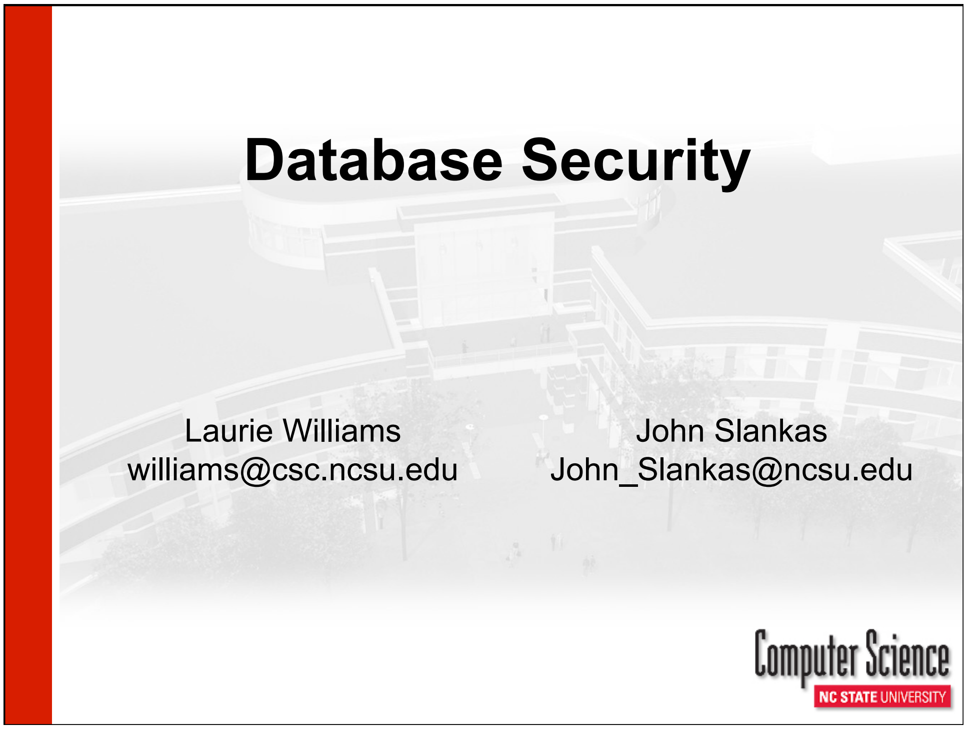
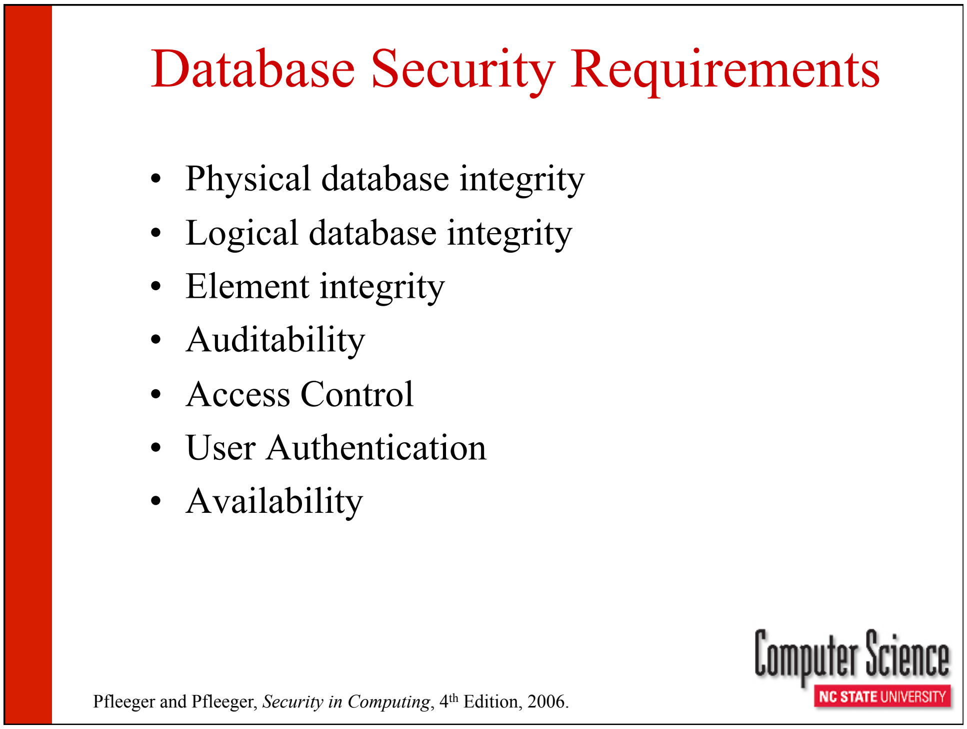


