

SPRINT 3 -- Tarea S3.01. Manipulación de tablas

Nivel 1

Ejercicio 1

Tu tarea es diseñar y crear una tabla llamada "credit_card" que almacene detalles cruciales sobre las tarjetas de crédito. La nueva tabla debe ser capaz de identificar de forma única cada tarjeta y establecer una relación adecuada con las otras dos tablas ("transaction" y "company"). Después de crear la tabla será necesario que ingreses la información del documento denominado "datos_introducir_credit". Recuerda mostrar el diagrama y realizar una breve descripción del mismo.

```

15 • use transactions;
16
17 • CREATE TABLE IF NOT EXISTS credit_card (
18     id VARCHAR(100) PRIMARY KEY,
19     iban VARCHAR(100),
20     pan VARCHAR(100),
21     pin VARCHAR(100),
22     cvv VARCHAR(100),
23     expiring_date VARCHAR(100));
24
25 -- select * from credit_card; -- 275 rows
26

```

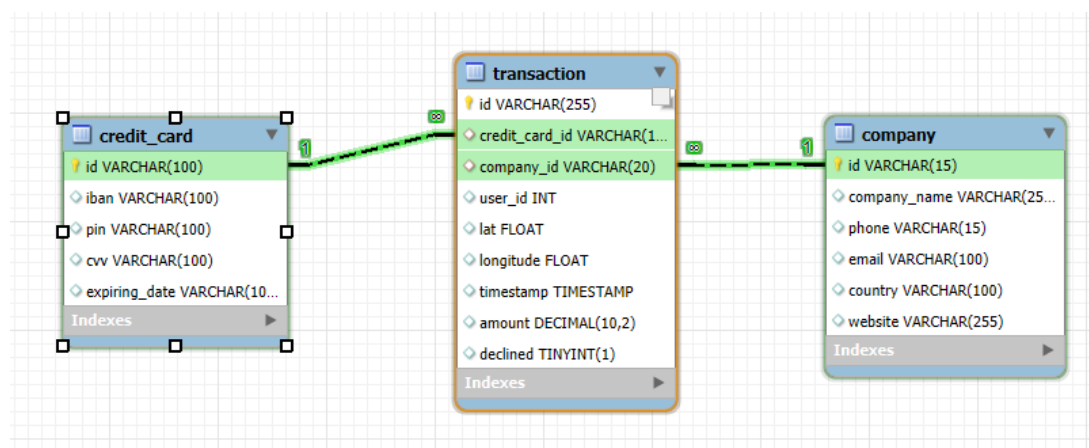
id	iban	pan	pin	cvv	expiring_date
CcU-2938	TR301950312213576817638661	5424465566813633	3257 984	10/30/22	
CcU-2945	DO26854763748537475216568689	5142423821948828	9080 887	08/24/23	
CcU-2952	BG45IVQL52710525608255	4556 453 55 5287	4598 438	06/29/21	
CcU-2959	CR7242477244335841535	372461377349375	3583 667	02/24/23	
CcU-2966	BG72LKTQ15627628377363	448566 886747 7265	4900 130	10/29/24	

credit_card 2 x Apply

Output

#	Time	Action	Message	Duration / I
906	12:10:27	INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ('CcU-...	1 row(s) affected	0.000 sec
907	12:10:38	select * from credit_card LIMIT 0, 1000	275 row(s) returned	0.000 sec

Diagrama de la BBDD transactions



Breve descripción

Este modelo está compuesto por **3 tablas**:

- **transaction**: tabla de Hechos

- **credit_card y company:** tablas de **Dimensiones**

En conjunto forman un **modelo** de tipo **Estrella**.

Campos y tipos de datos de cada Tabla:

transaction

<u>id</u>	varchar(255) PK
credit_card_id	varchar(15)
company_id	varchar(20)
lat	float
longitude	float
timestamp	timestamp
amount	decimal(10,2)
declined	tinyint(1)

credit_card

<u>id</u>	varchar(100) PK
iban	varchar(100)
pin	varchar(100)
cvv	varchar(100)
expiring_date	varchar(100)

company

<u>id</u>	varchar(15) PK
company_name	varchar(255)
phone	varchar(15)
email	varchar(100)
country	varchar(100)
website	varchar(255)

