

En este sprint, se simula una situación empresarial en la que debes realizar diversas manipulaciones en las tablas de la base de datos. A su vez, tendrás que trabajar con índices y vistas. En esta actividad, continuarás trabajando con la base de datos que contiene información de una empresa dedicada a la venta de productos online. En esta tarea, empezarás a trabajar con información relacionada con tarjetas de crédito.

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.

- Primero creamos la tabla:

```
CREATE TABLE IF NOT EXISTS credit_card(  
  id VARCHAR(20) PRIMARY KEY,  
  iban VARCHAR(50) NULL,  
  pan VARCHAR(30) NULL,  
  pin VARCHAR(4) NULL,  
  cvv VARCHAR(4) NULL,  
  expiring_date VARCHAR(15) NULL);
```

ion Output

Time	Action	Message
1 18:17:31	CREATE TABLE IF NOT EXISTS credit_card(id VARCHAR(20) PRIMARY KEY, iban VARCHAR(50) NULL, pan VARCHAR(30) NULL, pin VARCHAR...	0 row(s) affected, 1 warning(s): 1050 Table 'credit_card' already exists

- Le asigné el formato "VARCHAR" a la columna "expiring_date" en vista de que el formato de los datos proporcionados no coincide con el tipo DATE(YYYY-MM-DD).

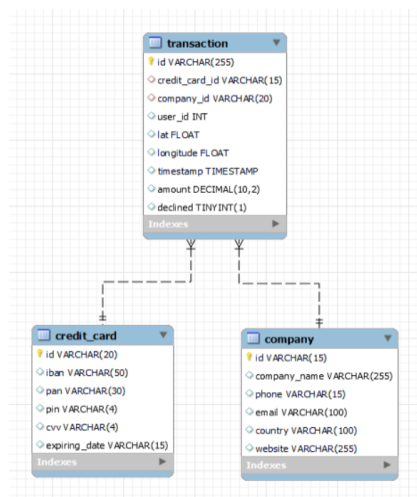
- Cargamos los datos y creé las conexiones entre las tablas:

```
27 • ALTER TABLE transaction  
28   ADD FOREIGN KEY (credit_card_id)  
29   REFERENCES credit_card(id);  
30  
31 -- Ejercicio 2:
```

Output

#	Time	Action	Message
1	18:20:38	ALTER TABLE transaction ADD FOREIGN KEY (credit_card_id) REFERENCES credit_card(id)	587 row(s) affected Records: 587 Duplicates: 0 Warnings: 0

- De momento, este es el esquema de nuestra base de datos:



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.

- Consultamos los datos:

```
35 • SELECT *
36 FROM credit_card
37 WHERE id = "CcU-2938";
38
```

Result Grid						
Filter Rows: <input type="text"/> Edit: Export/Import: Wrap Cell Content:						
id	iban	pan	pin	cvv	expiring_date	
CcU-2938	TR301950312213576817638661	5424465566813633	3257	984	10/30/22	
NULL	NULL	NULL	NULL	NULL	NULL	

credit_card 1 x			
Output			
Action Output			
#	Time	Action	Message
1	10:48:48	SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 1000	1 row(s) returned

- Los modificamos y consultamos nuevamente para visualizar los cambios:

```
35 • UPDATE credit_card
36 SET iban = "R323456312213576817699999"
37 WHERE id = "CcU-2938";
38
39 • SELECT *
40 FROM credit_card
41 WHERE id = "CcU-2938";
```

Result Grid						
Filter Rows: <input type="text"/> Edit: Export/Import: Wrap Cell Content:						
id	iban	pan	pin	cvv	expiring_date	
CcU-2938	R323456312213576817699999	5424465566813633	3257	984	10/30/22	
NULL	NULL	NULL	NULL	NULL	NULL	

credit_card 2 x			
Output			
Action Output			
#	Time	Action	Message
1	10:53:03	UPDATE credit_card SET iban = "R323456312213576817699999" WHERE id = "CcU-2938"	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0
2	10:53:26	SELECT * FROM credit_card WHERE id = "CcU-2938" LIMIT 0, 1000	1 row(s) returned