Database Security

Laurie Williams
williams@csc.ncsu.edu

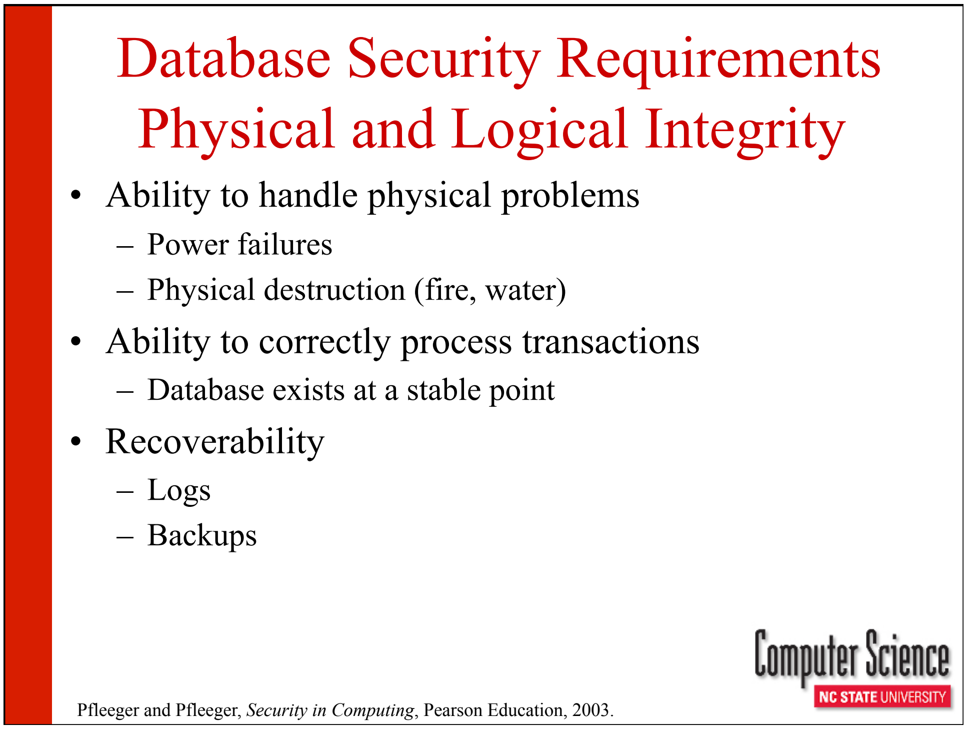
John Slankas
John_Slankas@ncsu.edu



Database Security Requirements

- Physical database integrity
- Logical database integrity
- Element integrity
- Auditability
- Access Control
- User Authentication
- Availability

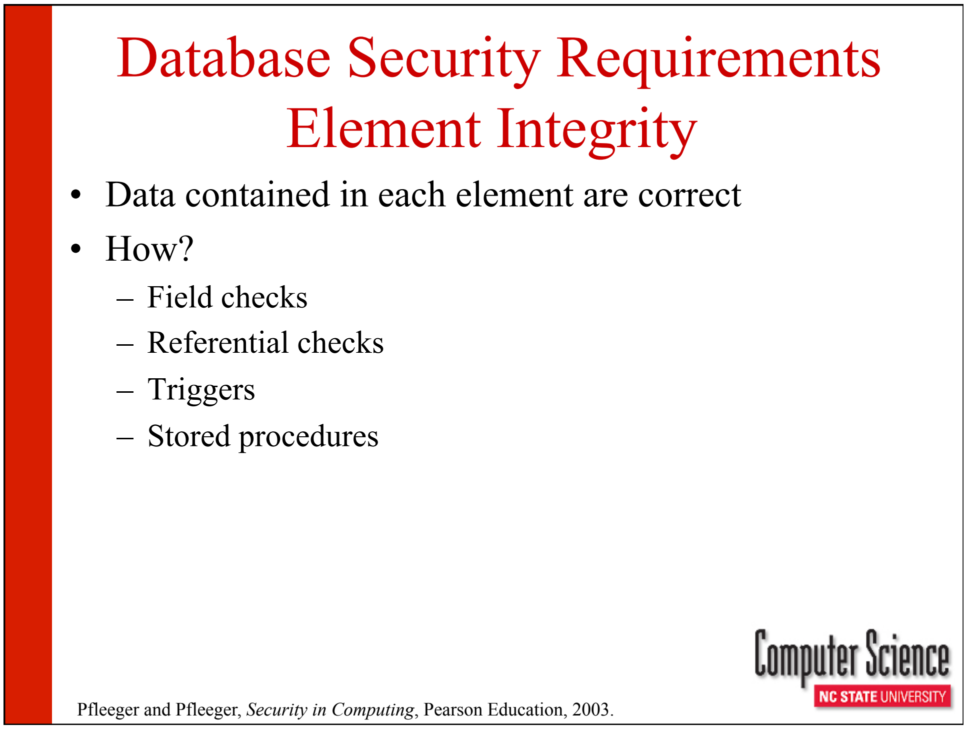
Pfleeger and Pfleeger, *Security in Computing*, 4th Edition, 2006.



Database Security Requirements Physical and Logical Integrity

- Ability to handle physical problems
  - Power failures
  - Physical destruction (fire, water)
- Ability to correctly process transactions
  - Database exists at a stable point
- Recoverability
  - Logs
  - Backups

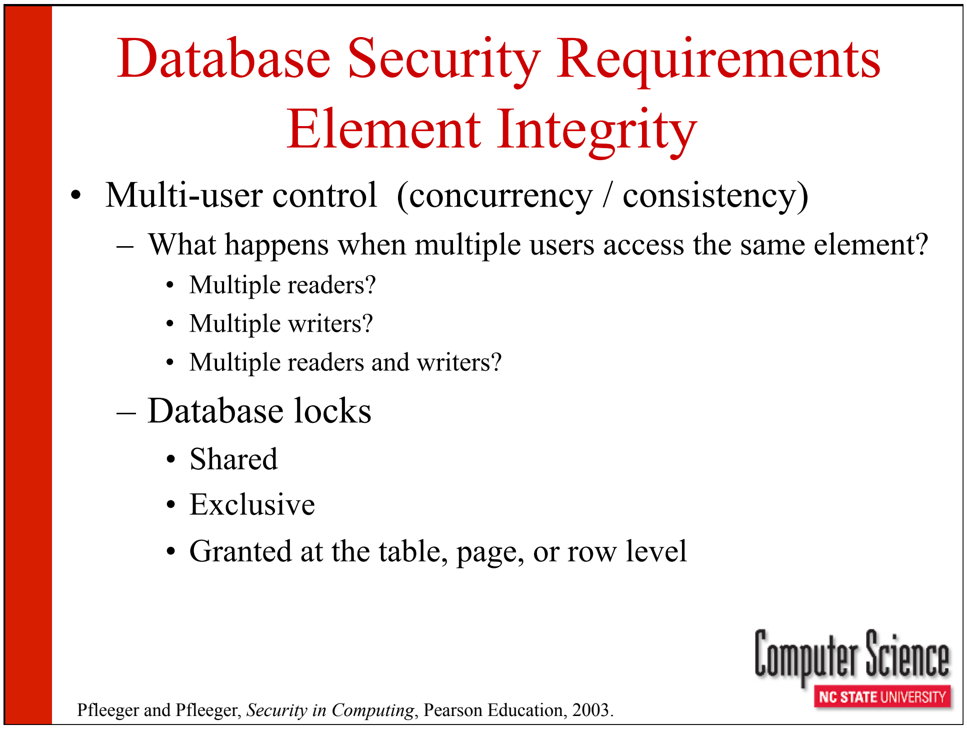
Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.



