Practical Exam: Loan Insights

EasyLoan offers a wide range of loan services, including personal loans, car loans, and mortgages.

EasyLoan offers loans to clients from Canada, United Kingdom and United States.

The analytics team wants to report performance across different geographic areas. They aim to identify areas of strength and weakness for the business strategy team.

They need your help to ensure the data is accessible and reliable before they start reporting.

Database Schema

The data you need is in the database named `lending`.

Task 1

The analytics team wants to use the `client` table to create a dashboard for client details. For them to proceed, they need to be sure the data is clean enough to use.

The `client` table below illustrates what the analytics team expects the data types and format to be.

Write a query that makes the `client` table match the description provided. Your query should not update the `client` table.

Column Name	Description
client_id	Unique integer (set by the database, can’t take any other value)
date_of_birth	Date of birth of the client, as a date (format: YYYY-MM-DD)
employment_status	Current employment status of the client, either employed or unemployed, as a lower case string
country	The country where the client resides, either USA, UK or CA, as an upper case string

Unknown integration DataFrame as client

```
SELECT
    client_id,
    date_of_birth::date AS date_of_birth,
    -- Handle employment_status: correct 'emplouyed' to 'employed', convert 'employed' and 'unemployed' to lowercase
    CASE
        WHEN LOWER(employment_status) = 'emplouyed' THEN 'employed'
        WHEN LOWER(employment_status) = 'employed' THEN 'employed'
        WHEN LOWER(employment_status) = 'full-time' THEN 'employed'
        WHEN LOWER(employment_status) = 'part-time' THEN 'employed'
        WHEN LOWER(employment_status) = 'unemployed' THEN 'unemployed'
        ELSE NULL -- Handle other unexpected values
    END AS employment_status,
    -- Convert country to upper case and ensure it is either 'USA', 'UK', or 'CA'
    CASE
        WHEN UPPER(country) IN ('USA', 'UK', 'CA') THEN UPPER(country)
        ELSE NULL -- Handle unexpected values
    END AS country
FROM
    client;
```

	client_id	date_of_birth
0	1	1963-07-08T00:00:00.000
1	2	1957-02-07T00:00:00.000
2	3	1993-02-21T00:00:00.000
3	4	1978-03-19T00:00:00.000
4	5	2000-10-02T00:00:00.000
5	6	1974-08-05T00:00:00.000
6	7	1980-07-14T00:00:00.000
7	8	1995-06-24T00:00:00.000
8	9	1962-02-21T00:00:00.000
9	10	1992-05-28T00:00:00.000
10	11	2001-11-30T00:00:00.000
11	12	1981-02-05T00:00:00.000
12	13	1960-07-16T00:00:00.000
13	14	1962-06-15T00:00:00.000
14	15	1990-06-27T00:00:00.000

300 rows

Task 2

You have been told that there was a problem in the backend system as some of the `repayment_channel` values are missing.

The missing values are critical to the analysis so they need to be filled in before proceeding.

Luckily, they have discovered a pattern in the missing values:

- Repayment higher than 4000 dollars should be made via `bank account`.
- Repayment lower than 1000 dollars should be made via `mail`.

Write a query that makes the `repayment` table match this criteria.

Unknown integration   DataFrame as `repayment`

```
SELECT
    repayment_id,
    repayment_amount,
    CASE
        WHEN repayment_channel = '-' THEN
            CASE
                WHEN repayment_amount > 4000 THEN 'bank account'
                WHEN repayment_amount < 1000 THEN 'mail'
                ELSE repayment_channel
            END
        ELSE repayment_channel
    END AS repayment_channel
FROM repayment;
```

	repayment_id	repayment_amount
0	1	
1	2	
2	3	
3	4	
4	5	
5	6	
6	7	
7	8	
8	9	
9	10	
10	11	
11	12	
12	13	
13	14	
14	15	

1,500 rows

Task 3

Starting on January 1st, 2022, all US clients started to use an online system to sign contracts.

The analytics team wants to analyze the loans for US clients who used the new online system.

Write a query that returns the data for the analytics team. Your output should include `client_id`, `contract_date`, `principal_amount` and `loan_type` columns.

