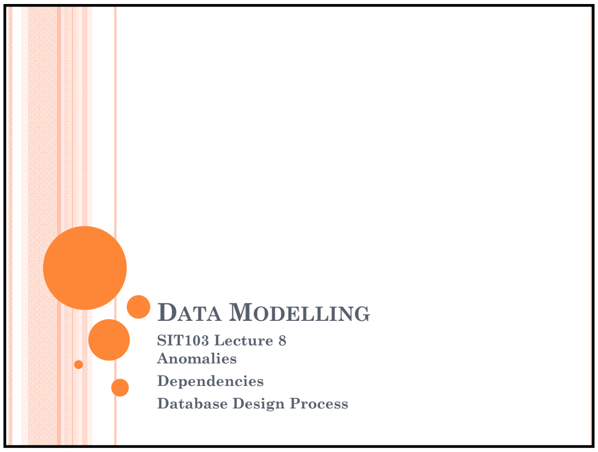
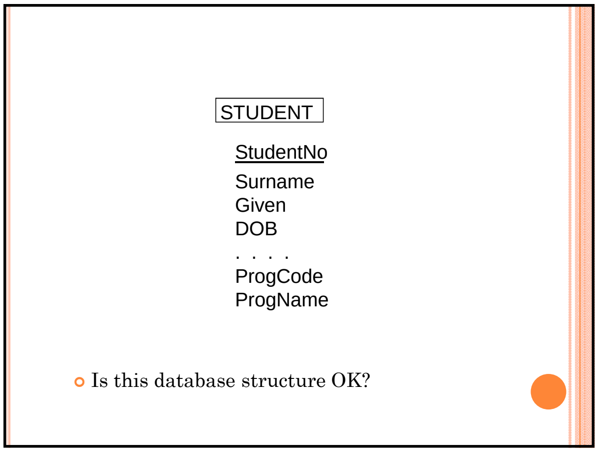


Data Modelling

SIT103 Lecture 8
Anomalies

Dependencies

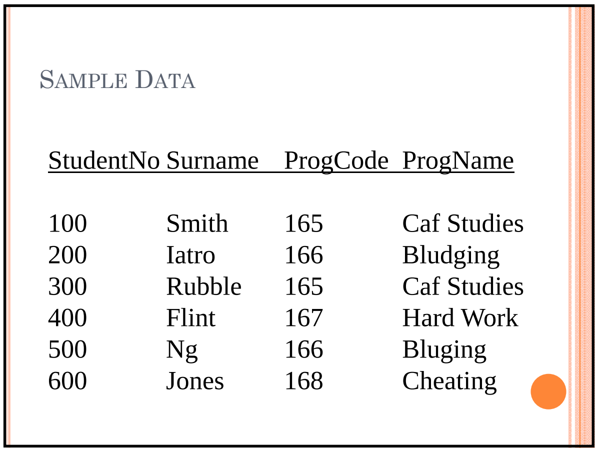
Database Design Process



STUDENT

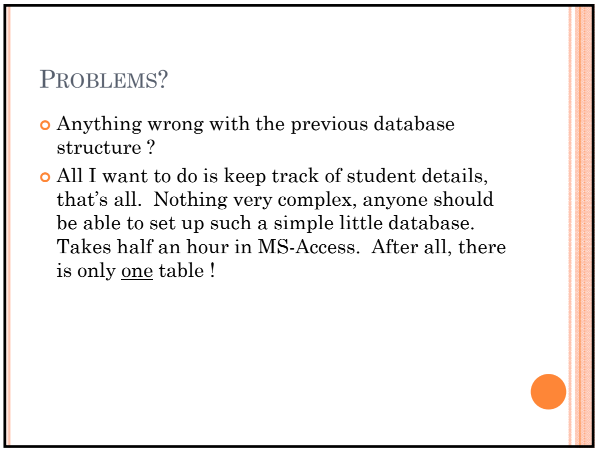
<u>StudentNo</u>
Surname
Given
DOB
. . . .
ProgCode
ProgName

- Is this database structure OK?



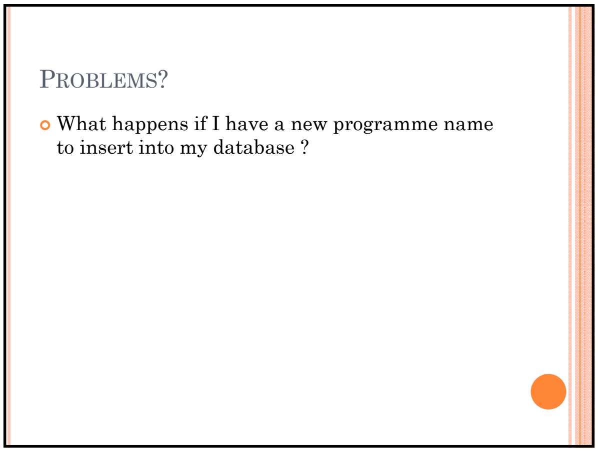
Sample Data

StudentNo	Surname	ProgCode	ProgName
100	Smith	165	Caf Studies
200	Iatro	166	Bludging
300	Rubble	165	Caf Studies
400	Flint	167	Hard Work
500	Ng	166	Bluging
600	Jones	168	Cheating



