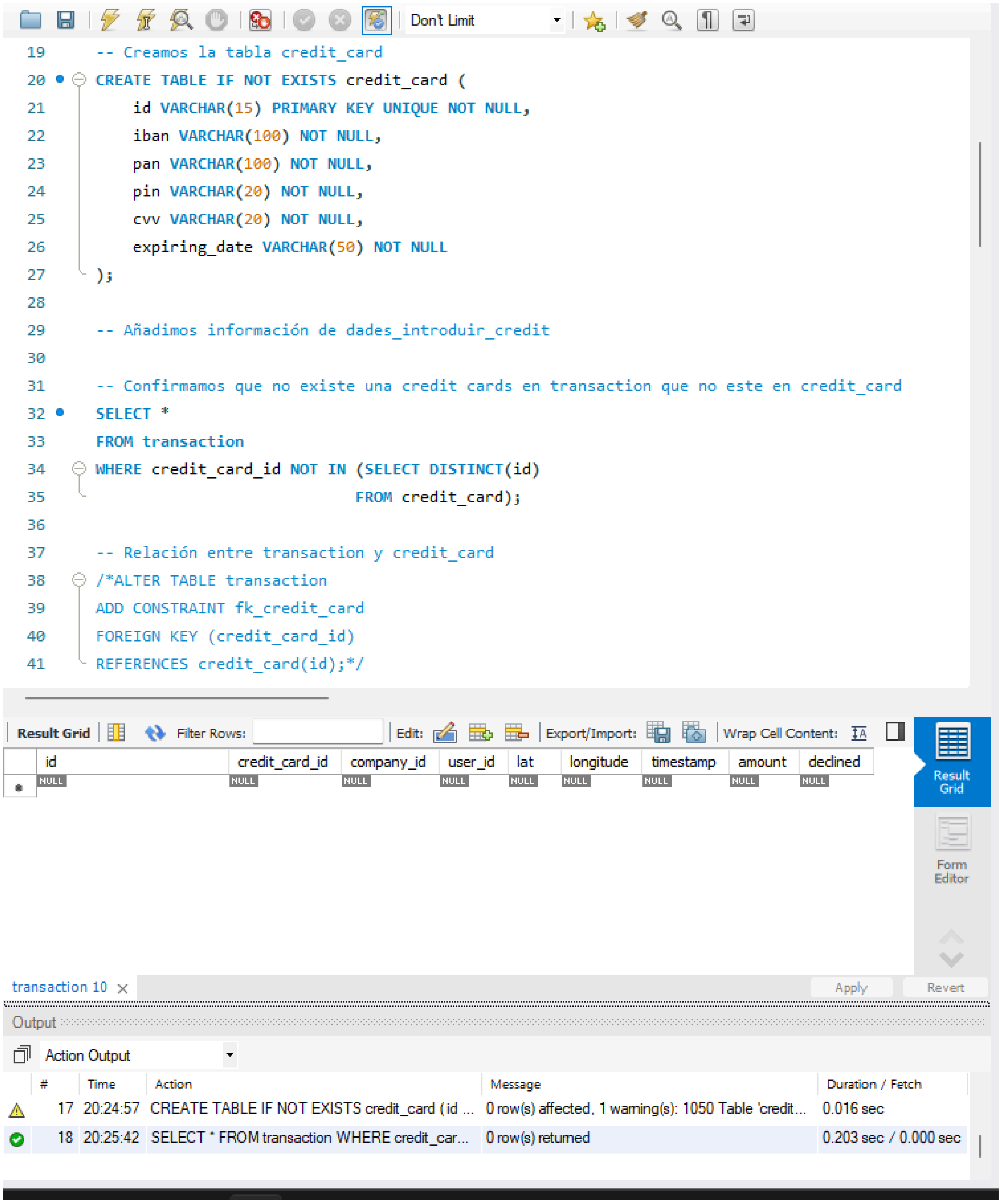
<u>Tasca S3.01. Manipulació de taules</u>

Nivell 1

Exercici 1

En esta tarea, se elaboró la tabla credit_card. Viendo los valores escritos en el archivo dades_introduir_credit, se decidió que todas las columnas fueran de tipo VARCHAR. Una vez creada la tabla, se creó la relación con la tabla transaction.

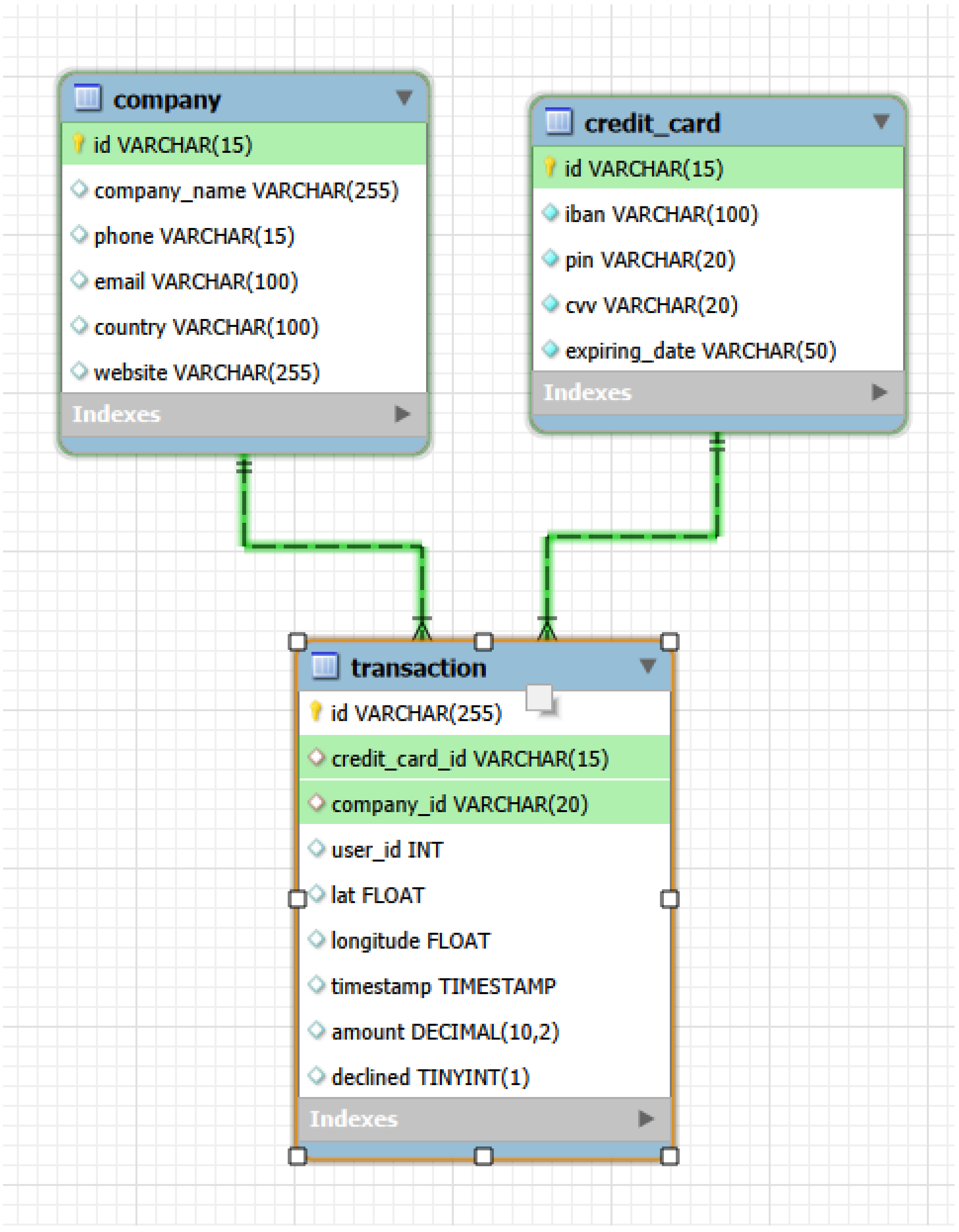


```sql
-- Creamos la tabla credit_card
CREATE TABLE IF NOT EXISTS credit_card (
    id VARCHAR(15) PRIMARY KEY UNIQUE NOT NULL,
    iban VARCHAR(100) NOT NULL,
    pan VARCHAR(100) NOT NULL,
    pin VARCHAR(20) NOT NULL,
    cvv VARCHAR(20) NOT NULL,
    expiring_date VARCHAR(50) NOT NULL
);

-- Añadimos información de dades_introduir_credit

-- Confirmamos que no existe una credit cards en transaction que no este en credit_card
SELECT *
FROM transaction
WHERE credit_card_id NOT IN (SELECT DISTINCT(id)
                             FROM credit_card);

-- Relación entre transaction y credit_card
/*ALTER TABLE transaction
ADD CONSTRAINT fk_credit_card
FOREIGN KEY (credit_card_id)
REFERENCES credit_card(id);*/
```

