IT Academy – Datos
Sprint 4 – 25/2/25
Lucía Cané

Descarga los archivos CSV, estudiales y diseña una base de datos con un esquema de estrella que contenga, al menos 4 tablas de las que puedas realizar las siguientes consultas:

```sql
CREATE DATABASE sprint4;
USE sprint4;

CREATE TABLE companies (
company_id CHAR(6) PRIMARY KEY,
company_name VARCHAR(100),
phone VARCHAR(50),
email VARCHAR(100),
country VARCHAR(100),
website VARCHAR(100));

SHOW VARIABLES LIKE "secure_file_priv";
# Moví los archivos .csv a la carpeta que figura en el output: 'C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\'

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

Output

Action Output

#	Time	Action	Message
1	10:11:51	DROP DATABASE sprint4	7 row(s) affected
2	10:12:23	CREATE DATABASE sprint4	1 row(s) affected
3	10:12:26	USE sprint4	0 row(s) affected
4	10:12:28	CREATE TABLE companies (company_id CHAR(6) PRIMARY KEY, company_name VARCHAR(100), ph...	0 row(s) affected
5	10:12:32	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/companies.csv' INTO TABLE...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0

```sql
CREATE TABLE creditcard(
id CHAR(8) PRIMARY KEY,
user_id INT,
iban VARCHAR(100),
pan VARCHAR(100),
pin CHAR(4),
cvv CHAR(3),
track1 VARCHAR(100),
track2 VARCHAR(100),
expiring_date CHAR(8));

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv'
INTO TABLE creditcard
FIELDS TERMINATED BY ','
IGNORE 1 ROWS;

UPDATE creditcard
SET expiring_date = DATE_FORMAT(STR_TO_DATE(expiring_date, '%m/%d/%y'), '%d-%m-%y');

ALTER TABLE creditcard
MODIFY COLUMN expiring_date DATE;
```

Output

Action Output

#	Time	Action	Message
6	10:13:59	CREATE TABLE creditcard(id CHAR(8) PRIMARY KEY, user_id INT, iban VARCHAR(100), pan VARC...	0 row(s) affected
7	10:14:03	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/credit_cards.csv' INTO TAB...	275 row(s) affected Records: 275 Deleted: 0 Skipped: 0 Warnings: 0
8	10:14:12	UPDATE creditcard SET expiring_date = DATE_FORMAT(STR_TO_DATE(expiring_date, '%m/%d/%y'),...	275 row(s) affected Rows matched: 275 Changed: 275 Warnings: 0
9	10:14:14	ALTER TABLE creditcard MODIFY COLUMN expiring_date DATE	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

```sql
CREATE TABLE products (
id INT PRIMARY KEY,
product_name VARCHAR(100),
price VARCHAR(100),
colour VARCHAR(100),
weight VARCHAR(100),
warehouse_id VARCHAR(100));

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

CREATE TABLE transactions (
id VARCHAR(150) PRIMARY KEY,
card_id VARCHAR(10),
business_id CHAR(6),
timestamp timestamp,
amount DECIMAL(8,2),
declined BOOL,
product_ids VARCHAR(150),
user_id INT,
lat VARCHAR(150),
longitude VARCHAR(150));
```

Output

Action Output

#	Time	Action	Message
10	10:15:06	CREATE TABLE products (id INT PRIMARY KEY, product_name VARCHAR(100), price VARCHAR(10...	0 row(s) affected
11	10:15:08	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/products.csv' INTO TABLE ...	100 row(s) affected Records: 100 Deleted: 0 Skipped: 0 Warnings: 0
12	10:15:10	CREATE TABLE transactions (id VARCHAR(150) PRIMARY KEY, card_idVARCHAR(10), business_idC...	0 row(s) affected

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

CREATE TABLE users (
id INT PRIMARY KEY,
name VARCHAR(150),
surname VARCHAR(150),
phone VARCHAR(150),
email VARCHAR(150),
birth_date VARCHAR(150),
country VARCHAR(150),
city VARCHAR(150),
postal_code VARCHAR(150),
address VARCHAR(150));

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

Output

Action Output

#	Time	Action	Message
13	10:15:55	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/transactions.csv' INTO TAB...	587 row(s) affected Records: 587 Deleted: 0 Skipped: 0 Warnings: 0
14	10:15:57	CREATE TABLE users (id INT PRIMARY KEY, name VARCHAR(150), surname VARCHAR(150), phon...	0 row(s) affected
15	10:16:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_ca.csv' INTO TABLE...	75 row(s) affected Records: 75 Deleted: 0 Skipped: 0 Warnings: 0

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

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

UPDATE users
SET birth_date = DATE_FORMAT(STR_TO_DATE(birth_date, '%b %d, %Y'), '%Y-%m-%d');

ALTER TABLE users
MODIFY COLUMN birth_date DATE;

SET FOREIGN_KEY_CHECKS=0;
```

