COURSE NAME: DBMS
COURSE CODE:23AD2102A

TOPIC:

DATA SECURITY USING ACCESS CONTROL

Session - 13

AIM OF THE SESSION

To familiarize students with the advance and complex Subqueries in PostgreSQL.

INSTRUCTIONAL OBJECTIVES

This Session is designed to:

1. Discuss the subqueries.
2. Various guidelines and types of subqueries.

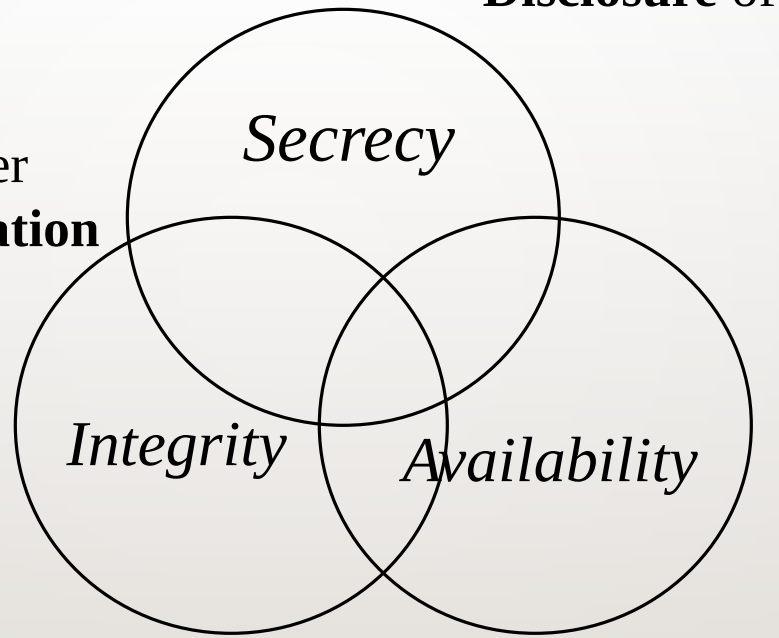
LEARNING OUTCOMES

At the end of this session, you should be able to understand the basic concepts of Subqueries and learn how to write complex subqueries with PostgreSQL commands.

SECURITY OBJECTIVES

Prevent/detect/deter improper **Disclosure** of information

Prevent/detect/deter Improper **modification** of information



Prevent/detect/deter improper **Denial of access** to services

DATABASES

- Collection of
  - interrelated data and
  - set of programs to access the data
- Convenient and efficient processing of data
- Database Application Software

DATABASE SECURITY

- Protect Sensitive Data from
  - Unauthorized disclosure
  - Unauthorized modification
  - Denial of service attacks
- Security Controls
  - Security Policy
  - Access control models
  - Integrity protection
  - Privacy problems
  - Fault tolerance and recovery
  - Auditing and intrusion detection

Protection of Data Confidentiality

- Access control – which data users can access
- Information flow control – what users can do with the accessed data
- Data Mining

Access Control

- Ensures that all direct accesses to object are authorized
- Protects against accidental and malicious threats by regulating the read, write and execution of data and programs

ACCESS CONTROL

Requires:

- Proper user identification
- Information specifying the access rights is protected form modification

Access Control

- Access control components:
  - Access control policy: specifies the authorized accesses of a system
  - Access control mechanism: implements and enforces the policy

ACCESS CONTROL

- Subject: active entity that requests access to an object
  - e.g., user or program
- Object: passive entity accessed by a subject
  - e.g., record, relation, file
- Access right (privileges): how a subject is allowed to access an object
  - e.g., subject *s* can read object *o*

Protection Object

- Database
- Relation
- Record
- Attribute
- Element

Advantages vs. disadvantages of supporting different granularity levels

Relation-Level Granularity

Confidential relation

Person-name	Company-name	Salary
Smith	BB&C	$43,982
Dell	Bell	$97,900
Black	BB&C	$35,652

