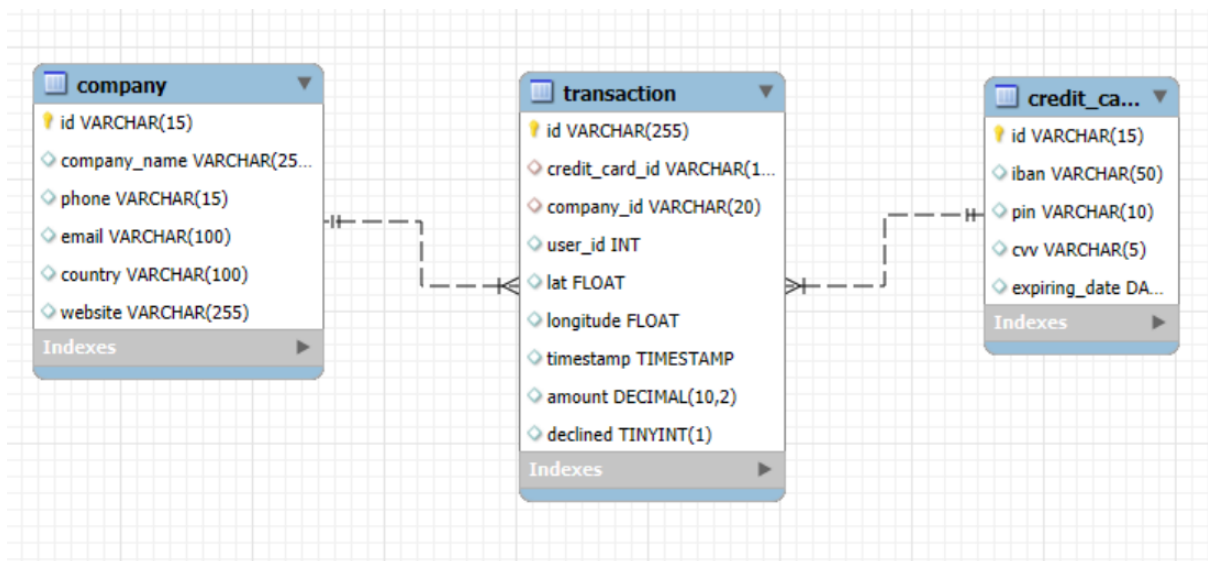


Sprint 3

- Nivel 1

Ejercicio 1.



Nombre de la base de datos: transactions

Concepto del Modelo: Esquema en estrella.

- **Tabla de Hechos:** transaction
- **Tablas de Dimensiones:** company, credit_card

Tabla company (Dimensión):

- **id:** Clave de la dimensión company (Clave Primaria - PK).
- **company_name:** Nombre completo de la empresa.
- **country:** País donde está ubicada la empresa.
- **phone:** Número de teléfono de contacto de la empresa.
- **email:** Dirección de correo electrónico de la empresa.
- **website:** URL del sitio web de la empresa.

Tabla credit_card (Dimensión):

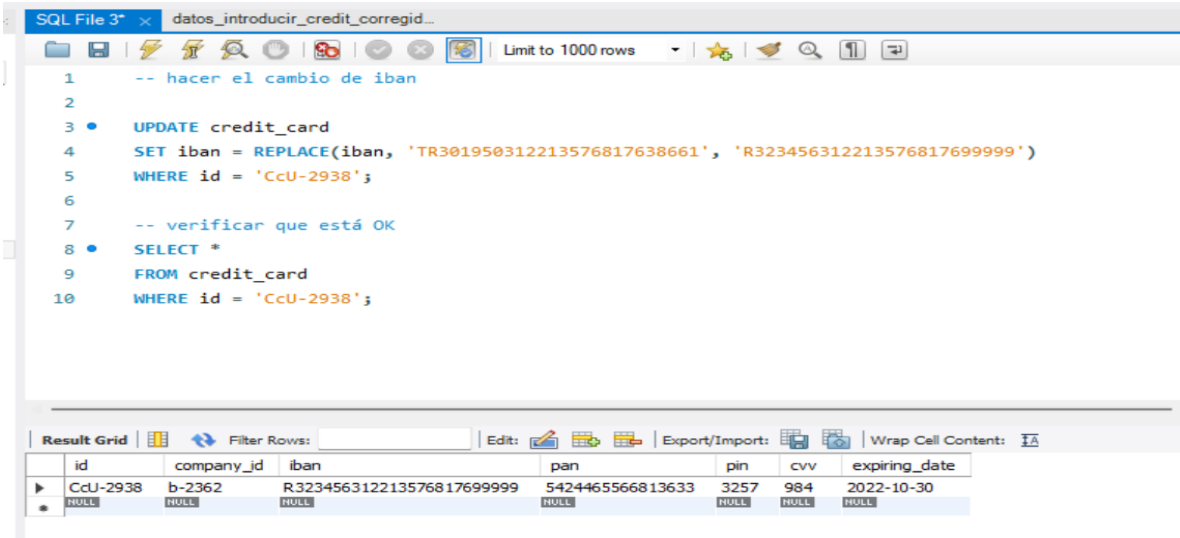
- **id**: Clave de la dimensión credit_card (Clave Primaria - PK).
- **iban**: Número de Cuenta Bancaria Internacional (IBAN) asociado a la tarjeta.
- **pan**: Número de Identificación Principal de la Cuenta (PAN) de la tarjeta.
- **pin**: Número de Identificación Personal (PIN) de la tarjeta (Nota: en un sistema real, el PIN nunca se almacenaría directamente por seguridad).
- **cvv**: Código de Verificación de Valor (CVV) de la tarjeta
- **expiring_date**: Fecha de caducidad de la tarjeta de crédito.

