

Sprint 4

Creación de bases de datos

The screenshot shows the MySQL Workbench interface. In the top left, there's a 'Nivell 1' badge with three stars. The left sidebar shows 'SCHEMAS' with 'biblioteca' and 'db_companies' selected. The main area has a code editor with the following SQL script:

```
1
2
3
4
5
6 • CREATE DATABASE db_companies;
7
8
9
10
11
12
13
14
15
16
```

The 'Output' tab at the bottom shows the results of the execution:

#	Time	Action	Message
1	10:31:35	CREATE TABLE db_companies.companies (company_id CHAR (6) PRIMARY KEY, company_name VARCHAR (100), phone VARCHAR (15), email VARCHAR (100), country VARCHAR (100), website VARCHAR (100));	Error Code: 1049. Unknown database 'db_companies'
2	10:31:41	CREATE DATABASE db_companies	1 row(s) affected
3	10:31:44	CREATE TABLE db_companies.companies (company_id CHAR (6) PRIMARY KEY, company_name VARCHAR (100), phone VARCHAR (15), email VARCHAR (100), country VARCHAR (100), website VARCHAR (100));	0 row(s) affected

Se crea la base de datos llamada db_companies para guardar las 4 tablas a las que luego le queremos hacer consultas a través de un diagrama de estrella.

The screenshot shows the MySQL Workbench interface. The left sidebar shows 'SCHEMAS' with 'biblioteca' and 'db_companies' selected. The main area has a code editor with the following SQL script:

```
2
3
4
5 • CREATE TABLE db_companies.companies
6   (
7     company_id CHAR (6) PRIMARY KEY,
8     company_name VARCHAR (100),
9     phone VARCHAR (15),
10    email VARCHAR (100),
11    country VARCHAR (100),
12    website VARCHAR (100));
13
14
15
16
17
```

The 'Output' tab at the bottom shows the results of the execution:

#	Time	Action	Message
1	10:31:35	CREATE TABLE db_companies.companies (company_id CHAR (6) PRIMARY KEY, company_name VARCHAR (100), phone VARCHAR (15), email VARCHAR (100), country VARCHAR (100), website VARCHAR (100));	Error Code: 1049. Unknown database 'db_companies'
2	10:31:41	CREATE DATABASE db_companies	1 row(s) affected
3	10:31:44	CREATE TABLE db_companies.companies (company_id CHAR (6) PRIMARY KEY, company_name VARCHAR (100), phone VARCHAR (15), email VARCHAR (100), country VARCHAR (100), website VARCHAR (100));	0 row(s) affected

Se crea la tabla companies en la data base db_companies con las respectivas características de sus valores. (Con VARCHAR (100) se permiten todo tipo de valores hasta 100 características).

```

10
11
12
13
14 • SHOW VARIABLES LIKE 'secure_file_priv';
15
16
17
18

```

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

Variable_name	Value
secure_file_priv	C:\ProgramData\MySQL\MySQL Server 8.0\Upl...

Result 9 ×

Output

Action Output

#	Time	Action	Message
30	11:31:33	SELECT * FROM companies LIMIT 0, 50000	100 row(s) returned
31	11:32:03	CREATE DATABASE db_companies	Error Code: 1007. Can't create database 'db_com...' (errno: 13)
32	11:37:08	SHOW VARIABLES LIKE 'secure_file_priv'	1 row(s) returned

Para poder acceder a la carpeta que contiene el archivo de companies.csv se ha de crear un acceso seguro a través del comando secure_file_priv. Aquí se copia la ruta de acceso y se abre en carpetas, luego se guardan ahí los archivos a los que posteriormente vamos a querer acceder.

Después se cargan estos datos, a través de esta ruta de acceso:

SCHEMAS

Filter objects

db_companies

- Tables
 - companies
 - Columns
 - company_id
 - company_name
 - phone
 - email
 - country
 - website
- Indexes
- Foreign Keys

Administration Schemas

Information

Schema: db_companies

Object Info Session

```

15
16
17 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv"
18 INTO TABLE db_companies.companies
19 FIELDS TERMINATED BY ','
20 ENCLOSED BY ''
21 IGNORE 1 LINES;
22
23 • SELECT * FROM companies;

```

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

company_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.co...
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.co...
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.co...
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.co.uk	Germany	https://cnn.com/use...

companies 3 ×

Output

Action Output

#	Time	Action	Message
3	11:40:52	SHOW VARIABLES LIKE 'secure_file_priv'	1 row(s) returned
4	11:40:55	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/co..."	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0
5	11:40:59	SELECT * FROM companies LIMIT 0, 50000	100 row(s) returned

Se cargan adentro de la base de datos que hemos creado previamente y se establece la manera en la que queremos que Sql lea los valores contenidos en este archivo.

FIELDS TERMINATED BY ',' indica que los campos están terminados en coma (,).

ENCLOSED BY "" indica que otros están rodeados de comillas. Esto se establece para que a la hora de leer el archivo no se guarden los registros de manera incorrecta.

IGNORE 1 LINES indica que se ignore la primera línea de datos ya que esta contiene el nombre de las columnas y no los registros.

Despues leemos lo que contiene la nueva tabla companies con SELECT.

Estos pasos se repiten con las otras tablas que queremos insertar en la base de datos db_companies.

