

Sprint 3: Gestió de taules, índex i vistes

Tasca S3.01. Manipulació de taules

Nivell 1

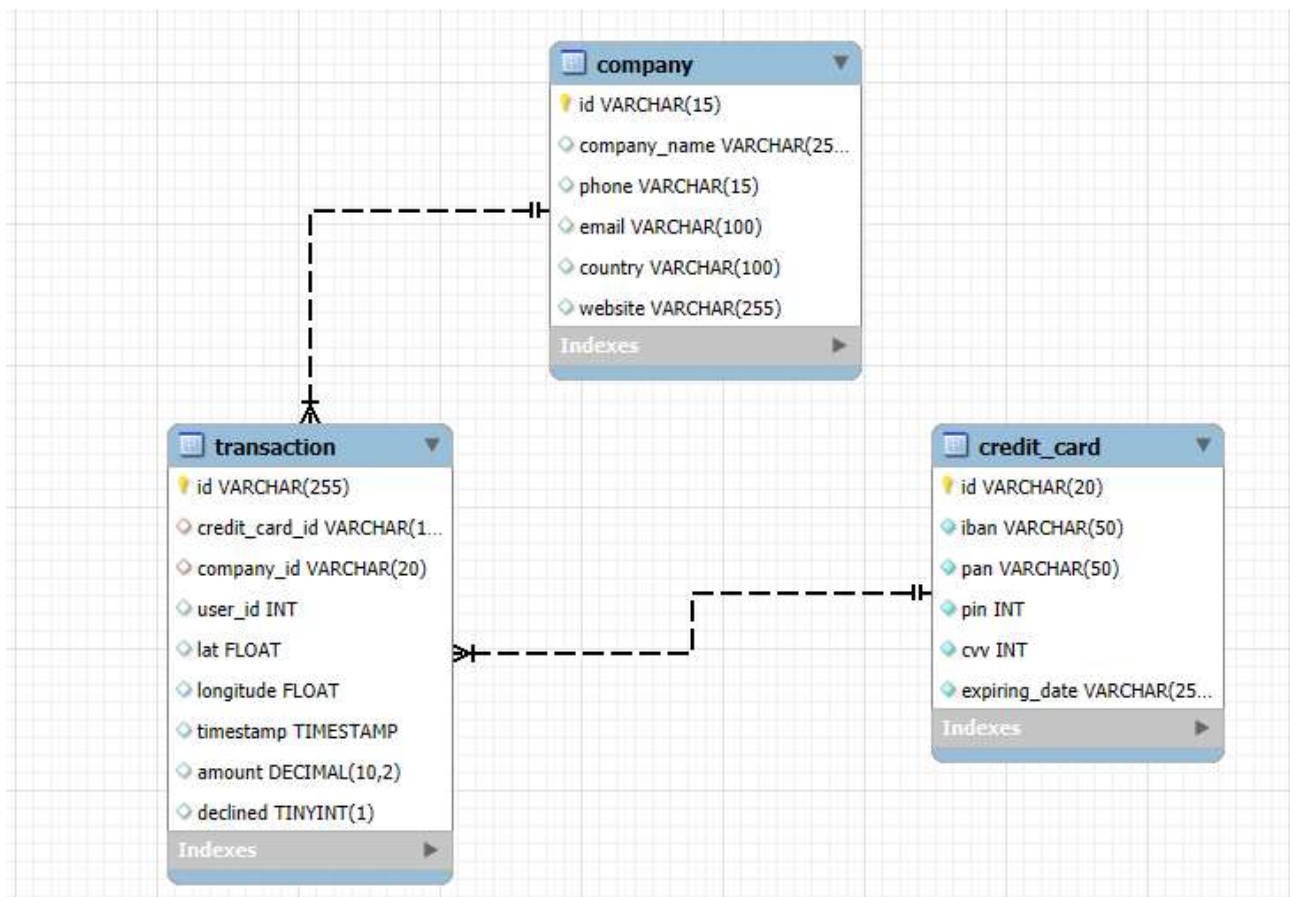
Exercici 1

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.

```
1  -- Nivel 1 / Ejercicio 1 --
2  CREATE TABLE IF NOT EXISTS credit_card (
3      id VARCHAR (20),
4      iban VARCHAR (50) NOT NULL,
5      pan VARCHAR (50) NOT NULL,
6      pin INT NOT NULL,
7      cvv INT NOT NULL,
8      expiring_date VARCHAR (255) NOT NULL,
9      PRIMARY KEY (id));
10
11  #INSERT INTO credit_card ()
12  -- VALUES () -- se ejecutan los datos para cargar la información proporcionada relativa a la tabla 'credit_card'
13
14  ALTER TABLE transaction
15  ADD CONSTRAINT FK_TransactionCreditCard
16  FOREIGN KEY (credit_card_id) REFERENCES credit_card(id); -- Se define una restricción de FK en la tabla hijo (transaction)
```

