```sql
CREATE TABLE transactions (
    id  VARCHAR(100) PRIMARY KEY NOT NULL,
    card_id VARCHAR(50) NULL,
    business_id VARCHAR(50) NULL,
    `timestamp` TIMESTAMP NULL,
    amount DECIMAL(10,2) NULL,
    declined TINYINT(1) NOT NULL,
    product_ids VARCHAR(100) NULL,
    user_id VARCHAR(100) NULL,
    lat  VARCHAR(100) NULL,
    longitude VARCHAR(100) NULL
);
-- ----- Cargamos los datos de transactions.
LOAD DATA
INFILE "C://transactions.csv"
INTO TABLE transactions
FIELDS TERMINATED BY ';'
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

1

Output

Action Output

#	Time	Action	Message
1	12:48:04	LOAD DATA INFILE "C://transactions.csv" -- "C://Users//Usuario//OneDrive//Documentos//IT ACADEMY//tran...	100000 row(s) affected
2	12:49:35	SELECT * FROM tienda_online.transactions	100000 row(s) returned

```sql
SELECT *
FROM transactions;
```

2

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

id	card_id	business_id	timestamp	amount	declined	product_ids	user_id	lat	longitude
00043A49-2949-494B-A5DD-A5BAE3BB19DD	CcS-9294	b-2458	2024-08-28 07:16:46	395.43	0	16, 26, 97, 87	4713	46.19992926158272	1.4355402821327607
000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	CcS-5019	b-2370	2016-12-21 20:07:18	155.63	0	66, 69, 87	438	41.59720554463741	12.22175994259365
00045D6B-ED2E-4F2F-8186-CEE074D875D0	CcS-6699	b-2390	2020-07-14 15:37:45	326.01	0	30, 11, 16, 81	2118	29.757295899964348	-95.37963676492454
000481C3-1C26-4FEF-83A0-4CD0EB004BBD	CcS-6696	b-2230	2017-09-04 19:44:53	161.60	0	72	2115	53.54888376797025	-113.50305274646564
00051AA4-9CBE-4268-B070-C38062A1B3E2	CcS-7606	b-2266	2017-01-05 18:19:25	148.91	0	18	3025	52.20836951654172	5.690806474241335
0008A312-EDFE-4A4F-BC99-E9C92EC3CA4D	CcU-3358	b-2598	2023-09-23 04:51:43	294.59	0	35, 33, 19	215	53.553485560256014	-113.4991339874781
0009A151-9BCF-4E31-9053-A468FF77FAAB	CcS-7509	b-2546	2023-12-31 00:06:36	383.63	0	93, 55, 28, 91	2928	51.93615735576977	5.342650184655181
0009D494-6245-4DF9-955D-2C084191CFFB	CcS-8483	b-2526	2017-07-18 07:52:02	197.80	0	55, 8, 72	3902	45.49200875032752	-73.57063686984957

transactions 2

Output

Action Output

#	Time	Action	Message
1	13:02:21	SELECT * FROM transactions	100000 row(s) returned

Creamos la tabla “transactions” y le insertamos los datos del archivo CSV(transactions).

para comprobar mostramos la tabla.

```sql
CREATE TABLE companies (
    company_id VARCHAR(100) PRIMARY KEY NOT NULL,
    company_name VARCHAR(50) NULL,
    phone VARCHAR(50) NULL,
    email VARCHAR(50) NULL,
    country VARCHAR(50) NULL,
    website VARCHAR(50) NULL
);
-- ----- Cargamos los de companies y comprobamos.
LOAD DATA
INFILE "C://companies.csv"
INTO TABLE companies
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

1

Output

Action Output

#	Time	Action	Message
1	13:15:04	CREATE TABLE companies (company_id VARCHAR(100) PRIMARY KEY NOT NULL, company_name VARCHA...	0 row(s) affected
2	13:15:25	LOAD DATA INFILE "C://companies.csv" INTO TABLE companies FIELDS TERMINATED BY ',' ENCLOSED BY '"' ...	100 row(s) affected Records:

```sql
SELECT *
FROM companies;
```

