

NIVELL 1

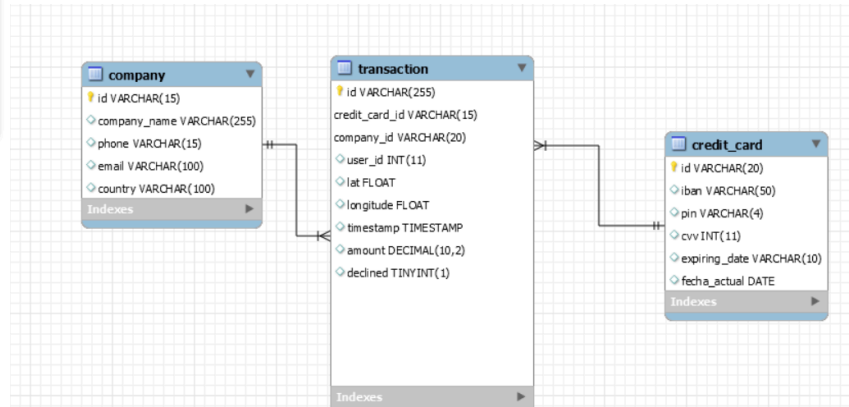
EXERCICI 1

Relacion Cardinalidad

La tabla 'transaction', es la tabla de hechos.

La tabla 'company', está relacionada 1 a N con 'transaction' a través de 'id' con 'company_id'.

La tabla 'credit_card', está relacionada 1 a N con 'transaction' a través de 'id' con 'company_id'.



La nueva tabla 'credit_card' contiene:

- 'id' que es la Primary Key, ya que contiene valores únicos.
- 'iban' que es el número bancario.
- 'pan' que es el número de la tarjeta.
- 'pin' es la clave del usuario para acceder a la tarjeta.
- 'ccv' es el código de valor de verificación de la tarjeta.
- 'expiring_date' que es la fecha de expiración de la tarjeta

La tabla 'credit_card', se relaciona con la tabla 'transactions' a través del campo 'id' y 'credit_card_id'. A su vez, como la tabla 'company' también está relacionada con 'transactions', las tres tablas estarán relacionadas como vemos en el diagrama.

Para la creación de la tabla, hemos tenido que escribir el siguiente código:

```
USE transactions;

-- Creamos la tabla credit_card
CREATE INDEX idx_credit_card_id ON transaction(credit_card_id);

CREATE TABLE IF NOT EXISTS credit_card (
    id VARCHAR(20) PRIMARY KEY,
    iban VARCHAR(50),
    pan VARCHAR(20),
    pin VARCHAR(4),
    cvv VARCHAR(10),
    expiring_date DATE,
    actual_date DATE
);

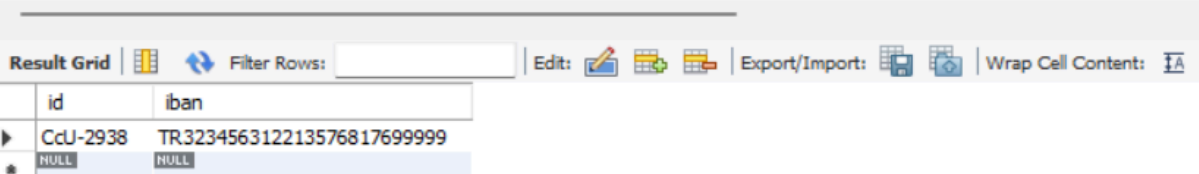
alter table transaction
add FOREIGN KEY FK_credit_id (credit_card_id)
references credit_card(id);
```

Posteriormente, hemos cargado todos los datos en cada campo para realizar las consultas.

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.

```
2  -- Actualizamos IBAN del id Ccu-2938
3
4  • UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE id = 'CcU-2938';
5
6  • SELECT id,iban
7    FROM credit_card
8   WHERE id = 'CcU-2938';
```



The screenshot shows a database interface with a toolbar and a result grid. The toolbar includes options for 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The result grid displays the following data:

id	iban
CcU-2938	TR323456312213576817699999
NULL	NULL

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

```
-- Añadimos la información del user_id = '9999'
USE credit_card;
INSERT INTO credit_card (id) VALUES ('CcU-9999');
```

Result Grid									
Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:			
id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	
10881D1D-5B23-A76C-55EF-C568E49A9DD	CcU-9999	b-9999	9999	829.999	-117.999	2024-03-05 12:30:00	111.11	0	

EXERCICI 4

```
30 -- Eliminamos la columna 'pan'
31 ALTER TABLE credit_card DROP COLUMN pan;
32
```

Result Grid |   Filter Rows: | Edit:    | Export

id	iban	pin	cvv	expiring_date
CcU-2938	TR323456312213576817699999	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	9080	887	08/24/23

EXERCICI 1

[illegible]

Borramos de la tabla 'transaction' todo el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02. Comprobamos que se ha borrado mostrando la segunda query y vemos que como resultado da NULL en todos los campos.