Tuple-level Granularity

Works

Person-name	Company-name	Salary	
Smith	BB&C	$43,982	Public
Dell	Bell	$97,900	Conf.
Black	BB&C	$35,652	Public

Attribute-Level Granularity

Works

Person-name Publ.	Company-name Publ.	Salary Conf.
Smith	BB&C	$43,982
Dell	Bell	$97,900
Black	BB&C	$35,652

Cell-Level Granularity

Works

Person-name		Company-name		Salary	
Smith	P	BB&C	P	$43,982	C
Dell	C	Bell	C	$97,900	C
Black	P	BB&C	C	$35,652	C

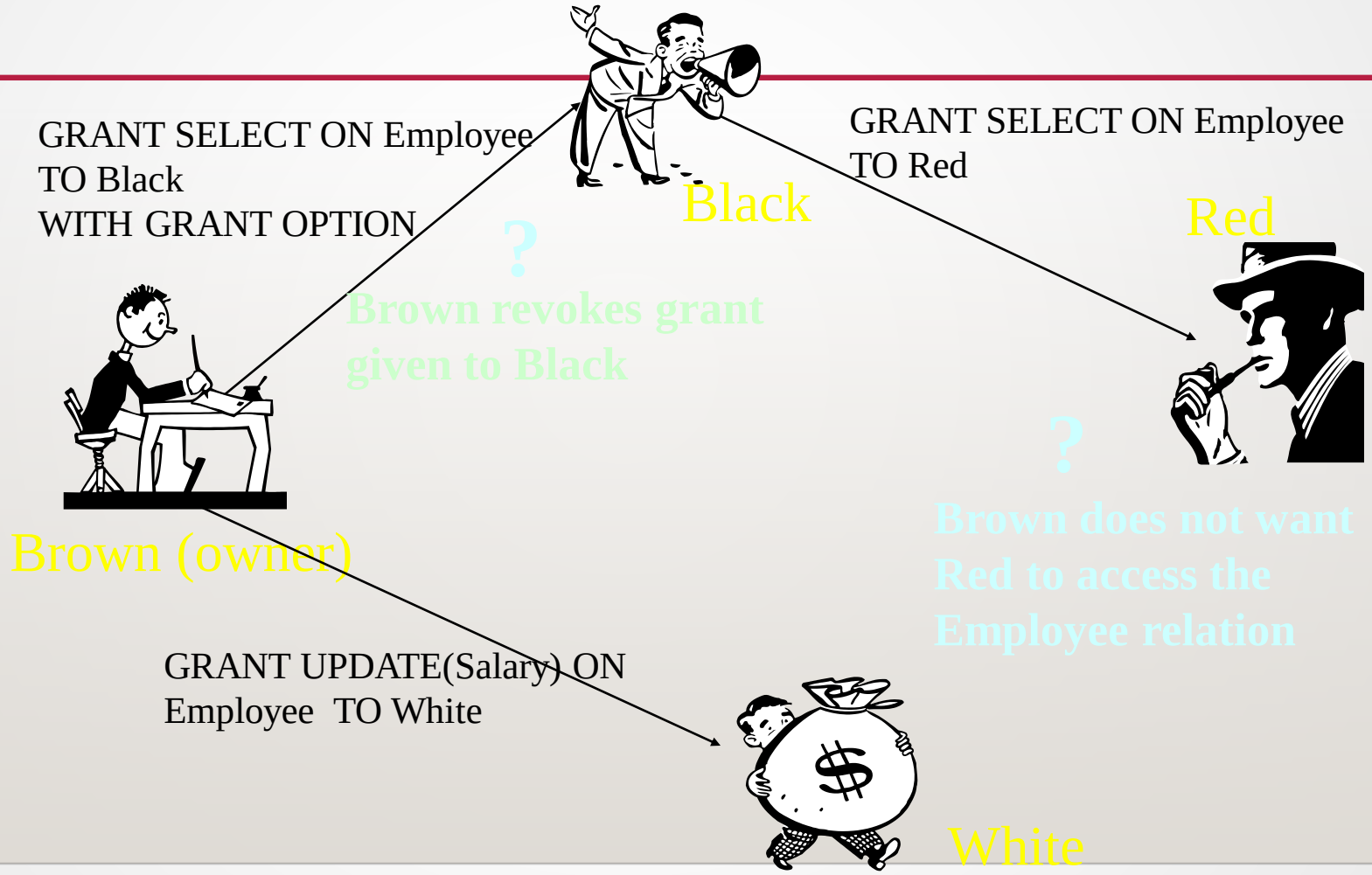
Access Control Policies

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
- Role-Based Access Control (RBAC)

Discretionary Access Control (DAC)

- *For* each subject access right to the objects are defined
  - (subject, object, +/- access mode)
  - (Black, Employee-relation, read)
- User based
- Grant and Revoke
- Problems:
  - Propagation of access rights
  - Revocation of propagated access rights

DAC BY GRANT AND REVOKE



Implementation

Access Control List (column) (ACL)

File 1	**File 2**
Joe:Read	Joe:Read
Joe:Write	Sam:Read
Joe:Own	Sam:Write
	Sam:Own

Capability List (row)

Joe: File 1/Read, File 1/Write, File 1/Own, File 2/Read
Sam: File 2/Read, File 2/Write, File 2/Own

Access Control Triples

Subject	Access	Object
Joe	Read	File 1
Joe	Write	File 1
Joe	Own	File 1
Joe	Read	File 2
Sam	Read	File 2
Sam	Write	File 2
Sam	Own	File 2

Access Control Mechanisms

- Security through Views
- Stored Procedures
- Grant and Revoke
- Query modification

Security Through Views

- Assign rights to access predefined views

  CREATE VIEW ***Outstanding-Student***
  AS SELECT NAME, COURSE, GRADE
  FROM ***Student***
  WHERE GRADE > B

Problem:

Difficult to maintain updates.

Stored Procedures

- Assign rights to execute compiled programs
- **GRANT RUN ON <program> TO <user>**

Problem:

Programs may access resources for which the user who runs the program does not have permission.

Grant and Revoke

GRANT <privilege> ON <relation>

To <user>

[WITH GRANT OPTION]

- GRANT SELECT * ON *Student* TO Matthews
- GRANT SELECT *, UPDATE(GRADE) ON *Student* TO FARKAS
- GRANT SELECT(NAME) ON *Student* TO Brown

