIGN KEY**.

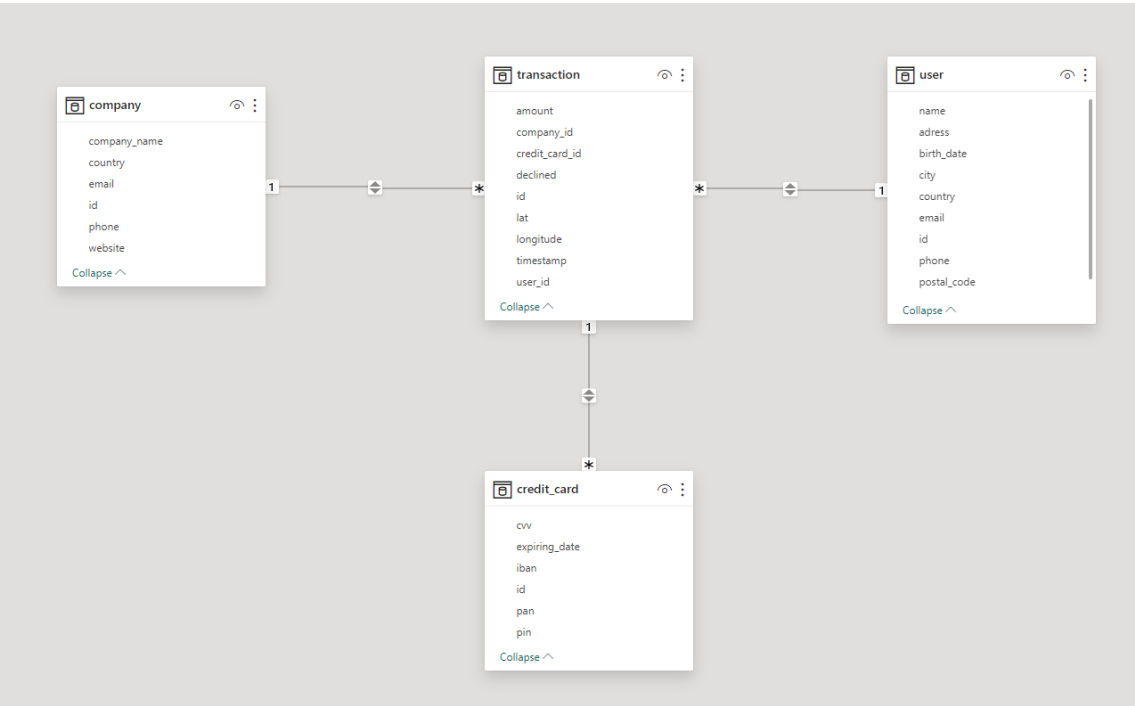


```
21
22 CREATE TABLE IF NOT EXISTS user (
23     id INT PRIMARY KEY,
24     name VARCHAR(100),
25     surname VARCHAR(100),
26     phone VARCHAR(150),
27     email VARCHAR(150),
28     birth_date VARCHAR(100),
29     country VARCHAR(150),
30     city VARCHAR(150),
31     postal_code VARCHAR(100),
32     address VARCHAR(255)
33 );
34
35 ALTER TABLE transaction
36 ADD FOREIGN KEY (user_id) REFERENCES user(id);
37
```

Finalmente ejecutaremos los archivos proporcionados para llenar estas tablas de datos.*