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

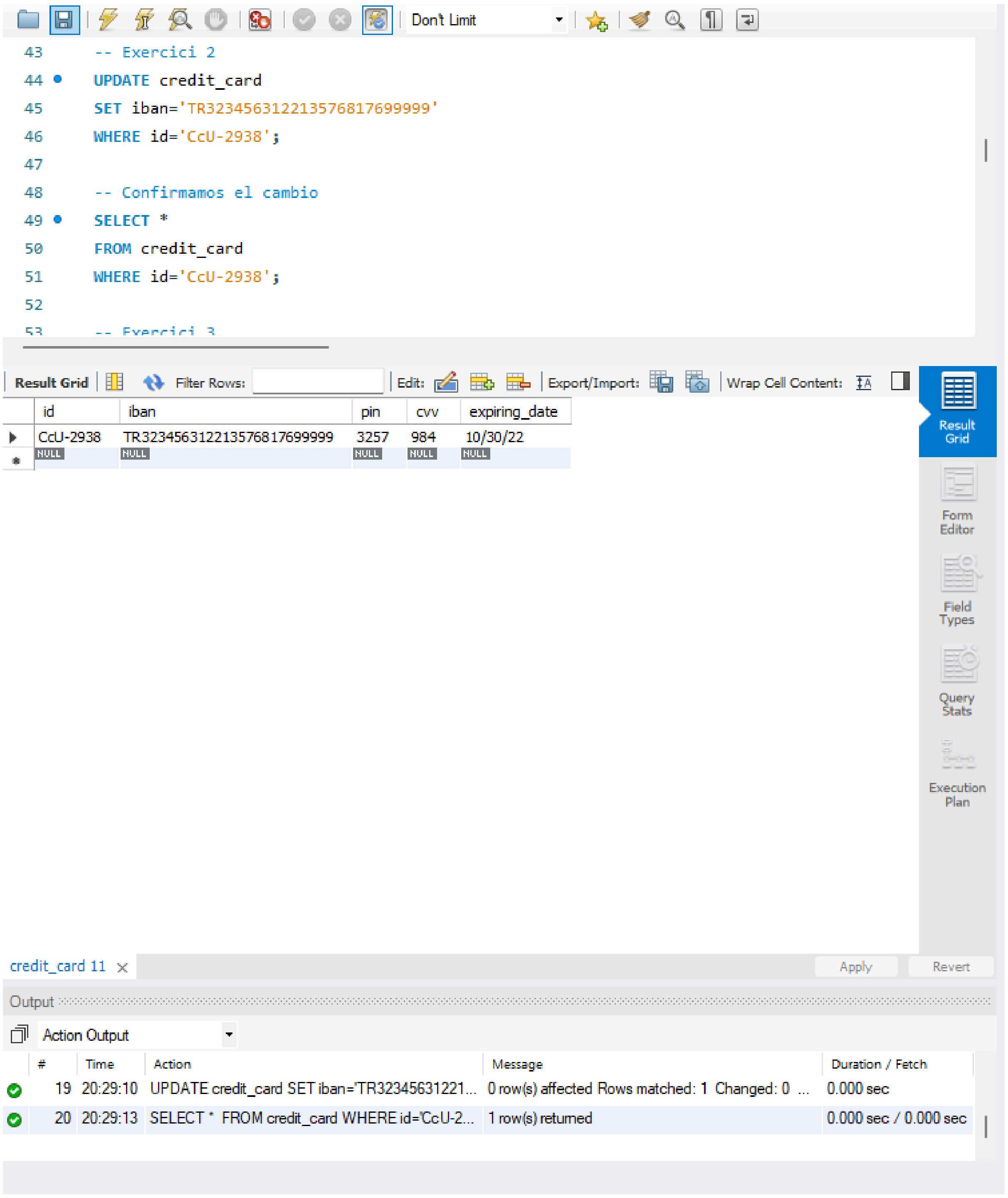
#	Time	Action	Message	Duration / Fetch
17	20:24:57	CREATE TABLE IF NOT EXISTS credit_card (id ...	0 row(s) affected, 1 warning(s): 1050 Table 'credit...	0.016 sec
18	20:25:42	SELECT * FROM transaction WHERE credit_car...	0 row(s) returned	0.203 sec / 0.000 sec

En la siguiente figura se muestran las relaciones entre las tres tablas. Podemos observar una relación n-to-1 entre las tablas transaction y company, unidas en transaction.company_id y company.id. Esto se explica a que una misma compañía puede haber realizado diversas transacciones con la empresa dedicada a la venta de productos en línea. También observamos la relación n-to-1 entre las tablas transaction y credit_card, unidas en transaction.credit_card_id y credit_card.id. Esto se explica a que una misma tarjeta de crédito puede usarse en más de una transacción.



Exercici 2

Cambiamos el número de cuenta.



```
-- Exercici 2
UPDATE credit_card
SET iban='TR323456312213576817699999'
WHERE id='CcU-2938';

-- Confirmamos el cambio
SELECT *
FROM credit_card
WHERE id='CcU-2938';

-- Exercici 3
```