La **tabla user** incluye tres datasets de diferentes users de diferentes países:

The screenshot shows the MySQL Workbench interface with two main panes. The left pane displays the database schema for 'db_companies'. It shows the 'user' table with columns: id (int PK), name (varchar(100)), surname (varchar(100)), phone (varchar(100)), email (varchar(100)), birth_date (varchar(100)), country (varchar(100)), city (varchar(100)), postal_code (varchar(100)), and address (varchar(100)). The right pane shows the SQL editor and its output.

Top SQL Editor Output:

```
26
27
28
29
30
31 • CREATE TABLE db_companies.user
32   (id INT PRIMARY KEY,
33    name VARCHAR (100),
34    surname VARCHAR (100),
35    phone VARCHAR (100),
36    email VARCHAR (100),
37    birth_date VARCHAR (100),
38    country VARCHAR (100),
39    city VARCHAR (100),
40    postal_code VARCHAR (100),
41    address VARCHAR (100));
```

Action Output:

#	Time	Action	Message
1	13:03:20	CREATE TABLE db_companies.user (id INT PRIMARY KEY, name VARCHAR (1...)	0 row(s) affected

Bottom SQL Editor Output:

```
43
44 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv"
45   INTO TABLE db_companies.user
46   FIELDS TERMINATED BY ','
47   ENCLOSED BY ""
48   LINES TERMINATED BY '\r\n'
49   IGNORE 1 LINES;
50
51 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv"
52   INTO TABLE db_companies.user
53   FIELDS TERMINATED BY ','
54   OPTIONALLY ENCLOSED BY ""
55   LINES TERMINATED BY '\r\n'
56   IGNORE 1 LINES;
57
58 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv"
59   INTO TABLE db_companies.user
60   FIELDS TERMINATED BY ','
61   OPTIONALLY ENCLOSED BY ""
62   LINES TERMINATED BY '\r\n'
63   IGNORE 1 LINES;
```

Action Output:

#	Time	Action	Message
2	12:54:40	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/use..."	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 1
3	12:54:42	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/use..."	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 1
4	12:54:44	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/use..."	150 row(s) affected Records: 150 Deleted: 0 Skipped: 1

Aquí tuvimos que agregar la línea

LINES TERMINATED BY '\r\n', ya que los archivos csv proporcionados no culminan cada fila con una cierta estructura. Algunas terminan con comillas ("), otras con punto(.) Otras vacías, por lo cual hay que señalarle a sql que cada vez que encuentre un salto de línea (\n) → Carácter de salto de línea, se observa el siguiente registro.

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree shows the 'db_companies' schema selected. Under 'Tables', there are 'companies' and 'user'. The 'user' table has columns: id (int PK), name (varchar(100)), surname (varchar(100)), phone (varchar(100)), email (varchar(100)), birth_date (varchar(100)), country (varchar(100)), city (varchar(100)), postal_code (varchar(100)), and address (varchar(100)). The main pane displays the SQL query:

```
59
60 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv"
61 INTO TABLE db_companies.user
62 FIELDS TERMINATED BY ','
63 OPTIONALLY ENCLOSED BY ""
64 LINES TERMINATED BY '\r\n'
65 IGNORE 1 LINES;
66
67 • SELECT *
68 FROM user;
```

Below the query is a 'Result Grid' showing four rows of data:

	id	name	surname	phone	email	birth_date	country	city	pc
▶	1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonmail.edu	Nov 17, 1985	United States	Lowell	73
▶	2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	Aug 23, 1992	United States	Des Moines	59
▶	3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.org	Apr 29, 1998	United States	Columbus	56
▶	4	Howard	Stafford	1-411-740-3269	ornare.egestas@cloud.edu	Feb 18, 1989	United States	Kailua	77

The 'Output' pane shows the log of actions:

#	Time	Action	Message
3	13:04:22	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/us...	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0
4	13:04:23	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/us...	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0
5	13:04:25	SELECT * FROM user LIMIT 0, 50000	275 row(s) returned

Se ve que todos los registros están ordenados según el numero de la Primary Key. El datatype de id de user es INT para poder garantizar este orden.

La tabla **credit_cards**:

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree shows the 'db_companies' schema selected. Under 'Tables', there are 'companies', 'user', and 'credit_cards'. The 'credit_cards' table has columns: id (varchar(100) PRIMARY KEY), user_id (int), iban (varchar(100)), pan (varchar(100)), pin (varchar(100)), cvv (int), track1 (varchar(100)), track2 (varchar(100)), and expiring_date (varchar(100)). The main pane displays the SQL query:

```
61
62
63
64
65 • CREATE TABLE db_companies.credit_cards
66 (id VARCHAR (100) PRIMARY KEY,
67 user_id INT,
68 iban VARCHAR (100),
69 pan VARCHAR (100),
70 pin VARCHAR (100),
71 cvv INT,
72 track1 VARCHAR (100),
73 track2 VARCHAR (100),
74 expiring_date VARCHAR (100));
```

The 'Output' pane shows the log of actions:

#	Time	Action	Message
1	13:45:14	CREATE TABLE db_companies.credit_cards (id VARCHAR (100) PRIMARY KEY...,	0 row(s) affected

Aquí cabe destacar que la columna user_id tiene la misma característica que la Primary Key de user para posteriormente poder hacer conexiones. (INT)

SCHEMAS

```

79
80
81 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv"
82 INTO TABLE db_companies.credit_cards
83 FIELDS TERMINATED BY ','
84 IGNORE 1 LINES;
85
86 • SELECT*
87 FROM credit_cards;
88