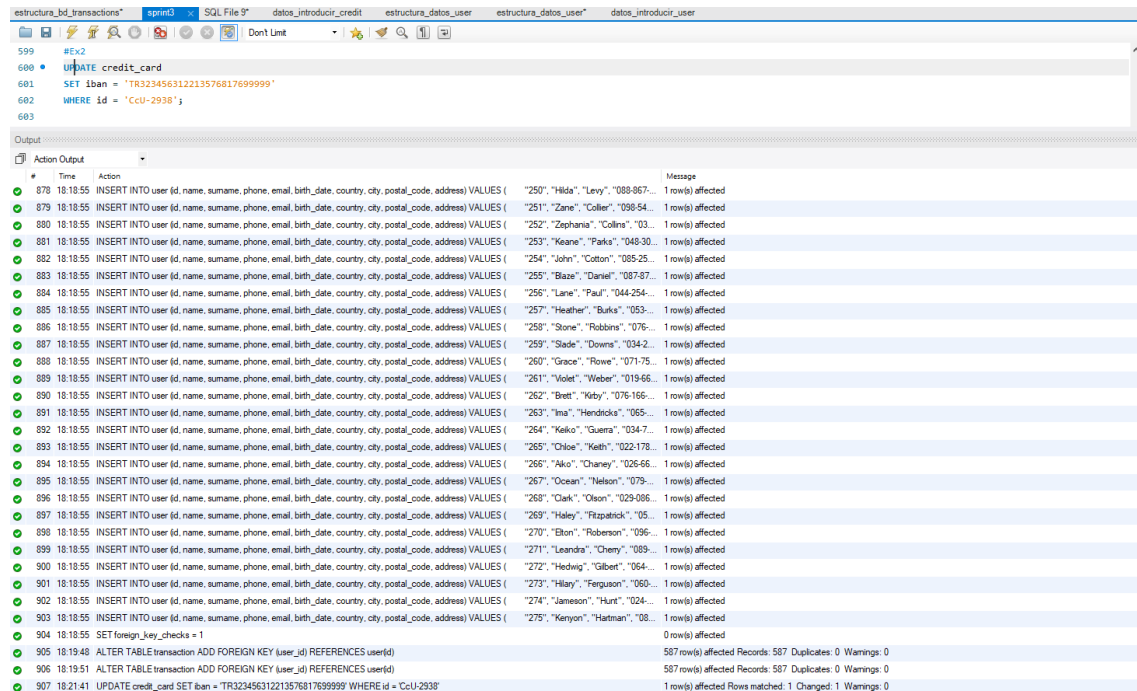
*Es necesario realizar este paso antes de establecer la **FOREING KEY** emtre las tablas.

Obtenemos esta base de datos en forma de estrella donde la tabla *transaction* es la **FACT TABLE**, y las tablas *company*, *user* y *credit_card* son **DIMENSION TABLES**.



Ex2

Modificaremos el valor de una celda mediante las instrucciones **UPDATE** y **SET**, con la condición en la cláusula **WHERE** del usuario que nos interesa.



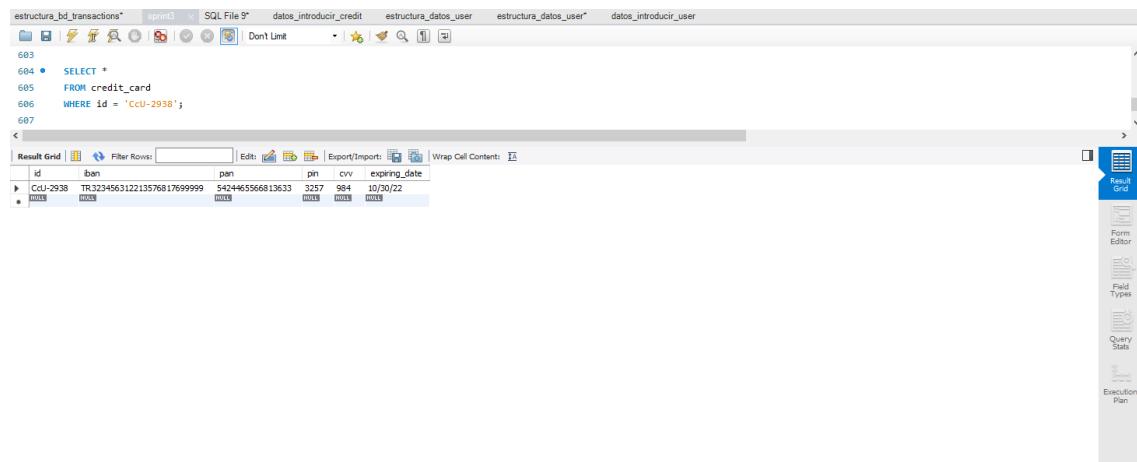
The screenshot shows a SQL IDE with a query editor and an output window. The query editor contains the following SQL statement:

```
599 #Ex2
600 UPDATE credit_card
601 SET iban = 'TR323456312213576817699999'
602 WHERE id = 'Ccu-2938';
603
```

The output window displays the execution results, showing a list of actions and their messages. The final message indicates that the UPDATE statement was successful, affecting 1 row and changing 1 row.

