

Tasca S3.01. Manipulació de taules

NIVELL 1

N1EX_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.

Descripció dels camps de la taula que crearem, descrivim les variables i la seva longitud. variables.

Id: VARCHAR(20): ens ajuda a emmagatzemar l'identificador de la targeta, en aquest cas es un identificador alfanumèric, considerat con una cadena de text.

iban: VARCHAR(34): es una seqüència llarga de números i lletres (alfanumèrica) que a l'igual que id, es considera cadena de text, perquè no farem operacions matemàtiques.

pan VARCHAR(19): El PAN (Primary Account Number) és un número de targeta, habitualment de 16 dígitos però pot arribar fins a 19. Tot i ser numèric, s'emmagatzema com a text per evitar problemes de format, ja que no s'hi faran càlculs.

pin VARCHAR(4): El PIN és un número de 4 dígitos. S'utilitza el tipus cadena de text perquè no es fan operacions matemàtiques i pot contenir zeros a l'esquerra.

cvv VARCHAR(3): representa el codi de seguretat de la targeta i consisteix en un número de 3 dígitos. Encara que podria utilitzar-se el tipus INT, habitualment es defineix com a cadena de text per preservar possibles zeros a l'esquerra i evitar problemes de format.

Expiring_date DATE: per emmagatzemar la data en format 'YYYY-MM-DD'.

El codi crea una taula de targetes, transforma correctament el format de data, assegura la qualitat de les dades i finalment estableix una relació sòlida entre targetes i transaccions.

Creació de la taula credit_card CREATE TABLE, una millora i bona pràctica seria fer la taula CRATE TABLE IF EXISTS, per evitar errors si la taula ja existís, primer es desa expiring_date com a text i després es converteix a format DATE amb UPDATE i STR_TO_DATE (%m/%d/%y). Això assegura la coherència de les dades i permet

operar amb dates. Finalment, s'ajusta la columna amb ALTER TABLE MODIFY COLUMN i es verifica amb DESCRIBE.

S'estableix la relació a la taula transaction mitjançant la incorporació d'una clau forana (FOREIGN KEY) que vincula les transaccions amb la taula credit_card a través de l'atribut credit_card_id referenciant credit_card.id. Aquesta mesura garanteix la integritat referencial, assegurant que cap transacció pugui estar associada a una targeta de crèdit inexistent.

```

1  -- SPRINT_3 --
2  -- NIVELL 1 --
3  /* Exercici 1 : La teva tasca és dissenyar i crear una taula anomenada "credit_card" que emmagatzemi detalls
4  crucials sobre les targetes de crèdit. La nova taula ha de ser capaç d'identificar de manera única cada targeta
5  i establir una relació adequada amb les altres dues taules ("transaction" i "company"). Després de crear la taula
6  serà necessari que ingressis la informació del document denominat "dades_introduir_credit". Recorda mostrar el diagrama i
7  realitzar una breu descripció d'aquest.*/
8  • USE transactions;
9  • CREATE TABLE credit_card (
10     id VARCHAR(20) PRIMARY KEY,
11     iban VARCHAR(34),
12     pan VARCHAR(19),
13     pin VARCHAR(4),
14     cvv VARCHAR(3),
15     expiring_date VARCHAR(255) # Es fa servir cadena de text per emmagatzemar les dates, i convertir-les posteriorment.
16 );
17
18 • UPDATE credit_card
19 SET expiring_date = STR_TO_DATE(`expiring_date`, '%m/%d/%y')
20 WHERE id IS NOT NULL;
21

```

Output

#	Time	Action	Message	Duration / Fetch
20	17:24:00	UPDATE credit_card SET expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%y') ...	5000 row(s) affected Rows matched: 5000 Changed: 5000 Warnings: 0	0.156 sec
21	17:24:07	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.000 sec
22	17:24:07	SELECT * FROM credit_card	5000 row(s) returned	0.000 sec / 0.000 sec
23	17:27:04	SET SQL_SAFE_UPDATES = 1	0 row(s) affected	0.000 sec

```

1/
18 • UPDATE credit_card
19 SET expiring_date = STR_TO_DATE(`expiring_date`, '%m/%d/%y')
20 WHERE id IS NOT NULL;
21
22 • ALTER TABLE credit_card MODIFY COLUMN expiring_date DATE;
23 • DESCRIBE credit_card;
24

