Query languages: Languages in which user request some info. from database.

Procedural

Here user instruct the system to perform a sequence of operations in order to produce desired results. Users tells what data to be retrieved & how to be retrieved

Nonprocedural

Here user describe the desired info. without giving the specific procedure for obtaining that info.

- Procedural → Relational Algebra
- Nonprocedural → Relational Calculus

↓ SQL theoretical

Relational model is a conceptual / framework & RDBMS its implementation

Relational Algebra & Relational calculus are mathematical System or Algorithms based on relational Model

Relational model	RDMS
RA, RC	SQL
Algo	Code
Conceptual theoretical	Implementation

operat...

② a query langage

Relational Algebra is one of the two formal associated with relational model. Like any other mathematical system, it defines a no. of operators and use of relation (tables) as operands.

→ Every operator in relation algebra take one or two relations as input argument & generate a single relation as a result without a name. It does not consider duplicacy as it is based on set theory.

→ In each query we describe a step by step procedure for computing the desired result. So it is called procedural QL.

→ no use of English words.

Fundamental operations

Select (σ)
Project (π)
Union ($\cup$)
Set difference ($-$)
Cartesian product ($\times$)
Rename (ρ)

① select operation (σ): select operation is a unary operation, so can take only one table at a time. Denoted by σ (sigma). Main idea of select is to find those tuples/rows in a relation which satisfies a given condition or predicate.

Eg) Syntax $\sigma_{\text{condition/predication}}(\text{relation/tablename})$

→ It has same function as of where clause in SQL.
→ min. no. of tuples selected is 0
→ max. no. of tuples selected is all tuples.

Student

Sid	Name	Branch
1	A	CS
2	B	ME
3	C	CS
4	D	CS
5	E	EE
6	F	EC
7	G	EE

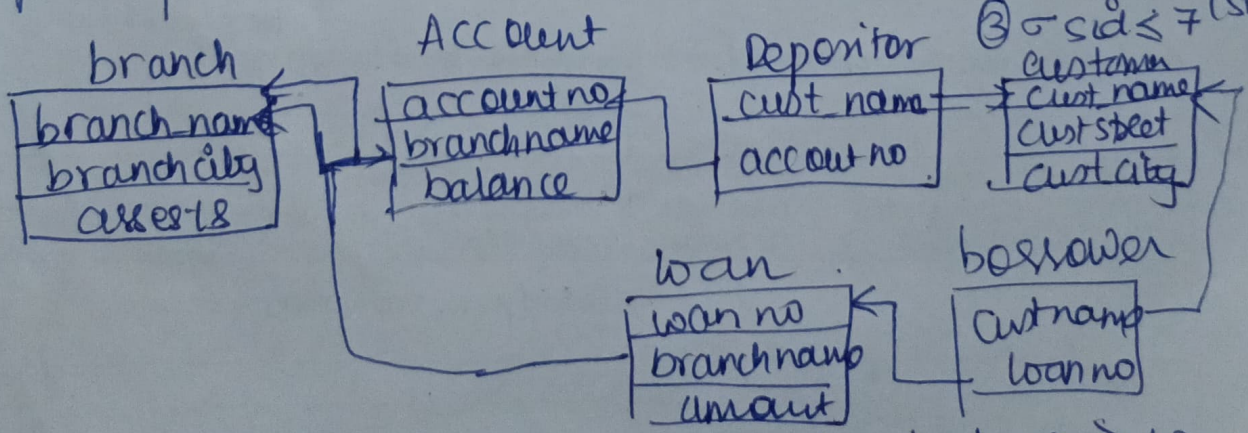
① $\sigma_{\text{Branch}='CS'}(\text{Student})$

The predicate P will involve.
Attributes from schema R of r
Comparison operators, $<, >, \leq, \geq, =, \neq$
Logical operators: $\wedge$ (AND), $\vee$ (OR), $\neg$ (Not)

② $\sigma_{\text{Branch}=\text{Civil}}(\text{Student})$

③ $\sigma_{\text{sid}\leq 7}(\text{Student})$

Eg)



① Find the details of accounts having balance ≥ 10000

$\sigma_{\text{balance}\geq 10,000}(\text{Account})$

(2) Find the details of the customer who in delhi (4)

$\sigma_{cust\_city = 'delhi'}(Customer)$

(3) Find the details of those loans having amount ≤ 5000 and from north delhi branch.

$\sigma_{branch\ name = 'North\ Delhi' \wedge amount \leq 5000}(loan)$

$\sigma_{amount \leq 5000}(\sigma_{branch\ name \neq 'North\ Delhi'}(loan))$

$\sigma_{branchname = 'Northern'}(\sigma_{amount \leq 5000}(loan))$ → Commutative in nature.

(4) Find those branch details which are in delhi or having assets more than 10,000?

$\sigma_{branch\ city = 'delhi' \vee assets > 10,000}(branch)$

(2) Project (or π): Project operation is a unary operation, takes one table at a time, fundamental operations. This operation keeps certain columns (attributes) from a relation and discards the other columns. Project means vertical partitioning. π (pi) symbol is used to represent project operation.

Syntax: $\pi_{(Column\_name)}(tablename)$

$\pi_{<attributelist>}(R)$ →

→ It works as select clause of SQL

→ Project is not commutative

(1) Find all branch name of the bank. (5)

$\Pi_{branchname}(branch)$

(2) Find all account_no along with the balance?

$\Pi_{account\_no,\ balance}(Acccount)$

(3) Find name of all the customer who have loan?

$\Pi_{cust\_name}(borraer)$

(4) Find all the details about branch

(branch)

Select & project

(1) Find those account no where balance is less than 10000?

$\Pi_{accountno.}(\sigma_{balance<10000}(Account))$

(2) Find those loan no. which are from CP branch with amount > 1000

$\Pi_{loan\ no.}(\sigma_{branch\ name='CP' \wedge amount>1000}(loan))$

(3) Find branch name and branchcity with assests more than 100000?

$\Pi_{branchname,branchcity}(\sigma_{assests > 1,00,000}(branch))$

(4) Find all the names of students from CS branch?

$\sigma_{branch='cs'}(\Pi_{names}(Student))$

$\Pi_{name}(\sigma_{branch='cs'}(Student))$