Database Security Requirements Element Integrity

- Data contained in each element are correct
- How?
  - Field checks
  - Referential checks
  - Triggers
  - Stored procedures

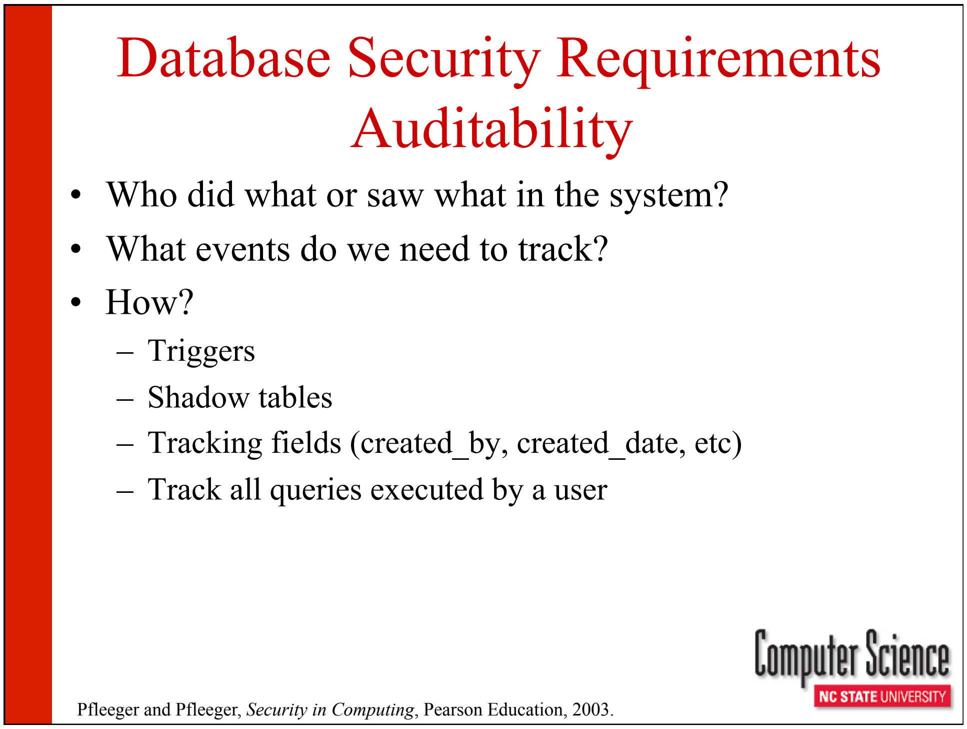
Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.



Database Security Requirements
Element Integrity

- Multi-user control (concurrency / consistency)
  - What happens when multiple users access the same element?
    - Multiple readers?
    - Multiple writers?
    - Multiple readers and writers?
  - Database locks
    - Shared
    - Exclusive
    - Granted at the table, page, or row level

Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.



Database Security Requirements
Auditability

- Who did what or saw what in the system?
- What events do we need to track?
- How?
  - Triggers
  - Shadow tables
  - Tracking fields (created_by, created_date, etc)
  - Track all queries executed by a user

Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.

Database Security Requirements Access Control

- Who can see what and do what in the system?
- How?
  - GRANT|REVOKE *privilege* on *object* to *user*|*role*
  - Use roles. Assign permissions there.
- Review default users and roles

Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.

Database Security Requirements User Authentication

- Every user must be identified
- Appropriate passwords
- Time of day checks

Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.

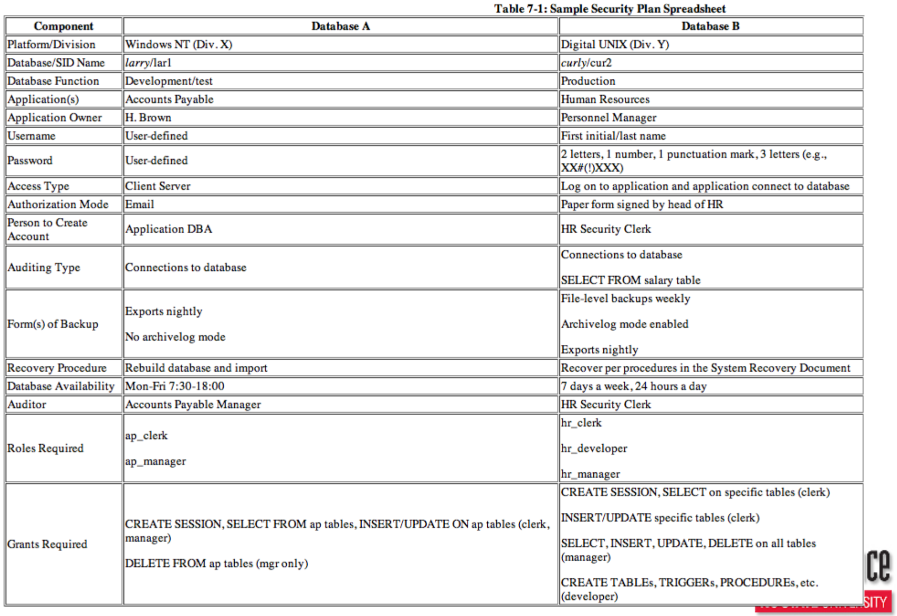
Database Security Requirements Availability

- Users must be able to access the database when needed
- What are the requirements of your system?
  - Time
  - Performance

Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.

