



# Manipulación de tablas S3.01

Elian Daghoum Dorado



# Índice

Nivel 1	03
Nivel 2	08
Nivel 3	12



## Nivel 1

### Ejercicio 1

Tu tarea es diseñar y crear una tabla llamada "credit\_card" que almacene detalles cruciales sobre las tarjetas de crédito. La nueva tabla debe ser capaz de identificar de manera única cada tarjeta y establecer una relación adecuada con las otras dos tablas ("transaction" y "company"). Después de crear la tabla será necesario que ingreses la información del documento denominado "dades\_introduir\_credit". Recuerda mostrar el diagrama y realizar una breve descripción del mismo.

1/ Creamos la tabla siguiendo los datos que nos han dado

```
CREATE TABLE IF NOT EXISTS credit_card (  
  id VARCHAR(20) PRIMARY KEY,  
  iban VARCHAR(34) NOT NULL,  
  pan VARCHAR(16) NOT NULL,  
  pin CHAR(4) NOT NULL,  
  cvv CHAR(3) NOT NULL,  
  expiring_date VARCHAR(10) NOT NULL  
);
```

2/ Conectamos la tabla credit\_card a transaction creando la FK

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



## Nivel 1

### Ejercicio 1

3/ Revisamos que se ha creado correctamente

```
SELECT * FROM credit_card;
```

	id	iban	pan	pin
►	CcS-4857	XX4857591835292505850771	2314242385113924	1819
	CcS-4858	XX8581768137002436094025	6582720299715533	3964
	CcS-4859	XX7826930491423553609370	8861684536289642	4983
	CcS-4860	XX5559590368835304645299	2481155515498459	6876
	CcS-4861	XX2035182877195191627307	1308930301149557	5710
	CcS-4862	XX4774721462463645409758	6715617009807829	4042
	CcS-4863	XX1476829664245046207111	3140879819451394	5969
	CcS-4864	XX8380298893385731196159	5793672133649114	8481
	CcS-4865	XX7085078596101025280599	5101552687251312	7847



## Nivel 1

## Ejercicio 1

```
4 CREATE TABLE IF NOT EXISTS credit_card (  
5     id VARCHAR(20) PRIMARY KEY,  
6     iban VARCHAR(34) NOT NULL,  
7     pan VARCHAR(16) NOT NULL,  
8     pin CHAR(4) NOT NULL,  
9     cvv CHAR(3) NOT NULL,  
10    expiring_date VARCHAR(10) NOT NULL  
11 );  
12  
13 -- generando conexiones  
14  
15 ALTER TABLE transaction  
16 ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);  
17  
18 SELECT * FROM credit_card;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap C

	id	iban	pin	cvv	expiring_date
▶	CcS-4857	XX4857591835292505850771	1819	467	2025-09-27
	CcS-4858	XX8581768137002436094025	3964	817	2028-12-28
	CcS-4859	XX7826930491423553609370	4983	277	2026-11-26
	CcS-4860	XX5559590368835304645299	6876	661	2027-07-27
	CcS-4861	XX2035182877195191627307	5710	398	2026-04-25
	CcS-4862	XX4774721462463645409758	4042	174	2026-11-27

credit\_card 19 x

Output

Action Output

#	Time	Message	Action
✓ 27	13:02:12	5001 row(s) returned	SELECT * FROM credit_card



## Nivel 1

### Ejercicio 2

El departamento de Recursos Humanos ha identificado un error en el número de cuenta asociado a la tarjeta de crédito con ID CcU-2938. La información que debe mostrarse para este registro es: TR323456312213576817699999. Recuerda mostrar que el cambio se realizó.

1/ Actualizamos la tabla cambiando el iban al campo con el id marcado

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

```
SELECT * FROM credit_card  
Where id = "CcU-2938";
```

2/ Revisamos que la actualización se ha realizado correctamente

	id	iban	pin	cvv	expiring_date
►	CcU-2938	TR323456312213576817699999	3257	984	2022-10-30



## Nivel 1

## Ejercicio 2

