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.

```
CREATE TABLE credit_card (
    id VARCHAR(10),
    iban VARCHAR(50),
    pan VARCHAR(20),
    pin INT,
    cvv INT,
    expiring_date VARCHAR(15)
);

#Me dejo el drop en caso que necesite modificar algo de la tabla y prefiera comenzar de cero
#DROP TABLE credit_card;

SELECT *
FROM credit_card
```

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR30195031221357681763861	5424465566813633	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	5142423821948828	9080	887	08/24/23
CcU-2952	BG45IVQL52710525608255	4556 453 55 5287	4598	438	06/29/21
CcU-2959	CR7242477244335841535	372461377349375	3583	667	02/24/23
CcU-2966	BG72LKTQ156276283777363	448566 886747 7265	4900	130	10/29/24
CcU-2973	PT87806228135092429456346	544 58654 54343 384	8760	887	01/30/25
CcU-2980	DE39241881883086277136	402400 7145845969	5075	596	07/24/22
CcU-2987	GE89681434837748781813	3763 747687 76666	2298	797	10/31/23
CcU-2994	BH62714428368066765294	344283273252593	7545	595	02/28/22
CcU-3001	CY49087426654774581266832110	511722 924833 2244	9562	867	09/16/22
CcU-3008	LU5072166936161192330	4485744464433884	1856	740	04/05/25
CcU-3015	PS11939821629571596834245821	3784 662233 17389	3246	822	01/31/22
CcU-3022	GT91695162850556977423121857	5164 1379 4842 3951	5610	342	04/25/25
CcU-3029	AZ62317413982441418123739746	3429 279566 77631	9708	505	09/02/23
CcU-3036	AZ39336002925842865843941994	3768 451556 48766	2232	565	10/27/25
CcU-3043	TN6488143310514852179535	455676 6437463635	5969	196	06/07/25
CcU-3050	FR5167744369175836831854477	4024007123722	4834	126	10/09/23
CcU-3057	LU931822574697545215	3484 621767 21237	6805	848	09/14/25
CcU-3064	PS14696554544925337762727313	3467 732741 26810	3865	498	06/03/25
CcU-3071	NO8923814763512	3464 789562 23352	6625	661	12/20/23
CcU-3078	IS025127145884623279548733	4539 322 74 2377	9405	720	03/08/23

Hago una comprobación de que toda la información se ha cargado en la tabla credit_card
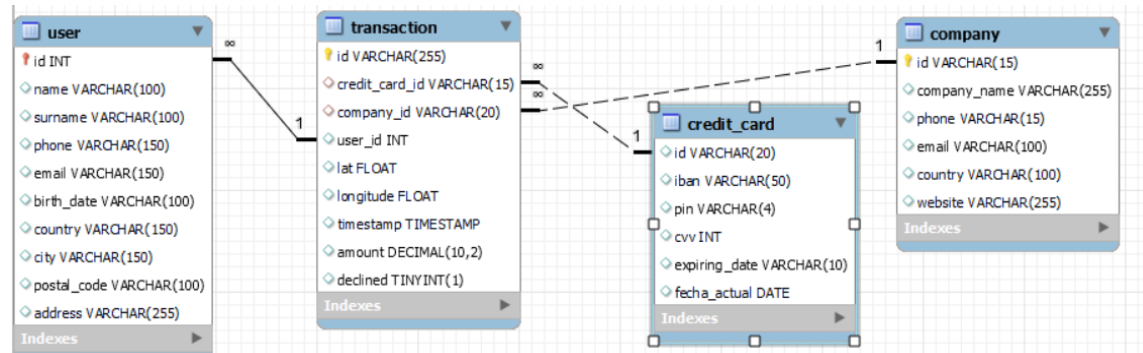
En el diagrama vemos que company se relaciona con transaction a través de company.id con transaction.company_id, y que credit card se relaciona con transaction a través del credit_card.id con transaction.credit_card_id.

- Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb el: IBAN CcU-2938. Es requereix actualitzar la informació que identifica un compte bancari a nivell internacional (identificat com "IBAN"): TR323456312213576817699999. Recorda mostrar que el canvi es va realitzar.

Primero localizamos la informacion de la ID: Ccu-2938

```
SELECT *
FROM credit_card
WHERE id = "CcU-2938"
```

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR30195031221357681763 8661	5424465566813633	3257	984	10/30/22

Dejo apuntada la información por si hubiera un error no perder los datos anteriores.

Pasamos a actualizar la información:

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

Realizamos de nuevo el primer select para comprar que se ha actualizado la información correctamente. Y vemos que está bien.

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

- Exercici 3

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

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

Procedemos a actualizar la tabla transaction, probando actualizarla he visto que por temas de foreing keys me estaba dando el problema que necesitamos una compañía con el ID "b-9999" para poder añadir este nuevo usuario

```sql
INSERT INTO company (id)
VALUES ('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);
```

Una vez añadido el nuevo id en company.id y añadido toda la información del nuevo usuario, procedemos a asegurarnos que la información se ha añadido correctamente a la tabla

```sql
SELECT *
FROM transaction
WHERE company_id = "b-9999";
```

transaction.

Lanzamos un query con el id de la compañía y obtenemos el resultado correcto:

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

- Exercici 4

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

Para borrar la columna pan necesitaremos hacer un alter de la tabla credit_card y hacer un drop en la columna pan.

```sql
ALTER TABLE credit_card
DROP COLUMN pan;
```

Revisamos con una simple query que este todo correcto:

```
SELECT *
FROM credit_card;
```

Confirmamos que la columna pan ha sido eliminada.

	id	iban	pin	cvv	expiring_date
▶	CcU-2938	TR32345631221357681769999	3257	984	10/30/22
	CcU-2945	DO26854763748537475216568689	9080	887	08/24/23
	CcU-2952	BG45IVQL52710525608255	4598	438	06/29/21
	CcU-2959	CR72424772443335841535	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
	CcU-3001	CY49087426654774581266832110	9562	867	09/16/22
	CcU-3008	LU50721669361611923	1856	740	04/05/25
	CcU-3015	PS11939821629571596834245682	3246	822	01/31/22
	CcU-3022	GT91695162850556977423121857	5610	342	04/25/25
	CcU-3029	AZ62317413982441418123739746	9708	505	09/02/23
	CcU-3036	AZ39336002925842865843941994	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
	CcU-3064	PS14696554549253377627273133	3865	498	06/03/25
	CcU-3071	NO8923814763512	6625	661	12/20/23
	CcU-3078	IS025127145884623279548733	9405	720	03/08/23

Nivell 2

- Exercici 1

Elimina el registre amb IBAN 02C6201E-D90A-1859-B4EE-*88D2986D3B02 de la base de dades.

Primero buscamos el id en concreto dentro de la tabla de transaction

```
SELECT *
FROM transaction
WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02"
```

	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.92	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Una vez ya tenemos ubicado correctamente el registro, procedemos a borrarlo.

```sql
DELETE FROM transaction
WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

Haremos un delete desde la tabla transaction hacia la ID = "02C6201E-D90A-1859-B4EE-*88D2986D3B02 "

Y comprobamos el output:

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

Y vemos que todo el registro ha sido eliminado.

- 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

Aquí utilizamos el command Create View. Dentro de la vista haremos una query que juntará ambas tablas, agruparemos para poder generar el agregado del Average en el transaction.amount y lo ordenamos por descendiente para verlo de mayor a menor

