**RELATIONAL ALGEBRA TEMPLATE**

**BY: Saksham Jain**

**Ques 1: Find all account ids from Benesov district in Central Bohemia.**

Ans 1:

SELECT a.id

FROM Accounts as a

JOIN Districts as d ON d.id = a.district\_id

WHERE d.name = 'Benesov' AND d.region = 'central Bohemia';

**Ques 2: Find all client-account pairs with the relationship between account and client, the district they are in and the date of account creation for accounts created in 1995.**

Ans 2:

SELECT c.id AS client\_id, a.id AS account\_id, di.name AS district\_name, a.date

AS account\_date

FROM Clients as c

JOIN Dispositions as d ON c.id = d.client\_id

JOIN Accounts as a ON d.account\_id = a.id

JOIN Districts as di ON c.district\_id = di.id

WHERE EXTRACT(YEAR FROM a.date) = 1995 ORDER BY c.id, a.id

**Ques 3: Find the accounts which can be accessed by 2 or more clients (client or disponent).**

Ans 3:

SELECT d.account\_id AS account\_id, COUNT(d.client\_id) as count\_client

FROM Dispositions AS d

GROUP BY d.account\_id

HAVING count\_client >= 2;

**Ques 4: Find the account ids of accounts with a cash credit (Transaction operation ‘VKLAD’) of more than 10000 or a loan of more than 500000.**

Ans 4:

SELECT account\_id

FROM Transactions

WHERE operation = 'VKLAD' AND amount > 10000

UNION

SELECT account\_id

FROM Loans

WHERE amount > 500000;

**Ques 5: Find all account ids who have made loan payments (orders, k\_symbol=’UVER’) minus those who have paid their contract on time (Loans status A).**

Ans 5:

SELECT DISTINCT o.account\_id

FROM Orders o

LEFT JOIN Loans l ON o.account\_id = l.account\_id AND l.status = 'A' WHERE o.k\_symbol = 'UVER' AND l.account\_id IS NULL;

**Ques 6: Find the account ids of accounts with a cash withdrawal (operation ‘VYBER’) of more than 1000 and a loan of more than 50000.**

Ans 6:

SELECT distinct a.id AS account\_id

FROM Accounts AS a

JOIN Transactions AS t ON a.id = t.account\_id

JOIN Loans AS l ON a.id = l.account\_id

WHERE t.operation = 'VYBER' AND t.amount > 1000 AND l.amount > 50000;

**Ques 7: Find the client id and district id of clients with a loan duration of more than 36 months and a loan amount greater than 250000.**

Ans 7:

SELECT distinct c.id, c.district\_id

FROM Clients as c

JOIN Dispositions as d ON c.id = d.client\_id

JOIN Accounts as a ON d.account\_id = a.id

Join loans as l on l.account\_id = a.id

WHERE l.amount > 250000 AND l.duration > 36;

**Ques 8: Find all account id, total withdrawal transaction sum for accounts with total withdrawals more than 75000.**

Ans 8:

SELECT account\_id, SUM(amount) AS total\_withdrawals

FROM Transactions

WHERE type = 'VYDAJ'

GROUP BY account\_id

HAVING total\_withdrawals > 75000;

**Ques 9: . Find the total amount of loans taken out by accounts in 1995 and 1996 grouped by district.**

Ans 9:

SELECT SUM(l.amount) AS total\_loan\_amount, d.name

FROM Loans as l

JOIN Accounts as a ON l.account\_id = a.id

JOIN Districts as d ON a.district\_id = d.id

WHERE YEAR(l.date) = 1995 OR YEAR(l.date) = 1996 GROUP BY d.name

**Ques 10: Find the client and card ids of all gold cards.**

Ans 10:

SELECT c.id AS client\_id, cc.id AS card\_id

FROM Clients as c

JOIN Dispositions as d ON c.id = d.**client\_id**

JOIN Accounts as a ON d.account\_id = a.id

JOIN Cards as cc ON d.id = cc.disp\_id WHERE cc.type = 'gold';

