

# 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 ingessis la informació del document denominat "dades\_introduir\_credit". Recorda mostrar el diagrama i realitzar una breu descripció d'aquest.

Comentario: Primero creamos la tabla

```
10 • USE transactions;
11
12 • CREATE TABLE credit_card(
13     id VARCHAR(45) NOT NULL,
14     iban VARCHAR(45),
15     pan VARCHAR(45),
16     pin CHAR(4),
17     cvv CHAR(3),
18     expiring_date DATE,
19     PRIMARY KEY (id)
20 );
21
22 • SHOW COLUMNS FROM credit_card;
```

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

Field	Type	Null	Key	Default	Extra
id	varchar(45)	NO	PRI	NULL	
iban	varchar(45)	YES		NULL	
pan	varchar(45)	YES		NULL	
pin	char(4)	YES		NULL	
cvv	char(3)	YES		NULL	
expiring_date	date	YES		NULL	

Luego insertamos los valores. Es importante comentar que hemos modificado las fechas antes de importarlas para que su formato sea aaaa-mm-dd, ya que este es el formato de fecha que MySQL requiere a la hora de usar el tipo de datos "Date".

```

285 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
286 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
287 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
288 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
289 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
290 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
291 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
292 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
293 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
294 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
295 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
296 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
297 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
298 • INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES (
299
300 • SELECT * FROM credit_card;
301

```

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	5424465566813633	3257 984	2022-10-30	
CcU-2945	DO26854763748537475216568689	5142423821948828	9080 887	2023-08-24	
CcU-2952	BG45IVQL52710525608255	4556 453 55 5287	4598 438	2021-06-29	
CcU-2959	CR7242477244335841535	372461377349375	3583 667	2023-02-24	
CcU-2966	BG72LKTQ15627628377363	448566 886747 7265	4900 130	2024-10-29	

Por último, convertimos la columna "credit\_card\_id" de la tabla **transaction** en Foreign Key, para que pueda relacionarse con la columna "id" de la tabla **credit\_card** que acabamos de crear:

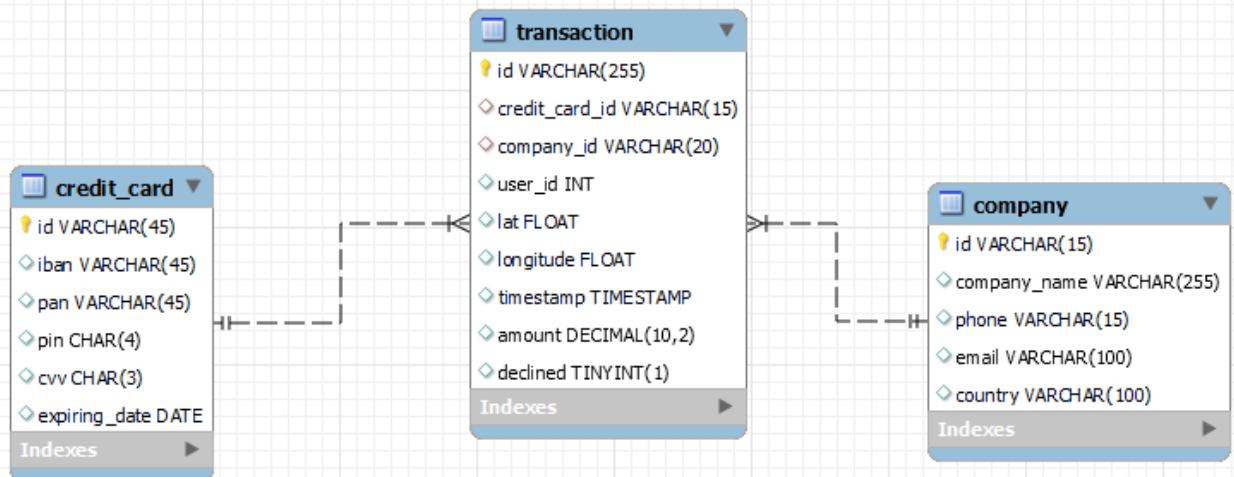
```

302 • ALTER TABLE transaction
303 ADD CONSTRAINT
304 FOREIGN KEY (credit_card_id)
305 REFERENCES credit_card(id);
306
307 • SHOW COLUMNS FROM transaction;
308

