15CSE302 Database Management Systems

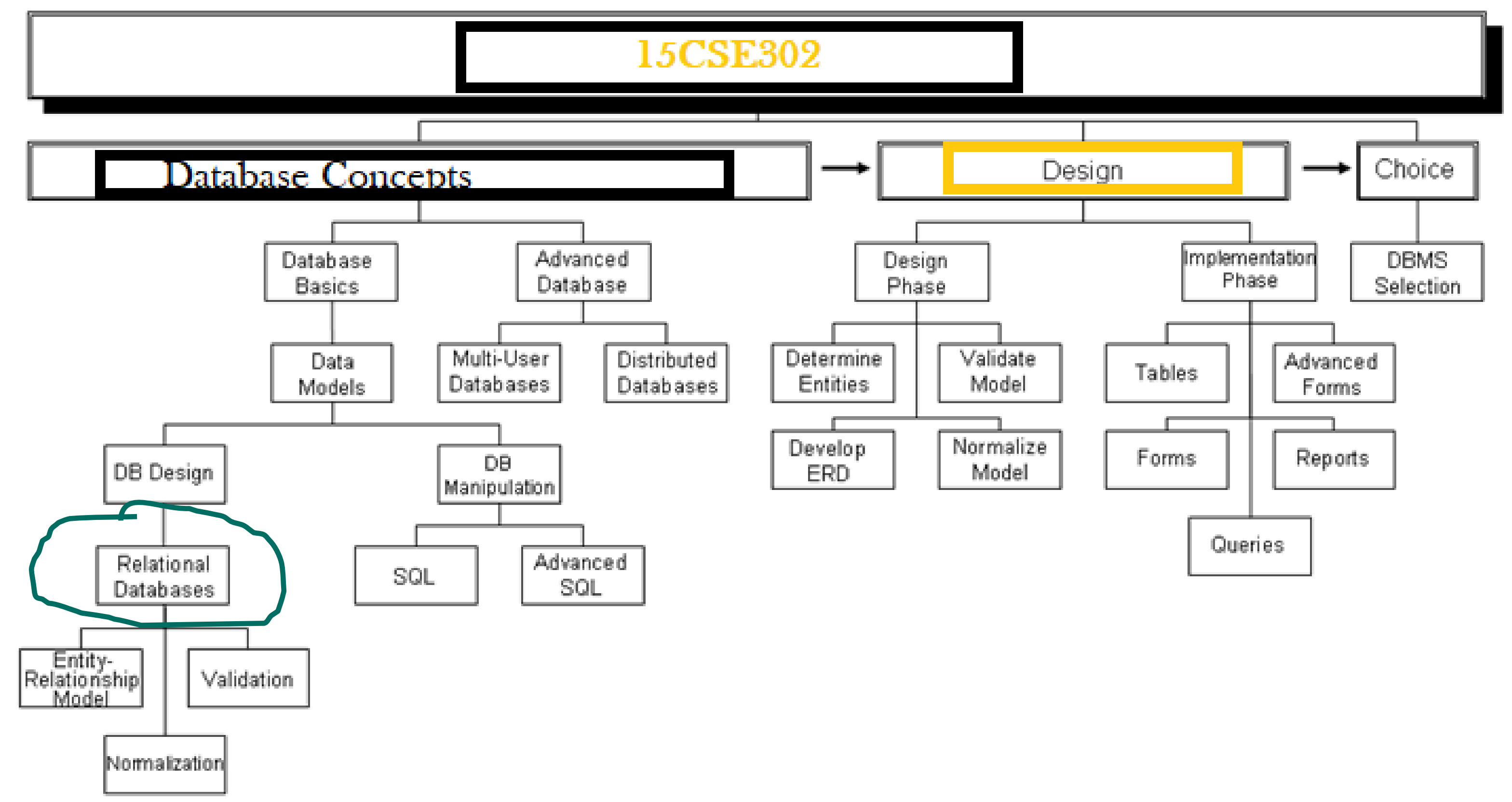
Lecture 5 Relational Algebra session 2

B.Tech /III Year CSE/V Semester

L T P C 2 0 2 3

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
Advanced Database
Design Phase
Implementation Phase
DBMS Selection
Data Models
Multi-User Databases
Distributed Databases
Determine Entities
Validate Model
Tables
Advanced Forms
DB Design
DB Manipulation
Develop ERD
Normalize Model
Forms
Reports
Queries
Relational Databases
SQL
Advanced SQL
Entity-Relationship Model
Validation
Normalization

Brief Recap of Previous Lecture

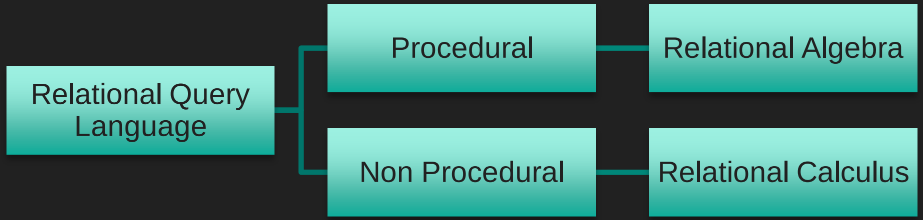
- **Relational Query Languages**
- **Relational Algebra -Selection and Projection**
- **RelaX tool**

Today’s Lecture

- Relational Algebra
  - Set operations
  - Cartesian Product
  - Join
  - Rename
- RelaX tool

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.)



