

SPRINT 3

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.

Primer he creat una taula amb nom: *credit_card*; -si ara en Action Output surten errors perquè he preparat l'explicació un cop fet l'exercici-. La Primary Key es Id (registre únic i no es repeteix) el ser lletres, símbols i números he agafat el tipo com CHAR aquesta registre no pot ser null; Iban i pan l'he determinat el format com a VARCHAR amb diferent longitud perquè les dades no son totalment homogènies, es a dir, hi ha números i lletres, en canvi, en pin i cvv he "acotat" més les dades, posant INT perquè els valor son enters i amb la mateixa longitud. Respecte expiring_date, tot i ser una data no he determinat coma Data perquè em donava problemes els "-" i per facilitar la carga l'he considerat CHAR.

```

10      # Nivell 1
11      #Exercici 1
12  • create table credit_card (
13      id char (8) not null primary key,
14      iban varchar (100) not null,
15      pan varchar(25) not null,
16      pin int not null,
17      cvv int not null,
18      expiring_date char(10) not null
19  );

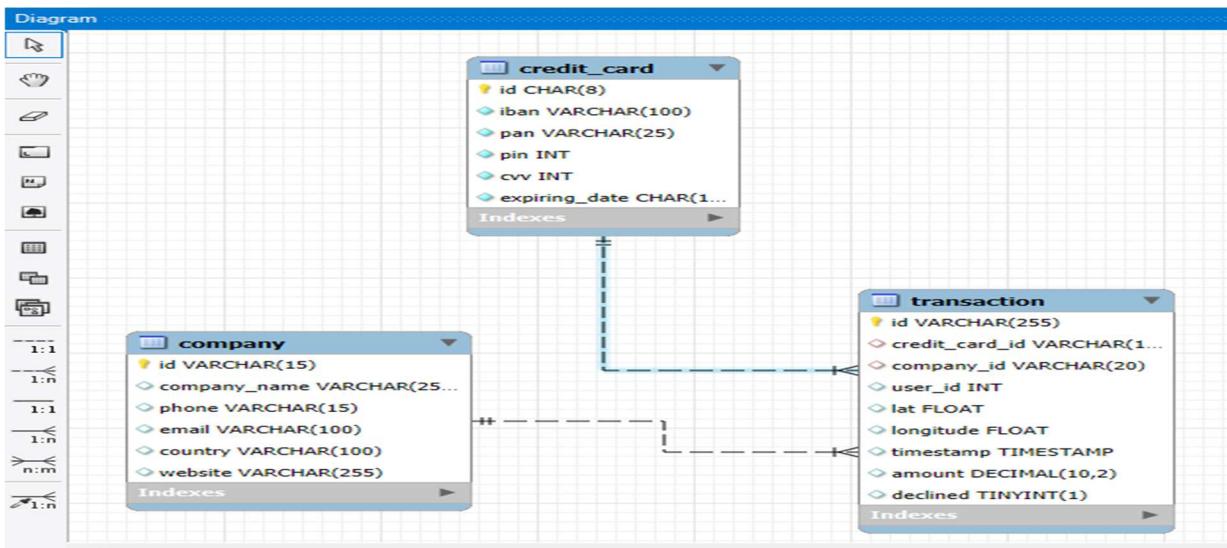
```

Action	Time	Message	Duration / Fetch
select database()	16:07:38	1 row(s) returned	0.000 sec / 0.000 sec
SELECT TABLE_SCHEMA, TABLE_NAMEcredit_card FROM INFORMATION_SCHEMA....	16:07:38	Error Code: 1054. Unknown column 'TABLE_NAMEcredit_card' in field list'	0.000 sec
select* from credit_card	16:07:44	0 row(s) returned	0.000 sec / 0.000 sec
select* from credit_card	16:08:29	5000 row(s) returned	0.000 sec / 0.000 sec
select* from credit_card where credit_card.id='CcS-4862'	16:08:57	1 row(s) returned	0.000 sec / 0.000 sec
show warnings	16:10:07	0 row(s) returned	0.000 sec / 0.000 sec
select* from credit_card	16:10:17	5000 row(s) returned	0.000 sec / 0.000 sec
alter table transaction add constraint fk_transaction_credit_card_id foreign key (credit_card....	16:17:22	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	2.250 sec
select* from credit_card	16:17:30	5000 row(s) returned	0.000 sec / 0.015 sec
create table credit_card (id char (8) not null primary key, iban varchar (100) not null, p... Error Code: 1050. Table 'credit_card' already exists	16:21:09	0.000 sec	
create table credit_card (id char (8) not null primary key, iban varchar (100) not null, p... Error Code: 1050. Table 'credit_card' already exists	16:21:13	0.000 sec	

He "carregat" les dades del document denominat "dades_introduir_credit"; després he buscat les relacions entre les taules, per tant, en la taula *transaction* he afegit un fk (*credit_card*) que esta relacionada amb la nova taula *credit_card* a través de la columna ID.

