

# Database Schema Design

30 December 2009  
Lecture 11

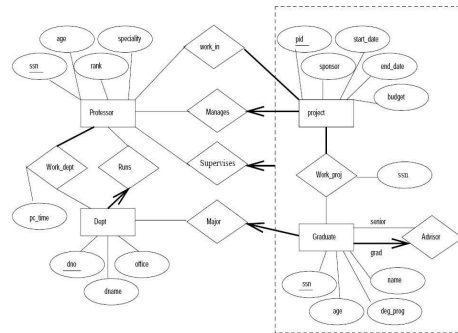
# HCI

- Read article by Gould and Lewis
  - Read 1 and 3
- Early focus on Users and Tasks
- Empirical Measurements
- Iterative Design
- Do you agree?
- For us
  - Current State – What did we do?

# Topics for Today

- Designing Database Schemas
  - Balancing
  - Designing Schemas
  - Access Steps
- Schemas and Normalization
- Source: PS98 9.1-9.3

# Example: ERD



# Designing Database Schemas

- Two methods discussed
  - Normalization
  - ERD
- We end up with
  - Relations
  - Constraints

# Designing Database Schemas

- The first step: make sure they are balanced
- Checking for balance means
  - Check all writes to the data source
  - Check all reads to the data source
  - Ensure all reads have a write before
  - Ensure all writes have a read after
- This is done *per data source*

## Balancing Data Sources

- Problems we may find:
  - Data flows which read data never written
  - Data flows which write something never read
  - Reads and Writes with race conditions

December 30, 2009

ISE 323: Information Systems Engineering 1

7

## Balancing Data Sources

- Read but not written
  - Usually impossible, it's probably a mistake
  - Maybe it's coming from the outside
  - Maybe it's derived from some other database values
  - Maybe there is something else going on

December 30, 2009

ISE 323: Information Systems Engineering 1

8

## Balancing Data Sources

- Written but not read
  - Might be superfluous
  - Maybe it's for later use
  - Maybe it's for external use or auditing
  - Discuss it with the team

December 30, 2009

ISE 323: Information Systems Engineering 1

9

## Balancing Data Sources

- Race conditions
  - A read and a write from different transactions compete to read or write the data source
  - Define some protocol for deciding what to do
  - Maybe ignore new changes and rely on old data
  - Maybe use locking to prevent this

December 30, 2009

ISE 323: Information Systems Engineering 1

10

## So far

- Designing Database Schemas
  - Balancing
  - Designing Schemas
  - Access Steps
- Schemas and Normalization

December 30, 2009

ISE 323: Information Systems Engineering 1

11

## Techniques for Schema Building

- Two techniques we'll talk about:
  - Normalization based
  - ERD based

December 30, 2009

ISE 323: Information Systems Engineering 1

12

## Normalization Techniques

- Start with a single data source
    - Either start with one large relation and break it down
  - Or
  - Start with a list of constraints and fields and do some narrowing first
- You end up with similar results
- Write down field dependencies
    - Get expert input
    - Don't assume the dictionaries are pre-normalized or complete
  - Put together a complete schema
    - Assemble the schemas from each data source
  - Look for overlap or identical relations
    - Identical relations
      - Collapse them into one
    - Similar relations
      - Leave them alone?
      - Combine them?
      - Hierarchy?

December 30, 2009

ISE 323: Information Systems Engineering 1

13

## ERD Technique

Using ERD, the steps are

1. Choose a data source from the DFD
  2. Get the inputs and outputs for the data source
  3. Figure out entities and relationships
  4. Build the ERD for the data source
  5. Repeat 1-4 for the next data source
  6. Each added data source may modify the relationships in the existing ERD
- Iteratively we develop a full data source schema
    - This leads to the database schemas as discussed above
  - In general
    - Each field is an attribute
    - Some attributes are used to enforce constraints

December 30, 2009

ISE 323: Information Systems Engineering 1

14

## So far

- Designing Database Schemas
  - Balancing
  - Designing Schemas
  - Access Steps
- Schemas and Normalization

December 30, 2009

ISE 323: Information Systems Engineering 1

15

## Defining Access Steps

- An access step is a channel for defining access to the data source from a transaction
- They are transaction level definitions
  - Every read and write needs one or more access steps
  - A single read or write line may imply several reads and writes to the database

December 30, 2009

ISE 323: Information Systems Engineering 1

16

## Defining Access Steps

- Find the appropriate relations
  - See which relation sets are for the given data source
  - If it's via normalization it's easier, if it's via ERD it's harder
  - Search by field name, type, or relation name
- Choosing the Participating Relations
  - Go over the relations found and decide which ones will really be used
  - Look at the keys, schemas, and fields
- Define the access details
  - Type of access
  - Key search fields
  - Fields accessed
- Exceptional Circumstances
  - Not all fields need to be read or written
  - Some reads and writes can be combined
  - Some may connect multiple times to the same relation

December 30, 2009

ISE 323: Information Systems Engineering 1

17

## Alternative: SQL

- An alternative: use SQL
- Access Steps are meant to make it clearer what each data flow is meant to read, write, update, or delete
  - Using SQL we can write that information down concisely

December 30, 2009

ISE 323: Information Systems Engineering 1

18

## Subschemas for the Database

- Subschemas
  - Subset of the full database schema
  - Used in incremental development
  - When there are multiple information systems
  - When there are multiple user types
  - Can be based on views
- Steps
  - Define the users
  - Define the transactions for the secondary or new system
  - Find the relations and fields which belong
- This can be done at the field and table access control level
- Also can be done using views and stored procedures

December 30, 2009

ISE 323: Information Systems Engineering 1

19

## So Far

- Designing Database Schemas
  - Balancing
  - Designing Schemas
  - Access Steps
- Schemas and Normalization

December 30, 2009

ISE 323: Information Systems Engineering 1

20

## Database Schemas

- We talked about relations, fields, and values in the Database Systems course
  - The PS book goes over this too in Hebrew
- We'll talk about converting from DFDs with data stores and data flows into database relations
- First: Some theory about good database schemas

December 30, 2009

ISE 323: Information Systems Engineering 1

21

## Good Schema Design

- Consider the following schema: Employees may work in one or more Departments.