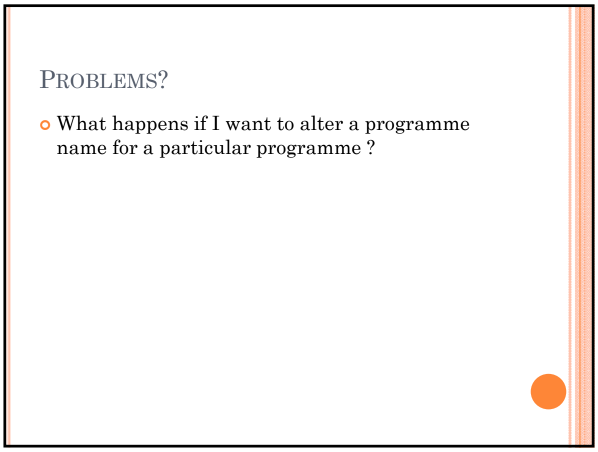
PROBLEMS?

- Anything wrong with the previous database structure ?
- All I want to do is keep track of student details, that's all. Nothing very complex, anyone should be able to set up such a simple little database. Takes half an hour in MS-Access. After all, there is only one table !



PROBLEMS?

- What happens if I have a new programme name to insert into my database ?



PROBLEMS?

- What happens if I want to alter a programme name for a particular programme ?

PROBLEMS ?

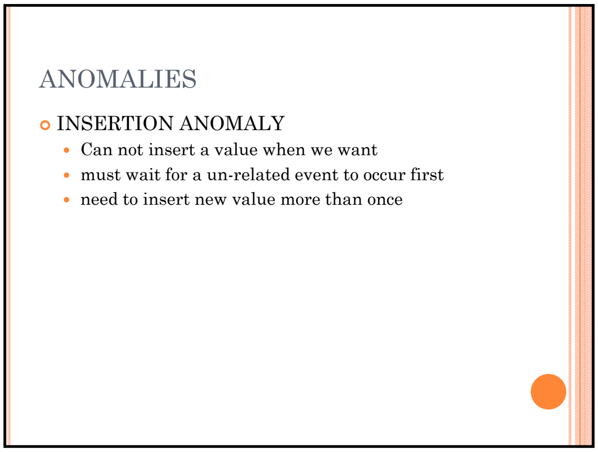
- What happens if all the students in a programme drop out ?

PROBLEMS ?

- Are all copies of the same programme name the same ?

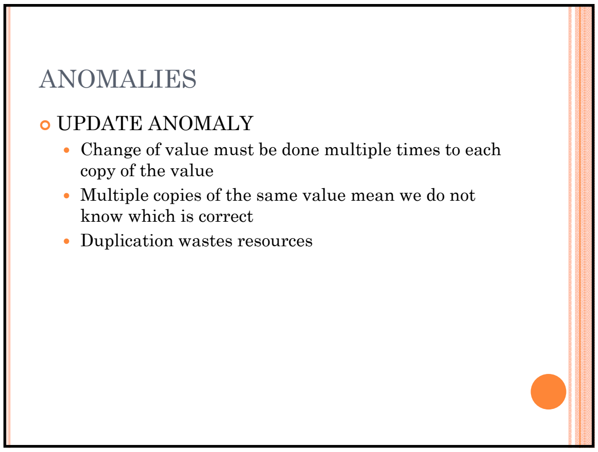
ANOMALIES

- An Inconsistency
- A database structure that will be prone to errors in the data.
- Not necessarily wrong, but will promote erroneous data rather than prevent it.
- Will usually also involve <u>more</u> effort in programs/users that maintain the file.



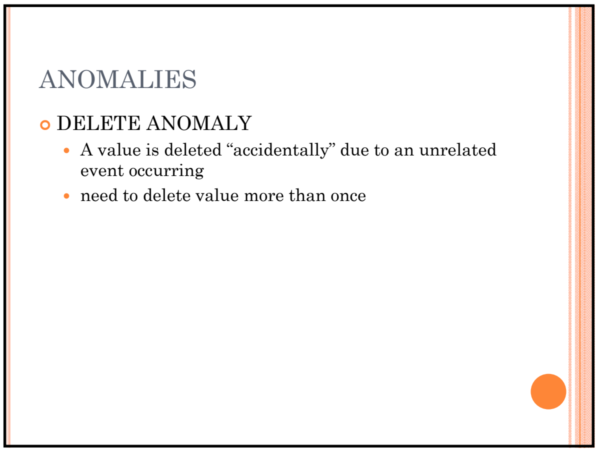
ANOMALIES

- INSERTION ANOMALY
  - Can not insert a value when we want
  - must wait for a un-related event to occur first
  - need to insert new value more than once



