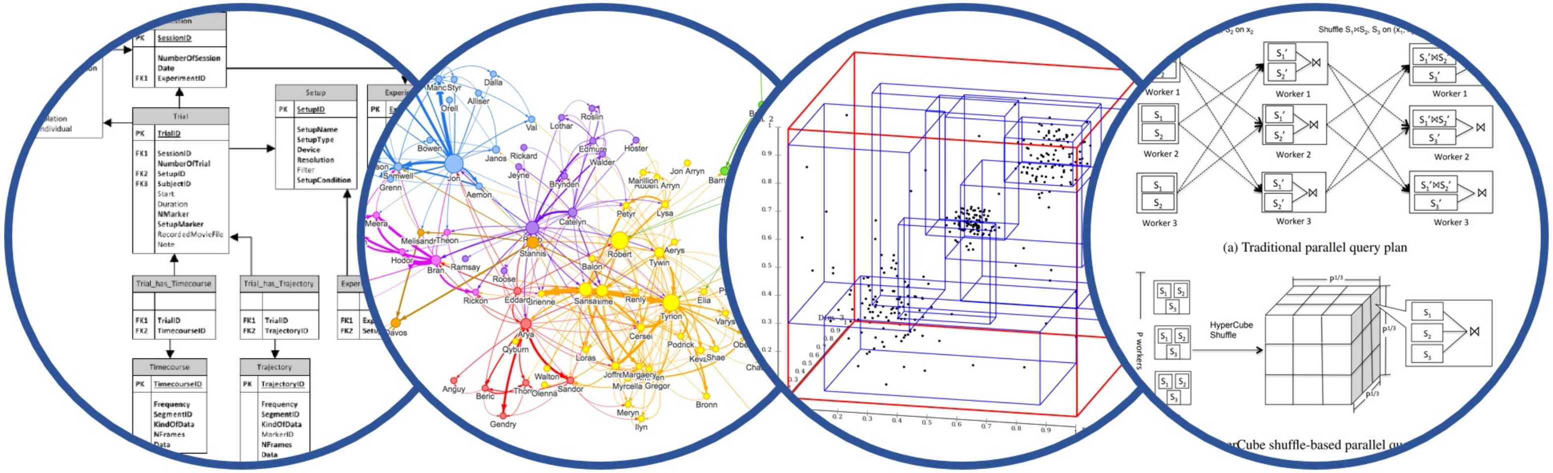


Introduction to Data Management

Practical Aspects

Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle

Announcements

- HW5 is due on Friday

- HW6 has two parts:
  - Part 1 due 5/17. No late days (for quick feedback)
  - Part 2 due 5/24. Much more work than part 1

Data Privacy Laws

Law

Some data is protected by law:

- HIPPA
- GDPR
- FERPA

HIPPA

Health Information Portability and Accountability Act

- Mandatory for healthcare and health insurance institutions
- Privacy Rule to protect Protected Health Information
- Security Rule to ensure administrative, physical, and technical safeguards

GDPR

General Data Protection Regulation (GDPR)

- European Union
- Corporate disclosure of what user data is stored
- Only recently implemented (a few years ago)

FERPA

Family Education Rights and Privacy Act

- Mandatory for education institutions
  - Requires written consent to disclose academic info
  - Allows the release of directory information
- Allows institutions to disclose “directory information” without consent (institution policies can be stronger)
  - Name
  - Email
  - Photographs
  - Phone Number

Privacy Leaks via Linking

Anonymity

- Common practice for making a dataset private: remove Personal Identifiable Information (PII)
- But by linking data from distinct datasets one can reveal private information
- In her PhD thesis* (2001) Latanya Sweeney described a famous example

* https://dspace.mit.edu/handle/1721.1/8589

Latanya Sweeney's Finding

- Massachusetts: GIG* is responsible for health insurance of state emps;

*Group Insurance Commission

Latanya Sweeney’s Finding

- Massachusetts: GIG* is responsible for health insurance of state emps; public data

```
GIC(zip, dob, sex,
 diagnosis, procedure,...)
```