#	Time	Action	Message	
578	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("250", "Hilda", "Levy", "089-867-...	1 row(s) affected
579	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("251", "Zane", "Collier", "090-54-...	1 row(s) affected
580	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("252", "Zephania", "Collins", "03...	1 row(s) affected
581	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("253", "Keane", "Parks", "048-30...	1 row(s) affected
582	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("254", "John", "Cotton", "085-25...	1 row(s) affected
583	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("255", "Blaze", "Daniel", "087-87...	1 row(s) affected
584	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("256", "Lane", "Paul", "044-254-...	1 row(s) affected
585	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("257", "Heather", "Barks", "053...	1 row(s) affected
586	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("258", "Stone", "Robbins", "076...	1 row(s) affected
587	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("259", "Slade", "Downs", "034-2...	1 row(s) affected
588	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("260", "Grace", "Rowe", "071-75...	1 row(s) affected
589	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("261", "Violet", "Weber", "019-66...	1 row(s) affected
590	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("262", "Brett", "Kitty", "076-166...	1 row(s) affected
591	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("263", "Ina", "Hendricks", "065...	1 row(s) affected
592	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("264", "Kelso", "Guerra", "034-7...	1 row(s) affected
593	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("265", "Olivia", "Keith", "022-178...	1 row(s) affected
594	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("266", "Aiko", "Chen", "025-66...	1 row(s) affected
595	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("267", "Ocean", "Nelson", "079-...	1 row(s) affected
596	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("268", "Clark", "Olson", "029-086...	1 row(s) affected
597	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("269", "Haley", "Ritzpatrick", "05...	1 row(s) affected
598	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("270", "Elton", "Roberson", "096...	1 row(s) affected
599	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("271", "Leandra", "Chen", "089...	1 row(s) affected
600	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("272", "Hedwig", "Gilbert", "064...	1 row(s) affected
601	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("273", "Hilary", "Ferguson", "060...	1 row(s) affected
602	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("274", "Jameson", "Hunt", "024...	1 row(s) affected
603	18:18:55	INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ("275", "Kanyon", "Hartman", "08...	1 row(s) affected
604	18:18:55	SET foreign_key_checks = 1		0 row(s) affected
605	18:19:48	ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES user(id)		587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
606	18:19:51	ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES user(id)		587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0
607	18:21:41	UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE id = 'Ccu-2938'		1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

Mediante un **SELECT** podemos comprobar que el cambio se ha realizado correctamente.



The screenshot shows a SQL IDE with a query editor and a result grid. The query editor contains the following SQL statement:

```
603
604 SELECT *
605 FROM credit_card
606 WHERE id = 'Ccu-2938';
607
```

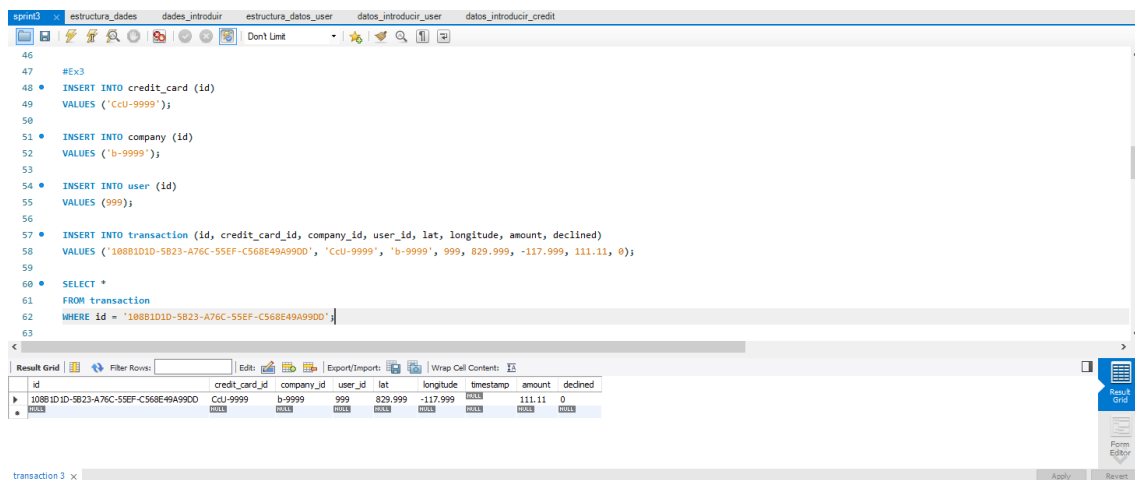
The result grid displays the following data:

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

Ex3

Usaremos el comando **INSERT INTO**, como no alteramos todas las columnas (timestamp no aparece) es necesario que especifiquemos las mismas después del **INSERT INTO**.

Para introducir datos nuevos en la tabla *transaction* es necesario que los nuevos valores introducidos en los campos *user_id*, *credit_card_id* y *company_id* estén registrados en sus **DIMENSION TABLES** correspondientes, ya que todos estos campos son **FOREIGN KEYS**, y sin existir en esas tablas no se pueden introducir.



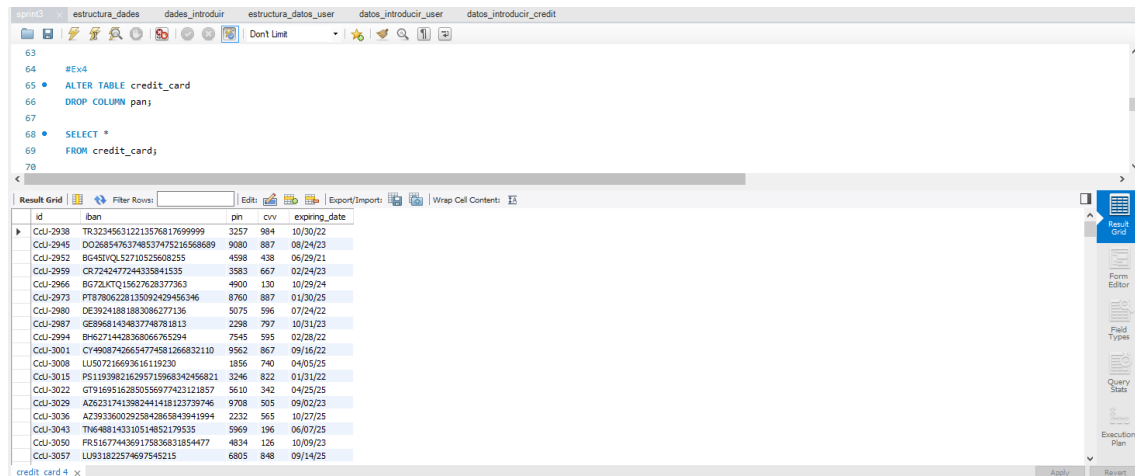
```
46
47 #Ex3
48 INSERT INTO credit_card (id)
49 VALUES ('CcU-9999');
50
51 INSERT INTO company (id)
52 VALUES ('b-9999');
53
54 INSERT INTO user (id)
55 VALUES (999);
56
57 INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
58 VALUES ('10881D1D-5B23-A76C-55EF-C568E49A9900', 'CcU-9999', 'b-9999', 999, 829.999, -117.999, 111.11, 0);
59
60 SELECT *
61 FROM transaction
62 WHERE id = '10881D1D-5B23-A76C-55EF-C568E49A9900';
63
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
10881D1D-5B23-A76C-55EF-C568E49A9900	CcU-9999	b-9999	999	829.999	-117.999	111.11	0	

Hacemos un **SELECT** para comprobar que se han introducido los datos.

Ex4

Para eliminar una columna de una tabla usamos el comando **ALTER TABLE – DROP COLUMN**. Después con un **SELECT** comprobamos que se ha eliminado la columna.