- # Afegiu una FK a la Taula Transaction. Relació PK (Credit_card.id) amb FK (transaction.Credit_card.id)
- alter table transaction
add constraint fk_transaction_credit_card_id
foreign key (credit_card_id)
references credit_card (id);

Action	Time	Message
alter table transaction add constraint fk_transaction_credit_card_id foreign key (credit_card....	17:35:57	Error Code: 1826. Duplicate foreign key constraint name 'fk_transaction_credit_card_id'



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.

Per aquest segon exercici he vist que ens demanava canvis en una dada d'un registre de la taula *Credit_card*, al haver-hi una relació entre aquesta taula i la taula *transaction* a través de la KF, s'ha de tenir en compte els canvis que facis en una taula s'actualitzi en l'altre. Per tant, com que no ho he tingut en compte inicialment, he decidit borrar/eliminar la foreign Key de la taula *transaction* i fer-ho de nou amb totes les condicions segons l'anunciat d'aquest segon exercici. He fet el següent:

```
alter table transaction
drop foreign key fk_transaction_credit_card_id;
```

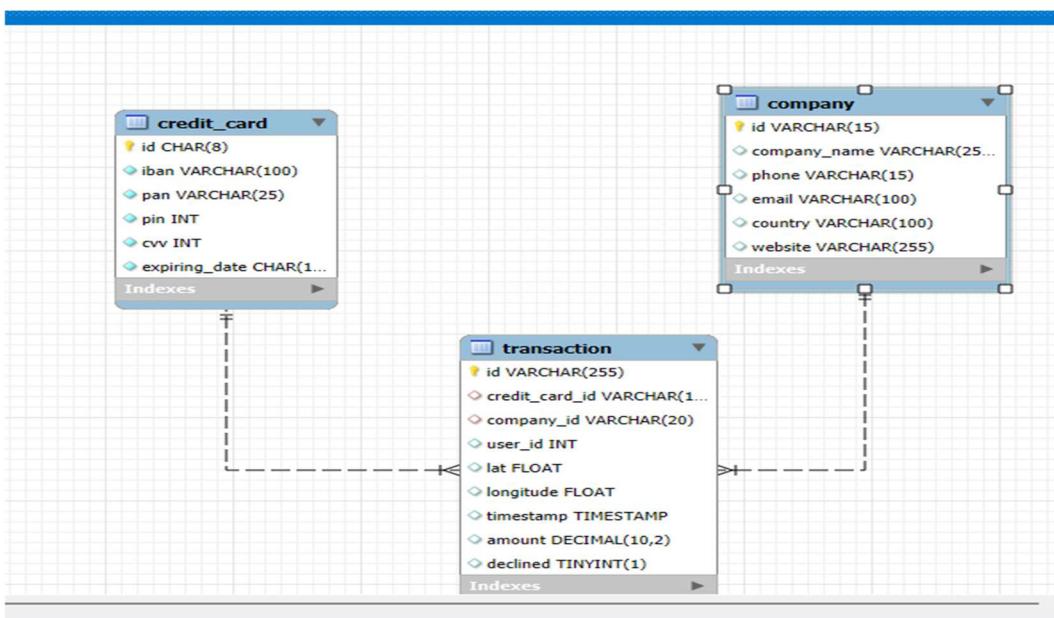
Info	Columns	Indexes	Triggers	Foreign keys	Partitions	Grants	DDL
Column	Type						
↳ id	varchar(255)						select,insert,update,references
↳ credit_card_id	varchar(15)						select,insert,update,references
↳ company_id	varchar(20)						select,insert,update,references
↳ user_id	int						select,insert,update,references
↳ lat	float						select,insert,update,references
↳ longitude	float						select,insert,update,references
↳ timestamp	timestamp						select,insert,update,references
↳ amount	decimal(10,2)						select,insert,update,references
↳ declined	tinyint(1)						select,insert,update,references

Crear-la de nou la FK amb les opcions que necessitem per l'exercici.

```
38 • alter table transaction
39   add constraint fk_transaction_credit_card_id
40     foreign key (credit_card_id)
41       references credit_card (id)
42       on update cascade # Si canvies l'id de la targeta, s'actualitza a transaction
43       on delete cascade; #Si elimines la targeta, s'elimina la transacció associada
44
45
46
47
48
49
50
51
52
53
54
55
56
57
```

Action Output			Message
#	Time	Action	
112	17:51:14	alter table transaction drop foreign key fk_transaction_credit_card_id	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

El diagrama queda igual:



Per insertar el nou registre he pensat 2 maneres, la primera – que crec que es més eficient a nivell SQL- i es:

```
33     ### Primer forma:
34 •   update credit_card
35     set iban='TR323456312213576817699999'
36     where id = 'CcU-2938';
37     ### comprovacio
38 •   select*
39     from credit_card
40     where id = 'CcU-2938';
41
42
43
44
45
```

Result Grid						
	id	iban	pin	cvv	expiring_date	fecha_actual
▶	CcU-2938	TR323456312213576817699999	3257	984	10/30/22	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