Relaciones entre tablas del modelo

- **company-transaction**
Relación **uno a muchos (1:N)** a través de:
 - o **id** es la **clave primaria (PK)** en la tabla **company**.
 - o **company_id** es la **clave foránea (FK)** en la tabla **transaction**
 - o Cada registro en **transaction** está asociado a **una única empresa** pero una misma empresa puede tener **muchas transacciones** (registros)
- **credit_card-transaction**
Relación **uno a muchos (1:N)** a través de:
 - o **id** es la **clave primaria (PK)** en la tabla **credit_card**.
 - o **credit_card_id** es la **clave foránea (FK)** en la tabla **transaction**

- Cada transacción está asociada a una única tarjeta de crédito pero una tarjeta puede estar vinculada a muchas transacciones.

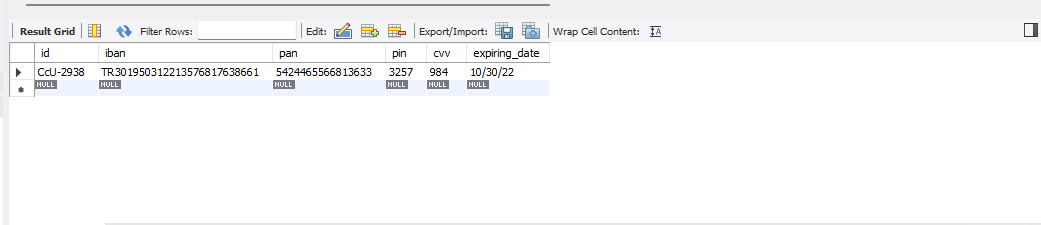
Ejercicio 2

El departamento de Recursos Humanos ha identificado un error en el número de cuenta del usuario con ID CcU-2938. La información que debe mostrarse para este registro es: R323456312213576817699999. Recuerda mostrar que el cambio se realizó.

```

34 -- Ejercicio 2
35 -- El departamento de Recursos Humanos ha identificado un error en el número de cuenta del usuario con ID CcU-2938.
36 -- La información que debe mostrarse para este registro es: R323456312213576817699999. Recuerda mostrar que el cambio se realizó.
37
38 • SELECT * FROM credit_card
39 WHERE id = 'CcU-2938';
40

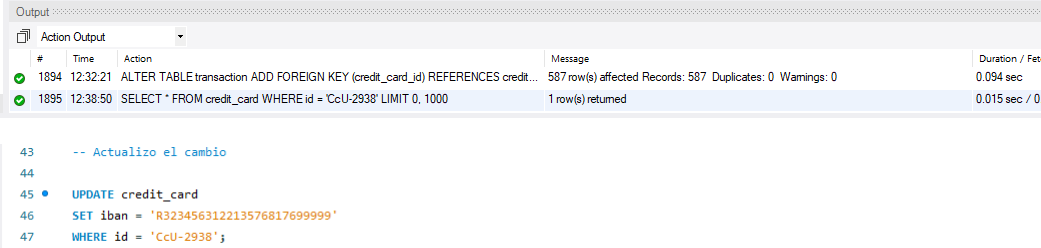
```



```

43 -- Actualizo el cambio
44
45 • UPDATE credit_card
46 SET iban = 'R323456312213576817699999'
47 WHERE id = 'CcU-2938';
48

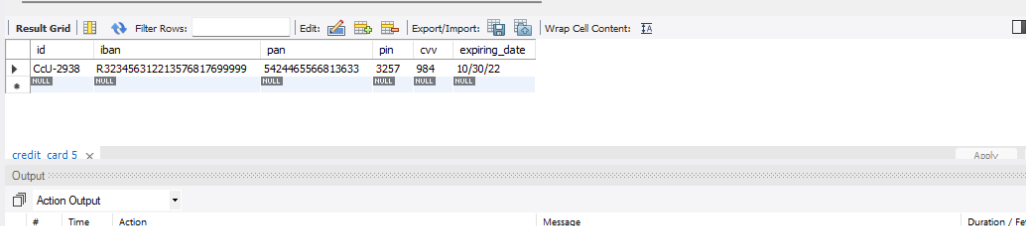
```



```

49 -- Rechequeo que se haya realizado el cambio
50
51 • SELECT * FROM credit_card
52 WHERE id = 'CcU-2938';
53