```

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

	id	user_id	iban	pan	pin	cvv	track1
▶	CcU-2938	275	TR301950312213576817638661	5424465566813633	3257	984	%88383712448554646^WovsxejDpw
	CcU-2945	274	DO26854763748537475216568689	5142423821948828	9080	887	%84621311609958661^UftuyfSeimx
	CcU-2952	273	BG45IVQL52710525608255	4556 453 55 5287	4598	438	%B2183285104307501^CdyytUJwx
	CcU-2959	272	CR7242477244335841535	372461377349375	3583	667	%B7281111956795320^XocddijBcke

credit_cards 24 x | Apply

Output

Action Output

#	Time	Action	Message
1	13:45:14	CREATE TABLE db_companies.credit_cards (id VARCHAR (100) PRIMARY KEY...	0 row(s) affected
2	13:46:17	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/cred...	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0
3	13:46:19	SELECT* FROM credit_cards LIMIT 0, 50000	275 row(s) returned

Object Info Session

La tabla transactions:

SCHEMAS

```

93 • CREATE TABLE db_companies.transactions
94   (id varchar(100) Primary Key,
95    card_id varchar(15),
96    business_id varchar(20),
97    timestamp timestamp,
98    amount decimal(10,2),
99    declined tinyint(1),
100   product_ids varchar(100),
101   user_id INT,
102   lat float,
103   longitude float);
104
105
106
107
108
109
110
111 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv"
112 INTO TABLE db_companies.transactions

```

Output

Action Output

#	Time	Action	Message
1	10:34:58	CREATE TABLE db_companies.transactions (id varchar(100) Primary Key, card_id...	0 row(s) affected, 1 warning(s): 1681 Integer display width

Object Info Session

Aquí opto por copiar el estilo de la tabla de transactions de otra data base.

Contiene valores como timestamp, con el datatype **timestamp** que es una hora y una fecha formateada.

Contiene un valor monetario (amount) a la que se le asigna **decimal (10,2)** lo que permite ingresar un total de 10 números con dos números después del decimal.

El datatype **tinyint(1)** es un boolean que almacena registros en función de que si son verdaderos o falsos.

Float permite valores numéricos con varios decimales cuando no se requiere una mayor precisión.

```

111 •  LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv"
112      INTO TABLE db_companies.transactions
113      FIELDS TERMINATED BY ','
114      LINES TERMINATED BY '\r\n'
115      IGNORE 1 LINES;
116
117 •  SELECT *
118      FROM transactions;
119
120

```

id	card_id	business_id	timestamp	amount	declined	product_ids	user_id	type
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	71, 1, 19	92	1
0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	17	1
063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	2022-01-06 21:25:27	92.61	0	47, 67, 31, 5	25	1
0668296C-CDB9-A883-76BC-2E4C44F8C8AE	CcU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	26	1

transactions 26 x

Output

Action Output

#	Time	Action	Message
1	13:58:03	CREATE TABLE db_companies.transactions (id varchar(100) Primary Key, card_id...	0 row(s) affected, 1 warning(s): 1681 Integer display width is too small; column was truncated
2	13:59:31	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv" INTO TABLE db_companies.transactions	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0
3	13:59:34	SELECT * FROM transactions LIMIT 0, 50000	587 row(s) returned

Se le agregan las Foreign Keys para obtener el diagrama de estrella:

Schemas

- biblioteca
- db_companies
- Tables
- companies
- credit_cards
- transactions
- user
- Views
- Stored Procedures
- Functions
- hospitales
- olympics
- sakila
- sys

Table: transactions

Columns:

id	varchar(100)	PK
card_id	varchar(15)	
business_id	varchar(20)	
timestamp	timestamp	
amount	decimal(10,2)	
declined	tinyint(1)	
product_ids	varchar(100)	
user_id	int	
type	float	

Object Info Session

```

105
106 •  ALTER TABLE transactions
107     ADD CONSTRAINT `transactionycredit`
108     FOREIGN KEY (`card_id`)
109     REFERENCES `credit_cards`(`id`);
110
111 •  ALTER TABLE transactions
112     ADD CONSTRAINT `transactionycompany`
113     FOREIGN KEY (`business_id`)
114     REFERENCES `companies`(`company_id`);
115
116 •  ALTER TABLE transactions
117     ADD CONSTRAINT `transactionyuser`
118     FOREIGN KEY (`user_id`)
119     REFERENCES `user`(`id`);

120

```

Action Output

#	Time	Action	Message
1	10:34:00	ALTER TABLE transactions ADD CONSTRAINT transactionycredit FOREIGN KEY (card_id) REFERENCES credit_cards(id);	587 row(s) affected Records: 587 Duplicates: 0 Warning: 0
2	10:35:19	ALTER TABLE transactions ADD CONSTRAINT transactionycompany FOREIGN KEY (business_id) REFERENCES companies(company_id);	Error Code: 1146. Table 'db_companies.transactions' doesn't have a primary key
3	10:35:26	ALTER TABLE transactions ADD CONSTRAINT transactionycompany FOREIGN KEY (business_id) REFERENCES companies(company_id);	587 row(s) affected Records: 587 Duplicates: 0 Warning: 0
4	10:36:28	ALTER TABLE transactions ADD CONSTRAINT transactionyuser FOREIGN KEY (user_id) REFERENCES user(id);	587 row(s) affected Records: 587 Duplicates: 0 Warning: 0