```
23
24 • SELECT * FROM credit_card
25 Where id = "CcU-2938";
26
27 • UPDATE credit_card
28 SET iban = 'TR323456312213576817699999'
29 WHERE id = "CcU-2938";
30
31 • SELECT * FROM credit_card
32 Where id = "CcU-2938";
33
```

<

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

	id	iban	pin	cvv	expiring_date
▶	CcU-2938	TR323456312213576817699999	3257	984	2022-10-30
*	NULL	NULL	NULL	NULL	NULL

credit\_card 21 x

Output

Action Output

#	Time	Message	Action
✓ 29	13:04:43	1 row(s) returned	SELECT * FROM credit_card Where id = "CcU-2938"



## Nivel 1

### Ejercicio 3

En la tabla "transaction" ingresa un nuevo usuario con la siguiente información:

Id	108B1D1D-5B23-A76C-55EF-C568E49A99DD
credit_card_id	CcU-9999
company_id	b-9999
user_id	9999
lat	829.999
longitude	-117.999
amount	111.11
declined	0

1/ Para poder introducir este usuario debemos crear registros en el resto de tablas.

```
INSERT INTO credit_card (id)
VALUES ('CcU-9999');
```

```
INSERT INTO company (id)
VALUES ('b-9999');
```

2/ Insertamos el usuario con la información dada.  
Como no nos dan información de la hora, dejamos ese valor NULL

```
INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude,
VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', -117.999, 111.11, 0));
```

3/ Revisamos que se ha insertado correctamente

	id	credit_card_id	company_id	user_id	lat
▶	108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999





## Nivel 1

### Ejercicio 3

```
40 • INSERT INTO credit_card (id)
41   VALUES ('CcU-9999');
42
43 • INSERT INTO company (id)
44   VALUES ('b-9999');
45
46 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, timestamp, amount, declined)
47   VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', NULL, '111.11', '0');
48
49 • SELECT * FROM transaction
50   Where id = "108B1D1D-5B23-A76C-55EF-C568E49A99DD";
51
```

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

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
	108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	NULL	111.11	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 17 x Apply

Output

Action Output

#	Time	Message	Action	Duration /
23	12:49:34	1 row(s) affected	INSERT INTO credit_card (id) VALUES ('CcU-9999')	0.016 sec



## Nivel 1

### Ejercicio 4

Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit\_card. Recuerda mostrar el cambio realizado.

	id	iban	pan	pin
▶	CcS-4857	XX4857591835292505850771	2314242385113924	1819
	CcS-4858	XX8581768137002436094025	6582720299715533	3964
	CcS-4859	XX7826930491423553609370	8861684536289642	4983
	CcS-4860	XX5559590368835304645299	2481155515498459	6876

1/ Alteramos la tabla para eliminar la columna mencionada

```
alter table credit_card drop column pan;  
SELECT * FROM credit_card;
```

2/ Revisamos que la actualización se ha realizado correctamente








	id	iban	pin	cvv	expiring_date
▶	CcS-4857	XX4857591835292505850771	1819	467	2025-09-27
	CcS-4858	XX8581768137002436094025	3964	817	2028-12-28
	CcS-4859	XX7826930491423553609370	4983	277	2026-11-26



## Nivel 1

## Ejercicio 4

```
57 • alter table credit_card drop column pan;  
58 • SELECT * FROM credit_card;  
59  
60
```

<   Filter Rows:  | Edit:    | Export/Import:   | Wrap Cell Cont

	id	iban	pin	cvv	expiring_date
▶	CcS-4857	XX4857591835292505850771	1819	467	2025-09-27
	CcS-4858	XX8581768137002436094025	3964	817	2028-12-28
	CcS-4859	XX7826930491423553609370	4983	277	2026-11-26
	CcS-4860	XX5559590368835304645299	6876	661	2027-07-27
	CcS-4861	XX2035182877195191627307	5710	398	2026-04-25
	CcS-4862	XX4774721462463645409758	4042	174	2026-11-27

credit\_card 23 x

Output

📄 Action Output ▼

	#	Time	Message	Action
✓	31	13:08:31	5001 row(s) returned	SELECT * FROM credit_card



## Nivel 2

### Ejercicio 1

Elimina de la tabla transaction el registro con ID  
000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de datos.

1/ Alteramos la tabla para eliminar la columna mencionada

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

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

2/ Revisamos que se ha eliminado correctamente

	id	credit_card_id	company_id	user_id	lat	longitude
*	NULL	NULL	NULL	NULL	NULL	NULL

	#	Time	Message	Action
✓	1	10:06:13	0 row(s) returned	SELECT * FROM trar



## Nivel 2

### Ejercicio 1