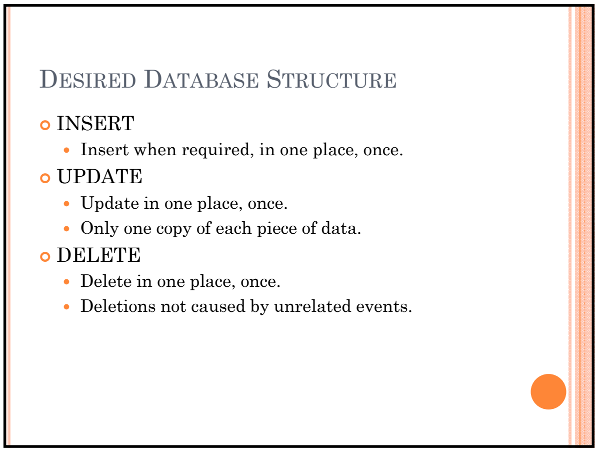
ANOMALIES

- UPDATE ANOMALY
  - Change of value must be done multiple times to each copy of the value
  - Multiple copies of the same value mean we do not know which is correct
  - Duplication wastes resources



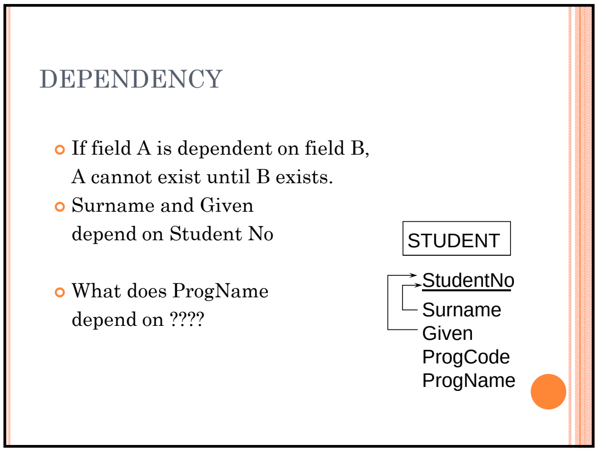
ANOMALIES

- DELETE ANOMALY
  - A value is deleted “accidentally” due to an unrelated event occurring
  - need to delete value more than once



Desired Database Structure

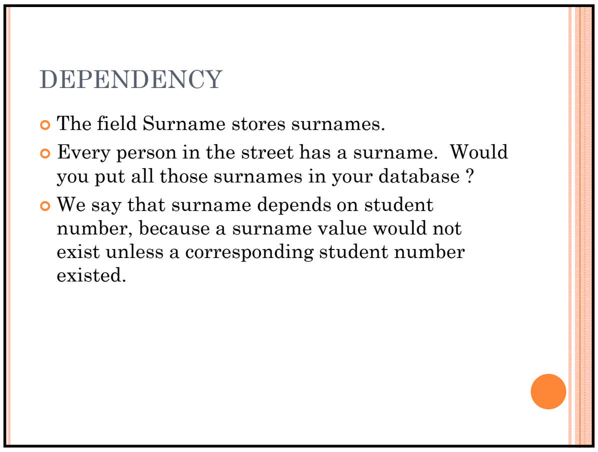
- INSERT
  - Insert when required, in one place, once.
- UPDATE
  - Update in one place, once.
  - Only one copy of each piece of data.
- DELETE
  - Delete in one place, once.
  - Deletions not caused by unrelated events.



DEPENDENCY

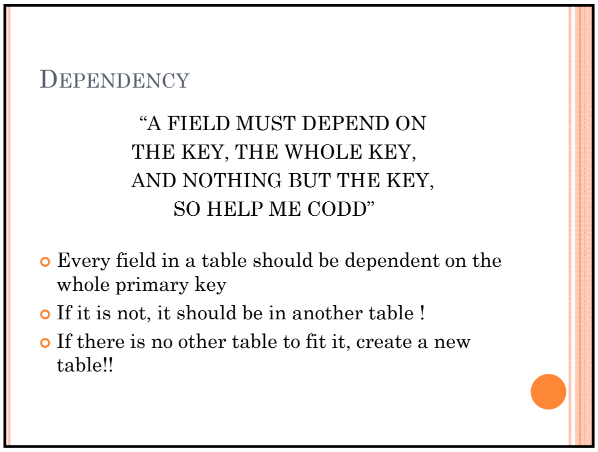
- If field A is dependent on field B, A cannot exist until B exists.
- Surname and Given depend on Student No
- What does ProgName depend on ????