Luego miramos los datos y regularizamos las columnas que contienen las fechas para poder trabajar con ellas:

```

123
124 • ALTER TABLE transactions
125     MODIFY COLUMN timestamp date;
126
127 • SELECT *
128     FROM transactions;
129
130
131

```

Result Grid

	id	card_id	business_id	timestamp	amount	declined	product_ids	user_id
▶	02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	2021-08-28	466.92	0	71, 1, 19	92
	0466A42E-47CF-8D24-FD01-C0B689713128	CcU-4219	b-2302	2021-07-26	49.53	0	47, 97, 43	170
	063FBA79-99EC-66FB-29F7-25726D1764A5	CcU-2987	b-2250	2022-01-06	92.61	0	47, 67, 31, 5	275
	0668296C-CDB9-A883-76BC-2E4C44F8C8AE	CcU-3743	b-2618	2022-01-26	394.18	0	89, 83, 79	265

transactions 9 ×

Action Output

#	Time	Action	Message
✖	6 11:40:50	UPDATE transactions SET timestamp = STR_TO_DATE(timestamp, '%m/%d/%Y');	Error Code: 1175. You are using safe update mode.
✓	7 11:41:16	ALTER TABLE transactions MODIFY COLUMN timestamp date	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓	8 11:41:39	SELECT * FROM transactions LIMIT 0, 50000	587 row(s) returned

SCHEMAS

- Filter objects
- biblioteca
- db_companies
 - Tables
 - companies
 - credit_cards
 - estado_tarjetas
 - productos
 - transactions
 - transactions_products
 - user

Administration Schemas

Information

Table: credit_cards

Columns:

id	varchar(100) PK
user_id	int
iban	varchar(100)
pan	varchar(100)
pin	varchar(100)
cvv	int
track1	varchar(100)
track2	varchar(100)
expiring_date	varchar(100)

Result Grid

	pin	cvv	track1	track2	expiring_date	
▶	5813633	3257	984	%B8383712448554646^WovsxejDpwiev^8604...	%B7653863056044187=800716333673	2022-10-30
	1948828	9080	887	%B4621311609958611^UfufyfsSelmnx^06106...	%B4149568437843501=51071403301	2023-08-24
	5 5287	4598	438	%B2183285104307501^CddytytcUxwfdq^5907...	%B6778580257827162=69068597400?	2021-06-29
	349375	3583	667	%B7281111956795320^XocddijBcked^09016...	%B4246154489281853=280522391678	2023-02-24

credit_cards 1 ×

Action Output

#	Time	Action	Message
✓	1 11:49:56	UPDATE credit_cards SET expiring_date = STR_TO_DATE(expiring_date, '%m/%d/%Y');	275 row(s) affected Rows matched: 275 Changed: 275
✓	2 11:49:58	SELECT * FROM credit_cards LIMIT 0, 50000	275 row(s) returned

Luego de formatear la manera en la que se escribieron las fechas, modificamos el datatype en la que se guardaron para que sean fechas

SCHEMAS

Filter objects

- companies
- credit_cards
- Columns
 - id
 - user_id
 - iban
 - pan
 - pin
 - cvv
 - track1

Administration Schemas Information

Table: credit_cards

Columns:

id	varchar(100) PK
user_id	int
iban	varchar(100)
pan	varchar(100)
pin	varchar(100)
cvv	int
track1	varchar(100)
track2	varchar(100)
expiring_date	date

127
128
129
130
131
132
133 • ALTER TABLE credit_cards
134 MODIFY COLUMN expiring_date date;
135
136
137
138
139
140
141
142
143

Output:

Action Output

#	Time	Action	Message
2	11:54:18	UPDATE USER SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y')	275 row(s) affected Rows matched: 275 Changed: 275 W...
3	11:54:22	SELECT * FROM user LIMIT 0, 50000	275 row(s) returned
4	11:56:23	ALTER TABLE credit_cards MODIFY COLUMN expiring_date date	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

Object Info Session

131
132
133 • UPDATE USER
134 SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y');
135
136 • SELECT *
137 FROM user;
138
139

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

	name	phone	email	birth_date	country	city	postal_code	address
▶	ble	1-282-581-0551	interdum.enim@protonmail.edu	1985-11-17	United States	Lowell	73544	348-7818 Sa...
	onnell	(718) 257-2412	integer.vitae.nibh@protonmail.org	1992-08-23	United States	Des Moines	59464	903 Sit Ave...
	ison	(522) 598-1365	interdum.feugiat@aol.org	1998-04-29	United States	Columbus	56518	736-2063 Tel...
	ford	1-411-740-3269	ornare.egestas@icloud.edu	1989-02-18	United States	Kailua	77417	Ap #545-224...

user 2 ×

Output:

Action Output

#	Time	Action	Message
1	11:54:05	UPDATE USER SET birth_date = STR_TO_DATE(birth_date, '%b %d %Y')	Error Code: 1411. Incorrect datetime value:...
2	11:54:18	UPDATE USER SET birth_date = STR_TO_DATE(birth_date, '%b %d, %Y')	275 row(s) affected Rows matched: 275 Ch...
3	11:54:22	SELECT * FROM user LIMIT 0, 50000	275 row(s) returned

Schemas

Filter objects

db_companies

- Tables
 - companies
 - credit_cards
 - estado_tarjetas
 - productos
 - transactions
 - transactions_products
 - user
- Views

Administration Schemas

Information

Table: user

Columns:

id	int PK
name	varchar(100)
surname	varchar(100)
phone	varchar(100)
email	varchar(100)
birth_date	date
country	varchar(100)
city	varchar(100)
postal_code	varchar(100)
address	varchar(100)

