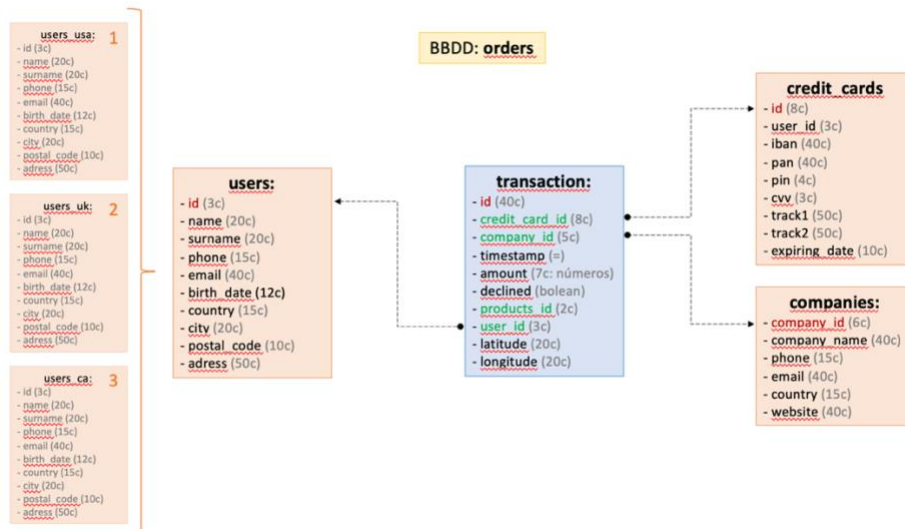


## Nivell 1

Descàrrega els arxius CSV, estudia'ls i dissenya una base de dades amb un esquema d'estrella que contingui, almenys 4 taules de les quals puguis realitzar les següents consultes:

- Primeramente, me he hecho un esquema de las tablas (en PowerPoint) para tener una idea de cómo se relacionan teniendo en cuenta los datos de los que disponemos, así como saber con qué tipo de variables trabajaré:

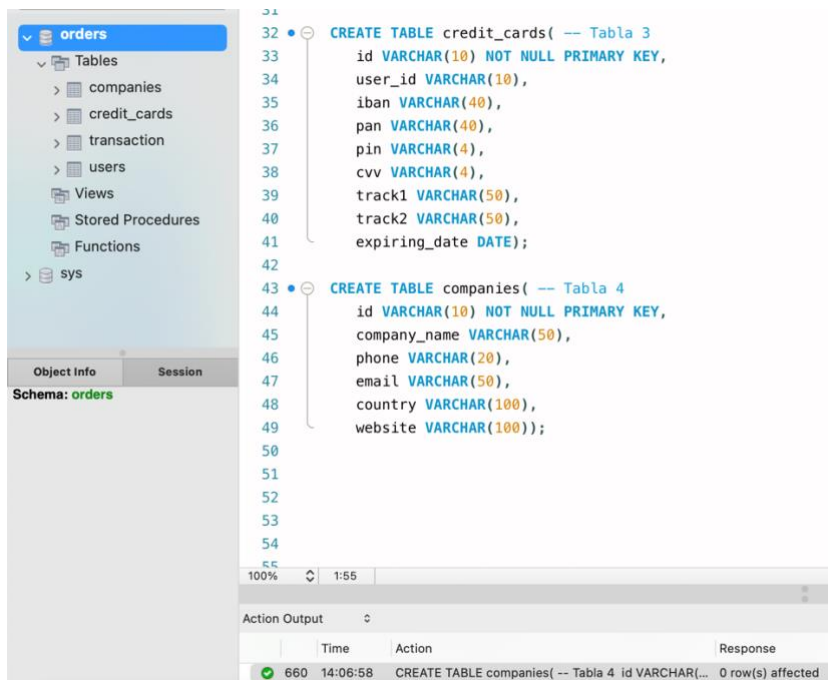


- Creamos la BBDD y las tablas:

```
SCHEMAS
Filter objects
orders
sys
orders
Tables
transaction
users
Views
Stored Procedures
Functions
sys
Object Info
Session
Schema: orders

1 -- Nivel 1: Dissenya una base de dades amb un esquema d'estrella:
2
3 • CREATE DATABASE IF NOT EXISTS orders -- Creo la base de datos
4   DEFAULT CHARACTER SET = 'utf8mb4' DEFAULT COLLATE 'utf8mb4_general_ci';
5
6 • use orders; -- y la selecciono para comenzar a crear las tablas que contendrá
7 • CREATE TABLE transaction( -- Tabla 1
8     id VARCHAR(40) NOT NULL PRIMARY KEY, -- a cada tabla le asigno su Primary Key
9     credit_card_id VARCHAR(10),
10    company_id VARCHAR(10),
11    timestamp TIMESTAMP,
12    amount DECIMAL(10,2),
13    declined BOOLEAN,
14    products_id VARCHAR(10),
15    user_id VARCHAR(10),
16    latitude VARCHAR(20),
17    longitude VARCHAR(20));
18
19 • CREATE TABLE users( -- Tabla 2
20     id VARCHAR(10) NOT NULL PRIMARY KEY,
21     name VARCHAR(20),
22     surname VARCHAR(20),
23     phone VARCHAR(20),
24     email VARCHAR(40),
25     birth_date DATE,
26     country VARCHAR(100),
27     city VARCHAR(100),
28     postal_code VARCHAR(10),
29     address VARCHAR(100));
30

Action Output
Time Action Response
112 11:45:28 CREATE TABLE users( -- Tabla 2 id VARCHAR(10) NOT NU... 0 row(s) affected
```



En la tabla 'transaction' elegí la mayoría de datos en formato VARCHAR (con una amplitud dependiente de los caracteres que tuviera nuestro campo) porque la mayoría de campos contenían números y letras o caracteres. En el caso del campo 'timestamp' como su nombre indica su tipo de dato ha de ser TIMESTAMP, 'amount' es decimal porque se trata de dinero y declined es un booleano porque solo acepta valores de 0 o 1.

En la tabla 'users' todos los campos son VARCHAR por lo mismo que en la anterior, excepto el campo 'birth\_date' que como su nombre indica es una fecha, ya que así será más fácil de trabajar y manipular en el futuro. Nombre, apellido y país los dejo como VARCHAR en lugar de texto por si existiera alguno de ellos que contuviera un carácter especial no contemplado en otros países (por ejemplo la 'ñ' de 'España' que no existe en los teclados de países de habla inglesa).

En el caso de la tabla 'credit\_card', al igual que en 'users' he definido todos los campos como VARCHAR exceptuando el campo 'expiring\_date' que es una fecha. En este caso tanto 'cvv' como 'pin' podrían ser considerados INT (enteros) pero entonces tendríamos problemas si existiera algún código que fuera, por ejemplo: 0001 porque lo detectaría como 1. El IBAN (International Bank Account Number), utilizado para identificar de forma estandarizada cualquier cuenta sea del país que sea, es una serie alfanumérica formada por un máximo de 34 caracteres dependiendo del país, por eso lo he definido como VARCHAR (40) para darle algo de margen. El CVV o código de verificación de la tarjeta es un código de tres o cuatro números utilizado para verificar que el usuario de la compra es el poseedor de la tarjeta por eso lo he definido como VARCHAR (4) porque es el máximo número de caracteres que puede aceptar. El PIN es el código de seguridad de la tarjeta que he definido como VARCHAR (4) porque suele ser de 4 números.