STUDENT
StudentNo
Surname
Given
ProgCode
ProgName



DEPENDENCY

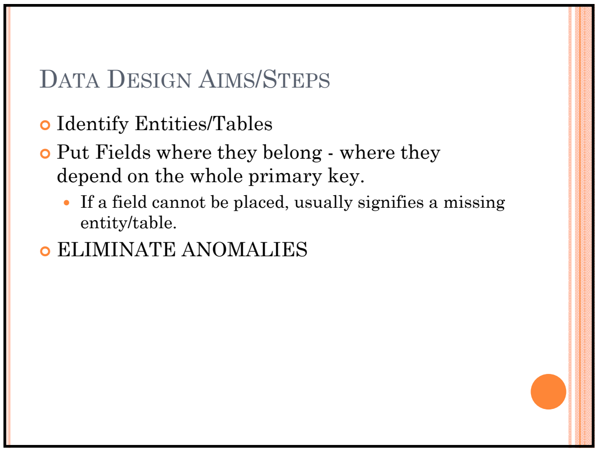
- The field Surname stores surnames.
- Every person in the street has a surname. Would you put all those surnames in your database ?
- We say that surname depends on student number, because a surname value would not exist unless a corresponding student number existed.



Dependency

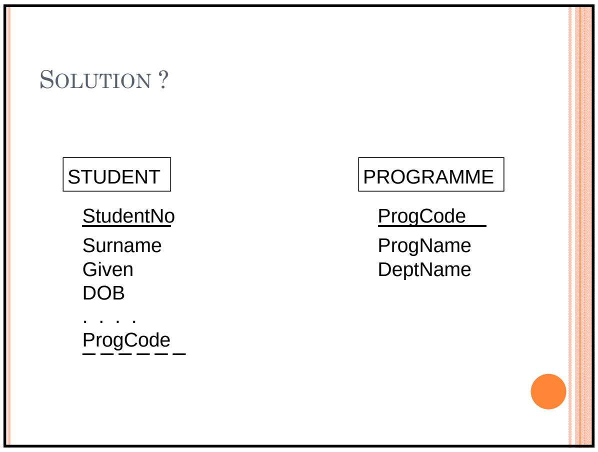
"A FIELD MUST DEPEND ON
THE KEY, THE WHOLE KEY,
AND NOTHING BUT THE KEY,
SO HELP ME CODD"

- Every field in a table should be dependent on the whole primary key
- If it is not, it should be in another table !
- If there is no other table to fit it, create a new table!!



Data Design Aims/Steps

- Identify Entities/Tables
- Put Fields where they belong - where they depend on the whole primary key.
  - If a field cannot be placed, usually signifies a missing entity/table.
- ELIMINATE ANOMALIES



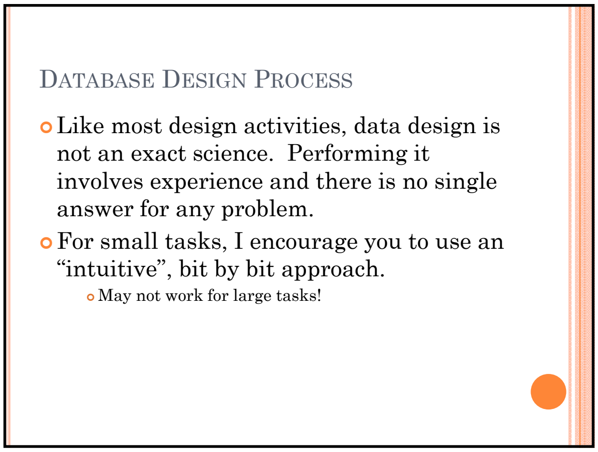
Solution ?