```

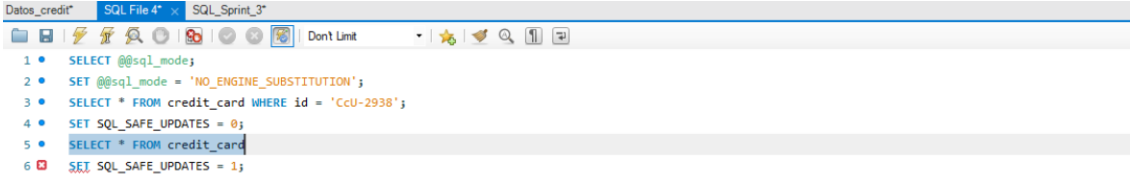
Result Grid

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI		
iban	varchar(34)	YES			
pin	varchar(4)	YES			
cvv	varchar(3)	YES			
expiring_date	date	YES			

Result 1 x

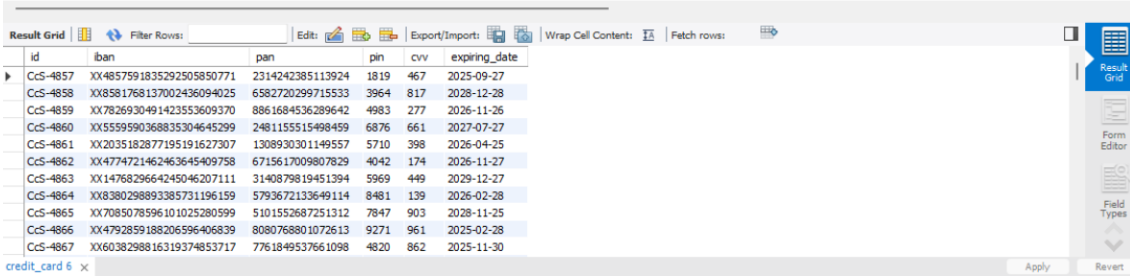
Output

#	Time	Action	Message	Duration / Fetch
1	18:34:21	ALTER TABLE credit_card MODIFY COLUMN expiring_date DATE	5001 row(s) affected Records: 5001 Duplicates: 0 Warnings: 0	0.172 sec
2	18:38:48	DESCRIBE credit_card	5 row(s) returned	0.016 sec / 0.000 sec
3	18:39:10	DESCRIBE credit_card	5 row(s) returned	0.000 sec / 0.000 sec

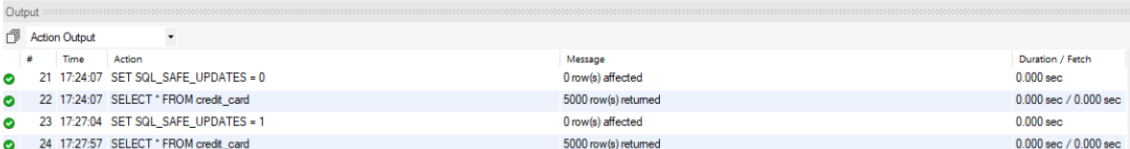


```

1 • SELECT @@sql_mode;
2 • SET @@sql_mode = 'NO_ENGINE_SUBSTITUTION';
3 • SELECT * FROM credit_card WHERE id = 'CCU-2938';
4 • SET SQL_SAFE_UPDATES = 0;
5 • SELECT * FROM credit_card;
6 • SET SQL_SAFE_UPDATES = 1;
  
```

id	iban	pan	pin	cvv	expiring_date
CcS-4857	XX4857591835292505850771	2314242385113924	1819	467	2025-09-27
CcS-4858	XX8581768137002436094025	6582720299715533	3964	817	2028-12-28
CcS-4859	XX7826930491423553609370	8861684536289642	4983	277	2026-11-26
CcS-4860	XX5559590368835304645299	2481155515498459	6876	661	2027-07-27
CcS-4861	XX2035182877195191627307	1308930301149557	5710	398	2026-04-25
CcS-4862	XX4774721462463645409758	6715617009807829	4042	174	2026-11-27
CcS-4863	XX1476829664245046207111	3140879819451394	5969	449	2029-12-27
CcS-4864	XX8380298893385731196159	5793672133649114	8481	139	2026-02-28
CcS-4865	XX7085078596101025280599	5101552687251312	7847	903	2028-11-25
CcS-4866	XX4792859188206596406839	8080768801072613	9271	961	2025-02-28
CcS-4867	XX6038298816319374853717	7761849537661098	4820	862	2025-11-30



#	Time	Action	Message	Duration / Fetch
21	17:24:07	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.000 sec
22	17:24:07	SELECT * FROM credit_card	5000 row(s) returned	0.000 sec / 0.000 sec
23	17:27:04	SET SQL_SAFE_UPDATES = 1	0 row(s) affected	0.000 sec
24	17:27:57	SELECT * FROM credit_card	5000 row(s) returned	0.000 sec / 0.000 sec