*Group Insurance Commission

Latanya Sweeney’s Finding

- Massachusetts: GIG* is responsible for health insurance of state emps; public data

```
GIC(zip, dob, sex,
 diagnosis, procedure,...)
```

- Sweeney paid $20 and bought voter registration list for Cambridge, MA

```
VOTER(name, party, ...,
      zip, dob, sex)
```

*Group Insurance Commission

Latanya Sweeney’s Finding

- Massachusetts: GIG* is responsible for health insurance of state emps; public data

GIC(**zip**, **dob**, **sex**,
diagnosis, procedure,...)

- Sweeney paid $20 and bought voter registration list for Cambridge, MA

VOTER(**name**, party, ...,
zip, **dob**, **sex**)

- William Weld** lived in Cambridge: in VOTER

*Group Insurance Commission
**former governor

Latanya Sweeney’s Finding

- Massachusetts: GIG* is responsible for health insurance of state emps; public data

GIC(**zip**, **dob**, **sex**, diagnosis, procedure,...)

- Sweeney paid $20 and bought voter registration list for Cambridge, MA

VOTER(**name**, party, ..., **zip**, **dob**, **sex**)

- William Weld** lived in Cambridge: in VOTER
- 6 people had same **dob**

*Group Insurance Commission
**former governor

Latanya Sweeney’s Finding

- Massachusetts: GIG* is responsible for health insurance of state emps; public data

GIC(**zip**, **dob**, **sex**,
diagnosis, procedure,...)

- Sweeney paid $20 and bought voter registration list for Cambridge, MA

VOTER(**name**, party, ...,
zip, **dob**, **sex**)

- William Weld** lived in Cambridge: in VOTER
- 6 people had same **dob**
- 3 had also **sex**=‘M’

*Group Insurance Commission
**former governor

Latanya Sweeney’s Finding

- Massachusetts: GIG* is responsible for health insurance of state emps; public data

GIC(**zip**, **dob**, **sex**,
diagnosis, procedure,...)

- Sweeney paid $20 and bought voter registration list for Cambridge, MA

VOTER(**name**, party, ...,
zip, **dob**, **sex**)

- William Weld** lived in Cambridge: in VOTER
- 6 people had same **dob**
- 3 had also **sex**=‘M’
- Weld only one in that **zip**

*Group Insurance Commission
**former governor

Latanya Sweeney’s Finding

- Massachusetts: GIG* is responsible for health insurance of state emps; public data

GIC(**zip**, **dob**, **sex**,
diagnosis, procedure,...)

- Sweeney paid $20 and bought voter registration list for Cambridge, MA

VOTER(**name**, party, ...,
zip, **dob**, **sex**)

- William Weld** lived in Cambridge: in VOTER
- 6 people had same **dob**
- 3 had also **sex**=‘M’
- Weld only one in that **zip**

Sweeney learned Weld’s medical records !

*Group Insurance Commission
**former governor

Latanya Sweeney’s Finding

- The best common practice is still to remove PII
- Law specifies which attributes are considered PII

Privacy Leaks via Aggregates

Implicit Disclosure

FERPA says:

- These might be public*
  - Name
  - Email
  - Photographs
  - Phone Number

- Grades are private;
- Grade averages from larger groups are OK

* Each university may impose further restrictions

Which Queries Should be Permitted?

Student(sid, name, email)
Takes(sid, cid, grade)
Course(cid, …)

Which Queries Should be Permitted?

Student(sid, name, email)
Takes(sid, cid, grade)
Course(cid, …)

Alice’s grade in cse414:

```
SELECT T.grade
FROM Students S, Takes T
WHERE S.sid = T.cid
    and T.cid = ‘cse414’
  and S.name = ‘Alice’
```

Which Queries Should be Permitted?

Student(sid, name, email)
Takes(sid, cid, grade)
Course(cid, …)

Alice’s grade in cse414:

No

```
SELECT T.grade
FROM Students S, Takes T
WHERE S.sid = T.cid
  and T.cid = ‘cse414’
 and S.name = ‘Alice’
```

Which Queries Should be Permitted?