#Creamos la tabla "credit_card", considerando los valores a insertar, de la información proporcionada "dades_introduir_credit" y el tipo de datos para cada registro. Luego de ejecutada la instrucción de INSERT, con los valores correspondientes, se genera una restricción mediante la instrucción ADD CONSTRAINT, para establecer la clave foránea (credit_card_id) en la tabla hijo (transaction) que hace referencia hacia la tabla padre (credit_card).

Una vez creada la tabla "credit_card" y ejecutadas las instrucciones necesarias, realizamos el diagrama de relaciones. Donde se puede observar la cardinalidad establecida (1:N), entre la nueva tabla de dimensiones (credit_card), que se suma a la ya existente, (company) y la tabla de hechos (transaction). Es decir, que las relaciones serán de (1) tarjeta de crédito puede realizar entre 1 o muchas transacciones, y en el sentido opuesto, solo un id de la tabla "transaction" corresponderá a una sola y única operación efectuada con esa tarjeta de crédito.



Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte associat a la targeta de crèdit amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: TR323456312213576817699999. Recorda mostrar que el canvi es va realitzar.

```

17  #Exercici 2 ---
18  *  SELECT *
19    FROM credit_card
20    WHERE id = 'CcU-2938';
21

```

id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL	NULL

#Efectuamos la instrucción SELECT para observar los datos contenidos en el registro, y filtramos mediante la instrucción WHERE con el ID “ CcU-2938”, para obtener la línea correspondiente.

Verificado esto, ejecutamos la instrucción UPDATE, de la tabla “credit_card” para actualizar el registro del “iban” indicado. Y mediante WHERE filtramos la orden para ese registro ID en particular. Luego, mediante la orden SELECT chequeamos que el UPDATE del campo solicitado, se ejecutó correctamente.

```

22 * UPDATE credit_card
23     SET iban = 'TR323456312213576817699999'
24     WHERE id = 'CcU-2938';
25
26 * SELECT *
27     FROM credit_card
28     WHERE id = 'CcU-2938';
29
30

```

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

```

33 #Exercici 3--
34 * INSERT INTO transaction
35     (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
36     VALUES
37     ('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0');
38
39

```

Output

#	Time	Action	Message
1	11:20:58	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, decli...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('transactions'...

#Al ejecutar la instrucción INSERT INTO en la tabla “transaction”, obtenemos un error en la acción a realizar:

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`transactions`.`transaction`, CONSTRAINT `transaction_ibfk_1` FOREIGN KEY (`company_id`) REFERENCES `company` (`id`))

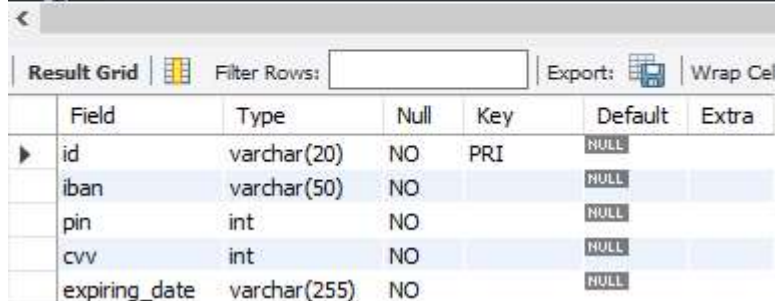
Debido a la restricción de las claves foráneas, (credit_card_id / user_id) el o los valores a insertar en la tabla hijo (transaction), deben existir en las tablas padre (credit_card/company), o la inserción fallará. Para solucionar esto, se debería solicitar la información correspondiente, para completar los

campos y sus registros, en las tablas "credit_card" y "company", para luego poder realizar la carga de los valores solicitados, en la tabla "transaction".