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

- El primer paso para crear el nuevo usuario es agregar los id a sus respectivas tablas.
- Y luego si agregamos los valores indicados:

```

46 • INSERT INTO company(id)
47   VALUES('b-9999');
48
49 • INSERT INTO credit_card(id)
50   VALUES('CcU-9999');
51
52 • INSERT INTO transaction(id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
53   VALUES('108B1D1D-5B23-A76C-55EF-C568E49A99DD', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0');
54
55 • SELECT *
56   FROM transaction
57  WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD';

```

Result Grid

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
108B1D1D-5B23-A76C-55EF-C568E49A99DD	CcU-9999	b-9999	9999	829.999	-117.999	NULL	111.11	0

ransaction 5 ×

Output

#	Time	Action	Message
1	11:15:37	INSERT INTO company(id) VALUES(b-9999)	1 row(s) affected
2	11:15:37	INSERT INTO credit_card(id) VALUES(CcU-9999)	1 row(s) affected
3	11:15:37	INSERT INTO transaction(id, credit_card_id, company_id, user_id, lat, longitude, amount, declined) VALU...	1 row(s) affected
4	11:15:37	SELECT * FROM transaction WHERE id = '108B1D1D-5B23-A76C-55EF-C568E49A99DD' LIMIT 0, 1000	1 row(s) returned

Se observa que el campo **timestamp** queda en NULL, ya que no se ha proporcionado ningún valor. Entiendo es que esta columna debería capturar automáticamente la fecha y hora de la transacción, información que no tenemos disponible en este caso.

Ejercicio 4

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

- Eliminamos la columna y visualizamos la tabla para verificar:

```

62 • ALTER TABLE credit_card
63   DROP COLUMN pan;
64
65 • SELECT*
66   FROM credit_card;

```

Result Grid

id	iban	pin	cvv	expiring_date
CcU-2938	R323456312213576817699999	3257	984	10/30/22
CcU-2945	DO26854763748537475216568689	9080	887	08/24/23
CcU-2952	BG45IVQL52710525608255	4598	438	06/29/21
CcU-2959	CR7242477244335841535	3583	667	02/24/23
CcU-2966	BG72LKTQ15627628377363	4900	130	10/29/24
CcU-2973	PT878N677813502749456146	8760	887	01/30/25

credit_card 6 ×

Output

#	Time	Action	Message
1	11:19:08	ALTER TABLE credit_card DROP COLUMN pan	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
2	11:19:34	SELECT* FROM credit_card LIMIT 0, 1000	276 row(s) returned

Nivel 2

Ejercicio 1

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

```
73 • DELETE FROM transaction
74 WHERE ID = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
75
```

Output			
Action Output			
#	Time	Action	Message
1	11:21:54	DELETE FROM transaction WHERE ID = '02C6201E-D90A-1859-B4EE-88D2986D3B02'	1 row(s) affected

- Comprobamos:

```
80 • SELECT *
81 FROM transaction
82 WHERE ID = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
```

Result Grid									
id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined	

Output			
Action Output			
#	Time	Action	Message
1	18:29:10	SELECT * FROM transaction WHERE ID = '02C6201E-D90A-1859-B4EE-88D2986D3B02' LIMIT 0, 1000	0 row(s) returned

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.

- Creamos y visualizamos la vista:

```
92 • CREATE VIEW VistaMarketing AS
93 SELECT c.company_name, c.phone, c.country, AVG(t.amount) AS MediaCompras
94 FROM transaction t JOIN company c
95 ON t.company_id = c.id
96 GROUP BY c.company_name, c.phone, c.country
97 ORDER BY MediaCompras ASC;
98
99 • SELECT * FROM VistaMarketing;
```


Result Grid			
company_name	phone	country	MediaCompras
Aliquam Erat Volutpat LLP	06 01 63 57 66	Italy	27.665000
Aliquet Vel Vulputate Incorporated	04 40 43 84 18	Netherlands	36.550000
Dui Quis Institute	06 93 28 72 81	New Zealand	43.220000
Tempor Diam Institute	05 01 94 80 87	Netherlands	45.360000
Sed Est Corp.	04 58 02 37 91	Canada	46.250000

Output			
Action Output			
#	Time	Action	Message
1	16:18:23	CREATE VIEW VistaMarketing AS SELECT c.company_name, c.phone, c.country, AVG(t.amount) AS Me...	0 row(s) affected
2	16:18:23	SELECT * FROM VistaMarketing LIMIT 0, 1000	101 row(s) returned