```



```

1891 12:30:47 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2938', 'R323456312213576817699999', '5424465566813633', '3257', '984', '10/30/22') 1 row(s) affected 0.000 sec
1892 12:30:47 INSERT INTO credit_card (id, iban, pan, pin, cvv, expiring_date) VALUES ( 'CcU-2938', 'R323456312213576817699999', '5424465566813633', '3257', '984', '10/30/22') 1 row(s) affected 0.000 sec
1893 12:31:40 select * from credit_card LIMIT 0, 1000 275 row(s) returned 0.000 sec / 0.
1894 12:32:21 ALTER TABLE transaction ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card (id) ON DELETE CASCADE 587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0 0.094 sec
1895 12:38:50 SELECT * FROM credit_card WHERE id = 'CcU-2938' LIMIT 0, 1000 1 row(s) returned 0.015 sec / 0.
1896 12:44:42 UPDATE credit_card SET iban = 'R323456312213576817699999' WHERE id = 'CcU-2938' 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec
1897 12:45:31 SELECT * FROM credit_card WHERE id = 'CcU-2938' LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.

```

Ejercicio 3

En la tabla "transaction" ingresa un nuevo usuario con la siguiente información:

Id	108B1D1D-5B23-A76C-55EF-C568E49A99DD
----	--------------------------------------

credit_card_id	CcU-9999
company_id	b-9999
user_id	9999
lato	829.999
longitud	-117.999
amunt	111.11
declined	0

```

59 • SELECT *
60 FROM credit_card
61 WHERE id = 'CcU-9999';
62
63 • INSERT INTO credit_card(id)
64 VALUES ('CcU-9999');
65
66 -- | chequeo la insercion
67

```

id	iban	pan	pin	cvv	expiring_date
CcU-9999	NOLE	NOLE	NOLE	NOLE	NOLE
NOLE	NOLE	NOLE	NOLE	NOLE	NOLE

#	Time	Action	Message	Duration / Fet
✓ 1899	12:52:35	select * from transaction where id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD' LI...	0 row(s) returned	0.000 sec / 0
✓ 1900	12:53:38	select * from transaction where credit_card_id = 'CcU-9999' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0
✓ 1901	12:54:00	select * from transaction where credit_card_id = 'CcU-9999' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0
✓ 1902	12:57:17	SELECT * FROM credit_card WHERE id = 'CcU-9999' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0
✓ 1903	12:58:04	INSERT INTO credit_card(id) VALUES ('CcU-9999')	1 row(s) affected	0.000 sec

```

68 -- 2. Chequear si existe y sino insertar en company el id 'b-9999'
69
70 • SELECT *
71 FROM company
72 WHERE id = 'b-9999';
73
74 -- INSERT INTO
75 -- VALUES ;
76

```

id	company_name	phone	email	country	website
NOLE	NOLE	NOLE	NOLE	NOLE	NOLE

#	Time	Action	Message	Duration / Fet
✗ 1904	12:58:20	INSERT INTO credit_card(id) VALUES ('CcU-9999')	Error Code: 1062. Duplicate entry 'CcU-9999' for key 'credit_card.PRIMARY'	0.000 sec
✓ 1905	12:58:29	SELECT * FROM credit_card WHERE id = 'CcU-9999' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0
✓ 1906	13:03:37	SELECT * FROM company WHERE id = 'b-9999' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0
✓ 1907	13:04:13	SELECT * FROM company WHERE id = 'b-9999' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0

```

68 -- 2. Chequear si existe y sino insertar en company el id 'b-9999'
69
70 • SELECT *
71 FROM company
72 WHERE id = 'b-9999';
73
74 • INSERT INTO company (id) -- -- y rechequeo la insercion. OK.
75 VALUES ('b-9999');
76

```

Result Grid

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

company 13 x

Output

Action Output

#	Time	Action	Message	Duration / Fect
1907	13:04:13	SELECT * FROM company WHERE id = 'b-9999' LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.0
1908	13:05:49	INSERT INTO company (id) VALUES 'b-9999'	Error Code: 1064. You have an error in your SQL syntax; check the manual that comes...	0.000 sec
1909	13:06:32	INSERT INTO company (id) VALUES (b-9999)	1 row(s) affected	0.016 sec
1910	13:06:38	SELECT * FROM company WHERE id = 'b-9999' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.0

```

77 -- 3- insert :
78 • INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
79 VALUES ('10881D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0' );
80
81 • SELECT * FROM transaction WHERE id = '10881D1D-5B23-A76C-55EF-C568E49A99DD';
82

