SQL - SPRINT 4

Alicia Ripoll

IT ACADEMY

Barcelona Activa

SPRINT 4

NIVEL 1

- Diseño de una BBDD en base a varios archivos CSV

```sql
CREATE DATABASE orders;
USE orders;

-- USUARIOS-------------------------------------
CREATE TABLE users (
    id INT NOT NULL PRIMARY KEY,
    name VARCHAR(50) NULL,
    surname VARCHAR(50) NULL,
    phone VARCHAR(50) NULL,
    email VARCHAR(100) NULL,
    birth_date VARCHAR(20) NULL,
    country VARCHAR(50) NOT NULL,
    city VARCHAR(50) NULL,
    postal_code VARCHAR(20) NULL,
    address VARCHAR(100) NULL
    );

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/american_users.csv'
INTO TABLE users
FIELDS TERMINATED BY ","
ENCLOSED BY '"'
LINES TERMINATED BY "\n"
IGNORE 1 ROWS;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/european_users.csv'
INTO TABLE users
FIELDS TERMINATED BY ","
ENCLOSED BY '"'
LINES TERMINATED BY "\n"
IGNORE 1 ROWS;
```

Output

Action Output

#	Time	Action	Message
445	10:22:15	CREATE DATABASE orders	1 row(s) affected
446	10:22:19	USE orders	0 row(s) affected
447	10:22:41	CREATE TABLE users (id INT NOT NULL PRIMARY KEY, name VARCHAR(50) NULL, surname VARC...	0 row(s) affected
448	10:22:45	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/american_users.csv' INTO TABLE...	1010 row(s) affected Records: 1010 Deleted: 0 Skipped: 0 Warnings: 0
449	10:22:47	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/european_users.csv' INTO TABL...	3990 row(s) affected Records: 3990 Deleted: 0 Skipped: 0 Warnings: 0

```sql
-- EMPRESAS -----------------------------------------------
CREATE TABLE company (
    company_id VARCHAR(20) NOT NULL PRIMARY KEY,
    company_name VARCHAR(50) NULL,
    phone VARCHAR(50) NULL,
    email VARCHAR(100) NULL,
    country VARCHAR(50) NOT NULL,
    website VARCHAR(150) NULL
    );

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv'
INTO TABLE company
FIELDS TERMINATED BY ","
ENCLOSED BY '"'
LINES TERMINATED BY "\n"
IGNORE 1 ROWS;
```

Output

Action Output

#	Time	Action	Message
453	10:25:23	CREATE TABLE company (company_id VARCHAR(20) NOT NULL PRIMARY KEY, company_name VAR...	0 row(s) affected
454	10:25:26	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv' INTO TABLE com...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

```sql
-- TARJETAS------------------------------------------
CREATE TABLE credit_cards (
    id VARCHAR(20) NOT NULL PRIMARY KEY,
    user_id INT NULL,
    iban VARCHAR(100) NULL,
    pan VARCHAR(100) NULL,
    pin VARCHAR(5) NULL,
    cvv VARCHAR(5) NULL,
    track1 VARCHAR(100) NULL,
    track2 VARCHAR(100) NULL,
    expiring_date VARCHAR(50) NOT NULL
    );


LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
INTO TABLE credit_cards
FIELDS TERMINATED BY ","
ENCLOSED BY '"'
LINES TERMINATED BY "\n"
IGNORE 1 ROWS;
```

Output

Action Output

#	Time	Action	Message
474	10:54:00	CREATE TABLE credit_cards (id VARCHAR(20) NOT NULL PRIMARY KEY, user_id INT NULL, iban V...	0 row(s) affected
475	10:54:08	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv' INTO TABLE cr...	5000 row(s) affected Records: 5000 Deleted: 0 Skipped: 0 Warnings: 0

SPRINT 4

NIVEL 1

```sql
-- TRANSACCIONES----------------------------------------
CREATE TABLE transactions (
    id VARCHAR(50) NOT NULL PRIMARY KEY,
    card_id VARCHAR(100) NOT NULL,
    business_id VARCHAR(50) NOT NULL,
    timestamp timestamp,
    amount DECIMAL(10,2),
    declined TINYINT NOT NULL,
    products_id VARCHAR(50) NOT NULL,
    user_id INT NOT NULL,
    lat FLOAT,
    longitude FLOAT,
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (card_id) REFERENCES credit_cards(id),
    FOREIGN KEY (business_id) REFERENCES company(company_id)
    );

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv'
INTO TABLE transactions
FIELDS TERMINATED BY ";"
LINES TERMINATED BY "\n"
IGNORE 1 ROWS;
```