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.

```

53 • CREATE VIEW VistaMarketing AS
54
55 SELECT DISTINCT company.company_name, company.phone, company.country,AVG(transaction.amount) as avg_trans
56 FROM company
57 JOIN transaction
58 ON company.id = transaction.company_id
59 GROUP BY company.company_name, company.phone, company.country
60 ORDER BY avg_trans DESC;
61
62 -- Comprobamos la vista para ver si da resultado
63 • SELECT *
64 FROM VistaMarketing;
65

```

company_name	phone	country	avg_trans
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

EXERCICI 3

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

```

66 #EXERCICI 3
67 #Filtra la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"
68
69 • SELECT company_name
70 FROM VistaMarketing
71 WHERE country = 'Germany';
72
73

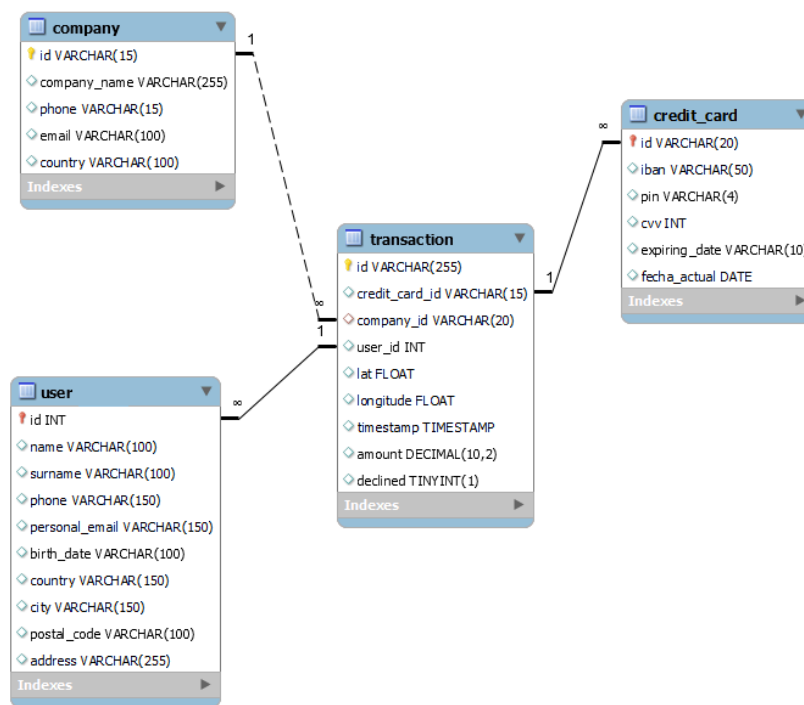
```

company_name
Aliquam PC
Ac Industries
Rutrum Non Inc.
Nunc Interdum Incorporated
Augue Foundation
Ac Fermentum Incorporated
Auctor Mauris Corp.
Convallis In Incorporated

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):



Paso 1:

Creemos la tabla user y una vez creada le añadimos toda la información a la tabla:

```
USE transactions;

-- 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)
);
```

```

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",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "2",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "3",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "4",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "5",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "6",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "7",
INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "8",

.

.

.

• INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "272",
• INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "273",
• INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "274",
• INSERT INTO user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES ( "275",

• SET foreign_key_checks = 1;

```

Paso 2:

Eliminamos la columna website de la tabla company:

```

101
102 -- Eliminamos la columna website de la tabla company
103
104 • ALTER TABLE company DROP COLUMN website;
105

```

id	company_name	phone	email	country
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.couk	Norway
b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom
b-2250	Amet Nulla Donec Corporation	07 15 25 14 74	mattis.integer.eu@protonmail.net	Italy
b-2254	Nascetur Ridiculus Mus Inc.	06 26 87 61 84	suspendisse.dui@icloud.net	United States

Paso 3:

Creamos la columna fecha_actual en la tabla credit_card:

```

107 -- Añadimos la columna fecha_actual a la tabla credit_card
108
109 • ALTER TABLE credit_card
110 ADD COLUMN fecha_actual DATE;
111

