

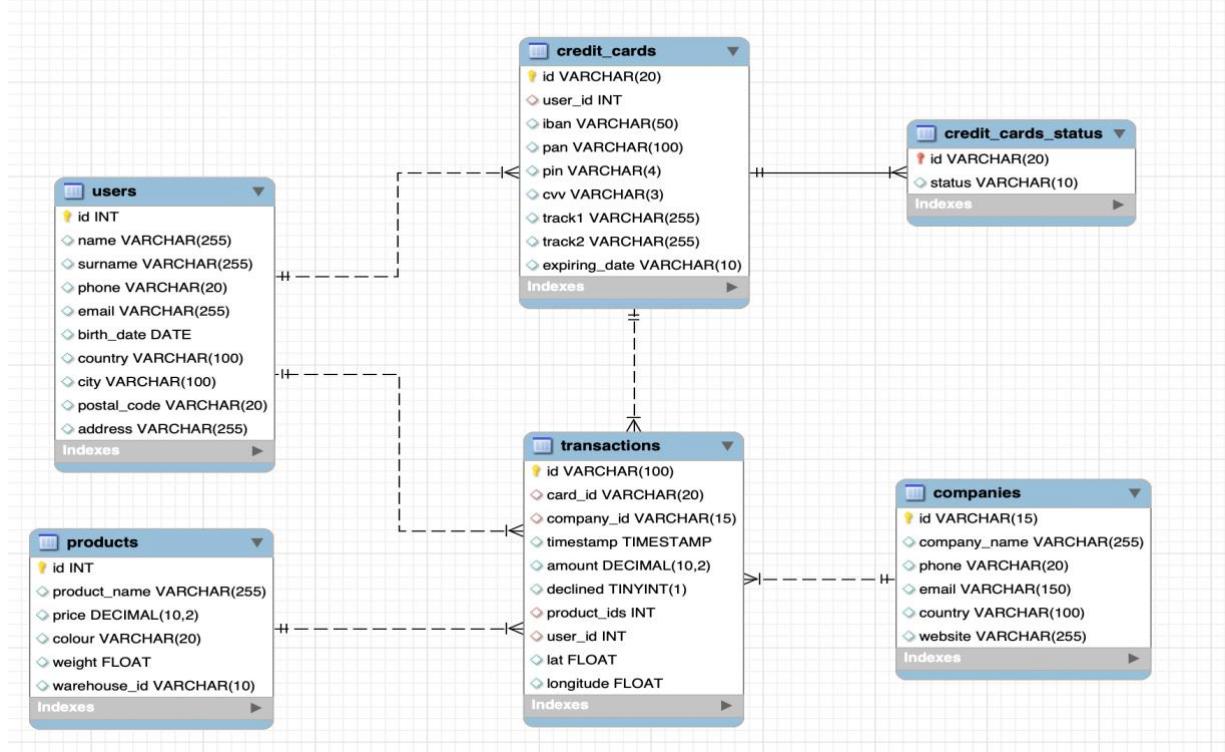
Name: Andre Mazurkiewitz

SPRINT 4

- **Level 1**

With the given .csv files I designed the DB as the following image.

*The products and credit_card_status were not in the initial model, they were created along with the exercises and the image was taken at the end of the exercises.



The data was loaded to MySQL using the following queries, but before I had cleaned the CSVs using Excel to make sure the files were good to be inserted into MySQL.

```
-- INSERTING THE DATA

SET GLOBAL local_infile = 1;

LOAD DATA LOCAL INFILE '/Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint4/companies.csv'
INTO TABLE companies
FIELDS TERMINATED BY ','
ENCLOSED BY ""
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

LOAD DATA LOCAL INFILE '/Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint4/american_users.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
-- ENCLOSED BY ""
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

LOAD DATA LOCAL INFILE '/Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint4/european_users.csv'
INTO TABLE users
FIELDS TERMINATED BY ','
-- ENCLOSED BY ""
LINES TERMINATED BY '\n'
```

- **Level 1 Exercise 1**

```

1 • use sprint4;
2
3 • SELECT count(*) AS num_transactions, (
4     SELECT u.email
5         FROM users AS u
6             WHERE t.user_id = u.id) AS user_email
7     FROM transactions AS t
8     GROUP BY t.user_id
9     HAVING num_transactions > 80
10    ORDER BY num_transactions DESC;

```

Result Grid

num_transactions	user_email
847	ornare.egestas@icloud.edu
834	sodales@yahoo.edu
831	nam@hotmail.com
829	enim commodo@google.ca
821	conubia.nostra@icloud.com
817	eu@google.co.uk

Action Output

Time	Action	Response	Duration / Fetch Time
10:17:18	use sprint4	0 row(s) affected	0.00025 sec
10:17:18	SELECT count(*) AS num_transactions, (SELECT u.email FRO...	100 row(s) returned	0.037 sec / 0.000021...