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

credit_card 11

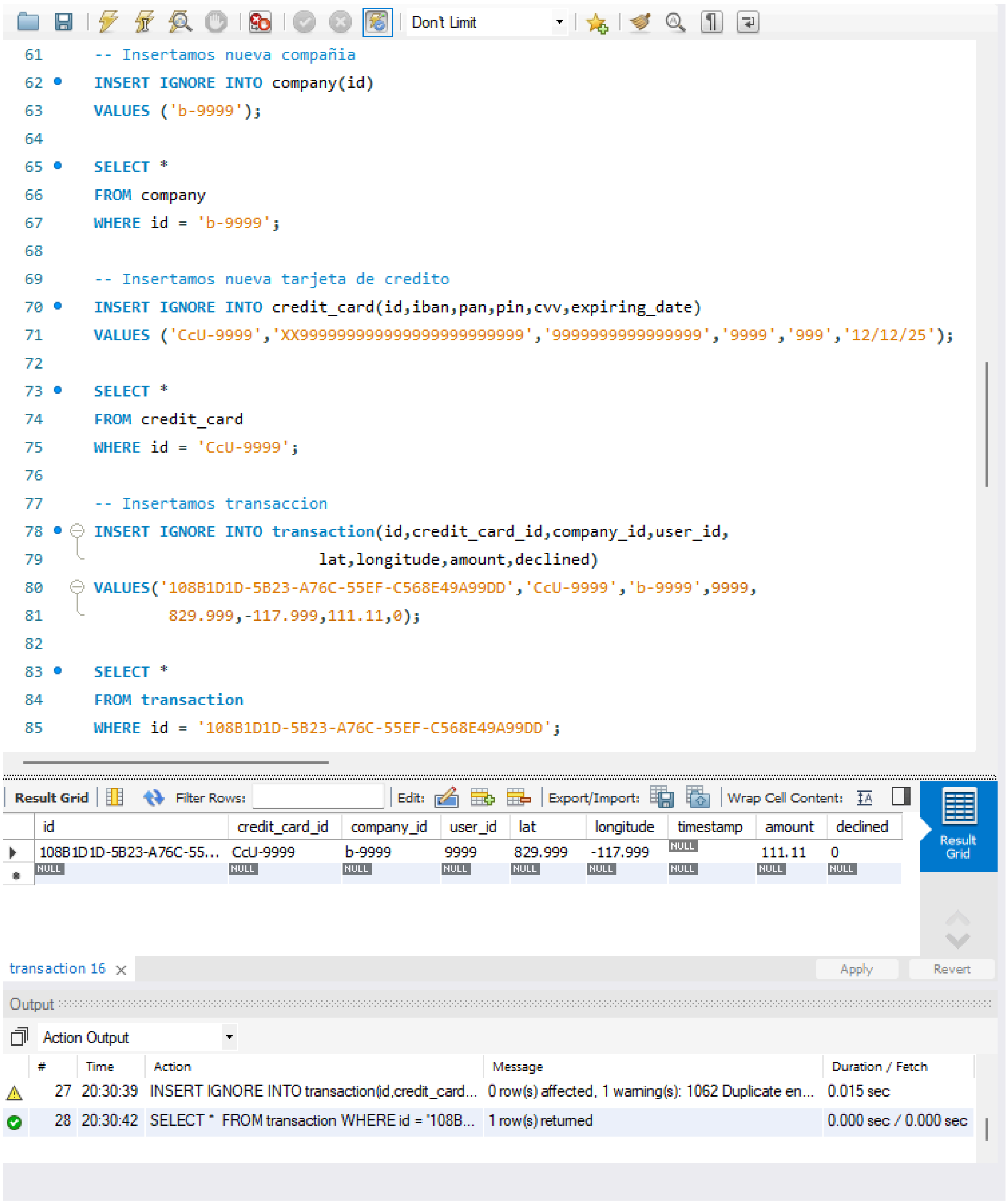
Output

Action Output

```
#   Time      Action                                        Message                                            Duration / Fetch
19  20:29:10  UPDATE credit_card SET iban='TR32345631221...  0 row(s) affected Rows matched: 1  Changed: 0  ...  0.000 sec
20  20:29:13  SELECT * FROM credit_card WHERE id='CcU-2...   1 row(s) returned                                  0.000 sec / 0.000 sec
```

Exercici 3

Para ingresar la nueva transacción, nos dimos cuenta que teníamos que agregar una nueva empresa y una nueva tarjeta de crédito a sus respectivas tablas, debido a que existen relaciones con keys entre las tres tablas. Primero agregamos la compañía de id ‘b-9999’, luego la tarjeta de crédito de id ‘CcU-9999’ y finalmente la nueva transacción.



```sql
-- Insertamos nueva compañia
INSERT IGNORE INTO company(id)
VALUES ('b-9999');

SELECT *
FROM company
WHERE id = 'b-9999';

-- Insertamos nueva tarjeta de credito
INSERT IGNORE INTO credit_card(id,iban,pan,pin,cvv,expiring_date)
VALUES ('CcU-9999','XX999999999999999999999999','9999999999999999','9999','999','12/12/25');

SELECT *
FROM credit_card
WHERE id = 'CcU-9999';

-- Insertamos transaccion
INSERT IGNORE 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);

SELECT *
FROM transaction
WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';
```

Result Grid

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55...	CcU-9999	b-9999	9999	829.999	-117.999	NULL	111.11	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 16