Database Security Plan

Table 7-1: Sample Security Plan Spreadsheet



Component	Database A	Database B
Platform/Division	Windows NT (Div. X)	Digital UNIX (Div. Y)
Database/SID Name	*larry*/lar1	*curly*/cur2
Database Function	Development/test	Production
Application(s)	Accounts Payable	Human Resources
Application Owner	H. Brown	Personnel Manager
Username	User-defined	First initial/last name
Password	User-defined	2 letters, 1 number, 1 punctuation mark, 3 letters (e.g., XX#(!)XXX)
Access Type	Client Server	Log on to application and application connect to database
Authorization Mode	Email	Paper form signed by head of HR
Person to Create Account	Application DBA	HR Security Clerk
Auditing Type	Connections to database	Connections to database SELECT FROM salary table
Form(s) of Backup	Exports nightly No archivelog mode	File-level backups weekly Archivelog mode enabled Exports nightly
Recovery Procedure	Rebuild database and import	Recover per procedures in the System Recovery Document
Database Availability	Mon-Fri 7:30-18:00	7 days a week, 24 hours a day
Auditor	Accounts Payable Manager	HR Security Clerk
Roles Required	ap_clerk ap_manager	hr_clerk hr_developer hr_manager

Grants Required	CREATE SESSION, SELECT FROM ap tables, INSERT/UPDATE ON ap tables (clerk, manager)
	DELETE FROM ap tables (mgr only)
		CREATE SESSION, SELECT on specific tables (clerk)
		INSERT/UPDATE specific tables (clerk)
		SELECT, INSERT, UPDATE, DELETE on all tables (manager)
		CREATE TABLEs, TRIGGERs, PROCEDUREs, etc. (developer)

http://oreilly.com/catalog/orasec/chapter/ch07.html

Principle of Least Privilege

- Only give users the absolute minimum privileges to complete their job.
- If X service doesn't need access to all tables in Y database… then don't give it access to all tables.
- Do not give accounts privileges that aren't needed

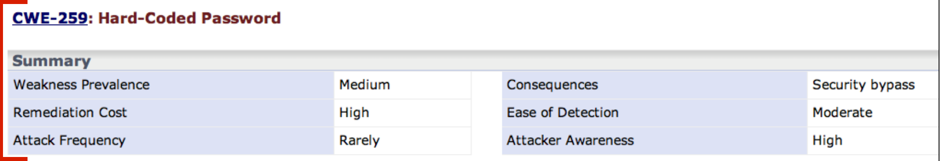
cio.uiowa.edu/ITsecurity/education/documents/DatabaseSecurity.ppt

Strong Passwords

- CWE 521: Weak Password Requirements
- Length
  - Each character you add to your password increases the protection
  - 8 or more characters are the minimum for a strong password; 14 characters or longer are ideal.
- Complexity
  - An ideal password combines both length and different types of symbols (alpha, numeric, mixed case)
- Does not contain user name
- Expiration
  - CWE 262: Not Using Password Aging
- No password reuse.

http://cwe.mitre.org/data/des/521.html and http://cwe.mitre.org/data/des/262.html

Hardcoded Password



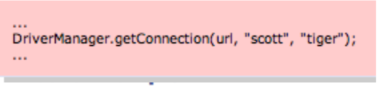
CWE-259: Hard-Coded Password

Summary			
Weakness Prevalence	Medium	Consequences	Security bypass
Remediation Cost	High	Ease of Detection	Moderate
Attack Frequency	Rarely	Attacker Awareness	High

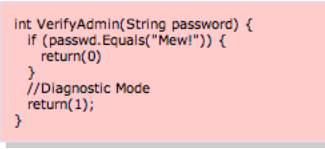
- Condition: The software contains a hard-coded password, which it uses for its own inbound authentication or for outbound communication to external components.
- Consequence: If the password is the same across all your software, then every customer becomes vulnerable if (rather, when) your password becomes known.

http://cwe.mitre.org/data/definitions/259.html

Hardcoded Password -2



```
...
DriverManager.getConnection(url, "scott", "tiger");
...
```

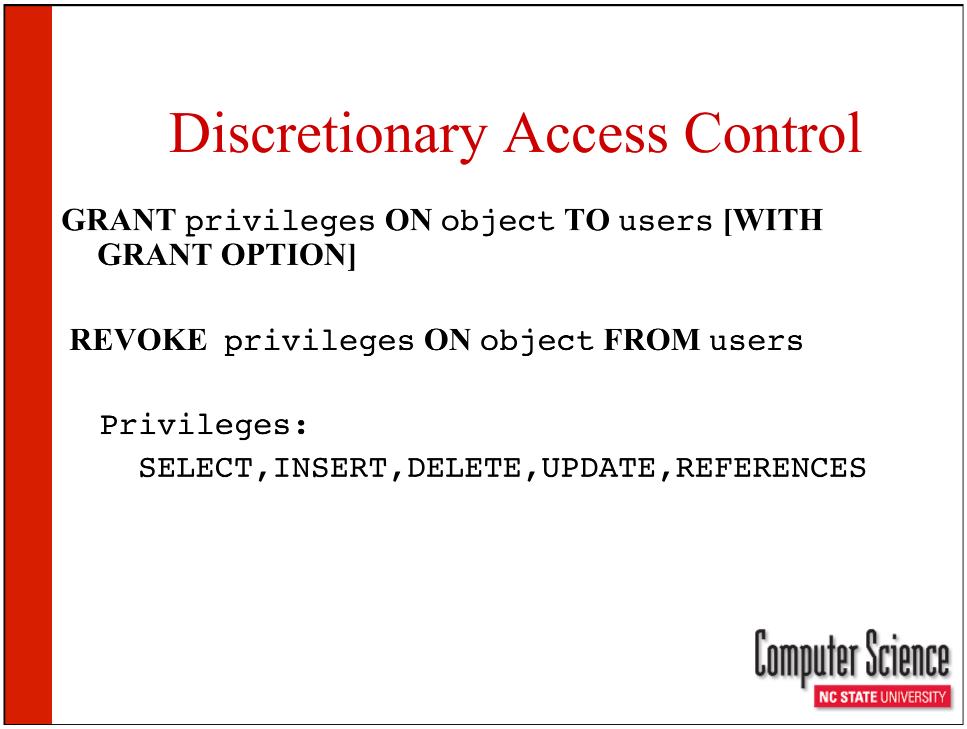


```
int VerifyAdmin(String password) {
  if (passwd.Equals("Mew!")) {
    return(0)
  }
  //Diagnostic Mode
  return(1);
}
```