Una segona forma de fer-ho; Insertes el registre nou que ens ha indicat RRHH; com que ja hi es i volem només canviar les dades fem:

```
56 •   insert into credit_card (id, iban, pan, pin, cvv, expiring_date)
57     values ('CcU-2938', 'TR323456312213576817699999', '5424465566813633', 3257, 984, '10/30/22')
58     on duplicate key update
59     iban= values(iban),
60     pan= values(pan),
61     pin= values(pin),
62     cvv= values(cvv),
63     expiring_date= values(expiring_date);
```

El resultat es:

```

65
66 •   select*
67   from credit_card
68   where credit_card.id='CcU-2938';
69
70
71
72
73
74
75
76

```

credit_card 66 ×

Output

Action Output

#	Time	Action
129	19:04:06	select* from credit_card where credit_card.id='CcU-2938'

Message 1 row(s) returned

Duration / Fetch 0.000 sec / 0.000 sec

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

El modificar informació d'una taula ('transaction') que té vincles amb altres he anat a veure com eren aquestes relacions entre les FK i les altres taules ('company' i 'credit_card'); respecte la primera, he vist que no està "preparada" per una actualització en cas de modificar/eliminar algun registre:

DDL for transactions.transaction

```

1  CREATE TABLE `transaction` (
2    `id` varchar(255) NOT NULL,
3    `credit_card_id` varchar(15) DEFAULT NULL,
4    `company_id` varchar(20) DEFAULT NULL,
5    `user_id` int DEFAULT NULL,
6    `lat` float DEFAULT NULL,
7    `longitude` float DEFAULT NULL,
8    `timestamp` timestamp NULL DEFAULT NULL,
9    `amount` decimal(10,2) DEFAULT NULL,
10   `declined` tinyint(1) DEFAULT NULL,
11   PRIMARY KEY (`id`),
12   KEY `company_id` (`company_id`),
13   KEY `fk_transaction_credit_card_id` (`credit_card_id`),
14   CONSTRAINT `fk_transaction_credit_card_id` FOREIGN KEY (`credit_card_id`) REFERENCES `credit_card` (`id`) ON DELETE CASCADE ON UPDATE
15   CONSTRAINT `transaction_ibfk_1` FOREIGN KEY (`company_id`) REFERENCES `company` (`id`)
16 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

```

En aquest exercici, se'ns demana "ingressar un nou usuari" i això es tindria que visualitzar en la taula de "company". L'exercici no es dóna totes les dades per complamentar els camps de la taula "company"; però crec que la taula ja està preparada per admetre nous usuaris; ho veiem en l'estructura de quan es va crear la taula, perquè excepte la Pk les altres poden assignar un valor nul per defecte (DEFAULT):

DDL for transactions.company

```

1   CREATE TABLE `company` (
2     `id` varchar(15) NOT NULL,
3     `company_name` varchar(255) DEFAULT NULL,
4     `phone` varchar(15) DEFAULT NULL,
5     `email` varchar(100) DEFAULT NULL,
6     `country` varchar(100) DEFAULT NULL,
7     `website` varchar(255) DEFAULT NULL,
8     PRIMARY KEY (`id`)
9   ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

```

Per tant, aquí podríem crear un nou client a la taula 'Company' i després afegir el nou ingrés d'un usuari. En aquest cas, com que l'enunciat no ens dóna informació sobre l'empresa i la taula 'company' permet excepte Id (not Null) no "indicar" dades/camps, faré la primer opció: crear l'empresa a "company" i després afegir la transacció de l'exercici.

```

93      # Insertem el nou client a 'company'
94  •    insert into company ('id')
95    values ('b-9999');
96
97
98
99
100
101

```

Output

#	Time	Action	Message
6	12:41:03	insert into company ('id') value ('b-9999')	Error Code: 1054. Unknown column 'b-9999' in field list
7	12:43:08	insert into company ('id') values ('b-9999')	1 row(s) affected

Comprovació que i es:

```

96
97  •    select*
98    from company
99    where id= 'b-9999';

```

Result Grid

id	company_name	phone	email	country	website
b-9999	NULL	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL

company 5

Output

#	Time	Action	Message
7	12:43:08	insert into company ('id') values ('b-9999')	1 row(s) affected
8	12:45:19	select*from company where id= 'b-9999'	1 row(s) returned

Ara, tenim que tractar la relació entre el FK de taula *Transaction* i PK d'una altre taula com és *Credit_card* (*per deixar-ho tot lligat*); aquí cal tenir en compte que si fem canvis el FK no s'actualitzaran a PK; a més, segons l'enunciat de l'exercici de la transacció no tenim dades per omplir els camps de *credit_card* (iban, pan, pin, cvv...) només tenim id; per tant, he decidit modificar la taula '*Credit_card*' per permetre default Null, serà una taula més flexible i menys restriccions per poder fer el cas.

```

100
101  •    #modificar les dades de tbla company per poder fer transaccions
102  alter table `credit_card`
103    modify `iban` varchar(100) default null,
104    modify `pan` varchar(25) default null,
105    modify `pin` int default null,
106    modify `cvv` int default null,
107    modify `expiring_date` char(10) default null;
108
109
110
111
112
113
114
115
116
117
118