- **Level 1 Exercise 2**

```

1 • use sprint4;
2
3 • SELECT AVG(t.amount), cc.iban
4     FROM transactions AS t
5     LEFT JOIN companies AS c ON t.company_id = c.id
6     LEFT JOIN credit_cards AS cc ON t.card_id = cc.id
7     WHERE c.company_name = "Donec Ltd"
8     GROUP BY cc.iban

```

result Grid

AVG(t.amount)	iban
356.246667	XX911406401125586307586805
142.960000	SK9446370242474562577506
257.370000	XX77675281784595297555640
139.590000	XX413827362289719304908990
240.410000	XX347787246070769610780308
188.580000	XX688768436543000R94R54R02

Action Output

Time	Action	Response	Duration / Fetch Time
10:21:03	use sprint4	0 row(s) affected	0.00033 sec
10:21:03	SELECT AVG(t.amount), cc.iban FROM transactions AS t LEFT J...	371 row(s) returned	0.218 sec / 0.000095..

- **Level 2**

To create and insert the data into the table credit_card_status I used the following queries.

```

CREATE TABLE IF NOT EXISTS credit_cards_status (
    id VARCHAR(20) PRIMARY KEY,
    status VARCHAR(10)
    FOREIGN KEY (id) REFERENCES credit_cards(id)
);

INSERT INTO credit_cards_status (id, status)
    SELECT card_id,
        CASE WHEN SUM(CASE WHEN declined = 1 THEN 1 ELSE 0 END) = 3 THEN 'INACTIVE'ELSE 'ACTIVE' END AS status
    FROM (
        SELECT card_id, declined, ROW_NUMBER() OVER(PARTITION BY card_id ORDER BY timestamp DESC) as lastTransactions
        FROM transactions) as lastTransactionsTT
    WHERE lastTransactions <=3
    GROUP BY card_id;

```

- **Level 2 Exercise 1**

4995 cards are active.

```
1 •  use sprint4;
2
3 •  SELECT COUNT(*)
4   FROM credit_cards_status
5   WHERE status = "ACTIVE";
```

Result Grid

COUNT(*)
4995

Action Output

Time	Action	Response	Duration / Fetch Time
10:35:18	use sprint4	0 row(s) affected	0.00030 sec
10:35:18	SELECT COUNT(*) FROM credit_cards_status WHERE status = "ACTIVE";	1 row(s) returned	0.0027 sec / 0.00000...

- **Level 3 Exercise 1**

```
CREATE TABLE IF NOT EXISTS products(
    id INT PRIMARY KEY,
    product_name VARCHAR(255),
    price DECIMAL(10, 2),
    colour VARCHAR(20),
    weight FLOAT,
    warehouse_id VARCHAR(10)
);

LOAD DATA LOCAL INFILE '/Users/andre/Documents/1data_analist/CURSO_IT_ACADEMY/sprint4/products.csv'
INTO TABLE products
FIELDS TERMINATED BY ','
ENCLOSED BY ""
LINES TERMINATED BY '\n'
IGNORE 1 LINES;

ALTER TABLE transactions
ADD FOREIGN KEY (product_ids) REFERENCES products(id);
```

```
1 •  use sprint4;
2
3 •  SELECT DISTINCT(t.product_ids) AS product_id, p.product_name,COUNT(*) AS unities_sold
4   FROM transactions AS t
5   LEFT JOIN products AS p ON t.product_ids = p.id
6   GROUP BY product_ids;
```

Result Grid

product_id	product_name	unities_sold
1	arya stark	100
3	duel tourney Lannister	1027
4	warden south duel	960
5	skywalker ewok	1030
6	dooku solo	987
7	north of Casterly	1020
8	Winterfell	1005

Action Output

Time	Action	Response	Duration / Fetch Time
10:52:49	use sprint4	0 row(s) affected	0.00032 sec
10:52:49	SELECT DISTINCT(t.product_ids) AS product_id, p.product_name,...	100 row(s) returned	0.051 sec / 0.000028...

URL Github: https://github.com/andre-mazus/DataAnalysisSpecialization/blob/main/sprint_4_exercises.sql