Ethics and Privacy

Road Map

- so far – mechanisms that protect data in the database
- now – the decisions about personal data the end up in your schemas, access rules, and retention policies
  - security vs privacy
  - PII and anonymity
  - privacy compliance by design
  - ownership and custodianship
  - unintended uses of data

From Protection to Responsibility

- we have mechanisms to protect data
  - transactions and isolation keep it correct
  - validation keeps bad data out
  - privileges, views, and placeholders keep out those who shouldn't get in
  - auditing records what happened
- questions the mechanisms don't address
  - should we even store the data at all?
  - who should see the data?
  - how long should we keep the data?
  - what do we owe the people it describes?

Security Isn't Enough

- a librarian browses a celebrity's loan history out of curiosity
- all the checks passed
  - authenticated and authorized – legitimate account with legitimate access
  - no injection, no anomaly – legitimate queries run
- but this shouldn't have happened
  - library patrons don't consider their loan history to be public
  - the confidentiality of library records is protected by law
- need limits on how authorized people can use data, not just who can access it
  - the harm came from an authorized insider, not an attacker
  - need-to-know access – narrower than "librarians see all loans"
  - purpose limits – loan records are for lending, not curiosity
  - monitoring – requires auditing of reads, not just changes

Security vs Privacy

- *security* is “can this person get to the data”?
  - protecting data from unauthorized access, alteration, or loss
  - for a database
    - who can access which tables
    - can the database survive attacks and failures
- *privacy* is “should this data be used this way, or even kept in the first place?”
  - appropriate collection, use, and disclosure of personal data – including by those authorized to see it
  - for a database
    - should this column exist?
    - who should see it and for what purpose?
    - how long should it be kept?
- a database can be completely secure and still fail on privacy

Privacy Matters

- people are harmed when personal data is misused
  - discrimination (insurance, jobs, housing)
  - manipulation (targeted persuasion)
  - chilling effects – people avoid reading or searching for what they think is being watched
- people have expectations about data collection and use
  - consenting to one purpose doesn’t mean consenting to another
  - meeting those expectations is what earns trust for organizations
- many laws require it
  - keeping data longer than needed, using it for a purpose people weren’t told about, or ignoring a request to delete it can be legal failures
    - obligation may be set directly law (e.g. GDPR) or the law may enforce published promises (e.g. FTC’s deceptive practices)
  - privacy obligations apply even when no attacker is involved
- database professionals make it happen
  - decisions about what’s collected, who sees it, how long it is kept are carried out in the database

Issues of Scale

Large data collections change the risk –

- ordinary failures have bigger impacts
  - breaches – one incident exposes millions
  - errors – one bad rule or data load affects many records
- the collection reveals more than its parts
  - aggregation – many harmless facts together reveal a lot
  - linking – more outside data to match against
  - surveillance – a record of what many people do
  - value – the collection becomes an asset others want

Issues of Scale

issue	examples of resulting harms
aggregation	social: profiles reveal health, beliefs, or habits people never disclosed, enabling discrimination or manipulation
linking	social: "anonymous" data is re-identified, exposing the people in it
	legal: released data may still count as personal data under privacy law, with the obligations that come with it
breaches	social: identity theft and fraud for many people at once
	legal: breach notification, fines, and lawsuits
errors	social: many people wrongly denied, billed, or flagged
	legal: liability when wrong data drives decisions about people (e.g. credit reports)
surveillance	social: chilling effects: people change what they read, search, or say
	legal: records can be demanded by law enforcement
value	social: data sold or transferred beyond what people expected
	legal: disputes over whether a transfer needs people's consent

Personally Identifiable Information (PII)

- *personally identifiable information (PII)* is data that identifies a person, alone or in aggregate
  - *direct identifiers* identify someone on their own
    - e.g. name, email, library card number, government ID number, …
  - *quasi-identifiers* are harmless on their own but identify in aggregate
    - e.g. zip code, birth date, gender, …
  - *sensitive attributes* are not identifying but can be harmful if linked to a person
    - e.g. loan history, health, religion, political views, …

Personally Identifiable Information

- label direct identifiers, quasi-identifiers, sensitive attributes
- if you deleted card_no, name, and email, could someone still be identified? what columns would you use?

card_no	name	email	birth_date	postal_code	gender	branch	favorite_genre	loans_this_year
			2001-03-14	99258	F	Downtown	mystery	31
			1987-11-02	99201	M	Northside	history	4

Protecting PII in a Database

- don't collect what you don't need
- know which columns hold it
- expose it only through views and grants that need it
- keep it out of logs, error messages, and test copies of the database