2

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

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.com/site
b-2226	Magna A Neque Industries	04 14 44 64 62	risus.donec.nibh@icloud.org	Australia	https://whatsapp.com/group/9
b-2230	Fusce Corp.	08 14 97 58 85	risus@protonmail.edu	United States	https://pinterest.com/sub/cars
b-2234	Convallis In Incorporated	06 66 57 29 50	mauris.ut@aol.couk	Germany	https://cnn.com/user/110
b-2238	Ante Iaculis Nec Foundation	08 23 04 99 53	sed.dictum.proin@outlook.ca	New Zealand	https://netflix.com/settings
b-2242	Donec Ltd	01 25 51 37 37	at.iaculis@hotmail.couk	Norway	https://nytimes.com/user/110

companies 3

Output

Action Output

#	Time	Action	Message
1	13:18:10	SELECT * FROM companies	100 row(s) returned

Creamos la tabla “companies” y le insertamos los datos del archivo CSV(companies).

para comprobar mostramos la tabla.

1

```sql
CREATE TABLE credit_cards (
    id VARCHAR(100) PRIMARY KEY NOT NULL,
    user_id VARCHAR(50) NULL,
    iban VARCHAR(100) NULL,
    pan VARCHAR(100) NULL,
    pin VARCHAR(50) NULL,
    cvv VARCHAR(50) NULL,
    track1 VARCHAR(250) NULL,
    track2 VARCHAR(250) NULL,
    expiring_date VARCHAR(50) NULL
);
-- ----- Cargamos los de credit_cards y comprobamos.
LOAD DATA
INFILE "C://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
1	13:37:44	CREATE TABLE credit_cards (id VARCHAR(100) PRIMARY KEY NOT NULL, user_id VARCHAR(50) NULL, iban ...	0 row(s) affected
2	13:39:03	LOAD DATA INFILE "C://credit_cards.csv" INTO TABLE credit_cards FIELDS TERMINATED BY ',' ENCLOSED BY '"' ...	5000 row(s) affected

2

```sql
SELECT *
FROM credit_cards;
```

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

id	user_id	iban	pan	pin	cvv	track1	track2	expiring_date
CcS-4857	276	XX4857591835292505850771	2314242385113924	1819	467	%B2314242385113924^LWCBUD...	%B2314242385113924=2410...	09/27/25
CcS-4858	277	XX8581768137002436094025	6582720299715533	3964	817	%B6582720299715533^TIQMVIT...	%B6582720299715533=2411...	12/28/28
CcS-4859	278	XX7826930491423553609370	8861684536289642	4983	277	%B8861684536289642^COFBGD...	%B8861684536289642=2502...	11/26/26
CcS-4860	279	XX5559590368835304645299	2481155515498459	6876	661	%B2481155515498459^TIUJTUT...	%B2481155515498459=2602...	07/27/27
CcS-4861	280	XX2035182877195191627307	1308930301149557	5710	398	%B1308930301149557^HPOBNZ...	%B1308930301149557=2805...	04/25/26

credit_cards 5

Output

Action Output

#	Time	Action	Message
1	13:43:23	SELECT * FROM credit_cards	5000 row(s) returned

Creamos la tabla “credit_cards” y le insertamos los datos del archivo CSV(credit_cards).

para comprobar mostramos la tabla.