```
65 • DELETE FROM transaction
66 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
67
68 • SELECT * FROM transaction
69 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
70
```

< Filter Rows:  | Edit: | Export/Import: | Wrap Cell Content:

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

transaction 27 x

Output



Action Output

#	Time	Message
✓ 1	10:06:13	0 row(s) returned

Action

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



## Nivel 2

### Ejercicio 2

La sección de marketing desea tener acceso a información específica para realizar análisis y estrategias efectivas. Se ha solicitado crear una vista que proporcione detalles clave sobre las compañías y sus transacciones. Será necesaria que crees una vista llamada VistaMarketing que contenga la siguiente información: Nombre de la compañía. Teléfono de contacto. País de residencia. Promedio de compra realizado por cada compañía. Presenta la vista creada, ordenando los datos de mayor a menor media de compra.

1/ Creamos la Vista con los requisitos pedidos

```
CREATE VIEW VistaMarketing AS
SELECT c.id, c.company_name, c.phone, c.country,
ROUND(AVG(t.amount),2) AS 'Media de ventas'
FROM company c
JOIN transaction t ON c.id = t.company_id
WHERE t.declined = 0
GROUP BY c.id
;
```

2/ Presentamos la vista creada, ordenando los datos de mayor a menor media de compra

```
SELECT * FROM VistaMarketing
ORDER BY 'Media de ventas' DESC;
```

3/ Revisamos que se ha creado correctamente

	id	company_name	phone	country
▶	b-2222	Ac Fermentum Incorporated	06 85 56 52 33	Germany
	b-2226	Magna A Neque Industries	04 14 44 64 62	Australia



## Nivel 2

## Ejercicio 2

```
77 • CREATE VIEW VistaMarketing AS
78 SELECT c.id, c.company_name, c.phone, c.country,
79 ROUND(AVG(t.amount),2) AS 'Media de ventas'
80 FROM company c
81 JOIN transaction t ON c.id = t.company_id
82 WHERE t.declined = 0
83 GROUP BY c.id
84 ;
85
86 -- Presenta la vista creada, ordenando los datos de mayor a menor media de compra.
87
88 • SELECT * FROM VistaMarketing
89 ORDER BY 'Media de ventas' DESC;
90
```

<

Result Grid | | Filter Rows:  | Export: | Wrap Cell Content:

	id	company_name	phone	country	Media de ventas
▶	b-2222	Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
	b-2226	Magna A Neque Industries	04 14 44 64 62	Australia	260.04
	b-2230	Fusce Corp.	08 14 97 58 85	United States	265.23
	b-2234	Convallis In Incorporated	06 66 57 29 50	Germany	257.75
	b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	New Zealand	260.34
	b-2242	Donec Ltd	01 25 51 37 37	Norway	265.01
	b-2246	Sed Nunc Ltd	02 62 64 73 48	United Kingdom	254.13

VistaMarketing 29 x

Output

Action Output

#	Time	Message	Action
✓ 37	13:20:23	101 row(s) returned	SELECT * FROM VistaMarketing ORDER BY 'Media



## Nivel 2

### Ejercicio 3

Filtra la vista VistaMarketing para mostrar sólo las compañías que tienen su país de residencia en "Germany"

1/ Alteramos la tabla para filtrar las compañías de "Germany"

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

2/ Revisamos que se ha eliminado correctamente

	id	company_name	phone	country	Media de ventas
▶	b-2222	Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
	b-2234	Convallis In Incorporated	06 66 57 29 50	Germany	257.75
	b-2302	Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.32
	b-2306	Augue Foundation	06 88 43 15 63	Germany	253.51
	b-2358	Ac Industries	09 34 65 40 60	Germany	255.15
	b-2550	Auctor Mauris Corp.	05 62 87 14 41	Germany	254.77
	b-2566	Aliquam PC	01 45 73 52 16	Germany	253.14
	b-2614	Putrum Non Inc	02 66 31 61 09	Germany	255.14





## Nivel 2

### Ejercicio 3