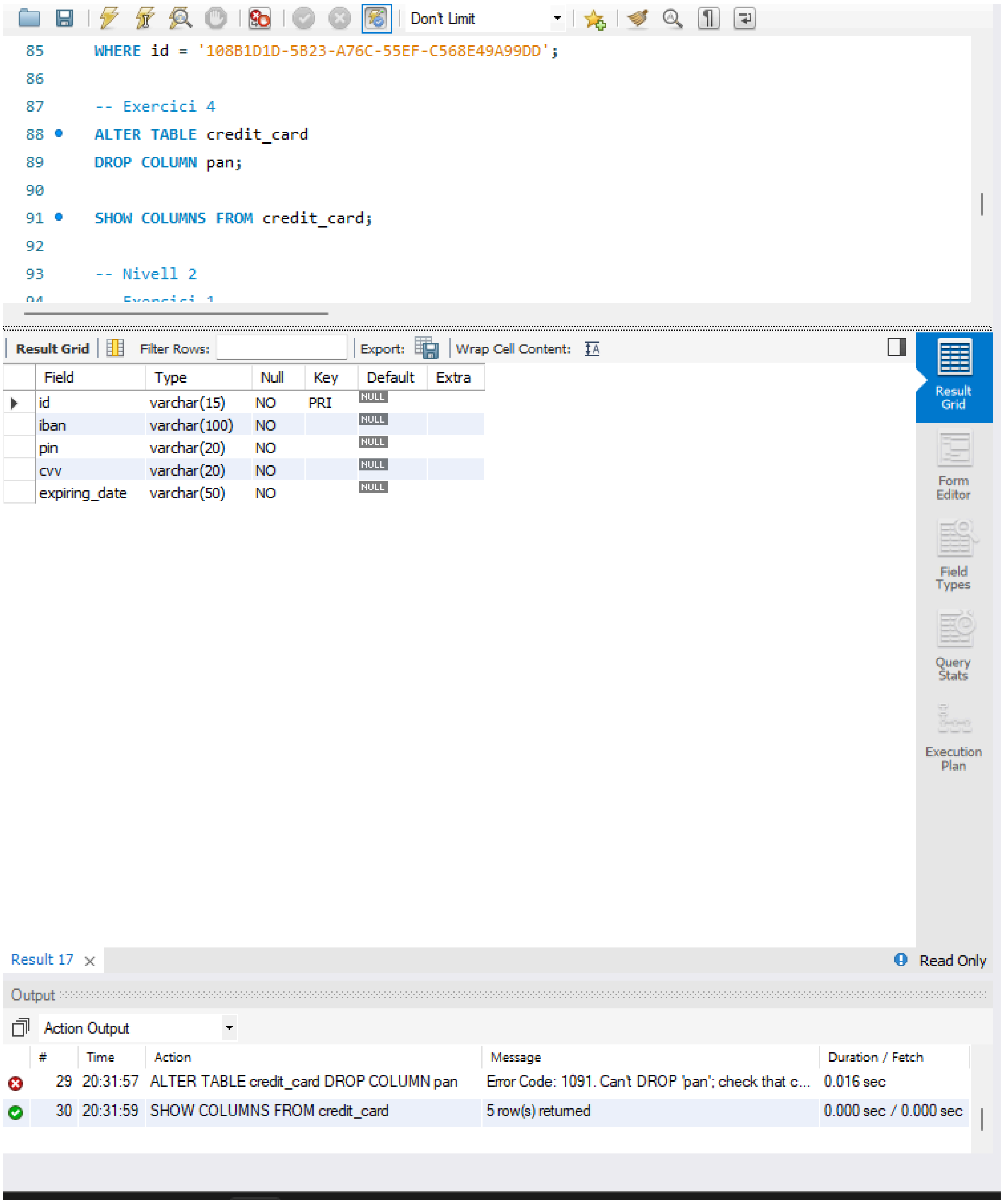
Output

Action Output

	#	Time	Action	Message	Duration / Fetch
⚠	27	20:30:39	INSERT IGNORE INTO transaction(id,credit_card...	0 row(s) affected, 1 warning(s): 1062 Duplicate en...	0.015 sec
✓	28	20:30:42	SELECT * FROM transaction WHERE id = '108B...	1 row(s) returned	0.000 sec / 0.000 sec

Exercici 4

Eliminamos la columna ‘pan’ de la tabla credit_card.



```
WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';

-- Exercici 4
ALTER TABLE credit_card
DROP COLUMN pan;

SHOW COLUMNS FROM credit_card;

-- Nivell 2
```

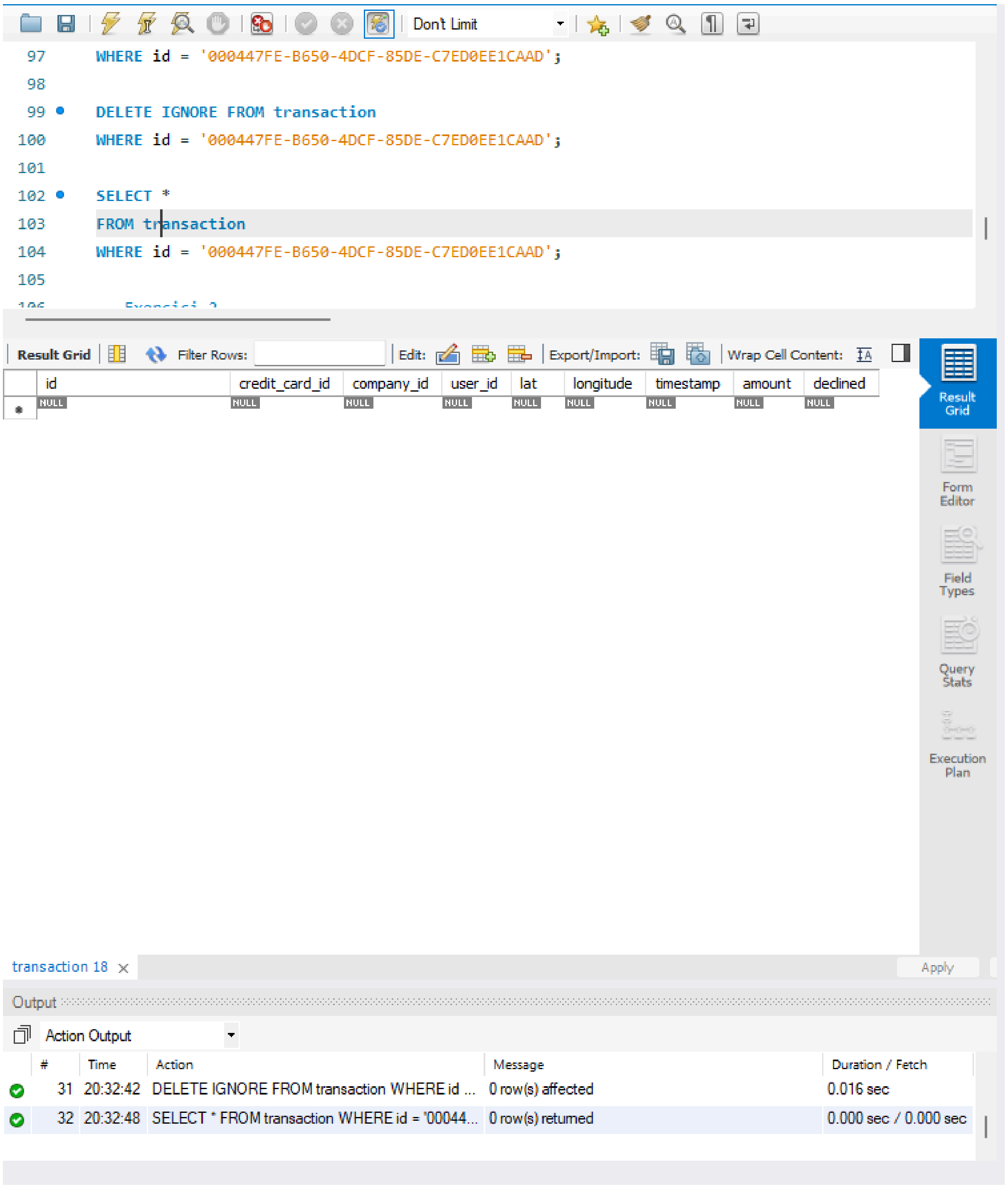
Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	NULL	
iban	varchar(100)	NO		NULL	
pin	varchar(20)	NO		NULL	
cvv	varchar(20)	NO		NULL	
expiring_date	varchar(50)	NO		NULL	

#	Time	Action	Message	Duration / Fetch
29	20:31:57	ALTER TABLE credit_card DROP COLUMN pan	Error Code: 1091. Can't DROP 'pan'; check that c...	0.016 sec
30	20:31:59	SHOW COLUMNS FROM credit_card	5 row(s) returned	0.000 sec / 0.000 sec

Nivell 2

Exercici 1

Eliminamos la transacción indicada.