El resultat és una taula amb un total de **5 000 registres**

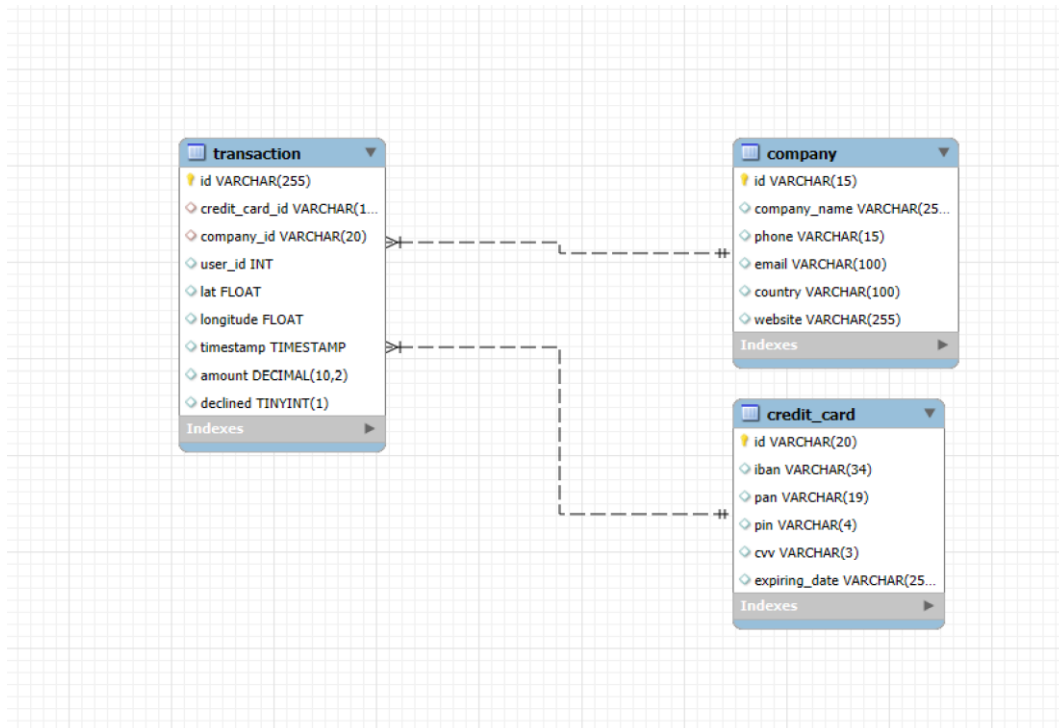
id	iban	pan	pin	cvv	expiring_date
CcS-4857	XX4857591835292505850771	2314242385113924	1819	467	2025-09-27

on **expiring_date**, esta en el format **DATE**,

Nota:

Tot i que la columna expiring_date és VARCHAR, MySQL intenta validar els seus valors com a dates si sql_mode inclou STRICT_TRANS_TABLES. Per evitar-ho temporalment, utilitza SET @@sql_mode = 'NO_ENGINE_SUBSTITUTION'; així MySQL no aplicarà la validació de dates per defecte. Un cop fet això, pots convertir la dada a DATE amb una sentència UPDATE. Si apareix l'error 1175 ("Safe Update mode"), desactiva'l provisionalment amb SET SQL_SAFE_UPDATES = 0; executa l'actualització i torna a activar el mode segur després.

Descripció del diagrama relacional:



El diagrama següent presenta l'estructura relacional del sistema de gestió de transaccions amb targeta de crèdit, que integra les dades relatives a les targetes, les transaccions efectuades i les empreses on s'han realitzat aquestes operacions.

Està compost per tres taules: credit_card, transaction i company.

1.- Taula crèdit card: compte les dades sensibles de les targetes de crèdit utilitzades en el sistema.

Atributs (ID, IBAN, PAN, PIN, CVV, data d'expiració (expiring_date)).

Clau primaria: id (VARCHAR(20)).

2.- Taula Transaction: registra cadascuna de les transaccions realitzades amb una targeta de crèdit.

Atributs (USER ID, LAT, LONGITUDE, TIMESTAMPS, AMOUNT, DECLINED)

*DECLINED (Estat indicant si va ser aprovada o no).

Clau primaria: id (VARCHAR(255))

Compte **dos claus foranes:**

- credit_card_id → Referència a credit_card.id
- company_id → Referència a company.id

Això permet establir relacions **u a molts (1:N):**

- Una targeta pot tenir moltes transaccions.