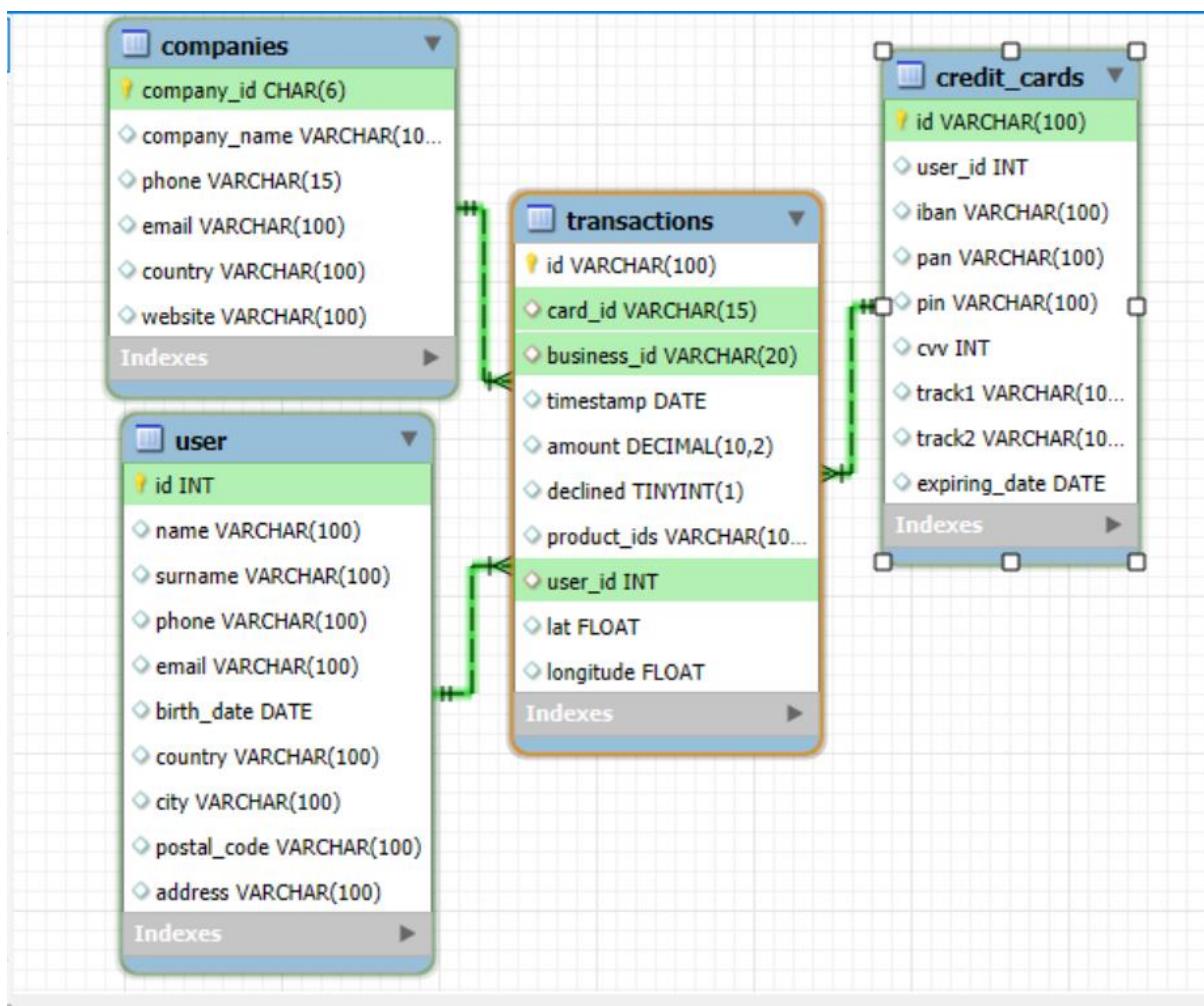
137
138
139
140
141
142
143
144
145 • ALTER TABLE user
146 MODIFY COLUMN birth_date date;
147
148
149
150
151
152
153

Action Output

#	Time	Action	Message
3	11:54:22	SELECT * FROM user LIMIT 0, 50000	275 row(s) returned
4	11:56:23	ALTER TABLE credit_cards MODIFY COLUMN expiring_date date	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0
5	11:59:04	ALTER TABLE user MODIFY COLUMN birth_date date	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

Object Info Session

Llegamos a este diagrama en la que la tabla de transacciones se sitúa en la mitad y las otras tablas de dimensiones están alrededor:



- Exercici 1

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

```
155
156 •  SELECT *
157   FROM user
158   WHERE user.id IN ( SELECT user_id
159     FROM transactions
160     WHERE declined = 0
161     GROUP BY user_id
162     HAVING COUNT(id) > 30);
163
```

id	name	surname	phone	email	birth_date	country	city	postal_code
92	Lynn	Riddle	1-387-885-4057	vitae.aliquet@outlook.edu	1984-09-21	United States	Bozeman	61871
267	Ocean	Nelson	079-481-2745	aenean@yahoo.com	1991-12-26	Canada	Charlottetown	85X 3P4
272	Hedwig	Gilbert	064-204-8788	sem.eget@icloud.edu	1991-04-16	Canada	Tuktoyaktuk	Q4C 3G7

user 6 x

Output

Action Output

#	Time	Action	Message
1	12:31:55	SELECT * FROM user WHERE user.id IN (SELECT user_id FROM transaction...)	3 row(s) returned

Aquí podemos ver que hay 3 usuarios que han efectuado (declined = 0) más que 30 transacciones.