**Ques 11: Find account id, transaction type and average transaction amount for each account and each transaction type (credit, debit).**

Ans 11:

SELECT Transactions.account\_id, Transactions.type,

AVG(Transactions.amount) AS avg\_amount

FROM Transactions

GROUP BY Transactions.account\_id, Transactions.type;

**Ques 12: Find the district id, name, and number of accounts for all districts with more than 20 accounts.**

Ans 12:

SELECT d.name, d.id, COUNT(a.id) AS num\_accounts

FROM Districts as d

INNER JOIN Accounts as a ON d.id = a.district\_id

GROUP BY d.id, d.name

HAVING num\_accounts >= 20

**Ques 13: Find the loan ids of all loans with a monthly payment greater than 7500.**

Ans 13:

SELECT id

FROM Loans

WHERE amount / duration > 7500;

**Ques 14: Find the name, id, total loan and population for districts with a total loan (combined loan of all accounts from district) greater than twice the district population.**

Ans 14:

SELECT d.name, d.id, (SELECT SUM(amount)

FROM Loans l

JOIN Accounts a ON l.account\_id = a.id

WHERE a.district\_id = d.id) AS total\_loan, d.population

FROM Districts d

HAVING total\_loan > d.population \* 2;

**Ques 15: Find the number of transactions for each account-operation pair in 1996.**

Ans 15:

SELECT distinct account\_id, operation, COUNT(\*) AS num\_transactions

FROM Transactions

WHERE year(date) = '1996'and operation !='NULL' GROUP BY account\_id,.operation;

**Ques 16: Find the account id, loan id and total amount paid in loan payments (k\_symbol=’UVER’) by accounts with loans of more than 50000.**

Ans 16:

SELECT account\_id, id AS loan\_id, (payments \* duration) AS total\_amount\_paid

FROM Loans

WHERE amount > 50000

**Ques 17: Find the number and total amount of lease payments (k\_symbol=’LEASING’) in districts with 8 or more cities.**

Ans 17:

SELECT COUNT(\*) AS num\_payments, SUM(o.amount) AS total\_amount FROM Orders o

JOIN Accounts a ON o.account\_id = a.id

JOIN Districts d ON a.district\_id = d.id

WHERE o.k\_symbol = 'LEASING' AND d.num\_cities >= 8 Group by d.id ;

**Ques 18: Find the account id, district name, district id, average district salary and average credit transaction (type=’PRIJEM’) amount for all accounts whose average credit transaction is greater than the average salary for their district.**

Ans 18:

SELECT a.id AS account\_id, d.name AS district\_name, d.id AS district\_id,

AVG(d.avg\_salary) AS avg\_district\_salary, AVG(t.amount) AS avg\_credit\_transaction

FROM Accounts a

JOIN Districts d ON a.district\_id = d.id

JOIN Transactions t ON a.id = t.account\_id

WHERE t.type = 'PRIJEM'

GROUP BY a.id, d.name, d.id

HAVING AVG(t.amount) > AVG(d.avg\_salary);

**Ques 19: Find all loan payment transactions (k\_symbol=’UVER’) along with the id of the loan, id of the account, and the date of the transaction.**

Ans 19:

SELECT Loans.id AS loan\_id, Accounts.id AS account\_id, Transactions.date

FROM Loans

JOIN Accounts ON Loans.account\_id = Accounts.id

JOIN Transactions ON Accounts.id = Transactions.account\_id WHERE Transactions.k\_symbol = 'UVER';

**Ques 20: Find number of credit cards of each type for all districts with population greater than 250000.**

Ans 20:

SELECT d.name AS district\_name, c.type, COUNT(c.id) AS num\_cards

FROM Districts d

JOIN Clients cl ON d.id = cl.district\_id

JOIN Dispositions ds ON cl.id = ds.client\_id Cards c ON ds.id = c.disp\_id

WHERE d.population > 250000 GROUP BY d.name, c.type;