```

Result Grid

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	NULL	111.11	0

transaction 14 x

Output

Action Output

#	Time	Action	Message	Duration / Fect
1909	13:06:32	INSERT INTO company (id) VALUES (b-9999)	1 row(s) affected	0.016 sec
1910	13:06:38	SELECT * FROM company WHERE id = 'b-9999' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.0
1911	13:08:03	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amo...	1 row(s) affected	0.016 sec
1912	13:09:00	SELECT * FROM transaction WHERE id = '10881D1D-5B23-A76C-55EF-C568E49A99...	1 row(s) returned	0.000 sec / 0.0

Ejercicio 4

Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit_card. Recuerda mostrar el cambio realizado.

```

84  -- Ejercicio 4 --
85
86  -- Desde recursos humanos te solicitan eliminar la columna "pan" de la tabla credit_card.
87  -- Recuerda mostrar el cambio realizado.
88  --
89  -- 1.Verifico que este el campo 'pan' en la Tabla credit_card original
90
91  • DESCRIBE credit_card;

```

Field	Type	Null	Key	Default	Extra
id	varchar(100)	NO	PRI	NULL	
iban	varchar(100)	YES		NULL	
pan	varchar(100)	YES		NULL	
pin	varchar(100)	YES		NULL	
cvv	varchar(100)	YES		NULL	
expiring_date	varchar(100)	YES		NULL	

Result 16 x

Output

#	Time	Action	Message	Duration / Fets
✓ 1911	13:08:03	INSERT INTO transaction (id, credit_card_id, company_id, user_id, lat, longitude, amo...	1 row(s) affected	0.016 sec
✓ 1912	13:09:00	SELECT * FROM transaction WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99...	1 row(s) returned	0.000 sec / 0
✓ 1913	13:13:02	SELECT * FROM credit_card LIMIT 0, 1000	276 row(s) returned	0.000 sec / 0
✓ 1914	13:13:26	DESCRIBE credit_card	6 row(s) returned	0.000 sec / 0

```

93  -- 2. Elimino campo 'pan' y chequeo su ejecucion con DESCRIBE credit_card
94
95  • ALTER TABLE credit_card
96  DROP COLUMN pan;
97

```

Field	Type	Null	Key	Default	Extra
id	varchar(100)	NO	PRI	NULL	
iban	varchar(100)	YES		NULL	
pin	varchar(100)	YES		NULL	
cvv	varchar(100)	YES		NULL	
expiring_date	varchar(100)	YES		NULL	

Result 17 x

Output

#	Time	Action	Message	Duration / Fets
✓ 1913	13:13:02	SELECT * FROM credit_card LIMIT 0, 1000	276 row(s) returned	0.000 sec / 0
✓ 1914	13:13:26	DESCRIBE credit_card	6 row(s) returned	0.000 sec / 0
✓ 1915	13:15:50	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
✓ 1916	13:15:56	DESCRIBE credit_card	5 row(s) returned	0.000 sec / 0

Nivel 2 -----

Ejercicio 1

Elimina de la tabla transacción el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos.

```

98  -- Nivel 2 -----
99
100 -- Ejercicio 1
101 -- Elimina de la tabla transacción el registro con ID 02C6201E-D90A-1859-B4EE-88D2986D3B02 de la base de datos.
102
103 • SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
104
105

```

Result Grid

Filter Rows:

Edit:

Export/Imports:

Wrap Cell Content:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.92	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

transaction 18

Output

Action Output

#	Time	Action	Message	Duration / Fets
1914	13:13:26	DESCRIBE credit_card	6 row(s) returned	0.000 sec / 0.
1915	13:15:50	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
1916	13:15:56	DESCRIBE credit_card	5 row(s) returned	0.000 sec / 0.
1917	13:19:29	SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3...	1 row(s) returned	0.000 sec / 0.