Protecting What the Database Lets Out

- *anonymous data* can't reasonably be linked to a person by anyone
  - e.g. aggregate counts over large groups
- *pseudonymous data* has ids replaced by consistent codes
  - e.g. replace borrower card number 123 with code BX83E everywhere it appears
  - records still link to each other so relationships are preserved
    - e.g. can still see that one person borrowed 10 books but not who that person is
  - mapping table is PII – whoever has the mapping can reverse it
  - linked rows are what make data identifying
    - the more rows tied to the same pseudonym, the more likely that combination is unique

Protecting What the Database Lets Out

- *de-identified data* has ids removed or generalized
  - e.g. export member list with names, usernames, and emails removed and birth date changed to just birth year
  - joining tables before de-identifying preserves per-row context
    - e.g. each loan keeps info about its borrower
  - "these rows are about the same person" is lost
    - e.g. can no longer tell that two loans belong to the same borrower
  - reduces the risk of identification but doesn't eliminate it
    - links can partly survive e.g. loans with an unusual combination of borrower info probably belong to the same person
    - preserving links through pseuodnyms increases risk

Protecting What the Database Lets Out

- removing names isn't sufficient to make data anonymous
  - quasi-identifiers can be linked to other data that has names (voter lists, social media, …)
  - 1997 Weld example
- design responses
  - export only what the purpose needs
  - generalize quasi-identifiers (birth year, 3-digit zip, …)
  - replace ids with pseudonyms
  - report counts only for groups above a minimum size

Privacy Compliance By Design

- privacy laws differ in detail but most impose similar kinds of obligations
- build the database from the start to meet those obligations
  - retrofitting is expensive

obligation	what it means for the database
a stated purpose for collecting	every column has a reason – no columns "just in case"
minimization	collect and expose only what's needed: views, least privilege
retention limits	deletion schedules, including for backups and logs
individual rights: access, correction, erasure	ability to find, export, and delete one person's data across every table
breach notification	auditing, so you know what was taken and who it belongs to
reach follows the people, not the organization	where your users are matters, not just where your servers are

Privacy Laws

- GDPR – General Data Protection Regulation (EU, 2018)
  - covers all personal data of people in the EU, wherever the organization is
  - gives people rights to access, correct, and erase their data
  - requires prompt reporting of breaches to the authorities
- HIPAA – Health Insurance Portability and Accountability Act (US, 1996)
  - covers health information held by health care provides, health plans, and their business partners
  - requires using and disclosing only the minimum necessary
  - applies based on who holds the data, not how sensitive it is
- FERPA – Family Educational Rights and Privacy Act (US, 1974)
  - covers educational records held by schools and colleges that receive US Dept of Education funding
  - gives students rights to see and ask to correct their records
  - requires student's consent for most disclosures
- as a database professional, need to recognize when a law likely applies and what kinds of obligations it brings
  - legal experts interpret the details

Reasonable Care

- laws often require “appropriate” or “reasonable” measures
- *due care* is taking the precautions a reasonable professional would take given the risks
  - e.g. least privilege, placeholders, encrypted backups, applying security updates, collecting only what is needed
- *due diligence* is finding out what the risks are and checking that the precautions are adequate and still working
  - e.g. assessing a new system’s data before building it, reviewing who has which grants, testing that backups restore, checking how a vendor will protect data you share
- legally, failing to take reasonable care is negligence
  - regulators can act without a breach
  - after a breach it decides liability
- show you’ve take reasonable care with records
  - why a column exists, why data is kept for a given period, who approved which access, …

Library Example

The library promises it doesn’t keep members’ reading history.

```
CREATE TABLE LOANS (
    loan_id        INT  NOT NULL AUTO_INCREMENT,
    book_id        INT  NOT NULL,
    member_id      INT  NOT NULL,
    checked_out_on DATE NOT NULL,
    due_on         DATE NOT NULL,
    returned_on    DATE NULL,
    PRIMARY KEY (loan_id),
    FOREIGN KEY (book_id)   REFERENCES BOOKS (book_id)     ON DELETE RESTRICT,
    FOREIGN KEY (member_id) REFERENCES MEMBERS (member_id) ON DELETE CASCADE
);
```

  - every returned loan stays forever, along with its member_id
- options
  - clear member_id once a loan is returned
  - delete loans after a retention period
- costs and wrinkles
  - fines and disputes need recent history
  - ON DELETE RESTRICT on books blocks deleting any book that has ever been loaned – if loan history is cleared, books not borrowed recently can be deleted
  - the audit trail records member ids