

S3_01_Manipulació_de_taulas

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.

The screenshot shows the MySQL Workbench interface. On the left, a code editor displays the SQL script for creating the credit_card table and establishing foreign keys:

```
9 • USE transactions;
10 • CREATE TABLE credit_card (
11     id VARCHAR(20),
12     iban VARCHAR(50),
13     pan VARCHAR(50),
14     pin VARCHAR(4),
15     cvv INT,
16     expiring_date VARCHAR(20)
17 );
18
19 • ALTER TABLE credit_card
20     ADD PRIMARY KEY (ID);
21
22 • ALTER TABLE transaction
23     ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
```

To the right of the code editor, there is explanatory text about the `ALTER` statement:

To use `\L`, you need `ALTER`, `CREATE`, and `INSERT` privilege for the table. Renaming a table requires `ALTER` and `DROP` or old table, `ALTER`, `CREATE`, and `INSERT` on the new table.

Following the table name, specify the alterations to be made. If no alterations are given, `\L` does nothing.

The syntax for many of the permissible alterations is similar to the clauses of the `\L` statement. `column_definition` clauses have the same syntax for `ADD` and `CHANGE` as for `\L`. For more information, see the MySQL manual.

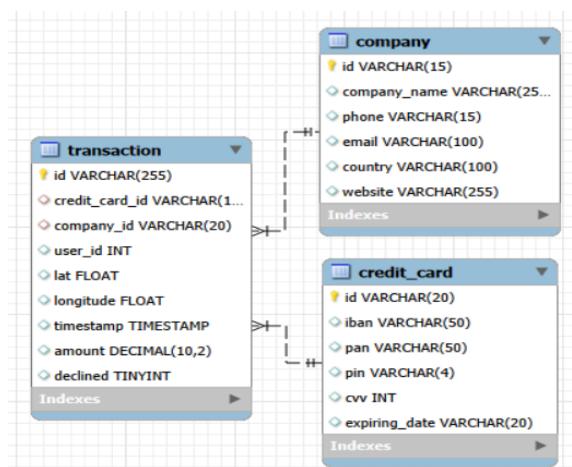
Below the code editor, the "Output" tab shows the execution results:

#	Time	Action	Message	Duration / File
5005	09:59:26	ALTER TABLE credit_card ADD PRIMARY KEY (ID)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.312 sec
5006	09:59:29	ALTER TABLE transaction ADD FOREIGN KEY (credit_card_id) RE... 100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0		2.265 sec

Primerament, crearem la taula amb les columnes amb una estructura basada en l'esquema de l'exercici 1 del Nivell 3, per no haver de formatar les columnes. En aquest cas, hem optat per afegir la relació entre taules a posteriori de la creació de credit_card, tot i que podria haver definit la clau primaria dins de la mateixa sentència de creació de la taula. Després, afegim la relació a la clau secundària a la foranea. Hem introduït les dades de "dades_introduir_credit.sql" desde un altre script, per no sobrecarregar l'arxiu sobre el que treballem. Com que cada dada està definida en una ordre diferent, executem tot l'script per a introduir-les a la taula.

Ens queda el següent esquema de taules:

Podem veure que ara, de la taula de fets (transaction), en neixen dues relacions a taules de dimensions (amb les dades de les companyies i de les targetes de credit, respectivament), relacionades amb les seves claus



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.

```
31 • UPDATE credit_card
32     SET iban = "TR323456312213576817699999"
33     WHERE id = "CcU-2938";
34
35 • select * FROM credit_card
36     WHERE id = "CcU-2938";
37
--  -  -  -
```

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window containing the provided SQL script. Below it is a 'Result Grid' window displaying the updated record in the 'credit_card' table. The grid has columns: id, iban, pan, pin, cvv, and expiring_date. The row for 'CcU-2938' now has 'iban' set to 'TR323456312213576817699999'. Below these windows is an 'Output' pane titled 'credit_card 2'. It contains two log entries: one for the update operation (5009) and one for the subsequent select query (5010). Both entries show successful execution with no errors.

Actualitzem el valor a la taula i comprovem que el canvi s'ha realitzat amb èxit.

Exercici 3

En la taula "transaction" ingressa una nova transacció 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

The screenshot shows the MySQL Workbench interface. In the code editor, a new transaction is being inserted into the 'transaction' table. The SQL command is as follows:

```
41  /* 108B1D1D-5B23-A76C-55EF-C568E49A99DD
42  credit_card_id CcU-9999
43  company_id b-9999
44  user_id 9999
45  lat 829.999
46  longitude -117.999
47  amount 111.11
48  declined 0*/
49
50 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ("108B1D1D-5B23-A76C-
```

Below the code editor is an 'Output' pane. It shows the execution of a 'SELECT * FROM transactions.company' query (log entry 5014) which returns 100 rows. Then, the insertion query (log entry 5015) is executed, but it fails with an error message: 'Error Code: 1452 Cannot add or update a child row: a foreign key constraint fails (transactions)'.