```

Output

#	Time	Action	Message
9	13:00:56	insert into `transaction` ('id','credit_card_id','company_id','user_id','lat','lon...)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails: 'transactions'.'credit_card_id' cannot be null
10	14:42:31	alter table `credit_card` modify `iban` varchar(100) default null, modify `pan` ...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Ara podem crear el credit_card:

```
107
108      # creo la nova targeta:
109 •  INSERT INTO `credit_card` (`id`)
110     VALUES ('CcU-9999');
```

Ara ja podem ingressar la nova transacció amb el nou client i la nova credit_card_id:

```
113      #ingressa un nou usuari amb la seguent informació
114 •  insert into `transaction` (`id`,`credit_card_id`,`company_id`,`user_id`,`lat`,`longitude`,`amount`,`declined`)
115   values ('108B1D1D-5B23-A76C-55EF-C568E49A99DD','CcU-9999','b-9999',9999,829.999,-117.999,111.11,0);
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
```

Output

#	Time	Action
13	14:49:46	INSERT INTO `credit_card` (`id`) VALUES ('CcU-9999')
14	14:52:44	insert into `transaction` (`id`, `credit_card_id`, `company_id`, `user_id`, `lat`, `longitude`, `amount`, `declined`)

Message
Error Code: 1062. Duplicate entry 'CcU-9999' for key 'credit_card.PRIMARY' 1 row(s) affected

Comprovació:

```
116      #comprovació:
117 •  select*
118   from transaction
119   where credit_card_id='CcU-9999';
```

Result Grid

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	NULL	111.11	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 7 ×

Output

#	Time	Action
17	14:56:34	select* from transaction where credit_card_id='CcU-9999'
18	14:56:36	select* from transaction where credit_card_id='CcU-9999'

Message
1 row(s) returned
1 row(s) returned

Exercici 4

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

```
120
121      #Exercici 4
122      #Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card. Recorda mostrar el canvi realitzat.
123 •  alter table credit_card
124     drop column pan;
```

Output

#	Time	Action
1	17:29:12	select* from credit_card
2	17:29:39	alter table credit_card drop column pan

Message
5001 row(s) returned
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Duration / Fetch
0.000 sec / 0.011 sec
0.032 sec

Comprovació:

```
126
127 • select*
128   from credit_card;
129
130
131
132
133
134
135
136
137
```

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

	id	iban	pin	cvv	expiring_date
▶	CcS-4857	XX485759183529250580771	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
	CcS-4862	XX4774721462463645409758	4042	174	11/27/26
	CcS-4863	XX1176920661216016207111	5050	440	12/27/26

credit_card 2 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	17:29:39	altertable credit_card drop column pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.032 sec
3	17:32:01	select* from credit_card	5001 row(s) returned	0.000 sec / 0.000 sec

Nivell 2

Exercici 1

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

The screenshot shows the MySQL Workbench interface. The top part is a script editor with the following content:

```
127 #Exercici 1
128 #Elimina de la taula transaction el registre amb ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de dades.
129 • delete from transaction
130 where id='000447FE-B650-4DCF-85DE-C7ED0EE1CAAD';
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
```

The bottom part is an "Output" pane titled "Action Output". It contains the following log entries:

#	Time	Action	Message
4	17:38:18	select * from transaction where id = '000447FE-B650-4DCF-85DE-C7ED0EE1...'	1 row(s) returned
5	17:45:20	delete from transaction where id='000447FE-B650-4DCF-85DE-C7ED0EE1...'	1 row(s) affected

Comprovació:

#	Time	Action	Message
5	17:45:20	delete from transaction where id='000447FE-B650-4DCF-85DE-C7ED0EE1...'	1 row(s) affected
6	17:46:37	select * from transaction where id = '000447FE-B650-4DCF-85DE-C7ED0EE...'	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.

Creació query:

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a code editor window displaying the following SQL query:

```
142 • select c.company_name, c.phone, c.country, avg(t.amount) as mitjana
143   from transaction t
144   join company c
145     on t.company_id=c.id
146   group by c.company_name, c.phone, c.country
147   order by mitjana desc;
```

Below the code editor is a "Result Grid" table with four columns: company_name, phone, country, and mitjana. The data is as follows:

company_name	phone	country	mitjana
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.867160
Pretium Neque Corp.	07 77 48 55 28	Australia	276.158330
Urna Convallis Associates	06 01 24 77 04	United States	274.235011
At Associates	09 56 61 10 65	New Zealand	272.214870
Metus Vitae Associates	08 25 44 40 66	Australia	270.080965
Aliquet Diam Limited	02 76 61 47 46	United States	269.599181
Nec Luctus LLC	02 14 71 75 73	Norway	268.604837
Neque Tellus Incorporated	04 43 18 34 19	Ireland	267.850372
Tortor Nunc Commodo Company	05 35 92 77 16	United States	267.836085

Below the table, the status bar shows "Result 15" and "Output". Under "Output", there is an "Action Output" section with a log entry: "# 27 11:34:16 select* from VistaMarketing order by mitjana desc" and a message "101 row(s) returned".

