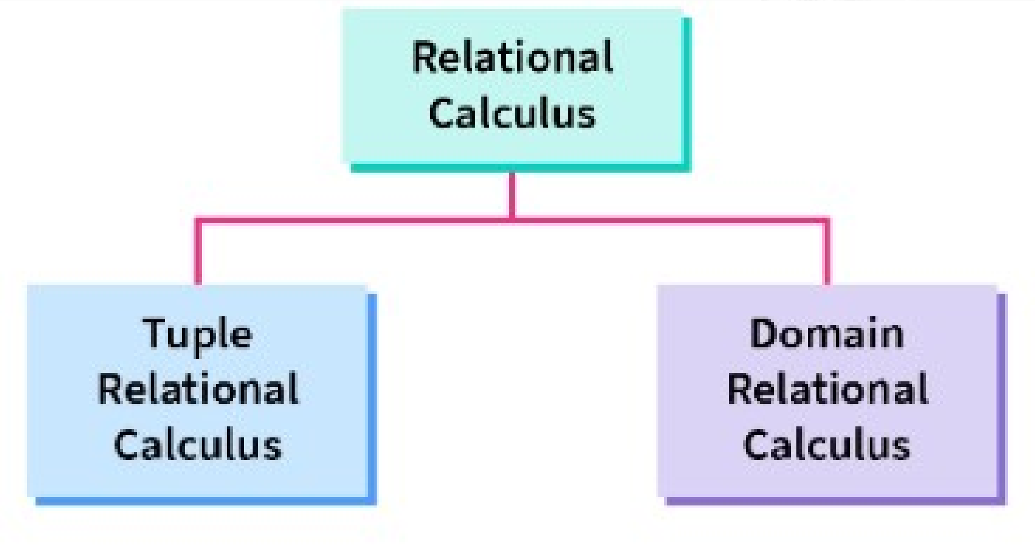
Relational Calculus

- Relational Calculus in database management system (DBMS) is all about ***"What you want ?"***.
- Relational calculus does not tell us how to get the results from the Database, but it just cares about what we want.
- The theory of Relational calculus was introduced by computer scientist and mathematician Edgar Codd.



Relational Calculus

Tuple Relational Calculus

Domain Relational Calculus

Tuple Relational Calculus

- Tuple Relational Calculus is a **non-procedural query language** unlike relational algebra.
- Tuple Calculus provides only the description of the query but it does not provide the methods to solve it.
- Thus, it explains what to do but not how to do.
- In Tuple Calculus, a query is expressed as :

  {t| P(t)}

  where t = resulting tuples,
  P(t) = known as Predicate and these are the conditions that are used to fetch t

- Thus, it generates set of all tuples t, such that Predicate P(t) is true for t.

- P(t) may have various conditions logically combined with OR (∨), AND (∧), NOT(¬).
  It also uses quantifiers:
  $\exists\, t \in r\, (Q(t))$ = ”there exists” a tuple in t in relation r such that predicate Q(t) is true.
  $\forall\, t \in r\, (Q(t))$ = Q(t) is true “for all” tuples in relation r.

Example

- Let’s say we have a table called “Employees” with the following attributes:

- EmployeeID
- Name
- Salary
- DepartmentID

To retrieve the names of all employees who earn more than $50,000 per year, we can use the following TRC query:

$$\{ t \mid \text{Employees}(t) \ \wedge \ t.\text{Salary} > 50000 \}$$

In this query, the “Employees(t)” expression specifies that the tuple variable t represents a row in the “Employees” table. The “ ∧ ” symbol is the logical AND operator, which is used to combine the condition “t.Salary > 50000” with the table selection.

Example contd..