- Exercici 2

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

```
167
168 •  WITH laempresa AS (SELECT *
169   FROM companies
170   WHERE company_name = "Donec Ltd")
171   SELECT company_name, business_id, round(avg(amount),2) as promedio, card_id, iban
172   FROM transactions
173   JOIN laempresa ON laempresa.company_id = transactions.business_id
174   JOIN credit_cards ON credit_cards.id = transactions.card_id
175   WHERE declined = 0
```

company_name	business_id	promedio	card_id	iban
Donec Ltd	b-2242	42.82	CcU-2973	PT87806228135092429456346

Result 7 x

Output

Action Output

#	Time	Action	Message
1	12:31:55	SELECT * FROM user WHERE user.id IN (SELECT user_id FROM transaction...)	3 row(s) returned
2	12:32:30	WITH laempresa AS (SELECT * FROM companies WHERE company_name = "...")	1 row(s) returned

Aqui podemos ver que solo 1 credit_card ha efectuado (declined=0) una transaccion a la empresa Donec Ltd.

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:

Exercici 1

Quantes targetes estan actives?

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree shows the 'credit_cards' schema with a table named 'estado_tarjetas'. The 'Columns' section of this table contains two columns: 'card_id' (varchar(15)) and 'estado' (varchar(8)). In the center, the SQL editor displays the following code:

```
156
157 • CREATE TABLE estado_tarjetas AS
158     SELECT card_id,
159         CASE
160             WHEN COUNT(*) = 3 AND SUM(declined = 1) = 3 THEN 'Inactiva'
161             ELSE 'Activa'
162         END AS estado
163     FROM (SELECT card_id, declined,
164                 ROW_NUMBER() OVER (PARTITION BY card_id ORDER BY timestamp DESC) AS rn
165             FROM transactions
166         ) AS ultimas
167     WHERE rn <= 3
168     GROUP BY card_id;
169
170
171
172
173
174
```

Below the code, the 'Output' pane shows the results of the query execution. The first message is an error: 'Error Code: 1054. Unknown column 'credit_id' in 'window''. The second message is a success: '275 row(s) affected Records: 275 Duplicates: 0 Warnings'.

Aqui lo primero que hacemos es **CREATE** la tabla **estado_tarjeta** que adopta los valores de las consultas que siguen. Esta tabla contendrá card_id y asignara una etiqueta de **estado** a cada card_id según si fueron declinadas las últimas tres veces o no.

(WHEN **COUNT(*)** = 3 AND **SUM(declined = 1)** = 3 THEN '**Inactiva**'

ELSE 'Activa'

END AS estado)

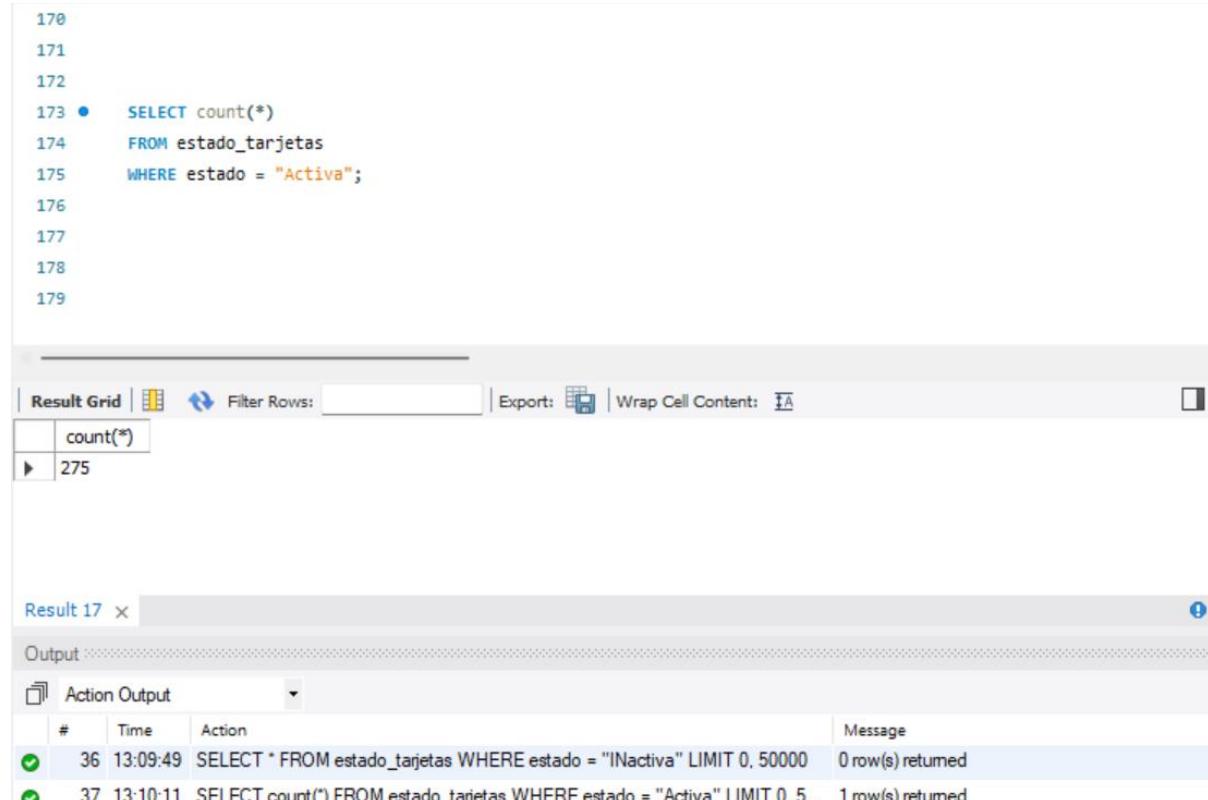
Posteriormente se agrega con la segunda subconsulta de la que se quiere sacar los datos. En esta solo se tienen en cuenta las últimas tres filas según su timestamp.

```

FROM (SELECT card_id, declined, ROW_NUMBER() OVER (PARTITION BY card_id
ORDER BY timestamp DESC) AS rn
FROM transactions ) AS ultimas
WHERE rn <= 3
GROUP BY card_id;