Creació view:

```
151
152
153 • create view VistaMarketing as
154   select c.company_name, c.phone, c.country, avg(t.amount) as mitjana
155   from transaction t
156   join company c
157     on t.company_id=c.id
158   group by c.company_name, c.phone, c.country;
159
160
```

Consulta vista:

The screenshot shows the MySQL Workbench interface again. The code editor contains the following query:

```
156 • select*
157   from VistaMarketing
158   order by mitjana desc;
```

Below the code editor is a "Result Grid" table with the same four columns: company_name, phone, country, and mitjana. The data is identical to the previous result grid:

company_name	phone	country	mitjana
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.867160
Pretium Neque Corp.	07 77 48 55 28	Australia	276.158330
Urna Convallis Associates	06 01 24 77 04	United States	274.235011
At Associates	09 56 61 10 65	New Zealand	272.214870
Metus Vitae Associates	08 25 44 40 66	Australia	270.080965
Aliquet Diam Limited	02 76 61 47 46	United States	269.599181

Below the table, the status bar shows "VistaMarketing 14" and "Output". Under "Output", there is an "Action Output" section with a log entry: "# 26 11:32:42 select c.company_name,c.phone,c.country,avg(t.amount) as mitjana from t..." and a message "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"

Comprovació:

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window with the following SQL query:

```
162 •    select*
163      from VistaMarketing
164      where country ='Germany'
165      order by mitjana desc;
166
167
168
169
170
171
```

Below the code editor is a "Result Grid" table with the following data:

	company_name	phone	country	mitjana
▶	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
	Auctor Mauris Corp.	05 62 87 14 41	Germany	254.765518
	Augue Foundation	06 88 43 15 63	Germany	253.505000
	Aliquam PC	01 45 73 52 16	Germany	253.136923

At the bottom of the interface, there is an "Output" panel showing the following log entry:

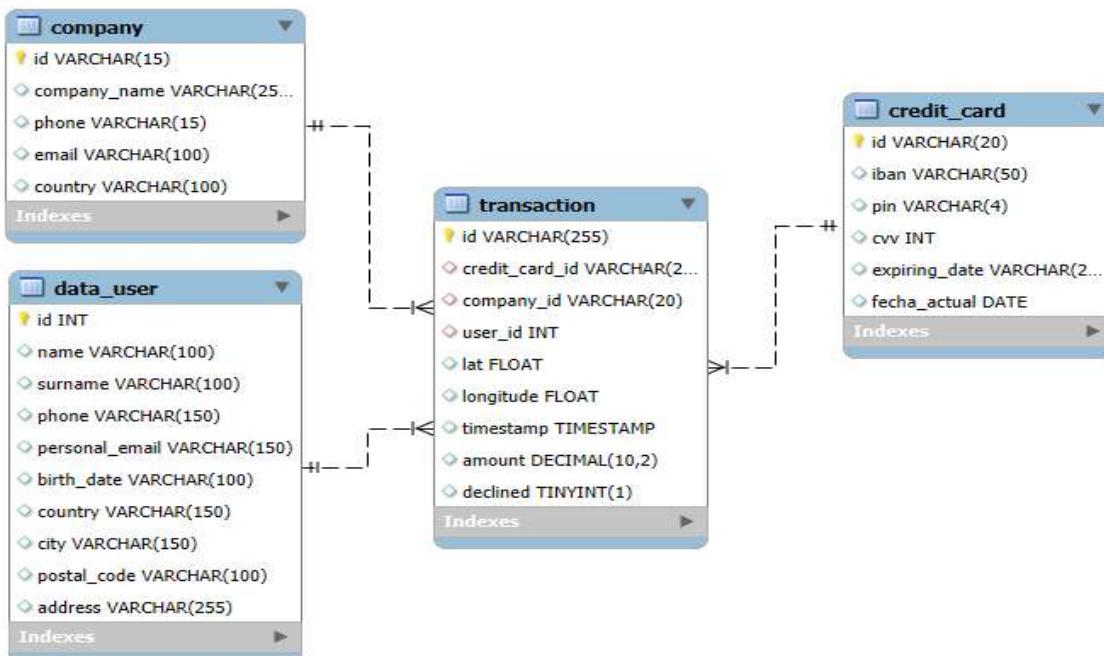
```
VistaMarketing 16 ×
Output
Action Output
# Time Action
28 11:35:20 select c.company_name,c.phone,c.country,avg(t.amount) as mitjana from t... 101 row(s) returned
```

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:

Comprovació:



Jo he partit de l'idea de crear la taula 'data_user', segons l'arxiu que ens dóna l'anunciat; he mirat com era inicialment l'arxiu i després, he anat fent les modificacions que facin falta per adaptar-lo el diagrama final. Primer he creat la nova taula 'data_user'; tenint en compte l'arxiu "estructura_dades_user":

```

160  #Nivell 3
161  #Exercici 1
162  #la setmana vinent tindràs una nova reunió amb els gerents de màrqueting. Un company del teu equip va realitzar
163  #modificacions en la base de dades, però no recorda com les va realitzar. Et demana que l'ajudis a deixar els
164  #comandos executats per a obtenir el següent diagrama:
165
166 • CREATE TABLE IF NOT EXISTS user (
167     id CHAR(10) PRIMARY KEY,
168     name VARCHAR(100),
169     surname VARCHAR(100),
170     phone VARCHAR(150),
171     email VARCHAR(150),
172     birth_date VARCHAR(100),
173     country VARCHAR(150),
174     city VARCHAR(150),
175     postal_code VARCHAR(100),
176     address VARCHAR(255)
177 );
178