Set Operators

- Relation is a set of tuples => set operations should apply
- Result of combining two relations with a set operator is a relation => all its elements must be tuples having same structure
- Hence, scope of set operations limited to *union compatible relations*

Union Compatible Relations

- Two relations are *union compatible* if
  - Both have same number of columns
  - Names of attributes are the same in both
  - Attributes with the same name in both relations have the same domain
- Union compatible relations can be combined using
  - union
  - Intersection
  - set difference

Example Tables:

Person (SSN, Name, Address, Hobby)

Professor (Id, Name, Office, Phone)

are not union compatible.

Π_{Name} (Person) and Π_{Name} (Professor)

Π_{Name} (Person) - Π_{Name} (Professor)

Union of two relations

- Relations *r, s:*

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- r ∪ s:

A	B
α	1
α	2
β	1
β	3

Set difference of two relations

- Relations *r*, *s*:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

$r - s$

A	B
α	1
β	1

Set intersection of two relations

- Relation r, s:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- $r \cap s$

A	B
α	2

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

Joining two relations -- Cartesian-product

Relations *r, s*:   *r* x *s*:

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

If r and s are two relations, r × s is the set of all concatenated tuples <x,y>,

- where x is a tuple in r and
- y is a tuple in s

(r and s need not be union compatible)

r × s is expensive to compute:

Factor of two in the size of each row

Quadratic in the number of rows

Cartesian-product – naming issue

Relations *r*, *s*:

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

r x *s*:

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

Composition of Operations

- Can build expressions using multiple operations
- Example: $\sigma_{A=C}\,(r \times s)$

- *r x s*

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

- $\sigma_{A=C}\,(r \times s)$

A	B	C	D	E
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b

Renaming a Table

- Allows us to refer to a relation, (say E) by more than one name.

$$\rho_X(E)$$

returns the expression *E* under the name *X*

- Relations *r*

A	B
α	1
β	2

r

- $r \times \rho_s(r)$

$r.A$	$r.B$	$s.A$	$s.B$
α	1	α	1
α	1	β	2
β	2	α	1
β	2	β	2

Joining two relations – Natural Join

- Let r and s be relations on schemas R and S respectively. Then, the "natural join" of relations R and S $r \bowtie s$ is a relation on schema $R \cup S$ obtained as follows:
    - Consider each pair of tuples t_r from r and t_s from s.
    - If t_r and t_s have the same value on each of the attributes in $R \cap S$, add a tuple t to the result, where
        - t has the same value as t_r on r
        - t has the same value as t_s on s

Natural Join Example

- Relations r, s:

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ε

s

- Natural Join

  $r \bowtie s$

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

$$\pi_{r.B,\ C,\ r.D,\ E}\,(\sigma_{\,r.B\,=\,s.B\ \wedge\ r.D\,=\,s.D}\ (r \times s)))$$

Summary of Relational Algebra Operators

Symbol (Name)	Example of Use
σ (Selection)	$\sigma_{\text{salary} >= 85000}(\textit{instructor})$ Return rows of the input relation that satisfy the predicate.
Π (Projection)	$\Pi_{\textit{ID, salary}}(\textit{instructor})$ Output specified attributes from all rows of the input relation. Remove duplicate tuples from the output.
x (Cartesian Product)	*instructor* x *department* Output pairs of rows from the two input relations that have the same value on all attributes that have the same name.
∪ (Union)	$\Pi_{\textit{name}}(\textit{instructor}) \cup \Pi_{\textit{name}}(\textit{student})$ Output the union of tuples from the *two* input relations.
- (Set Difference)	$\Pi_{\textit{name}}(\textit{instructor}) - \Pi_{\textit{name}}(\textit{student})$ Output the set difference of tuples from the two input relations.
⋈ (Natural Join)	*instructor* ⋈ *department* Output pairs of rows from the two input relations that have the same value on all attributes that have the same name.

[RelaX - relational algebra calculator 0.19.1](https://dbis-uibk.github.io/relax/calc.htm) Demo

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

Summary

- Relational Algebra
  - ★ Set operations
  - ★ Cartesian Product
  - ★ Join
  - ★ Rename
- RelaX tool

Next Lecture

- E R Model

References

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

Thank You

Happy to answer any questions ! ! !