Store passwords outside of the code in a strongly-protected, encrypted configuration file or database that is protected from access by all outsiders, including other local users on the same system.

Properly protect the key (CWE-320). If you cannot use encryption to protect the file, then make sure that the permissions are as restrictive as possible.

http://cwe.mitre.org/data/definitions/259.html



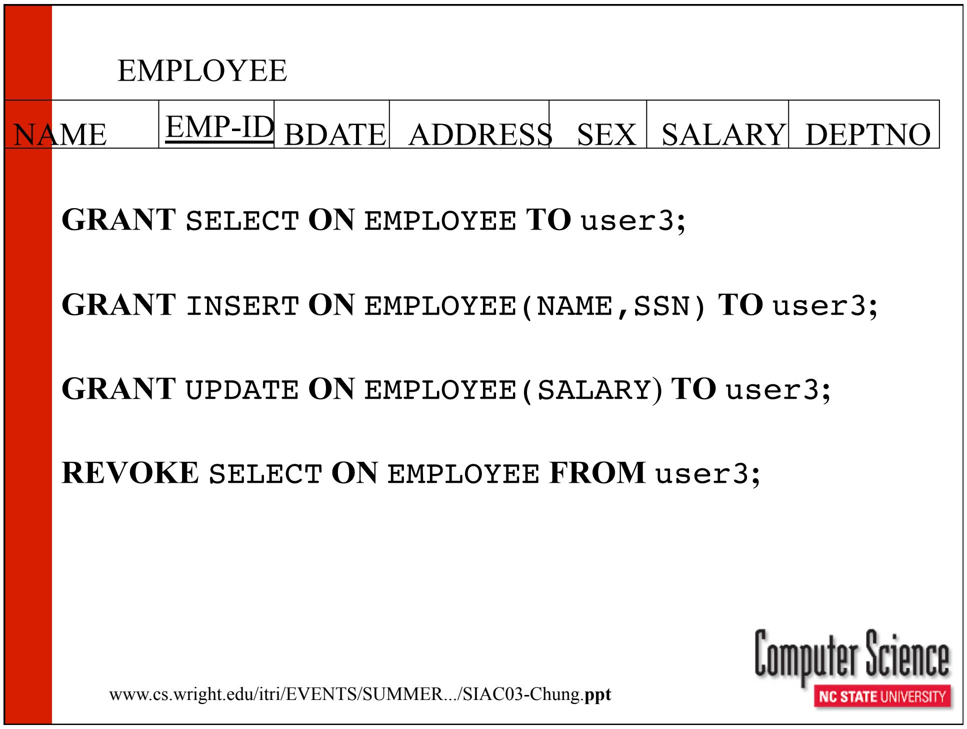
Discretionary Access Control

GRANT privileges **ON** object **TO** users **[WITH GRANT OPTION]**

REVOKE privileges **ON** object **FROM** users

```
Privileges:
   SELECT,INSERT,DELETE,UPDATE,REFERENCES
```

Computer Science
NC STATE UNIVERSITY



EMPLOYEE

NAME	EMP-ID	BDATE	ADDRESS	SEX	SALARY	DEPTNO

GRANT SELECT **ON** EMPLOYEE **TO** user3;

GRANT INSERT **ON** EMPLOYEE(NAME,SSN) **TO** user3;

GRANT UPDATE **ON** EMPLOYEE(SALARY) **TO** user3;

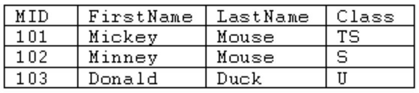
REVOKE SELECT **ON** EMPLOYEE **FROM** user3;

www.cs.wright.edu/itri/EVENTS/SUMMER.../SIAC03-Chung.**ppt**

Computer Science
NC STATE UNIVERSITY

Multi-level Access

- Users may be granted “top secret” “secret” “confidential” or “unclassified” access (decreasing access)
- Database records can be marked accordingly
  - User with TS access sees all three rows
  - User with S access sees Minney and Donald
  - User with U access sees Donal

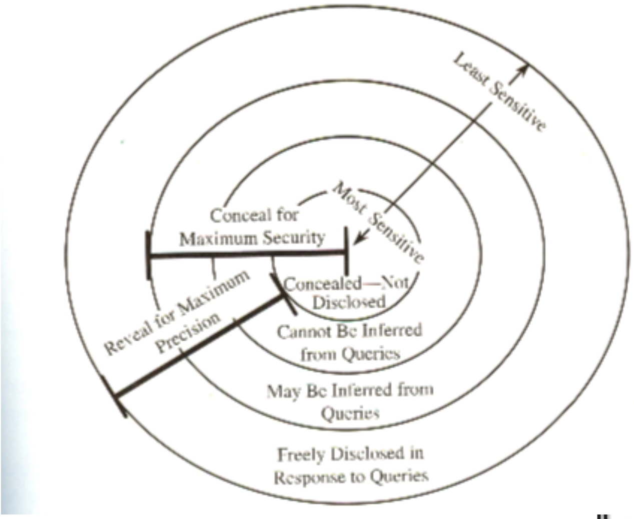


MID	FirstName	LastName	Class
101	Mickey	Mouse	TS
102	Minney	Mouse	S
103	Donald	Duck	U