```
97
98 • SELECT * FROM VistaMarketing
99 WHERE country = "Germany";
100
```

< **Result Grid** Filter Rows:  | Export: | Wrap Cell Content:

	id	company_name	phone	country	Media de ventas
▶	b-2222	Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
	b-2234	Convallis In Incorporated	06 66 57 29 50	Germany	257.75
	b-2302	Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.32
	b-2306	Augue Foundation	06 88 43 15 63	Germany	253.51
	b-2358	Ac Industries	09 34 65 40 60	Germany	255.15
	b-2550	Auctor Mauris Corp.	05 62 87 14 41	Germany	254.77
	b-2566	Aliquam PC	01 45 73 52 16	Germany	253.14

VistaMarketing 31 x

Output

Action Output ▼

#	Time	Message	Action
✓ 39	13:22:51	8 row(s) returned	SELECT * FROM VistaMarketing WHERE country = "Germany"

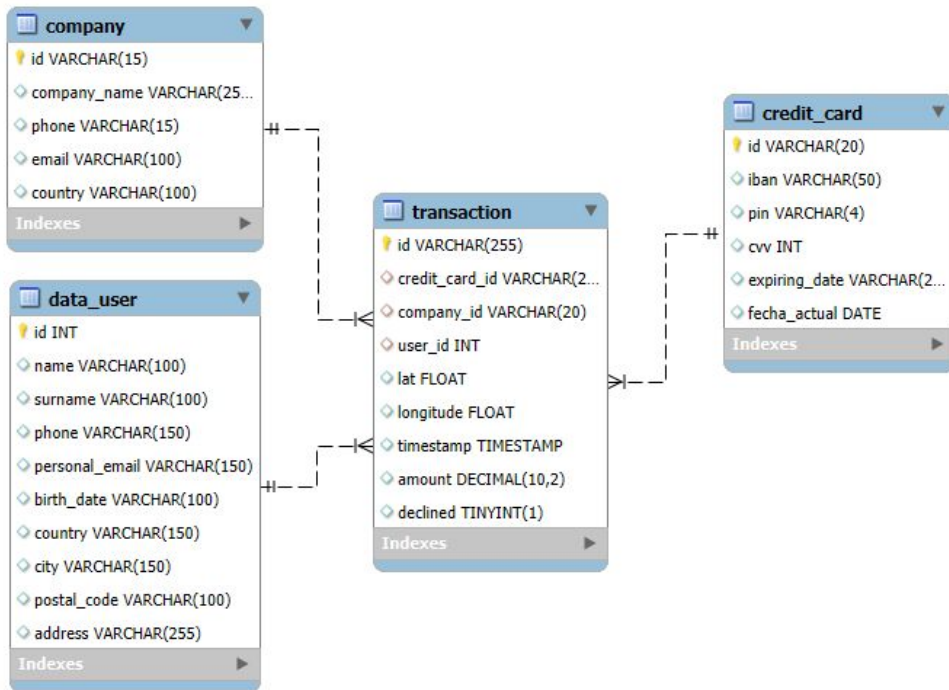


## Nivel 3

### Ejercicio 1

La semana próxima tendrás una nueva reunión con los gerentes de marketing. Un compañero de tu equipo realizó modificaciones en la base de datos, pero no recuerda cómo las realizó. Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama.

Es necesario que describas el "paso a paso" de las tareas realizadas. Es importante realizar descripciones sencillas, simples y fáciles de comprender.





## Nivel 3

### Ejercicio 1

La semana próxima tendrás una nueva reunión con los gerentes de marketing. Un compañero de tu equipo realizó modificaciones en la base de datos, pero no recuerda cómo las realizó. Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama.

Es necesario que describas el "paso a paso" de las tareas realizadas. Es importante realizar descripciones sencillas, simples y fáciles de comprender.

1/ Comenzamos creando la tabla con la que trabajar

```
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)  
);
```

2/ introducimos los datos del archivo "dades\_introduir\_user"

```
1 • INSERT INTO user (id, name, surname, phone, email, bi  
2 • INSERT INTO user (id, name, surname, phone, email, bi  
3 • INSERT INTO user (id, name, surname, phone, email, bi  
4 • INSERT INTO user (id, name, surname, phone, email, bi
```



## Nivel 3

### Ejercicio 1

3/ Revisamos que la tabla se ha creado bien nos encontramos con un problema

```
SHOW CREATE TABLE transaction;  
SHOW CREATE TABLE user;
```

	Table	Create Table
►	user	CREATE TABLE `user` ( `id` int NOT NULL, `name` varchar(100) DE

4/ Como id.user es char, y user\_id.transaction es Int, hay que igualarlos

