Practical Assignment: Access Control and Inference Control in Database Security

1. Overview

This practical assignment integrates Access Control Models (DAC, RBAC) and Inference Control techniques to secure sensitive data in a database system. Students will implement security controls, simulate attacks, and apply inference control mechanisms .

The Employees table contains sensitive salary data and represents a vulnerable schema for inference attacks.

EmpID	FullName	Salary
1	Ali	120000
2	Asser	110000
3	Mona	100000
4	Fatma	90000
5	Gehad	80000
6	Ahmed	70000

Part 1 — DAC Implementation

1. Create SQL Server logins for user_public and user_admin.
2. Map them to users in the database (general and admin1).
3. Create roles (public_role, admin_role).
4. Grant only limited access to public_role and full access to admin_role.
5. Test access by executing queries as both users.

Expected: Public users can only view anonymized data, while admin can view mapped data.

6. Create a situation where a user in public_role **indirectly gains access** to a restricted table through a *view* or *inherited PUBLIC permission*.

- How the attack works (e.g., executing a view that joins Employees to a public table).
- How to fix it ?

Part 2 — RBAC Implementation

1. Create roles read_onlyX and insert_onlyX.
2. Assign users to these roles.
3. Use GRANT and REVOKE to enforce least privilege.
4. Verify access by executing different operations.

Expected: Different roles have different privileges according to the least privilege principle.

5. Role Hierarchy & Composite Role (role inheritance)

Create a `power_user` role that *inherits* privileges from both `read_onlyX` and `insert_onlyX`. Assign a user to `power_user` and show that they gain both sets of privileges. Then remove one underlying role (`REVOKE`) and show how the `power_user` behavior changes.

Part 3 — Inference Attack Simulation

Demonstrate how a malicious user can align ordered results from vPublicNames and vPublicSalaries to infer individual salaries without direct access to the mapping table.

Part 4 — Inference Control by Randomization

1. Regenerate Public IDs using NEWID().
2. Restrict access to AdminMap table.
3. Deny CREATE VIEW to public_role.
4. Verify that the inference attack no longer works.

Part 5 — Functional Dependency Inference

Given FDs:

- FD_1: EmpID → Dept,
- FD_2: Title → Grade
- FD_3: Dept, Grade → Bonus

Students must:

1. Compute closure Q^+ of {Dept, Title}.

2. Show Bonus $\in Q^+$.
3. Decide whether to reject or transform the query.

Part 6 — Inference via Aggregates

1. Use AVG views including and excluding a target user to infer sensitive values.
2. Apply K-anonymity rule to prevent inference.
3. Demonstrate that inference is blocked after applying protection.

11. Evaluation Rubric

Task	Marks
DAC & RBAC Implementation	2
Inference attack demonstration	2
Randomization & control enforcement	2
FD closure analysis	2
Aggregate attack and defense	2

ملحوظة: المناقشات بالاسبوع السادس بالكلية أثناء السكاشن

درجة الأسيمنت : 5 من إجمالي أعمال السنة

90% من الدرجة تحدد علي مناقشتك

10 % علي الكود وتشغيله بشكل صحيح