Security versus Precision

- Precision – protect all sensitive data while revealing as much non-sensitive data as possible … such as in aggregated or anonymized form
  - List of grades for students with one or more felony charges
  - Average income of all men and all women

Security versus Precision



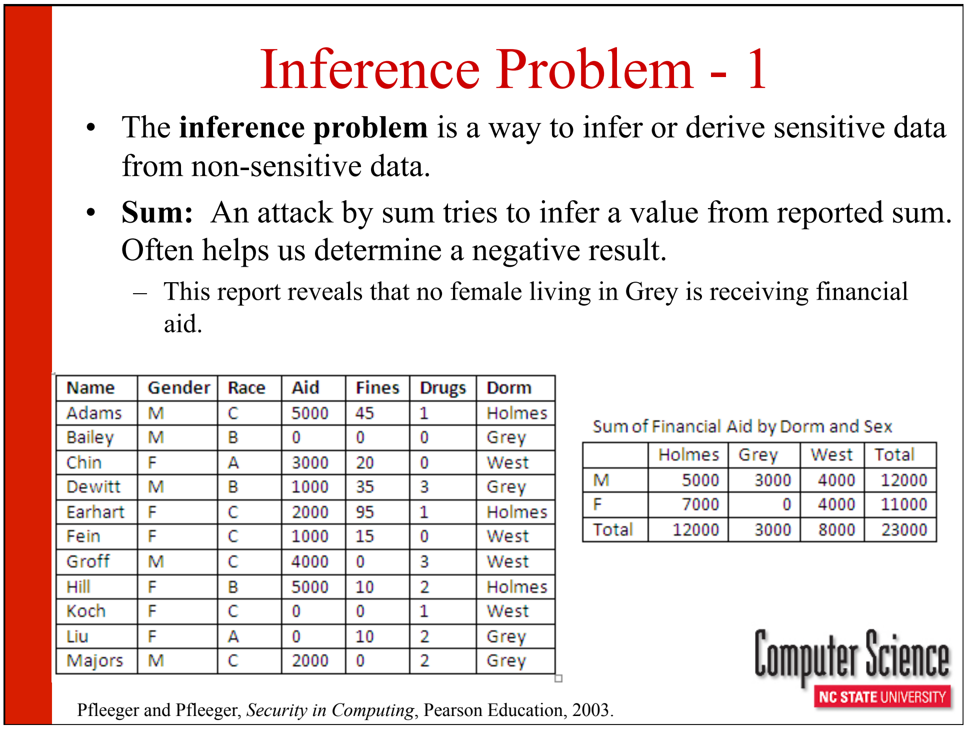
Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.

Computer Science
NC STATE UNIVERSITY

Internal Data Security

- Limit fields that can be queried
- Cleanse / Make data anonymous
- Encrypt data

Computer Science
NC STATE UNIVERSITY



Inference Problem - 1

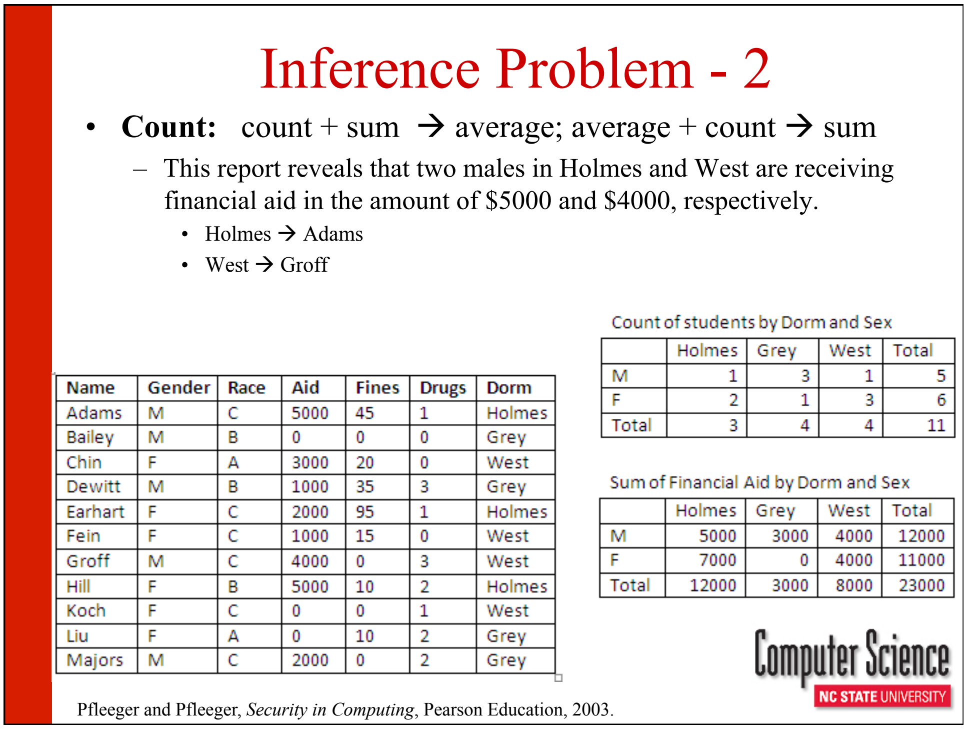
- The **inference problem** is a way to infer or derive sensitive data from non-sensitive data.
- **Sum:** An attack by sum tries to infer a value from reported sum. Often helps us determine a negative result.
  - This report reveals that no female living in Grey is receiving financial aid.