GRANT command applies to base relations as well as views

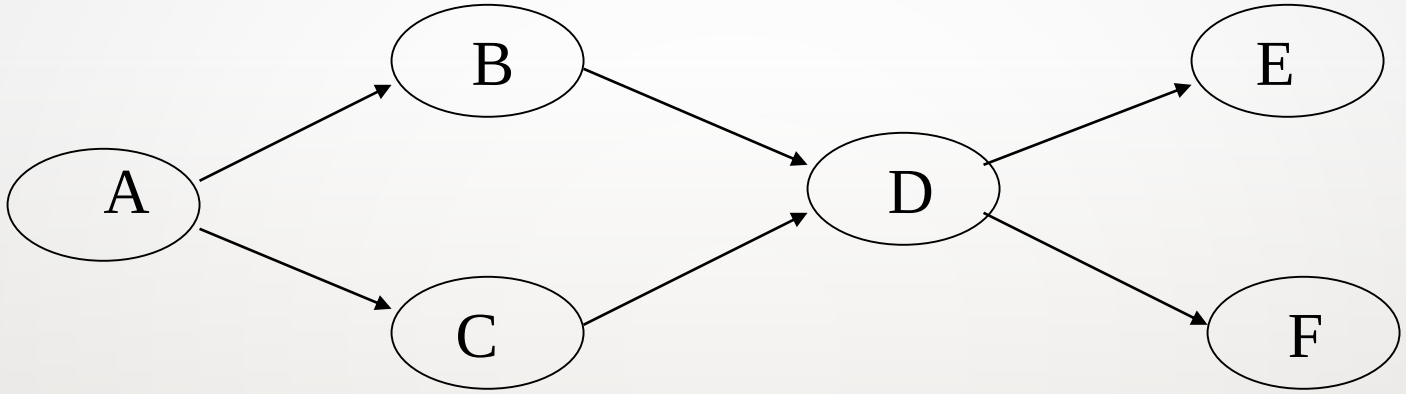
Grant and Revoke

REVOKE <privileges> [ON <relation>]

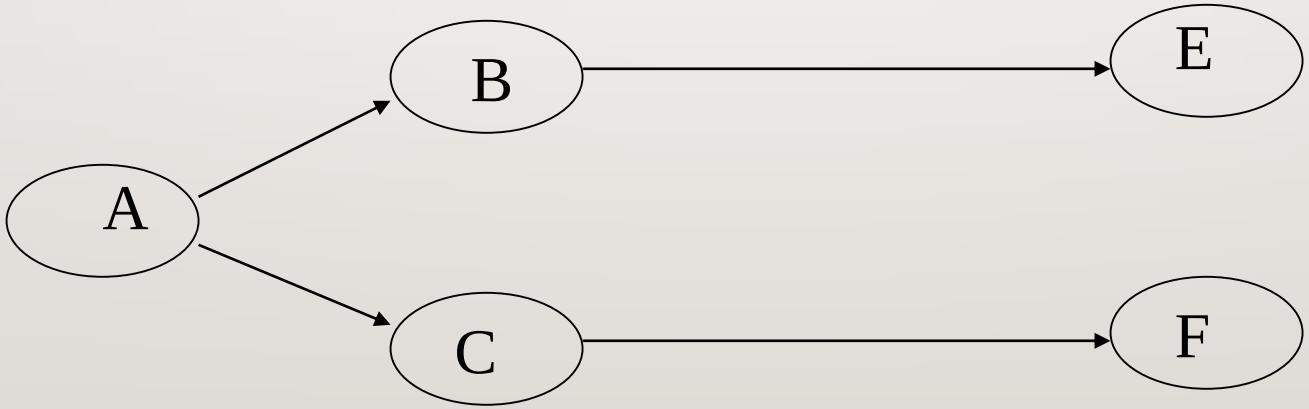
FROM <user>

- REVOKE SELECT* ON *Student* FROM Blue
- REVOKE UPDATE ON *Student* FROM Black
- REVOKE SELECT(NAME) ON *Student* FROM Brown