```sql
CREATE TABLE users (
    id VARCHAR(100) PRIMARY KEY NOT NULL,
    `name` VARCHAR(100) NULL,
    surname VARCHAR(100) NULL,
    phone VARCHAR(100) NULL,
    email VARCHAR(100) NULL,
    birth_date VARCHAR(50) NULL,
    country VARCHAR(50) NULL,
    city VARCHAR(50) NULL,
    postal_code VARCHAR(100) NULL,
    address VARCHAR(150) NULL
);
```

1

Output

Action Output

#	Time	Action	Message
1	16:14:17	CREATE TABLE users (id VARCHAR(100) PRIMARY KEY NOT NULL, `name` VARCHAR(100) NULL, surname...	0 row(s) affected

```sql
INFILE "C://american_users.csv"
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
-- ----- Cargamos los datos de european_users y comprobamos.
LOAD DATA
INFILE "C://european_users.csv"
INTO TABLE users
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
-- -----
SELECT *
FROM users;
```

2

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

id	name	surname	phone	email	birth_date	country	city	postal_code	address
1	Zeus	Gamble	1-282-581-0551	interdum.enim@prot...	Nov 17, 1985	United States	New York	10001	348-7818 Sagittis St.
10	Robert	Mccarthy	(324) 746-6771	fermentum@proton...	Apr 30, 1984	United States	San Jose	95101	P.O. Box 773
100	Melodie	Mclean	1-677-221-7152	risus.varius@google.ca	Sep 15, 1989	United States	San Jose	95101	Ap #644-8492 Sag...
1000	Amkjrv	Qbulrxbp	+48-258-9936	amkjrv.qbulrxbp@ex...	May 17, 1970	Germany	Stuttgart	70173	215 Qbulrxbp St
1001	Nfvrlb	Oydaiwbg	+94-121-2522	nfvrlb.oydaiwbg@ex...	Mar 4, 1994	Germany	Cologne	50667	121 Oydaiwbg St

users 9

Output

Action Output

#	Time	Action	Message
1	16:18:34	LOAD DATA INFILE "C://american_users.csv" INTO TABLE users FIELDS TERMINATED BY ',' ENCLOSED BY ""...	1010 row(s) affected Records:
2	16:18:37	LOAD DATA INFILE "C://european_users.csv" INTO TABLE users FIELDS TERMINATED BY ',' ENCLOSED BY ""...	3990 row(s) affected Records:
3	16:18:44	SELECT * FROM users	5000 row(s) returned

Creamos la tabla “users”
y le insertamos los datos de los archivos
CSV(american_users)
y
CSV(european_users).

para comprobar mostramos la tabla.

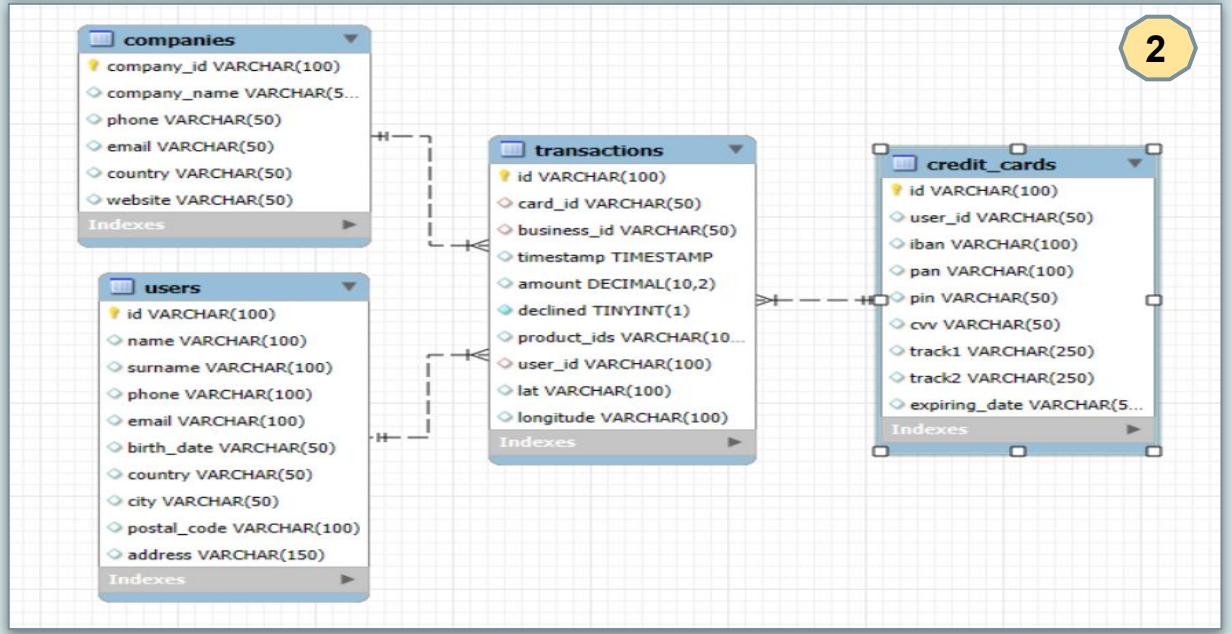
```sql
ALTER TABLE transactions
    ADD CONSTRAINT  FK_business_id
    FOREIGN KEY (business_id) REFERENCES companies(company_id);
-- -----
ALTER TABLE transactions
    ADD CONSTRAINT  FK_card_id
    FOREIGN KEY (card_id) REFERENCES credit_cards(id);
-- -----
ALTER TABLE transactions
    ADD CONSTRAINT  FK_user_id
    FOREIGN KEY (user_id) REFERENCES users(id);
```

1

Output

Action Output

#	Time	Action	Message
1	16:40:49	ALTER TABLE transactions ADD CONSTRAINT FK_business_id FOREIGN KEY (business_id) REFERENCES comp...	100000 row(s)
2	16:40:53	ALTER TABLE transactions ADD CONSTRAINT FK_card_id FOREIGN KEY (card_id) REFERENCES credit_...	100000 row(s)
3	16:41:03	ALTER TABLE transactions ADD CONSTRAINT FK_user_id FOREIGN KEY (user_id) REFERENCES users(id)	100000 row(s)



Una vez creadas las tablas, añadimos las FK a la tabla de hechos “transactions” para relacionarla con cada id de las tablas de dimensión.

Para comprobar mostramos el diagrama.