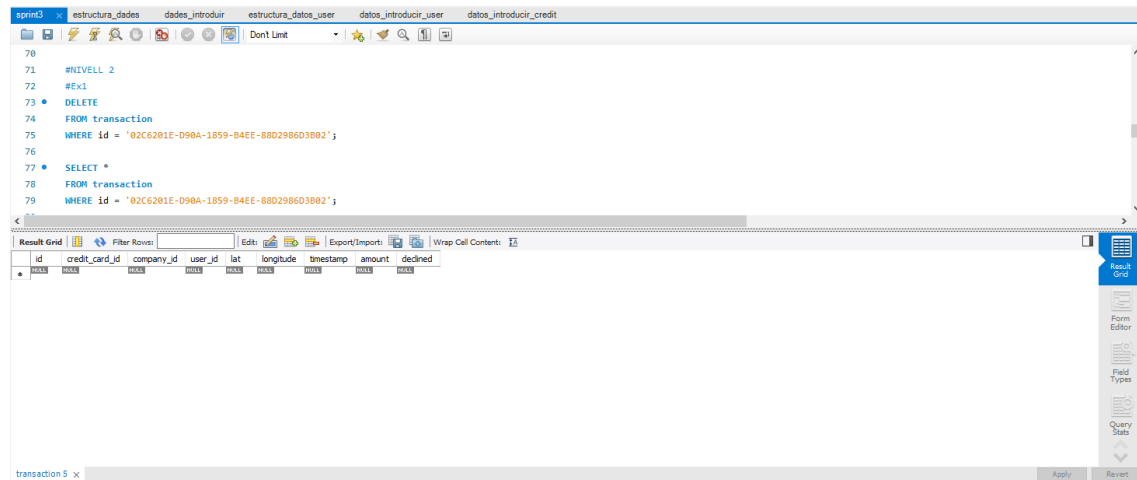
```
63
64 #Ex4
65 ALTER TABLE credit_card
66 DROP COLUMN pan;
67
68 SELECT *
69 FROM credit_card;
70
```

id	iban	pin	cvr	expiring_date
CcU-2938	TR323456312213576817699999	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	9080	887	08/24/23
CcU-2952	BC45VQL52710525603255	4998	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	BN4371442836866765204	7545	595	02/28/22
CcU-3001	CY4908742665474581266832110	9562	867	09/16/22
CcU-3008	LU507216693616119230	1856	740	04/05/25
CcU-3015	PS119398216295715968342456821	3246	822	01/31/22
CcU-3022	GT91695162850556977423121857	5610	342	04/25/25
CcU-3029	AJ52317413982441418123739746	9708	505	09/02/23
CcU-3036	AZ39330002925842865843941994	2232	565	10/27/25
CcU-3043	TN6488143310514852179535	5969	196	06/07/25
CcU-3050	FR5167744369175836831854477	4834	126	10/09/23
CcU-3057	LU931822574697545215	6805	848	09/14/25

NIVELL 2

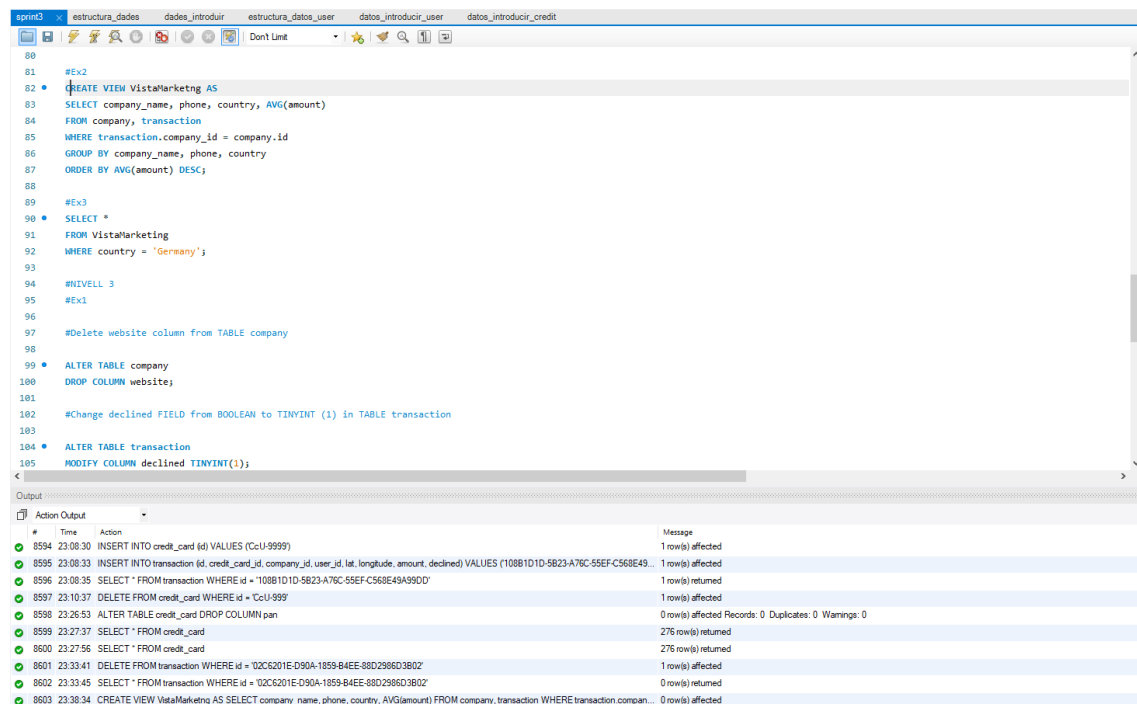
Ex1

Usamos el comando **DELETE** con la condición **WHERE id = id que queremos eliminar**. Se realiza un **SELECT** para comprobar que se ha eliminado.



Ex2

Para crear una **VIEW** usaremos el comando **CREATE VIEW AS**, con un **SELECT** donde introduciremos la selección que queremos para la **VIEW**.



Ex3

Para filtrar la **VIEW** que acabamos de crear utilizaremos una cláusula **WHERE**.

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