Student(sid, name, email)
Takes(sid, cid, grade)
Course(cid, …)

No

Alice’s grade in cse414:

```
SELECT T.grade
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
  and S.name = ‘Alice’
```

Average grade
of students in 414

```
SELECT avg(T.grade)
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
```

Which Queries Should be Permitted?

Student(sid, name, email)
Takes(sid, cid, grade)
Course(cid, …)

Alice’s grade in cse414:

No

```
SELECT T.grade
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
  and S.name = ‘Alice’
```

Maybe?

Average grade
of students in 414

```
SELECT avg(T.grade)
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
```

Which Queries Should be Permitted?

Student(sid, name, email)
Takes(sid, cid, grade)
Course(cid, …)

Alice’s grade in cse414:

No

```
SELECT T.grade
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
  and S.name = ‘Alice’
```

Average grade
of students in 414

Maybe?

```
SELECT avg(T.grade)
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
```

Average grade
of students in 414
other than Alice!

```
SELECT avg(T.grade)
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
   and S.name != ‘Alice’
```

Which Queries Should be Permitted?

Student(sid, name, email)
Takes(sid, cid, grade)
Course(cid, …)

Alice’s grade in cse414:

No

```
SELECT T.grade
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
  and S.name = ‘Alice’
```

Maybe?

Average grade
of students in 414

```
SELECT avg(T.grade)
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
```

No

Average grade
of students in 414
other than Alice!

```
SELECT avg(T.grade)
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
   and S.name != ‘Alice’
```

Which Queries Should be Permitted?

Student(sid, name, email)
Takes(sid, cid, grade)
Course(cid, …)

Alice’s grade in cse414:

No

```
SELECT T.grade
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
  and S.name = ‘Alice’
```

~~Maybe?~~

Average grade
of students in 414

No

```
SELECT avg(T.grade)
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
```

Average grade
of students in 414
other than Alice!

No

```
SELECT avg(T.grade)
FROM Students S, Takes T
WHERE S.sid = T.cid
   and T.cid = ‘cse414’
   and S.name != ‘Alice’
```

Discussion

Make sure you understand how the privacy leak happened. Example:

- Sum of all grades = S
- Alice’s grade = A
- 100 students in classs

- Avg grade in class: $S/100 = 3.49$
- Avg grade w/o Alice: $(S-A)/99 = 3.5$

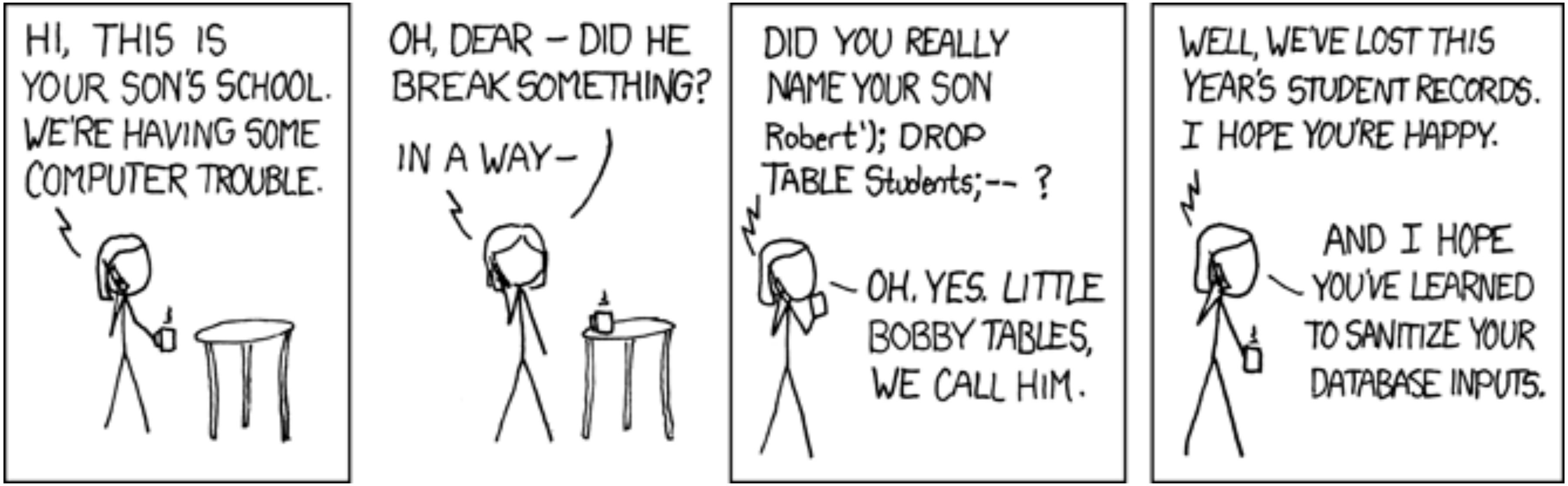
- Solve for A: $A = 2.5$

Today's Solutions

- Bucketize data and release only information on large groups
- Add noise: differential privacy

SQL Injection

SQL Injection

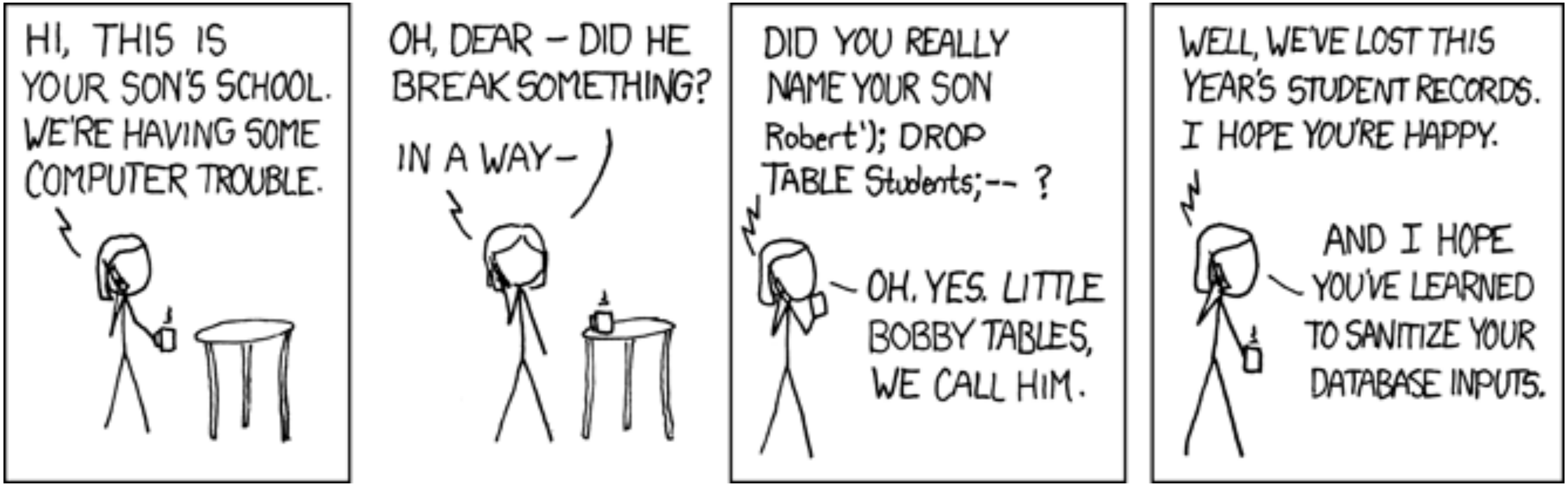

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

SQL Injection

- In the application, a SQL query is a string
- Part of that string is input by the user
- A malicious user can enter a string that changes the SQL query

Demo

SQL Injection


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

SQL Injection

Considered a “solved” problem

- Parameterize queries using ’?’
- Use ‘prepared’ statements

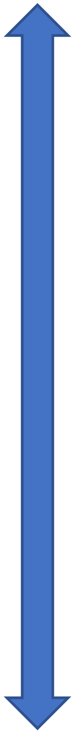
Storing Passwords

Storing Passwords

- Passwords are special
  - High potential for additional security compromises
  - Only operation that should be done is equality comparison

Storing Passwords

(bobtheninja246, password)



If you do this, Ted Codd will roll in his grave.