Ejercicio 3

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


```
94 • SELECT *
95 FROM VistaMarketing
96 WHERE country = 'Germany';
```

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

company_name	phone	country	MediaCompras
Convallis In Incorporated	06 66 57 29 50	Germany	156.730000
Auctor Mauris Corp.	05 62 87 14 41	Germany	184.310000
Ac Fermentum Incorporated	06 85 56 52 33	Germany	206.465000
Augue Foundation	06 88 43 15 63	Germany	240.800000
Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.025238
Quibusque Mauris Inc.	02 66 34 61 88	Germany	266.000000

VistaMarketing 9 x

Output

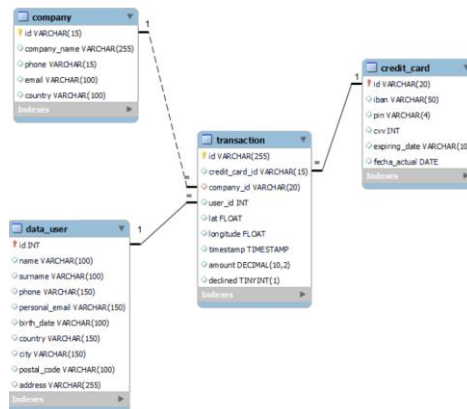
 Action Output

#	Time	Action	Message
1	12:02:05	SELECT * FROM VistaMarketing WHERE country = 'Germany' LIMIT 0, ...	8 row(s) returned

Nivel 3

Ejercicio 1

La próxima semana tendrás una nueva reunión con los gerentes de marketing. Un compañero de tu equipo realizó modificaciones en la base de datos, pero no recuerda cómo las realizó. Te pide que le ayudes a dejar los comandos ejecutados para obtener el siguiente diagrama:



- Primero creamos la tabla “user” y cargamos los datos.

```
106 • CREATE INDEX idx_user_id ON transaction(user_id);
107
108 • CREATE TABLE IF NOT EXISTS user (
109     id INT PRIMARY KEY,
110     name VARCHAR(100),
111     surname VARCHAR(100),
112     phone VARCHAR(150),
113     email VARCHAR(150),
114     birth_date VARCHAR(100),
115     country VARCHAR(150),
116     city VARCHAR(150),
117     postal_code VARCHAR(100),
118     address VARCHAR(255));
119
120 -- Cargamos los datos, dentro de ellos agregué el id '9999' el cual corresponde a uno de los ejercicios anteriores
121 -- Hacemos la conexión entre las tablas
122
123 • ALTER TABLE transaction
124     ADD FOREIGN KEY (user_id)
125     REFERENCES user(id);
126
```

Output

#	Time	Action	Message
1	13:17:58	CREATE INDEX idx_user_id ON transaction(user_id)	0 row(s) affected. Records: 0 Duplicates: 0 Warnings: 0
2	13:17:58	CREATE TABLE IF NOT EXISTS user (id INT PRIMARY KEY, name VARCHAR(100), surname VARCHAR(100), phone VARCHAR(150), email VARCHAR(150), birth_date VARCHAR(100), country VARCHAR(150), city VARCHAR(150), postal_code VARCHAR(100), address VARCHAR(255));	0 row(s) affected. 1 warning(s): 1050 Table 'user' already exists
3	13:17:58	ALTER TABLE transaction ADD FOREIGN KEY (user_id) REFERENCES user(id)	587 row(s) affected. Records: 587 Duplicates: 0 Warnings: 0

En la carga de datos agregué manualmente la id “9999”, ya que lo habíamos agregado anteriormente a la tabla “transaction”:

```
279 • INSERT INTO user (id) VALUES ('9999');
```

Ahora pasamos a comparar los cambios realizados:

- Con respecto a la tabla company: vemos que se ha eliminado la columna “website”, por lo tanto:

```
130 • ALTER TABLE company
131 DROP COLUMN website;
132
```

Output					
Action Output					
#	Time	Action	Message	Duration / F	
1	13:22:25	ALTER TABLE company DROP COLUMN we...	0 row(s) affected Records: 0 Duplicates: 0 W...	0.031 sec	