Tabla transaction (Tabla de Hechos):

- **id**: Identificador único de la transacción (Clave Primaria - PK).
- **credit_card_id**: Clave Foránea (FK) que referencia al id de la tabla de dimensión credit_card.
- **company_id**: Clave Foránea (FK) que referencia al id de la tabla de dimensión company.
- **user_id**: Identificador único del usuario que realizó la transacción
- **lat**: Latitud geográfica donde se realizó la transacción.
- **longitude**: Longitud geográfica donde se realizó la transacción.
- **timestamp**: Fecha y hora exacta en que se registró la transacción
- **amount**: Valor monetario de la transacción
- **declined**: Indicador (booleano/tinyint) que señala si la transacción fue declinada o no.

Relación entre tablas :

- **Relación 1:N (Uno a Muchos) entre company (Dimensión) y transaction (Hechos):**
 - Cada fila en la tabla de hechos transaction se relaciona con una única fila en la tabla de dimensión company a través de transaction.company_id.
 - Una empresa puede estar asociada a muchas transacciones.
- **Relación 1:N (Uno a Muchos) entre credit_card (Dimensión) y transaction (Hechos):**
 - Cada fila en la tabla de hechos transaction se relaciona con una única fila en la tabla de dimensión credit_card a través de transaction.credit_card_id.
 - Una tarjeta de crédito puede estar asociada a muchas transacciones.

Ejercicio 2.

The screenshot shows a SQL IDE window titled "datos_introducir_credit_corregid...". The SQL editor contains the following code:

```
1 -- hacer el cambio de iban
2
3 • UPDATE credit_card
4   SET iban = REPLACE(iban, 'TR301950312213576817638661', 'R323456312213576817699999')
5   WHERE id = 'CcU-2938';
6
7 -- verificar que está OK
8 • SELECT *
9   FROM credit_card
10  WHERE id = 'CcU-2938';
```

The result grid at the bottom shows the following data:

	id	company_id	iban	pan	pin	cvv	expiring_date
▶	CcU-2938	b-2362	R323456312213576817699999	5424465566813633	3257	984	2022-10-30
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ejercicio 3.

```
-- Ejercicio 3. En la taula "transaction" ingresa un nou usuari amb la següent informació:

• INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
VALUES ('10881D1D-5B23-A76C-55EF-C568E49A990D', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0');

-- Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`transactions`.`transaction`, CONSTRAINT `transaction_ibfk_1`
-- FOREIGN KEY (`company_id`) REFERENCES `company` (`id`)) -> Necesito primero insertar la compañía en tabla company

• INSERT INTO company (id, company_name, country, phone, email, website)
VALUES ('b-9999', 'Nueva Compañía de Prueba', 'Desconocido', 'N/A', 'info@nuevacompania.com', 'http://www.nuevacompania.com');
-- Relleno los otros campos con valores creados relevantes

-- verifico si existe el id en credit_card
• select*
from credit_card
where id = 'CcU-9999';

-- no existe, lo creo

• INSERT INTO credit_card (id, iban, pin, cvv, expiring_date)
VALUES ('CcU-9999', 'ES991234567890123456789012', '3257', '984', '2028-12-31');

-- Inserto la transacción.
• INSERT INTO transaction (Id, credit_card_id, company_id, user_id, lat, longitude, amount, declined)
VALUES ('10881D1D-5B23-A76C-55EF-C568E49A990D', 'CcU-9999', 'b-9999', '9999', '829.999', '-117.999', '111.11', '0');
```

Ejercicio 4.