Unknown integration DataFrame as u

```sql
SELECT l.client_id, c.contract_date, l.principal_amount, l.loan_type
FROM loan l
JOIN contract c ON l.contract_id = c.contract_id
JOIN client cl ON l.client_id = cl.client_id
WHERE c.contract_date >= '2022-01-01' -- Starting from January 1, 2022
  AND cl.country = 'USA'; -- Filtering only for clients in the USA
```

	client_id	contract_date
0	267	2022-03-08T00:00:00.000
1	50	2022-01-13T00:00:00.000
2	280	2022-01-02T00:00:00.000
3	79	2022-01-24T00:00:00.000
4	245	2022-01-03T00:00:00.000
5	181	2022-02-16T00:00:00.000
6	194	2022-01-03T00:00:00.000
7	251	2022-04-14T00:00:00.000
8	128	2022-03-27T00:00:00.000
9	211	2022-03-18T00:00:00.000
10	86	2022-03-14T00:00:00.000
11	267	2022-03-07T00:00:00.000
12	50	2022-03-24T00:00:00.000
13	42	2022-01-10T00:00:00.000
14	252	2022-03-24T00:00:00.000

94 rows

Task 4

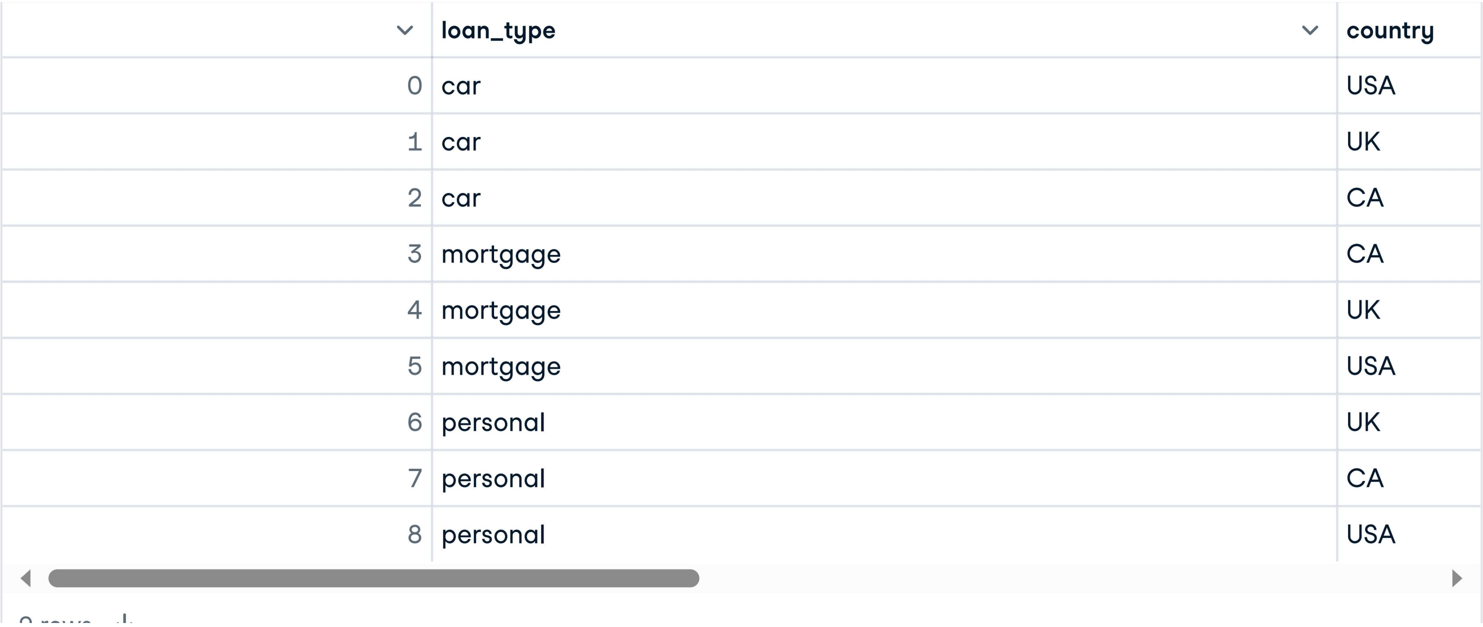
The business strategy team is considering offering a more competitive rate to the US market.

The analytic team want to compare the average interest rates offered by the company for the same loan type in different countries to determine if there are significant differences.

Write a query that returns the data for the analytics team. Your output should include `loan_type`, `country` and `avg_rate` columns.

Unknown integration   DataFrame as a

```sql
SELECT l.loan_type, c.country, AVG(l.interest_rate) AS avg_rate
FROM loan l
LEFT JOIN client c ON l.client_id = c.client_id
GROUP BY l.loan_type, c.country
ORDER BY l.loan_type;
```



	loan_type	country
0	car	USA
1	car	UK
2	car	CA
3	mortgage	CA
4	mortgage	UK
5	mortgage	USA
6	personal	UK
7	personal	CA
8	personal	USA

9 rows