- Una empresa pot rebre moltes transaccions.

3.- Taula company: Emmagatzemar les dades de les empreses que realitzen les transaccions.

Atributs (ID, nom de la empresa, telèfon, email, país i lloc web)

Clau primaria: id (VARCHAR(15)).

N1EX_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.

L'objectiu es corregir l'IBAN de la targeta de crèdit ID = CcU-2938.

Per modificar dades en una taula, s'utilitza UPDATE. En aquest cas, cal actualitzar el camp IBAN i aplicar un filtre WHERE per implicar només la targeta amb l'identificador especificat.

```
-- 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.*/  
  
UPDATE credit_card  
SET iban = 'TR323456312213576817699999'  
WHERE id = 'CcU-2938';
```

S'ha detectat i corregit un error al camp IBAN de la targeta de crèdit amb ID CcU-2938, segons notificació del Departament de Recursos Humans.

#	Time	Action	Message	Duration / Fetch
29	18:14:44	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_credit_card FOREIGN KEY ...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	1.812 sec
30	18:39:03	UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE id = 'CcU-2938'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
31	18:39:03	UPDATE credit_card SET iban = 'TR323456312213576817699999' WHERE id = 'CcU-2938'	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.000 sec
32	18:41:28	SELECT iban FROM credit_card WHERE id = 'CcU-2938'	1 row(s) returned	0.000 sec / 0.000 sec

Com a resultat: el número de compte ha estat actualitzat a TR323456312213576817699999.

N1EX_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

Per registrar una nova transacció, primer cal crear els registres corresponents a les taules company i credit_card, ja que transaction depèn d'aquestes per claus foranes. Es pot inserir una nova empresa i una targeta només amb l'id, deixant la resta de camps com a valors nuls o perduts.

Després d'aquesta acció, la transacció s'insereix a la taula transactions amb declined = 0, indicant que s'ha aprovat.

Finalment es fa servir un SELECT per verificar que la nova transacció ha sigut insertada correctament en la base de dades.

```
43 amount = 111.11
44 declined = 0 %/
45 # Primer crear la empresa en la taula company
46 • INSERT INTO company (
47     id,
48     company_name,
49     phone,
50     email,
51     country,
52     website
53 ) VALUES (
54     'b-9999',
55     NULL,
56     NULL,
57     NULL,
58     NULL,
59     NULL
60 );
61 • INSERT INTO credit_card (
62     id,
63     iban,
64     pan,
65     pin,
66     cvv,
67     expiring_date
68 ) VALUES (
69     'CcU-9999',
70     NULL,
71     NULL,
72     NULL,
73     NULL,
74     NULL
75 );
```

```
77 • INSERT INTO transaction (
78     id,
79     credit_card_id,
80     company_id,
81     user_id,
82     lat,
83     longitude,
84     timestamp,
85     amount,
86     declined
87 ) VALUES (
88     '10881D1D-5B23-A76C-55EF-C568E49A99DD',
89     'CcU-9999',
90     'b-9999',
91     9999,
92     829.999,
93     -117.999,
94     CURRENT_TIMESTAMP, # Es guarda l'hora actual
95     111.11,
96     0
97 );
98 • SELECT * FROM transaction WHERE id = '10881D1D-5B23-A76C-55EF-C568E49A99DD'; #Comprobació de la creació de la línia
```

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
10881D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	2025-09-22 13:05:46	111.11	0

Com a resultat s'ha creat el nou registre en a la taula transaction amb la informació proporcionada.

N1EX_4: Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card. Recorda mostrar el canvi realitzat

Per modificar l'estructura de la taula, on la taula ja té una estructura, es fa servir ALTER TABLE.

Seguit es fa servir DROP COLUMN, ja que es vol eliminar una columna específica de la taula credi_card.

S'ha d'estar segur en l'execució d'aquest tipus de sentència, pel fet que el codi eliminarà tota la columna i les dades es perdran.