```sql
SELECT
    u.id AS id_usuario,
    CONCAT(u.`name`," ", u.surname) AS nombre_completo
FROM users u
WHERE EXISTS( SELECT COUNT(t.id) AS num_transacciones
              FROM transactions t
              WHERE t.user_id = u.id
              AND t.declined = 0
              HAVING num_transacciones > 80
              )
;
```

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

id_usuario	nombre_completo
185	Molly Gilliam
289	Dxwgi Hwcru
318	Bnyr Astuw

Result 66

Output

Action Output

#	Time	Action	Message
1	14:06:03	SELECT u.id AS id_usuario, CONCAT(u.`name`," ", u.surname) AS nombre_completo FROM users u WHERE EX...	3 row(s) returned

N1.1

Usamos una query y una subquery que muestra a todos los usuarios con más de 80 transacciones.

Usamos la función concat para juntar las columnas “name” y “surname”. en una misma llamada “nombre_completo”.

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

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

iban	media_monto
XX38301781391962019936...	680.69
XX63770635739757039497...	680.01
XX97139397146529220231...	645.46
XX17184711692889237596...	628.89
XX22542463881854240622...	608.68
XX74889072905719571176...	607.29
TN96145635706673818931 22	605.41

Result 49

Output

Action Output

#	Time	Action	Message
1	12:29:52	SELECT cc.iban, ROUND(AVG(t.amount) ,2) AS media_monto FROM credit_cards cc JOIN transactions t ON c...	370 row(s) returned

N1.1

Usamos una query con JOINS de varias tablas para mostrar la media de amount por IBAN de las tarjetas de crédito en la compañía "Donec Ltd".

N2