Exercici 4

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

```
48 • ALTER TABLE credit_card
49 DROP COLUMN pan;
50
51 • SHOW COLUMNS FROM credit_card;
```



	Field	Type	Null	Key	Default	Extra
▶	id	varchar(20)	NO	PRI	NULL	
	iban	varchar(50)	NO		NULL	
	pin	int	NO		NULL	
	cvv	int	NO		NULL	
	expiring_date	varchar(255)	NO		NULL	

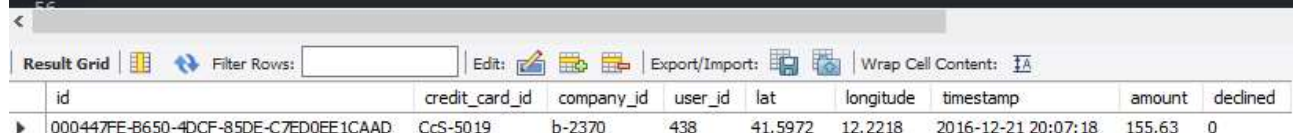
#Mediante la instrucción ALTER TABLE de la tabla "credit_card", y DROP COLUMN, realizamos el borrado de la columna "pan" con todos sus registros. Para verificar la acción, ejecutamos la instrucción SHOW COLUMNS y observamos que efectivamente la columna "pan" ha sido eliminada de la tabla "credit_card".

Nivell 2

Exercici 1

Elimina de la taula transaction el registre amb ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de dades.

```
53 #Nivell 2 - Exercici 1 ---
54 • SELECT * FROM transaction
55 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
```



id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
▶ 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	CcS-5019	b-2370	438	41.5972	12.2218	2016-12-21 20:07:18	155.63	0

#Verificamos los valores del registro que se nos solicita eliminar, mediante SELECT y filtramos con la instrucción WHERE para seleccionar el ID indicado.

```

56 * DELETE FROM transaction
57 WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
58
59 * SELECT * FROM transaction
60 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

Mediante la instrucción DELETE FROM, y el filtrado con WHERE, para el ID solicitado, ejecutamos y procedemos a eliminar los valores de ese registro. Verificamos luego, mediante SELECT y filtrando por el id solicitado, que el registro fue correctamente eliminado.

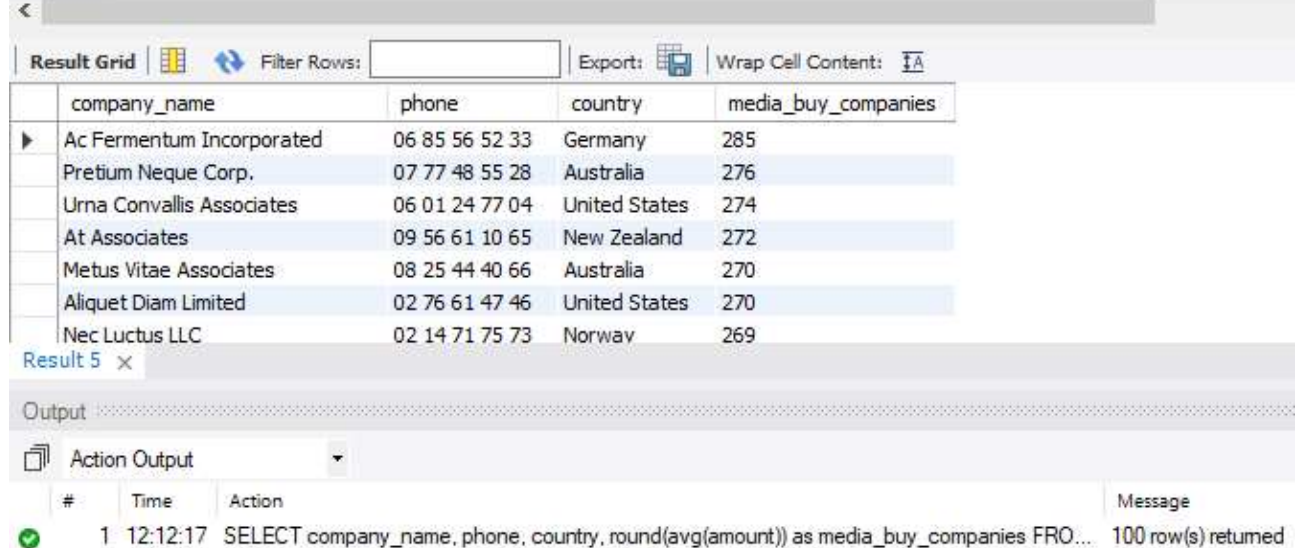
Exercici 2

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.