```

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

De esta manera, obtenemos el siguiente esquema:



**transaction** es una tabla de hechos, donde se registran las transacciones realizadas por las diferentes compañías. Esta tabla tiene una primary key ("id"), correspondiente al ID de cada transacción. Además, esta tabla tiene ahora dos foreign keys: "company\_id" y "credit\_card\_id", a través de las cuales se establece la relación con las tablas company y credit\_card, respectivamente.

El resto de la información dentro de la tabla **transaction** es relativa al credit\_card\_id, user\_id, latitud y longitud, fecha, hora, importe y si la compra se efectivizó o no.

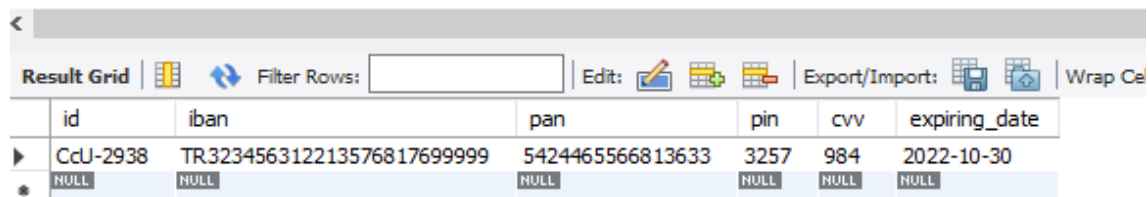
Las tablas **company** y **credit\_card** son tablas de dimensión donde obtenemos más información sobre las empresas y las tarjetas de crédito utilizadas para las transacciones.

La cardinalidad de las relaciones entre estas dos tablas y **transaction** es de 1 a N: una misma compañía puede realizar más de una transacción, y una transacción solo puede ser realizada por una única compañía; asimismo, una misma tarjeta de crédito puede utilizarse para más de una transacción, pero una transacción solo se puede realizar con una tarjeta de crédito.

## Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb el ID 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.

```
315 • UPDATE credit_card
316 SET iban = "TR323456312213576817699999"
317 WHERE ID = "CcU-2938";
318
319 • SELECT *
320 FROM credit_card
321 WHERE ID = "CcU-2938";
322
```



	id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	TR323456312213576817699999	5424465566813633	3257	984	2022-10-30
*	NULL	NULL	NULL	NULL	NULL	NULL

Comentario: Aquí arriba se muestra que el IBAN de la tarjeta de crédito con id "CcU-2938" fue exitosamente cambiado a TR323456312213576817699999.

## - 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

Comentario: Para poder agregar esta transacción, primero es necesario añadir una compañía y una credit card con los ids mencionados, ya que si no lo hacemos nos saldrá un error en SQL.

```
326 • INSERT INTO company
327   VALUES ("b-9999",
328           "Fake Company Inc",
329           "08 68 42 53 53",
330           "fake.company.inc@gmail.com",
331           "Argentina",
332           "https://linkedin.com/fakecompanyinc"
333          );
334
335 • SELECT *
336   FROM company
337  WHERE id = "b-9999";
338
```

Result Grid						
Filter Rows: <input type="text"/>						
Edit:      Export/Import:     Wrap Cell Content:						
	id	company_name	phone	email	country	website
▶	b-9999	Fake Company Inc	08 68 42 53 53	fake.company.inc@gmail.com	Argentina	https://linkedin.com/fakecompanyinc
*	NULL	NULL	NULL	NULL	NULL	NULL

```
339 • INSERT INTO credit_card
340   VALUES ("CcU-9999",
341           "AR45128465789502026154",
342           "3143587964857695",
343           "5521",
344           "989",
345           "2025-07-31"
346          );
347
348 • SELECT *
349   FROM credit_card
350  WHERE id = "CcU-9999";
351
```

