

## Sprint 3 - Thais Alcaide

### - 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.**

Utilitzo create table i afegeixo les següents columnes:

- Id, character varchar de 15, pel que veig a les dades no sobrepassa aquest llarg. Es una primary key i es imprescindible que s'afegeixi aquesta informació per identificar.
- Iban, character varchar de 50, porque els iban més llargs nomès arriben fins a 33.
- Pan, character varchar de 20, tenen llargada entre 12 i 19.
- Pin, varchar de 4, sempre té la mateixa longitud.
- Cvv, int
- Expiring\_date, volia fer valor date, pero les dades que tenim no estan amb el format correcte a l'arxiu d'origen (any, mes, dia) pel que he hagut de fer format varchar(15). Això pot donar problemes si més endavant volem fer filtres per data.
- Afegeixo un index per optimitzar el rendiment.

```

• create table credit_card (
    id varchar(15) PRIMARY KEY,
    iban varchar(50),
    pan varchar(20),
    pin varchar(4),
    cvv int,
    expiring_date varchar(10),
    INDEX(id)
);

```

```

create table credit_card (id varchar(15) PRIMARY KEY,  iban varchar(50),  pan varchar(20),  pin varchar(4),  cvv int,  expiring_date varchar(10),  INDEX(id)) 0 row(s) affected

```

Després he carregat totes les dades que ens ha proporcionat l'exercici.

Al revisar aquesta nova taula per possibles relacions amb la resta de taules, he vist que l'id esta relacionat amb la taula transaction a traves de credit\_card\_id com a foreign key, així que he modificat la taula transaction per afegir aquesta FK:

```

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

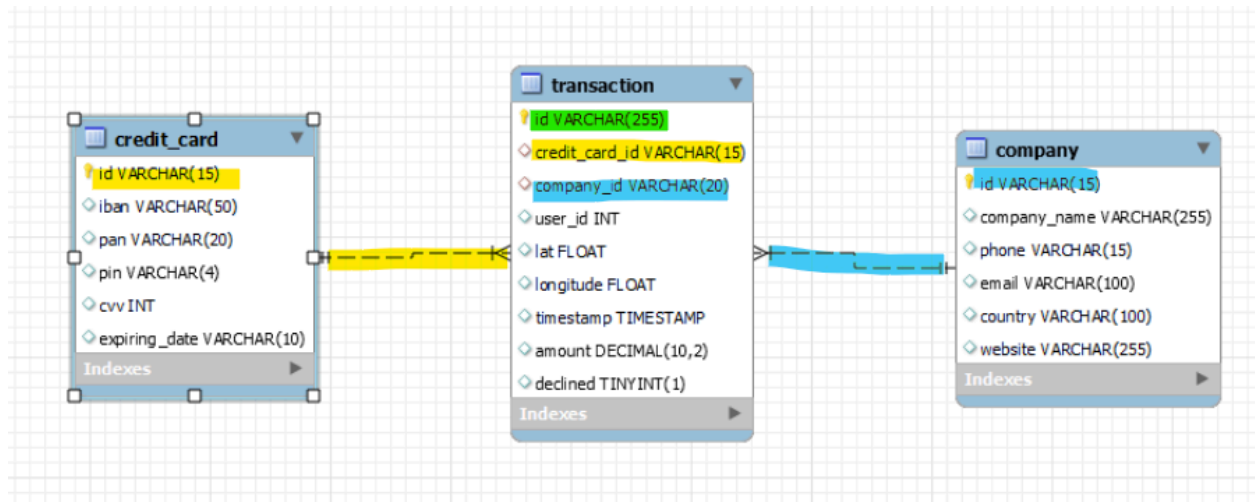
```

```

ALTER TABLE transaction ADD CONSTRAINT FK_credit_card FOREIGN KEY (credit_card_id) REFERENCES credit_card(id) 586 row(s) affected Records: 586 Duplicates: 0 Warnings: 0

```

L'esquema final que queda son 3 taules en format estrella, on la taula transaccions actua de nexa principal i està directament relacionada amb les altres dues taules a les columnes credit\_card\_id i company\_id. Tenen una relació 1 a N, la taula credit\_card a l'id amb la taula transaction a credit\_card\_id. I la taula company a l'id amb la taula transaction a company\_id. He marcat amb colors per fer més intuïtives les relacions del mateix color.



## - Exercici 2

El departament de Recursos Humans ha identificat un error en el número de compte de l'usuari amb ID CcU-2938. La informació que ha de mostrar-se per a aquest registre és: **R323456312213576817699999**. Recorda mostrar que el canvi es va realitzar.