Output

Action Output

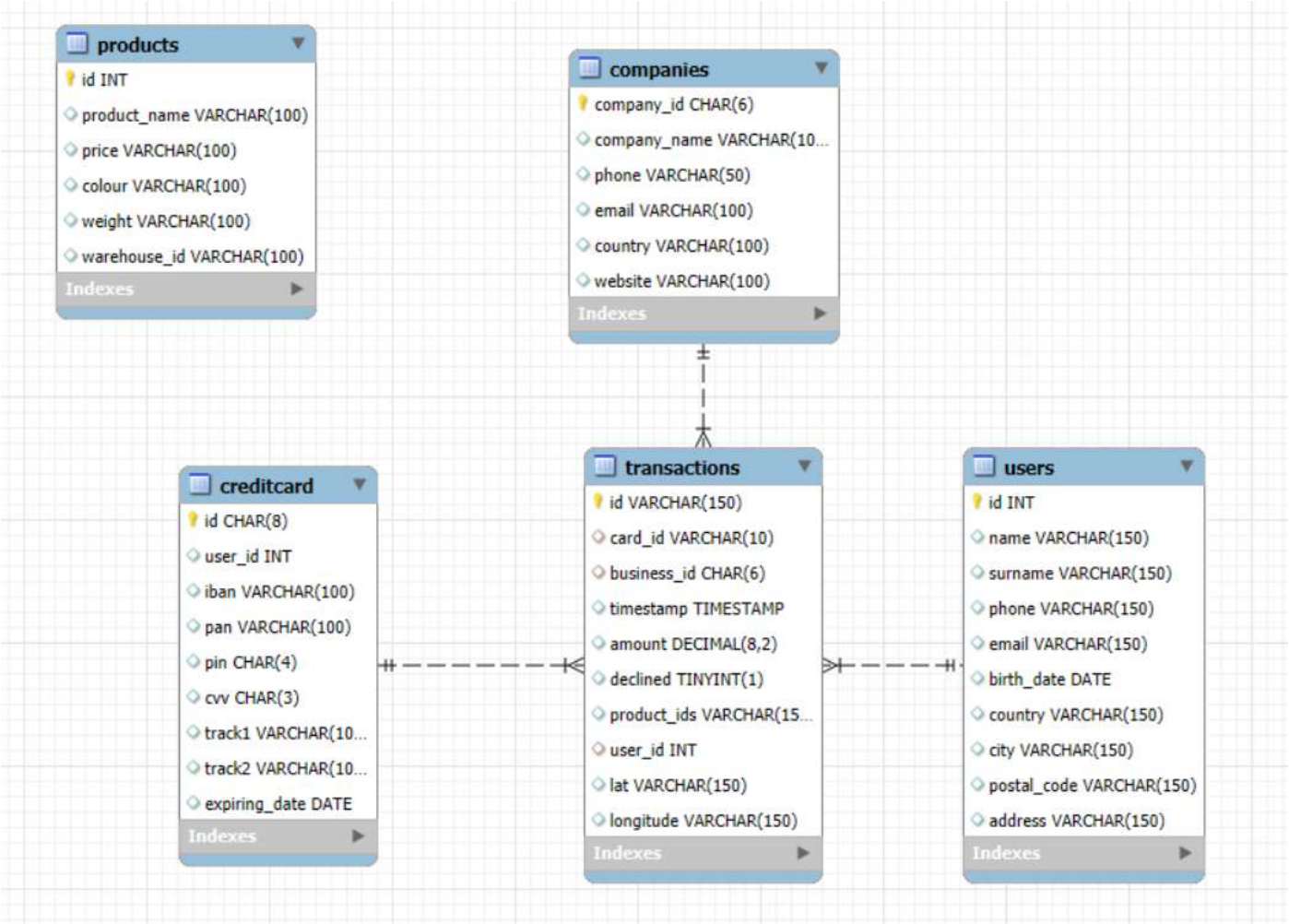
#	Time	Action	Message
16	10:16:50	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_uk.csv' INTO TABLE...	50 row(s) affected Records: 50 Deleted: 0 Skipped: 0 Warnings: 0
17	10:16:52	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/users_usa.csv' INTO TABL...	150 row(s) affected Records: 150 Deleted: 0 Skipped: 0 Warnings: 0
18	10:16:56	UPDATE users SET birth_date = DATE_FORMAT(STR_TO_DATE(birth_date, '%b %d, %Y'), '%Y-%m-%d')	275 row(s) affected Rows matched: 275 Changed: 275 Warnings: 0
19	10:17:00	ALTER TABLE users MODIFY COLUMN birth_date DATE	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0
20	10:17:10	SET FOREIGN_KEY_CHECKS=0	0 row(s) affected

```
114 •   ALTER TABLE transactions
115     ADD FOREIGN KEY (card_id) REFERENCES creditcard(id);
116 •   ALTER TABLE transactions
117     ADD FOREIGN KEY (user_id) REFERENCES users(id);
118 •   ALTER TABLE transactions
119     ADD FOREIGN KEY (business_id) REFERENCES companies(company_id);
120
```