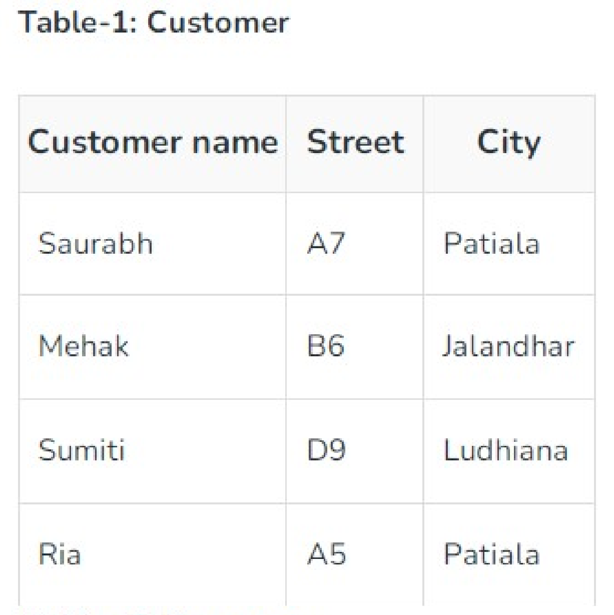


Table-1: Customer

Customer name	Street	City
Saurabh	A7	Patiala
Mehak	B6	Jalandhar
Sumiti	D9	Ludhiana
Ria	A5	Patiala

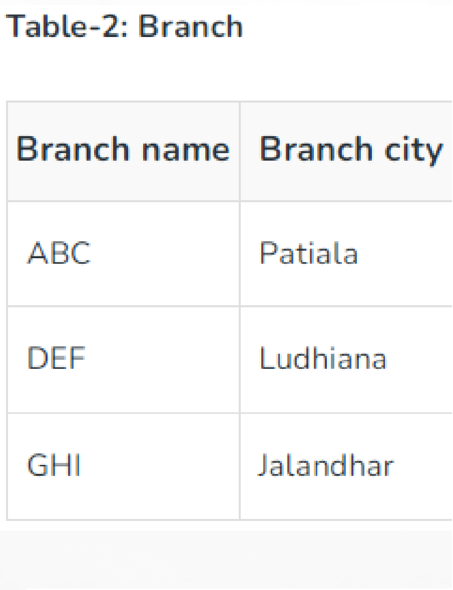


Table-2: Branch

Branch name	Branch city
ABC	Patiala
DEF	Ludhiana
GHI	Jalandhar

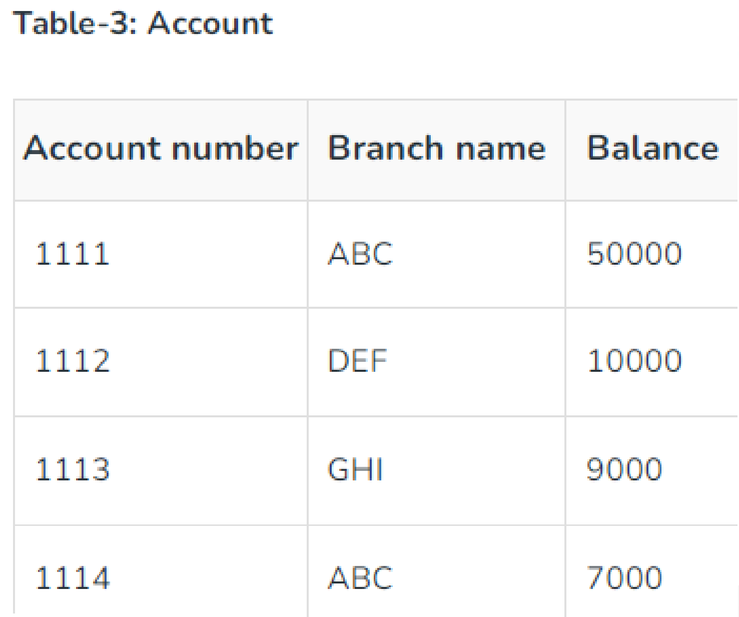


Table-3: Account

Account number	Branch name	Balance
1111	ABC	50000
1112	DEF	10000
1113	GHI	9000
1114	ABC	7000

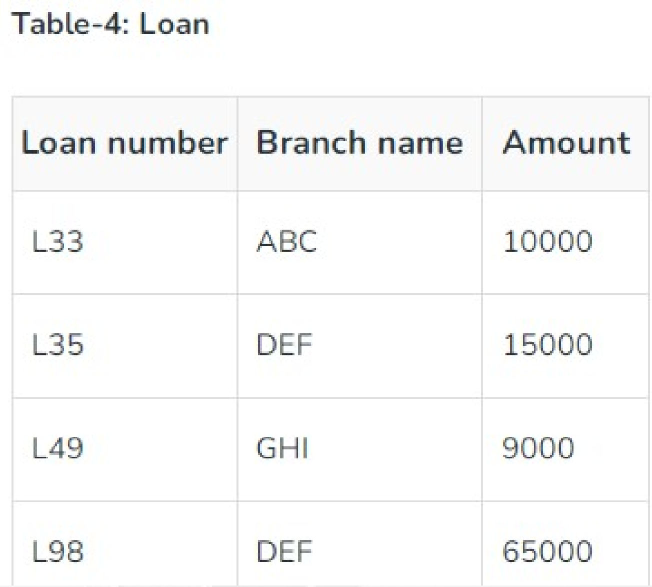


Table-4: Loan

Loan number	Branch name	Amount
L33	ABC	10000
L35	DEF	15000
L49	GHI	9000
L98	DEF	65000