Result Grid						
Filter Rows: <input type="text"/>						
Edit:      Export/Import:						
	id	iban	pan	pin	cvv	expiring_date
▶	CcU-9999	AR45128465789502026154	3143587964857695	5521	989	2025-07-31
*	NULL	NULL	NULL	NULL	NULL	NULL

Una vez agregada esta información, podemos insertar el valor requerido en la tabla transaction:

```
352 • INSERT INTO transaction
353 VALUES (
354     "108B1D1D-5B23-A76C-55EF-C568E49A99DD",
355     "CcU-9999",
356     "b-9999",
357     9999,
358     829.999,
359     -117.999,
360     CURRENT_TIMESTAMP(),
361     111.11,
362     0)
363 ;
364
365 • SELECT *
366 FROM transaction
367 WHERE company_id = "b-9999";
368
```

Result Grid									
Filter Rows: <input type="text"/>									
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	2024-03-04 16:53:21	111.11	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### - Ejercicio 4

Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit\_card. Recordar mostrar el canvi realitzat.

```
370 • ALTER TABLE credit_card
371 DROP COLUMN pan;
372
373 • SHOW COLUMNS FROM credit_card;
374
```

Result Grid						
Filter Rows: <input type="text"/>						
Export:						
Wrap Cell C						
	Field	Type	Null	Key	Default	Extra
▶	id	varchar(45)	NO	PRI	NULL	
	iban	varchar(45)	YES		NULL	
	pin	char(4)	YES		NULL	
	cvv	char(3)	YES		NULL	
	expiring_date	date	YES		NULL	

Comentario: Aquí arriba se muestran las columnas de la tabla credit\_card. Como podemos ver, la columna "pan" ha sido eliminada.

## Nivell 2

### Exercici 1

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

Comentario: Primero vemos el registro con id de transacción 02C6201E-D90A-1859-B4EE-88D2986D3B02:

```
382 • SELECT *
383 FROM transaction
384 WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
```

Result Grid   Filter Rows:   Edit:   Export/Import:   Wrap Cell Content:									
	id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶	02C6201E-D90A-1859-B4EE-88D2986D3B02	CcJ-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

Y a continuación lo eliminamos:

```
379 • DELETE FROM transaction
380 WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
381
382 • SELECT *
383 FROM transaction
384 WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02";
385
```

Result Grid   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

En la captura de arriba se puede ver que el registro con el id de transacción "02C6201E-D90A-1859-B4EE-88D2986D3B02" ya no figura en la tabla.

### 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.

```
394 • CREATE VIEW VistaMarketing AS
395 SELECT
396     c.company_name AS company,
397     c.phone,
398     c.country,
399     AVG(t.amount) AS avg_expense
400 FROM company c
401 JOIN transaction t
402     ON c.id = t.company_id
403 GROUP BY company, c.phone, c.country
404 ORDER BY 4 DESC;
405
406 • SELECT *
407 FROM VistaMarketing;
408
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
company	phone	country	avg_expense	
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 Ltd	00 00 47 65 00	Sweden	407.300000	

#	Time	Action	Message
965	17:11:38	CREATE VIEW VistaMarketing AS SELECT c.company_name AS company, c.phone, c.country, AVG(t.amount) AS a...	0 row(s) affected
966	17:11:41	SELECT * FROM VistaMarketing LIMIT 0, 1000	101 row(s) returned

Comentario: En la captura de arriba se muestra la vista solicitada por el equipo de Marketing, con el nombre de la compañía, teléfono, país y la media de gastos. La lista está ordenada de mayor a menor media de gastos, y si bien se ven unas pocas compañías en la captura, en la esquina inferior derecha se muestra que la lista contiene 101 empresas.



### Exercici 3

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