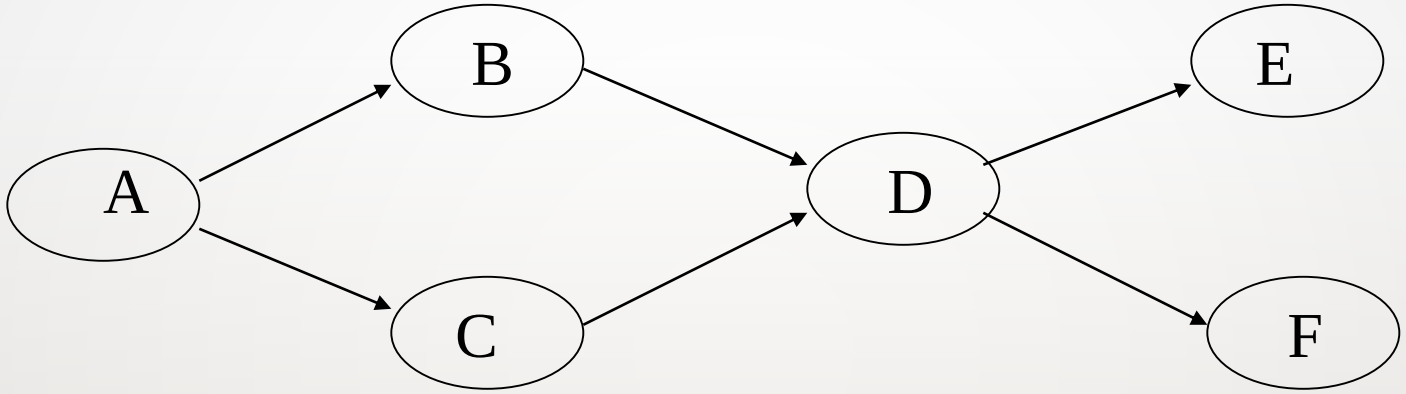
Non-cascading Revoke



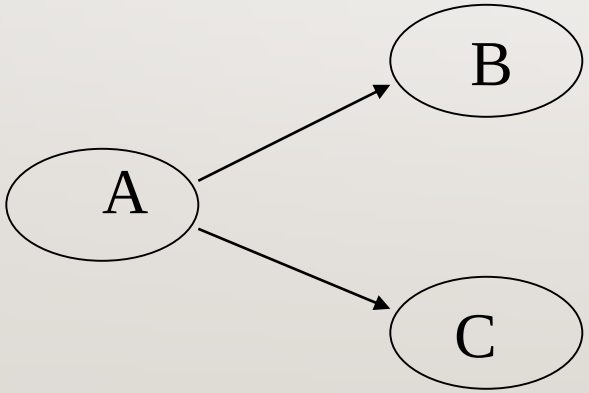
A revokes D's privileges



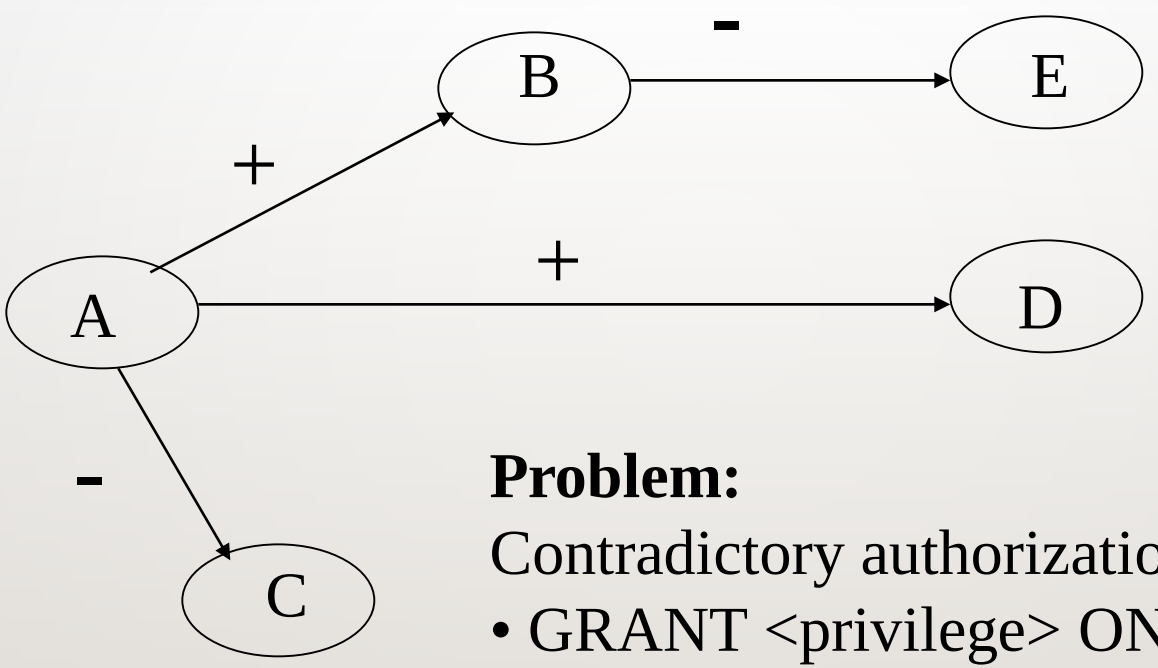
Cascading Revoke



A revokes D's privileges



Positive and Negative Authorization

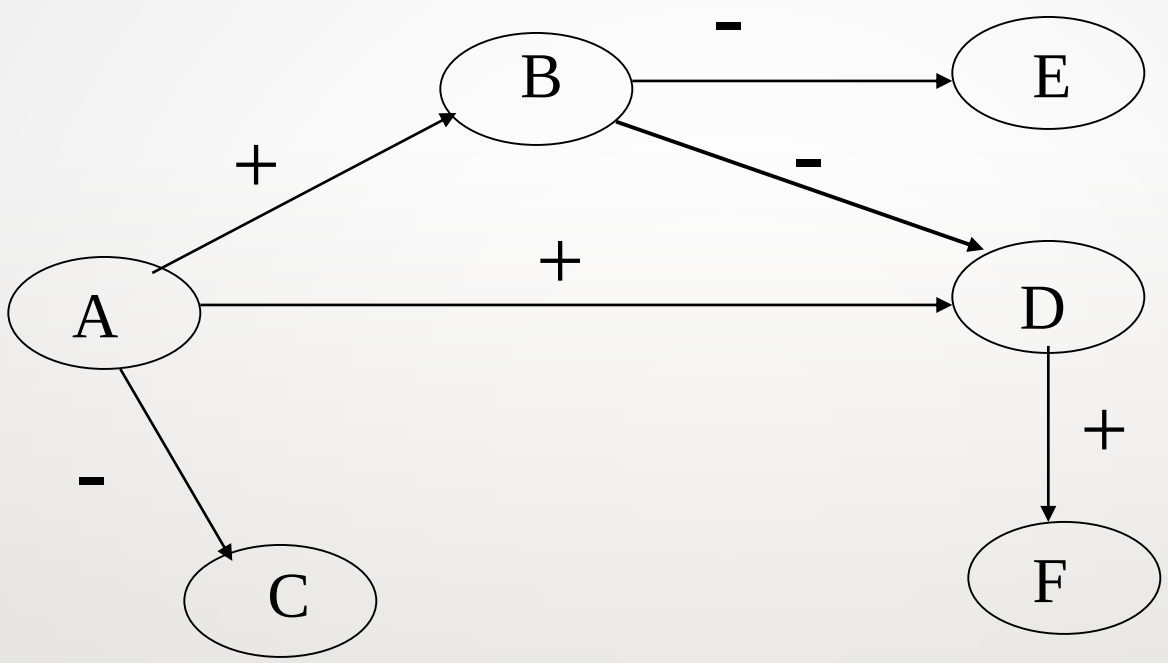


Problem:

Contradictory authorizations

- GRANT <privilege> ON *X* TO <user>
- DENY <privilege> ON *X* TO <user>

Negative Authorization



What should happen with the privilege given by D To F?

Query Modification

- GRANT SELECT(NAME) ON *Student* TO Blue WHERE COURSE="CSCE 590"

- **Blue's query:**
  SELECT *
  FROM *Student*

- **Modified query:**
  SELECT NAME
  FROM *Student*
  WHERE COURSE="CSCE 590"

DAC Overview

- Advantages:
  - Intuitive
  - Easy to implement
- Disadvantages:
  - Inherent vulnerability (look TH example)
  - Maintenance of ACL or Capability lists
  - Maintenance of Grant/Revoke
  - Limited power of negative authorization

Mandatory Access Control (MAC)

- Security label
  - Top-Secret, Secret, Public
- Objects: security classification
  - File 1 is Secret, File 2 is Public
- Subjects: security clearances
  - Brown is cleared to Secret, Black is cleared to Public
- Dominance (≥)
  - Top-Secret ≥ Secret ≥ Public

MAC

- Access rights: defined by comparing the security classification of the requested objects with the security clearance of the subject
- If access control rules are satisfied, access is permitted
- Otherwise access is rejected
- Granularity of access rights!