```sql
CREATE VIEW VistaMarketing AS
SELECT company.company_name, company.phone, company.country, AVG(transaction.amount)
FROM company
JOIN transaction ON transaction.company_id = company.id
GROUP BY company.company_name, company.phone, company.country
ORDER BY AVG(transaction.amount) DESC;
```
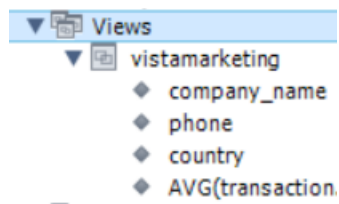
Una vez creada la vista comprobamos que se ha creado correctamente y hacemos una query de la propia vista

Dejo screen del output (no aparecen todas las empresas debido a que hay 101 rows de data):

company_name	phone	country	AVG(transaction.amount)
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 Etiam 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

- Exercici 3

Filtra la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"

Aqui simplemente haremos una query sobre la vista filtrando el company.country a Germany

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

Resultado final:

	company_name	phone	country	AVG(transaction.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

Comprobamos que obtenemos 8 compañias de procedencia alemana.

Nivell 3

- Exercici 1

La setmana vinent tindràs 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 les següents modificacions (s'espera que realitzin 6 canvis):
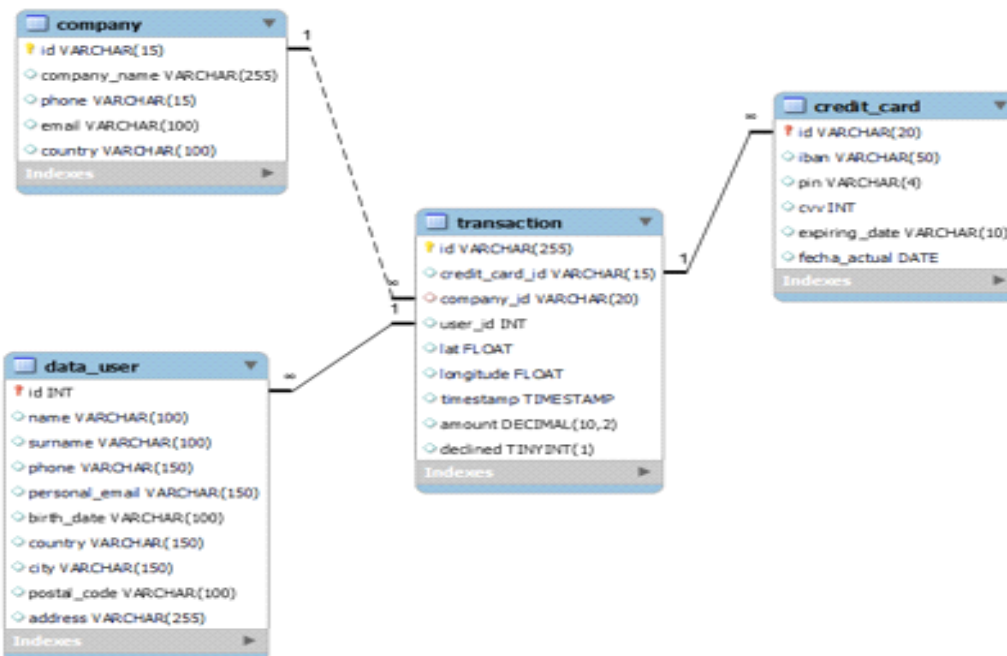
Procedemos a modificar la tabla company:

```sql
ALTER TABLE company
MODIFY id VARCHAR(15),
MODIFY company_name VARCHAR(255),
MODIFY phone VARCHAR(15),
MODIFY email VARCHAR(100),
MODIFY country VARCHAR(100);
```

Modificamos la tabla transaction:

```sql
ALTER TABLE transaction
MODIFY id VARCHAR(255),
MODIFY credit_card_id VARCHAR(15),
MODIFY company_id VARCHAR(20),
MODIFY user_id INT,
MODIFY lat FLOAT,
MODIFY longitude FLOAT,
MODIFY timestamp TIMESTAMP,
MODIFY amount DECIMAL(10,2),
MODIFY declined TINYINT(1) ;
```

Modificamos la tabla credit_card y añadimos la columna que faltaba:

```sql
ALTER TABLE credit_card
MODIFY id VARCHAR(20),
MODIFY iban VARCHAR(50),
MODIFY pin VARCHAR(4),
MODIFY cvv INT,
MODIFY expiring_date VARCHAR(10);
ALTER TABLE credit_card
ADD COLUMN fecha_actual DATE;
```

Creamos la tabla User