```
88
89 #Ex3
90 SELECT *
91 FROM VistaMarketing
92 WHERE country = 'Germany';
93
```

The results grid displays the following data:

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

NIVELL 3

Ex1

Los cambios necesarios a realizar son los siguientes:

- Delete website column from table *company*.
- Change declined **FIELD** from **BOOLEAN** to **TINYINT (1)** in table *transaction*.
- Change name of table *user* to *data_user*.
- Change name of field *email* to *personal_email* on table *data_user*.

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

```
93
94
95 #NIVELL 3
96 #Ex1
97
98 #Delete website column from TABLE company
99
100 ALTER TABLE company
101 DROP COLUMN website;
102
103 #Change declined FIELD from BOOLEAN to TINYINT (1) in TABLE transaction
104
105 ALTER TABLE transaction
106 MODIFY COLUMN declined TINYINT(1);
107
108 #Change name of TABLE user to data_user
109
110 ALTER TABLE user
111 RENAME TO data_user;
112
113 #Change name of FIELD email to personal_email on TABLE user
114
115 ALTER TABLE data_user
116 RENAME COLUMN email TO personal_email;
117
118 #Change id, iban, pin, expiring_date FIELDS VARCHAR lenght from TABLE credit_card
119 #Create FIELD fecha_actual DATE on TABLE credit_card
120
```

The output window shows the execution results of these commands:

#	Time	Action	Message
8605	23:41:01	CREATE VIEW VistaMarketing AS SELECT company_name, phone, country, AVG(amount) FROM company, transaction WHERE transaction.compan...	Error Code: 1050. Table 'VistaMarketing' already exists
8606	23:41:05	SELECT * FROM VistaMarketing WHERE country = 'Germany'	Error Code: 1146. Table 'transactions.vistamarketing' doesn't exist
8607	23:42:14	SELECT * FROM VistaMarketing WHERE country = 'Germany'	8 row(s) returned
8608	23:51:02	ALTER TABLE company DROP COLUMN website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
8609	23:51:20	ALTER TABLE transaction MODIFY COLUMN declined TINYINT(1)	0 row(s) affected. 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. Records: 0 Duplicate
8610	23:52:17	ALTER TABLE user RENAME TO data_user	0 row(s) affected
8611	23:52:25	ALTER TABLE data_user RENAME COLUMN email TO personal_email	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

- Change *id*, *iban*, *pin*, *expiring_date* fields **VARCHAR** lenght from table *credit_card*.
- Create field *fecha_actual* (data type **TIMESTAMP**) con **DEFAULT** fecha de hoy (**CURRENT_TIMESTAMP**).