Es pot observar en el Result Grid que al cridar la taula credit_card, la columna pan no existeix. **Com a resultat**, s'ha eliminat el camp PAN

The screenshot shows a SQL IDE interface. The top pane displays a SQL script with the following content:

```
94 CURRENT_TIMESTAMP, # Es guarda l'hora actual
95 111.11,
96 0
97 );
98 SELECT * FROM transaction WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD'; #Comprobació de la creació de la línia
99
100 -- Exercici 4--
101 -- Des de recursos humans et sol·liciten eliminar la columna "pan" de la taula credit_card. Recorda mostrar el canvi realitzat. --
102
103 ALTER TABLE credit_card
104 DROP COLUMN pan;
105 SELECT * FROM credit_card;
```

The bottom pane shows the 'Result Grid' with the following data:

id	iban	pin	cvv	expiring_date
CcS-4857	XX4857591835292505850771	1819	467	2025-09-27
CcS-4858	XX8581768137002436094025	3964	817	2028-12-28
CcS-4859	XX7826930491423553609370	4983	277	2026-11-26
CcS-4860	XX5559590368835304645299	6876	661	2027-07-27
CcS-4861	XX2035182877195191627307	5710	398	2026-04-25

Below the Result Grid, the 'Action Output' pane shows the following message:

#	Time	Action	Message	Duration / Fetch
1	13:38:51	SELECT * FROM credit_card	5001 row(s) returned	0.000 sec / 0.015 sec

NIVELL_2

N2EX_1: Elimina de la taula transaction el registre amb ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de dades.

La verificació abans d'eliminar el registre ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD es realitza mitjançant una consulta SELECT. Posteriorment, s'executa la instrucció DELETE FROM transaction WHERE id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD', per suprimir aquest registre específic de la base de dades. Finalment, es torna a llançar una consulta SELECT amb el mateix identificador per comprovar que ja no apareix als resultats. Aquest procediment permet validar que l'eliminació s'ha dut a terme correctament i sense impactar altres registres.

The first screenshot shows a SQL script with a comment in Catalan: '-- Elimina de la taula transaction el registre amb ID 000447FE-B650-4DCF-85DE-C7ED0EE1CAAD de la base de dades.--'. The script contains a SELECT statement to verify the record's existence. The 'Result Grid' shows one record with ID '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD'. The 'Output' pane shows the execution of the SELECT statement, returning 1 row.

The second screenshot shows the same script with an additional DELETE statement. The 'Output' pane shows three actions: a SELECT statement returning 1 row, a DELETE statement affecting 1 row, and another SELECT statement returning 0 rows, confirming the record's deletion.

Com a resultat ara tenim la eliminació del registre id = '000447FE-B650-4DCF-85DE-C7ED0EE1CAAD'

N2EX_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 de la vista es farà amb la sentència SQL CREATE VIEW nom_de_la_vista AS, la qual es una taula virtual basada en una consulta fem servir SELECT, no te la capacitat de emmagatzemat les dades per si mateixa.

En aquest cas, ens demana que s'inclouï: company_name, phone, country que es trobem en la taula company. I després per fer el càlcul de la mitjana de les compres es calcularà amb la funció AVG() d'amount que es troba a la taula transaction.

Necessitem FROM i JOIN en aquest cas es fa servir un INNER JOIN per fer servir només les transaccions que tenen una coincidència en les dues taules.

Per agrupar GRUOP BY company_name, phone, country per fer el càlcul del promig de les compres de cada companyia.

I per últim, es demanen ordenar de major a menor segons el resultat del promig de vendes i es fa servir ORDER BY.

```

113
114 -- EXERCICI 2 --
115 /* La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives.
116 S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions.
117 Serà necessària que creïs una vista anomenada VistaMarketing que contingui la següent informació: Nom de la companyia,
118 Telèfon de contacte, País de residència, Mitjana de compra realitzat per cada companyia. */
119
120 • CREATE VIEW VistaMarketing AS
121 SELECT
122     c.company_name AS Entitat,
123     c.phone AS Telèfon,
124     c.country AS País,
125     ROUND(AVG(t.amount),2) AS Mitjana_Compra
126 FROM
127     company c
128 INNER JOIN
129     transaction t ON c.id = t.company_id
130 GROUP BY
131     c.company_name, c.phone, c.country
132 ORDER BY
133     Mitjana_Compra DESC;
134
135 • SELECT * FROM VistaMarketing;

```

Output

#	Time	Action	Message	Duration / Fetch
1	19:16:11	CREATE VIEW VistaMarketing AS SELECT	c.company_name AS Entitat, c.phone AS T... 0 row(s) affected	0.000 sec

Mitjançant aquesta vista, el departament de màrqueting disposarà de les dades necessàries per analitzar les companyies i les seves transaccions, ordenades segons la mitjana de compra.