Por último, en la tabla 'companies' todos los datos son de tipo VARCHAR por lo anteriormente explicado.

- Al intentar cargar los csv para añadirlos a las tablas me daba error ya que la exportación e importación de datos estaba desactivada porque las versiones de MySQL anteriores a la 8.0 no tienen un archivo llamado my.cnf, y por eso al ejecutar: `SHOW VARIABLES LIKE "secure_file_priv"` Nos devuelve el valor NULL.

Variable_name	Value
secure_file_priv	NULL

Para solucionar esto:

1º Creé el archivo `/etc/my.cnf` que no existía y le añadí los siguientes valores:

```
[mysqld]
secure_file_priv=""
local_infile=1

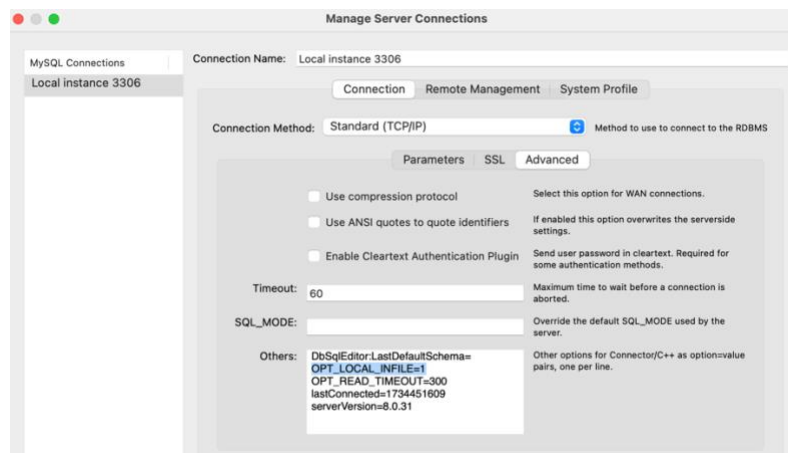
[client]
loose-local-infile=1
```

2º Me aseguré que la variable 'local\_infile' estuviera activa igualando su valor a true:

```
58 • SET GLOBAL local_infile = true;
59 • SHOW VARIABLES LIKE 'local_infile';
```

Variable_name	Value
local_infile	ON

3º Como continuaba dándome error, edité la conexión en la subpestaña 'Advanced' de la casilla 'Others' añadiendo: `OPT_LOCAL_INFILE=1`



Y entonces funcionó:

```
53 • SHOW VARIABLES LIKE 'secure_file_priv';
```

Variable_name	Value
secure_file_priv	

- Ahora sí, cargo los csv para añadirlos a las tablas:

### Tabla 1: 'transaction':

Al intentar cargar los datos el campo 'timestamp' me salen valores a 0, por eso cambio el campo de TIMESTAMP a VARCHAR(50) en la tabla creada por mí, importo los datos de los archivos csv y modifíco el formato de los datos del campo para volver a cambiar la variable de VARCHAR a TIMESTAMP:

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'transaction' table is selected under the 'orders' database. The table structure is displayed, showing columns: id (varchar(40) PK), credit\_card\_id (varchar(10)), company\_id (varchar(10)), timestamp (timestamp), amount (decimal(10,2)), declined (tinyint(1)), products\_id (varchar(10)), user\_id (varchar(10)), latitude (varchar(20)), and longitude (varchar(20)).

The main pane shows a SQL script with the following steps:

```

61 -- Al intentar cargar los datos el campo 'timestamp' (de 'transactions') tengo valores 0, cambio el tipo a VARCHAR(50) en la tabla:
62 ALTER TABLE orders.transaction
63 CHANGE COLUMN timestamp
64 timestamp VARCHAR(50);
65
66 -- Ahora importo los datos desde 'transaction.csv' a las tabla creada:
67 LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2-Especialización/ISprints/54/Datos/transactions.csv'
68 INTO TABLE orders.transaction
69 FIELDS TERMINATED BY ';' -- Delimitador de las columnas
70 LINES TERMINATED BY '\r\n' -- Fin de línea (para archivos que vienen de Windows)
71 IGNORE 1 ROWS; -- Ignorar la primera línea (encabezados)
72
73 -- Y modifíco el formato del campo 'timestamp' para volver a cambiar la variable de VARCHAR a TIMESTAMP
74 SET SQL_SAFE_UPDATES=0; -- para deshabilitar la opción de actualización segura, porque sino no me deja realizar el cambio
75
76 UPDATE orders.transaction
77 SET timestamp = DATE_FORMAT(STR_TO_DATE(timestamp, '%d/%m/%y %H:%i'), '%y-%m-%d %H:%i'); -- str_to_date cambia las fechas de formato
78
79 SET SQL_SAFE_UPDATES=1; -- vuelvo a habilitar la opción de actualización segura
80
81 ALTER TABLE orders.transaction -- y vuelvo a cambiar el tipo de variable
82 CHANGE COLUMN timestamp
83 timestamp TIMESTAMP;
84

```

The bottom pane shows the 'Action Output' with the following message:

```

121 11:54:19 ALTER TABLE orders.transaction -- y vuelvo a cambiar el tipo de variable 587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

```

Me sale un WARNING al importar 'transaction':

The screenshot shows the 'Action Output' pane with the following message:

```

116 11:52:54 LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2... 587 row(s) affected, 102 warning(s): 1265 Data truncated for column 'products_id'

```

Pero al visualizar la tabla todo está correcto y como menciona la columna 'products\_id' deduzco que es porque esta tiene varios valores separados por comas en algunas de sus filas.

14 • SELECT \* FROM orders.transaction;

The screenshot shows the SQL Server Enterprise Manager interface. The 'transaction' table is selected, and the 'Result Grid' is displayed, showing the first 10 rows of data. The columns are: id, credit\_card\_id, company\_id, timestamp, amount, declined, products\_id, user\_id, latitude, and longitude.

The data rows are as follows:

id	credit_card_id	company_id	timestamp	amount	declined	products_id	user_id	latitude	longitude
02C6201E-D90A-1859-B4EE-88...	CcU-2938	b-2362	2021-08-28 23:42:00	466.92	0	71, 1, 19	92	819.184.589.824	-125.275.561.984
0466A42E-47CF-8D24-FD01-C0...	CcU-4219	b-2302	2021-07-26 07:29:00	49.53	0	47, 97, 43	170	-439.694.885.888	-1.175.251.835.904
063FBA79-99EC-66FB-29F7-257...	CcU-2987	b-2250	2022-01-06 21:25:00	92.61	0	47, 67, 31	275	-81.222.680.576	-129.049.879.552
0668296C-CDB9-A883-76BC-2E...	CcU-3743	b-2618	2022-01-26 02:07:00	394.18	0	89, 83, 79	265	-343.593.055.232	-100.555.928.064
06CD9AA5-9B42-D684-DDDD-A...	CcU-2959	b-2346	2021-10-26 23:00:00	279.93	0	43, 31	92	337.381.445.632	158.298.210.304
07A46D48-31A3-7E87-65B9-0D...	CcU-3225	b-2386	2021-06-28 21:11:00	340.87	1	47, 23	272	388.341.526.528	92.190.545.408
09DE92CE-6F27-2BB7-13B5-93...	CcU-3071	b-2298	2021-05-11 20:40:00	303.05	1	67, 7	275	711.705.612.288	105.756.752.896

The bottom pane shows the 'Action Output' with the following message:

```

122 11:56:10 SELECT * FROM orders.transaction 587 row(s) returned

```

Tabla 2: 'users':

Al intentar cargar los datos el campo 'birth\_date' me salen valores a 0, por eso cambio el campo de DATE a VARCHAR(20) en la tabla creada por mí. También modifico el campo 'id' de VARCHAR a INT porque me daba errores de ordenación al intentar cargar los datos.

