→ Integrity Constraints :- It is a condition that can be applied on a database schema to restrict the data acc to need

↓ if condition is satisfied

↓

Than only it can be stored in the database.

↓

Entity Integrity constraints

Referential Integrity constraints

→ EIC :- It states no primary key value can be null. Having null values for primary key implies that we can not identify any tuples.

→ RIC :- It ensure that value that appears in one relation for given set of attributes also appear for certain set of attributes in another relation

→ Domain constraints :

↓

A set of atomic values.

↳ A domain is a set of collection of data appear in column.

Triggers: It executes automatically as a side effect of modification in database.

It has 2 requirements to design

- Specify condition under which trigger is to be executed
- specify the actions to be taken when trigger executes

Query Language

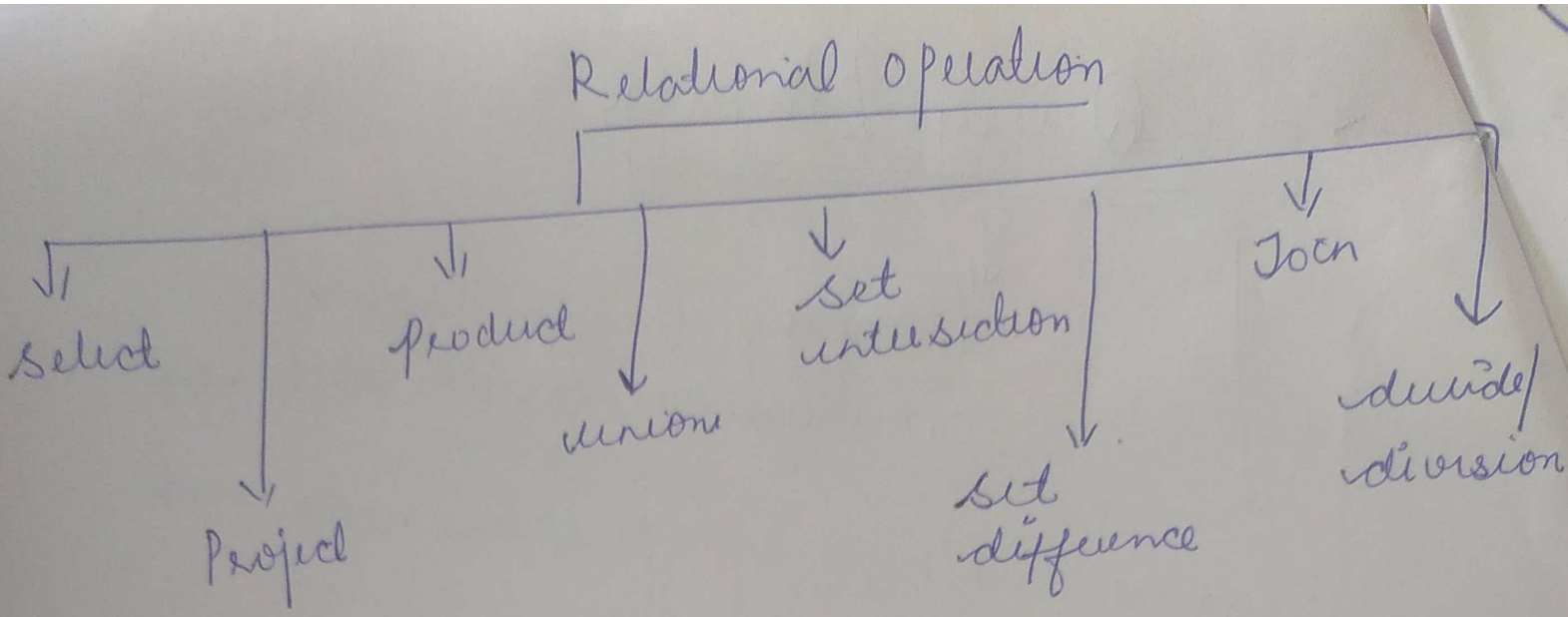
Language through which we communicate with database.

Reading and writing operations into and from existing database.

QL

- Relational Algebra
  - Step by step process to obtain the result of query.
  - Operators are used to perform queries.
- Relational calculus.

- RA is procedural Query language which takes relation as input and generate output as relation
- Operators can be unary or binary.



1) Select:- (σ) (sigma)

BranchName	LOAN_NO	Amount
Downtown	L-17	600
Redwood	L-21	200
Perridge	L-15	1500
Hill	L-23	400
Perridge	L-16	1300

$$\sigma_{Branchname = \text{"perridge"}}(LOAN)$$

Select tuple where Perridge is written

BN	LOAN No.	AMOUNT
Perridge	L-15	1500
Perridge	L-16	1300

Project operation :- (Π) (Pie)

↓

It shows list of those attributes which we want to appear in result, Rest of attributes are eliminated from the table.

Loan No.	Amount
L-17	600
L-21	200
L-15	1500
L-23	400
L-16	1300

rows or tuples

$\Pi_{loan\_number, amount}(loan)$ ← Table Name

3) The Union operation :- (∪) → eliminates the duplicate tuples

↓ The names of all Bank customers who have either an account and loan or both.

r and s tables

$r \cup s$ | r, s relations

→ constraint for union of 2 relations is that both relations must have same set of attribute

$\Pi_{customername}(borrower) \cup \Pi_{customername}(depositer)$

Customer Name
JOHN
SMITH
Hayes

← Name of customers who have either a loan or an account

4.) Set Difference operation :- (e) ——— "

↓

To find tuples that are in one relation but not in another.

eg → we have two tuples R and S. The operations contain all tuples that are in R but not in S

(r - s)

customer name
John
Thomas
Lindsay

← customer with account but no loan

5.) Product operation or Cartesian product operation

(×) ← denoted

↓

Allows us to combine info. from any two relations r1 and r2

e.g- r1 × r2

The relation schema for

r = borrower × loan

∇ Branch-name = "Perryridg" (borrower × loan)

↓ it will find all customers who have loan at Perridg branch.

Re-name operation :- The results of relation algebra expressions do not have name that we can use to refer them

So Give them name by using "rename" operator

$\rho_x(E)$ ⟶ It will return the result of expression E under the name x

7) Set intersection :- $(\cap)$

We want to find all customers who have both loan and an account.

$r \cap s = r - (r - s)$

$\Pi_{cust\text{-}name}(borrower) \cap \Pi_{C\text{-}N}(depositor)$

- Project → It projects the content of either a particular column or more than single column

- In formal Relational model Terminology a Row is called tuple
- A column header is called A Attribute
- A table is called Relation

- Select: The select operation is used to select a subset of tuples from a relation that satisfies selection condition.

The select operator is unary ie it is applied to single operation

Q: Difference Between Select / Project operator.