Output

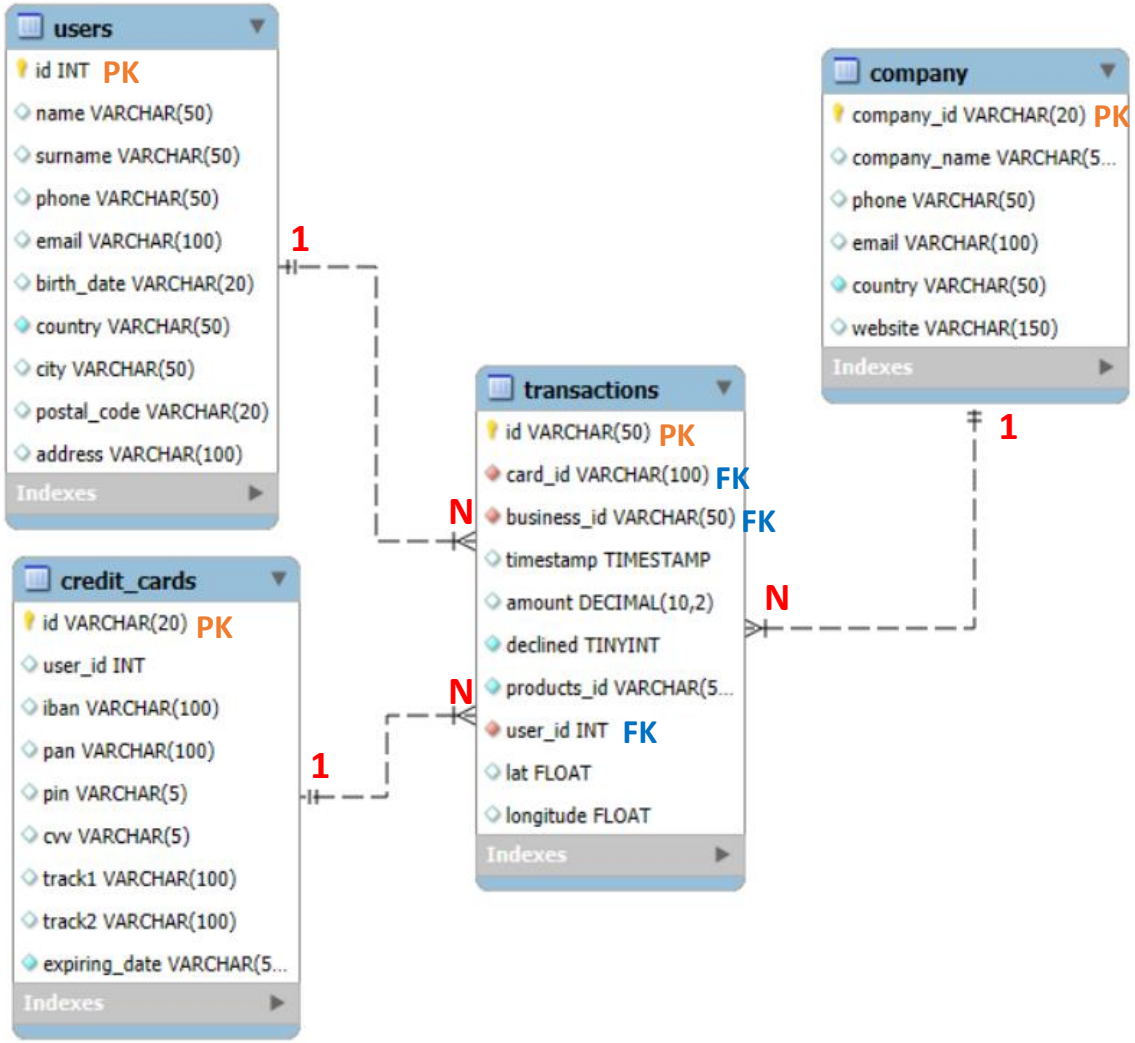
Action Output

#	Time	Action	Message
461	10:35:12	CREATE TABLE transactions (id VARCHAR(50) NOT NULL PRIMARY KEY, card_id VARCHAR(100) NOT...	0 row(s) affected
462	10:35:14	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv' INTO TABLE tra...	100000 row(s) affected Records: 100000 Deleted: 0 Skipped: 0 Warnings: 0

