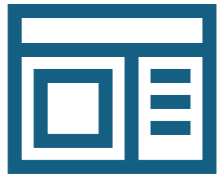
CS 161 Software Projects

Instructor: Dominic Abucejo

Spring 2025

MH 225

SJSU SAN JOSÉ STATE UNIVERSITY



Agenda

- HW #4
- New topics (Databases)
- Team/Group Meetings (~25 minutes)

Back-End Data Repository Issues

- Redundancy and inconsistency
  - Multiple copies of data (good for backup)
  - Different versions that don't match (bad)
  - Consistent updates and deletions
- Access
  - How to access data
  - Timeliness
- Disparity
  - Data stored in multiple and scattered locations
  - Data stored in different formats and media

Back-End

Back-End Data Repository Issues, *cont'd*

- Concurrency
  - Handle multiple clients accessing and updating the data simultaneously.
- Security
  - Prevent unauthorized accesses and updates.
- Integrity
  - Data values must meet constraints .
    - Example: Minimum and maximum value ranges
  - Data values must agree with each other.
    - Example: Medical records for a male patient should not include pregnancy data.

Data Modeling

- A data model is a diagram that shows what data an application works with and how the data is used.
  - Model data that will be persisted (written to and read from a data repository).
  - In the Model-View-Controller architecture, generally only the model objects are persisted.
- For databases, objects are called entities.
  - Model entities and their relationships to each other.
  - Similar to classes and their relationships.

Data Modeling, *cont'd*

- Conceptual data model
  - A high-level user-oriented description of the data.
    - entities
    - relationships among the entities
    - who will use the data (access control)
    - how it will be used (use cases)
- Logical data model
  - A diagram showing names, attributes, keys, and relations.

Data Modeling, *cont'd*

- Physical data model
  - A configuration specification or a script to build the repository.
  - Used by repository developers and maintainers.

Conceptual Data Model Example

- Student, teacher, and class entities and their attributes.
  - Student
    - id
    - name
    - which teachers
  - Teacher
    - id
    - name
    - which classes taught
  - Class
    - class code
    - subject name
    - class room number

Conceptual Data Model Example, *cont'd*

- Sample queries (derived from use cases)
  - What teachers does this student have?
  - What classes does this teacher teach?
  - Who is the teacher of this class?
  - Which students are in this class?
  - Which students are in each of the classes taught by this teacher?

The Relational Data Model

- Data element: values that are stored in the repository (i.e., the database)
  - Values are typed.
  - A value can be null.
- Entity: a group of data elements that together are meaningful for a person or an application
  - Similar to objects.
  - Each data element is the value of an attribute of the entity.

The Relational Data Model, *cont'd*

- Table: a conceptual two-dimensional structure that contains entities of a particular type.
  - AKA relation
- Each row (AKA record) contains the attribute values of one entity.
- Each column (AKA field) holds an attribute value.
- Table ⇔ relation
- Row ⇔ entity
- Rows and columns ⇔ records and fields

Logical Data Model

- Initial version

Each table has a primary key (PK) field whose value in each record uniquely identifies that record.

Student

Id	Name	Teacher_id_1	Teacher_id_2	Teacher_id_3
1001	Doe, John	7003	7012	7008
1005	Novak, Tim	7012	7008	*null*
1009	Klein, Leslie	*null*	*null*	*null*
1014	Jane, Mary	7051	*null*	*null*
1021	Smith, Kim	7003	7012	7051

PK

Teacher

Id	Name	Class_code	Subject	Room
7003	Rogers, Tom	926	Java programming	101
7008	Thompson, Art	908	Data structures	114
7012	Lane, John	951	Software engineering	210
7012	Lane, John	974	Operating systems	109
7051	Flynn, Mabel	931	Compilers	222

PK

- Student
  - id
  - name
  - which teachers
- Teacher
  - id
  - name
  - which classes taught
- Class
  - class code
  - subject name
  - class room number

John Lane teaches two classes.

Normalization

- Relational tables need to be normalized.
  - Improve the stability of the model.
    - More resilient to change.
  - Faster record insertions and updates.
  - Improve data quality.
- There are six normal forms, but we will only consider the first two.
  - Each normal form includes the lower normal forms.
    - Example: A database in second normal form is also in first normal form.

First Normal Form (1NF)

- Separate multi-valued data elements.
  - Break the name fields into last name and first name fields.

