

It's preferred to get the exercises resolved using Python but if you feel more comfortable using another language feel free to do it.

Please send the results in 2 different folders with the name Exercise 1 and Exercise 2 including all the files required to run the code and the results.

Exercise 1

Look at the database diagram to understand how the tables are related and create a script that queries the data and export the results.

- A) Aggregated by partner code and product code get:
- Average loan size (TotalDisbursementAmount)
 - Total number of loans
 - Total amount disbursed (TotalDisbursementAmount)
 - Total number of unique borrowers

Output format: CSV delimited by comma “,” including headers.

- B) Aggregated by partner code and partner borrower Id get:
- Total number of loans
 - Total amount disbursed (TotalDisbursementAmount)
 - Average loan size (TotalDisbursementAmount)

Only show the top 10 borrowers per partner with more disbursed loans.

Output format: CSV delimited by comma “,” including headers.

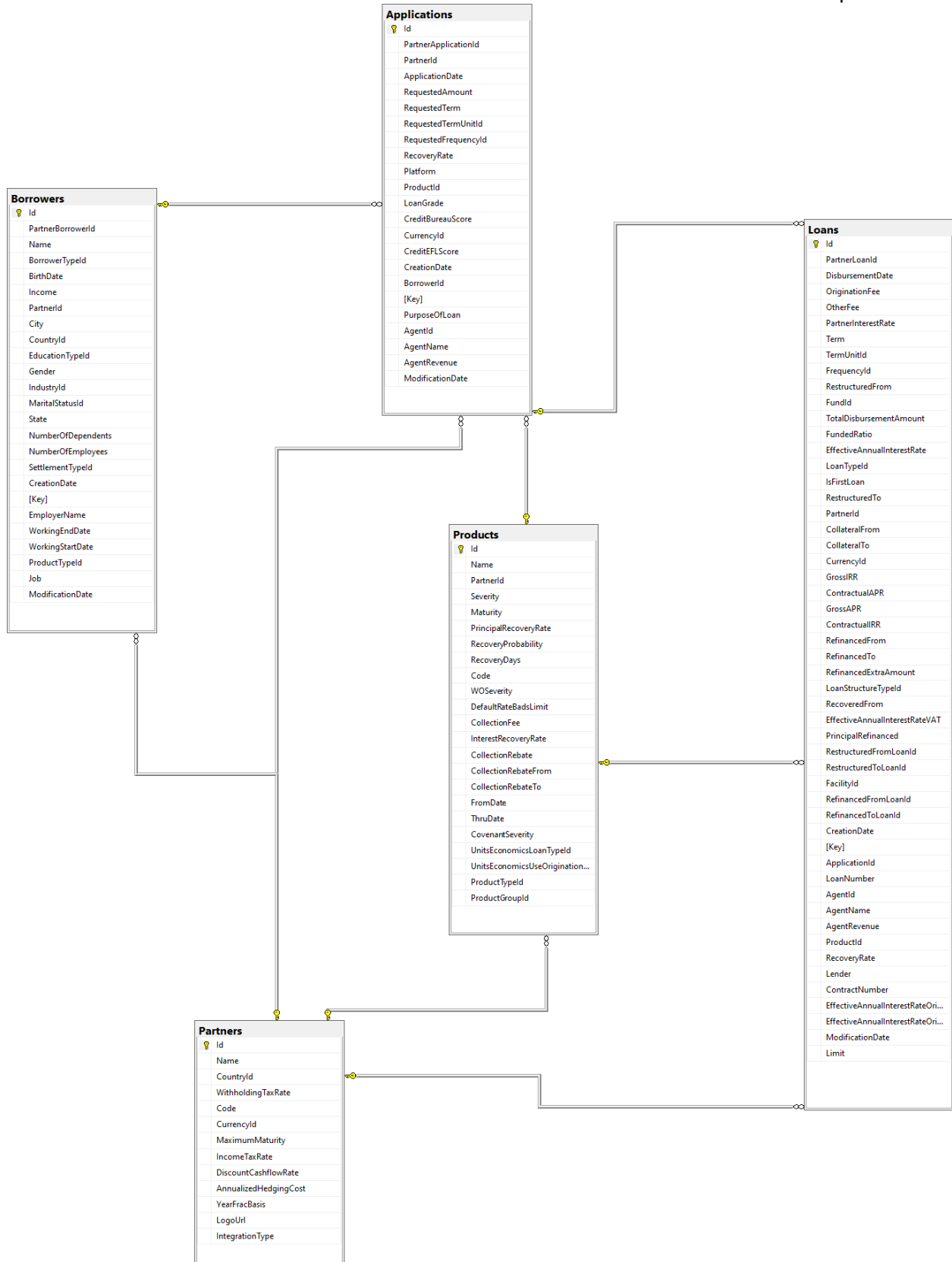
Database: SQL Server

Endpoint: testdb.ckv1htjmgn5s.us-east-1.rds.amazonaws.com

Port: 1433

User: test

Password: SoftDevTest123!