```
ALTER TABLE user  
MODIFY COLUMN id INT;
```

5/ Creamos las conexiones uniendo los user id de ambas tablas

```
ALTER TABLE transaction  
ADD FOREIGN key (user_id) references user(id);
```



## Nivel 3

### Ejercicio 1

6/ cambiamos algunos nombres para ajustarnos a la tabla de la imagen

```
ALTER TABLE user RENAME TO data_user;
```

```
ALTER TABLE data_user  
RENAME COLUMN email TO personal_email;
```

```
ALTER TABLE company  
DROP COLUMN website;
```

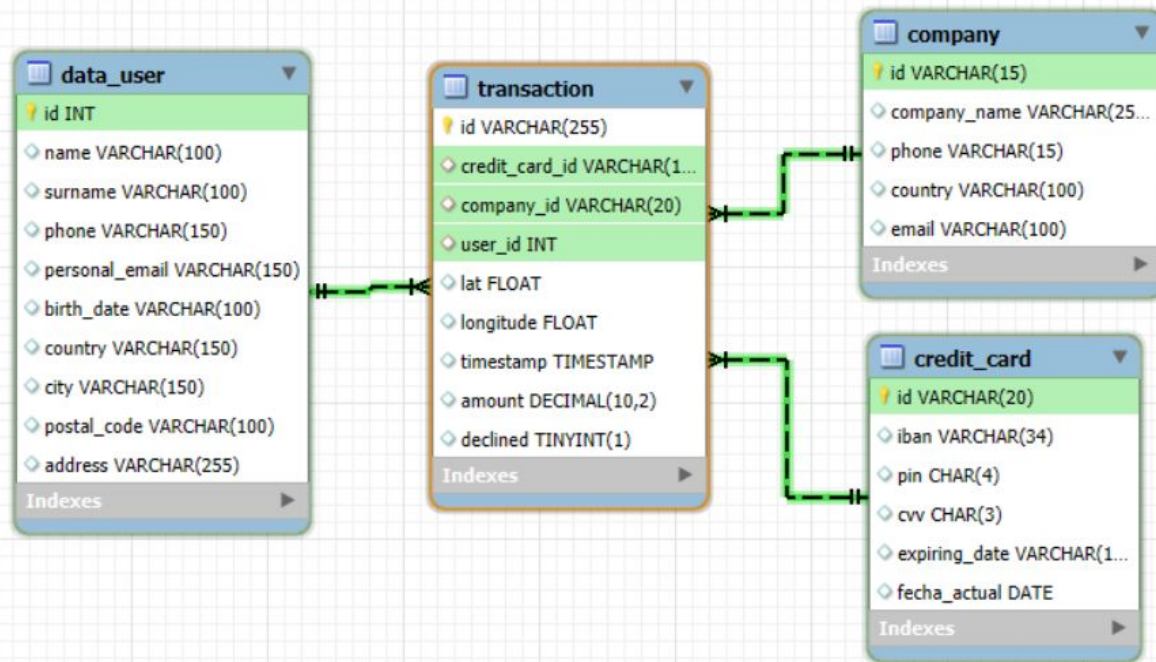
```
ALTER TABLE credit_card  
ADD COLUMN fecha_actual DATE DEFAULT (CURDATE());
```



## Nivel 3

### Ejercicio 1

6/ Revisamos que las conexiones son correctas





## Nivel 3

## Ejercicio 2

La empresa también os pide crear una vista llamada "InformeTecnico" que contenga la siguiente información:

- ID de la transacción
- Nombre del usuario/a
- Apellido del usuario/a
- IBAN de la tarjeta de crédito usada.
- Nombre de la compañía de la transacción realizada.
- Asegúrese de incluir información relevante de las tablas que conoceréis y utilice alias para cambiar de nombre columnas según sea necesario.

Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.

1/ Creamos la Vista con los requisitos pedidos y añadimos algunos campos como la cantidad de la transacción o el lugar donde se hizo

```
CREATE VIEW InformeTecnico AS
SELECT u.name AS Nombre, u.surname AS Apellido, cr.iban,
t.id AS "id transacción", t.amount AS Cantidad,
c.company_name AS 'Beneficiario transacción',
u.country AS 'Lugar transacción'
FROM transaction t
JOIN credit_card cr ON cr.id = t.credit_card_id
JOIN data_user u ON u.id = t.user_id
JOIN company c ON c.id = t.company_id
WHERE t.declined = 0
GROUP BY t.id;