Object Info

Session

Table: **users**

Columns:

id

int PK

name

varchar(20)

surname

varchar(20)

phone

varchar(20)

email

varchar(40)

birth\_date

varchar(20)

country

varchar(100)

city

varchar(100)

postal\_code

varchar(10)

adress

varchar(100)

```
84
85 -- Al intentar cargar los datos el campo 'birth_date' (de 'users') me salen valores a 0, cambio el tipo de DATE a VARCHAR(20)
86 • ALTER TABLE orders.users
87   CHANGE COLUMN birth_date
88     birth_date VARCHAR(20);
89
90 -- Tambien cambio el campo 'id' de VARCHAR a INT porque me daba errores de ordenación
91 • ALTER TABLE orders.users
92   CHANGE COLUMN id
93     id INT;
94
100% 1:94
```

Action Output

	Time	Action	Response
133	12:14:30	ALTER TABLE orders.users CHANGE COLUMN id INT	150 row(s) affected Records: 150 Duplicates: 0 Warnings: 0

Importo los datos de los archivos csv:

orders

Tables

companies

credit\_cards

transaction

users

Views

Stored Procedures

Functions

sys

Object Info

Session

Table: users

Columns:

id

int PK

name

varchar(20)

surname

varchar(20)

phone

varchar(20)

email

varchar(40)

birth\_date

varchar(20)

country

varchar(100)

city

varchar(100)

postal\_code

varchar(10)

adress

varchar(100)

95

96

97 -- Luego importo los datos desde los diferentes archivos de csv (users\_usa, users\_uk y users\_ca) a la tabla creada:

98 • LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2-Especialización/iSprints/S4/Datos/users\_usa.csv'

99 INTO TABLE orders.users

100 FIELDS TERMINATED BY ',' -- Delimitador de las columnas

101 OPTIONALLY ENCLOSED BY '"' -- Delimitador de texto (comillas dobles), para que me coja bien la fecha y la dirección

102 LINES TERMINATED BY '\r\n' -- Fin de línea (para archivos que vienen de Windows)

103 IGNORE 1 LINES; -- Ignorar la primera línea (encabezados)

104

105 • LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2-Especialización/iSprints/S4/Datos/users\_uk.csv'

106 INTO TABLE orders.users

107 FIELDS TERMINATED BY ',' -- Delimitador de las columnas

108 OPTIONALLY ENCLOSED BY '"' -- Delimitador de texto (comillas dobles), para que me coja bien la fecha y la dirección

109 LINES TERMINATED BY '\r\n' -- Fin de línea (para archivos que vienen de Windows)

110 IGNORE 1 LINES; -- Ignorar la primera línea (encabezados)

111

112 • LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2-Especialización/iSprints/S4/Datos/users\_ca.csv'

113 INTO TABLE orders.users

114 FIELDS TERMINATED BY ',' -- Delimitador de las columnas

115 OPTIONALLY ENCLOSED BY '"' -- Delimitador de texto (comillas dobles), para que me coja bien la fecha y la dirección

116 LINES TERMINATED BY '\r\n' -- Fin de línea (para archivos que vienen de Windows)

117 IGNORE 1 LINES; -- Ignorar la primera línea (encabezados)

118

100% 1:96

Action Output

Time

Action

Response

136

12:17:43

LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2... 75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0

Modifico el formato de los datos del campo para volver a cambiar la variable de VARCHAR a DATE y visualizo la tabla:

Object Info

Session

Table: users

Columns:

id

int PK

name

varchar(20)

surname

varchar(20)

phone

varchar(20)

email

varchar(40)

birth\_date

date

country

varchar(100)

city

varchar(100)

postal\_code

varchar(10)

adress

varchar(100)

```
118 • SET SQL_SAFE_UPDATES=0; -- para deshabilitar la opción de actualización segura, porque sino no me deja realizar el cambio
119
120 • UPDATE orders.users
121   SET birth_date = DATE_FORMAT(STR_TO_DATE(birth_date, '%b %d, %Y'), '%Y-%m-%d') -- str_to_date cambia las fechas de formato
122   WHERE users.id IS NOT null;
123
124 • SET SQL_SAFE_UPDATES=1; -- vuelvo a habilitar la opción de actualización segura
125
126 • ALTER TABLE orders.users -- y vuelvo a cambiar el tipo de variable para 'birth date' a DATE
127   CHANGE COLUMN birth_date
128     birth_date DATE;
129
100% 4:126
```

Action Output

Time	Action	Response
141 12:23:56	ALTER TABLE orders.users -- y vuelvo a cambiar el tipo de	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0



15 • **SELECT \* FROM orders.users;**

100% 1:7

Result Grid Filter Rows: Search Edit: Export/Import:

	id	name	surname	phone	email	birth_date	country	city	postal_code	adress
▶ 1	Zeus	Gamble		1-282-581-0551	interdum.enim@protonmail.edu	1985-11-17	United States	Lowell	73544	348-7818 Sagittis St.
2	Garrett	Mconnell		(718) 257-2412	integer.vitae.nibh@protonmail.org	1992-08-23	United States	Des Moines	59464	903 Sit Ave
3	Ciaran	Harrison		(522) 598-1365	interdum.feugiat@aol.org	1998-04-29	United States	Columbus	56518	736-2063 Tellus St.
4	Howard	Stafford		1-411-740-3269	omare.egestas@icloud.edu	1989-02-18	United States	Kailua	77417	Ap #545-2244 Erat. Rd.
5	Hayfa	Pierce		1-554-541-2077	et.malesuada.fames@hotmail.org	1998-09-26	United States	Sandy	31564	341-2821 Ultrices Av.
6	Joel	Tyson		(718) 288-8020	gravida.nunc.sed@yahoo.ca	1989-10-15	United States	Nashville	96838	888-2799 Amet Street
7	Rafael	Jimenez		(817) 689-0478	eget@outlook.ca	1981-12-04	United States	Hillsboro	29874	8627 Malesuada Rd.

users 5

Action Output

	Time	Action	Response
✓ 142	12:26:34	SELECT * FROM orders.users	275 row(s) returned

Tabla 3: 'credit\_cards':

Primero cambio el campo 'user\_id' a INT para que coincida con el campo 'id' de la tabla 'users' y el campo 'expiring\_date' a VARCHAR para que acepte los datos y luego lo volveré a cambiar a DATE:

Object Info	Session
Table: credit_cards	
Columns:	
id	varchar(10) PK
user_id	int
iban	varchar(40)
pan	varchar(40)
pin	varchar(4)
cvv	varchar(4)
track1	varchar(50)
track2	varchar(50)
expiring_date	date

```

145 -- Cambio el campo 'user_id' (de 'credit_cards') de VARCHAR a INT para que sea igual al campo 'id' de la tabla 'users':
146 • ALTER TABLE orders.credit_cards
147   CHANGE COLUMN user_id
148   user_id INT;
149
150 -- También cambio el tipo de dato de 'expiring_date' de DATE a VARCHAR(20) en la tabla:
151 • ALTER TABLE orders.credit_cards
152   CHANGE COLUMN expiring_date
153   expiring_date VARCHAR(20);

