USE financial9\_99; *-- from now on, the default database schema we use is: financial*

*-- checking type of relationship*

SELECT

account\_id,

*count*(trans\_id) as amount

FROM trans

GROUP BY account\_id

ORDER BY 2 DESC

*-- Write a query that prepares a summary of the granted loans in the following dimensions:*

*-- year, quarter, month,*

*-- year, quarter,*

*-- year,*

*-- total.*

*-- Display the following information as the result of the summary:*

*-- total amount of loans,*

*-- average loan amount,*

*-- total number of given loans.*

SELECT *\**

FROM financial9\_99.loan

SELECT

*extract*(YEAR FROM date) as year,

*extract*(QUARTER FROM date) as quarter,

*extract*(MONTH FROM date) as month,

*sum*(amount) as loan\_total\_amount,

*avg*(amount) as loan\_avg\_amount,

*count*(amount) as loan\_count\_amount

FROM financial9\_99.loan

group by year, quarter, month with rollup

order by year, quarter, month;

*-- Write a query that ranks accounts according to the following criteria:*

*-- number of given loans (decreasing),*

*-- amount of given loans (decreasing),*

*-- average loan amount,*

*-- Only fully paid loans are considered.*

WITH account\_ranking AS (

SELECT

account\_id,

*sum*(amount) as loan\_amount,

*count*(amount) as loan\_count,

*avg*(amount) as loan\_avg

FROM financial9\_99.loan

WHERE status IN ('A', 'C')

GROUP BY account\_id

)

SELECT

*\**,

*DENSE\_RANK*() OVER (ORDER BY loan\_amount DESC) AS rank\_loans\_amount,

*DENSE\_RANK*() OVER (ORDER BY loan\_count DESC) AS rank\_loans\_count

FROM account\_ranking;

*-- Fully paid loans*

*-- Find out the balance of repaid loans, divided by client gender.*

*-- Additionally, use a method of your choice to check whether the query is correct.*

select *count*(distinct loan\_id),

*sum*(amount),

gender

from loan

join financial9\_99.account a on loan.account\_id = a.account\_id

join financial9\_99.disp d on a.account\_id = d.account\_id

join financial9\_99.client c on c.client\_id = d.client\_id

where status in ('A', 'C')

AND d.type = 'OWNER'

group by gender

order by *sum*(amount);

*-- Client analysis - part 1*

*-- Modifying the queries from the exercise on repaid loans, answer the following questions:*

*-- Who has more repaid loans - women or men? - Women*

*-- What is the average age of the borrower divided by gender?*

*-- Hints:*

*-- Save the result of the previously written and then modified query, for example, to a temporary table, and conduct the analysis on it.*

WITH gender\_rate AS (

SELECT

*COUNT*(DISTINCT loan.loan\_id) AS loan\_count,

*SUM*(loan.amount) AS total\_amount,

c.gender

FROM loan

JOIN financial9\_99.account a ON loan.account\_id = a.account\_id

JOIN financial9\_99.disp d ON a.account\_id = d.account\_id

JOIN financial9\_99.client c ON c.client\_id = d.client\_id

WHERE loan.status IN ('A', 'C')

AND d.type = 'OWNER'

GROUP BY c.gender

),

avg\_age AS (

SELECT

*AVG*(*FLOOR*(*DATEDIFF*(loan.date, c.birth\_date) / 365.25)) AS average\_age,

c.gender

FROM loan

JOIN financial9\_99.account a ON loan.account\_id = a.account\_id

JOIN financial9\_99.disp d ON a.account\_id = d.account\_id

JOIN financial9\_99.client c ON c.client\_id = d.client\_id

WHERE loan.status IN ('A', 'C')

AND d.type = 'OWNER'

GROUP BY c.gender

)

SELECT

gender\_rate.gender,

gender\_rate.loan\_count,

gender\_rate.total\_amount,

avg\_age.average\_age

FROM gender\_rate

JOIN avg\_age ON avg\_age.**gender** = gender\_rate.**gender**;

*-- Client analysis - part 2*

*-- Make analyses that answer the questions:*

*-- which area has the most clients,*

*-- in which area the highest number of loans was paid, - 1*

*-- in which area the highest amount of loans was paid. - 1*

*-- Select only owners of accounts as clients.*

select *count*(distinct loan\_id),

*sum*(amount),a.district\_id

from loan

join financial9\_99.account a on loan.account\_id = a.account\_id

join financial9\_99.disp d on a.account\_id = d.account\_id

join financial9\_99.client c on c.client\_id = d.client\_id

where status in ('A', 'C')