Username	Password
bobtheninja246	password
xXxDragonSlayerxXx	password
420_E-Sports_Masta	qwertyuiop

Storing Passwords

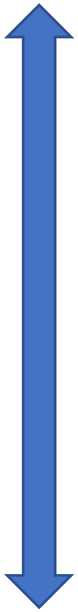
- Quick overview of hashing
  - Hash(input) → hash value
    - Hash function takes input and generates “scrambled” output, that is always equal for the same input
  - Hashing is deterministic
  - Ideally hashing is noninvertible
    - Secure hash functions make it impossible to derive the input value from the hash value
  - Ideally hash values are uniformly spread out
    - Useful for hash tables!

Storing Passwords

Hash it!

(bobtheninja246, hash(password))

(bobtheninja246, FCgJFI9ryz)



Username	Hash
bobtheninja246	FCgJFI9ryz
xXxDragonSlayerxXx	FCgJFI9ryz
420_E-Sports_Masta	p8mel6usIF

Storing Passwords

Hash it!

(bobtheninja246, hash(password))

(bobtheninja246, FCgJFI9ryz)

Issues/pitfalls:

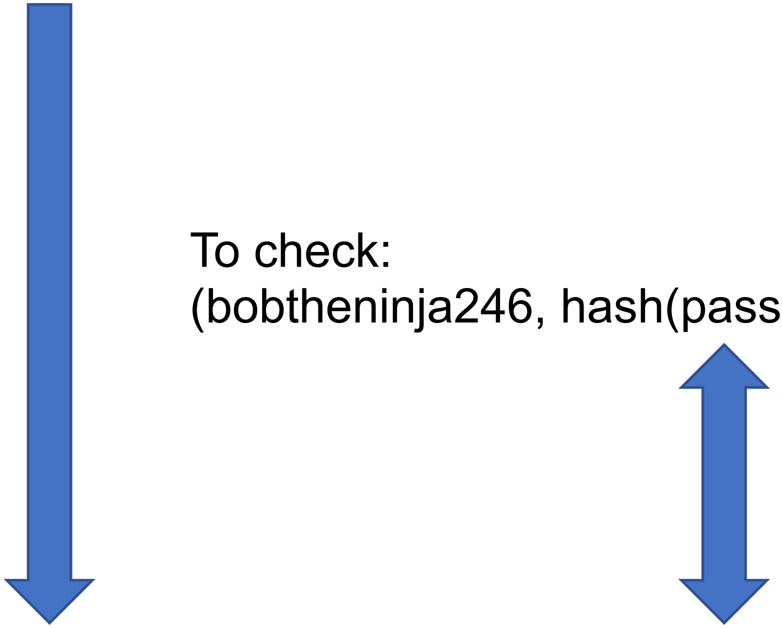
- Hashing functions have precomputed “rainbow tables”
- Patterns can occur for the same passwords

Username	Hash
bobtheninja246	FCgJFI9ryz
xXxDragonSlayerxXx	FCgJFI9ryz
420_E-Sports_Masta	p8mel6usIF

Storing Passwords

Salt it and hash it!

(bobtheninja246, hash(password * random salt), random salt)



To check:
(bobtheninja246, hash(password * stored salt))

Username	Hash	Salt
bobtheninja246	HHxrd5o7Cn	WUKhhIFBLc
xXxDragonSlayerxXx	7rYFQIowpW	mq5rFL6JzF
420_E-Sports_Masta	cQF4DdSFfn	S8e0zpATNR

Discussion

- These are just the fundamentals: companies outsource password management because it can get very complicated.
- In HW6 you are asked to do simple password management