Intentem introduir les dades a través d'una nova ordre INSERT INTO, però desgraciadament, els vincles de les claus secundàries no ens permeten fer l'acció, perquè ni la companyia, ni la tarjeta de crèdit, consten a les taules de dimensions. Ja esbrinarem la resta de dades de la tarjeta i el client, però per ara, introduïm les dades que tenim (id de tarjeta i companyia) a les respectives taules, per poder afegir les dades sense problemes.

```

49
50 •  INSERT INTO company (id) VALUES ("b-9999");
51 •  INSERT INTO credit_card (id) VALUES ("CcU-9999");
52 •  INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALUES ("108B1D1D-5B23-A76C-
53

```

Output:

Action Output	#	Time	Action	Message
5016 10:55:08	INSERT INTO company (id) VALUES ("b-9999")			1 row(s) affected
5017 10:55:11	INSERT INTO credit_card (id) VALUES ("CcU-9999")			1 row(s) affected
5018 10:55:13	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VAL...			1 row(s) affected

Per sort, no tenim altres CONSTRAINT que ens limitin al crear dades amb columnes amb valor NULL.

Exercici 4

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

```

57 •  ALTER TABLE credit_card DROP COLUMN pan;
58 •  SELECT * FROM credit_card;

```

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

id	iban	pin	cvv	expiring_date
CcS-4857	XX4857591835292505850771	1819	467	09/27/25
CcS-4858	XX8581768137002436094025	3964	817	12/28/28
CcS-4859	XX7826930491423553609370	4983	277	11/26/26
CcS-4860	XX5559590368835304645299	6876	661	07/27/27
CcS-4861	XX2035182877195191627307	5710	398	04/25/26

credit card 3 x

Output:

Action Output	#	Time	Action	Message
5018 10:55:13	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VAL...			1 row(s) affected
5019 10:59:15	ALTER TABLE credit_card DROP COLUMN pan			0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
5020 10:59:20	SELECT * FROM credit_card			5001 row(s) returned

Per tal d'eliminar la columna "pan", farem servir l'ordre ALTER TABLE / DROP COLUMN. És satisfactori veure que el canvi s'ha portat a terme sense problemes.

Nivell 2

Exercici 1

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

```
64 •    DELETE FROM transaction WHERE id='000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
65 •    SELECT * FROM transaction
66      WHERE id='000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
67
```

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

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

transaction 4 x | Apply | Revert

Output | Action Output | # Time Action | Message

#	Time	Action	Message
5021	11:03:04	DELETE FROM transaction WHERE id='000447FE-B650-4DCF-85DE-C7ED0EE1CAAD'	1 row(s) affected
5022	11:03:50	SELECT * FROM transaction WHERE id='000447FE-B650-4DCF-85DE-C7ED0EE1CAAD'	0 row(s) returned

En aquest cas, eliminem les dades d'aquesta transacció en particular, amb l'ordre DELETE FROM, indicant les dades del "id" que ens han donat.

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.

```
75 •    CREATE OR REPLACE VIEW VistaMarketing AS
76      SELECT c.company_name, c.phone, c.country, ROUND(AVG(t.amount), 2) AS mitja_amount
77      FROM company c
78      JOIN transaction t ON t.company_id=c.id
79      GROUP BY c.id
80      ORDER BY AVG(amount) DESC;
81
82 •    SELECT * FROM VistaMarketing;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | VistaMarketing 4

company_name	phone	country	mitja_amount
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
Prætium Neque Corp.	07 77 48 55 28	Australia	276.16
Urna Convallis Associates	06 01 24 77 04	United States	274.24
At Associates	09 56 61 10 65	New Zealand	272.21
Metus Vitae Associates	08 25 44 40 66	Australia	270.08
Aliquet Diam Limited	02 76 61 47 46	United States	269.60
Nec Luctus LLC	02 14 71 75 73	Norway	268.60
Neque Tellus Incorporated	04 43 18 34 19	Ireland	267.85
Tellus Nulla Company	05 35 02 77 16	United States	267.84

VistaMarketing 4 x | Output | Action Output | # Time Action | Message

#	Time	Action	Message
5	10:18:02	CREATE OR REPLACE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, ROUND...	0 row(s) affected
6	10:18:31	SELECT * FROM VistaMarketing	101 row(s) returned

L'ordre per crear la vista es CREATE VIEW, però primerament, he oblidat d'ordenar els resultats de manera de major a menor, segons la mitja. Afegint "OR REPLACE", hem re-escrit la vista perquè s'adeqüi a la demanda.

Exercici 3

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