**Ques 21: Find the client ids of clients who do not have accounts with a cash withdrawal (operation ‘VYBER’) of more than 3500.**

Ans 21:

SELECT c.id

FROM Clients c

WHERE NOT EXISTS ( SELECT 1

FROM Dispositions d

JOIN Transactions t ON d.account\_id = t.account\_id

WHERE d.client\_id = c.id

AND t.operation = 'VYBER'

AND t.amount > 3500 );

**Ques 22: Find the id and name of all districts with accounts which made loan payment transactions (k\_symbol=’UVER’) of more than 8000 minus the accounts who have paid off their loans on time (Loans Status A).**

Ans 22:

SELECT DISTINCT d.id, d.name

FROM districts d

JOIN clients c ON c.district\_id = d.id

JOIN dispositions dp ON dp.client\_id = c.id

JOIN accounts a ON a.id = dp.account\_id

JOIN orders t ON t.account\_id = a.id

WHERE t.k\_symbol = 'UVER' and t.amount > 8000 and a.id NOT IN

(SELECT a.id

From accounts a

JOIN loans l ON l.account\_id = a.id WHERE l.status = 'A')

**Ques 23: Find id, and type of cards held by accounts with a cash credit (Transaction operation ‘VKLAD’) of more than 49500 or a loan of more than 500000.**

Ans 23:

SELECT DISTINCT c.id, cd.type

FROM Clients c

JOIN Dispositions d ON c.id = d.client\_id

Cards cd ON d.id = cd.disp\_id

JOIN Accounts a ON a.id = d.account\_id

JOIN Transactions t ON t.account\_id = a.id

WHERE (t.operation = 'VKLAD' AND t.amount > 49500)

OR (a.id IN (SELECT l.account\_id FROM Loans l WHERE l.amount > 500000))

**Ques 24: Find the client ids of all clients with an account which has a loan of more than 100000 and a monthly payment greater than 7500.**

Ans 24:

SELECT DISTINCT c.id

FROM Clients c

JOIN Dispositions d ON c.id = d.client\_id

JOIN Loans l ON l.account\_id = d.account\_id

WHERE l.amount > 100000 AND l.payments > 7500;

**Ques 25: Find the account id and avg statement payment (Transaction k\_symbol=’SLUZBY’) for accounts whose average statement payment is greater than the average statement payments of each district (avg statement payment of district is average statement payment across all accounts in district).**

Ans 25:

WITH district\_avg AS (

SELECT d.id, AVG(t.amount) AS avg\_statement\_payment

FROM Districts d

JOIN Accounts a ON a.district\_id = d.id

JOIN Transactions t ON t.account\_id = a.id

WHERE t.k\_symbol = 'SLUZBY'

GROUP BY d.id

), account\_avg AS (

SELECT a.id, AVG(t.amount) AS avg\_statement\_payment

FROM Accounts a

JOIN Transactions t ON t.account\_id = a.id

WHERE t.k\_symbol = 'SLUZBY'

GROUP BY a.id

)

SELECT a.id, account\_avg.avg\_statement\_payment

FROM Accounts a account\_avg ON a.id = account\_avg.id

WHERE account\_avg.avg\_statement\_payment > (

SELECT AVG(district\_avg.avg\_statement\_payment)

FROM district\_avg

WHERE district\_avg.id = a.district\_id

);

**Ques 26. Find the client ids of all clients who have not paid more than 5000 total (sum of all payments) in insurance payments (Orders k\_symbol = ‘POJISTNE’).**

Ans 26:

SELECT c.id AS client\_id

FROM Clients c

WHERE c.id NOT IN (

SELECT d.client\_id

FROM Accounts a

Join dispositions d on d.account\_id = a.id

JOIN Orders o ON a.id = o.account\_id

WHERE o.k\_symbol = 'POJISTNE'

GROUP BY d.client\_id

HAVING SUM(o.amount) > 5000 );