Name	Gender	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45	1	Holmes
Bailey	M	B	0	0	0	Grey
Chin	F	A	3000	20	0	West
Dewitt	M	B	1000	35	3	Grey
Earhart	F	C	2000	95	1	Holmes
Fein	F	C	1000	15	0	West
Groff	M	C	4000	0	3	West
Hill	F	B	5000	10	2	Holmes
Koch	F	C	0	0	1	West
Liu	F	A	0	10	2	Grey
Majors	M	C	2000	0	2	Grey

Sum of Financial Aid by Dorm and Sex

	Holmes	Grey	West	Total
M	5000	3000	4000	12000
F	7000	0	4000	11000
Total	12000	3000	8000	23000

Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.



Inference Problem - 2

- **Count:** count + sum → average; average + count → sum
  - This report reveals that two males in Holmes and West are receiving financial aid in the amount of $5000 and $4000, respectively.
    - Holmes → Adams
    - West → Groff

Count of students by Dorm and Sex

	Holmes	Grey	West	Total
M	1	3	1	5
F	2	1	3	6
Total	3	4	4	11

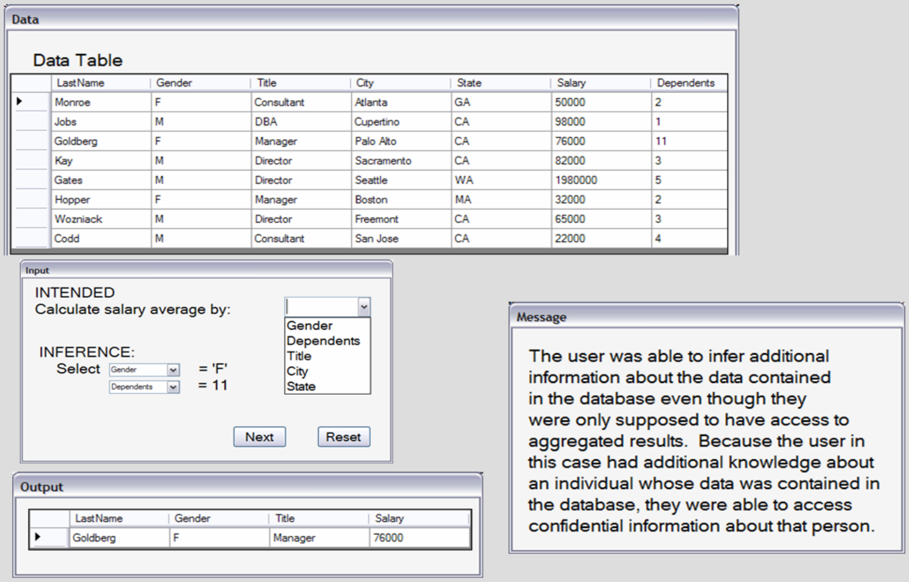
Name	Gender	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45	1	Holmes
Bailey	M	B	0	0	0	Grey
Chin	F	A	3000	20	0	West
Dewitt	M	B	1000	35	3	Grey
Earhart	F	C	2000	95	1	Holmes
Fein	F	C	1000	15	0	West
Groff	M	C	4000	0	3	West
Hill	F	B	5000	10	2	Holmes
Koch	F	C	0	0	1	West
Liu	F	A	0	10	2	Grey
Majors	M	C	2000	0	2	Grey

Sum of Financial Aid by Dorm and Sex

	Holmes	Grey	West	Total
M	5000	3000	4000	12000
F	7000	0	4000	11000
Total	12000	3000	8000	23000

Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.

Inference Problem - 3



Data Table

	LastName	Gender	Title	City	State	Salary	Dependents
▸	Monroe	F	Consultant	Atlanta	GA	50000	2
	Jobs	M	DBA	Cupertino	CA	98000	1
	Goldberg	F	Manager	Palo Alto	CA	76000	11
	Kay	M	Director	Sacramento	CA	82000	3
	Gates	M	Director	Seattle	WA	1980000	5
	Hopper	F	Manager	Boston	MA	32000	2
	Wozniack	M	Director	Freemont	CA	65000	3
	Codd	M	Consultant	San Jose	CA	22000	4

INTENDED
Calculate salary average by: [Gender / Dependents / Title / City / State]

INFERENCE:
Select Gender = 'F'
Dependents = 11

	LastName	Gender	Title	Salary
▸	Goldberg	F	Manager	76000

The user was able to infer additional information about the data contained in the database even though they were only supposed to have access to aggregated results. Because the user in this case had additional knowledge about an individual whose data was contained in the database, they were able to access confidential information about that person.

science.kennesaw.edu/~mguimara/8080/dbsecurity.ppt

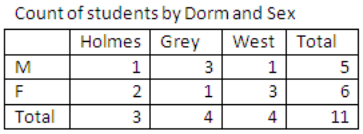
Controls for Statistical Inference Attacks

- Controls are applied to queries
  - Difficult to determine if query discloses sensitive data
- Controls are applied to individual items within the database (security vs. precision)
  - **Suppression:** sensitive data values are not provided; query is rejected without response
    - Many results suppressed; precision high
  - **Concealing:** answer provided is close to by not exactly the actual value
    - More results provided; precision low

Computer Science
NC STATE UNIVERSITY

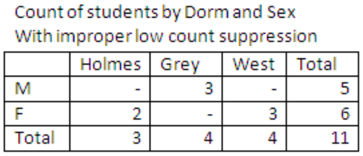
Limited Response Suppression

- The n-item k-percent rule eliminates certain low-frequency elements from being displayed
- When one cell is suppressed in a table with totals for rows and columns, must suppress at least one additional cell on the row and one on the column to provide some confusion.