Output

Action Output

#	Time	Action	Message
25	10:19:39	ALTER TABLE transactions ADD FOREIGN KEY (card_id) REFERENCES creditcard(id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
26	10:19:42	ALTER TABLE transactions ADD FOREIGN KEY (user_id) REFERENCES users(id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
27	10:19:46	ALTER TABLE transactions ADD FOREIGN KEY (business_id) REFERENCES companies(company_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0



Nivel 1 - Ejercicio 1

Realiza una subconsulta que muestre a todos los usuarios con más de 30 transacciones utilizando al menos 2 tablas.

```sql
# Nivel 1 - Ejercico 1

SELECT id, name
FROM users
WHERE id IN(SELECT user_id FROM transactions
GROUP BY user_id HAVING COUNT(user_id) > 30 );
```

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

id	name
92	Lynn
267	Ocean
272	Hedwig
275	Kenyon
NULL	NULL

users 2

Output

Action Output

#	Time	Action	Message
31	10:22:01	ALTER TABLE transactions DROP FOREIGN KEY transactions_jbfk_6	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
32	10:23:38	SELECT id, name FROM users WHERE id IN(SELECT user_id FROM transactions GROUP BY user_j...	4 row(s) returned

Nivel 1 - Ejercicio 2

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

```sql
# Nivel 1 - Ejercico 2
SELECT c.iban AS IBAN, t.declined AS Declined, ROUND(AVG(t.amount),2) AS Amount
FROM transactions AS t
JOIN companies AS co ON co.company_id=t.business_id
JOIN creditcard AS c ON t.card_id=c.id
WHERE co.company_name= "Donec Ltd"
GROUP BY c.iban, c.id, co.company_name, c.iban, t.declined;
```

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

IBAN	Declined	Amount
PT87806228135092429456346	1	364.61
PT87806228135092429456346	0	42.82

Result 3

Output

Action Output

#	Time	Action	Message
32	10:23:38	SELECT id, name FROM users WHERE id IN(SELECT user_id FROM transactions GROUP BY user_i...	4 row(s) returned
33	10:24:46	SELECT c.iban AS IBAN, t.declined AS Declined, ROUND(AVG(t.amount),2) AS Amount FROM transa...	2 row(s) returned

Nivel 2 - Ejercicio 1

Crea una nueva tabla que refleje el estado de las tarjetas de crédito basado en si las últimas tres transacciones fueron declinadas y genera la siguiente consulta:

¿Cuántas tarjetas están activas?

```sql
# Nivel 2 - Ejercico 1
CREATE TABLE creditcard_state (
card_id CHAR(8),
last3 INT,
state VARCHAR(50));

ALTER TABLE creditcard_state
ADD PRIMARY KEY (card_id);

ALTER TABLE creditcard
ADD FOREIGN KEY (id) REFERENCES creditcard_state(card_id);

INSERT INTO creditcard_state (card_id, last3)
SELECT DISTINCT card_id, sum(declined) AS last3
FROM (SELECT card_id, declined, RANK() OVER (partition by card_id ORDER BY timestamp DESC) AS RN
FROM transactions) AS rankedtransactions
WHERE RN <=3
GROUP BY card_id;
```