Exercise 2

Create a client that send to our API 1 Borrower, 1 Application, 1 Loan and 1 Payment.

API workflow:

- 1) Create the Borrower.
- 2) Take the Id from the response and use it in the applications URL.
- 3) Create the Application.
- 4) Take the Id from the response and use it in the applicationId field in the Loan body.
- 5) Create the Loan.
- 6) Take the Id from the response and use it in the payments URL.
- 7) Create the Payment

Borrower data:

- externalBorrowerId: Create it by your own.
- name: Exercise 2
- borrowerTypeId: 70BBD5E3-EDB7-4C28-A1C4-0F894EEA4467
- countryId: Choose one from the list of available values you can find in the endpoint <https://dev.orca.accialcapital.com/api/integration/tables/countries>
- educationTypeId: Choose one from the list of available values you can find in the endpoint <https://dev.orca.accialcapital.com/api/integration/tables/educationtypes>

Application data:

- externalApplicationId: Create it by your own.
- applicationDate: Create it by your own.
- requestedTerm: 3
- requestedTermUnitId: 246d03c8-ea99-406e-a32e-e5764e634a63
- productId: 462de2eb-0b30-4685-9686-0ac43295f72e
- currencyId: Choose one from the list of available values you can find in the endpoint <https://dev.orca.accialcapital.com/api/integration/tables/currencies>

Loan data:

- externalLoanId: Create it by your own.
- applicationId: applicationId you got from the Application create response.
- loanTypeId: 3A0B59D1-864A-43A6-B860-56968EBB13E6
- disbursementDate: Create it by your own.
- interestRate: 0.2
- term: 3
- totalDisbursementAmount: 300
- fundId: 77a9139d-caab-43fd-aafb-3effc7652877
- termUnitId: 6575C4FF-32DC-4B62-BB9E-AC2C34DD55D3
- currencyId: Same Id of the Application.

- loanStructureTypeId: E060ECE8-0446-45F3-B0ED-43A1F597755D
- paymentSchedule: Array with 3 elements.
 - number: 1
 - date: 1 month after disbursementDate
 - expectedPrincipal: 100
 - expectedInterest: 20
 - expectedVAT: 10
 - number: 2
 - date: 2 months after disbursementDate
 - expectedPrincipal: 100
 - expectedInterest: 20
 - expectedVAT: 10
 - number: 3
 - date: 3 months after disbursementDate
 - expectedPrincipal: 100
 - expectedInterest: 20
 - expectedVAT: 10

Payment data:

- paymentDate: Any date after the disbursement date.
- concepts: Array with 3 elements.
 - conceptId: 91DC91A3-3D97-489B-AE32-8861D9D67D2C
 - amount: 100
 - conceptId: 1BC5E871-3AE7-465A-9525-1D60DED6B381
 - amount: 10
 - conceptId: 650F41F7-8CBB-4868-8843-B7020962F556
 - amount: 20

Borrowers endpoint: <https://dev.orca.accialcapital.com/api/integration/borrowers>

Applications endpoint:
<https://dev.orca.accialcapital.com/api/integration/borrowers/{borrowerId}/applications>

Loans endpoint: <https://dev.orca.accialcapital.com/api/integration/loans>

Payments endpoint: <https://dev.orca.accialcapital.com/api/integration/loans/{loanId}/payments>

The API authentication is done via a JWT token.

You have to include an Authorization header with the value: "Bearer {API Key}"

API Documentation: <https://dev.orca.accialcapital.com/swagger-ui/index.html>

API Key:

eyJhbGciOiJSUzI1NiIsImtpZCI6Ijlk4QkFBQThDNjZFQTY5NTkiLCJrdHkiOiJSU0EifQ.eyJQYXJ0bmVySWQzOiI0iYMDZERjNERC05Qjc1LTRDNjAtOTFGNy1ERjQ1MzE4NUEzN0YiLCJleHAiOiE2MzQzMDE4OTgsImhhdCI6MTYwMjc2NDg5OCwianRpljoiYjA1NGFhMDYtZGUzZC00MzcwLTlmZTYtZmQ4NTZjZjVhODIhliwicm9sZSI6IiBhcnRuZXIifQ.UVKkfwED6DeAB_8Dx72nLRfZSQrms-2IJXYYDDJBbuJEAPphOsJdC5qM9Kp4RzzLI5PIhet34nu3gyoll-Xylyor0DhW-g0UE1H_g7_ok8hKwU5ZkJ3Y-3vZnJz0GWWKhSMPbts0Mfqxbdlh5gSMacj45vV1MHxtLnC9cULPLBe4ojZ7BDnpJDdia1zGZ6J2a_yG9wNth8K4y8dDHUktLzH8LWdLOo_ROVYIE5ki58pxF7qQF2KIOGbnTWBTB3zBLOgUGg8f2yzt5cMUaxr2Pu_i8aeeypNS7KLoVUTIHLxUYI0cKi6Wkrx9Onc-JdPDF_bo6l60JH9pvqiFNlgPKCg