```

id	iban	pin	cvv	expiring_date	fecha_actual
----	------	-----	-----	---------------	--------------

Paso 4:

Modificamos el nombre del campo email por personal_email de la tabla user:

```
113 -- Modificamos el nombre del campo email por personal_email de la tabla user
114
115 • ALTER TABLE user
116   CHANGE email personal_email varchar(150);
117
```

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

id	name	surname	phone	personal_email	birth_date	country	city	postal_code	address
----	------	---------	-------	----------------	------------	---------	------	-------------	---------

Paso 5:

Así quedarían todas las tablas que tenemos actualmente:

Tabla credit_card:

```
118 • DESCRIBE credit_card;
```

Result Grid | Filter Rows: | Export: | Wrap

	Field	Type	Null	Key	Default	Extra
▶	id	varchar(20)	NO	PRI	NULL	
	iban	varchar(50)	YES		NULL	
	pin	varchar(4)	YES		NULL	
	cvv	int(11)	YES		NULL	
	expiring_date	varchar(10)	YES		NULL	
	fecha_actual	date	YES		NULL	

Tabla company:

```
118 • DESCRIBE company;
```

Result Grid | Filter Rows: | Export: | Wrap

	Field	Type	Null	Key	Default	Extra
▶	id	varchar(15)	NO	PRI	NULL	
	company_name	varchar(255)	YES		NULL	
	phone	varchar(15)	YES		NULL	
	email	varchar(100)	YES		NULL	
	country	varchar(100)	YES		NULL	

Tabla transaction:

```
118 • DESCRIBE transaction;
```

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI	NULL	
credit_card_id	varchar(15)	YES		NULL	
company_id	varchar(20)	YES	MUL	NULL	
user_id	int(11)	YES	MUL	NULL	
lat	float	YES		NULL	
longitude	float	YES		NULL	
timestamp	timestamp	YES		NULL	
amount	decimal(10,2)	YES		NULL	
declined	tinyint(1)	YES		NULL	

Tabla user:

```
118 • DESCRIBE user;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
surname	varchar(100)	YES		NULL	
phone	varchar(150)	YES		NULL	
personal_email	varchar(150)	YES		NULL	
birth_date	varchar(100)	YES		NULL	
country	varchar(150)	YES		NULL	
city	varchar(150)	YES		NULL	
postal_code	varchar(100)	YES		NULL	
address	varchar(255)	YES		NULL	

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.

```
126 • CREATE VIEW InformeTecnico AS
127
128 SELECT t.id as transaccion_id, u.name as nombre, u.surname as apellido, cc.iban,
129 c.company_name as compañía, t.timestamp as fecha_transaccion, t.declined
130 FROM transaction t
131 JOIN company c
132 ON c.id = t.company_id
133 JOIN user u
134 ON u.id = t.user_id
135 JOIN credit_card cc
136 ON cc.id = t.credit_card_id;
137
138 • SELECT *
139 FROM InformeTecnico
140 ORDER BY transaccion_id DESC;
141
142 -- Si quisieramos hacer la selección de todos los campos a través del WHERE y AND en vez de usar JOIN
143 /*
144 SELECT t.id as transaccion_id, u.name as nombre, u.surname as apellido, cc.iban,
145 c.company_name as compañía, t.timestamp as fecha_transaccion, t.declined
146 FROM transaction t, company c, user u, credit_card cc
147 WHERE c.id = t.company_id
148 AND u.id = t.user_id
149 AND cc.id = t.credit_card_id;
150 */
```

Result Grid

transaccion_id	nombre	apellido	iban	compañía	fecha_transaccion	declr
FE96CE47-8D59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries	2021-06-15 00:26:29	1
FE809ED4-2DB6-55AC-C915-929516E46468	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated	2021-11-09 21:35:40	0
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated	2021-06-13 11:41:17	0
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC	2022-03-16 02:35:05	0
FD2E8957-4148-8EEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.	2022-03-13 00:27:34	0
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated	2022-02-06 22:48:41	0

Se ha creado la vista InformeTecnico y se ha modificado los campos de la siguiente manera:

- ID de la transacció → transaccion_id
- Nom de l'usuari/ària → nombre
- Cognom de l'usuari/ària → apellido
- IBAN de la targeta de crèdit usada. → iban
- Nom de la companyia de la transacció realitzada. → compañía

También hemos visto oportuno agregar los campos de 'fecha_transaccion' y 'declined' para poder ver reflejado que día se realizó la transacción (gracias al campo 'fecha_transaccion') y si esa transacción ha sido aprobada o no (gracias al campo 'declined').