Output

Action Output

#	Time	Action	Message
34	10:25:32	CREATE TABLE creditcard_state (card_id CHAR(8), last3 INT, state VARCHAR(50))	0 row(s) affected
35	10:25:36	ALTER TABLE creditcard_state ADD PRIMARY KEY (card_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
36	10:25:38	ALTER TABLE creditcard ADD FOREIGN KEY (id) REFERENCES creditcard_state(card_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
37	10:25:41	INSERT INTO creditcard_state (card_id, last3) SELECT DISTINCT card_id, sum(declined) AS last3 FR...	275 row(s) affected Records: 275 Duplicates: 0 Warnings: 0

```sql
UPDATE creditcard_state
SET state = CASE
    WHEN (last3) >= 3 THEN "INACTIVE"
    WHEN (last3) < 3 THEN "ACTIVE"
END;

SELECT count(*) AS Activecards
FROM creditcard_state
WHERE state = "ACTIVE";
```

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

Activecards
275

Result 4

Output

Action Output

#	Time	Action	Message
38	10:26:19	UPDATE creditcard_state SET state = CASE WHEN (last3) >= 3 THEN "INACTIVE" WHEN (last3) < ...	275 row(s) affected Rows matched: 275 Changed: 275 Warnings: 0
39	10:26:21	SELECT count(*) AS Activecards FROM creditcard_state WHERE state = "ACTIVE" LIMIT 0, 50000	1 row(s) returned

Nivel 3 - Ejercicio 1

Crea una tabla con la que podamos unir los datos del nuevo archivo products.csv con la base de datos creada, teniendo en cuenta que desde transaction tienes product_ids. Genera la siguiente consulta:

Necesitamos conocer el número de veces que se ha vendido cada producto.

```sql
# Nivel 3 - Ejercico 1

CREATE TABLE productspertransaction (
transaction_id VARCHAR(150),
product_id INT);

ALTER TABLE productspertransaction
ADD PRIMARY KEY(transaction_id, product_id);
ALTER TABLE productspertransaction
ADD FOREIGN KEY(transaction_id) REFERENCES transactions(id);
ALTER TABLE productspertransaction
ADD FOREIGN KEY (product_id) REFERENCES products(id);

CREATE TEMPORARY TABLE numbers AS
 ( select 1 as n
 union select 2 as n
 union select 3 as n
 union select 4 as n );
```

Output

Action Output

#	Time	Action	Message
40	10:27:16	CREATE TABLE productspertransaction (transaction_id VARCHAR(150), product_id INT)	0 row(s) affected
41	10:27:22	ALTER TABLE productspertransaction ADD PRIMARY KEY(transaction_id, product_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
42	10:27:25	ALTER TABLE productspertransaction ADD FOREIGN KEY(transaction_id) REFERENCES transaction...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
43	10:27:31	ALTER TABLE productspertransaction ADD FOREIGN KEY (product_id) REFERENCES products(id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
44	10:27:45	CREATE TEMPORARY TABLE numbers AS (select 1 as n union select 2 as n union select 3 as n ...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0

```sql
INSERT INTO productspertransaction (transaction_id, product_id)
SELECT
    t.id,
    SUBSTRING_INDEX(SUBSTRING_INDEX(t.product_ids, ',', n), ',', -1) AS product_id
FROM transactions t
JOIN numbers ON (CHAR_LENGTH(t.product_ids) - CHAR_LENGTH(REPLACE(t.product_ids, ',', '')) >= n - 1);

SELECT product_id, COUNT(transaction_id) AS UnitsSold
FROM productspertransaction
GROUP BY product_id
ORDER BY product_id;
```

