15CSE302 Database Management Systems

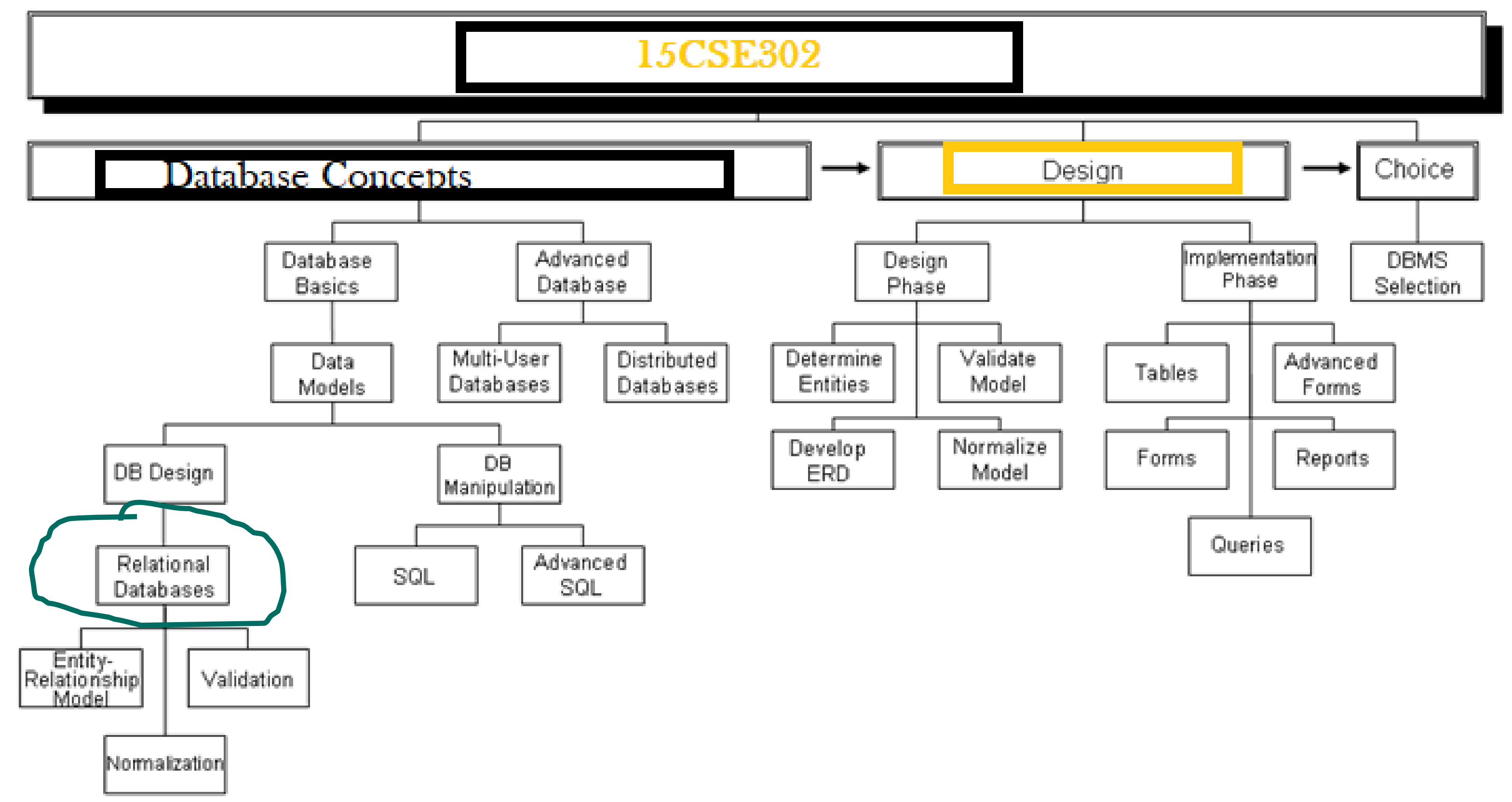
Lecture 4 Relational Algebra

B.Tech /III Year CSE/V Semester

LTPC 2023

DBMS Team
Dr G Jeyakumar
Bindu K R
Dr Priyanka Kumar
R. Manjusha
Department of CSE
Amrita School of Engineering