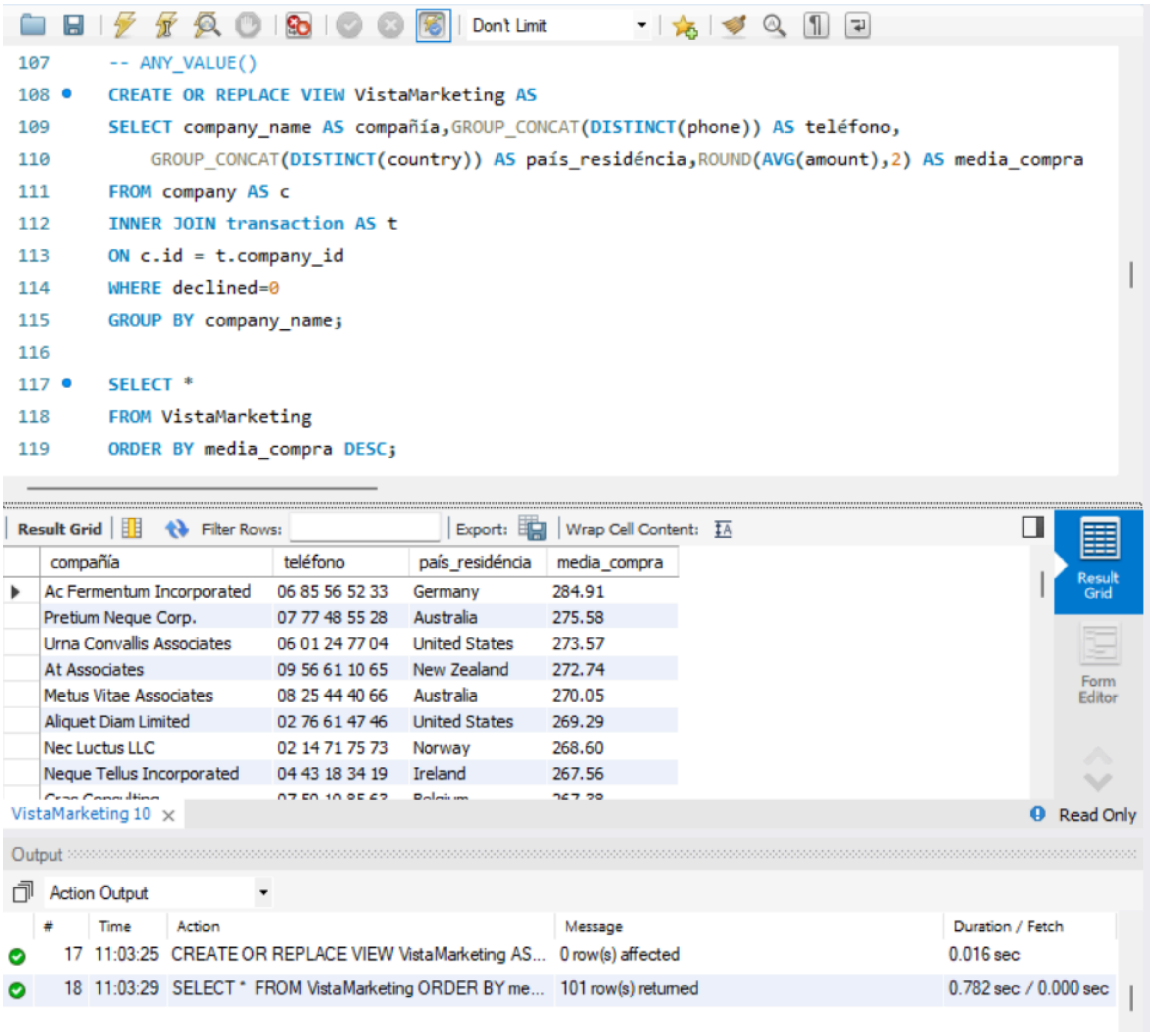
```
WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';

DELETE IGNORE FROM transaction
WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';

SELECT *
FROM transaction
WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
```

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

```
#   Time      Action                                        Message               Duration / Fetch
31  20:32:42  DELETE IGNORE FROM transaction WHERE id ...   0 row(s) affected     0.016 sec
32  20:32:48  SELECT * FROM transaction WHERE id = '00044...  0 row(s) returned   0.000 sec / 0.000 sec
```

Exercici 2

Elaboramos la vista que se nos pedía. Utilizamos un INNER JOIN para relacionar las tablas y para poder hacer el GROUP BY con las otras columnas, decidimos utilizar GROUP_CONCAT(DISTINCT()). Solo consideramos las transacciones donde declined=0.



```
-- ANY_VALUE()
CREATE OR REPLACE VIEW VistaMarketing AS
SELECT company_name AS compañía,GROUP_CONCAT(DISTINCT(phone)) AS teléfono,
    GROUP_CONCAT(DISTINCT(country)) AS país_residéncia,ROUND(AVG(amount),2) AS media_compra
FROM company AS c
INNER JOIN transaction AS t
ON c.id = t.company_id
WHERE declined=0
GROUP BY company_name;

SELECT *
FROM VistaMarketing
ORDER BY media_compra DESC;
```

compañía	teléfono	país_residéncia	media_compra
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.91
Pretium Neque Corp.	07 77 48 55 28	Australia	275.58
Urna Convallis Associates	06 01 24 77 04	United States	273.57
At Associates	09 56 61 10 65	New Zealand	272.74
Metus Vitae Associates	08 25 44 40 66	Australia	270.05
Aliquet Diam Limited	02 76 61 47 46	United States	269.29
Nec Luctus LLC	02 14 71 75 73	Norway	268.60
Neque Tellus Incorporated	04 43 18 34 19	Ireland	267.56