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

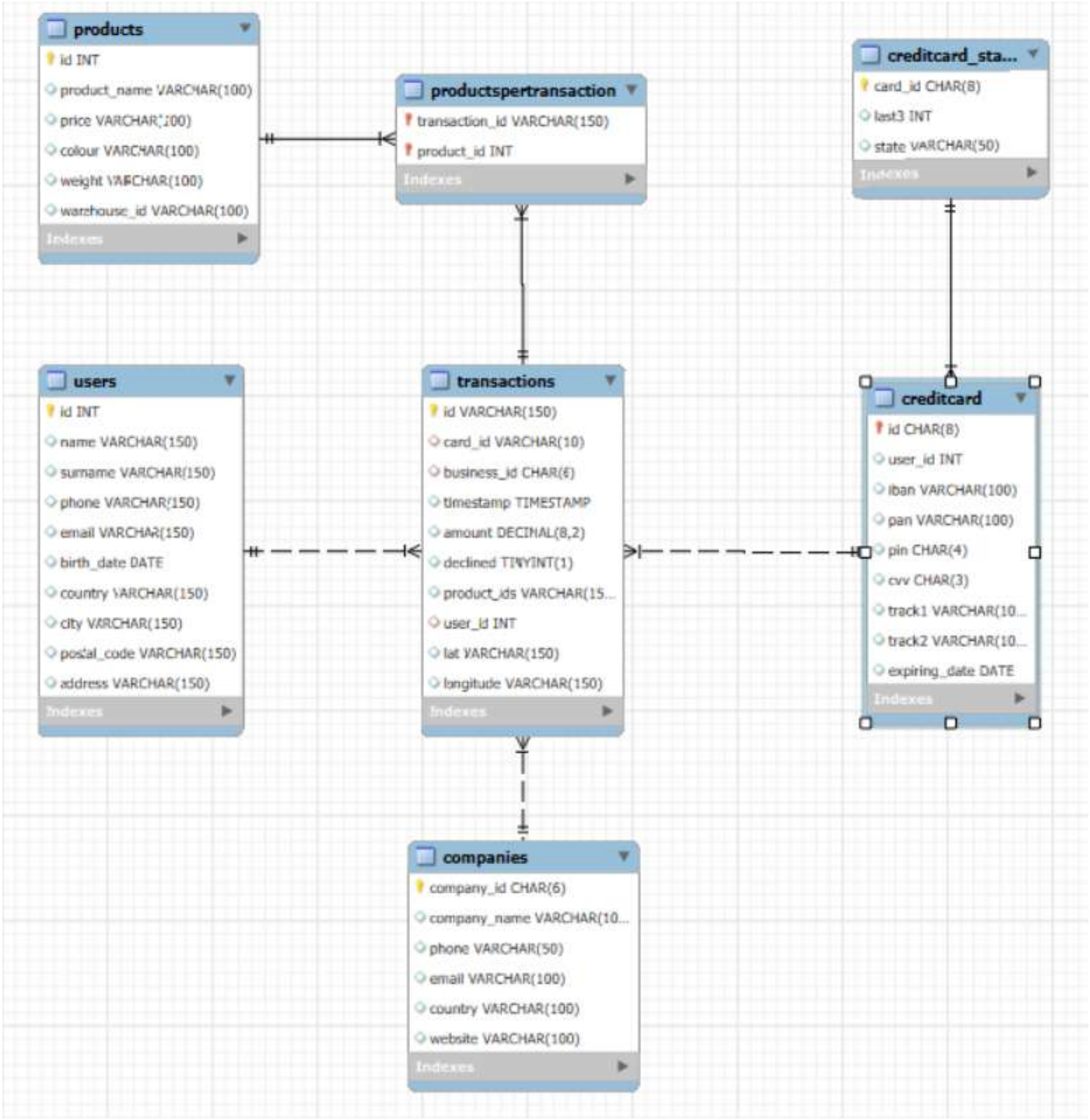
product_id	UnitsSold
1	61
2	65
3	51
5	49
7	54
11	48
13	60
17	61

Result 5

Output

Action Output

#	Time	Action	Message
45	10:29:08	INSERT INTO productspertransaction (transaction_id, product_id) SELECT t.id, SUBSTRING_IN...	1457 row(s) affected Records: 1457 Duplicates: 0 Warnings: 0
46	10:29:14	SELECT product_id, COUNT(transaction_id) AS UnitsSold FROM productspertransaction GROUP BY p...	26 row(s) returned


products
id INT
product_name VARCHAR(100)
price VARCHAR(100)
colour VARCHAR(100)
weight VARCHAR(100)
warehouse_id VARCHAR(100)
Indexes
productspertransaction
transaction_id VARCHAR(150)
product_id INT
Indexes
creditcard_sta...
card_id CHAR(8)
last3 INT
state VARCHAR(50)
Indexes
users
id INT
name VARCHAR(150)
surname VARCHAR(150)
phone VARCHAR(150)
email VARCHAR(150)
birth_date DATE
country VARCHAR(150)
city VARCHAR(150)
postal_code VARCHAR(150)
address VARCHAR(150)
Indexes
transactions
id VARCHAR(150)
card_id VARCHAR(10)
business_id CHAR(6)
timestamp TIMESTAMP
amount DECIMAL(8,2)
declined TINYINT(1)
product_ids VARCHAR(15...
user_id INT
lat VARCHAR(150)
longitude VARCHAR(150)
Indexes
creditcard
id CHAR(8)
user_id INT
iban VARCHAR(100)
pan VARCHAR(100)
pin CHAR(4)
cvv CHAR(3)
track1 VARCHAR(10...
track2 VARCHAR(10...
expiring_date DATE
Indexes
companies
company_id CHAR(6)
company_name VARCHAR(10...
phone VARCHAR(50)
email VARCHAR(100)
country VARCHAR(100)
website VARCHAR(100)
Indexes