SCHEMAS

Filter objects

biblioteca

classmodels

hospitales

minos

sakila

sys

transactions

Tables

company

credit_card

transaction

Columns

id

credit_card_id

Administration

Schemas

Information

Column: id

Collation: utf8mb4_0900_ai_ci

Definition:

id varchar(255) PK

Limit to 1000 rows

DELETE FROM transaction

WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';

-- chequeo el borrado del registro

SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';

Result Grid

Filter Rows

Edit

Export/Imports

Wrap Cell Content

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
02C6201E-D90A-1859-B4EE-88D2986D3B02	CcU-2938	b-2362	92	81.9185	-12.5276	2021-08-28 23:42:24	466.92	0

transaction 19 x

Output

Action Output

#	Time	Action	Message	Duration / Fets
1916	13:15:56	DESCRIBE credit_card	5 row(s) returned	0.000 sec / 0.
1917	13:19:29	SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';	1 row(s) returned	0.000 sec / 0.
1918	13:21:22	DELETE FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';	1 row(s) affected	0.015 sec
1919	13:21:30	SELECT * FROM transaction WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';	0 row(s) returned	0.000 sec / 0.

Ejercicio 2

La sección de marketing desea tener acceso a información específica para realizar análisis y estrategias efectivas. Se ha solicitado crear una vista que proporcione detalles clave sobre las compañías y sus transacciones. Será necesaria que crees una vista llamada VistaMarketing que contenga la siguiente información: Nombre de la compañía. Teléfono de contacto. País de residencia. Media de compra realizado por cada compañía. Presenta la vista creada, ordenando los datos de mayor a menor promedio de compra.

```

125 • CREATE VIEW VistaMarketing AS
126 SELECT c.company_name, c.phone, c.country, round(avg(t.amount),2) AS averige_buy_amount
127 FROM company c
128 JOIN transaction t ON c.id = t.company_id
129 WHERE declined = 0 -- transacciones aceptadas
130 GROUP BY c.company_name, c.phone, c.country
131 ORDER BY averige_buy_amount DESC
132 ;
133
134 • SELECT * FROM VistaMarketing;

```

company_name	phone	country	averige_buy_amount
Eget Ipsum Ltd	03 67 44 56 72	United States	481.86
Sed Id Limited	07 28 18 18 13	United States	477.51
Neque Tellus Incorpor...	04 43 18 34 19	Ireland	477.10
Nunc Sit Incorporated	07 28 42 63 63	Norway	461.83
Non Magna LLC	06 71 73 13 17	United Kingdom	458.74
...

VistaMarketing 22 x

Output

#	Time	Action	Message	Duration / Fe
1926	13:27:45	ALTER VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, roun...	0 row(s) affected	0.000 sec
1927	13:27:53	select * from VistaMarketing LIMIT 0, 1000	101 row(s) returned	0.016 sec / (

Ejercicio 3

Filtra la vista VistaMarketing para mostrar sólo las compañías que tienen su país de residencia en "Germany"

```

137 -- Ejercicio 3
138 -- Filtra la vista VistaMarketing para mostrar sólo las compañías que tienen su país de residencia en "Germany"
139
140 • SELECT * FROM VistaMarketing
141 WHERE country = 'Germany';
142
143

```

company_name	phone	country	averige_buy_amount
Ac Industries	09 34 65 40 60	Germany	396.15
Auctor Mauris Corp.	05 62 87 14 41	Germany	308.99
Ac Fermentum Incorp...	06 85 56 52 33	Germany	293.57
Aliquam PC	01 45 73 52 16	Germany	280.34
Rutrum Non Inc.	02 66 31 61 09	Germany	266.90
Nunc Interdum Incorp...	05 18 15 48 13	Germany	242.95
Convallis In Corpora...	06 66 57 29 50	Germany	60.99
Augue Foundation	06 88 43 15 63	Germany	15.05

VistaMarketing 23 x

Output

#	Time	Action	Message	Duration / Fe
1927	13:27:53	select * from VistaMarketing LIMIT 0, 1000	101 row(s) returned	0.016 sec / 0.1
1928	13:52:43	SELECT * FROM VistaMarketing WHERE country = 'Germany' LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.1

Nivel 3

Ejercicio 1


```

137 -- Nivel 3 -----
138
139 -- Ejercicio 1
140 /* La próxima semana tendrás una nueva reunión con los gerentes de marketing.
141    Un compañero de tu equipo realizó modificaciones en la base de datos, pero no recuerda cómo las realizó.
142    Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama: */
143 -- Paso a Paso:
144
145 -- 1. Creamos la tabla user (incluida en los recursos del ejercicio original en moodle)
146 -- 2. Inserto los datos (incluida en los recursos del ejercicio original en moodle)
147 -- 3. Chequeo la tabla