STUDENT

StudentNo
Surname
Given
DOB
. . . .
ProgCode

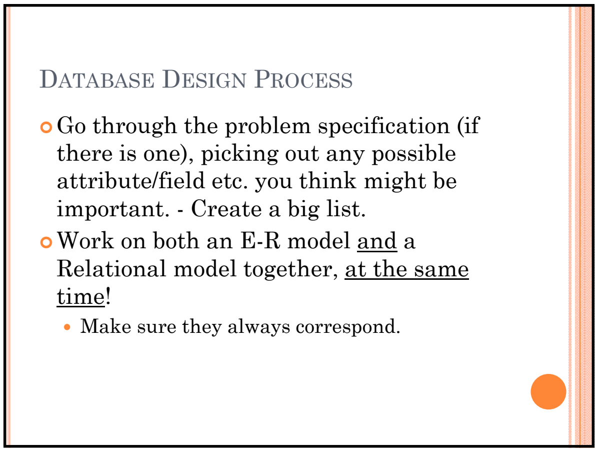
PROGRAMME

ProgCode
ProgName
DeptName



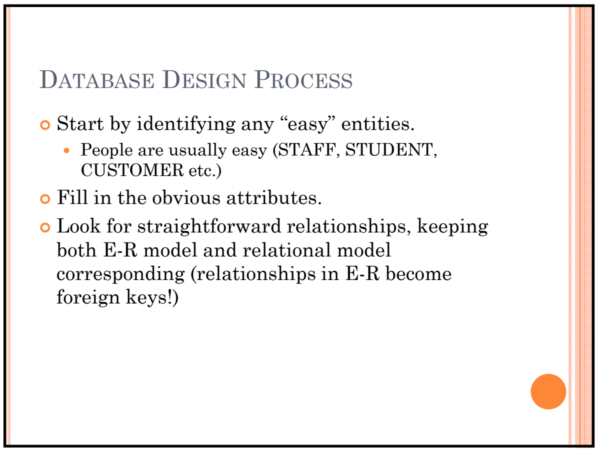
Database Design Process

- Like most design activities, data design is not an exact science. Performing it involves experience and there is no single answer for any problem.
- For small tasks, I encourage you to use an “intuitive”, bit by bit approach.
  - May not work for large tasks!



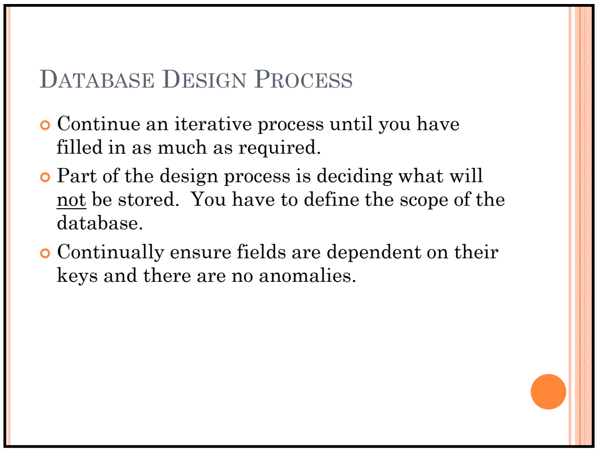
Database Design Process

- Go through the problem specification (if there is one), picking out any possible attribute/field etc. you think might be important. - Create a big list.
- Work on both an E-R model and a Relational model together, at the same time!
  - Make sure they always correspond.



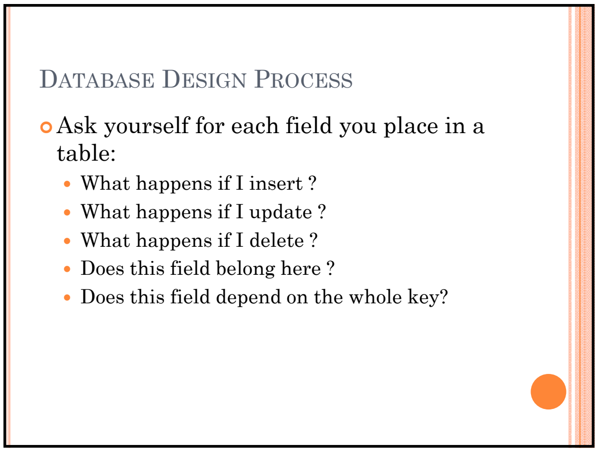
Database Design Process

- Start by identifying any “easy” entities.
  - People are usually easy (STAFF, STUDENT, CUSTOMER etc.)
- Fill in the obvious attributes.
- Look for straightforward relationships, keeping both E-R model and relational model corresponding (relationships in E-R become foreign keys!)



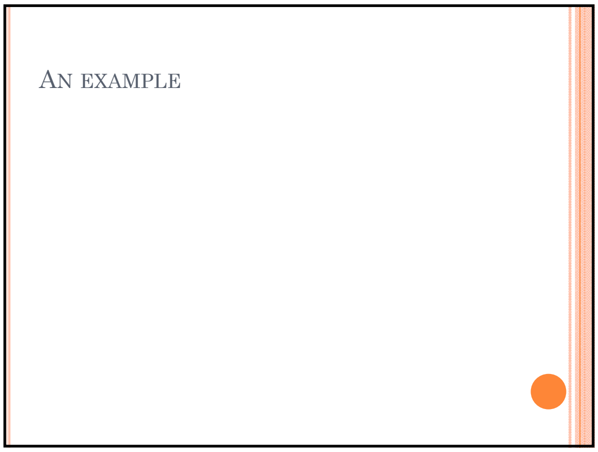
Database Design Process

- Continue an iterative process until you have filled in as much as required.
- Part of the design process is deciding what will not be stored. You have to define the scope of the database.
- Continually ensure fields are dependent on their keys and there are no anomalies.