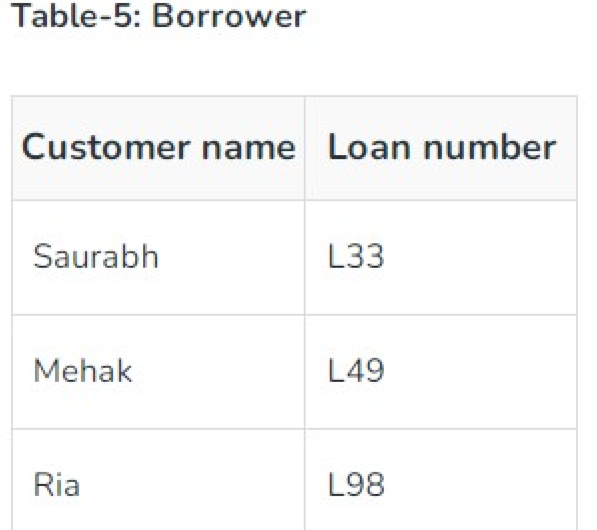


Table-5: Borrower

Customer name	Loan number
Saurabh	L33
Mehak	L49
Ria	L98

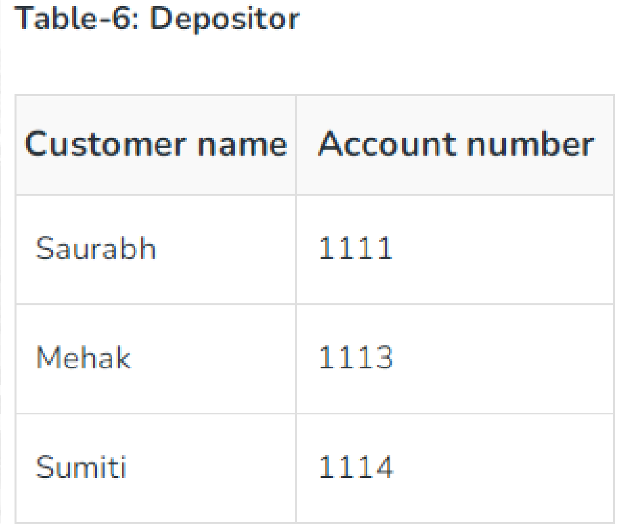


Table-6: Depositor

Customer name	Account number
Saurabh	1111
Mehak	1113
Sumiti	1114

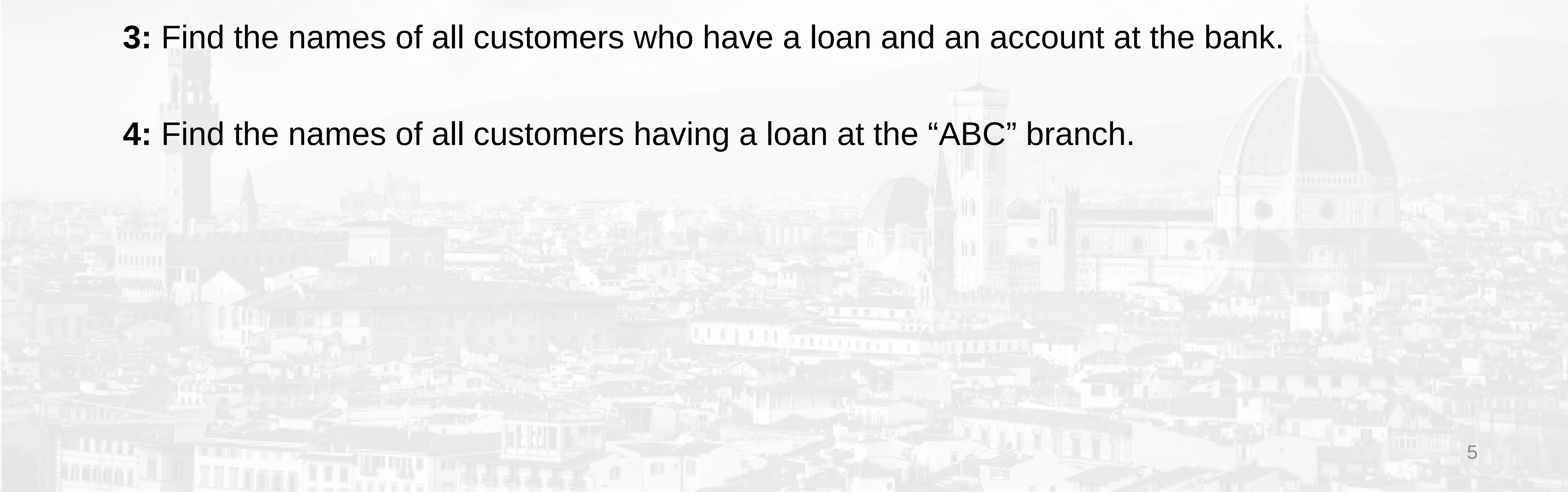
Exercise

1: Find the loan number, branch, amount of loans of greater than or equal to 10000 amount.

2: Find the loan number for each loan of an amount greater or equal to 10000.

3: Find the names of all customers who have a loan and an account at the bank.

4: Find the names of all customers having a loan at the “ABC” branch.



Solution

- **1:** Find the loan number, branch, amount of loans of greater than or equal to 10000 amount.

$$\{t|\ t \in \text{loan} \ \wedge\ t[\text{amount}]>=10000\}$$

- **2:** Find the loan number for each loan of an amount greater or equal to 10000.

$$\{t|\ \exists\ s \in \text{loan}(t[\text{loan number}] = s[\text{loan number}] \ \wedge\ s[\text{amount}]>=10000)\}$$

- **3:** Find the names of all customers who have a loan and an account at the bank.

$$\begin{aligned}\{t \mid \exists\ s \in\ & \text{borrower}(\ t[\text{customer-name}] = s[\text{customer-name}]) \\ & \wedge\ \exists\ u \in \text{depositor}(\ t[\text{customer-name}] = u[\text{customer-name}])\}\end{aligned}$$

- **4:** Find the names of all customers having a loan at the “ABC” branch.

$$\begin{aligned}&\{t \mid \exists\ s \in \text{borrower}(t[\text{customer-name}] = s[\text{customer-name}] \\ &\ \wedge\ \exists\ u \in\ \text{loan}(u[\text{branch-name}] = \text{“ABC”}\ \wedge\ u[\text{loan-number}] = s[\text{loan-number}]))\}\end{aligned}$$

Domain Relational Calculus

- **Domain Relational Calculus** is a non-procedural query language equivalent in power to Tuple Relational Calculus.
- Domain Relational Calculus provides only the description of the query but it does not provide the methods to solve it.
- Domain Relational Calculus uses domain Variables to get the column values required from the database based on the predicate expression or condition.

- In Domain Relational Calculus, a query is expressed as,

{ < x1, x2, x3, ..., xn > | P (x1, x2, x3, ..., xn) }

where, < x1, x2, x3, …, xn > represents resulting domains variables and P (x1, x2, x3, …, xn) represents the condition or formula equivalent to the Predicate calculus.

- Note: The domain variables those will be in resulting relation must appear before | within ≺ and ≻ and all the domain variables must appear in which order they are in original relation or table.