SQL File 3* SQL File 7* SQL File 9* x

Limit to 500 rows

```
1 ALTER TABLE credit_card
2 DROP COLUMN pan;
3
4 • describe credit_card;
```

2 • describe credit_card;

Result Grid						
		Filter Rows:			Export:	Wrap Cell Content:
	Field	Type	Null	Key	Default	Extra
▶	id	varchar(15)	NO	PRI	NULL	
	company_id	varchar(15)	YES	MUL	NULL	
	iban	varchar(50)	YES		NULL	
	pin	varchar(10)	YES		NULL	
	cvv	varchar(5)	YES		NULL	
	expiring_date	date	YES		NULL	

- Nivel 2

Ejercicio 1.

The screenshot shows a SQL IDE with a query editor and a result grid. The query editor contains the following SQL code:

```

1 • select *
2   from transaction
3  WHERE id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';
4
5 • delete from transaction where id = '02C6201E-D90A-1859-B4EE-88D2986D3B02';

```

The result grid shows the following data:

id	credit_card_id	company_id	user_id	lat	longitude	timestamp	amount	declined
02C6201E-D90A-1859-B4EE-88D2986D3B02	CdU-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

Ejercicio 2.

The screenshot shows a SQL IDE with a query editor and a result grid. The query editor contains the following SQL code:

```

10
11 -- Ejercicio 2. La secció de màrqueting desitja tenir accés a informació específica per a realitzar anàlisi i estratègies efectives.
12 -- S'ha sol·licitat crear una vista que proporcioni detalls clau sobre les companyies i les seves transaccions.
13 -- Serà necessària que creis una vista anomenada VistaMarketing que contingui la següent informació: Nom de la companyia. Telèfon de contacte.
14 -- País de residència. Mitjana de compra realitzat per cada companyia. Presenta la vista creada, ordenant les dades de major a menor mitjana de compra.
15
16 • CREATE VIEW VistaMarketing AS
17   SELECT company_name as 'Nombre de la companyia', phone as 'Teléfono de contacto', country as 'País de residencia', AVG (amount) as 'Media de compra'
18   FROM company
19   JOIN transaction
20   ON company.id = transaction.company_id
21   GROUP BY company_name, phone, country
22   ORDER BY AVG (amount) DESC;
23
24 -- Ejercicio 3. Filtra la vista VistaMarketing per a mostrar només les companyies que tenen el seu país de residència en "Germany"

```

The result grid shows the following data:

Field	Type	Null	Key	Default	Extra
id	varchar(15)	NO	PRI	NULL	
company_id	varchar(15)	YES	MUL	NULL	
ban	varchar(50)	YES		NULL	
pin	varchar(10)	YES		NULL	
civ	varchar(5)	YES		NULL	

Ejercicio 3.

VEL 1 NIVEL 2* SQL File 10*

Limit to 500 rows

```

1 • SELECT *
2 FROM vistamarketing
3 WHERE País_de_residencia = 'Germany';

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Nombre_de_la_compañía	Teléfono_de_contacto	País_de_residencia	Media_de_compra
Aliquam PC	01 45 73 52 16	Germany	385.265000
Ac Industries	09 34 65 40 60	Germany	289.645000
Rutrum Non Inc.	02 66 31 61 09	Germany	266.900000
Nunc Interdum Incorporated	05 18 15 48 13	Germany	244.025238
Augue Foundation	06 88 43 15 63	Germany	240.800000
Ac Fermentum Incorporated	06 85 56 52 33	Germany	206.465000
Auctor Mauris Corp.	05 62 87 14 41	Germany	184.310000
Convallis In Incorporated	06 66 57 29 50	Germany	156.730000

206.465000

- Nivel 3**Ejercicio 1.****Proceso de normalización y establecimiento de relaciones:****1. Incorporación de la Entidad data_user:**

Se creó la tabla data_user (inicialmente 'User', luego renombrada) y se cargaron los datos de los usuarios.

2. Ajuste de tipos de datos para Claves Foráneas:

Se identificaron y corrigieron inconsistencias en las longitudes de los tipos de datos entre las claves foráneas en transaction y sus respectivas claves primarias (ej., transaction.credit_card_id ajustado de VARCHAR(15) a VARCHAR(20) para coincidir con credit_card.id). Esto requirió eliminar temporalmente claves foráneas existentes en transaction, modificar las columnas afectadas, y

prepararlas para su recreación.

3. Limpieza y ajuste de datos.

A pesar de tener los tipos de datos correctos, existían valores en las columnas de clave foránea de transaction que no tenían una entrada correspondiente en las tablas principales (por ejemplo, user_id = 9999 en transaction no existía en data_user.id). Esto causaba el Error Code: 1452 al intentar crear las claves foráneas.

Se realizaron consultas para identificar estos registros huérfanos y se aplicaron soluciones. En el caso el campo user_id permite NULL y se completó la info del usuario de manera estándar. Se podría haber eliminado el registro o conservado siendo posteriormente modificado el ID para que coincida con el ID de tipo incremental de la tabla data_user. Esto finalmente dependerá de la preferencia de la compañía para la que se realiza el análisis de datos.

IMPORTANTE: Garantizar que todos los datos cumplan con la regla de integridad referencial, es decir, que cada clave foránea apunte a una clave primaria existente.

Por lo que se verificó la correspondencia de los registros de transaction.user_id y data_user.id. Verificar si quedan muchos datos sin relacionar. Completar la info del usuario 9999.

4. Establecimiento final de relaciones de Clave Foránea:

Una vez que los tipos de datos fueron consistentes y los datos estaban depurados, se añadieron las claves foráneas desde la tabla transaction hacia data_user, company y credit_card, formalizando las relaciones de uno a muchos en el esquema.

Ejercicio 2.

CREATE VIEW InformeTecnico AS

