

A decorative graphic on the left side of the slide, consisting of a network of white lines and small circles on a teal background, resembling a circuit board or a neural network.

# WEEK 6

## INTRODUCTION TO TUPLE RELATIONAL CALCULUS

CS3319

# STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
  - List the two types of relational calculus
  - Define the term: Declarative Language
  - Given 1 table, write a simple relational calculus expression that returns some values from rows based on a condition(s) that must be true.

# RELATIONAL CALCULUS

- 2 Types:

- Tuple Relational Calculus
- Domain Relational Calculus

*what we want  
→  
how to get what we want.*

- In relational calculus we write ONE declarative statement that states WHAT is to be returned rather than how (in what order). Thus we do NOT say things like first return all the Employee IDs of people with the last name Simpson THEN return the hours they work on a project, THEN retrieve the project name as we would in Relational Algebra → In Relational Calculus we do express it in ONE statement.
- Declarative languages: describe the desired results without explicitly listing commands or steps that must be performed
- It is a Nonprocedural language
- Any query that can be written in relational algebra can also be written in relational calculus (i.e. the expressive power is identical).

# TUPLE RELATIONAL CALCULUS

- General Form:

*a row* →  $\{t \mid \text{COND}(t)\}$

*same condition*  
*return a row that condition for t is true*

where  $t$  is a tuple variable and  $\text{COND}(t)$  is a Boolean expression involving  $t$

- $\text{COND}(t)$  **must** evaluate to be TRUE for  $t$  in order to return that tuple.

- Example:

- To find all employees whose age is  $> 45$ , we write the following:

*t is in emp table*  
 $\{t \mid \text{EMPLOYEE}(t) \text{ AND } t.\text{Age} > 45\}$

- If we only want first and last name for employees over 45 we write:

$\{t.\text{Lname}, t.\text{Fname} \mid \text{EMPLOYEE}(t) \text{ and } t.\text{Age} > 45\}$

## Answer

EmpID	Fname	Lname	Age	Salary
22	Homer	Simpson	56	1000
66	Francine	Smith	47	3000

## EMPLOYEE

EmpID	Fname	Lname	Age	Salary
22	Homer	Simpson	56	1000
33	Ned	Flanders	34	2000
44	Brian	Griffin	7	4000
55	Wilma	Flintstone	43	3000
66	Francine	Smith	47	3000

# TUPLE RELATIONAL CALCULUS

- General Form:

$$\{t \mid \text{COND}(t)\}$$

where  $t$  is a tuple variable and  $\text{COND}(t)$  is a Boolean expression involving  $t$

- $\text{COND}(t)$  **must** evaluate to be TRUE for  $t$  in order to return that tuple.

- Example:

- To find all employees whose age is  $> 45$ , we write the following:

$$\{t \mid \text{EMPLOYEE}(t) \text{ AND } t.\text{Age} > 45\}$$

- If we only want first and last name for employees over 45 we write:

$$\{t.\text{Lname}, t.\text{Fname} \mid \text{EMPLOYEE}(t) \text{ and } t.\text{Age} > 45\}$$

**Answer**

Fname	Lname	Age	Salary
Homer	Simpson	56	1000
Francine	Smith	47	3000

## EMPLOYEE

EmpID	Fname	Lname	Age	Salary
22	Homer	Simpson	56	1000
33	Ned	Flanders	34	2000
44	Brian	Griffin	7	4000
55	Wilma	Flintstone	43	3000
66	Francine	Smith	47	3000

in a relational  
columns, we don't  
show  
duplicates.

- In general, we must specify three things:

1. The **range relation (table)** of  $t$
2. The **condition** which must evaluate to TRUE
3. A **set of attributes** to be retrieved if you don't want all the attributes OR just the tuple variable  $t$  (for all the attributes)

**$\{t.Lname, t.Fname \mid \text{EMPLOYEE}(t) \text{ and } t.Age > 45\}$**

**QUESTION:** Retrieve the birth date and address of the employee whose name is 'Jon Mortensen' assuming this is one of your tables (relations):

Employee

FName	Minit	Lname	<u>SSN</u>	BDate	Address	Sex	Salary	SuperSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

**ANSWER:**

**$\{t.Bdate, t.Address \mid \text{EMPLOYEE}(t) \text{ AND } t.Fname = \text{"Jon"} \text{ and } t.Lname = \text{"Mortensen"}\}$**