```

Solo se busca las ultimas rows que tienen que tener el numero 1, 2 y 3. **WHERE rn <= 3**



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

```

170
171
172
173 •   SELECT count(*)
174     FROM estado_tarjetas
175     WHERE estado = "Activa";
176
177
178
179

```

Below the code editor is a result grid window titled "Result Grid". It contains one row of data:

	count(*)
▶	275

At the bottom of the interface, there is a "Result 17" tab showing the execution log:

Output

Action Output

#	Time	Action	Message
36	13:09:49	SELECT * FROM estado_tarjetas WHERE estado = "INactiva" LIMIT 0, 50000	0 row(s) returned
37	13:10:11	SELECT count(*) FROM estado_tarjetas WHERE estado = "Activa" LIMIT 0, 5...	1 row(s) returned

275 tarjetas están activas. (todas)

También se crea una Foreign Key con la tabla de credit_cards:

```
167  
168  
169  
170  
171  
172  
173  
174 • ALTER TABLE estado_tarjetas  
175     ADD CONSTRAINT `creditcardyestado`  
176     FOREIGN KEY (`card_id`)  
177     REFERENCES `credit_cards` (`id`);  
178  
179  
180  
181  
182  
183  
184
```

Output			
Action Output			Message
#	Time	Action	
✗	151 12:58:24	CREATE TABLE estado_tarjetas AS SELECT card_id, CASE WHEN COUNT(*)...	Error Code: 1064. You have an error in your SQL syntax;
✓	152 13:01:53	ALTER TABLE estado_tarjetas ADD CONSTRAINT `creditcardyestado` FOREIGN ...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0



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:

Exercici 1

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

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree view shows the 'db_companies' schema selected, containing tables like 'companies', 'credit_cards', 'estado_tarjetas', 'productos', 'transactions', and 'user'. A 'Views' and 'Stored Procedures' section is also present. The 'Information' tab is selected. In the center, the 'Table: productos' section shows the table definition with columns: id (int PK), product_name (varchar(100)), price (decimal(10,2)), colour (varchar(100)), weight (decimal(10,1)), and warehouse_id (varchar(100)). Below this, the 'Columns' section lists the same columns with their types. On the right, the 'Output' pane shows the SQL code for creating the table and the log of actions, including the creation of the table and its subsequent drop.

```
CREATE TABLE productos
(
    id INT PRIMARY KEY,
    product_name VARCHAR (100),
    price DECIMAL (10,2),
    colour VARCHAR (100),
    weight DECIMAL (10,1),
    warehouse_id VARCHAR (100));

```

#	Time	Action	Message
117	11:53:36	DROP TABLE 'db_companies';'productos'	0 row(s) affected
118	11:53:43	CREATE TABLE productos (id INT PRIMARY KEY, product_name VARCHAR (...)	0 row(s) affected

```

191
192 • LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv"
193     INTO TABLE db_companies.productos
194     FIELDS TERMINATED BY ","
195     LINES TERMINATED BY '\n'
196     IGNORE 1 LINES
197     (id, product_name, @precio, colour, weight, warehouse_id)
198     SET price = REPLACE(@precio, '$', '');
199
200 • SELECT *
201     FROM productos;

```

Result Grid

	id	product_name	price	colour	weight	warehouse_id
▶	1	Direwolf Stannis	161.11	#7c7c7c	1.0	WH-4
	2	Tarly Stark	9.24	#919191	2.0	WH-3
	3	duel tourney Lannister	171.13	#d8d8d8	1.5	WH-2
	4	warden south duel	71.89	#111111	3.0	WH-1
	5	skywalker ewok	171.22	#dbdbdb	3.2	WH-0

productos 48

Output

Action Output

#	Time	Action	Message
119	11:54:42	LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/pr...	100 row(s) affected Records: 100 Deleted: 0
120	11:54:45	SELECT * FROM productos LIMIT 0, 50000	100 row(s) returned

Primero se crea la tabla en la que se quieren insertar los datos de productos.csv.

Luego se insertan los datos teniendo en cuenta que el precio que esta guardado en el archivo con \$, se reemplaza con un espacio en blanco para que no genere problemas.

(id, product_name, @precio, colour, weight, warehouse_id)

SET price = REPLACE(@precio, '\$', '');

Se verifica que se hayan insertado correctamente.

