WEEK 8

STATISTICAL DATABASE SECURITY

STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
  - List at least 3 of the functions that would be accessible to a user who is doing statistical queries
  - Identify situations where statistical queries could reveal information that should be private
  - List 3 strategies that employed to avoid the loss of private data via statistical queries

STATISTICAL DATABASE SECURITY

- Often stats will be run on a database to find out data about customers, employees, etc…
- For example:

  Find the average salary & age of customers who live in Ontario.

- The user running these stats must have access to the data, however we might not want them to see the actual data per record, just run stats on the data (to keep the information private)

- Statistical users have access to certain functions like ***COUNT, MAX, MIN, AVERAGE,*** and ***STANDARD DEVIATION*** but not the individual data, called statistical queries

  - Thus, we can do this:

    SELECT COUNT(*) FROM employee WHERE sex='F'

  - BUT NOT THIS:

    SELECT * FROM employee WHERE sex = 'F'

    private info hidden.

- HOWEVER in some cases it is possible to deduce the values of an individuals records as follows:

- Example: We want to know Dr. Hanan Lutfiyya’s salary but we shouldn’t be allowed to find that out for her. We know that she is the only tenured Prof. in the computer science department who is female. We create a query as follows:

```
SELECT COUNT(*) AS Result FROM faculty WHERE Dept='Computer Science‘ AND
   Position='Tenured' AND Sex = 'F'
```

- and we get the following answer from our database:

```
RESULT
1
```

QUESTION: Now we know only 1 such Prof. exists, so what query do we write now:

```
SELECT MAX(salary) AS Result FROM faculty WHERE Dept='Computer
Science‘ AND Position='Tenured' AND Sex = 'F‘

OR

SELECT AVERAGE(salary) AS Result FROM faculty WHERE Dept='Computer
Science‘ AND Position='Tenured' AND Sex = 'F'
```

STRATEGIES FOR HANDLING SUCH LEAKS OF INFORMATION:

QUESTION: Do you have any ideas/suggestions for how you could handle such leaks of information?

limit queries that return tresholds lowers than certain

limit repeat queries with repeat

introduce noise to data.

REVIEW → TOOLS FOR DESIGNING ACCESS CONTROL FOR A TYPICAL COMMERCIAL DATABASE APPLICATION:

- Use VIEWS to narrow down exactly what a user needs
- Use roles if your db package has them. If not, design the roles off-line (the way you would use ER diagrams), and make sure you implement the roles with the other aspects of the application
- Most systems have ways of making applications/packages with specific interfaces. E.g. you would design the buttons on a bank machine interface for only those operations a casual user is allowed to do.