

Instructions

This challenge is designed for you to demonstrate two key requirements for all Mollie data analysts: analytical skills and commercial awareness.

Once we have received your answers to this challenge, you may be invited (after a positive review) to present your work to members of Mollie's Data Team and/or the team to which you are applying (15 mins presentation, 15 mins Q&A). You can present your answers to each part however you choose, but you may find it helpful to create slides.

Challenge Overview:

To demonstrate your capabilities in these areas, please complete the following 2 tasks:

- Part 1: analyse a sample dataset to provide recommendations around Mollie's strategy
- Part 2: write 2 SQL queries to combine table schemas provided

Tips!

Some tips to think about when working on the challenge.

- Try to stay away from generalities and focus on more in-depth analysis of the payment methods and channels
- Make sure to share all your conclusions + recommendations already in the report via email and use the same report when presenting in the interview - so we can assess it properly
- Make sure to tie your recommendations to Mollie as much as possible

Part 1: Commercial Analysis

Based on the two datasets provided, complete an analysis using a tool of your choice (e.g. SQL, Excel, R, Python) and write a report including findings, visualisations and recommendations to Mollie.

Be prepared to discuss your findings in an interview.

Notes:

- The total monthly costs for each acquisition channel is €10k
- Pricing for different payment methods is shown in the table to the right
- Fixed rate is expressed as a € fee per payment e.g. 0.2 = €0.20 per payment
- Variable rate is expressed as a percentage of volume e.g. 0.01 = 1% of vol

payment_method_id	fixed_rate	variable_rate
3	0.29	0
11	0.25	0.018
17	0.39	0
19	0.25	0

Data provided in this exercise have been fabricated for this case study.

The data is imperfect, just like in the real world ;)

Part 2: SQL skills

This is an opportunity to demonstrate your SQL skills. There is no actual data provided - answer the questions by writing SQL queries using the table schemas provided.

You have been provided with the schema for 4 tables, with column definitions and a preview of some rows from each table.

The 4 tables are:

- 1. organizations: properties of organizations i.e. Mollie's customers**
- 2. payments: data on the payments processed by each organization**
- 3. custom_pricing: data on an organization's custom pricing agreement for each payment method**
- 4. default_pricing: pricing per payment method for organizations without a custom pricing agreement. The schema for this table is identical to custom_pricing, but does not include customer_id**

Table 1: Organizations

- 1. **customer_id**: unique identifier for each organization
- 2. **organization_name**: name of the organization
- 3. **country_code**: registered country of the organization
- 4. **sales_manager_id**: identifier for the sales manager of that organization. NULL means acquired organically
- 5. **first_payment_date**: date of first payment of that organization

customer_id	country_code	organization_name	sales_manager_id	first_payment_date
345	GB	Electronics Ltd	358787	2018-01-01
346	NL	Furniture Store BV		2018-02-12
347	BE	Food delivery Inc	235567	2019-03-18
348	NL	Jewelry store		2019-04-25
...

Table 2: Payments

- 1. `payment_id`: unique identifier for each payment
- 2. `customer_id`: unique identifier for each organization
- 3. `payment_date`: date which payment was made on
- 4. `payment_method_id`: unique identifier for the payment method e.g. credit card, iDEAL, Bancontact etc.
- 5. `total_volume`: volume in euros processed by that customer, on each day for each payment method

Data shown below has been fabricated to help illustrate what the real table could look like.

payment_id	customer_id	payment_date	payment_method_id	total_volume
1145	345	2020-02-01	3	64.55
1146	345	2020-02-02	11	24.35
1147	346	2020-02-03	15	6400.00
1148	346	2020-02-01	3	500.00
...

Table 3: Custom Pricing

Note: the schema for default_pricing is identical to custom_pricing, but is not customer-specific and therefore does not include customer_id

- 1. customer_id: unique identifier for each organization
- 2. custom_pricing_id: unique identifier for each instance of custom pricing
- 3. payment_method_id: unique identifier for each payment method e.g. iDEAL = 3
- 4. starts_at: date that instance of custom pricing started on
- 5. ends_at: date that the instance of custom pricing ended on (if null, then still active)
- 6. fixed_rate: fee amount in euros charged per transaction
- 7. variable_rate: fee amount expressed as a fraction of transaction volume

Data shown below has been fabricated to help illustrate what the real table could look like.

customer_id	custom_pricing_id	payment_method_id	starts_at	ends_at	fixed_rate	variable_rate
1000	567	3	2018-01-01	2018-07-11	0.06	0
1000	568	3	2018-07-11		0.057	0
1000	569	11	2019-01-01		0.21	0.028
1001	570	11	2019-01-01		0.22	0.016
...

Question 2A

The sales team require a reporting tool to monitor *updates* to custom pricing. Write a query to generate a table showing updates to instances of custom pricing. The table should look as follows:

customer_id	payment_method_id	pricing_updated_at	new_fixed_rate	new_variable_rate	old_fixed_rate	old_variable_rate
1000	3	2020-01-01	0.2	0.0151	0.2	0.016
1000	3	2019-03-12	0.2	0.016	0.2	0.017
1000	3	2018-06-24	0.2	0.017	0.2	0.018
...

Question 2B

Write a query to infer the total fee for each payment shown in the table schema provided.

Include the following columns in your result set:

1. `payment_id`
2. `total_volume`
3. `total_fixed_fee`
4. `total_variable_fee`
5. `total_fee`

Hints:

- *Use the `custom_pricing` table to determine whether custom pricing was active for each payment*
- *If custom pricing was not active, use the default pricing for that payment method on the payment date*
- *`total_fee` is just the fixed and variable fees per payment combined*