MAC – BELL-LAPADULA (BLP) MODEL

- Single security property: a subject S is allowed a read access to an object O only if label(S) dominates label(O)
- Star-property: a subject S is allowed a write access to an object O only if label(O) dominates label(S)

No direct flow of information from high security objects to low security objects!

MULTILEVEL SECURITY

- Multilevel security → users at different security level, see different versions of the database
- Problem: *different versions* need to be kept consistent and coherent *without downward signaling channel* (covert channel)

MULTILEVEL RELATION

- Schema $R(A_1,C_1,\ldots,A_n,C_n,T_c)$
  - R: relation name
  - A_i: attribute name
  - C_i: security classes
  - T_c: Tuple security classes
- Instantiation of relation: sets of tuples of the form $<a_1,c_1,\ldots,a_n,c_n,t_c>$
  - a_i: attribute value
  - c_i: attribute classification label
  - t_c: tuple classification label

Multilevel Relation Example

SSN	λ(SSN)	Course	λ(Course)	Grade	λ(Grade)
111-22-3333	S	CSCE 786	S	A	TS
444-55-6666	S	CSCE 567	S	C	TS

Top-secret user sees all data

Secret user sees Secret-View:

SSN	λ(SSN)	Course	λ(Course)	Grade	λ(Grade)
111-22-3333	S	CSCE 786	S	null	S
444-55-6666	S	CSCE 567	S	null	S

SUMMARY

An aggregate function in SQL performs a calculation on multiple values and returns a single value. SQL provides many aggregate functions that include avg, count, sum, min, max, etc. An aggregate function ignores NULL values when it performs the calculation, except for the count function

SELF-ASSESSMENT QUESTIONS

1. Which of the following is true about sub-queries?

a) They execute after the main query executes.
b) They execute in parallel to the main query.
c) The user can execute the main query and then, if wanted, execute the sub-query.
d) They execute before the main query executes.

2. Which of the following clause is mandatorily used in a sub-query?

(a) SELECT
(b) WHERE
(c) ORDER BY
(d) GROUP BY

SELF-ASSESSMENT QUESTIONS

3. Which of the following multi-row operators can be used with a sub-query?

(a) IN
(b) ANY
(c) ALL
(d) ALL OF THE ABOVE

4. Which of the following is true about the result of a sub-query?

a) The result of a sub-query is generally ignored when executed.
b) The result of a sub-query doesn't give a result, it is just helpful in speeding up the main query execution.
c) The result of a sub-query is used by the main query.
d) The result of a sub-query is always NULL.

TERMINAL QUESTIONS

1. Describe various types of SQL complex subqueries.

2. List out the guidelines for creating the SQL subqueries.

3. Analyze the use of ALL,IN, or ANY operator while using subqueries in PostgreSQL.

REFERENCES FOR FURTHER LEARNING OF THE SESSION

Reference Books:

1. Database System Concepts, Sixth Edition, Abraham Silberschatz, Yale University Henry, F. Korth Lehigh University, S. Sudarshan Indian Institute of Technology, Bombay.

2. An Introduction to Database Systems by Bipin C. Desai

3. Fundamentals of Database Systems, 7th Edition, RamezElmasri, University of Texas at Arlington, Shamkant B. Navathe, University of Texasat Arlington.

Sites and Web links:

1. https://www.geeksforgeeks.org/postgresql-create-table/

2. https://www.tutorialsteacher.com/postgresql

THANK YOU

Team – DBMS