SELECT * FROM VistaMarketing;

```

132 UNLOCK DT
133 Mitjana_Compra DESC;
134
135 • SELECT * FROM VistaMarketing;
136

```

Result Grid

Entitat	Telèfon	País	Mitjana_Compra
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
Prebium Neque Corp.	07 77 48 55 28	Australia	276.16
Uma Convalis 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
Tortor Nunc Commodo Company	05 35 92 77 16	United States	267.84
Cras Consulting	07 50 10 85 63	Belgium	267.44
Eget Ipsum Ltd	03 67 44 56 72	United States	267.06
Sed LLC	01 63 16 26 52	Belgium	266.07
Amet Institute	06 33 40 21 33	Australia	265.54
Pede Ultrices Ltd	04 52 40 26 17	Belgium	265.36

Output

#	Time	Action	Message	Duration / Fetch
1	19:16:11	CREATE VIEW VistaMarketing AS SELECT	c.company_name AS Entitat, c.phone AS T... 0 row(s) affected	0.000 sec
2	19:16:54	SELECT * FROM VistaMarketing	101 row(s) returned	0.359 sec / 0.000 sec

Com a resultat, s'ha generat una taula que recull el detall de company_name, telèfon, país i la mitjana de compra, tal com sol·licitava el departament de màrqueting.

N2EX_3: Filtra la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"

Per filtrar la vista VistaMarketing de manera que només es mostrin les companyies que tenen el seu país de residència a "**Germany**", simplement s'ha de modificar la consulta per afegir una condició WHERE per filtrar pel país.

```
136
137 -- EXERCICI 3 --
138 * SELECT *
139 FROM VistaMarketing
140 WHERE País = 'Germany';
141
```

The screenshot shows a database management tool interface. At the top, a query is entered in a text editor. Below it, a 'Result Grid' displays the results of the query. The grid has four columns: 'Entitat', 'Telèfon', 'País', and 'Mitjana_Compra'. It lists eight companies, all from Germany, sorted by their average purchase value in descending order. At the bottom, an 'Output' window shows the execution log, including the creation of the view and the execution of the query, which returned 8 rows.

Entitat	Telèfon	País	Mitjana_Compra
Ac Fermentum Incorporated	06 85 56 52 33	Germany	284.87
Nunc Interdum Incorporated	05 18 15 48 13	Germany	259.32
Convallis In Incorporated	06 66 57 29 50	Germany	257.75
Ac Industries	09 34 65 40 60	Germany	255.15
Rutrum Non Inc.	02 66 31 61 09	Germany	255.14
Auctor Mauris Corp.	05 62 87 14 41	Germany	254.77
Augue Foundation	06 88 43 15 63	Germany	253.51
Aliquam PC	01 45 73 52 16	Germany	253.14