```

100% 1:144

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 677	14:10:04	ALTER TABLE orders.credit_cards CHANGE COLUMN...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.045 sec

Importo los datos del archivo csv, cambio el formato de fecha de 'expiring\_date', vuelvo a cambiar el tipo de variable a DATE y visualizo mi tabla:

```

144 -- Importo los datos desde 'credit_cards.csv' a la tabla creada:
145 • LOAD DATA LOCAL INFILE '/Users/Liss/Desktop/ItAcademy/2-Especialización/iSprints!/54/Datos/credit_cards.csv'
146 INTO TABLE orders.credit_cards
147 FIELDS TERMINATED BY ',' -- Delimitador de las columnas
148 LINES TERMINATED BY '\n' -- Fin de línea
149 IGNORE 1 LINES; -- Ignorar la primera línea (encabezados)
150
151 -- Y modifíco el formato del campo 'expiring_date' para volver a cambiar la variable de VARCHAR a DATE
152 • SET SQL_SAFE_UPDATES=0; -- para deshabilitar la opción de actualización segura
153
154 • UPDATE orders.credit_cards
155   SET expiring_date = DATE_FORMAT(STR_TO_DATE(expiring_date, '%m/%d/%Y'), '%Y-%m-%d') -- str_to_date cambia las fechas de formato
156   WHERE credit_cards.id IS NOT null;
157
158 • SET SQL_SAFE_UPDATES=1; -- vuelvo a habilitar la opción de actualización segura
159
160 • ALTER TABLE orders.credit_cards -- y vuelvo a cambiar el tipo de variable de 'expiring_date' a DATE
161   CHANGE COLUMN expiring_date
162   expiring_date DATE;
163

```

100% 1:140

Action Output

	Time	Action	Response
✓ 176	13:06:08	ALTER TABLE orders.credit_cards -- y vuelvo a cambiar el t...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

15 • **SELECT \* FROM orders.credit\_cards;**

id	user_id	iban	pan	pin	cvv	track1	track2	expiring_date
CcU-2938	275	TR301950312213576817638661	5424465566813633	3257	984	%B8383712448554646*WovsxejDpwiev*86041...	%B7653863056044187=800716333673	2022-10-30
CcU-2945	274	DO26854763748537475216568689	5142423821948828	9080	887	%B4621311609958661*UftuyfsSeimxn*061062...	%B4149568437843501=510714033071	2023-08-24
CcU-2952	273	BG45IVQL52710525608255	4556 453 55 5287	4598	438	%B2183285104307501*CddytycUxwfdq*59079...	%B6778580257827162=6906859740077	2021-06-29
CcU-2959	272	CR7242477244335841535	372461377349375	3583	667	%B7281111956795320*XocddyjBkeacd*090162...	%B42461544489281853=280522391678	2023-02-24
CcU-2966	271	BG72LKTQ15627628377363	448566 886747 7265	4900	130	%B4728932322756223*UhlgsuFbmwgj*72022...	%B2318571115599881=890821578475	2024-10-29
CcU-2973	270	PT87806228135092429456346	544 58654 54343 384	8760	887	%B4761405253275637*HjrnipoBlejrt*7108515...	%B7816169831446746=1310277279	2025-01-30
CcU-2980	269	DE39241881883086277136	402400 7145845969	5075	596	%B7320483593870549*OokzqxrHpsaed*49017...	%B2474313962214151=041221913175	2022-07-24

credit\_cards 7

Action Output

Time	Action	Response
177 13:08:58	SELECT * FROM orders.credit_cards	275 row(s) returned

Tabla 4: 'companies':

Importo los datos del archivo csv y visualizo mi tabla:

Table: **companies**

Columns:

- id: varchar(10) PK
- company\_name: varchar(50)
- phone: varchar(20)
- email: varchar(50)
- country: varchar(100)
- website: varchar(100)

```

142 -- Y también importo los datos desde 'companies.csv' a la tabla creada:
143 • LOAD DATA LOCAL INFILE '/Users/liss/Desktop/ItAcademy/2-Especialización/iSprints/54/Datos/companies.csv'
144 INTO TABLE orders.companies
145 FIELDS TERMINATED BY ',' -- Delimitador de las columnas
146 LINES TERMINATED BY '\r\n' -- Fin de línea
147 IGNORE 1 LINES; -- Ignorar la primera línea (encabezados)
148
149
100% 60:138

```

Action Output

Time	Action	Response
158 12:44:00	SELECT * FROM orders.credit_cards	0 row(s) returned

15 • **SELECT \* FROM orders.companies;**

id	company_name	phone	email	country	website
b-2222	Ac Fermentum Incorporated	06 85 56 52 33	donec.porttitor.tellus@yahoo.net	Germany	https://instagram.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110
b-2238	Ante iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.couk	Norway	https://mytimes.com/user/110
b-2246	Sed Nunc Ltd	02 62 64 73 48	nibh@yahoo.org	United Kingdom	https://cnn.com/one

companies 2

Action Output

Time	Action	Response
160 12:45:42	SELECT * FROM orders.companies	100 row(s) returned

- Añado las Foreign Keys que relacionaran las tablas entre ellas en este esquema de estrella:

Functions

sys

Object Info

Session

Table: **transaction**

Columns:

- id: varchar(40) PK
- credit\_card\_id: varchar(10)
- company\_id: varchar(10)
- timestamp: timestamp
- amount: decimal(10,2)
- declined: tinyint(1)
- products\_id: varchar(10)
- user\_id: int
- latitude: varchar(20)
- longitude: varchar(20)

```

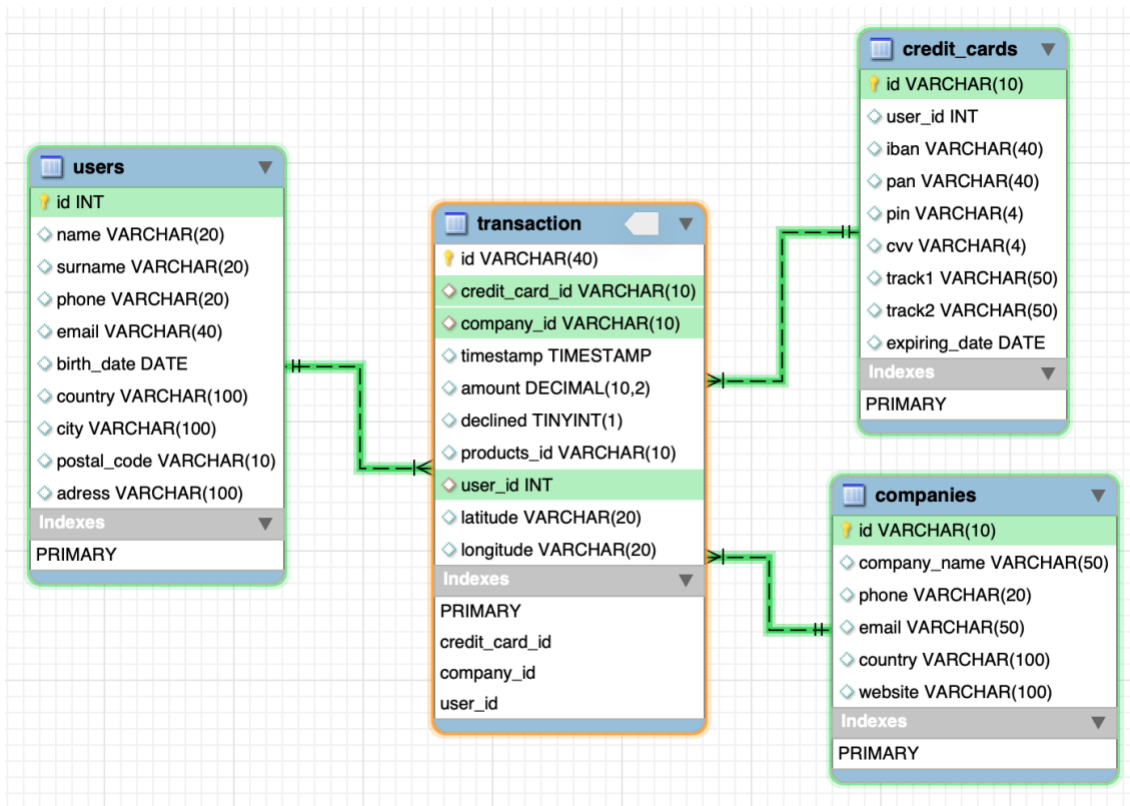
167 -- Ahora añado las Foreign Keys que relacionaran las tablas entre ellas en este esquema de estrella
168 • ALTER TABLE orders.transaction
169 ADD FOREIGN KEY (credit_card_id) REFERENCES credit_cards(id);
170
171 • ALTER TABLE orders.transaction
172 ADD FOREIGN KEY (company_id) REFERENCES companies(id);
173
174 • ALTER TABLE orders.transaction
175 CHANGE COLUMN user_id
176 user_id INT; -- cambio el campo 'user_id' a INT para que sea igual al campo 'id' de la tabla 'users'
177
178 • ALTER TABLE orders.transaction
179 ADD FOREIGN KEY (user_id) REFERENCES users(id);
180
100% 4:178

```

Action Output

Time	Action	Response
182 17:14:47	ALTER TABLE orders.transaction ADD FOREIGN KEY (user_id) REFERENCES users(id);	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

- Y así queda mi BBDD con esquema de estrella:



En el diagrama observamos que la tabla de hechos sería 'transaction' y las tablas de dimensiones 'companies', 'users' y 'credit\_cards'. La relación entre las tres tablas se establece a través de las Primary Keys de las tablas de dimensiones (id de 'companies', id de 'users' e id de 'credit\_cards') y las Foreign Keys de la tabla de hechos: `company_id` que se relaciona con id de 'company', `user_id` que se relaciona con la id de 'users' y `credit_card_id` que se relaciona con id de 'credit\_card'.

\*Aunque también había un archivo de csv que contenía datos de una tabla llamada 'products' por ahora no lo añadido a mi base de datos porque no lo usaré hasta el nivel 3.



## Nivell 1

### Exercici 1

Realitza una subconsulta que mostri tots els usuaris amb més de 30 transaccions utilitzant almenys 2 taules.

```
185 • SELECT u.name AS nombre_usuario, u.surname AS apellido_usuario, (SELECT COUNT(t.id)
186                                     FROM transaction AS t
187                                     WHERE t.user_id = u.id) AS transacciones
188 FROM users AS u
189 WHERE u.id IN (SELECT t.user_id
190               FROM transaction AS t
191               GROUP BY 1
192               HAVING COUNT(t.id) > 30);
```

100% 100:183

Result Grid Filter Rows: Search Export:

nombre_usuario	apellido_usuario	transacciones
Lynn	Riddle	39
Ocean	Nelson	52
Hedwig	Gilbert	76
Kenyon	Hartman	48

Result 3

Action Output

	Time	Action	Response	Duration / Fetch Time
8	13:24:27	SELECT u.name AS nombre_usuario, u.surna...	4 row(s) returned	0.022 sec / 0.000023...

### Exercici 2

Mostra la mitjana d'amount per IBAN de les targetes de crèdit a la companyia Donec Ltd, utilitza almenys 2 taules.

```
193 • SELECT c.company_name AS nombre_compañia, cc.id AS tarjeta_credito, cc.iban, ROUND(AVG(t.amount),2) AS importe
194 FROM credit_cards AS cc
195 JOIN transaction AS t
196 JOIN companies AS c
197 WHERE cc.id = t.credit_card_id
198 AND t.company_id IN (SELECT c.id
199                     FROM companies
200                     WHERE c.company_name = 'Donec Ltd')
201 GROUP BY 1, 2;
```

100% 1:203

Result Grid Filter Rows: Search Export:

nombre_compañia	tarjeta_credito	iban	importe
Donec Ltd	CcU-2973	PT87806228135092429456346	203.72

Result 14

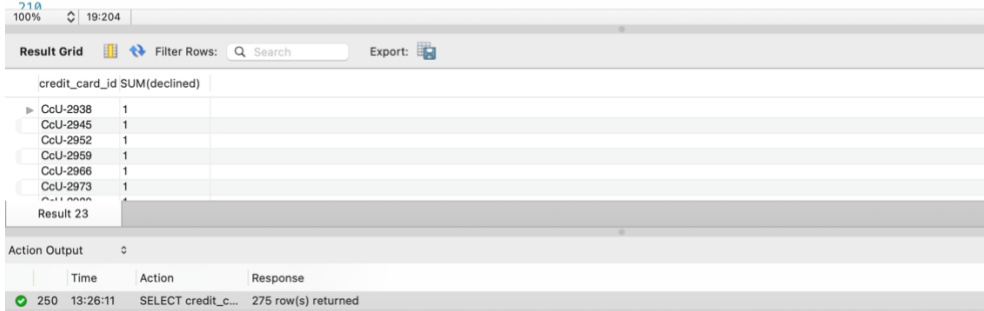
Action Output

	Time	Action	Response
209	19:51:44	SELECT c.company_name AS nombre_compañia, cc.id AS t...	1 row(s) returned