```
412 • SELECT *
413 FROM VistaMarketing
414 WHERE country = "Germany";
415
```

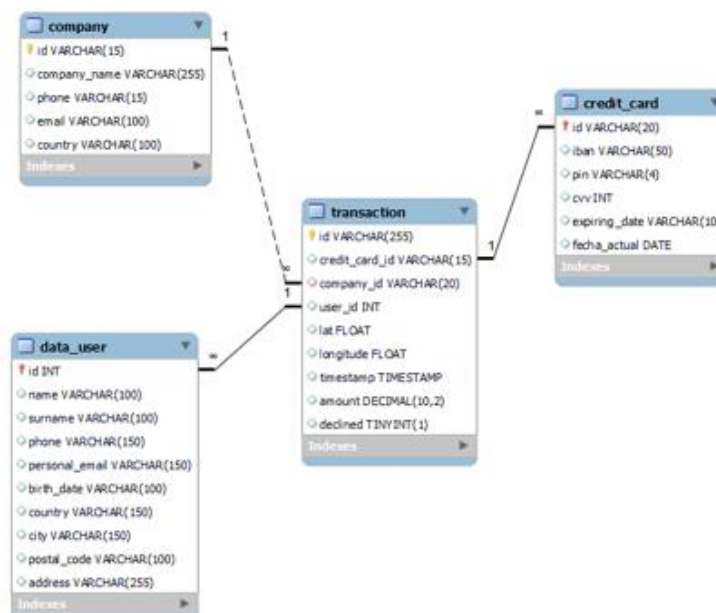
company	phone	country	avg_expense
Aliquam PC	01 45 73 52 16	Germany	385.265000
Ac Industries	09 34 65 40 60	Germany	289.645000
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000
Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.025238
Augue Foundation	06 88 43 15 63	Germany	240.800000
Ac Fermentum 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

Comentario: Aquí se muestran las 8 empresas con residencia en Alemania, ordenadas de mayor a menor gasto medio.

## 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):



Comentario: Lo primero que haremos será crear un índice en la columna “user\_id” de la tabla transaction, ya que estaremos utilizando este campo frecuentemente para unir la tabla transaction con “data\_user”. Luego, crearemos la tabla data\_user:

```
431 • CREATE INDEX idx_user_id ON transaction(user_id);
432 • CREATE TABLE IF NOT EXISTS data_user (
433     id INT PRIMARY KEY,
434     name VARCHAR(100),
435     surname VARCHAR(100),
436     phone VARCHAR(150),
437     email VARCHAR(150),
438     birth_date VARCHAR(100),
439     country VARCHAR(150),
440     city VARCHAR(150),
441     postal_code VARCHAR(100),
442     address VARCHAR(255)
443 );
444
445 • SHOW COLUMNS FROM data_user;
```

Result Grid						
Filter Rows:		Export:		Wrap Cell Content: <a href="#">IA</a>		
Field	Type	Null	Key	Default	Extra	
id	int	NO	PRI	NULL		
name	varchar(100)	YES		NULL		
surname	varchar(100)	YES		NULL		
phone	varchar(150)	YES		NULL		
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		

Luego añadimos una Foreign Key constraint a la columna “user\_id” de la tabla transaction, para poder relacionarla con columna “id” de la tabla data\_user:

```

447 • ALTER TABLE transaction
448     ADD CONSTRAINT
449     FOREIGN KEY (user_id)
450     REFERENCES data_user(id);
451
452 • SHOW COLUMNS FROM transaction;
453

```

Field	Type	Null	Key	Default	Extra
id	varchar(255)	NO	PRI	NULL	
credit_card_id	varchar(15)	YES	MUL	NULL	
company_id	varchar(20)	YES	MUL	NULL	
user_id	int	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	

Luego insertamos los valores y mostramos todos los registros de la tabla para corroborar que se hayan cargado correctamente:

```

718 • INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
719 • INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
720 • INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
721 • INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
722 • INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
723 • INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
724 • INSERT INTO data_user (id, name, surname, phone, email, birth_date, country, city, postal_code, address) VALUES (
725
726
727 • SET foreign_key_checks = 1;
728
729 • SELECT * FROM data_user;
730

```

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	Mconnell	(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.

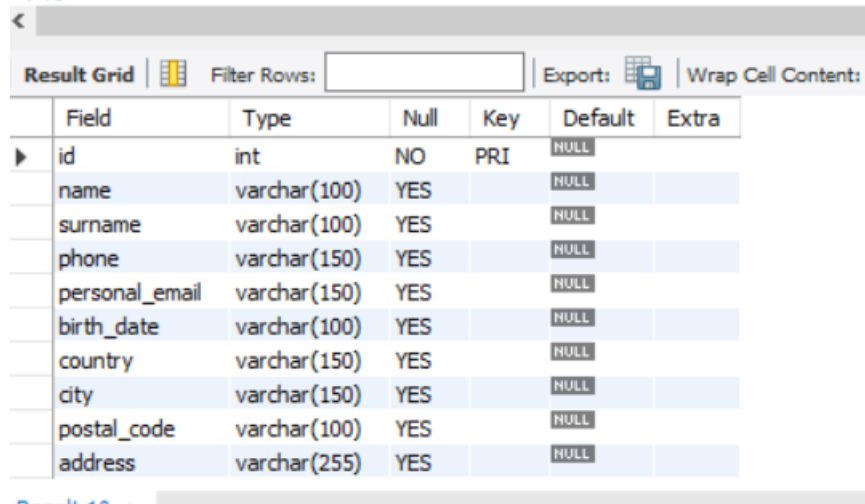
data\_user 3 x

Apply Revert

#	Time	Action	Message	Duration / Fetch
6	10:18:49	SHOW COLUMNS FROM data_user	10 row(s) returned	0.000 sec / 0.000 sec
7	10:22:00	SELECT * FROM data_user LIMIT 0, 1000	275 row(s) returned	0.000 sec / 0.000 sec

Ahora cambiamos el nombre de la columna “email” a “personal\_email” y mostramos el nombre actualizado de las columnas para corroborar que el cambio se ha hecho:

```
742 • ALTER TABLE data_user
743     RENAME COLUMN email to personal_email;
744
745 • SHOW COLUMNS FROM data_user;
746
```

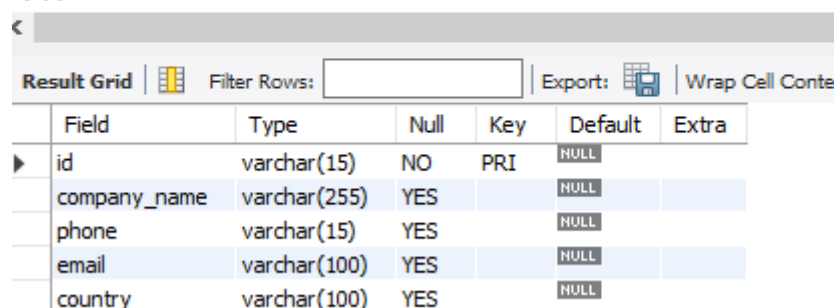


The screenshot shows a database client interface with a 'Result Grid' tab. The grid displays the columns of the 'data\_user' table. The columns are: id (int, NO, PRI, NULL), name (varchar(100), YES, NULL), surname (varchar(100), YES, NULL), phone (varchar(150), YES, NULL), personal\_email (varchar(150), YES, NULL), birth\_date (varchar(100), YES, NULL), country (varchar(150), YES, NULL), city (varchar(150), YES, NULL), postal\_code (varchar(100), YES, NULL), and address (varchar(255), YES, NULL). The 'personal\_email' column is highlighted, indicating it is the current row.

Field	Type	Null	Key	Default	Extra
id	int	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	

El siguiente paso será eliminar la columna “website” de la tabla company y, al igual que antes, mostramos el nombre actualizado de las columnas para corroborar que el cambio se ha hecho:

```
740 • ALTER TABLE company
741     DROP COLUMN website;
742
743 • SHOW COLUMNS FROM company;
744
```



The screenshot shows a database client interface with a 'Result Grid' tab. The grid displays the columns of the 'company' table. The columns are: id (varchar(15), NO, PRI, NULL), company\_name (varchar(255), YES, NULL), phone (varchar(15), YES, NULL), email (varchar(100), YES, NULL), and country (varchar(100), YES, NULL). The 'id' column is highlighted, indicating it is the current row.

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	

Por último, añadimos la columna “fecha\_actual” a la tabla credit\_card, cuyo tipo de datos es “Date”, y el valor por defecto es la fecha al momento de la creación de esta columna:

```
746 • ALTER TABLE credit_card
747 ADD fecha_actual DATE default (CURRENT_DATE);
748
749 • SELECT * FROM credit_card;
750
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	id	iban	pin	cvv	expiring_date	fecha_actual
▶	CcU-2938	TR323456312213576817699999	3257	984	2022-10-30	2024-03-04
	CcU-2945	DO26854763748537475216568689	9080	887	2023-08-24	2024-03-04
	CcU-2952	BG45IVQL52710525608255	4598	438	2021-06-29	2024-03-04
	CcU-2959	CR7242477244335841535	3583	667	2023-02-24	2024-03-04
	CcU-2966	BG72LKTQ15627628377363	4900	130	2024-10-29	2024-03-04
	CcU-2973	PT87806228135092429456346	8760	887	2025-01-30	2024-03-04
	CcU-2980	DE39241881883086277136	5075	596	2022-07-24	2024-03-04
	CcU-2987	GE89681434837748781813	2298	797	2023-10-31	2024-03-04
	CcU-2994	BH62714428368066765294	7545	595	2022-02-28	2024-03-04
	CcU-3001	CY49087426654774581266832110	9562	867	2022-09-16	2024-03-04
	CcU-3008	LU507216693616119230	1856	740	2025-04-05	2024-03-04
	CcU-3015	PS119398216295715968342456821	3246	822	2022-01-31	2024-03-04