```

117
118 #Change id, iban, pin, expiring_date FIELDS VARCHAR lenght from TABLE credit_card
119 #Create FIELD fecha_actual DATE on TABLE credit_card
120
121 • ALTER TABLE credit_card
122   MODIFY COLUMN id VARCHAR (20);
123
124 • ALTER TABLE credit_card
125   MODIFY COLUMN iban VARCHAR (50);
126
127 • ALTER TABLE credit_card
128   DROP COLUMN pan;
129
130 • ALTER TABLE credit_card
131   MODIFY COLUMN pin VARCHAR (4);
132
133 • ALTER TABLE credit_card
134   MODIFY COLUMN cvv INT;
135
136 • ALTER TABLE credit_card
137   MODIFY COLUMN expiring_date VARCHAR (10);
138
139 • ALTER TABLE credit_card
140   ADD COLUMN fecha_actual TIMESTAMP DEFAULT CURRENT_TIMESTAMP;
141
142 #Ex2
143 • CREATE VIEW InformeTecnico AS
144   SELECT transaction.id, user.name, user.surname, credit_card.iban, company.company_name, transaction.declined

```

#	Time	Action	Message
8613	23:59:22	ALTER TABLE credit_card MODIFY COLUMN iban VARCHAR (50)	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
8614	23:59:27	ALTER TABLE credit_card DROP COLUMN pan	Error Code: 1091. Can't DROP 'pan'; check that column/key exists
8615	23:59:36	ALTER TABLE credit_card MODIFY COLUMN pin VARCHAR (4)	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
8616	23:59:40	ALTER TABLE credit_card MODIFY COLUMN cvv INT	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
8617	23:59:44	ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR (10)	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
8618	23:59:50	ALTER TABLE credit_card ADD COLUMN fecha_actual TIMESTAMP DEFAULT CURRENT_TIMESTAMP	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
8619	00:00:18	SELECT * FROM credit_card	276 row(s) returned

Ex2

Finalmente creamos una **VIEW** con las condiciones especificadas.

```

142 #Ex2
143 • CREATE VIEW InformeTecnico AS
144   SELECT transaction.id, data_user.name, data_user.surname, credit_card.iban, company.company_name, transaction.declined
145   FROM company, transaction, data_user, credit_card
146   WHERE transaction.company_id = company.id AND transaction.user_id = data_user.id AND transaction.credit_card_id = credit_card.id;
147
148
149

```

#	Time	Action	Message
8615	23:59:36	ALTER TABLE credit_card MODIFY COLUMN pin VARCHAR (4)	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
8616	23:59:40	ALTER TABLE credit_card MODIFY COLUMN cvv INT	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
8617	23:59:44	ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR (10)	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0
8618	23:59:50	ALTER TABLE credit_card ADD COLUMN fecha_actual TIMESTAMP DEFAULT CURRENT_TIMESTAMP	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
8619	00:00:18	SELECT * FROM credit_card	276 row(s) returned
8620	00:02:57	CREATE VIEW InformeTecnico AS SELECT transaction.id, user.name, user.surname, credit_card.iban, company.company_name, transaction.declined...	Error Code: 1146. Table 'transactions.user' doesn't exist
8621	00:03:49	CREATE VIEW InformeTecnico AS SELECT transaction.id, data_user.name, data_user.surname, credit_card.iban, company.company_name, transac...	0 row(s) affected