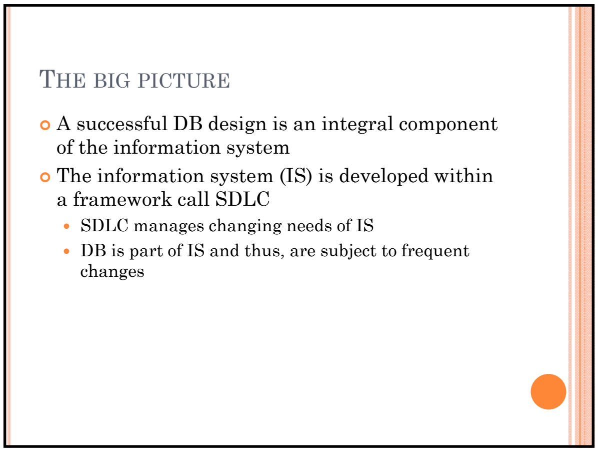


Database Design Process

- Ask yourself for each field you place in a table:
  - What happens if I insert ?
  - What happens if I update ?
  - What happens if I delete ?
  - Does this field belong here ?
  - Does this field depend on the whole key?

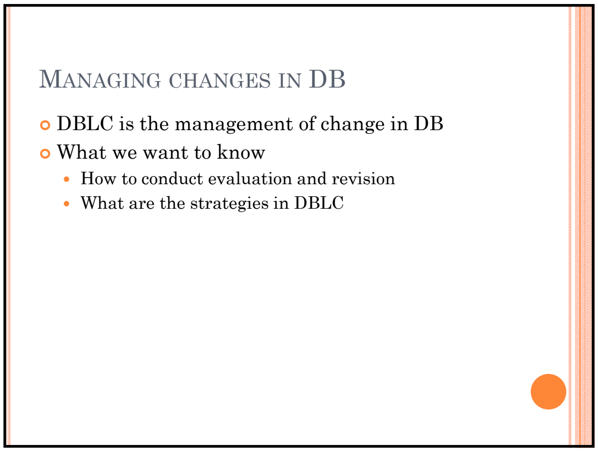


An example



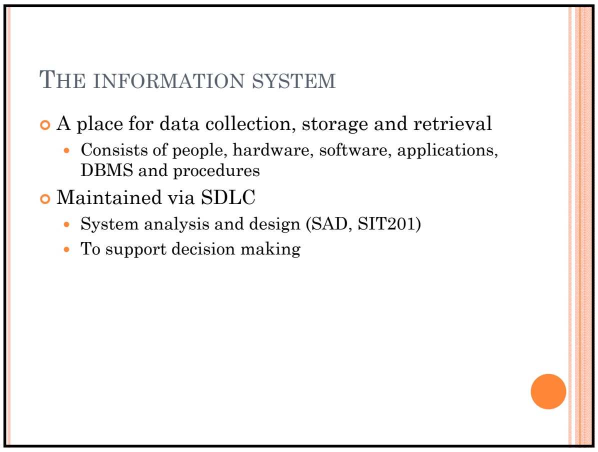
The big picture

- A successful DB design is an integral component of the information system
- The information system (IS) is developed within a framework call SDLC
  - SDLC manages changing needs of IS
  - DB is part of IS and thus, are subject to frequent changes



Managing changes in DB

- DBLC is the management of change in DB
- What we want to know
  - How to conduct evaluation and revision
  - What are the strategies in DBLC

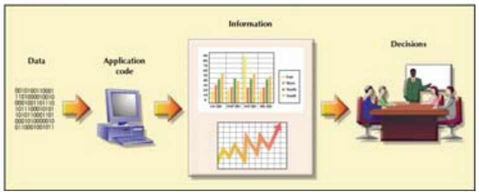


The information system

- A place for data collection, storage and retrieval
  - Consists of people, hardware, software, applications, DBMS and procedures
- Maintained via SDLC
  - System analysis and design (SAD, SIT201)
  - To support decision making

Conceptual view

FIGURE 8.1 Generating Information for Decision Making

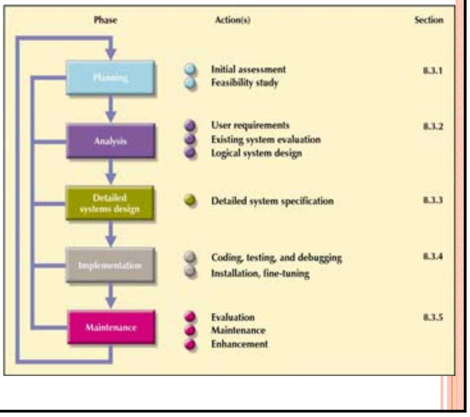


System development lifecycle

- The development and maintenance of IS
- The bigger framework which DBLC exists
- Divided into 5 phases
  - Planning
  - Analysis
  - Detailed systems design
  - Implementation
  - Maintenance