```
87 •  SELECT * FROM VistaMarketing
88 WHERE country = 'germany';
RQ
```

company_name	phone	country	mitja_amount
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.867160
Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.319156
Convallis In Incorporated	06 66 57 29 50	Germany	257.745376
Ac Industries	09 34 65 40 60	Germany	255.147288
Rutrum Non Inc.	02 66 31 61 09	Germany	255.136927

VistaMarketing 7 x Read Only

Output

Action Output

#	Time	Action	Message
5028	11:16:09	SELECT * FROM VistaMarketing	101 row(s) returned
5029	11:39:50	SELECT * FROM VistaMarketing WHERE country = 'germany'	8 row(s) returned

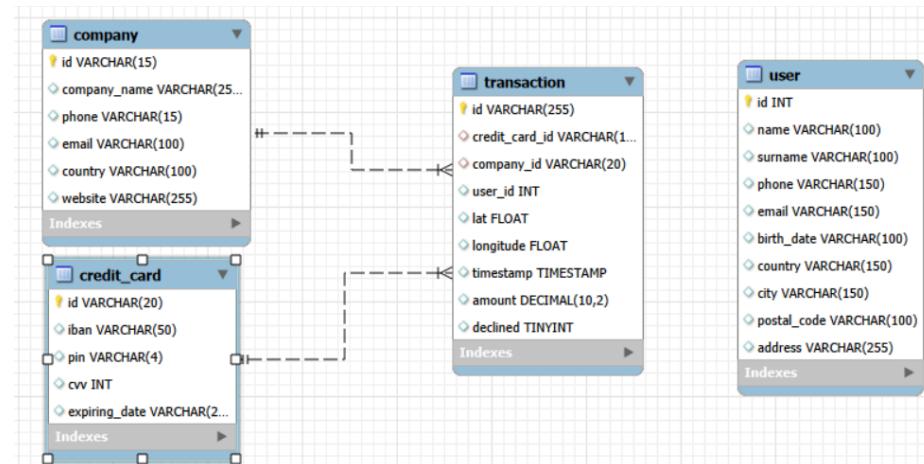
Per a fer el filtre, només hem de fer una cerca a la vista com si es tractés d'una taula, amb el filtre que ens demana a la columna "country"

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:

El primer que farem, serà crear la nova taula "user" i introduir les dades adjuntes. Un cop fet, comprovem amb ingenieria inversa quin és l'estat de la taula.



Com ja podem sospitar, la clau entre “user (id)” (PK) i transaction(user_id) (FK), no ha quedat establet i haurem d'establir la relació manualment. Per aconseguir-ho haurem de canviar el tipus de dades de les dues columnes a INT, com ens mostra el diorama. Però encara tenim problemes per fer el vincle entre les dues taules, generats per la transacció que hem introduït manualment, i llurs dades (**credit_card_id**, **company_id** i **user_id**) no tenen correspondències a les taules de dimensions. De nou, per poder fer el vincle, introduirem manualment a la taula user, l'escassa informació que tenim (id), per poder seguir endavant.

```

101 •  ALTER TABLE transaction
102     MODIFY user_id INT;
103
104 •  ALTER TABLE user
105     MODIFY id INT;
106
107 •  INSERT INTO user (id) VALUES (9999);
108
109 •  ALTER TABLE transaction
110     ADD FOREIGN KEY (user_id) REFERENCES user(id);
111

```

Output

Action Output	#	Time	Action	Message
10037 12:10:26	INSERT INTO user (id) VALUES (9999)			1 row(s) affected
10038 12:10:32	ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES user(id)			100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0

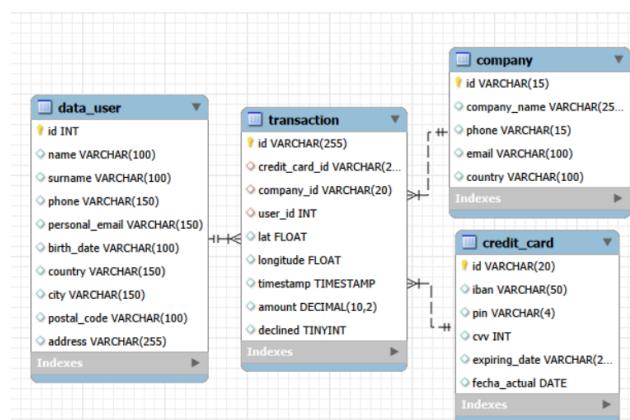
Observant l'esquema tal i com el tenim en aquest punt, prenem nota de la resta de canvis que hem de dur a terme per deixar la taula tal i com ens han demanat:

1. A la taula company, s'ha eliminat el camp “website”
2. La taula “user” ha canviat el nom a “data_user”
3. El camp “email” a canviat a “personal_email”
4. El camp “credit_card_id” te un format de VARCHAR (20) en comptes de 15
5. Hi ha un camp nou a la taula “credit_card”, “fecha_actual” DATE, que haurem de crear. Hi establirem la data actual com a valor.

