Umba SQL assessment

# Section 1 - Joins

***-- check users table***

|  |
| --- |
| select \* from users; |

***-- check loans table***

|  |
| --- |
| select \* from loans; |

***-- check transactions table***

|  |
| --- |
| select \* from transactions; |

***-- final query - use joins to retrieve the required result!***

|  |
| --- |
| select loans.id as loan\_id, loans.amount as loan\_amount, users.id as user\_id, users.created as user\_created, transaction\_fee, t\_type from transactions ***-- selected the required columns from three different tables and used 'alias' for column names in required format***  join loans on transactions.loan\_id = loans.id ***-- join transactions and loans table using 'loan\_id' column***  join users on loans.user\_id = users.id ***-- join loans and users table using 'user\_id' column***  where t\_type = 'disbursement' ***-- filter the data based on transaction\_type = 'disbursement'***  order by users.created desc; ***-- display results from new to old which means, the newly created user comes on top!*** |

# Section 2 - Aggregation

***--1 number of loans per month***

***-- used aggregate function 'COUNT' on 'id' column, then extracted(extract() method) month and year from the 'created' date column.***

***-- used 'group by' to group the results by month and the year. Finally sorted the results based on recency (latest created comes on top)!***

|  |
| --- |
| SELECT count(id), extract(month FROM created) as month, extract(year FROM created) as year from loans group by month, year  order by year desc, month desc; |

***--2 total amount of all loans per month***

***-- used aggregate function 'SUM' on 'amount' column, then extracted(extract() method) month and year from the 'created' date column.***

***-- used 'group by' to group the results by month and the year. Finally sorted the results based on recency (latest created comes on top)!***

|  |
| --- |
| select sum(amount), extract(month FROM created) as month, extract(year FROM created) as year from loans group by month, year  order by year desc, month desc; |

***--3 average amount of each loan per month***

***-- used aggregate function 'AVG' on 'amount' column and rounded it to 2 decimals, then extracted(extract() method) month and year from the 'created' date column.***

***-- used 'group by' to group the results by month and the year. Finally sorted the results based on recency (latest created comes on top)!***

|  |
| --- |
| select round(avg(amount),2) as avg\_amount, extract(month FROM created) as month, extract(year FROM created) as year from loans group by month, year  order by year desc, month desc; |

***--4 smallest loan size per month***

***-- used aggregate function 'MIN' on 'amount' column, then extracted(extract() method) month and year from the 'created' date column.***

***-- used 'group by' to group the results by month and the year. Finally sorted the results based on recency (latest created comes on top)!***

|  |
| --- |
| select min(amount) as min\_amount, extract(month FROM created) as month, extract(year FROM created) as year from loans  group by month, year  order by year desc, month desc; |

***--5 largest loan size per month***

***-- used aggregate function 'MAX' on 'amount' column, then extracted(extract() method) month and year from the 'created' date column.***

***-- used 'group by' to group the results by month and the year. Finally sorted the results based on recency (latest created comes on top)!***

|  |
| --- |
| select max(amount) as max\_amount, extract(month FROM created) as month,extract(year FROM created) as year from loans  group by month, year  order by year desc, month desc; |