## Nivell 2

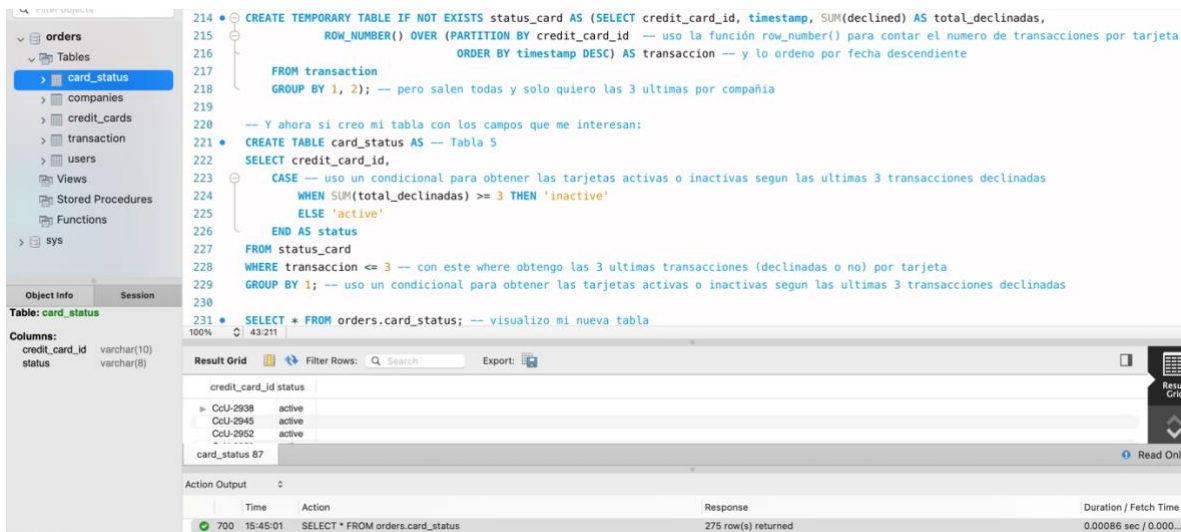
Crea una nova taula que reflecteixi l'estat de les targetes de crèdit basat en si les últimes tres transaccions van ser declinades i genera la següent consulta:

```
206 • SELECT credit_card_id, SUM(declined) -- compruebo cuantas tarjetas han sido declinadas más de una vez en general
207 FROM transaction
208 GROUP BY 1;
209 -- Ninguna tarjeta ha sido declinada más de una vez
```



credit_card_id	SUM(declined)
CcU-2938	1
CcU-2945	1
CcU-2952	1
CcU-2959	1
CcU-2966	1
CcU-2973	1

- Primero creo una tabla temporal, luego creo la tabla con los campos que me interesan y la visualizo:



```
214 • CREATE TEMPORARY TABLE IF NOT EXISTS status_card AS (SELECT credit_card_id, timestamp, SUM(declined) AS total_declinadas,
215 ROW_NUMBER() OVER (PARTITION BY credit_card_id -- uso la función row_number() para contar el numero de transacciones por tarjeta
216 ORDER BY timestamp DESC) AS transaccion -- y lo ordeno por fecha descendiente
217 FROM transaction
218 GROUP BY 1, 2); -- pero salen todas y solo quiero las 3 ultimas por compañía
219
220 -- Y ahora si creo mi tabla con los campos que me interesan:
221 • CREATE TABLE card_status AS -- Tabla 5
222 SELECT credit_card_id,
223 CASE -- uso un condicional para obtener las tarjetas activas o inactivas segun las ultimas 3 transacciones declinadas
224 WHEN SUM(total_declinadas) >= 3 THEN 'inactive'
225 ELSE 'active'
226 END AS status
227 FROM status_card
228 WHERE transaccion <= 3 -- con este where obtengo las 3 ultimas transacciones (declinadas o no) por tarjeta
229 GROUP BY 1; -- uso un condicional para obtener las tarjetas activas o inactivas segun las ultimas 3 transacciones declinadas
230
231 • SELECT * FROM orders.card_status; -- visualizo mi nueva tabla
```

credit_card_id	status
CcU-2938	active
CcU-2945	active
CcU-2952	active

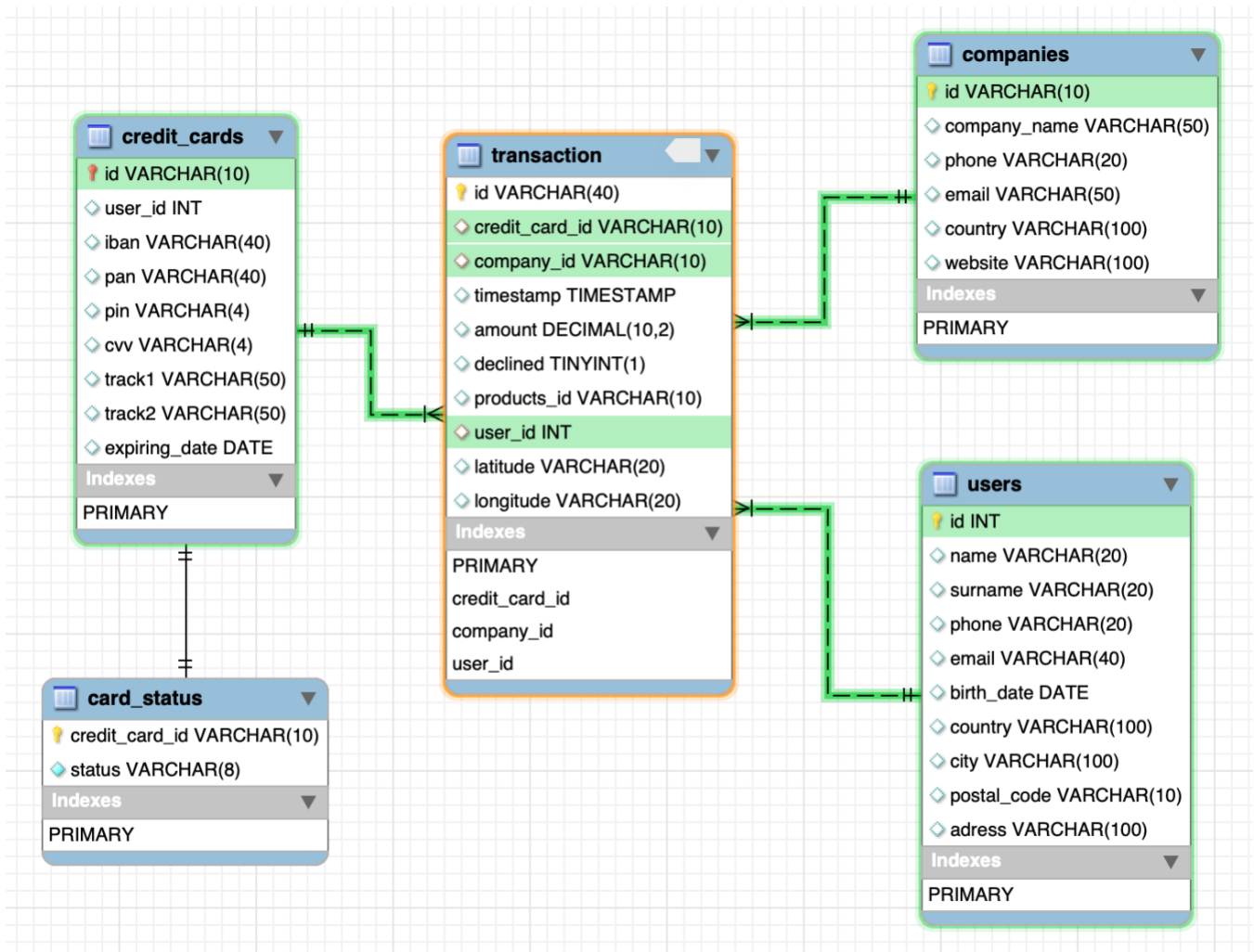
- Por último, añado su PK y la relaciono con una FK de la tabla 'credit\_cards':



```
232
233 -- Ahora creo la PK y la FK con la que la relacionaré:
234 • ALTER TABLE card_status
235 ADD PRIMARY KEY (credit_card_id);
236
237 • ALTER TABLE credit_cards
238 ADD FOREIGN KEY (id) REFERENCES card_status(credit_card_id);
```

credit_card_id	status
CcU-2938	active
CcU-2945	active
CcU-2952	active

- Y así queda nuestra BBDD ahora:



La nueva tabla 'card\_status' nos queda relacionada con la tabla de dimensiones 'credit\_cards' a través de su PK credit\_card\_id con la PK (id) de la tabla 'credit\_cards' en una relación de 1 a 1, de manera que si se actualizan los datos de las tarjetas de crédito se actualice el estado de las tarjetas activas o inactivas.