VistaMarketing 10 Read Only

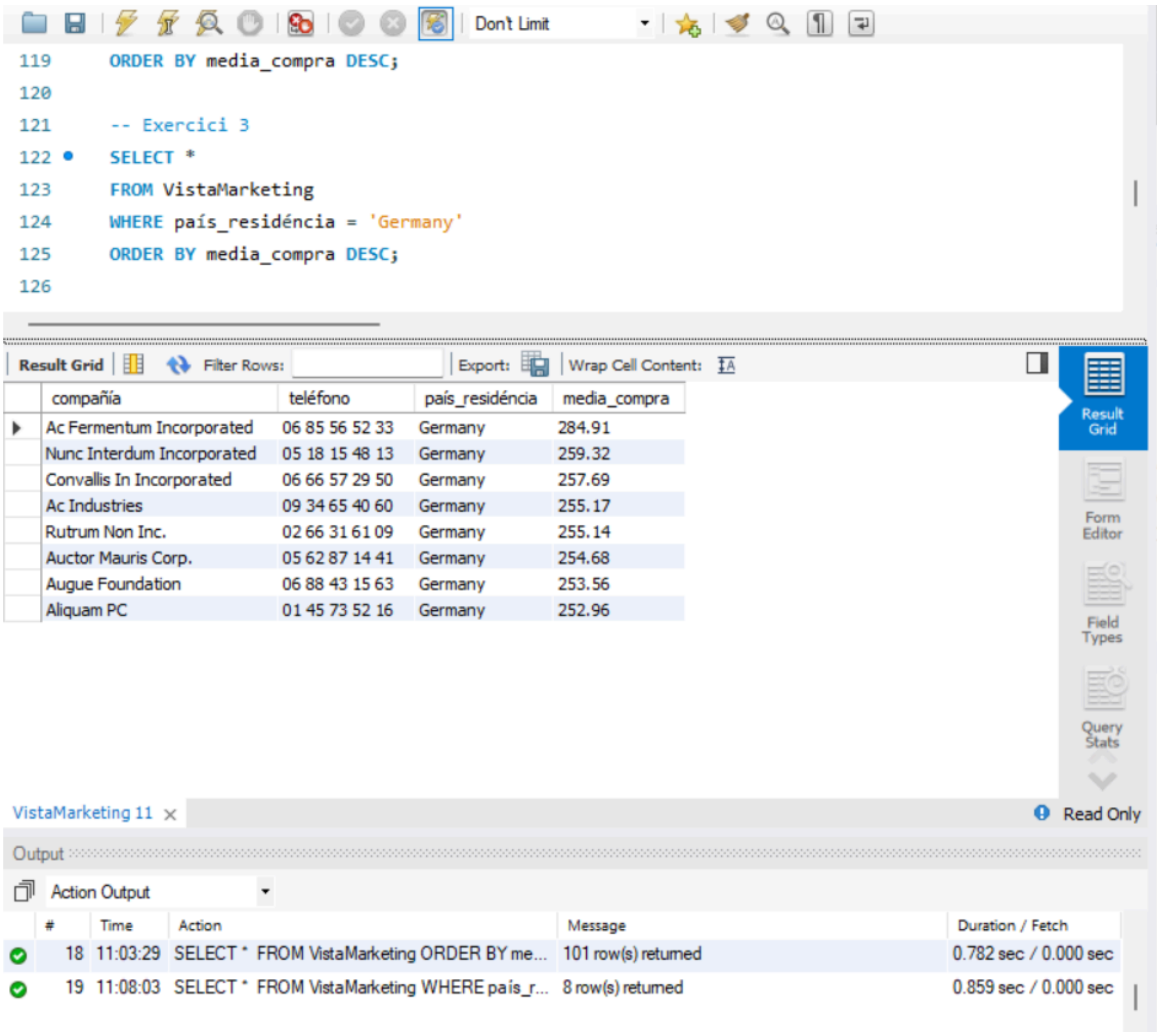
Output

Action Output

#	Time	Action	Message	Duration / Fetch
17	11:03:25	CREATE OR REPLACE VIEW VistaMarketing AS...	0 row(s) affected	0.016 sec
18	11:03:29	SELECT * FROM VistaMarketing ORDER BY me...	101 row(s) returned	0.782 sec / 0.000 sec

Exercici 3

Filtramos la vista para mostrar solo las compañías con país 'Germany'.



```
ORDER BY media_compra DESC;

-- Exercici 3
SELECT *
FROM VistaMarketing
WHERE país_residéncia = 'Germany'
ORDER BY media_compra DESC;
```

compañía	teléfono	país_residéncia	media_compra
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
Rutrum 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

VistaMarketing 11   Read Only

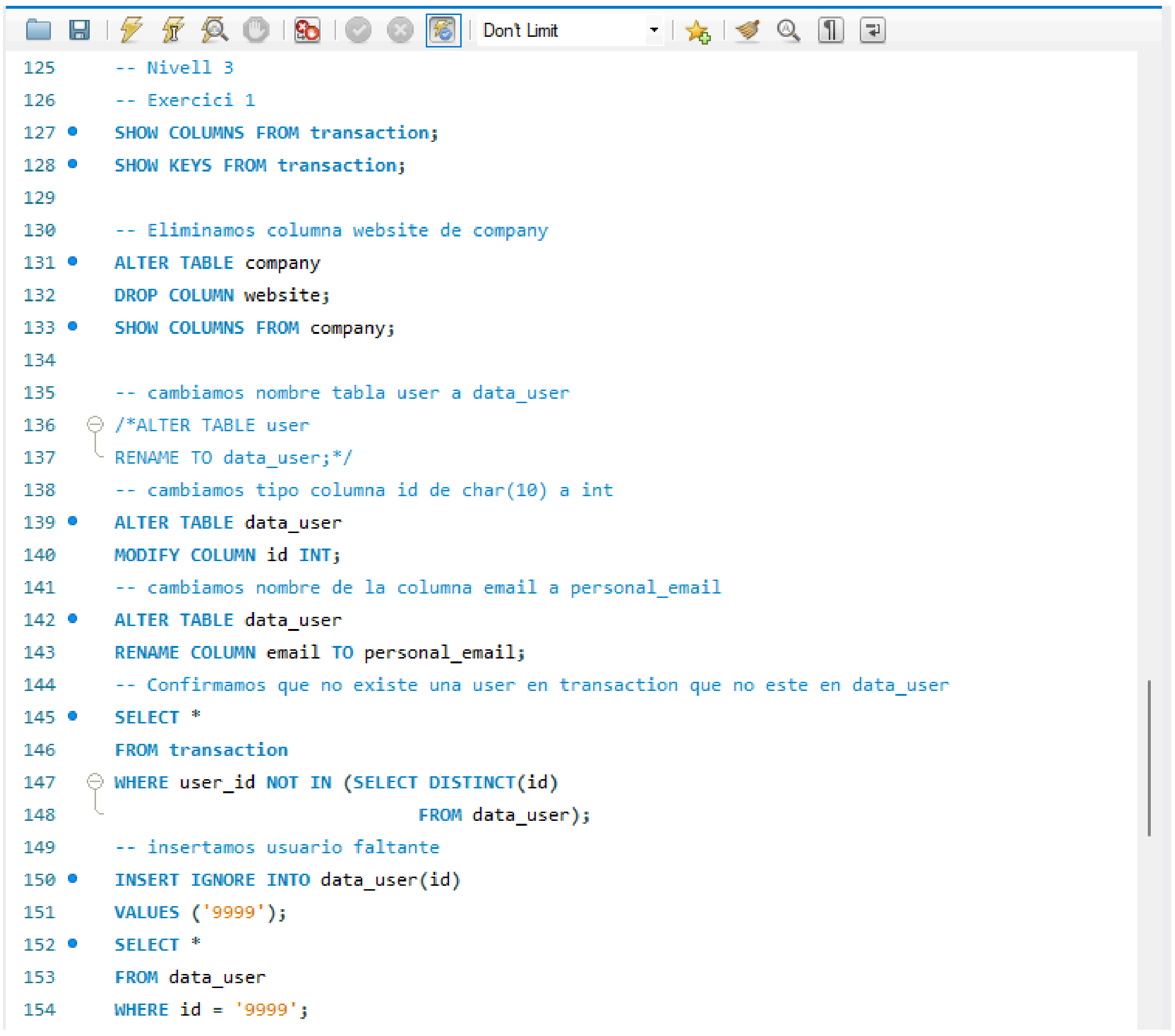
Output: Action Output

#	Time	Action	Message	Duration / Fetch
18	11:03:29	SELECT * FROM VistaMarketing ORDER BY me...	101 row(s) returned	0.782 sec / 0.000 sec
19	11:08:03	SELECT * FROM VistaMarketing WHERE país_r...	8 row(s) returned	0.859 sec / 0.000 sec