Syllabus



15CSE302

Database Concepts → Design → Choice

Database Basics

Data Models

DB Design

Relational Databases

Entity-Relationship Model

Validation

Normalization

DB Manipulation

SQL

Advanced SQL

Advanced Database

Multi-User Databases

Distributed Databases

Design Phase

Determine Entities

Validate Model

Develop ERD

Normalize Model

Implementation Phase

Tables

Advanced Forms

Forms

Reports

Queries

DBMS Selection

Brief Recap of Previous Lecture

- ❑ Structure of Relational Databases
- ❑ Keys
- ❑ Schema Diagrams

Today’s Lecture

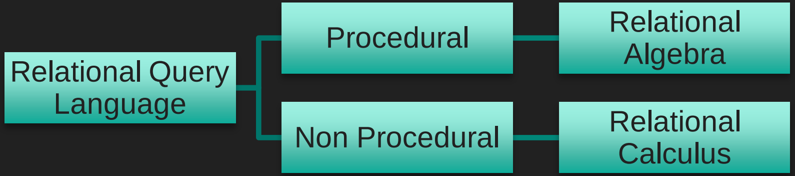
- ❑ **Relational Query Languages**
- ❑ **Relational Algebra**
- ❑ **RelaX tool**

Relational Query Language

- **Query languages**: Allow manipulation and retrieval of data from a database.
- Relational model supports simple, powerful QLs:
  - Strong formal foundation based on logic.
  - Allows for much optimization.
- Query Languages != programming languages!
  - QLs not expected to be "Turing complete".
  - QLs not intended to be used for complex calculations.
  - QLs support easy, efficient access to large data sets.

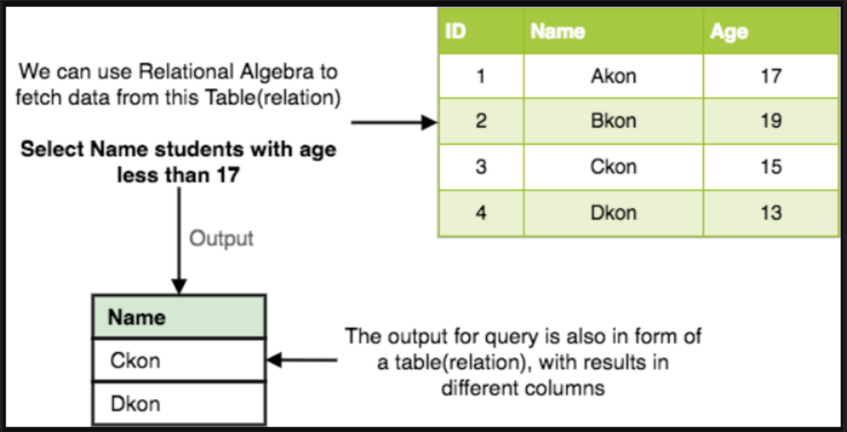
Relational Query Language

- A query language is a language in which a user requests information from the database.
- They are on a lever higher than that of standard programming languages.
- Two mathematical Query Languages form the basis for “real” languages (e.g. SQL), and for implementation:
  - Relational Algebra: More operational(procedural), very useful for representing execution plans.
  - Relational Calculus: Lets users describe what they want, rather than how to compute it. (Non-operational, declarative.)



Relational Algebra

- Algebra - language based on operators and a domain of values
- Operators map values taken from the domain into other domain values
- Hence, an expression involving operators and arguments produces a value in the domain
- When the domain is a set of all relations (and the operators are as described later), we get the *relational algebra*
- We refer to the expression as a *query* and the value produced as the *query result*
- **Relational algebra** is a procedural query language, which takes instances of relations as input and yields instances of relations as output.
- It uses operators to perform queries.



We can use Relational Algebra to fetch data from this Table(relation)

ID	Name	Age
1	Akon	17
2	Bkon	19
3	Ckon	15
4	Dkon	13

Select Name students with age less than 17

Output

Name
Ckon
Dkon

The output for query is also in form of a table(relation), with results in different columns

Relational Query Language

Two Categories

Procedural Languages

User instructs the system to perform
a sequence of operations on the database
to compute desired result.