VistaMarketing 5 x

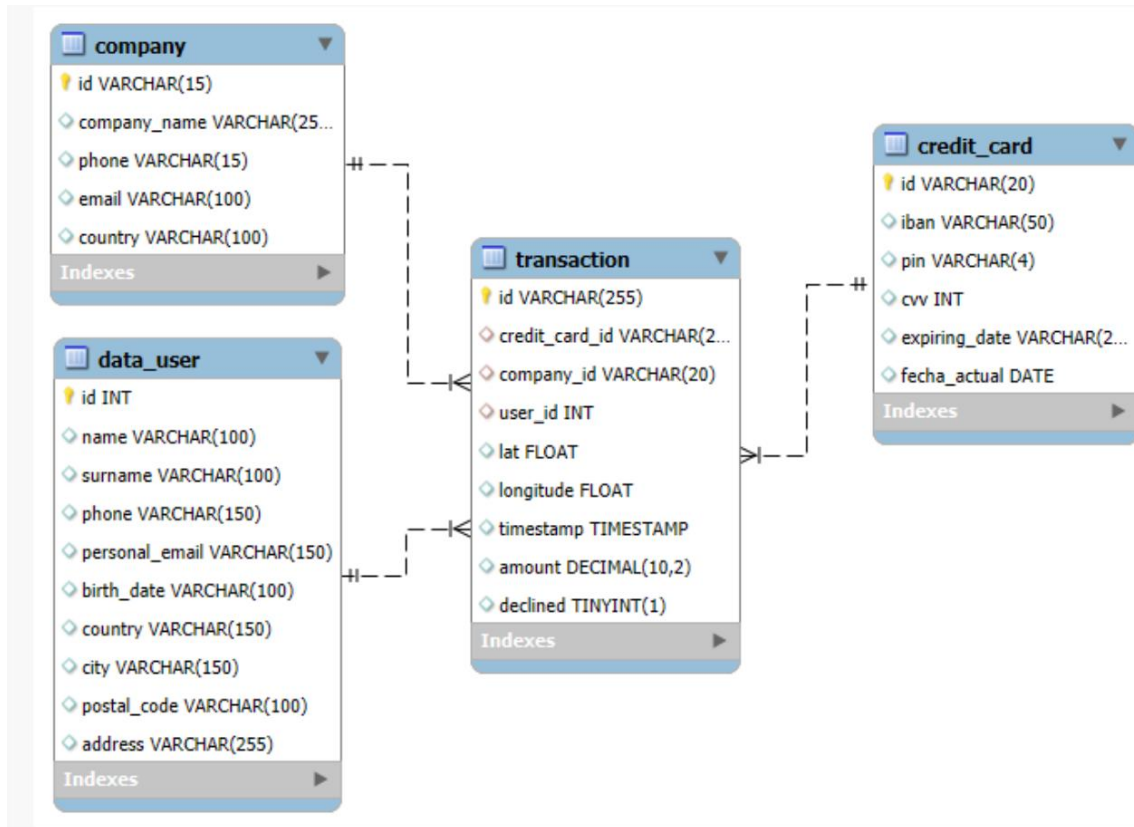
Output

#	Time	Action	Message	Duration / Fetch
1	19:16:11	CREATE VIEW VistaMarketing AS SELECT c.company_name AS Entitat, c.phone AS ...	0 row(s) affected	0.000 sec
2	19:16:54	SELECT * FROM VistaMarketing	101 row(s) returned	0.359 sec / 0.000 sec
3	19:17:50	SELECT * FROM VistaMarketing WHERE País = 'Germany'	8 row(s) returned	0.062 sec / 0.000 sec

Com a resultat, aquesta consulta retorna les **8 companyies d'Alemanya** amb la mitjana de compra més alta, ordenades de major a menor.

NIVELL_3

N3Ex_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:



La dificultat d'aquest exercici és que el camp id de la taula USER i user_id de la taula transaction tenen definicions incompatibles; per resoldre-ho, s'ha utilitzat ALTER TABLE user MODIFY id INT NOT NULL per igualar-los. Un cop resolt, **com a resultat s'ha creat la taula user**, després de crear el registre que té el user_id = 9999.

```
142 -- NIVELL 3 --
143 -- EXERCICI 1 --
144
145 CREATE TABLE IF NOT EXISTS user (
146     id CHAR(10) PRIMARY KEY,
147     name VARCHAR(100),
148     surname VARCHAR(100),
149     phone VARCHAR(150),
150     email VARCHAR(150),
151     birth_date VARCHAR(100),
152     country VARCHAR(150),
153     city VARCHAR(150),
154     postal_code VARCHAR(100),
155     address VARCHAR(255)
156 );
```

Output

#	Time	Action	Message	Duration / Fetch
1	10:20:09	CREATE TABLE IF NOT EXISTS user (id CHAR(10) PRIMARY KEY, name VARCHAR(100), ...	0 row(s) affected	0.031 sec

```
160 • INSERT INTO user (
161     id,
162     name,
163     surname,
164     phone,
165     email,
166     birth_date,
167     country,
168     city,
169     postal_code,
170     address
171 ) VALUES (
172     '9999',
173     NULL,
174     NULL,
175     NULL,
176     NULL,
177     NULL,
178     NULL,
179     NULL,
180     NULL,
181     NULL
182 );
```

```
183
184
185 • ALTER TABLE transaction
186     ADD CONSTRAINT fk_transaction_user FOREIGN KEY (user_id) REFERENCES user(id);
187
188
```

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
3	13:29:58	SELECT * FROM transaction	100000 row(s) returned	0.000 sec / 0.218 sec
4	13:30:31	INSERT INTO user (id, name, surname, phone, email, birth_date, country...	1 row(s) affected	0.016 sec
5	13:30:46	ALTER TABLE transaction ADD CONSTRAINT fk_transaction_user FOREIGN KEY (user_id)...	100000 row(s) affected Records: 100000 Duplicates: 0 Warnings: 0	1.468 sec

Relaciones entre tablas (resumen)

Transaction company: company_id → company.id

Transaction credit card: credit_card_id → credit_card.id

Transaction user: user_id → data_user.id

N3EX_2: L'empresa també us demana crear una vista anomenada "InformeTecnico" que contingui la següent informació: ID de la transacció, Nom de l'usuari, Cognom del l'usuari, IBAN de la targeta de crèdit feta servir, 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 de columnes segons calgui. Mostra el resultat de la vista, ordena els resultats de forma descendent en funció de la variable ID de la transacció.

Es crea un vista anomenada InformeTecnico per facilitar la consulta i l'anàlisi de transaccions realitzades pe l'usuari.