SPRINT 4

NIVEL 1

Análisis de relaciones



- **Modelo dimensional en estrella**: En el centro tenemos la tabla de hechos 'transactions', donde están registradas todas las compras realizadas.

- Contamos con **tres tablas de dimensiones** ('users','company','credit_cards'), que se pueden relacionar a la tabla de hechos mediante las FK.

- De la tabla de hechos a las de dimensiones, tenemos relaciones de 1 a muchos, siendo que cada dimensión nos ayuda a contextualizar cada transacción.

SPRINT 4

NIVEL 1

1.1

- Realizar una subconsulta que muestre todos los usuarios con más de 80 transacciones utilizando por lo menos 2 tablas.

```sql
SELECT *
FROM users u
WHERE id IN (
    SELECT t.user_id
    FROM transactions t
    WHERE declined = 0
    GROUP BY user_id
    HAVING COUNT(t.id) > 80
    );
```

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

id	name	surname	phone	email	birth_date	country	city	postal_code	address
185	Molly	Gilliam	0800 120 8023	donec@outlook.couk	Dec 21, 1993	United Kingdom	London	EC1A 1BB	P.O. Box 202, 5638 Mi Rd.
289	Dxwgi	Hwcru	+98-309-8797	dxwgi.hwcru@example.com	Aug 20, 1976	Germany	Stuttgart	70173	82 Hwcru Street
318	Bnyr	Astuw	+33-120-9644	bnyr.astuw@example.com	May 3, 1974	Italy	Genoa	16100	53 Astuw Street
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

users 116

Output

Action Output

#	Time	Action	Message
463	10:44:22	SELECT * FROM users u WHERE id IN (SELECT t.user_id FROM transactions t WHERE declined = 0 ...	3 row(s) returned

1.2

- Muestra la media de 'amount' por IBAN de las tarjetas de crédito de la compañía Donec Ltd, utiliza por lo menos 2 tablas.

```sql
SELECT cc.iban as IBAN, ROUND(AVG(t.amount),2) as Cantidad_Media
FROM credit_cards cc
JOIN transactions t
ON cc.id = t.card_id
WHERE t.business_id = (
    SELECT c.company_id
    FROM company c
    WHERE company_name = 'Donec Ltd'
    AND declined = 0)
GROUP BY IBAN;
```

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

IBAN	Cantidad_Media
XX8238042338901741490548 1	9.24
XX1957549299088485776208 5	348.80
XX2431116350902288725027 82	288.77
XX8998730308231330180357 71	325.41
XX7488907290571957117660 71	607.29
XX6150128278427867122171 8	514.33

Result 119

Output

Action Output

#	Time	Action	Message
466	10:48:01	SELECT cc.iban as IBAN, ROUND(AVG(t.amount),2) as Cantidad_Media FROM credit_cards cc JOIN transac...	370 row(s) returned

SPRINT 4

NIVEL 2

- Creación de una nueva tabla que refleje el estado de las tarjetas, basándonos en sus últimos tres movimientos

```sql
CREATE TABLE card_status as (
    SELECT card_id,
    SUM(declined) as reject_count,
    CASE
        WHEN sum(declined) = 3 THEN 'Rejected'
        ELSE 'Active'
    END AS state
    FROM (
        SELECT t.card_id, t.declined,
        ROW_NUMBER() OVER(PARTITION BY t.card_id ORDER BY t.timestamp DESC) as last_mov
        FROM transactions t
        ) as mov
    WHERE last_mov <= 3
    GROUP BY card_id
    );

ALTER TABLE card_status
ADD CONSTRAINT fk_card_id FOREIGN KEY (card_id) REFERENCES credit_cards(id);
```