SCHEMAS

- biblioteca
- db_companies
 - Tables
 - companies
 - credit_cards
 - estado_tarjetas
 - productos
 - transactions
 - transactions_products
 - user
 - Views

Administration Schemas

Table: transactions_products

Columns:

transaction_id	varchar(100)
producto_id	int

199
200
201
202
203
204
205 • CREATE TABLE transactions_products
(transaction_id VARCHAR(100),
206 producto_id INT,
207 FOREIGN KEY (transaction_id) REFERENCES transactions(id),
208 FOREIGN KEY (producto_id) REFERENCES productos(id)
209);
210
211
212
213
214
215
216

Output

Action Output

#	Time	Action	Message
120	11:54:45	SELECT * FROM productos LIMIT 0, 50000	100 row(s) returned
121	11:57:10	CREATE TABLE transactions_products (transaction_id VARCHAR(100), produ...	0 row(s) affected

Object Info Session

Se crea una nueva tabla intermedia para guardar las transacciones separadas por sus product_ids. Ya que en la tabla original de transacciones los product_ids están

guardadas en una secuencia separada por ,.

```

197
198
199
200
201 •   SELECT *
202     FROM transactions;
203
204
205
206

```

The screenshot shows the MySQL Workbench interface. At the top, there is a 'Result Grid' window displaying the results of a SELECT query on the 'transactions' table. The columns are: card_id, business_id, timestamp, amount, declined, product_ids, user_id, lat, and longitude. Below this is an 'Output' window showing the creation of a temporary table 'transactions_products' and a SELECT query returning 587 rows.

	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
▶	2986D3B02	CcU-2938	b-2362	2021-08-28 23:42:24	466.92	0	71, 1, 19	92	81.9185 -12.5276
	689713128	CcU-4219	b-2302	2021-07-26 07:29:18	49.53	0	47, 97, 43	170	-43.9695 -117.525
	26D1764A5	CcU-2987	b-2250	2022-01-06 21:25:27	92.61	0	47, 67, 31, 5	275	-81.2227 -129.05
	C44F8C8AE	CcU-3743	b-2618	2022-01-26 02:07:14	394.18	0	89, 83, 79	265	-34.3593 -100.556

transactions 49 ×

Output

Action Output

#	Time	Action	Message
✓	121	11:57:10 CREATE TABLE transactions_products (transaction_id VARCHAR(100), produ...	0 row(s) affected
✓	122	11:59:34 SELECT * FROM transactions LIMIT 0, 50000	587 row(s) returned

Para que se separen los product_ids hay que aplicar una función:

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree shows the 'transactions_products' table selected. In the center, a code editor displays the SQL script for creating the table and inserting data. Below the code editor is a 'Result Grid' window showing the data from the 'transactions_products' table. The 'Output' window at the bottom shows the execution of the script.

SCHEMAS

Table: transactions_products

Columns:

- transaction_id varchar(100)
- producto_id int

Result Grid

transaction_id	producto_id
02C6201E-D90A-1859-B4EE-88D2986D3B02	1
02C6201E-D90A-1859-B4EE-88D2986D3B02	19
02C6201E-D90A-1859-B4EE-88D2986D3B02	71
0466A42E-47CF-8D24-FD01-C0B689713128	43
0466A42E-47CF-8D24-FD01-C0B689713128	47

Output

Action Output

#	Time	Action	Message
✓	2	12:32:30 WITH laempresa AS (SELECT * FROM companies WHERE company_name = ...)	1 row(s) returned
✓	3	12:35:49 SELECT * FROM transactions_products LIMIT 0, 50000	1236 row(s) returned

Insertamos directamente en la tabla posteriormente creada cada transaccion y cada producto_id.

SELECT transactions.id AS transaction_id,

```

productos.id AS producto_id
FROM transactions
JOIN productos
ON FIND_IN_SET(productos.id,
REPLACE(transactions.product_ids, ',', ''))

WHERE declined = 0;

```

Aquí el **JOIN** se hace para que la tabla resultante contenga valores contrastados a través de los productos.id. Si un numero de productos.id está contenida en transactions los encuentra en la columna y cadena de valores product_id de transactions (**FIND_IN_SET**).

Sin embargo, para que SQL pueda leer la cadena de valores hay que quitarles el espacio después de cada coma. (**REPLACE(transactions.product_ids, ',', '')**)

La tabla resultante es una que contiene varias veces 1 transaction.id (transaction_id) con varios valores de productos separados, si es que ese producto estaba inicialmente en la tabla de productos.

```

212
213
214
215
216
217 •   SELECT producto_id, count(transaction_id)
218   FROM transactions_products
219   GROUP BY producto_id
220   ORDER BY producto_id ASC;
221

```

The screenshot shows a MySQL query results interface. At the top, there are several lines of code with line numbers 212 through 221. Line 217 is highlighted with a blue dot. Below the code is a 'Result Grid' table with two columns: 'producto_id' and 'count(transaction_id)'. The data is as follows:

producto_id	count(transaction_id)
1	51
2	56
3	43
5	42
7	44

Below the table is a 'Result 60 x' section with an 'Output' tab. Under 'Action Output', there are two log entries:

#	Time	Action	Message
149	12:35:16	SELECT * FROM transactions_products LIMIT 0, 50000	1236 row(s) returned
150	12:50:31	SELECT producto_id, count(transaction_id) FROM transactions_products GRO...	26 row(s) returned

Luego se cuentan las transaction_id de la nueva tabla y se ordena por producto_id. Eso cuenta cuantas veces aparece una transaction_id por cada numero (producto_id).