```

63 * SELECT company_name, phone, country, round(avg(amount)) as media_buy_companies
64 FROM company c
65 JOIN transaction t
66 ON c.id = t.company_id
67 GROUP BY c.company_name, c.phone, c.country
68 ORDER BY media_buy_companies DESC;

```



company_name	phone	country	media_buy_companies
Ac Fermentum Incorporated	06 85 56 52 33	Germany	285
Pretium Neque Corp.	07 77 48 55 28	Australia	276
Urna Convallis Associates	06 01 24 77 04	United States	274
At Associates	09 56 61 10 65	New Zealand	272
Metus Vitae Associates	08 25 44 40 66	Australia	270
Aliquet Diam Limited	02 76 61 47 46	United States	270
Nec Luctus LLC	02 14 71 75 73	Norway	269

Result 5 x

Output

Action Output

#	Time	Action	Message
1	12:12:17	SELECT company_name, phone, country, round(avg(amount)) as media_buy_companies FRO...	100 row(s) returned

#Ejecutamos las instrucciones SELECT/FROM/JOIN/GROUP BY/ORDER BY para obtener de los campos indicados, y las tablas que contengan la información requerida.


```
62 CREATE VIEW VistaMarketing AS
63 SELECT company_name, phone, country, round(avg(amount)) as media_buy_companies
64 FROM company c
65 JOIN transaction t
66 ON c.id = t.company_id
67 GROUP BY c.company_name, c.phone, c.country
68 ORDER BY media_buy_companies DESC;
69
```

Output

Action Output

#	Time	Action	Message
1	12:14:58	CREATE VIEW VistaMarketing AS SELECT company_name, phone, country, round(avg(amoun...	0 row(s) affected

#Comprobada la consulta anterior, realizamos mediante la función CREATE VIEW, la vista denominada “VistaMarketing”. Y ordenamos los datos, de la media de compras realizadas, de forma DESC (descendent).

Exercici 3

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

```
71 #Exercici 3
72 SELECT * FROM VistaMarketing
73 WHERE country = 'Germany';
```

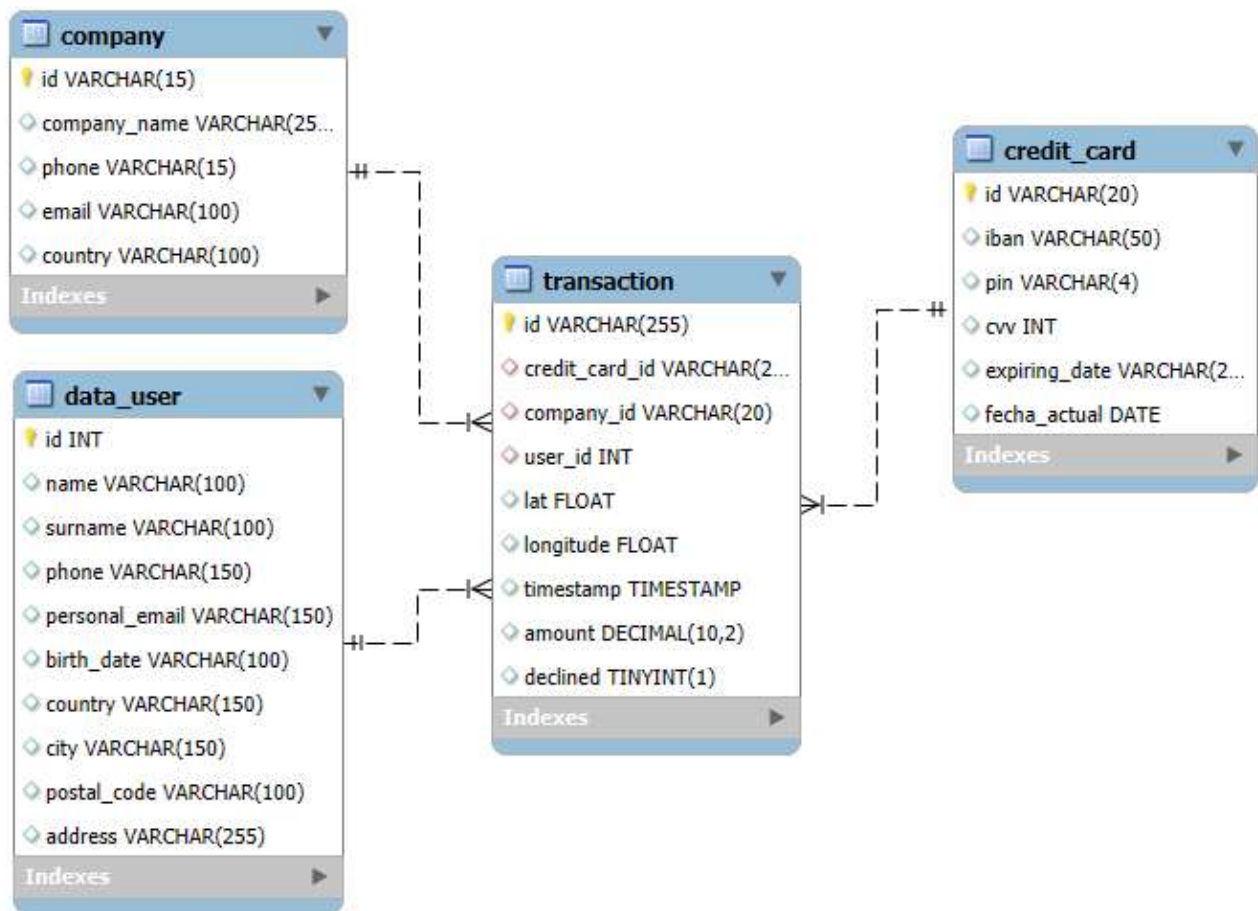
#Ejecutamos la instrucción SELECT/FROM de la vista creada anteriormente, (VistaMarketing) y filtrado mediante WHERE, los registros donde el país sea “Germany”. Obtenemos 8 filas.

	company_name	phone	country	media_buy_companies
▶	Ac Fermentum Incorporated	06 85 56 52 33	Germany	285
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	259
	Convallis In Incorporated	06 66 57 29 50	Germany	258
	Ac Industries	09 34 65 40 60	Germany	255
	Auctor Mauris Corp.	05 62 87 14 41	Germany	255
	Rutrum Non Inc.	02 66 31 61 09	Germany	255
	Augue Foundation	06 88 43 15 63	Germany	254
	Aliquam PC	01 45 73 52 16	Germany	253

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 el següent diagrama:



#A partir del diagrama indicado, primero creamos la tabla “user”, a partir de los campos indicados y los valores de los tipos de datos de cada uno.