CREATE VIEW per crear la “taula virtual” tenim un SELECT amb àlies per definir les columnes, s'indica la taula principal que es transaction, utilitzem JOIN per fer unions de taules, els paràmetres per fer la vista es trobem en diferents taules. Per l'últim la vista ordenarà les transaccions de manera descendent per l'ID més alt que apareix el primer, així els valors més antics quedaran al final.

```
Query 1 SQL_Sprint_3 SQL File 5 estructura datos user (1) datos introducir sprint3 user
Dont Limit
189 -- Ejercicio 2 --
190 /* L'empresa també us demana crear una vista anomenada "InformeTecnico" que contingui la següent informació:
191 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
192 Assegureu-vos d'incloure informació rellevant de les taules que coneixereu i utilitzeu àlies per canviar de nom columnes segons calgui.
193 Mostra els resultats de la vista, ordena els resultats de forma descendent en funció de la variable ID de transacció.*/
194
195 CREATE VIEW InformeTecnico AS
196 SELECT
197     t.id AS ID_Transaccio,
198     u.name AS Nom_Usuari,
199     u.surname AS Cognom_Usuari,
200     cc.iban AS IBAN_Targeta,
201     c.company_name AS Entitat
202 FROM
203     transaction t
204 JOIN
205     user u ON t.user_id = u.id
206 JOIN
207     credit_card cc ON t.credit_card_id = cc.id
208 JOIN
209     company c ON t.company_id = c.id
210 ORDER BY
211     t.id DESC;
212 SELECT * FROM InformeTecnico;
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
---	------	--------	---------	------------------

Com a resultat, es genera una vista que conté els camps ID_transacció, Nom_usuari, Cognom_usuari, IBAN_Targeta i Entitat.

212 • SELECT * FROM InformeTecnico;

Result Grid

ID_Transaccio	Nom_Usuari	Cognom_Usuari	IBAN_Targeta	Entitat
FFFD31D6-9495-47CE-B54A-7D88E1CC274B	Bmrgli	Tprvrmrc	XX794814451211289182490922	Turpis Company
FFFCF76D-ECF0-4985-A2D0-B2A7B75998FC	Dfrled	Vlqqdl	XX636251701647892036676034	Amet Nulla Donec Corporation
FFFC9E8D-27C7-4ADE-98F2-7533EF40F126	Securp	Faofvqfy	XX162677143304223631437567	Nunc Interdum Incorporated
FFFB270D-F53A-4D5D-9666-E5307C53CC84	Ggzjpa	Uirzjuh	XX395114267082019952567052	Viverra Donec Foundation
FFF9E3CE-234E-408C-A8EF-F9CAD577224A	Yshimq	Zpsjsleed	XX8845462156537570367941	Conwallis In Incorporated
FFF9E178-6CD2-4DF9-99B0-49AE068809B1	Jevexp	Xwcwzwm	XX321405515711654384711481	Mus Aenean Eget Foundation
FFF867C9-17B5-4B1F-AFD9-F8023AAA449E	Fqlngd	Lvhfqxyi	XX278446342932680979729426	Cras Vehicula Aliquet Industries
FFF7042D-18C6-4DD0-823C-4D90A4AC8F26	Njoraa	Egsqzui	XX405009272572550082027209	Placerat LLP
FFF660D4-4244-47F6-9210-ESD1DCB99D80	Lopzaj	Itgryfay	XX63376659736627454015125	Pede Cum Ltd
FFF5C660-4441-436D-BD2F-E6C53B618622	Gmnbrn	Oxdvhkdl	XX237820256172646394016483	At Associates
FFF54F54-B439-41F0-BDD2-F7332DC1ACAD	Gqcfyy	Mpfhltn	XX802723943240147612158718	Enim Condimentum Ltd
FFF42F7D-7A0D-46E2-AF72-5996903F8AA9	Ddkuqq	Ycsbpryu	XX926442301555195974199541	At Pede Corp.
FFF42620-1968-4A1F-B9D7-27843CB097A5	Sjbrvp	Biqnzdfj	XX395457232638959102336873	Amet Nulla Donec Corporation
FFF290EB-79D2-497C-BDCE-2EC63FE3E9AD	Wcfzys	Dkkthgtd	XX458989048222230717519935	Integer Mollis Corp.
FFF20FR4-9014-44C3-R938-F31F7786C49A	Afhint	Yshumtrk	XX148483552743443735712134	Ac Fermentum Incorporated

InformeTecnico 2 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	11:44:08	SELECT * FROM InformeTecnico	100000 row(s) returned	0.516 sec / 0.047 sec