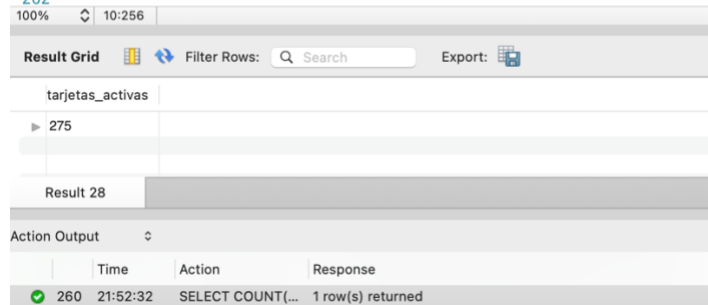
## Nivell 2

### Exercici 1

Quantes targetes estan actives?

- Todas las tarjetas están activas:

```
259 • SELECT COUNT(credit_card_id) AS tarjetas_activas
260 FROM card_status
261 WHERE status = 'active'; -- todas las tarjetas están activas
262
```

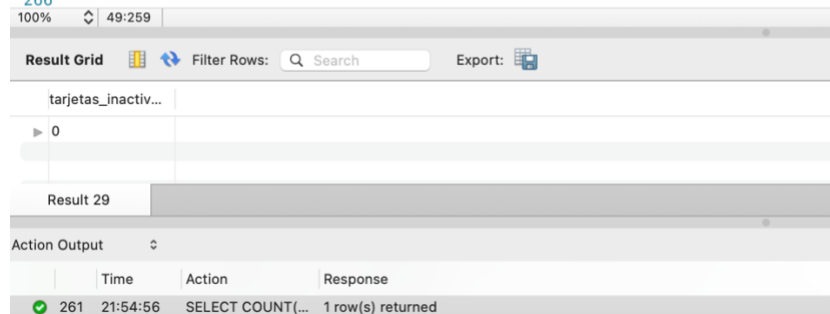


The screenshot shows a SQL query execution interface. The query is: `SELECT COUNT(credit_card_id) AS tarjetas_activas FROM card_status WHERE status = 'active';`. The result grid shows a single row with the value 275. The action output shows that the query returned 1 row(s).

Time	Action	Response
21:52:32	SELECT COUNT(...)	1 row(s) returned

- Lo comprobamos mirando que no haya tarjetas inactivas:

```
263 • SELECT COUNT(credit_card_id) AS tarjetas_inactivas
264 FROM card_status
265 WHERE status = 'inactive'; -- comprobamos mirando las tarjetas inactivas
266
```



The screenshot shows a SQL query execution interface. The query is: `SELECT COUNT(credit_card_id) AS tarjetas_inactivas FROM card_status WHERE status = 'inactive';`. The result grid shows a single row with the value 0. The action output shows that the query returned 1 row(s).

Time	Action	Response
21:54:56	SELECT COUNT(...)	1 row(s) returned

### Nivell 3

Crea una taula amb la qual puguem unir les dades del nou arxiu products.csv amb la base de dades creada, tenint en compte que des de transaction tens product\_ids. Genera la següent consulta:

- Primero creamos la tabla 'products' y le añadimos los datos del csv con el mismo nombre:

The screenshot shows a database management interface with a sidebar on the left containing a tree view of database objects. The 'products' table is selected under the 'orders' schema. The main pane displays SQL code for creating the table and loading data from a CSV file. The 'Result Grid' shows the first four rows of the loaded data.

```
274 CREATE TABLE products( -- Tabla 6
275     id VARCHAR(10),
276     product_name VARCHAR(40),
277     price VARCHAR(10),
278     colour VARCHAR(10),
279     weight DECIMAL(10,2),
280     warehouse_id VARCHAR(10));
281
282 -- cargamos los datos de la tabla desde el archivo csv:
283 LOAD DATA LOCAL INFILE '/Users/Liss/Desktop/ItAcademy/2-Especialización/iSprints!/S4/Datos/products.csv'
284 INTO TABLE orders.products
285 FIELDS TERMINATED BY ',' -- Delimitador de las columnas
286 LINES TERMINATED BY '\n' -- Fin de línea
287 IGNORE 1 ROWS; -- Ignorar la primera línea (encabezados)
288
289 -- visualizamos la tabla:
290 SELECT * FROM orders.products;
```

id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	\$161.11	#7c7c7c	1.00	WH-4
2	Tarly Stark	\$9.24	#919191	2.00	WH-3
3	duel tourney Lannister	\$171.13	#d8d8d8	1.50	WH-2
4	warden south duel	\$71.89	#111111	3.00	WH-1

- Elimino el símbolo de dólar de la columna 'price' de 'products' para convertirla en decimal y trabajar mejor con sus datos ya que se trata de dinero:

The screenshot shows the same database management interface. The SQL code pane now contains commands to update the 'price' column by removing dollar signs and changing its data type to DECIMAL. The 'Result Grid' shows the updated data, including a new row for 'skywalker ewok'.

```
292 -- sustituimos los símbolos '$' de la columna price y cambiamos el tipo de dato de VARCHAR a DECIMAL:
293 SET SQL_SAFE_UPDATES=0;
294
295 UPDATE products
296 SET price = REPLACE(price, "$", "")
297 WHERE price IS NOT NULL;
298
299 ALTER TABLE orders.products
300 CHANGE COLUMN price
301 price DECIMAL(10,2);
302
303 SET SQL_SAFE_UPDATES=1;
```

id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	161.11	#7c7c7c	1.00	WH-4
2	Tarly Stark	9.24	#919191	2.00	WH-3
3	duel tourney Lannister	171.13	#d8d8d8	1.50	WH-2
4	warden south duel	71.89	#111111	3.00	WH-1
5	skywalker ewok	171.22	#d8d8d8	3.20	WH-0



- También elimino los espacios después de las comas de 'products\_id' de la tabla 'transaction', para que al compararla con 'id' de la tabla 'products' no nos tenga en cuenta los espacios vacíos:

```

289 • SET SQL_SAFE_UPDATES=0; -- para desactivar la opción de actualización segura, porque sino no me deja realizar el cambio
290
291 • UPDATE transaction
292   SET products_id = REPLACE(products_id, " ", "")
293   WHERE products_id IS NOT NULL;
294
295 • SET SQL_SAFE_UPDATES=1; -- vuelvo a habilitar la opción de actualización segura
296
297 • SELECT * FROM orders.transaction; -- visualizo los cambios
298

```

id	credit_card_id	company_id	timestamp	amount	declined	products_id	user_id	latitude	longitude
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:00	466.92	0	71,1,19	92	819.184.589.824	-125.275.561.984
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26 07:29:00	49.53	0	47,97,43	170	-439.694.885.888	-1.175.251.835.904
063FBA79-99EC-66FB-29F7-257...	CcU-2987	b-2250	2022-01-06 21:25:00	92.61	0	47,67,31	275	-81.222.680.576	-129.049.879.552

transaction 92

Action Output

Time	Action	Response
16:03:51	SELECT * FROM orders.transaction	587 row(s) returned