```

```

149 • describe user;
150 • select * from user;
151
152 -- 4. La relacion User-transaction esta invertida (1 transaccion - muchos usuarios = t.user_id - u.id ==> debe ser: u.id - t.user_id -- 1:N--
153 -- Corregir:
154 -- DROP FK en user. Buscar el nombre de la restricción
155
156 • SHOW CREATE TABLE user;
157 -- 'user', 'CREATE TABLE `user` (\n `id` int NOT NULL,\n `name` varchar(100) DEFAULT NULL,\n `surname` varchar(100) DEFAULT NULL,\n `phone`
158
159 • ALTER TABLE user

```

Table	Create Table
transaction	CREATE TABLE `transaction` (`id` varchar(255) NOT NULL, `credit_card_id` varchar(15) DEFAULT NULL, `co...

Result 53 x Read Only

#	Time	Action	Message	Duration / Fetch
2247	10:50:45	CREATE TABLE `user` (`id` int NOT NULL, `name` varchar(100) DEFAULT NULL, ...	Error Code: 1064. You have an error in your SQL syntax; check the manual that comes...	0.015 sec
2248	10:51:54	show create table transaction	1 row(s) returned	0.000 sec / 0.000 sec

```

159 • ALTER TABLE user
160 DROP FOREIGN KEY user_ibfk_1;
161
162 -- 5. Chequeo si se dio de baja la FK en user
163
164 • SHOW CREATE TABLE user;
165
166 • CREATE TABLE `user` (
167   `id` int NOT NULL,
168   `name` varchar(100) DEFAULT NULL,
169   `surname` varchar(100) DEFAULT NULL,

```

```

170   `phone` varchar(150) DEFAULT NULL,
171   `email` varchar(150) DEFAULT NULL,
172   `birth_date` varchar(100) DEFAULT NULL,
173   `country` varchar(150) DEFAULT NULL,
174   `city` varchar(150) DEFAULT NULL,
175   `postal_code` varchar(100) DEFAULT NULL,
176   `address` varchar(255) DEFAULT NULL,
177   PRIMARY KEY (`id`)
178 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
179
180 -- 5. CREAM USER_ID COMO FK EN TRANSACTION

```

Table	Create Table
transaction	CREATE TABLE `transaction` (`id` varchar(255) NOT NULL, `credit_card_id` varchar(15) DEFAULT NULL, `co...

Result 53 x Read Only

#	Time	Action	Message	Duration / Fetch
2247	10:50:45	CREATE TABLE `user` (`id` int NOT NULL, `name` varchar(100) DEFAULT NULL, ...	Error Code: 1064. You have an error in your SQL syntax; check the manual that comes...	0.015 sec
2248	10:51:54	show create table transaction	1 row(s) returned	0.000 sec / 0.000 sec

```

180 -- 5. CREAM USER_ID COMO FK EN TRANSACTION
181 -- Modifico la tabla transaction para poner a user_id como FK en relacion con id de la tabla user
182
183 • ALTER TABLE transaction
184 ADD FOREIGN KEY (user_id) REFERENCES user(id);
185
186 -- Resultado: Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails. No permite hacer el cambio anterior.
187
188 -- .6 Chequeo si hay algun user_id en la tabla transaction que no exista en la tabla 'PARENT' user(id):
189
190 • SELECT DISTINCT user_id

```

```

190 • SELECT DISTINCT user_id
191 FROM transaction
192 WHERE user_id NOT IN (SELECT id FROM user);
193
194 -- user_id 9999 está en la tabla transaction ('CHILD') pero no existe en la tabla user ('PARENT'). Lo agrego a la tabla user.
195
196 • INSERT INTO user (id)
197 VALUES ('9999');
198
199 -- Vuelvo a ejecutar la modificacion:
200
201 • ALTER TABLE transaction
202 ADD FOREIGN KEY (user_id) REFERENCES user(id);
203
204 -- corregido!