```

Output ::::::::::::

Action Output			
#	Time	Action	Message
1	17:52:30	CREATE TABLE IF NOT EXISTS user (id CHAR(10) PRIMARY KEY, name ...)	0 row(s) affected

Renombrar la taula de user a data_user:

```

179 • rename table user to data_user;
180 • show tables;
181
182
183
184
185
186
187

```

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content: |

Tables_in_transactions
company
credit_card
data_user
transaction
vistamarketing

Result 2 ×

Output ::::::::::::

Action Output			
#	Time	Action	Message
3	18:12:39	rename table user to data_user	0 row(s) affected
4	18:13:01	show tables	5 row(s) returned

Modificar el tipus de la columna id:

Schemas

- tienda_online
- transactions
- Tables
 - company
 - credit_card
 - data_user**
 - Columns
 - ID
 - name
 - surname
 - phone
 - email
 - birth_date
 - country
 - city
 - postal_code

Column: ID

Definition: ID int PK

```

185  #modifiquem tipus de dades
186 • alter table data_user modify `ID` int;
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202

```

Output ::::::::::::

Action Output			
#	Time	Action	Message
7	18:29:12	alter table data_user modify `ID` int	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
8	18:29:15	alter table data_user modify `ID` int	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

Ara, vull importar les dades, però tenim arxiu amb les dades que es diu user i jo havia renombrat l'arxiu com a data_user, per tant, renombro de nou:

```
188      #renombrem com a user:
189 •    RENAME TABLE data_user TO user;
190
```

Carrego les dades -aquí ho he fet per Server-data Import-, tot i que es podria fer de forma directa i gaig comprovació:

The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window with the following SQL script:

```
198  #carguem dades, per Server...
199  #comprobem les dades
200 • select*
201  from user;
202
203
204
205
206
```

Below the code editor is a "Result Grid" table with 8 rows of data. The columns are: ID, name, surname, phone, email, birth_date, country, city, postal_code, and address. The data includes various names like Zeus, Garrett, Ciaran, Howard, Hayfa, Joel, Rafael, and Nissim, along with their respective details.

At the bottom of the interface, there is an "Output" pane showing the history of actions:

#	Time	Action	Message	Duration / Fetch
14	20:52:28	RENAME TABLE data_user TO user	0 row(s) affected	0.031 sec
15	20:54:21	select* from user	5000 row(s) returned	0.000 sec / 0.C

Canvio el nom de nou per adaptar-lo segons el diagrama:

The screenshot shows the MySQL Workbench interface with the "Schema Browser" open. In the "schemas" tree, under the "transactions" schema, the "data_user" table is highlighted with a red arrow. The "Object Info" tab at the bottom shows "No object selected".

In the main query editor area, the following SQL script is visible:

```
194  #renombrem com a data_user:
195 • RENAME TABLE user TO data_user;
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
```

Below the script is an "Output" pane showing the history of actions:

#	Time	Action	Message	Duration / Fetch
15	20:54:21	select* from user	5000 row(s) returned	
16	21:01:08	RENAME TABLE user TO data_user	0 row(s) affected	

Per tal que no em doni problemes en la carrega, en la nova taula *data_user*, afegiré la transacció que en l'exercici 3 del nivell 1 se'n ha dement fer. Afegeixo a *data_user* el *user_id* de la nova transacció.

```

210
211      # vaig afegir les dades que falten a data_user
212 •    insert into data_user ( `id` )
213     values ( '9999' );
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229

```

Output

Action Output			
#	Time	Action	Message
5	10:35:38	SELECT user_id FROM transaction WHERE user_id IS NOT NULL AND...	1 row(s) returned
6	10:36:17	insert into data_user (`id`) values ('9999')	1 row(s) affected

Ara falta fer la relació entre taules:

```

216
217 •    # afegir un constraint de fk en la taula transaction x rel. amb data_user
218 alter table transaction
219 add constraint fk_transaction_user_id
220 foreign key (user_id)
221 references data_user (id);
222
223
224
225
226
227
228
229
230
231
232
233
234
235

```

Output

Action Output			
#	Time	Action	Message
7	10:40:49	SELECT * FROM transactions.data_user	5001 row(s) returned
8	10:48:29	alter table transaction add constraint fk_transaction_user_id foreign key (user_id) references data_user (id)	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0

Ara anem a fer els canvis de tipus de totes les taules que falten per tal que quedi igual al diagrama de l'anunciat, afegim FK, adjunto columna fecha_actual.