- Y creo la tabla 'sales' que es una tabla puente que me relaciona las tablas 'products' y 'transaction':

```

317 • CREATE TABLE sales AS( -- tabla 7
318   SELECT t.id AS transaction_id, p.id AS product_id
319   FROM products AS p
320   JOIN transaction AS t
321   ON FIND_IN_SET(p.id, t.products_id));
322
323 -- visualizo la nueva tabla:
324 • SELECT * FROM orders.sales;
325
326 -- y compruebo que todos los valores son correctos, sabiendo que tenemos 578 transaction_id diferentes:
327 • SELECT COUNT(DISTINCT transaction_id)
328   FROM sales;

```

transaction_id	product_id
02C6201E-D90A-1859-B4EE-88D2986D3B02	71
02C6201E-D90A-1859-B4EE-88D2986D3B02	19
02C6201E-D90A-1859-B4EE-88D2986D3B02	1
0466A42E-47CF-8D24-FD01-C0B689713128	97
0466A42E-47CF-8D24-FD01-C0B689713128	47

sales 64

Action Output

Time	Action	Response
12:11:52	SELECT * FROM orders.sales	1355 row(s) returned

- Por último, reo la PK de la tabla 'products' y le añado las FKs a la tabla 'sales' que la relacionan con 'transaction' y 'products':

```

Table: products
Columns:
  id          varchar(10) PK
  product_name varchar(40)
  price       decimal(10,2)
  colour      varchar(10)
  weight      decimal(10,2)
  warehouse_id varchar(10)

314 • ALTER TABLE products
315   ADD PRIMARY KEY (id);
316
317 -- le añado las FKs a la tabla 'sales' que la relacionan con 'transaction' y 'products':
318 • ALTER TABLE sales
319   ADD FOREIGN KEY (transaction_id) REFERENCES transaction (id),
320   ADD FOREIGN KEY (product_id) REFERENCES products (id);

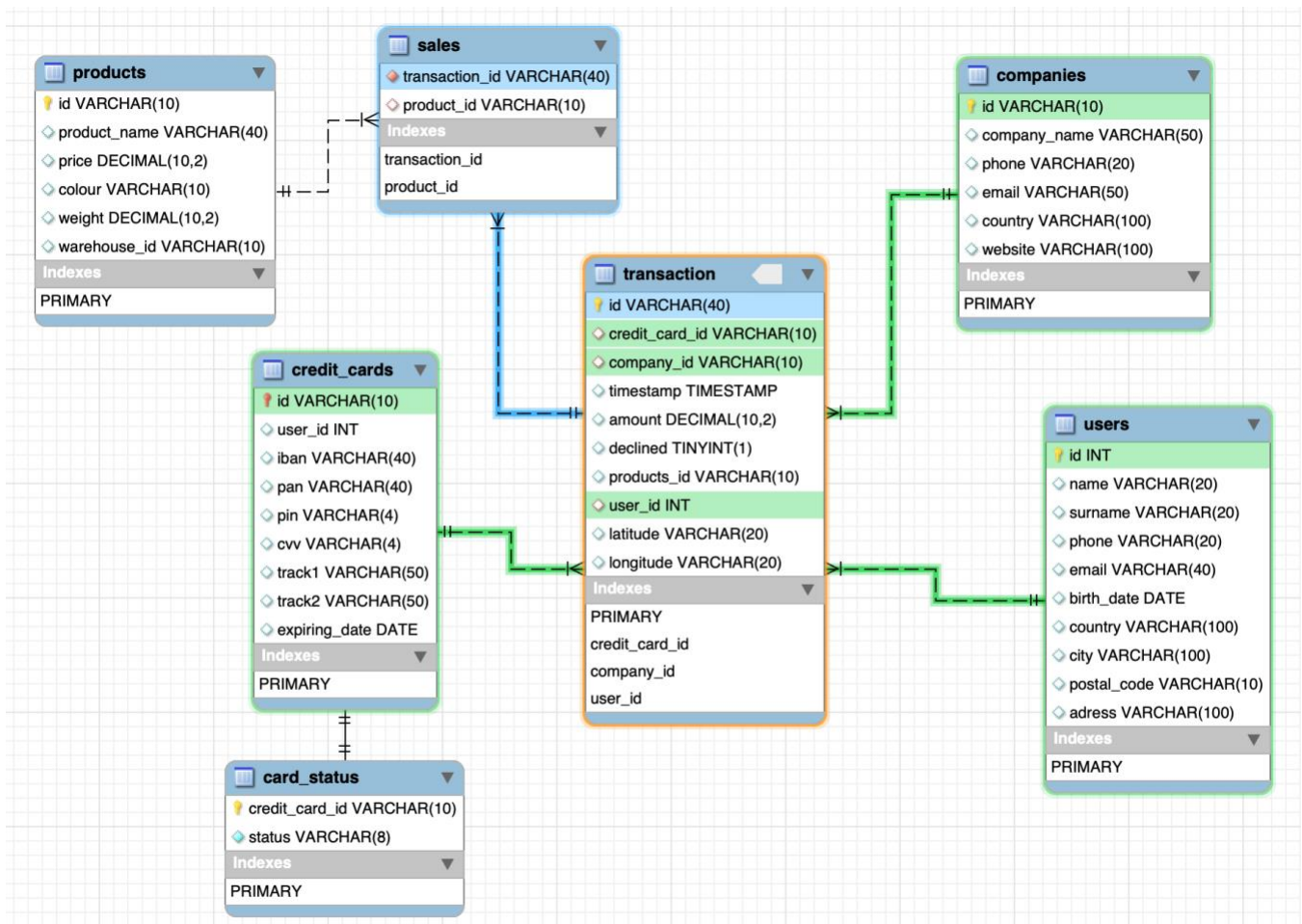
```

products 0

Action Output

Time	Action	Response
16:07:21	ALTER TABLE sales ADD FOREIGN KEY (transaction_id) REFERENCES transaction (id),...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

- Y así queda nuestra BBDD final:



La tabla puente 'sales' nos queda relacionada con la tabla de hechos 'transaction' a través de una FK llamada `transaction_id` que se relaciona con la PK (`id`) de la tabla 'transaction' y también se relaciona a través de la FK `producto_id` con la PK (`id`) de la tabla 'products' que es una nueva tabla de dimensiones que añadimos a nuestra BBDD.

## Nivell 3

### Exercici 1

Necessitem conèixer el nombre de vegades que s'ha venut cada producte.

```
325 • SELECT p.product_name AS producto, COUNT(s.product_id) AS cantidad_de_ventas
326 FROM sales AS s
327 JOIN products AS p
328 ON p.id = s.product_id
329 GROUP BY p.id
330 ORDER BY cantidad_de_ventas DESC;
```

100% 65:323

**Result Grid** Filter Rows: Search Export:

producto	cantidad_de_ventas
riverlands north	65
Winterfell	65
Direwolf riverlands the	63
Tarly Stark	61
skywalker ewok sith	60
Tully	60
duel	57

Result 1

ction Output

	Time	Action	Response
3	18:39:19	SELECT p.product_name AS producto, COUNT(s.product_id) AS cantidad_de_ventas FROM sales AS s JOIN produc...	26 row(s) returned