non-procedural language

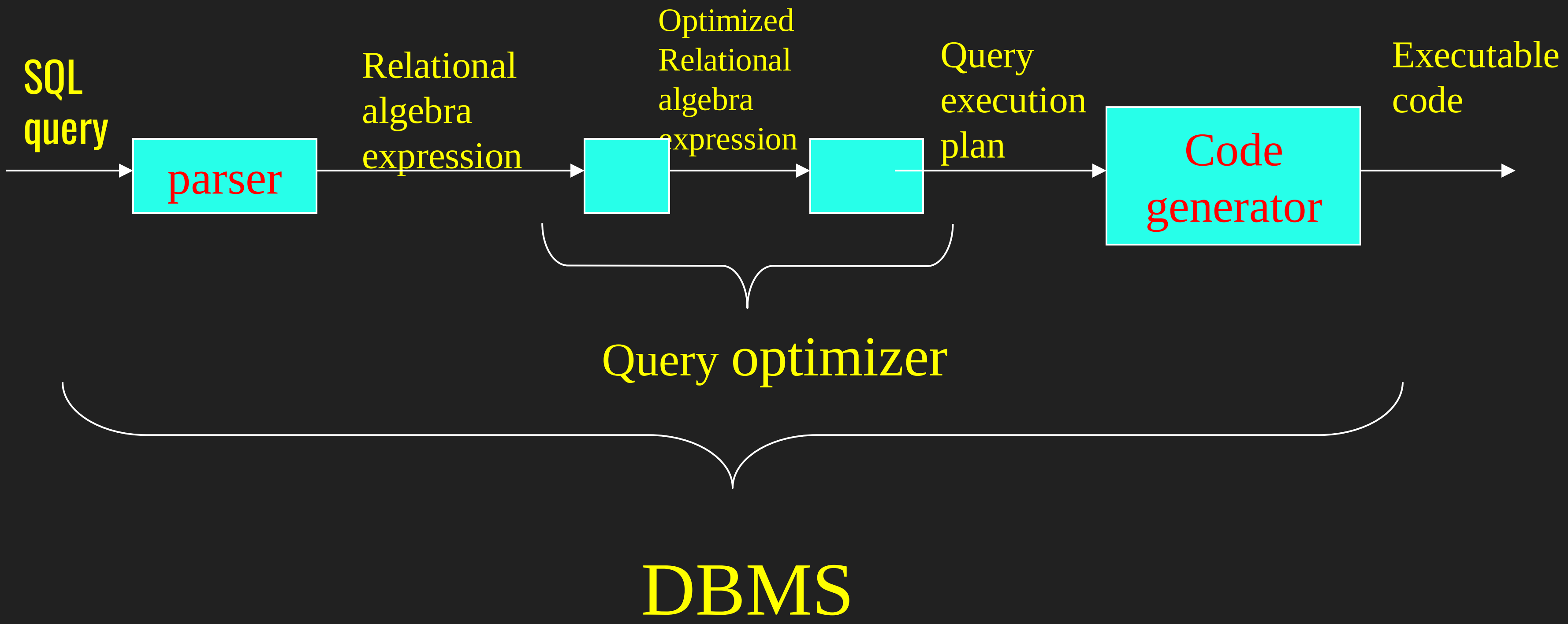
the user describes the desired results
without giving a specific procedure
for obtaining it.

* Query languages in practise use both the approaches.

Relational Algebra

- Domain: set of relations
- Basic operators: select, project, union, set difference, Cartesian product
- Derived operators: set intersection, division, join
- Procedural: Relational expression specifies query by describing an algorithm (the sequence in which operators are applied) for determining the result of an expression

Relational Algebra



Relational Operations

- The procedural query languages provide a set of operations that can be applied one or more relations.
- These operations have desired properties.
- These operations can be combined in a modular way.
- Common operations –
  - *Selection*
  - *Projection*
  - *Union*
  - *Difference*
  - *Intersection*
  - *Join (natural join and Cartesian Product)*

Relational Operations

- ❑ Basic operations:
  - □ *Selection* (σ) Selects a subset of rows from relation.
  - □ *Projection* (π) Deletes unwanted columns from relation.
  - □ *Cross-product* (X) Allows us to combine two relations.
  - □ *Set-difference* (-) Tuples in relation. 1, but not in relation. 2.
  - □ *Union* (U) Tuples in relation. 1 and in relation. 2.

Relational Operations

- Additional operations:
- Intersection(∩)
- *join(*⋈)
- division(÷)
- Renaming ρ

Since **each operation returns a relation**, operations can be *composed*!

(Relations are closed under the operators of the relational algebra.)

Select Operator

- □ Produce table containing subset of rows of argument table satisfying condition

$$\sigma_{condition}\ relation$$

- □ Example:

Person

$$\sigma_{Hobby='stamps'}(Person)$$

Person

Id	Name	Address	Hobby
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Id	Name	Address	Hobby
1123	John	123 Main	stamps
9876	Bart	5 Pine St	stamps

Selection Condition

- Operators: <, ≤, ≥, >, =, ≠
- Simple selection condition:
  - <*attribute*> *operator* <*constant*>
  - <*attribute*> *operator* <*attribute*>
- <*condition*> ∧ <*condition*> AND
- <*condition*> ∨ <*condition*> OR
- ! <condition> NOT

Selection Condition - Examples

- Display all persons whose id greater than 3000 or hobby is hiking
- Display all persons whose id between 3000 and 4000
- Display all persons whose hobby is not hiking
- Display all persons whose hobby is hiking

Person

Id	Name	Address	Hobby
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Selection Condition - Examples

- **Display all persons whose id greater than 3000 or hobby is hiking**

$\sigma_{Id>3000 \ \vee \ Hobby='hiking'}$ (Person)