File Edit View Query Database Server Tools Scripting Help

Navigator : hospital Query_Sakila_exam* Dades_introduir_Sprint2 Estructura_dades_Sprint2 Q_gral_Sprint2Rev Q_gral_Sprint2 Q_gral_Sprint

SCHEMAS

- practicajoin
- practicajoincompleta
- proves
- sakila
- sys
- tienda_online
- transactions**
 - Tables
 - company
 - credit_card
 - data_user**
 - Columns
 - Indexes
 - ForeignKeys
 - Triggers
- transaction

Administration Schemas

Table: credit_card

Columns:

id	varchar(20) PK
iban	varchar(50)
n	varchar(4)
cvv	int
expiring_date	varchar(20)
fecha_actual	date

Output

Action Output			
#	Time	Action	Message
11	11:36:08	Alter table 'credit_card' modify 'id' varchar(20), modify 'iban' varchar(50), mo...	5001 row(s) affected Records: 5001 Duplicates: 0 Warnings: 0
12	11:36:15	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_credit_card_id...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0

Seguim amb l'adaptació del tipus de columnes de les taules:

Elimino columna website.

```
hospital Query_Sakila_exam* Dades_introduir_Sprint2 Estructura_dades_Sprint2 Q_gral_Sprint2Rev Q_gral_Sprint2
242 # eliminar columna website de tabla company
243 • alter table company
244     drop column website;
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
```

Output:

Action Output	#	Time	Action	Message
	15	11:47:29	SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE T...	0 row(s) returned
	16	11:48:53	alter table company drop column website	0 rows(s) affected Records: 0 Duplicates: 0 Warnings: 0

Canvio el nom de la columna email per personal_email:

Navigator

SCHEMAS

- practicajoin
- practicajoincompleta
- proves
- sakila
- sys
- tienda_online
- transactions**
- Tables
 - company
 - credit_card
 - data_user**
- Columns
- Indexes
- Foreign Keys
- Triggers
- transaction

Administration Schemas

Information

Columns:

	id	int PK
	name	varchar(100)
	surname	varchar(100)
	phone	varchar(150)
	personal_email	varchar(150)
	birth_date	varchar(100)
	country	varchar(150)
	city	varchar(150)
	postal_code	varchar(100)
	address	varchar(255)

Object Info Session

```
hospital Query_Sakila_exam* Dades_introduir_Sprint2 Estructura_dades_Sprint2 Q_gral_Sprint2Rev Q_gral_Sprint2
245 #canviar el nom de data_user de columna email per personal_email
246 • alter table data_user
247     change email personal_email varchar(150);
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
```

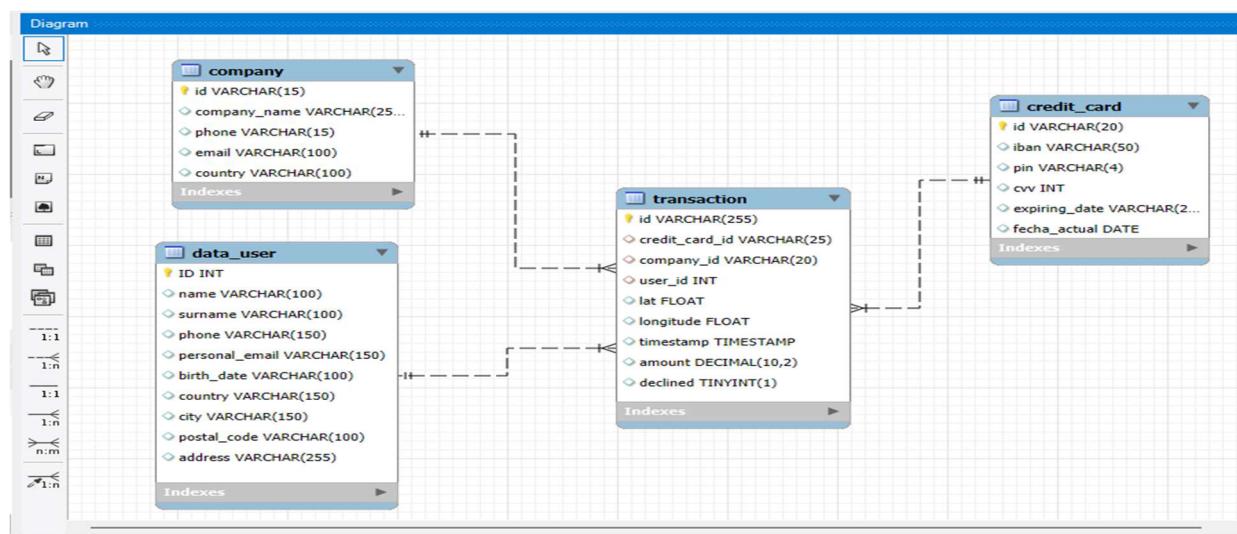
Output:

Action Output	#	Time	Action	Message
	16	11:48:53	altertable company drop column website	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
	17	11:58:00	alter table data_user change email personal_email varchar(150)	0 rows(s) affected Records: 0 Duplicates: 0 Warnings: 0

Canvio el tipus de columna Credit_car_id de taula transaction:

```
# canvi de tipus en la taula transaction// estava varchar 15 o poso a 20
• Alter table `transaction`
  modify`credit_card_id` varchar (25);
```

Resultat final:



Recordatori

En aquesta activitat, és necessari que descriguis el "pas a pas" de les tasques realitzades. És important realitzar descripcions senzilles, simples i fàcils de comprendre. Per a realitzar aquesta activitat hauràs de treballar amb els arxius denominats "estructura_dades_user" i "dades_introduir_user"

Recorda continuar treballant sobre el model i les taules amb les quals ja has treballat fins ara.