Si busquem l'informació actual del CcU-2938 ens dona això:

```

SELECT *
FROM credit_card
WHERE id = 'CcU-2938';
  
```

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

Per canviar la informació a una taula s'ha de fer un update, marcar el canvi al SET, en aquest cas l'iban que volem canviar, i especificar a on amb un WHERE.

```

9 • UPDATE credit_card
0 SET iban = 'R323456312213576817699999'
1 WHERE id = 'CcU-2938';
2
3

```

UPDATE credit\_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938' 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

Si ara demanem visualitzar l'informació del id CcU-2938 ens dona el nou iban:

id	iban	pan	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22
NULL	NULL	NULL	NULL	NULL	NULL

SELECT \* FROM credit\_card WHERE id = 'CcU-2938' LIMIT 0, 1000 1 row(s) returned

### - Exercici 3

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

Al intentar insertar aquesta informació a la taula em dona error. Després de mirar la taula original, es perquè les dades de CcU-9999 i b-9999 no existeixen. L'error que surt es perquè a la taula transaction l'id de la companyia i de la credit card son foreign keys i necessiten que els valors coincideixin amb valors a les taules pare (company i credit\_card). Per tant he de crear aquests registres a les taules corresponents:

Creo la companyia b-9999 amb un INSERT INTO a la taula company:

```

INSERT INTO company (id, company_name, phone, email, country, website)
VALUES ('b-9999', 'Barcelona Activa', '900 533 175', 'info@barcelonactiva.cat', 'Spain', 'https://www.barcelonactiva.cat/');

```

INSERT INTO company (id, company\_name, phone, email, country, website) VALUES ('b-9999', 'Barcelona Activa', '900 533 175', 'info@barcelonac... 1 row(s) affected

Creo la credit card CcU-9999 amb un INSERT INTO a la taula credit\_card:

```
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date)
VALUES ('CcU-9999', 'TR301950312213576817638662', '5424465566813634', '3258', '985', '10/30/23');
```

```
INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-9999', 'TR301950312213576817638662', '5424465566813634', '3... 1 row(s) affected
```

I finalment puc afegir la informació demanada a l'exercici a la taula transaction:

```
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);
```

```
INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ('108B1D1D-5B23-A76C-55EF-C568E49A9... 1 row(s) affected
```

## - Exercici 4

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

Per fer-ho s'utilitza el comando alter table, en aquest cas ens interessa de credit\_card, i per eliminar la columna pan s'utilitza Drop Column:

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

```
ALTER TABLE credit_card DROP COLUMN pan 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
```

Si ara volem veure les dades de la taula credit\_card, podem observar que ja no està la columna de pan:

Result Grid						
Filter Rows: <input type="text"/>						
	id	iban	pin	cvv	expiring_date	
▶	CcU-2938	R323456312213576817699999	3257	984	10/30/22	
	CcU-2945	DO26854763748537475216568689	9080	887	08/24/23	
	CcU-2952	BG45IVQL52710525608255	4598	438	06/29/21	
	CcU-2959	CR7242477244335841535	3583	667	02/24/23	
	CcU-2966	BG72LKTQ15627628377363	4900	130	10/29/24	

## Nivell 2

### Exercici 1

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

Per eliminar un registre s'utilitza el comando Delete from (nom de la taula, transaction) i WHERE per especificar quin vols eliminar, en aquest cas desde l'id que es la informació que ens donen:







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

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

1 row(s) affected

Si ara busquem aquest id veurem que la informació ha quedat eliminada:

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

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
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT * FROM transaction WHERE id='02C6201E-D90A-1859-B4EE-88D2986D3B02' LIMIT 0, 1000
```

0 row(s) returned




## 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.**

Per fer aquesta visualització necessitem fer un join de les taules transaction i company, ja que la informació que es demana està repartida entre aquestes dues taules. També calculem l'average i utilitzem el groupby perquè reculli tota la informació de cada empresa amb la seva mitjana. He utilitzat el següent codi per crear aquesta vista:

```
CREATE VIEW `vistamarketing` AS
SELECT company_name, phone, country, round(AVG(amount),2) AS 'Mitjana'
FROM company AS c
LEFT JOIN transaction AS t ON c.id=t.company_id
GROUP BY company_name, phone, country
```