```
-- Creamos la tabla user

CREATE INDEX idx_user_id ON transaction(user_id);

CREATE TABLE IF NOT EXISTS user (
        id INT 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),
        FOREIGN KEY(id) REFERENCES transaction(user_id)
    );
```

Y procedemos a rellenar la tabla con los datos:

```
SET foreign_key_checks = 0;

-- Insertamos datos de user
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "1", "Zeus", "Gamble", "1-
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "2", "Garrett", "Mcconnell"
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "3", "Ciaran", "Harrison",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "4", "Howard", "Stafford",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "5", "Hayfa", "Pierce", "1
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "6", "Joel", "Tyson", "(71
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "7", "Rafael", "Jimenez",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "8", "Nissim", "Franks", "
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "9", "Mannix", "Mcclain",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "10", "Robert", "Mccarthy"
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "11", "Joan", "Baird", "(9
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "12", "Benedict", "Wheeler
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "13", "Allegra", "Stanton"
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "14", "Sara", "Flynn", "1-
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "15", "Noelani", "Patrick"
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "16", "Eric", "Roth", "1-2
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "17", "Bruce", "Gill", "(7
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "18", "Russell", "Jimenez"
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "19", "Nicholas", "Travis"
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "20", "Kelsey", "Bates", "
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "21", "Hall", "Reeves", "(
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "22", "Allistair", "Holmes
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (    "23", "Kelsie", "Bass", "1
```

Hacemos una query simple para revisar que se han ingresado correctamente los datos:

275 rows returned.

id	name	surname	phone	email	birth_date	country	city	postal_code	address
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell	73544	348-7818 Sagittis St.
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59464	903 Sit Ave
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbus	56518	736-2063 Tellus St.
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.edu	Feb 18, 1989	United States	Kailua	77417	Ap #545-2244 Erat. Rd.
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hotmail.org	Sep 26, 1998	United States	Sandy	31564	341-2821 Ultrices Av.
6	Joel	Tyson	(718) 288-8020	gravida.nunc.sed@yahoo.ca	Oct 15, 1989	United States	Nashville	96838	888-2799 Amet Street
7	Rafael	Jimenez	(817) 689-0478	eget@outlook.ca	Dec 4, 1981	United States	Hillsboro	29874	8627 Malesuada Rd.
8	Nissim	Franks	(692) 157-3469	egestas.aliquam.fringilla@google.ca	Aug 1, 1993	United States	Jackson	61750	Ap #251-7144 Integer St.
9	Mannix	Mcclain	(590) 883-2184	aliquam.nisl@outlook.com	Jan 24, 1987	United States	Richmond	35987	647-3080 Lacus St.

- 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.

Procedemos a crear las VISTA:

```sql
CREATE VIEW InformeTecnico AS
SELECT transaction.id AS id , user.name AS Nombre_usuario, user.surname AS Apellido_Usuario, credit_card.id AS IBAN, company.compar
FROM transaction
JOIN user ON user.id = transaction.user_id
JOIN company ON company.id = transaction.company_id
JOIN credit_card ON credit_card.id = transaction.credit_card_id;
```

```sql
SELECT *
FROM InformeTecnico
ORDER BY id DESC;
```

Revisamos con una simple query que se ejecute la nueva vista creada:

id	Nombre_usuario	Apellido_Usuario	IBAN	Nombre_compañia
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	CcU-2945	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	CcU-4849	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	CcU-4331	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	CcU-3960	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	CcU-3232	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	CcU-4219	Nunc Interdum Incorporated
FBD7E0D6-BA6B-F5BC-0CA9-EA4B8760100C	Hedwig	Gilbert	CcU-3281	Mauris Id Inc.
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	CcU-4576	Arcu LLP
FAAD3FEC-1A17-F141-43D3-359A5BA7CB3B	Hedwig	Gilbert	CcU-3183	Lorem Eu Incorporated