Student

Id	Name		Teacher_id_1		Teacher_id_2		Teacher_id_3		_3
1001	Doe	John	7003	7003	7012	7012	7008	7008	
1005	Novak	Tim	7012	7012	7008	7008	null	null	
1009	Klein	Leslie	null	null	null	null	null	null	
1014	Jane	Mary	7051	7051	null	null	null	null	
1021	Smith	Kim	7003	7003	7012	7012	7051	7051	

Teacher

Id	Name		Class_code		Subject	Room	
7003	Rogers, Tom	Tom	926	926	Java programming	101	101
7008	Thompson, Art		908	908	Data structures	114	114
7012	Lane, John	John	951	951	Software engineering	210	210
7012	Lane, John	John	974	974	Operating systems	109	109
7051	Flynn, Mabel	Mabel	931	931	Compilers	222	222

Database normalization is a database design principle for organizing data in an organized and consistent way

First Normal Form, *cont'd*

- Move repeating data elements to a new table.

Student

Id	Last	First	Teacher_id_1	Teacher_id_2	Teacher_id_3
1001	Doe	John	7003	7012	7008
1005	Novak	Tim	7012	7008	*null*
1009	Klein	Leslie	*null*	*null*	*null*
1014	Jane	Mary	7051	*null*	*null*
1021	Smith	Kim	7003	7012	7051

Linking table

Student_Teacher

Student_id	Teacher_id
1001	7003
1001	7012
1001	7008
1005	7012
1005	7008
1014	7051
1021	7003
1021	7012
1021	7051

Teacher

Id	Last	First	Class_code	Subject	Room
7003	Rogers	Tom	926	Java programming	101
7008	Thompson	Art	908	Data structures	114
7012	Lane	John	951	Software engineering	210
7012	Lane	John	974	Operating systems	109
7051	Flynn	Mabel	931	Compilers	222

Problem!

- Suppose Prof. Lane decides he doesn't want to teach Operating Systems anymore and we delete that row.

Teacher

Id	Last	First	Class_code	Subject	Room
7003	Rogers	Tom	926	Java programming	101
7008	Thompson	Art	908	Data structures	114
7012	Lane	John	951	Software engineering	210
7012	Lane	John	974	Operating systems	109
7051	Flynn	Mabel	931	Compilers	222

- What other information do we lose as a result?
  - We lose the fact that the class is taught in Room 109.
- The problem arises because the Teacher table really contains two separate sets of data: teacher data and class data.

Second Normal Form (2NF)

- Keep related data together (cohesiveness).

Teacher

Id	Last	First
7003	Rogers	Tom
7008	Thompson	Art
7012	Lane	John
7051	Flynn	Mabel

Primary key (PK)

Class

Class_code	Teacher_id	Subject	Room
908	7008	Data structures	114
926	7003	Java programming	101
931	7051	Compilers	222
951	7012	Software engineering	210
974	7012	Operating systems	109

Primary key (PK)

Foreign key (FK)

- How would you do this relation with a linking table?

Final Database Structure, *cont'd*

- John Doe takes Java programming, software engineering, and data structures.

Student

Id	Last	First
1001	Doe	John
1005	Novak	Tim
1009	Klein	Leslie
1014	Jane	Mary
1021	Smith	Kim

Teacher

Id	Last	First
7003	Rogers	Tom
7008	Thompson	Art
7012	Lane	John
7051	Flynn	Mabel

Student_Class

Student_id	Class_code
1001	926
1001	951
1001	908
1005	974
1005	908
1014	931
1021	926
1021	974
1021	931

Class

Code	Teacher_id	Subject	Room
908	7008	Data structures	114
926	7003	Java programming	101
931	7051	Compilers	222
951	7012	Software engineering	210
974	7012	Operating systems	109

Final Database Structure, *cont'd*

- The Java Programming class has John Doe and Kim Smith.

Student

Id	Last	First
1001	Doe	John
1005	Novak	Tim
1009	Klein	Leslie
1014	Jane	Mary
1021	Smith	Kim

Teacher

Id	Last	First
7003	Rogers	Tom
7008	Thompson	Art
7012	Lane	John
7051	Flynn	Mabel

Student_Class

Student_id	Class_code
1001	926
1001	951
1001	908
1005	974
1005	908
1014	931
1021	926
1021	974
1021	931

Class

Code	Teacher_id	Subject	Room
908	7008	Data structures	114
926	7003	Java programming	101
931	7051	Compilers	222
951	7012	Software engineering	210
974	7012	Operating systems	109

Final Database Structure, *cont'd*

- Mabel Flynn teaches compilers.

Student

Id	Last	First
1001	Doe	John
1005	Novak	Tim
1009	Klein	Leslie
1014	Jane	Mary
1021	Smith	Kim

Teacher

Id	Last	First
7003	Rogers	Tom
7008	Thompson	Art
7012	Lane	John
7051	Flynn	Mabel

Student_Class

Student_id	Class_code
1001	926
1001	951
1001	908
1005	974
1005	908
1014	931
1021	926
1021	974
1021	931

Class

Code	Teacher_id	Subject	Room
908	7008	Data structures	114
926	7003	Java programming	101
931	7051	Compilers	222
951	7012	Software engineering	210
974	7012	Operating systems	109