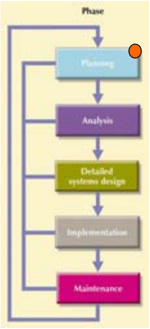
SDLC

FIGURE 8.2 The Systems Development Life Cycle (SDLC)



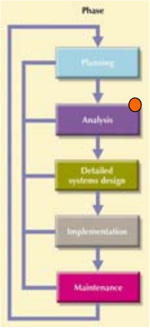
Phase	Action(s)	Section
Planning	Initial assessment Feasibility study	8.3.1
Analysis	User requirements Existing system evaluation Logical system design	8.3.2
Detailed systems design	Detailed system specification	8.3.3
Implementation	Coding, testing, and debugging Installation, fine-tuning	8.3.4
Maintenance	Evaluation Maintenance Enhancement	8.3.5

SDLC (CONT’D)



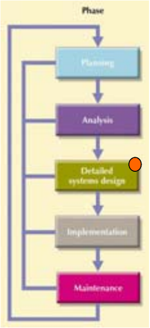
- Gets an overview of business rules
  - Assessment made from requirements obtained
  - Starting point for evaluating alternate solutions
  - Determines costs

SDLC (CONT’D)

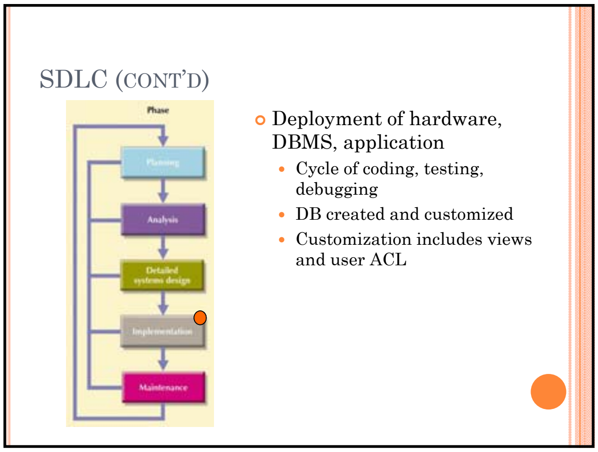


- Examine the problem defined during planning
  - Includes audit of user requirements
  - Study existing hardware and software
- Understanding of the system’s potential and problem

SDLC (CONT’D)

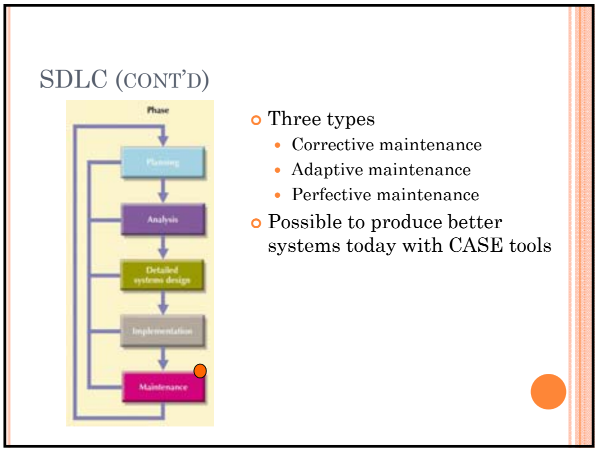


- Logical system design
  - Conceptual data model, processes and output
  - DFD, ERD, UML
- Detailed system design
  - Technical specifications
  - Planning of methodologies



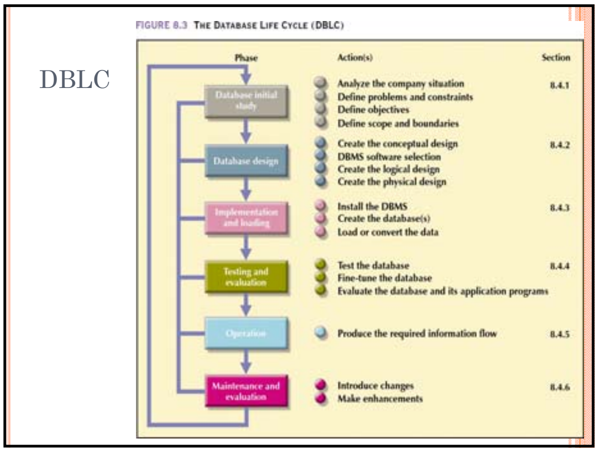
SDLC (CONT'D)

- Deployment of hardware, DBMS, application
  - Cycle of coding, testing, debugging
  - DB created and customized
  - Customization includes views and user ACL



SDLC (CONT'D)