```
SELECT *
FROM vistamarketing
ORDER BY Mitjana DESC;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 				
	company_name	phone	country	Mitjana
▶	Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
	Sed Id Limited	07 28 18 18 13	United States	477.51
	Neque Tellus Incorporated	04 43 18 34 19	Ireland	477.10
	Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
	Non Magna LLC	06 71 73 13 17	United Kingdom	458.74

vistamarketing 6 

SELECT \* FROM vistamarketing ORDER BY Mitjana DESC LIMIT 0, 1000 101 row(s) returned

### Exercici 3

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

Utilitzem el view que hem creat abans i afegim un filtre pel país demanat:

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

	company_name	phone	country	Mitjana
►	Ac Fermentum Incorporated	06 85 56 52 33	Germany	293.57
	Convallis In Incorporated	06 66 57 29 50	Germany	60.99
	Nunc Interdum Incorporated	05 18 15 48 13	Germany	242.95
	Augue Foundation	06 88 43 15 63	Germany	15.05
	Ac Industries	09 34 65 40 60	Germany	396.15
	Auctor Mauris Corp.	05 62 87 14 41	Germany	308.99
	...	...	...	...

vistamarketing 7

```
SELECT * FROM vistamarketing WHERE country = 'Germany' LIMIT 0, 1000
```

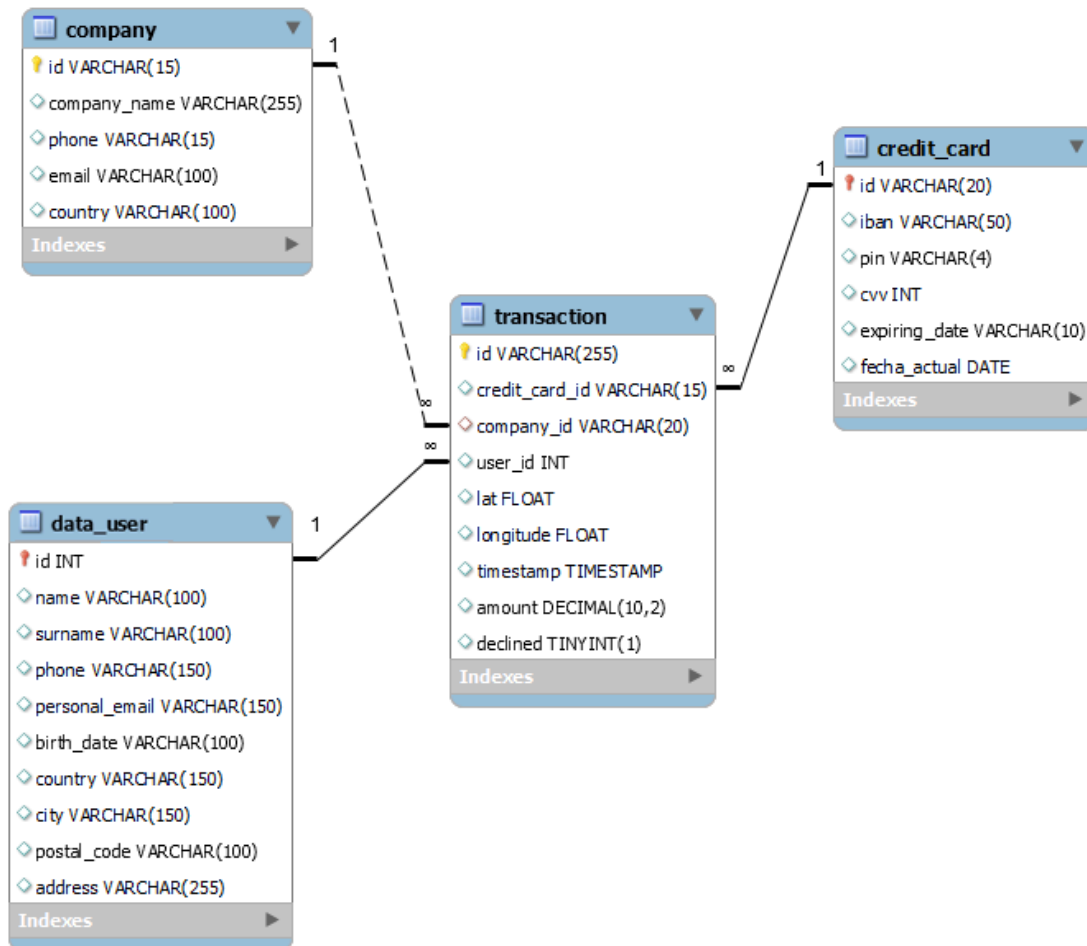
8 row(s) returned



## 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:



Els canvis necessaris per arribar a aquest nou diagrama son els següents:

- 1- Creació d'una nova taula, anomenada data\_user, on la primary key es la columna id i està relacionada 1 a N amb la taula transaction, columna user\_id. Utilitza formats varchar innecesariament grans que no son els més eficients. Aquesta taula va ser creada amb el següent comando que es proporciona als adjunts de l'exercici:

```
CREATE INDEX idx_user_id ON transaction(user_id)
```

```
CREATE INDEX idx_user_id ON transaction(user_id)
```

```
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
```

```
> 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)  
);
```

- 2- En aquesta mateixa taula també va canviar després el nom de la columna email a personal\_email, amb el comando alter table i change (el format varchar l'he deixat igual):

```
ALTER TABLE data_user  
CHANGE email personal_email VARCHAR(150);
```

```
ALTER TABLE data_user CHANGE email personal_email VARCHAR(150)
```

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

- 3- I també va canviar el nom de la taula a data\_user, en aquest cas s'utilitza rename:

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

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

0 row(s) affected

- 4- A la taula company va eliminar la columna website, això es fa amb un alter table i drop column:

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

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

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

- 5- A la taula transaction va treure el Foreign Key que vam afegir a credit\_card\_id a l'exercici del nivell 1. Es fa amb un alter table (en aquest cas era de la taula transaction) i s'especifica un drop foreign key amb el nom que vam assignar anteriorment:

```
ALTER TABLE transaction  
DROP FOREIGN KEY FK_credit_card;
```

```
ALTER TABLE transaction DROP FOREIGN KEY FK_credit_card
```

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

- 6- A la taula credit\_card ha canviat el format de la columna id de varchar 15 a 20, s'especifica la columna a modify column i afegeixes el nou format seguidament:

```
ALTER TABLE credit_card  
MODIFY COLUMN id VARCHAR(20);
```

```
ALTER TABLE credit_card MODIFY COLUMN id VARCHAR(20)
```

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

7- A la taula credit\_card ha afegit una nova columna anomenada fecha\_actual amb format date:

```
ALTER TABLE credit_card  
ADD fecha_actual DATE;
```

```
ALTER TABLE credit_card ADD fecha_actual DATE
```

```
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
```

## 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.**

Per crear aquesta vista que engloba dades de totes les taules s'han de fer varis left joins per anar afegint tota la informació a una sola taula, en aquest cas ho he fet desde la taula transaccions. He creat el següent codi, afegint àlies a tots els camps per fer la vista més còmoda als usuaris que utilitzin aquesta eina:

```
CREATE VIEW `informetecnico` AS
```

SELECT t.id AS "ID de la transacció", u.name AS "Nom de l'usuari/ària", u.surname AS "Cognom de l'usuari/ària", cc.iban AS "IBAN de la targeta de crèdit usada.", c.company\_name AS "Nom de la companyia de la transacció realitzada."

FROM transaction AS t

LEFT JOIN data\_user AS u ON u.id=t.user\_id

LEFT JOIN credit\_card AS cc ON cc.id=t.credit\_card\_id

LEFT JOIN company AS c ON c.id=t.company\_id;

```
SELECT *  
FROM informetecnico  
ORDER BY 'ID de la transacció' DESC;
```

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.
0466A42E-47CF-8D24-FD01-C0B689713128	William	Benjamin	MD1234119525145401270486	Nunc Interdum Incorporated
063FBA79-99EC-66FB-29F7-25726D1764A5	Kenyon	Hartman	GE89681434837748781813	Amet Nulla Donec Corporation
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	Chloe	Keith	BA542358041365401657	Non Institute
06CD9AA5-9B42-D684-DDDD-A5E394FEBA99	Lynn	Riddle	CR7242477244335841535	Ut Semper Foundation
07A46D48-31A3-7E87-65B9-0DA902AD109F	Hedwig	Gilbert	F19398462343991818	Lacus Quisque Associates
09DE92CE-6F27-2B87-13B5-9385B2B388E2	Kenyon	Hartman	NO8923814763512	Elit Etiam Laoreet Associates
0A476ED9-0C13-1962-F87B-D3563924B539	Sasha	Emerson	PK1842602717357586645371	Nunc Interdum Incorporated
0BEB80B7-9D66-1707-CE4B-9DC7E71914B5	Hedwig	Gilbert	CH4437804777669672438	Magna Incorporated
0C7C3A33-0947-3BC1-84FD-7BF3DD17508	Hedwig	Gilbert	CD7081008447841410311	Tritique Nunc Venenatis Institute

```
SELECT * FROM informetecnico ORDER BY 'ID de la transacció' DESC LIMIT 0, 1000
```

587 row(s) returned