```

79  #Nivell 3 - Exercici 1 ---
80  * CREATE TABLE IF NOT EXISTS user (
81      id INT NOT NULL PRIMARY KEY, -- se cambia el tipo de dato a un INT, para que concuerde con el expresado en el DIAGRAMA--
82      name VARCHAR(100),
83      surname VARCHAR(100),
84      phone VARCHAR(150),
85      email VARCHAR(150),
86      birth_date VARCHAR(100),
87      country VARCHAR(150),
88      city VARCHAR(150),
89      postal_code VARCHAR(100),
90      address VARCHAR(255)
91  );
92

```

Output

#	Time	Action	Message
1	12:35:59	CREATE TABLE IF NOT EXISTS user (id INT NOT NULL PRIMARY KEY, -- se cambia el tipo ...	0 row(s) affected

#Utilizamos los archivos proporcionados, "estructura_dades_user" i "dades_introduir_user". Para que el tipo de datos concuerde, cambiamos el tipo de dato del campo id, a un INT, dado que en el archivo suministrado, "estructura_dades_user" figura como CHAR.

```
Query 1 Sprint 3* estructura datos user x
1 CREATE TABLE IF NOT EXISTS user (
2     id CHAR(10) PRIMARY KEY,
3     name VARCHAR(100),
4     surname VARCHAR(100),
5     phone VARCHAR(150),
6     email VARCHAR(150),
7     birth_date VARCHAR(100),
8     country VARCHAR(150),
9     city VARCHAR(150),
10    postal_code VARCHAR(100),
11    address VARCHAR(255)
12 );
```

#Ya creada la tabla “user” procedemos a la carga de los datos, mediante la información contenida en el archivo “dades_introducir_user”. Y luego ejecutamos mediante ALTER TABLE/ADD CONSTRAINT, definimos la restricción correspondiente y poder establecer la relación entre claves, de la tabla “user” y la tabla “transaction”. Donde la FOREIGN KEY, de la tabla “transaction” será (user_id), la cual hace referencia al PRIMARY KEY (id) de la tabla “user”.