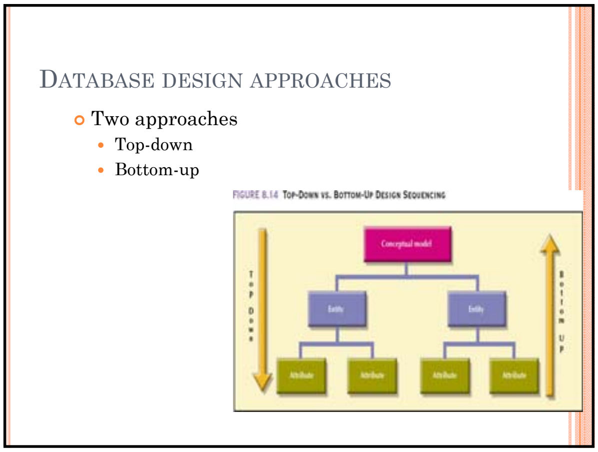
- Three types
  - Corrective maintenance
  - Adaptive maintenance
  - Perfective maintenance
- Possible to produce better systems today with CASE tools



DBLC

FIGURE 8.3 THE DATABASE LIFE CYCLE (DBLC)

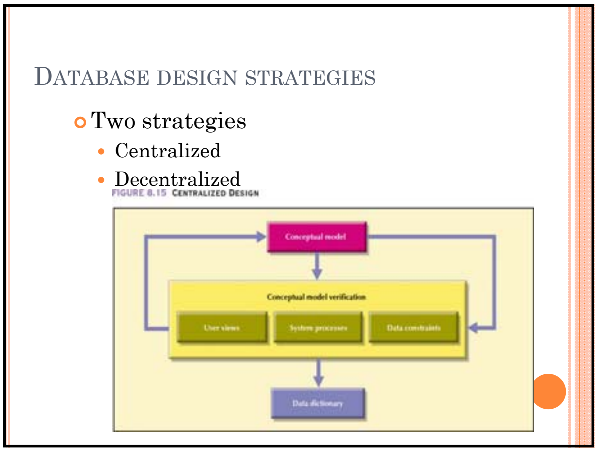
Phase	Action(s)	Section
Database initial study	Analyze the company situation Define problems and constraints Define objectives Define scope and boundaries	8.4.1
Database design	Create the conceptual design DBMS software selection Create the logical design Create the physical design	8.4.2
Implementation and loading	Install the DBMS Create the database(s) Load or convert the data	8.4.3
Testing and evaluation	Test the database Fine-tune the database Evaluate the database and its application programs	8.4.4
Operation	Produce the required information flow	8.4.5
Maintenance and evaluation	Introduce changes Make enhancements	8.4.6



Database design approaches

- Two approaches
  - Top-down
  - Bottom-up

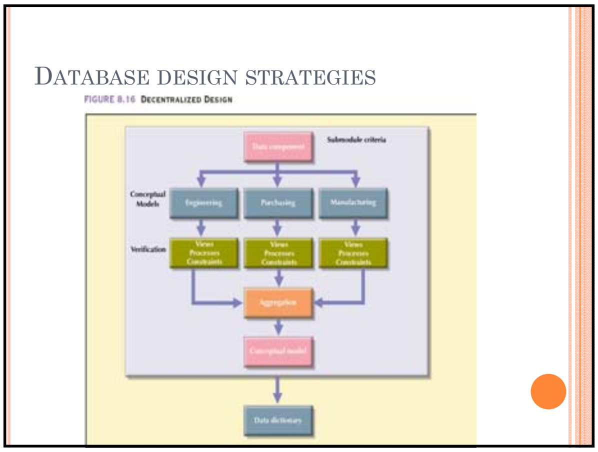
FIGURE 8.14 Top-Down vs. Bottom-Up Design Sequencing



Database design strategies

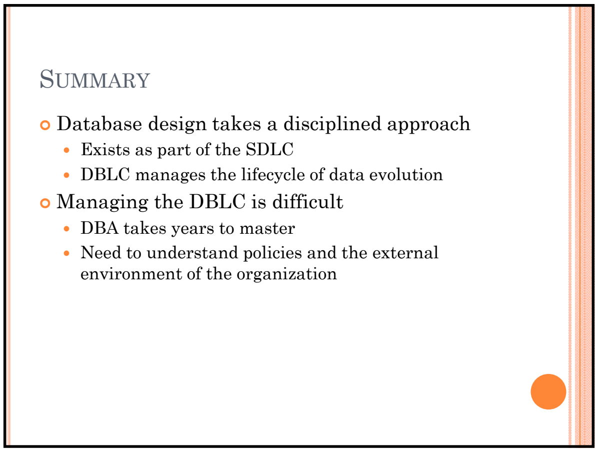
- Two strategies
  - Centralized
  - Decentralized

FIGURE 8.15 Centralized Design



Database design strategies

FIGURE 8.16 Decentralized Design



Summary

- Database design takes a disciplined approach
  - Exists as part of the SDLC
  - DBLC manages the lifecycle of data evolution
- Managing the DBLC is difficult
  - DBA takes years to master
  - Need to understand policies and the external environment of the organization