- Con respecto a la tabla credit_card: vemos que el 'cvv' tiene formato INT, que la columna 'expiring_date' tiene otra cantidad de caracteres y que se crea la columna 'fecha_actual' con el CURDATE para que siempre nos imprima la fecha de ese momento, por lo tanto, lo ajustamos:

```
142 • ALTER TABLE credit_card
143 MODIFY COLUMN cvv INT,
144 MODIFY COLUMN expiring_date VARCHAR(10),
145 ADD COLUMN fecha_actual DATE DEFAULT(CURDATE());
```

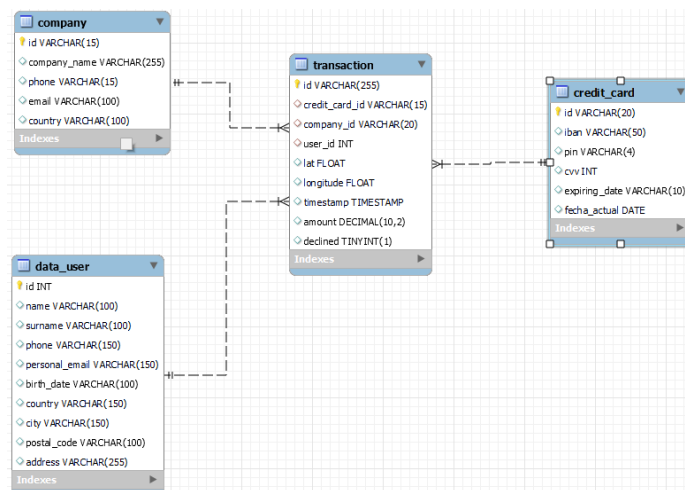
Output					
Action Output					
#	Time	Action	Message	Duration / F	
1	10:44:25	ALTER TABLE credit_card MODIFY COLUMN cvv INT, MODIFY COLUM...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0		

- Con respecto a la tabla 'user': la tabla tiene un nombre diferente y la columna 'email' tiene el nombre 'personal_email', por lo tanto:

```
143 • ALTER TABLE user
144 RENAME TO data_user,
145 CHANGE COLUMN email personal_email VARCHAR(150);
...
```

Output					
Action Output					
#	Time	Action	Message	Duration / F	
1	19:22:41	ALTER TABLE user RENAME TO data_user, CHANGE COLUMN email personal_email VARCHAR(150)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0		

Nuestra tabla quedaría de la siguiente manera:



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.

- Creamos la vista y la visualizamos:

```
162 • CREATE VIEW InformeTecnico AS
163 SELECT t.id AS TransactionID, t.timestamp AS TransactionDate, t.amount AS Amount, u.name UserName,
164 u.surname AS UserSurname, cc.iban AS IBAN, c.company_name AS Company
165 FROM transaction t
166 JOIN company c
167 ON t.company_id = c.id
168 JOIN data_user u
169 ON t.user_id = u.id
170 JOIN credit_card cc
171 ON t.credit_card_id = cc.id
172 GROUP BY t.id, t.timestamp, t.amount
173 ORDER BY TransactionID DESC;
174
175 • SELECT *
176 FROM InformeTecnico;
```

Result Grid							
Filter Rows:		Export:		Wrap Cell Content:			
TransactionID	TransactionDate	Amount	UserName	UserSurname	IBAN	Company	
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	2021-06-15 00:26:29	480.13	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries	
FE809ED4-2DB6-55AC-C915-929516E4646B	2021-11-09 21:35:40	219.83	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated	
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	2021-06-13 11:41:17	42.32	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated	
FD89D51B-AE8D-77DC-E450-B8083FBD3187	2022-03-16 02:35:05	200.72	Hilda	Levy	LT053237077744561475	Malesuada PC	
FD2E8957-414B-BEEC-E9AD-59AA7ABA6290	2022-03-13 00:27:34	78.29	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.	

InformeTecnico 3

Output

Action Output

#	Time	Action	Message
✓ 1	16:16:43	CREATE VIEW InformeTecnico AS SELECT t.id AS TransactionID, t.timestamp AS TransactionDate, t.am...	0 row(s) affected
✓ 2	16:16:43	SELECT * FROM InformeTecnico LIMIT 0, 1000	587 row(s) returned