Exercici 2

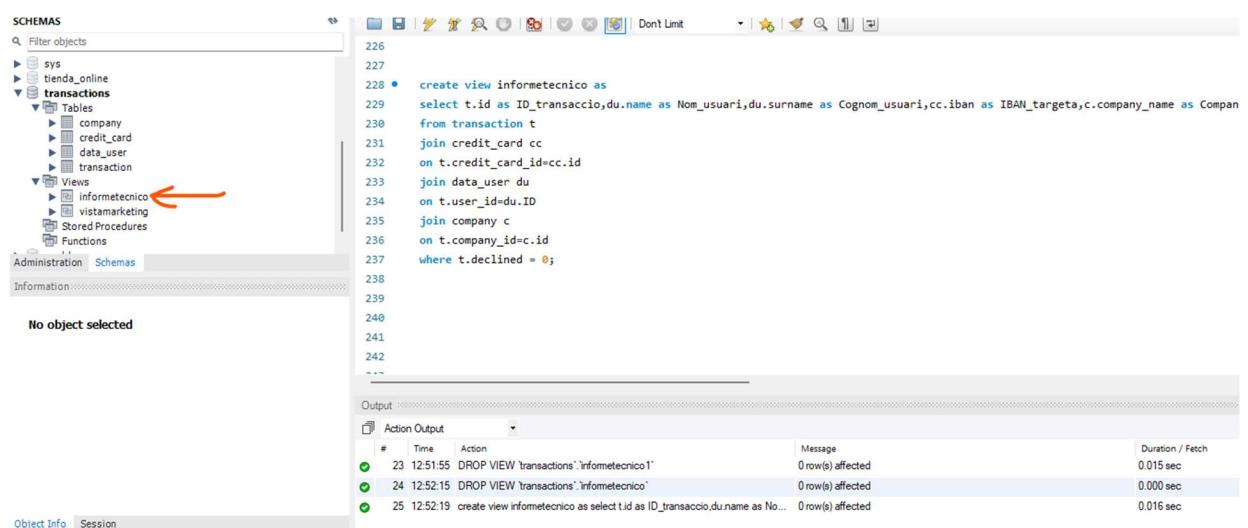
L'empresa també us demana crear una vista anomenada "InformeTecnico" que contingui la següent informació:

Comprovació:

- **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.

Creació de la join per la preparació de la view:



```
226
227
228 •  create view informetecnico as
229   select t.id as ID_transaccio,du.name as Nom_usuari,du.surname as Cognom_usuari,cc.iban as IBAN_targeta,c.company_name as Compan
230   from transaction t
231   join credit_card cc
232   on t.credit_card_id=cc.id
233   join data_user du
234   on t.user_id=du.ID
235   join company c
236   on t.company_id=c.id
237   where t.declined = 0;
238
239
240
241
242
243
```

#	Time	Action	Message	Duration / Fetch
23	12:51:55	DROP VIEW `transactions`.`informetecnico`	0 row(s) affected	0.015 sec
24	12:52:15	DROP VIEW `transactions`.`informetecnico`	0 row(s) affected	0.000 sec
25	12:52:19	create view informetecnico as select t.id as ID_transaccio,du.name as No...	0 row(s) affected	0.016 sec

Resultat amb la view "informetecnico":

```

238
239
240 • SELECT *
241   FROM informetecnico
242   ORDER BY ID_transaccio DESC;
243
244
245
246
247
248

```

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

ID_transaccio	Nom_usuari	Cognom_usuari	IBAN_targeta	Companyia_trans	Import
FFFD31D6-9495-47CE-B54A-7DB8E1CC274B	Bmrgli	Tprvrmrc	XX794814451211289182490922	Turpis Company	74.54
FFFCF76D-ECF0-4985-A2D0-B2A7B75998FC	Dfried	Vlqqjdl	XX636251701647892036676034	Amet Nulla Donec Corporation	148.91
FFFC9E8D-27C7-4ADE-98F2-7533EF4DF126	Securp	Faofvqfy	XX162677143304223631437567	Nunc Interdum Incorporated	234.22
FFFB270D-F53A-4D5D-9666-E5307C53CC84	Ggzjpa	Uirzjulh	XX395114267082019952567052	Viverra Donec Foundation	349.13
FFF9E3CE-234E-408C-A8EF-F9CAD577224A	Yshimq	Zpsjsleed	XX8845462156537570367941	Convallis In Incorporated	247.39
FFF9E178-6CD2-4DF9-99B0-49AE068809B1	Jevedx	Xwczwnm	XX321405515711654384711481	Mus Aenean Eget Foundation	438.13

informetecnico 4 ×

Output

#	Time	Action	Message
27	12:54:30	SELECT * FROM informetecnico ORDER BY ID_transaccio DESC	99763 row(s) returned
28	12:55:33	SELECT * FROM informetecnico ORDER BY ID_transaccio DESC	99763 row(s) returned