```

```

204 -- 7. Chequeo el esquema de la tabla transaction
205
206 • SHOW CREATE TABLE transaction;
207
208 • CREATE TABLE `transaction` (
209   `id` varchar(255) NOT NULL,
210   `credit_card_id` varchar(15) DEFAULT NULL,
211   `company_id` varchar(20) DEFAULT NULL,
212   `user_id` int DEFAULT NULL,
213   `lat` float DEFAULT NULL,
214   `longitude` float DEFAULT NULL,
215   `timestamp` timestamp NULL DEFAULT NULL,
216   `amount` decimal(10,2) DEFAULT NULL,
217   `declined` tinyint(1) DEFAULT NULL,
218   PRIMARY KEY (`id`),

```

```

214   `longitude` float DEFAULT NULL,
215   `timestamp` timestamp NULL DEFAULT NULL,
216   `amount` decimal(10,2) DEFAULT NULL,
217   `declined` tinyint(1) DEFAULT NULL,
218   PRIMARY KEY (`id`),
219   KEY `company_id` (`company_id`),
220   KEY `credit_card_id` (`credit_card_id`),
221   KEY `idx_user_id` (`user_id`),
222   CONSTRAINT `transaction_ibfk_1` FOREIGN KEY (`company_id`) REFERENCES `company` (`id`),
223   CONSTRAINT `transaction_ibfk_2` FOREIGN KEY (`credit_card_id`) REFERENCES `credit_card` (`id`),
224   CONSTRAINT `transaction_ibfk_3` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`),
225   CONSTRAINT `transaction_ibfk_4` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`)
226 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
227
228 -- corregido!

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Table	Create Table		
transaction	CREATE TABLE `transaction` (`id` varchar(255) NOT NULL, `credit_card_id` varchar(15) DEFAULT NULL, `co...		

Modificación de Esquema según solicitud de Ejercicio

```

230 -- 8. Chequear y modificar las tablas/campos de acuerdo con el esquema presentado en el ejercicio.
231 -- Tabla company
232
233 • ALTER TABLE company
234 DROP website;
235
236 -- Tabla credit_card
237 • select * from credit_card;
238
239 • ALTER TABLE credit_card
240 MODIFY COLUMN id VARCHAR(20);

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Table	Create Table		
transaction	CREATE TABLE `transaction` (`id` varchar(255) NOT NULL, `credit_card_id` varchar(15) DEFAULT NULL, `co...		

```

242 -- Error Code: 1833. Cannot change column 'id': used in a foreign key constraint 'transaction_ibfk_2' of table 'transactions.transaction'
243 -- Chequeo la tabla transaction para buscar el nombre de la restricción que no me permite modificar credit_card_id.id
244 • SHOW CREATE TABLE transaction;
245
246 • ALTER TABLE transaction
247 DROP FOREIGN KEY transaction_ibfk_2;
248
249 • ALTER TABLE credit_card
250 MODIFY COLUMN id VARCHAR(20);
251
252 -- Ahora sí se pudo modificar credit_card_id.id

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Table	Create Table		
transaction	CREATE TABLE `transaction` (`id` varchar(255) NOT NULL, `credit_card_id` varchar(15) DEFAULT NULL, `co...		

```

254 -- Restituyo el FK en la tabla transaction con el mismo nombre de CONSTRAINT
255
256 • ALTER TABLE transaction
257   ADD CONSTRAINT transaction_ibfk_2
258   FOREIGN KEY (credit_card_id) REFERENCES credit_card(id);
259
260 -- Chequeo que haya quedado correctamente como en el estado anterior de constraints para
261 -- mantener la integridad del modelo
262
263 • SHOW CREATE TABLE transaction;
264

```

Result 58 x

Output

Action Output

#	Time	Action	Message	Duration / Fe
2285	13:06:51	SHOW CREATE TABLE transaction	1 row(s) returned	0.000 sec / 1