```sql
CREATE TABLE estado_de_tarjeta (
    id_tarjeta VARCHAR(100) NOT NULL PRIMARY KEY,
    estado VARCHAR(50) NOT NULL,
    FOREIGN KEY (id_tarjeta) REFERENCES credit_cards(id)
);
-- ----- Insertamos los datos filtrados.
INSERT INTO estado_de_tarjeta (id_tarjeta, estado)
WITH crs AS (SELECT
                t.card_id,
                t.declined,
                ROW_NUMBER() OVER(PARTITION BY t.card_id ORDER BY t.`timestamp` DESC) AS num_transaccion
            FROM transactions t)
SELECT
    crs.card_id AS id_tarjeta,
    CASE
        WHEN SUM(crs.declined) = 3 THEN 'INACTIVA'
        ELSE 'ACTIVA'
    END AS estado
FROM crs
WHERE num_transaccion <= 3
GROUP BY id_tarjeta;
```

1

Output

Action Output

#	Time	Action	Message
1	13:12:29	CREATE TABLE estado_de_tarjeta (id_tarjeta VARCHAR(100) NOT NULL PRIMARY KEY, estado VARCHAR(...	0 row(s) affected
2	13:12:59	INSERT INTO estado_de_tarjeta (id_tarjeta, estado) WITH crs AS (SELECT t.card_id, t.declined, ROW_NUMBER() ...	5000 row(s) affected Records: 5000 Duplicates: 0 Warnings: 0

Usamos una query con Window Function para obtener las últimas transacciones de cada tarjeta.

Mediante el condicional CASE clasificamos en:
ACTIVA si al menos una de la últimas 3 transacciones ha sido aprobada.
INACTIVA si las ultimas 3 han sido declinadas.
Finalmente mostramos el número de tarjetas activas.

```sql
SELECT *
FROM estado_de_tarjeta;
```

2

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

id_tarjeta	estado
CcS-4857	ACTIVA
CcS-4858	ACTIVA
CcS-4859	ACTIVA
CcS-4860	ACTIVA
CcS-4861	ACTIVA

estado_de_tarjeta 70

Output

Action Output

#	Time	Action	Message
1	14:15:48	SELECT * FROM estado_de_tarjeta	5000 row(s) returned

```sql
SELECT COUNT(estado) AS num_tarjetas_activas
FROM estado_de_tarjeta
WHERE estado = 'ACTIVA';
```

3

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

num_tarjetas_activas
4995

Result 72

Output

Action Output

#	Time	Action	Message
1	14:21:59	SELECT COUNT(estado) AS num_tarjetas_activas FROM estado_de_tarjeta WHERE estado = 'ACTIVA'	1 row(s) returned

```sql
CREATE TABLE products (
    id VARCHAR(100) PRIMARY KEY NOT NULL,
    product_name VARCHAR(100) NULL,
    price VARCHAR(50) NULL,
    colour VARCHAR(100) NULL,
    weight DECIMAL(10,2) NULL,
    warehouse_id VARCHAR(100) NULL
);
-- ----- Cargamos los datos de products y comprobamos.
LOAD DATA
INFILE "C://products.csv"
INTO TABLE products
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

1

Output

Action Output

#	Time	Action	Message
1	14:29:41	CREATE TABLE products (id VARCHAR(100) PRIMARY KEY NOT NULL, product_name VARCHAR(100) NULL, ...	0 row(s) affected
2	14:30:17	LOAD DATA INFILE "C://products.csv" INTO TABLE products FIELDS TERMINATED BY ',' ENCLOSED BY '"' LIN...	100 row(s) affected Records:

```sql
SELECT *
FROM products;
```