Employees ( eid INT, ename CHAR(30), dno INT, dname CHAR(30), budget INT, daddress CHAR(40), PRIMARY KEY (eid, dno))

- Assume there is no separate Departments table
- What's the problem here?

December 30, 2009

ISE 323: Information Systems Engineering 1

22

## Anomalies

Employees ( eid INT, ename CHAR(30), dno INT, dname CHAR(30), budget INT, daddress CHAR(40), PRIMARY KEY (eid, dno))

eid	ename	dno	dname	budget	daddress
1	Adam	100	Toys	10,000	58 Hyacinth
2	Beth	100	Toys	10,000	58 Hyacinth
3	Cody	101	Sprokets	20,000	78 Oglethorpe
4	Dirk	101	Sprokets	20,000	78 Oglethorpe
5	Ernie	102	Joomlas	21,000	34 Juniper

What happens if we delete Adam and Beth?  
What if we want to add a new Employee?  
What if we want to add a new Department?  
What if we want to update the budget for Toys?

December 30, 2009

ISE 323: Information Systems Engineering 1

23

## Anomalies

- Three major kinds which can pop up:
  - Insertion Anomalies: Where inserting data involves adding nonessential or unrelated data to the intended data
  - Deletion Anomalies: Where deleting data involves deleting additional unintended information
  - Update Anomalies: Where updating data requires multiple edits

To prevent these anomalies, *normalization* techniques have been designed.

December 30, 2009

ISE 323: Information Systems Engineering 1

24

## Functional Dependencies



- Data Dependency: One field's value is dependent on another's
- Functional Dependency: Field A uniquely determines the values in B:  $F(A) \rightarrow B$ 
  - Many values may map to a particular B
  - Example:  $F(\text{sid}) \rightarrow \text{name}$
- One to One Dependency: Field A uniquely determines the values in B and no other A matches a particular B:  $A \leftrightarrow B$ 
  - Example: TZ and Passport Number
- Complex Dependency: Multiple A's are matched multiple B's, but there is a connection:  $A \leftrightarrow B$ 
  - Example: sid and courseid
- Transitive Dependency: A determines B, B determines C, so A determines C
  - Example:  $\text{departmentid} \rightarrow \text{managerid}$ ,  $\text{managerid} \rightarrow \text{salary}$  therefore  $\text{departmentid} \rightarrow \text{salary}$

December 30, 2009

ISE 323: Information Systems Engineering 1

25

## 1<sup>st</sup> and 2<sup>nd</sup> Normal Forms



- 1<sup>st</sup> normal form:
  - All rows are of uniform length
  - Essentially, it's a table
- 2<sup>nd</sup> normal form:
  - 1<sup>st</sup> normal form
  - There is a primary key
  - All fields are dependent on the whole primary key, not just part of it
- Example:
  - Table1 (**StudentID INT, CourseCode INT**, StudentName CHAR(20), CourseName CHAR(20), Grade REAL)
  - Student ID  $\rightarrow$  Student Name 
  - Course Code  $\rightarrow$  Course Name 
  - (Student ID  $\leftrightarrow$  Course Code)  $\rightarrow$  Grade

December 30, 2009

ISE 323: Information Systems Engineering 1

26

## 3<sup>rd</sup> Normal Form


- 3<sup>rd</sup> Normal Form
  - 2<sup>nd</sup> normal form
  - No non-key field is dependent on any other field in the relation
    - This excludes transitive dependencies on the primary key
- Example:
  - Table2 (**StudentID INT**, StudentName CHAR(20), DepartmentID INT, DepartmentBudget REAL)
  - Student ID  $\rightarrow$  Student Name
  - Student ID  $\rightarrow$  Department ID
  - Department ID  $\rightarrow$  Department Budget 
  - Student ID  $\rightarrow$  Department ID  $\rightarrow$  Department Budget 

December 30, 2009

ISE 323: Information Systems Engineering 1

27

## Boyce Codd Normal Form

- BCNF
  - 3<sup>rd</sup> Normal Form
  - For any non-trivial dependency in the relation  $F(A) \rightarrow B$ , A must be the full primary key (or a superset of it)
    - It's hard to find a case of BCNF and not 3<sup>rd</sup> Normal Form
- Example:
  - Table3 (**SupplierID INT, SupplierCode INT, PartCode INT**, AmountOrdered INT)
  - Supplier ID  $\leftrightarrow$  Supplier Code 

December 30, 2009

ISE 323: Information Systems Engineering 1

28

## Conclusion

- Designing Database Schemas
  - Balancing
  - Designing Schemas
  - Access Steps
- Schemas and Normalization

December 30, 2009

ISE 323: Information Systems Engineering 1

29