AND d.type = 'OWNER'

group by a.district\_id

order by *sum*(amount) desc;

select *count*(distinct loan\_id),

*sum*(amount),a.district\_id

from loan

join financial9\_99.account a on loan.account\_id = a.account\_id

join financial9\_99.disp d on a.account\_id = d.account\_id

join financial9\_99.client c on c.client\_id = d.client\_id

where status in ('A', 'C')

AND d.type = 'OWNER'

group by a.district\_id

order by *count*(distinct loan\_id) desc;

*-- Client analysis - part 3*

*-- Use the query created in the previous task and modify it to determine the percentage of each district in the total amount of loans granted.*

SELECT

*COUNT*(DISTINCT loan.loan\_id),

*SUM*(loan.amount),

a.district\_id,

(*SUM*(loan.amount) / (SELECT *SUM*(amount) FROM loan WHERE status IN ('A', 'C'))) \* 100 AS district\_percentage

FROM loan

JOIN financial9\_99.account a ON loan.account\_id = a.account\_id

JOIN financial9\_99.disp d ON a.account\_id = d.account\_id

JOIN financial9\_99.client c ON c.client\_id = d.client\_id

WHERE loan.status IN ('A', 'C')

AND d.type = 'OWNER'

GROUP BY a.district\_id

ORDER BY *COUNT*(DISTINCT loan.loan\_id) DESC;

*-- Selection - part 1*

*-- Client selection*

*-- Check the database for the clients who meet the following results:*

*-- their account balance is above 1000,*

*-- they have more than 5 loans,*

*-- they were born after 1990.*

*-- And we assume that the account balance is loan amount - payments.*

select *count*(loan.loan\_id),*sum*(amount - payments) as balance, c.client\_id

from loan

join account a on a.account\_id = loan.account\_id

join financial9\_99.disp d on a.account\_id = d.account\_id

join client c on d.client\_id = c.client\_id

where birth\_date > '1990-12-31'

AND loan.status IN ('A', 'C')

AND d.type = 'OWNER'

group by c.client\_id

having balance > 1000;

*-- order by count(loan.loan\_id) desc;*

*-- count(l.loan\_id) > 5*

*-- NEVIM PROC SE MI TADY NIC NEROBRAZUJE*

SELECT

c.client\_id,

*sum*(amount - payments) as client\_balance,

*count*(loan\_id) as loans\_amount

FROM loan as l

INNER JOIN

account a using (account\_id)

INNER JOIN

disp as d using (account\_id)

INNER JOIN

client as c using (client\_id)

WHERE True

AND l.status IN ('A', 'C')

AND d.type = 'OWNER'

*-- AND EXTRACT(YEAR FROM c.birth\_date) > 1990*

GROUP BY c.client\_id

HAVING

*sum*(amount - payments) > 1000

*-- and count(loan\_id) > 5*

ORDER BY loans\_amount DESC; *-- here we add descending sorting by number of loans*

*-- Expiring cards*

*-- Write a procedure to refresh the table you created (you can call it e.g. cards\_at\_expiration) containing the following columns:*

*-- client id,*

*-- card id,*

*-- expiration date - assume that the card can be active for 3 years after issue date,*

*-- client address (column A3 is enough)*

DELIMITER $$

DROP PROCEDURE IF EXISTS financial9\_99.*generate\_cards\_at\_expiration\_report\_EV*;

CREATE PROCEDURE financial9\_99.*generate\_cards\_at\_expiration\_report\_EV*(p\_date DATE)

BEGIN

TRUNCATE TABLE financial9\_99.cards\_at\_expiration\_EV;

INSERT INTO financial9\_99.cards\_at\_expiration\_EV

with cte as (

select c.client\_id,

d.client\_id,

*DATE\_ADD*(card.issued, INTERVAL 3 year) as expiration\_date,

d2.A3 as client\_adress

from card

join financial9\_99.disp d on card.disp\_id = d.disp\_id

join financial9\_99.client c on d.client\_id = card.client\_id

join financial9\_99.district d2 on c.district\_id = d2.district\_id

)

select *\** from cte

WHERE p\_date BETWEEN *DATE\_ADD*(expiration\_date, INTERVAL -7 DAY) AND expiration\_date;

END;

DELIMITER ;

CALL *generate\_cards\_at\_expiration\_report\_EV*('2001-01-01');

SELECT *\** FROM financial9\_99.cards\_at\_expiration\_EV;