Example

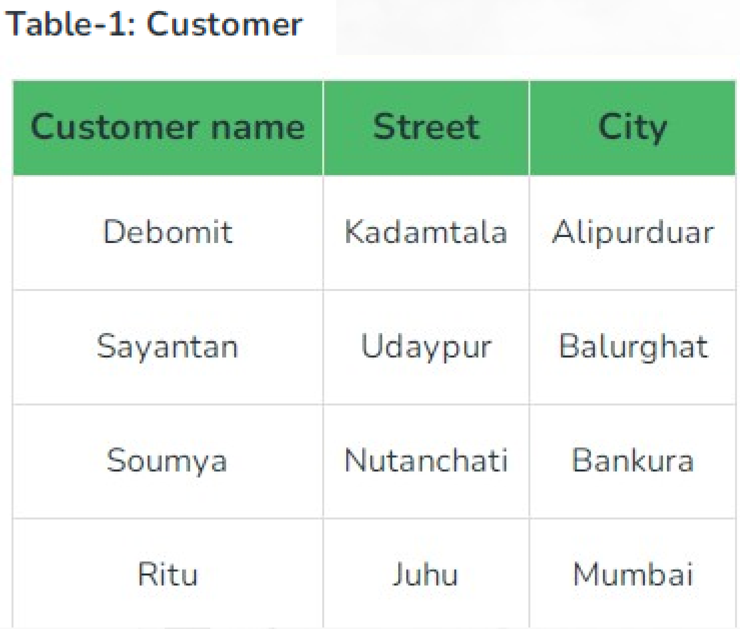
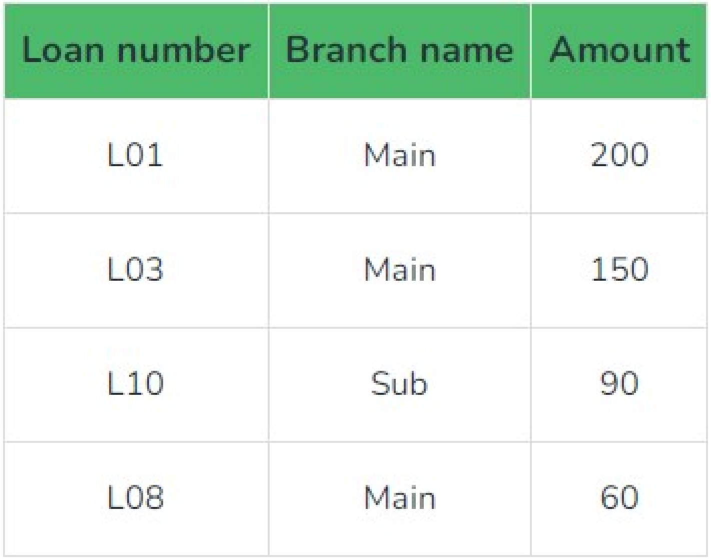


Table-1: Customer

Customer name	Street	City
Debomit	Kadamtala	Alipurduar
Sayantan	Udaypur	Balurghat
Soumya	Nutanchati	Bankura
Ritu	Juhu	Mumbai

Table-2: Loan



Loan number	Branch name	Amount
L01	Main	200
L03	Main	150
L10	Sub	90
L08	Main	60

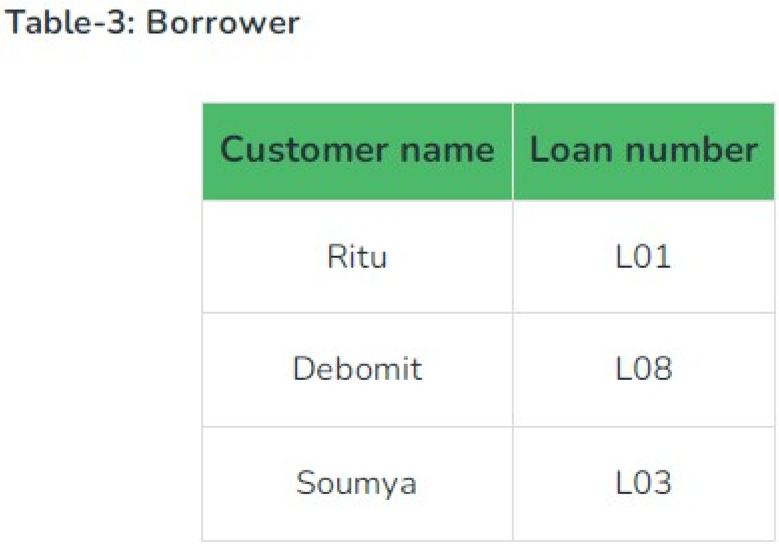


Table-3: Borrower

Customer name	Loan number
Ritu	L01
Debomit	L08
Soumya	L03

Queries

1: Find the loan number, branch, amount of loans of greater than or equal to 100 amount.

{l∈b, a |∧□l, b, a□ □ loan □ (a ≥ 100)}

2: Find the loan number for each loan of an amount greater or equal to 150.

{l∃|∈ ∧b, a (□l, b, a□ □ loan □ (a ≥ 150)}

3: Find the names of all customers having a loan at the "Main" branch and find the loan amount.

$$\{<c, a> \mid \exists\, l\, (<c, l> \in borrower \wedge \exists\, b\, (<l, b, a> \in loan \wedge (b = \text{“Main”})))\}$$