Count of students by Dorm and Sex

	Holmes	Grey	West	Total
M	1	3	1	5
F	2	1	3	6
Total	3	4	4	11



Count of students by Dorm and Sex
With improper low count suppression

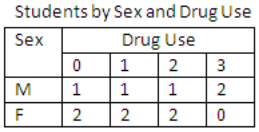
	Holmes	Grey	West	Total
M	-	3	-	5
F	2	-	3	6
Total	3	4	4	11

… Can only provide totals

Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.

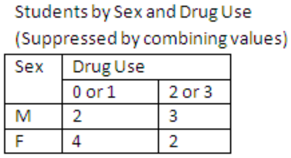
Other suppression and concealing

- Combine rows or columns to protect sensitive values



Students by Sex and Drug Use

Sex	Drug Use			
	0	1	2	3
M	1	1	1	2
F	2	2	2	0



Students by Sex and Drug Use
(Suppressed by combining values)

Sex	Drug Use	
	0 or 1	2 or 3
M	2	3
F	4	2

- Take a random sample (sample must be large enough to be valid)
  - Same sample set would be repeated for equivalent queries
- Query analysis
  - Query and its implications are analyzed
  - Can be difficult
  - Maintain query history for each user

- … no perfect solution to inference problem
- … recognizing the problem leads to being defensive

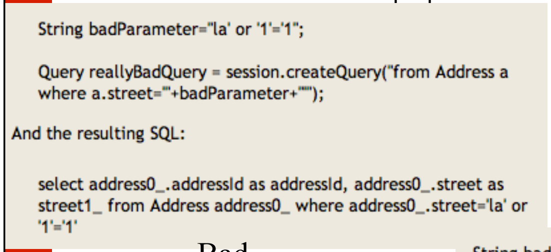
Pfleeger and Pfleeger, *Security in Computing*, Pearson Education, 2003.

Database Input Vulnerabilities (a.k.a. SQL Injection) Mitigation

- Validate Input
- Prepared Statements
- Stored Procedures
- Database frameworks
- QOUTENAME and REPLACE

Hibernate Framework

- Object/relational mapping
  - Cleanly connect Java objects and database tables
  - Requires adding a few Java classes
  - Uses Hibernate Query Language (HQL); similar to SQL
  - Uses prepared statements “under the covers”
    - So same issues as prepared statements



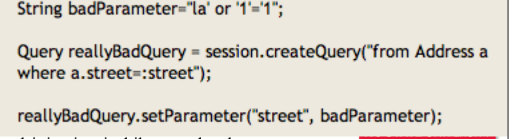
```
String badParameter="la' or '1'='1";

Query reallyBadQuery = session.createQuery("from Address a
where a.street='"+badParameter+"'");
```

And the resulting SQL:

```
select address0_.addressId as addressId, address0_.street as
street1_ from Address address0_ where address0_.street='la' or
'1'='1'
```

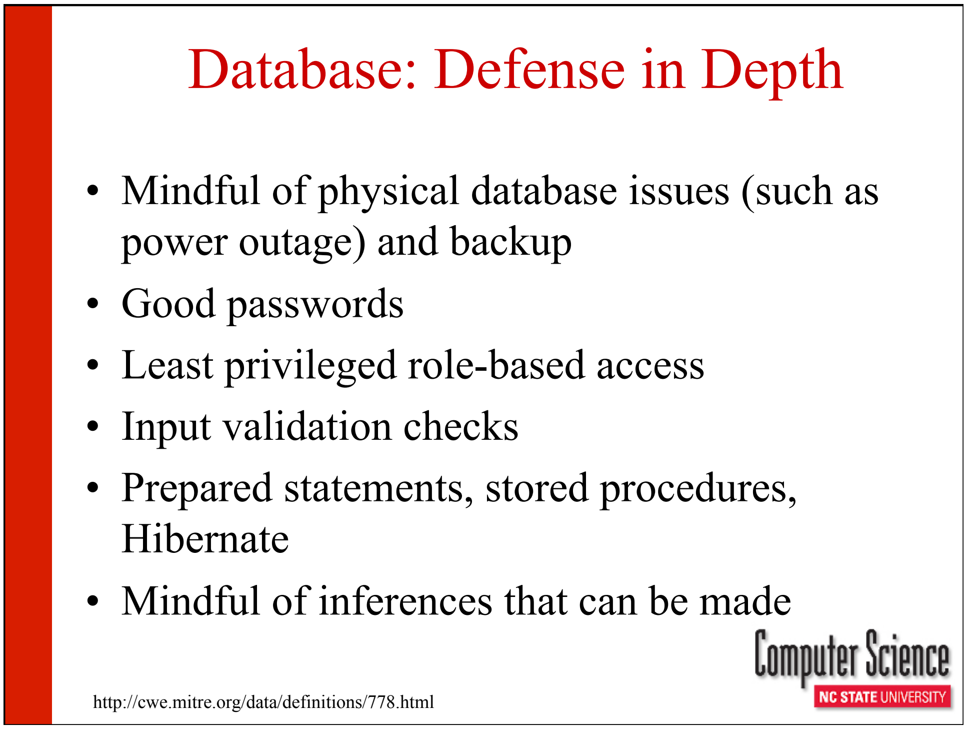
Bad

Good



```
String badParameter="la' or '1'='1";

Query reallyBadQuery = session.createQuery("from Address a
where a.street=:street");

reallyBadQuery.setParameter("street", badParameter);
```

http://blog.harpoontech.com/2008/10/how-to-avoid-sql-injection-in-hibernate.html



Database: Defense in Depth

- Mindful of physical database issues (such as power outage) and backup
- Good passwords
- Least privileged role-based access
- Input validation checks
- Prepared statements, stored procedures, Hibernate
- Mindful of inferences that can be made

http://cwe.mitre.org/data/definitions/778.html