Person

Id	Name	Address	Hobby
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Id	Name	Address	Hobby
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Selection Condition - Examples

❑ **Display all persons whose id between 3000 and 4000**

$\sigma_{Id>3000 \ \wedge \ Id <3999}$ **(Person)**

Person

Id	Name	Address	Hobby
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Id	Name	Address	Hobby
5556	Mary	7 Lake Dr	hiking

Selection Condition - Examples

- Display all persons whose hobby is not hiking

$\sigma_{(Hobby != 'hiking')}$ **(Person)**

Person

Id	Name	Address	Hobby
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Id	Name	Address	Hobby
1123	John	123 Main	stamps
1123	John	123 Main	coins
9876	Bart	5 Pine St	stamps

Selection Condition - Examples

- Display all persons whose hobby is hiking

$\sigma_{Hobby='hiking'}$ (Person)

Person

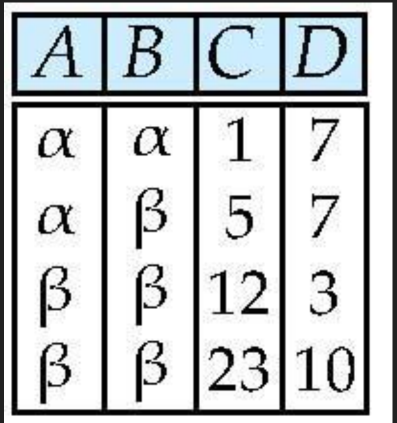
Id	Name	Address	Hobby
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Id	Name	Address	Hobby
5556	Mary	7 Lake Dr	hiking

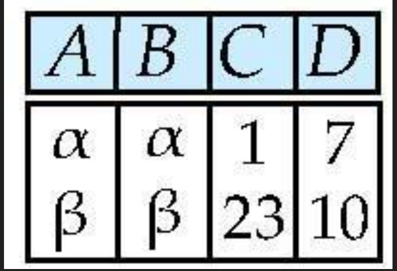
Select Operation – selection of rows (tuples)

Relation r



A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

- $\sigma_{A=B \wedge D > 5}(r)$



A	B	C	D
α	α	1	7
β	β	23	10

Project Operator Π

- □ **Produces table containing subset of columns of argument table**

$$\Pi_{\textit{attribute list}}(\textit{relation})$$

- □ Example:

Person $\qquad\qquad$ $\Pi_{\textit{Name, Hobby}}(\textbf{Person})$

Person

Id	**Name**	**Address**	**Hobby**
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Name	**Hobby**
John	stamps
John	coins
Mary	hiking
Bart	stamps

Project Operator Π

- Example: Person $\Pi_{Name,\ Address}(Person)$

Person

Id	Name	Address	Hobby
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Name	Address
John	123 Main
Mary	7 Lake Dr
Bart	5 Pine St

Result is a table (no duplicates)

Project Operator Π

Expression

$$\Pi_{Id,\ Name}\ (\sigma_{Hobby='stamps' \lor Hobby='coins'}\ (\text{Person})\ )$$

Person

Id	Name	Address	Hobby
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

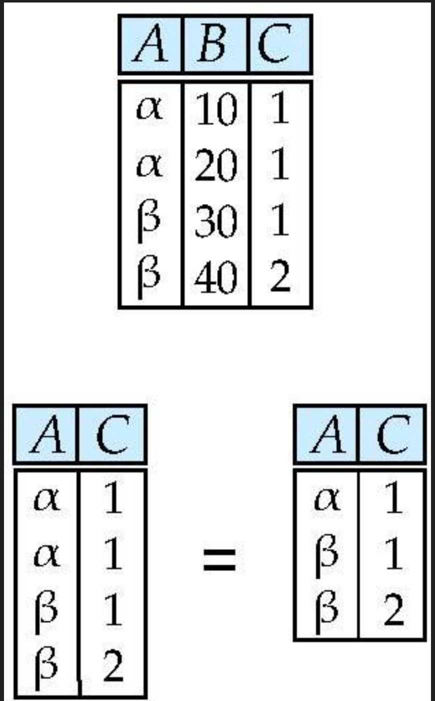
Person

Id	Name
1123	John
9876	Bart

Result is a table (no duplicates)

Project Operation – selection of columns (Attributes)

- Relation *r*:



A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

- $\prod_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Dept. of CSE., Amrita July 2020

Relax tool demo

https://dbis-uibk.github.io/relax/calc.htm#

Summary

- Relational Query Languages
- Relational Algebra
    - Selection
    - Projection

Next Lecture

- Relational Algebra
  - Set Operations
  - Cartesian Product
  - Join

References

- https://www.db-book.com/db6/index.html

Thank You

Happy to answer any questions ! ! !