Output

Action Output

#	Time	Action	Message
506	11:47:45	CREATE TABLE card_status as (SELECT card_id, SUM(declined) as reject_count, CASE WHEN sum(de...	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0
507	11:47:49	ALTER TABLE card_status ADD CONSTRAINT fk_card_id FOREIGN KEY (card_id) REFERENCES credit_c...	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0

2.1

- Cuantas tarjetas están activas?

```sql
SELECT *
FROM card_status
WHERE state = 'Active';
```

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

card_id	reject_count	state
CcS-4857	0	Active
CcS-4858	0	Active
CcS-4859	0	Active
CcS-4860	0	Active
CcS-4861	0	Active

card_status 128

Output

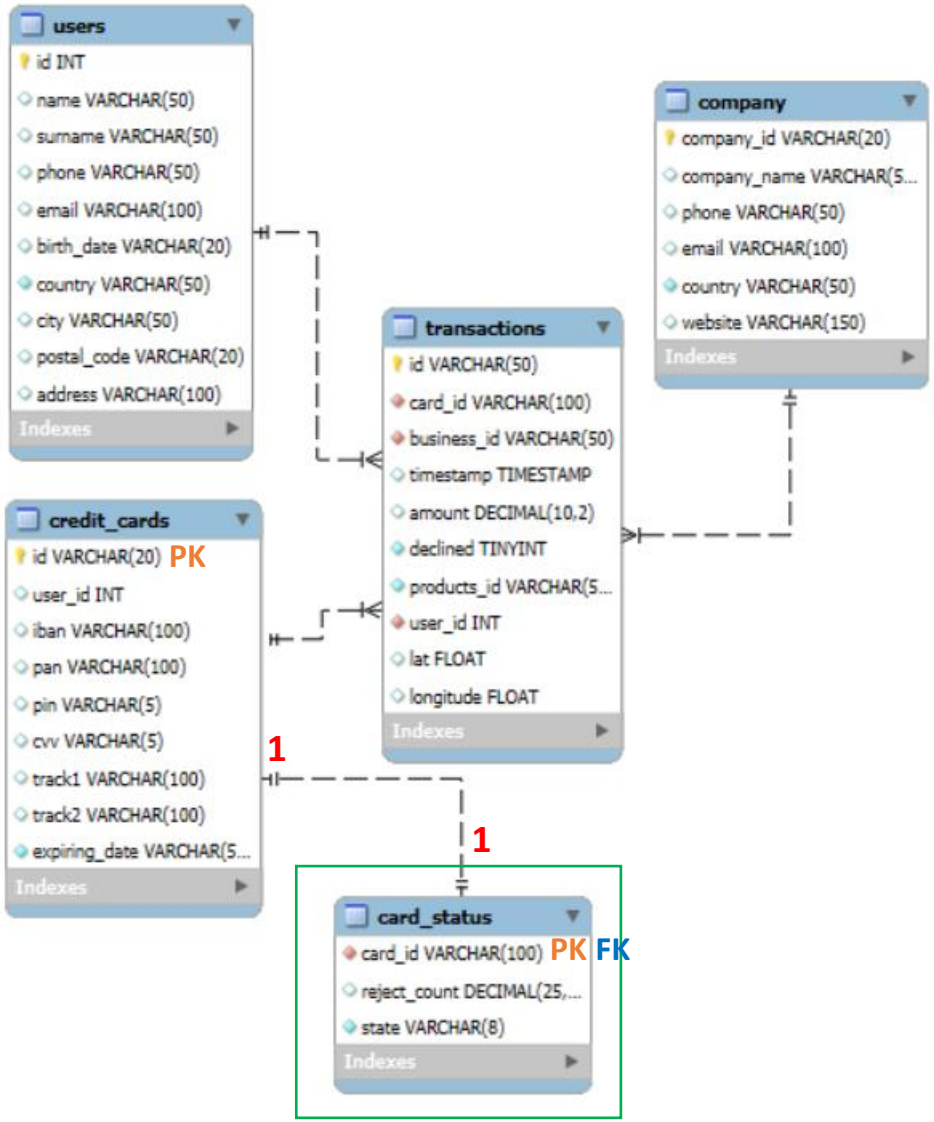
Action Output

#	Time	Action	Message
501	11:44:53	SELECT * FROM card_status WHERE state = 'Active'	4995 row(s) returned

SPRINT 4

NIVEL 2

Análisis de relaciones



- La nueva tabla de 'card_status' se puede considerar una **subdimensión** de la tabla de 'credit_cards', ya que nos aporta más detalle acerca al estado de cada tarjeta que tenemos en la dimensión.