```

269 • ALTER TABLE credit_card MODIFY COLUMN cvv INT;
270 • ALTER TABLE credit_card MODIFY COLUMN expiring_date VARCHAR(20);
271 • ALTER TABLE credit_card ADD COLUMN fecha_actual DATE;
272 -- Tabla user
273 • ALTER TABLE user RENAME data_user;
274 • ALTER TABLE user CHANGE email personal_email VARCHAR (150);
275 -- Tabla transaction - -- SIN CAMBIOS

```

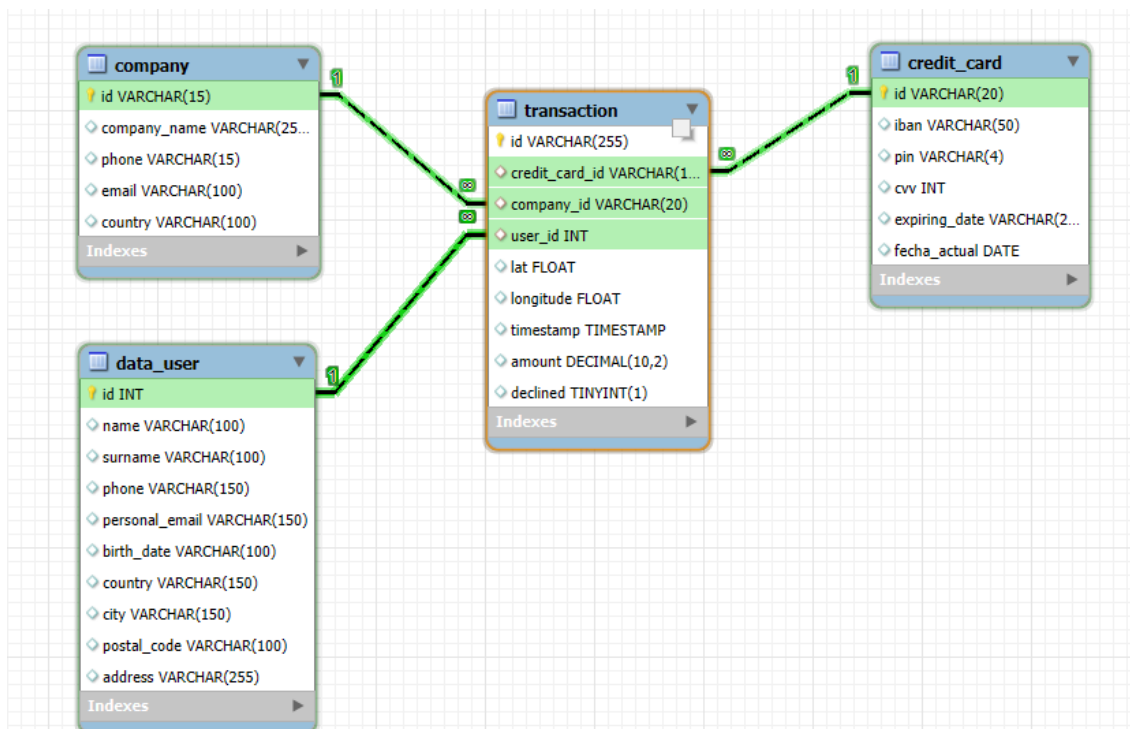
Result 58 x

Output

Action Output

#	Time	Action	Message	Duration / Fe
2281	13:00:46	SHOW CREATE TABLE transaction	1 row(s) returned	0.000 sec / 0
2282	13:01:45	ALTER TABLE transaction DROP FOREIGN KEY transaction_ibfk_2	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
2283	13:03:12	ALTER TABLE credit_card MODIFY COLUMN id VARCHAR(20)	276 row(s) affected Records: 276 Duplicates: 0 Warnings: 0	0.079 sec
2284	13:06:23	ALTER TABLE transaction ADD CONSTRAINT transaction_ibfk_2 FOREIGN KEY...	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0	0.109 sec
2285	13:06:51	SHOW CREATE TABLE transaction	1 row(s) returned	0.000 sec / 0

Diagrama de la base de datos



Ejercicio 2

La empresa también te solicita crear una vista llamada "InformeTecnico" que contenga la siguiente información:

- ID de la transacción
- Nombre del usuario/a
- Apellido del usuario/a
- IBAN de la tarjeta de crédito usada.
- Nombre de la compañía de la transacción realizada.
- Asegúrate de incluir información relevante de ambas tablas y utiliza alias para cambiar de nombre columnas según sea necesario.

Muestra los resultados de la vista, ordena los resultados de forma descendente en función de la variable ID de transacción.

```

303 • CREATE VIEW InformeTecnico AS
304 SELECT t.id AS Transaction_ID , u.name AS User_Name , u.surname AS User_Surname, cc.iban AS IBAN, c.company_name AS Company
305 FROM transaction t
306 JOIN data_user u ON t.user_id = u.id
307 JOIN credit_card cc ON t.credit_card_id = cc.id
308 JOIN company c ON t.company_id = c.id
309 ;

```

```

309
310 • SELECT * FROM InformeTecnico
311 ORDER BY Transaction_ID DESC;
312
313
314
315

```

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

Transaction_ID	User_Name	User_Surname	IBAN	Company
FE96CE47-8D59-381C-4E18-E3CA3D-44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E46468	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-88083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.

InformeTecnico 60 x