```
97 * ALTER TABLE transaction
98 ADD CONSTRAINT FKTransactionUser
99 FOREIGN KEY (user_id) REFERENCES user(id); -- Se define una restricción de FK en la tabla hijo (transaction)
100
```

Output

#	Time	Action	Message
5001	12:39:33	INSERT INTO user (id, name, sumame, phone, email, birth_date, country, city, postal_code, a...	1 row(s) affected
5002	12:48:34	ALTER TABLE transaction ADD CONSTRAINT FKTransactionUser FOREIGN KEY (user_id) ...	99999 row(s) affected Records: 99999 Duplicates: 0 Warnings: 0

#Modificamos la tabla “credit_card”, para que las tablas y sus campos/valores concuerden con el diagrama indicado. Para ello mediante la instrucción ALTER TABLE/ADD COLUMN añadimos una nueva columna, “fecha_actual” con tipo de dato DATE.

#Modificamos también la tabla “company”, para que concuerde con el diagrama. En este caso eliminamos la columna “website” mediante la instrucción ALTER TABLE/DROP COLUMN.

```
96 #MODIFICAR TABLA credit_card, insertamos nueva columna fecha_actual DATE
97 * ALTER TABLE credit_card
98 ADD COLUMN fecha_actual DATE;
99
100 #MODIFICAR TABLA company, eliminar columna website
101 * ALTER TABLE company
102 DROP COLUMN website;
```


Exercici 2

L'empresa també us demana 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.
- Assegureu-vos d'incloure informació rellevant de les taules que coneixereu i utilitzeu àlies per canviar de nom columnes segons calgui.

Mostra els resultats de la vista, ordena els resultats de forma descendent en funció de la variable ID de transacció.

```
104 SELECT t.id AS id_transaccion, u.name as nombre_usuario, u.surname as apellido_usuario,
105        cc.iban as iban_tarjeta_usada, c.company_name as nombre_compania_transaccion
106 FROM transaction t
107 JOIN user u
108 ON t.user_id = u.id
109 JOIN company c
110 ON t.company_id = c.id
111 JOIN credit_card cc
112 ON t.credit_card_id = cc.id
113 ORDER BY id_transaccion DESC;
```

	id_transaccion	nombre_usuario	apellido_usuario	iban_tarjeta_usada	nombre_compania_transaccion
▶	FFFD31D6-9495-47CE-B54A-7D88E1CC274B	Bmrgli	Tprvmrc	XX794814451211289182490922	Turpis Company
	FFFCF76D-ECF0-4985-A2D0-B2A7B75998FC	Dfrled	Vilqcdl	XX636251701647892036676034	Amet Nulla Donec Corporation
	FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faofvqfy	XX162677143304223631437567	Nunc Interdum Incorporated
	FFFB270D-F53A-4D5D-9666-E5307C53CC84	Ggzjpa	Uirzjulh	XX395114267082019952567052	Viverra Donec Foundation
	FFF9E3CE-234E-408C-A8EF-F9CAD577224A	Yshimq	Zpsjsleed	XX8845462156537570367941	Convallis In Incorporated
	FFF9E178-6CD2-4DF9-99B0-49AE068809B1	Jevepx	Xxwczwnm	XX321405515711654384711481	Mus Aenean Eget Foundation
	FFF867C9-17B5-4B1F-AFD9-F8023AAA449E	Fqlngd	Lvhfqyxi	XX278446342932680979729426	Cras Vehicula Aliquet Industries
	FFF7042D-18C6-4DDD-823C-4D90A4AC8F26	Njoraa	Eqsqucii	XX405009272572550082027209	Placerat LLP

#Primero ejecutamos la consulta, para obtener los datos e información solicitada. Utilizamos ALIAS para cada uno de los campos a visualizar en la nueva vista a crear. La consulta devuelve un total de 99999 filas.

```
101      #Exercici 2 --
102
103  * CREATE VIEW InformeTecnico AS
104  SELECT t.id AS id_transaccion, u.name as nombre_usuario, u.surname as apellido_usuario,
105  cc.iban as iban_tarjeta_usada, c.company_name as nombre_compania_transaccion
106  FROM transaction t
107  JOIN user u
108  ON t.user_id = u.id
109  JOIN company c
110  ON t.company_id = c.id
111  JOIN credit_card cc
112  ON t.credit_card_id = cc.id
113  ORDER BY id_transaccion DESC;
```

#Creamos la vista, “InformeTecnico” con los datos de la consulta verificada en el paso anterior. La información aparecerá ordenada por el valor de la columna denominada “id_transaccion” de mayor a menor.