```

116      -- A la taula company, s'ha eliminat el camp “website”
117 •  ALTER TABLE company DROP COLUMN website;
118      -- La taula “user” ha canviat el nom a “data_user”
119 •  RENAME TABLE user TO data_user;
120      -- El camp “email” a canviat a “personal_email”
121 •  ALTER TABLE data_user RENAME COLUMN email TO personal_email;
122      -- El camp “credit_card_id” te un format de VARCHAR (20) en comptes de 15
123 •  ALTER TABLE transaction
124     MODIFY credit_card_id VARCHAR(20);
125      -- Hi ha un camp nou a la taula “credit_card”, “fecha_actual” DATE, que haurem de crear.
126 •  ALTER TABLE credit_card
127     ADD COLUMN fecha_actual DATE DEFAULT (CURDATE());

```



Aquest és l'esquema resultant.

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 descendente en funció de la variable ID de transacció.

```
141 • CREATE OR REPLACE VIEW InformeTecnico AS
142     SELECT t.id Id_Transaccio,
143             name Nom,
144             surname Cognom,
145             iban IBAN,
146             company_name Nom_companyia,
147             timestamp Data_transaccio,
148             co.country pais,
149             amount Import,
150             s.Mitja_Companyia,
151             s.Total_Companyia
152     FROM transaction t
153     JOIN data_user du ON du.id = t.user_id
154     JOIN credit_card cc ON cc.id = t.credit_card_id
155     JOIN company co ON co.id = t.company_id
156     JOIN (
157         SELECT company_id, ROUND(AVG(amount),2) AS Mitja_Companyia, SUM(amount) AS Total_Companyia
158         FROM transaction
159         GROUP BY company_id
160     ) s ON s.company_id = t.company_id
161     ORDER BY t.id DESC;
162
163 • SELECT * FROM InformeTecnico;
```

The screenshot shows a database interface with a code editor and a results grid. The code editor contains the SQL script provided above. The results grid displays the data from the view, with columns: Id_Transaccio, Nom, Cognom, IBAN, Nom_companyia, Data_transaccio, and pais. The results show four transactions from different users, each with their details and the calculated average and total amounts for their respective companies. The results grid has a 'Result Grid' button, a 'Form Editor' button, and a 'Read Only' link. Below the results grid, there is an 'Output' section showing the log of actions taken, including the creation of the view and the execution of the select statement.

Id_Transaccio	Nom	Cognom	IBAN	Nom_companyia	Data_transaccio	pais
FFF731D6-9495-47CE-B54A-7D88E1CC274B	Bmrgli	Tprvvvmrc	XX794814451211289182490922	Turpis Company	2021-08-05 12:26:24	Netherlands
FFFCF76D-ECF0-4985-A2D0-B2A7B75998FC	Dfried	VlqqjdI	XX636251701647892036676034	Amet Nulla Donec Corporation	2023-06-17 19:10:30	Italy
FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faofvfy	XX162677143304223631437567	Nunc Interdum Incorporated	2019-10-30 02:31:37	Germany
FFFB270D-F53A-HD5D-9666-E5307C53CC84	Ggzjpa	Uirzjulh	XX395114267082019952567052	Viverra Donec Foundation	2024-05-13 03:42:03	United Kingdom
FF9E3CE-234E-408C-A8EF-F9CAD577224A	Yshimq	Zpsjsleed	XX8845462156537570367941	Convallis In Incorporated	2022-12-17 20:57:55	Germany

InformeTecnico x

Output

#	Time	Action	Message
6	10:59:24	CREATE OR REPLACE VIEW InformeTecnico AS SELECT t.id Id_Transaccio, name Nom, surname Co...	0 row(s) affected
7	10:59:27	SELECT * FROM InformeTecnico	100000 row(s) returned

Hem afegit alguna dada significativa com l'import de cada transacció, la data en que es va fer o el país de la companyia. També hem introduït la mitjana per venda de cada companyia, i el total de "amount" per companyia, a través d'una subconsulta amb JOIN.