SELECT * FROM InformeTecnico
ORDER BY "id transacción" DESC;
```

2/ Presentamos la vista creada, ordenando los datos de mayor a menor media de compra

	Nombre	Apellido	iban	id transacción	Cantidad
▶	Gxnmjn	Fjycossj	XX691799023254713153661899	00043A49-2949-494B-A5DD-A5BAE38B19DD	395.43
	Wolake	Ukynumly	XX715219809262791935275891	00045D6B-ED2E-4F2F-8186-CEE074D875D0	326.01
	Umhwoi	Ubm dxvmz	XX785022829713140880782522	000481C3-1C26-4FEF-83A0-4CD0EB004BBD	161.60
	Qpfyfa	Mtnpfdfq	XX634626564536875934323485	00051AA4-9CBE-4268-B070-C38062A1B3E2	148.91
	Keegan	Watson	RO76DAFO6583348580208155	0008A312-EDFE-4A4F-8C99-E9C92EC3CA4D	294.59
	Rregrc	Jfdvcgne	XX986535778187638923658336	0009A151-9BCF-4E31-9053-A468FF77FAAB	383.63
	Novmmv	Boznhdmm	XX738074164311515717218547	0009D494-6245-4DF9-955D-2C084191CFFB	197.80
	Hxzrri	Mzchxymz	XX714741860105067529287700	000A1DEC-CDB6-4AB2-A619-71DAB8D4A...	339.94
	Ghsngl	Gkwhehrz	XX343833941783938653185254	000A1E64-1414-40B0-9D92-5678A4D958E2	369.71



## Nivel 3

## Ejercicio 2

```
155 • CREATE VIEW InformeTecnico AS
156 SELECT u.name AS Nombre, u.surname AS Apellido, cr.iban,
157 t.id AS "id transacción", t.amount AS Cantidad,
158 c.company_name AS 'Beneficiario transacción',
159 u.country AS 'Lugar transacción'
160 FROM transaction t
161 JOIN credit_card cr ON cr.id = t.credit_card_id
162 JOIN data_user u ON u.id = t.user_id
163 JOIN company c ON c.id = t.company_id
164 WHERE t.declined = 0
165 GROUP BY t.id;
166
167 • SELECT * FROM InformeTecnico
168 ORDER BY "id transacción" DESC;
```

Result Grid							
Filter Rows: <input type="text"/>							
Export: <input type="button" value="Export"/> Wrap Cell Content: <input type="checkbox"/> Fetch rows: <input type="button" value="Fetch"/>							
	Nombre	Apellido	iban	id transacción	Cantidad	Beneficiario transacción	Lugar transacción
▶	Gxnmjn	Fjycossj	XX691799023254713153661899	00043A49-2949-4946-A5DD-A58AE38B19DD	395.43	Eget Tincidunt Dui Institute	France
	Wolake	Ukynumly	XX715219809262791935275891	00045D68-ED2E-4F2F-8186-CEE074D875D0	326.01	Neque Tellus Imperdiet Corp.	United States
	Umhwioi	Ubmndxvmz	XX785022829713140880782522	000481C3-1C26-4FEF-83A0-4CD0EB004EBD	161.60	Fusce Corp.	Canada
	Qpfyfa	Mbnpfdfq	XX634626564536875934323485	00051AA4-9CBE-4268-B070-C38062A1B3E2	148.91	Mus Aenean Eget Foundation	Netherlands
	Keegan	Watson	RO76DAFO6583348580208155	0008A312-EDFE-4A4F-8C99-E9C92EC3CA4D	294.59	Aliquam Iaculis Lacus Corp.	Canada
	Rreqrc	Jfdvcgne	XX986535778187638923658336	0009A151-9BCF-4E31-9053-A468FF77FAAB	383.63	Lorem Ipsum Dolor Corp.	Netherlands
	Novmmv	Boznhdn	XX738074164311515717218547	0009D494-6245-4DF9-955D-2C084191CFFB	197.80	Amet Luctus Vulputate Fou...	Canada

InformeTecnico 33 x

Output

Action Output

#	Time	Message
79	16:42:08	99762 row(s) returned

Action
SELECT * FROM InformeTecnico ORDER BY "id transacción" DESC





# Gracias.