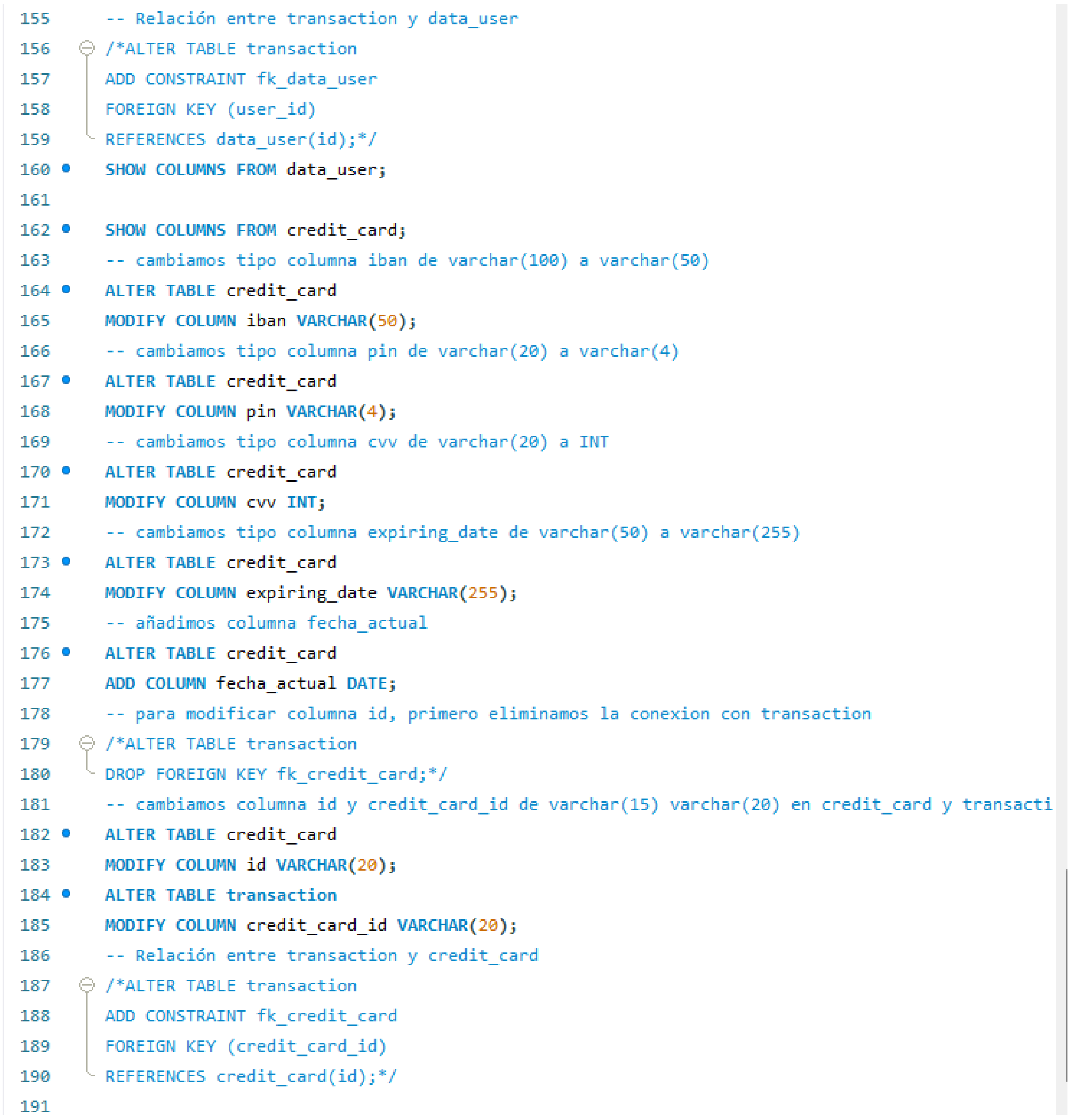
Nivell 3

Exercici 1

Realizamos diversos cambios para obtener el diagrama indicado.

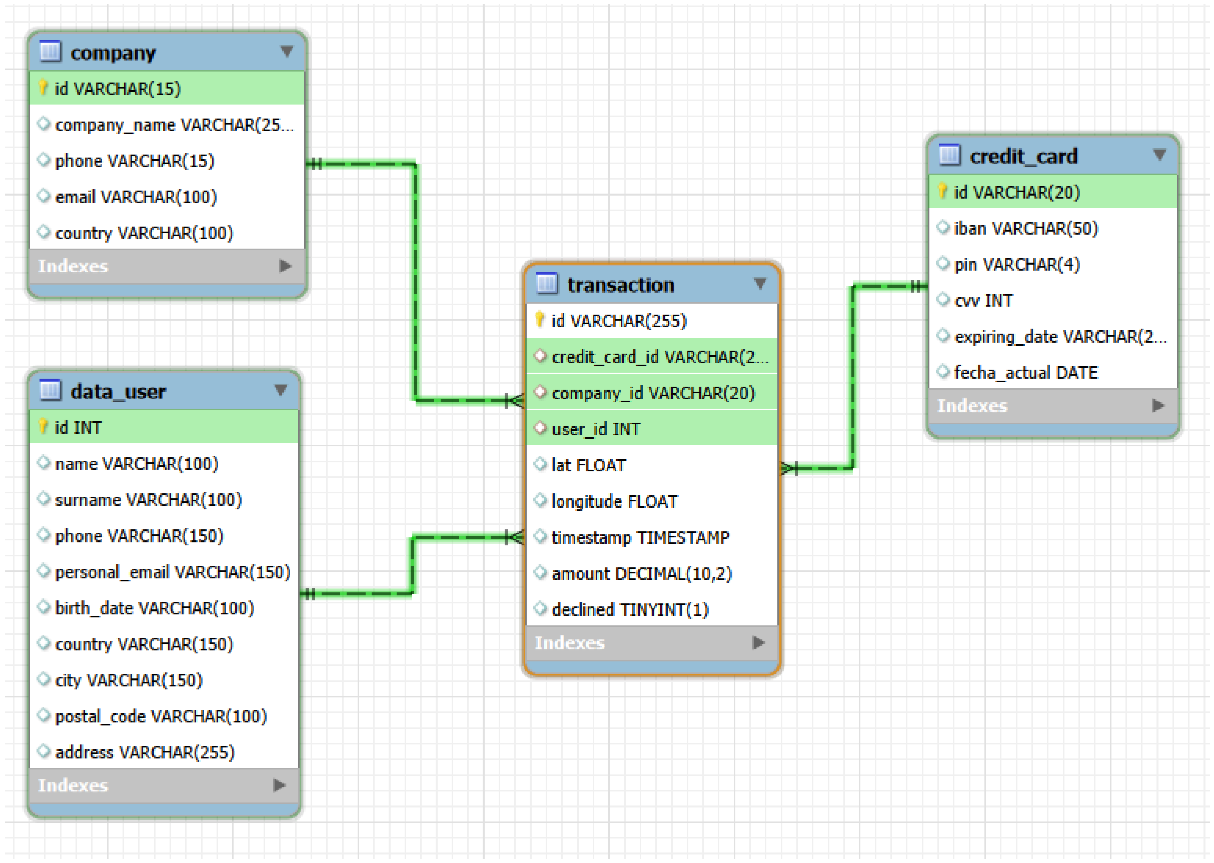


```sql
-- Nivell 3
-- Exercici 1
SHOW COLUMNS FROM transaction;
SHOW KEYS FROM transaction;

-- Eliminamos columna website de company
ALTER TABLE company
DROP COLUMN website;
SHOW COLUMNS FROM company;

-- cambiamos nombre tabla user a data_user
/*ALTER TABLE user
RENAME TO data_user;*/
-- cambiamos tipo columna id de char(10) a int
ALTER TABLE data_user
MODIFY COLUMN id INT;
-- cambiamos nombre de la columna email a personal_email
ALTER TABLE data_user
RENAME COLUMN email TO personal_email;
-- Confirmamos que no existe una user en transaction que no este en data_user
SELECT *
FROM transaction
WHERE user_id NOT IN (SELECT DISTINCT(id)
                            FROM data_user);
-- insertamos usuario faltante
INSERT IGNORE INTO data_user(id)
VALUES ('9999');
SELECT *
FROM data_user
WHERE id = '9999';
```