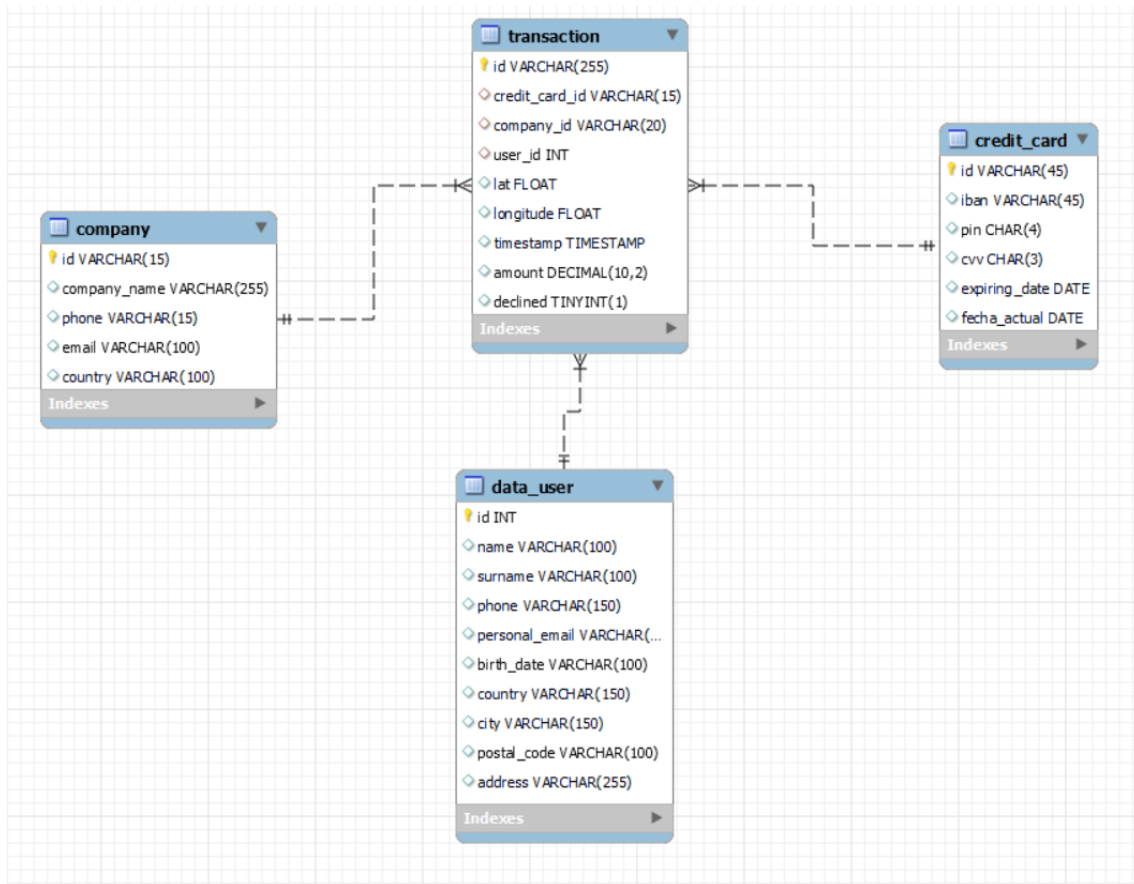
credit\_card 63 x

Output

Action Output

#	Time	Action	Message
✓ 1278	17:57:21	ALTER TABLE credit_card ADD fecha_actual DATE default (CURRENT_DATE)	276 row(s) affected Records: 276 Dupl
✓ 1279	17:57:24	SELECT * FROM credit_card LIMIT 0, 1000	276 row(s) returned

Tras efectuar los cambios mencionados, obtenemos el siguiente diagrama de estrella, con una tabla de hechos, **transaction** y tres tablas de dimensiones, **company**, **data\_user** y **credit\_card**.



## 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.

```

766 • CREATE VIEW InformeTecnico AS
767 SELECT
768     t.id AS transaction_id,
769     u.name AS user_name,
770     u.surname AS user_surname,
771     cr.iban,
772     co.company_name AS company
773 FROM transaction t
774 JOIN data_user u
775     ON t.user_id = u.id
776 JOIN credit_card cr
777     ON cr.id = t.credit_card_id
778 JOIN company co
779     ON co.id = t.company_id
780 ORDER BY 1 DESC;
781
782 • SELECT * FROM InformeTecnico;

```

Result Grid					
Filter Rows: <input type="text"/>					
Export: <input type="button" value=""/>					
Wrap Cell Content: <input type="button" value=""/>					
	transaction_id	user_name	user_surname	iban	company
▶	FE96CE47-8D59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
	FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
	FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
	FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
	FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.

InformeTecnico 24 x

Output

Action Output				
#	Time	Action	Message	
636	10:59:56	SELECT * FROM transaction WHERE id = "02C6201E-D90A-1859-B4EE-88D2986D3B02" LIMIT 0, 1000	0 row(s) returned	
637	11:00:01	SELECT * FROM InformeTecnico LIMIT 0, 1000	587 row(s) returned	

Comentario: Arriba se muestra la vista "InformeTecnico" que hemos creado. Si bien se muestran unos pocos resultados, la lista contiene 587 resultados, como se ve en la esquina inferior derecha. Como se ha requerido, los resultados están ordenados por el transaction\_id de manera descendente.