2

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

id	product_name	price	colour	weight	warehouse_id
1	Direwolf Stannis	$161.11	#7c7c7c	1.00	WH-4
10	Karstark Dorne	$119.52	#f4f4f4	2.40	WH--5
100	south duel	$40.43	#6d6d6d	3.00	WH--95
11	Karstark Dorne	$49.70	#141414	2.70	WH--6
12	duel Direwolf	$181.60	#a8a8a8	2.10	WH--7
13	palpatine chewbacca	$139.59	#2b2b2b	1.00	WH--8
14	Direwolf	$147.53	#c4c4c4	2.00	WH--9
15	Stannis warden	$194.29	#dbdbdb	1.50	WH--10
16	the duel warden	$180.91	#666666	3.00	WH--11

products 8

Output

Action Output

#	Time	Action	Message
1	14:32:25	SELECT * FROM products	100 row(s) returned

N3

Creamos la tabla “products” y le insertamos los datos del archivo CSV(products).

para comprobar mostramos la tabla.

N3

Usamos una query con Window Function para separar los porducts_ids de cada transacción en cada id_producto en cada transacción.
Usamos JSON_TABLE para separar.
Insertamos en la tabla intermedia que hemos creado y comprobamos.
Finalmente mostramos el número de ventas de cada producto.

1

```sql
CREATE TABLE transactions_products (
    id_transaction VARCHAR(100) NOT NULL ,
    id_product VARCHAR(100) NOT NULL,
    FOREIGN KEY (id_transaction) REFERENCES transactions(id),
    FOREIGN KEY (id_product) REFERENCES products(id) )
;
-- ----- extraemos los datos de la columna "transactions.product_ids" e insetamos en la tabala inermedia "transactions_products".
INSERT INTO transactions_products (id_transaction,id_product)
WITH temporal_table AS (    SELECT
                                t.id AS id_transaction,
                                ep.id_product
                            FROM transactions t,
                            JSON_TABLE (    CONCAT('[',t.product_ids,']'),
                                            '$[*]'
                                            COLUMNS (
                                                id_product INT PATH '$')
                                        )   AS ep )
SELECT *
FROM temporal_table;
```

Output

Action Output

#	Time	Action	Message
1	11:22:30	CREATE TABLE transactions_products (id_transaction VARCHAR(100) NOT NULL , id_product VARCHAR(100) ...	0 row(s) affected
2	11:22:39	INSERT INTO transactions_products (id_transaction,id_product) WITH temporal_table AS (SELECT t.id AS id_trans...	253391 row(s) affected Records: 253391 Duplicates: 0 Warnings: 0

2

```sql
SELECT *
FROM transactions_products;
```

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

id_transaction	id_product
00043A49-2949-494B-A5DD-A5BAE3BB19DD	16
00043A49-2949-494B-A5DD-A5BAE3BB19DD	26
00043A49-2949-494B-A5DD-A5BAE3BB19DD	97
00043A49-2949-494B-A5DD-A5BAE3BB19DD	87
000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	66
000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	69
000447FE-B650-4DCF-85DE-C7ED0EE1CAAD	87

transactions_products 25

Output

Action Output

#	Time	Action	Message
1	19:12:15	SELECT * FROM transactions_products	253391 row(s) returned

3

```sql
SELECT
    p.product_name AS nombre_producto,
    tp.id_product AS id_producto,
    COUNT( tp.id_product) AS num_ventas
FROM transactions_products tp
JOIN transactions t
    ON tp.id_transaction = t.id
JOIN products p
    ON tp.id_product = p.id
WHERE t.declined = 0
GROUP BY nombre_producto,
         id_producto
ORDER BY num_ventas DESC;
```

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

nombre_producto	id_producto	num_ventas
riverlands the duel	52	2642
Tully maester Tarly	29	2627
duel Direwolf	21	2603
the duel warden	16	2602
duel warden	33	2593
sith Jade	87	2591

Result 45

Output

Action Output

#	Time	Action	Message
1	12:10:14	SELECT p.product_name AS nombre_producto, tp.id_product AS id_producto, COUNT(tp.id_product) AS num_v...	100 row(s) returned