- La relación entre ambas tablas es de 1 a 1, ya que usamos el mismo campo como identificador único. Cada tarjeta sólo puede tener un estado (Activa o Rechazada).

- Precisamente por el tipo de relación entre ambas tablas, el **campo del id** es, al mismo tiempo, la PK de la tabla 'card_status' y la FK que la une a la tabla 'credit_cards'.

SPRINT 4

NIVEL 3

- Creación de una tabla intermedia que conecte los datos de 'products.csv' con el resto de la base de datos.

```sql
-- PRODUCTOS----------------------------------------
CREATE TABLE products (
    id INT NOT NULL PRIMARY KEY,
    product_name VARCHAR(50),
    price VARCHAR(10),
    colour VARCHAR(20),
    weight FLOAT,
    warehouse_id VARCHAR(10)
    );

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv'
INTO TABLE products
FIELDS TERMINATED BY ","
ENCLOSED BY '"'
LINES TERMINATED BY "\n"
IGNORE 1 ROWS;
```

Output

Action Output

#	Time	Action	Message
509	11:56:26	CREATE TABLE products (id INT NOT NULL PRIMARY KEY, product_name VARCHAR(50), price VA...	0 row(s) affected
510	11:56:28	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv' INTO TABLE produ...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

```sql
CREATE TABLE details_transactions (
    id INT AUTO_INCREMENT PRIMARY KEY,
    transaction_id VARCHAR(50) NOT NULL,
    product_id INT NOT NULL,
    FOREIGN KEY (transaction_id) REFERENCES transactions(id),
    FOREIGN KEY (product_id) REFERENCES products(id)
    );

INSERT INTO details_transactions (transaction_id,product_id)
SELECT t.id, j_query.product_id
FROM transactions t
JOIN JSON_TABLE(
  CONCAT('["', REGEXP_REPLACE (t.products_id, '[[:space:]]*,[[:space:]]*', '","'), '"]'),
  '$[*]' COLUMNS (product_id VARCHAR(50) PATH '$')
) AS j_query
WHERE
    t.products_id IS NOT NULL
    AND TRIM(t.products_id) <> ''
    AND j_query.product_id REGEXP '^[0-9]+$';
```