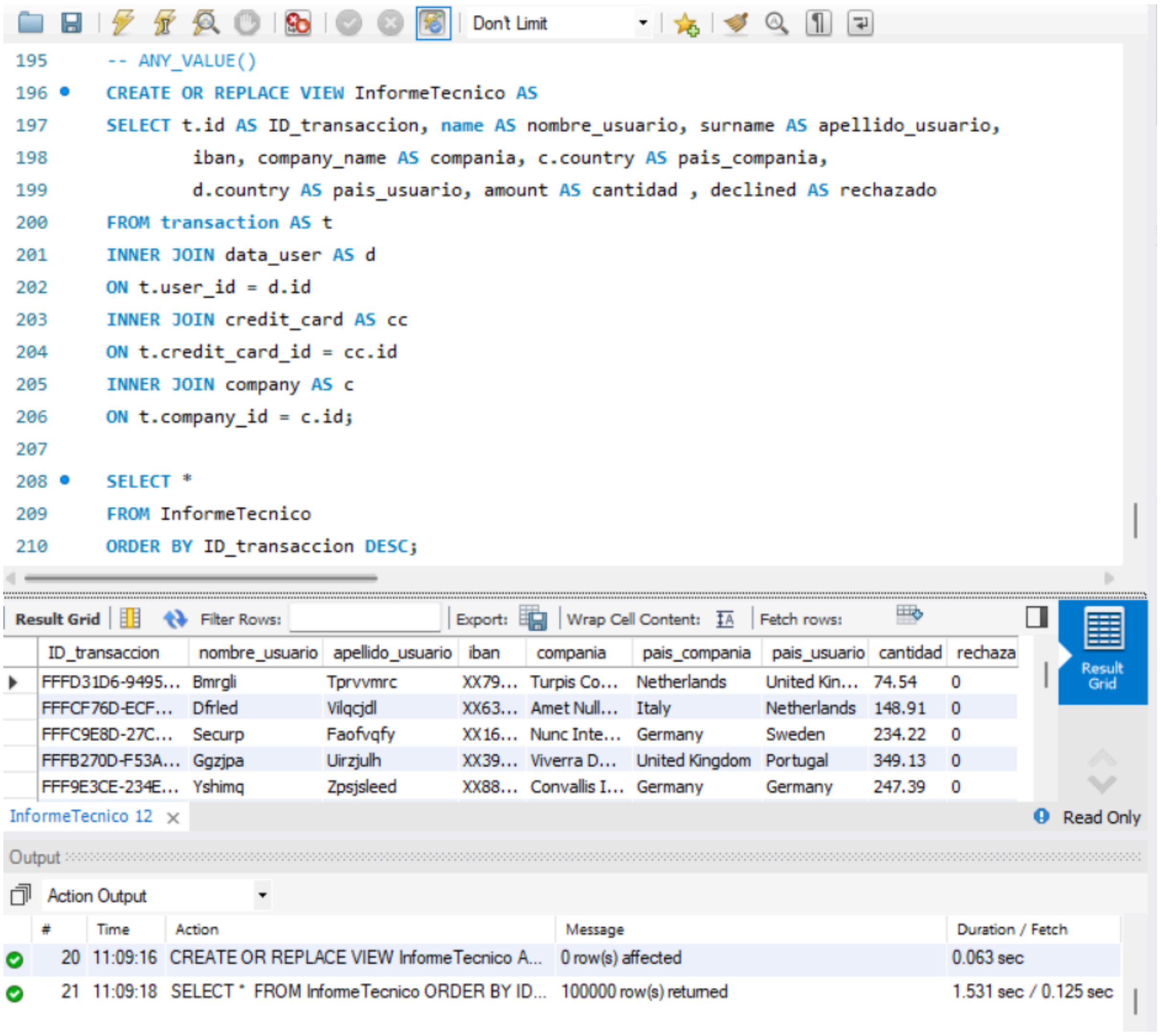
```
-- Relación entre transaction y data_user
/*ALTER TABLE transaction
ADD CONSTRAINT fk_data_user
FOREIGN KEY (user_id)
REFERENCES data_user(id);*/
SHOW COLUMNS FROM data_user;

SHOW COLUMNS FROM credit_card;
-- cambiamos tipo columna iban de varchar(100) a varchar(50)
ALTER TABLE credit_card
MODIFY COLUMN iban VARCHAR(50);
-- cambiamos tipo columna pin de varchar(20) a varchar(4)
ALTER TABLE credit_card
MODIFY COLUMN pin VARCHAR(4);
-- cambiamos tipo columna cvv de varchar(20) a INT
ALTER TABLE credit_card
MODIFY COLUMN cvv INT;
-- cambiamos tipo columna expiring_date de varchar(50) a varchar(255)
ALTER TABLE credit_card
MODIFY COLUMN expiring_date VARCHAR(255);
-- añadimos columna fecha_actual
ALTER TABLE credit_card
ADD COLUMN fecha_actual DATE;
-- para modificar columna id, primero eliminamos la conexion con transaction
/*ALTER TABLE transaction
DROP FOREIGN KEY fk_credit_card;*/
-- cambiamos columna id y credit_card_id de varchar(15) varchar(20) en credit_card y transacti
ALTER TABLE credit_card
MODIFY COLUMN id VARCHAR(20);
ALTER TABLE transaction
MODIFY COLUMN credit_card_id VARCHAR(20);
-- Relación entre transaction y credit_card
/*ALTER TABLE transaction
ADD CONSTRAINT fk_credit_card
FOREIGN KEY (credit_card_id)
REFERENCES credit_card(id);*/

```

Tras los cambios indicados, obtuvimos el siguiente diagrama.



Exercici 2

Creamos la vista indicada, usando múltiples INNER JOIN para unir las tablas. Por nuestra interpretación del enunciado, decidimos también agregar las columnas país de la compañía, país del usuario, cantidad de la transacción y si fue rechazada o no.



```sql
-- ANY_VALUE()
CREATE OR REPLACE VIEW InformeTecnico AS
SELECT t.id AS ID_transaccion, name AS nombre_usuario, surname AS apellido_usuario,
        iban, company_name AS compania, c.country AS pais_compania,
        d.country AS pais_usuario, amount AS cantidad , declined AS rechazado
FROM transaction AS t
INNER JOIN data_user AS d
ON t.user_id = d.id
INNER JOIN credit_card AS cc
ON t.credit_card_id = cc.id
INNER JOIN company AS c
ON t.company_id = c.id;

SELECT *
FROM InformeTecnico
ORDER BY ID_transaccion DESC;
```

ID_transaccion	nombre_usuario	apellido_usuario	iban	compania	pais_compania	pais_usuario	cantidad	rechaza
FFFD31D6-9495...	Bmrgli	Tprvvmrc	XX79...	Turpis Co...	Netherlands	United Kin...	74.54	0
FFFCF76D-ECF...	Dfrled	Vilqcjdl	XX63...	Amet Null...	Italy	Netherlands	148.91	0
FFFC9E8D-27C...	Securp	Faofvqfy	XX16...	Nunc Inte...	Germany	Sweden	234.22	0
FFFB270D-F53A...	Ggzjpa	Uirzjulh	XX39...	Viverra D...	United Kingdom	Portugal	349.13	0
FFF9E3CE-234E...	Yshimq	Zpsjsleed	XX88...	Convallis I...	Germany	Germany	247.39	0

InformeTecnico 12 — Read Only

Output — Action Output

#	Time	Action	Message	Duration / Fetch
20	11:09:16	CREATE OR REPLACE VIEW InformeTecnico A...	0 row(s) affected	0.063 sec
21	11:09:18	SELECT * FROM InformeTecnico ORDER BY ID...	100000 row(s) returned	1.531 sec / 0.125 sec

Revisión peer-to-peer
Revisado por Rubén Serra

- Al hacer la vista, tener en cuenta que puede ser importante filtrar por declined.
- En los ejercicios de la vista, poner el ORDER BY fuera de la creación de la vista.
- Cuando se haga el cambio de tipo en diversas columnas de una tabla, hacer todo al mismo tiempo en un MODIFY, para mayor eficiencia.