```

SELECT
  transaction.id AS ID_de_la_transacción,
  data_user.name AS Nombre_del_Usuario,
  data_user.surname AS Apellido_del_Usuario,
  credit_card.iban AS Iban_TarjetaCredito,
  company.company_name AS Nombre_De_Compañía
FROM
  transaction
JOIN
  data_user ON transaction.user_id = data_user.id
JOIN
  credit_card ON transaction.credit_card_id = credit_card.id
JOIN
  company ON transaction.company_id = company.id

```

SELECT *

```

FROM InformeTecnico
ORDER BY ID_de_la_transacción DESC

```

ID_de_la_transacción	Nombre_del_Usuario	Apellido_del_Usuario	Iban_TarjetaCredito	Nombre_De_Compañía
FE96CE47-BD59-381C-4E18-E3CA3D44E8FF	Kenyon	Hartman	DO26854763748537475216568689	Magna A Neque Industries
FE809ED4-2DB6-55AC-C915-929516E4646B	Molly	Gilliam	SE2813123487163628531121	Nunc Interdum Incorporated
FD9CBCCD-8E1E-8DA1-4606-7E3A6F3A5A65	Linus	Willis	KW9485332754781757886242955643	Nunc Interdum Incorporated
FD89D51B-AE8D-77DC-E450-B8083FBD3187	Hilda	Levy	LT053237077744561475	Malesuada PC
FD2E8957-414B-BEEC-E9AD-59AA7A8A6290	Hedwig	Gilbert	GE84848451582810541526	Neque Tellus Imperdiet Corp.
FCE2AB9A-271D-2BDC-9E49-8DD92A373391	Hakeem	Alford	MD1234119525145401270486	Nunc Interdum Incorporated
FBD7E0D6-8A6B-F5BC-0CA9-EA4B8760100C	Hedwig	Gilbert	MU4132333444534342541344788855	Mauris Id Inc.
FAC76A80-8448-69AA-E892-426C2F12621C	Slade	Poole	MT05JWCF58868200575771634583813	Arcu LLP
FAAD3FFC-1A17-E141-43D3-359A5BA7C83B	Hedwig	Gilbert	GE90157928843338134463	Lorem Eu Incorporated
FA053936-75D8-89FA-490D-9B624E1B920A	Hedwig	Gilbert	GT02497653655330848247645975	Non Justo Corp.
F85A7D75-2778-9D75-D776-3F41A828DE88	Sarah	Beck	VG1468087984174645729577	Ut Semper Foundation
F843DC08-C0B5-2444-1B4E-5966289FBA88	Jasper	Landry	VG1468087984174645729577	Ut Semper Foundation
F5ACD74B-4275-5AA1-2414-6EF417636898	Nora	Reeves	MD1234119525145401270486	Nunc Interdum Incorporated
F56FCA4A-0039-9F64-7376-85632B91121B	Lynn	Riddle	CR7242477244335841535	Ut Semper Foundation
F55B3CE1-3379-E0BF-5A89-6F4CC2C5479C	Sonya	McKee	EE541536644818872885	Arcu LLP
F4BCAE41-368E-EA8D-9C24-466F7CEB9F9A	Chester	Haynes	CY94263537405015481188625576	Malesuada PC
F2B3E645-2E6D-E891-9D05-33DBACE58DE4	Heather	Burks	SM6022751049715477062682363	Malesuada PC