Output

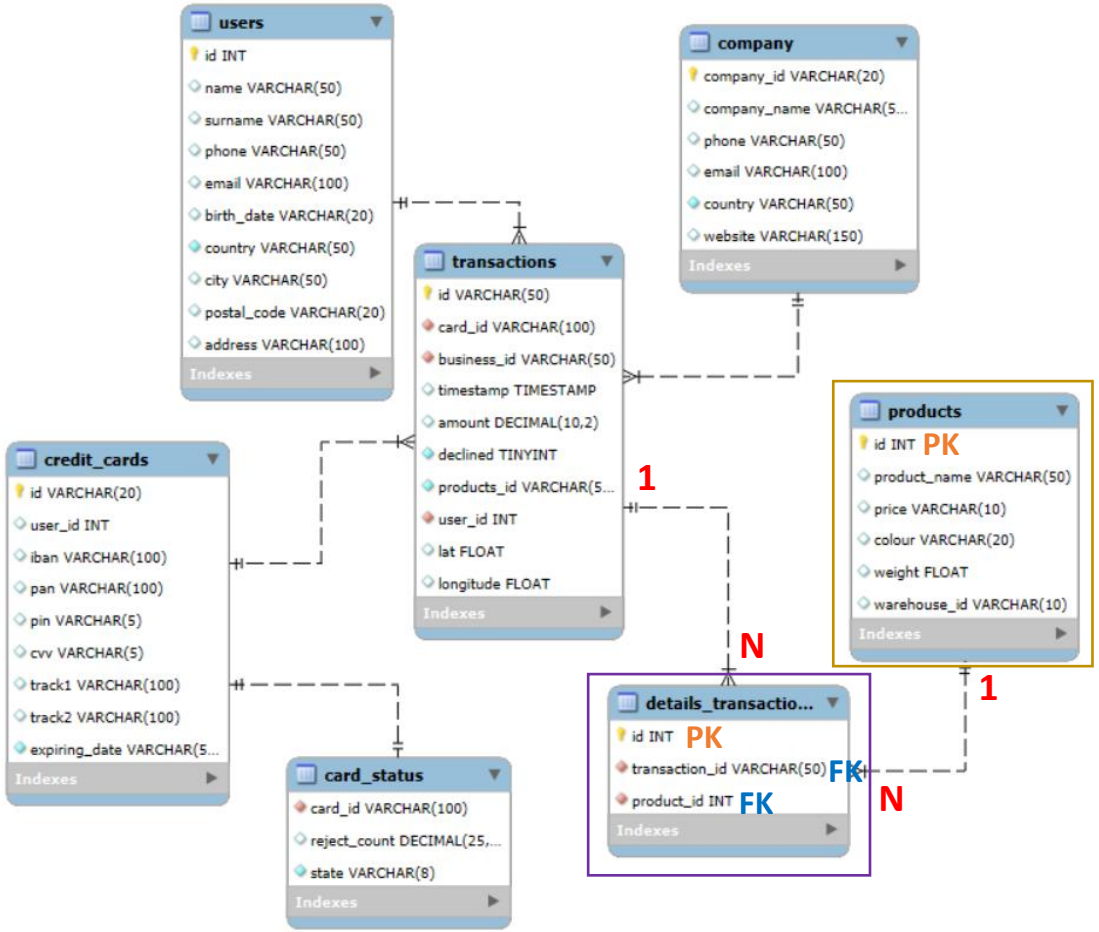
Action Output

#	Time	Action	Message
511	11:56:33	CREATE TABLE details_transactions (id INT AUTO_INCREMENT PRIMARY KEY, transaction_id VARCH...	0 row(s) affected
512	11:56:37	INSERT INTO details_transactions (transaction_id,product_id) SELECT t.id, j_query.product_id FROM transac...	253391 row(s) affected Records: 253391 Duplicates: 0 Warnings: 0

SPRINT 4

NIVEL 3

Análisis de relaciones



- Para la correcta conexión con la tabla ‘products’, era necesario la creación de una nueva tabla donde se viera el desglose de cada transacción; concretamente que id de productos contenía de manera separada. Dicha tabla de hechos intermedia es la de ‘details_transactions’, que nos aporta un nivel de detalle más concreto del contenido de cada transacción.

- La relacionamos con la tabla de hechos mediante el ‘id’ de ‘transactions’, siendo la cardinalidad de 1 a muchas (una transacción aparecerá en repetidas ocasiones en caso de contener más de un tipo de producto). Se relaciona mediante la FK de ‘transactions_id’.

- Por otro lado, la nueva tabla se relaciona con la dimensión de ‘products’ mediante la FK ‘producto_id’, porque ésta contiene la información concreta de cada producto. Es por ello que la cardinalidad entre las dos tablas es de muchos a 1 (un producto puede salir repetidas veces en ‘details_transactions’).

3.1

- Necesitamos saber el número de veces que se ha vendido cada producto

```
198     -- 3.1 Veces que se ha vendido cada producto
199 •   SELECT p.product_name as Producto, COUNT(dt.product_id) as Venta_productos
200     FROM details_transactions dt
201     JOIN products p
202     ON p.id = dt.product_id
203     GROUP BY p.id, Producto
204     ORDER BY Venta_productos DESC;
```

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

Producto	Venta_productos
riverlands the duel	2654
Tully maester Tarly	2635
duel Direwolf	2609
the duel warden	2608
mustafar jinn	2601
sith Jade	2598
duel warden	2597
rock Renly in	2597
riverlands north	2593
Stark Karstark	2589
Stannis warden so...	2587
warden south duel	2584
Dorne bastard	2584
chewbacca mustafar	2584
of north	2580
Lannister	2580

Result 75

Output

Action Output

#	Time	Action	Message
341	10:12:05	SELECT t.id, j_query.product_id FROM transactions t JOIN JSON_TABLE(CONCAT('[', REGEXP_REPLAC...	253391 row(s) returned