West Florida Mall

Case study

Introduction

- A new mall, west florida mall just had it’s grand opening 3 weeks ago.
- It is a part of a series of malls owned by a parent company
- We need a database to keep track of the management of the mall in terms of
    - Stores
    - Owners
    - Workers of the stores

Initial user specifications

1) Mall Information
   a) Mall’s name and address
   b) A mall must contain one or more stores
2) Store Information
   a) Store number
   b) Name
   c) Location
   d) Departements
   e) Owner
   f) Manager
   g) Each department is managed by a manager
   h) Each store will have only one store manager
   i) Each store is owned by only one owner
   j) Each store is located in one and only one mall

3) Store Manager Information

   a) Name
   b) Social security number
   c) Which store he or she is working for
   d) salary

4) Store Owner Information

   a) Name
   b) Address
   c) Office phone number
   d) May own more than one store

Step one

- Select one primary entity from the Database requirements description.

- Show attributes to be recorded for that entity.

- We will choose Mall as our primary entity.

Step two

- Use structured english for describing the Database :
    - Entities
    - Attributes
    - keys

The Entity

- For each Mall in the Database we record :
  - Name
  - Address
  - store_names

Attributes for Mall

- For each Mall there will be :
  - One and only one name
  - One and only one address
  - There may be more than one store_name recorded for each Mall

The keys

- For each Mall we assume that the mall name will be unique.

Relations

Mall

Name	Address

Mall-store

Name	store_name

Case study

Part two

Step three

- Examine attributes in the primary entity (with user assistance) to find out if information about one of the attributes is to be recorded.

- We need to store information about the attribute STORE

- So by turning the attribute STORE into an entity we need to repeat step two

The Entity

- For each STORE in the database we record:
  - Store name (sname)
  - Store number (snum)
  - Store location (sloc)
  - Departments (dept)

The attribute for STORE

- For each store there will be one and only one
    - Sname
    - Snum
    - Sloc
    - More than one dept

The keys

- We assume that the snum will be unique.

Step three (B)

- Define the relationship back to the original entity

- There is a relationship (located_in) between STORE and MALL

Step four

- If another entity is appropriate draw the second entity with it's attributes
- We select another entity STORE_MANAGER
- Repeat step two for that entity

The Entity

- For each STORE_MANAGER in the database we record:
    - Store manager name (sm_name)
    - Store manager social security number (sm_ssn)
    - Store manager salary (sm_salary)

The attributes for store_manager

- For each STORE_MANAGER there will be one and only one :
  - Sm_name
  - Sm_ssn
  - sm_salary

The keys

- For each STORE_MANAGER we will assume that the sm_snn will be unique.

Step five

- Connect entities with relationships if exist
- There is a relationship (manages) between STORE and STORE_MANAGER
- We select our next primary entity OWNER
- Repeating step two for OWNER

The Entity

- For each OWNER in the database we record:
    - Store owner name (so_name)
    - Store owner social security number (so_ssn)
    - Store owner office phone (so_of_phone)
    - Store owner address (so_address)

The attributes for OWNER

- For each OWNER there will be one and only one :
  - So_name
  - So_ssn
  - So_off_phone
  - so_address

The keys

- For each OWNER we will assume that the so_ssn will be unique.

Step five

- Connect entities with relationships if exist

- There is a relationship (owns) between STORE and OWNER

Relations

MALL

name	address

STORE

sloc	sname	snum

STORE_DEPT

snum	depts

OWNER

so_ssn	so_name	so_off_phone	so_address

STORE_MANAGER

sm_snn	sm_name	sm_salary

Case study

Part three

Step six

- State the exact nature of the relationships in structured English from all sides
- Example:
  - when a relationship is A:B::1:M, there is a relationship from A(1) to B(M) and from B(M) back to A(1)

The relationship located_in

- From MALL to STORE
  - A mall must have at least one store and can have many stores
  - Malls, which are recorded in the database, must have many (one or more)stores located in them
  - 1(full):N
- From STORE to MALL
  - Many stores (one or more) must be in one mall
  - Stores, which are recorded in the database, must be in one mall
  - M(full):1

Mapping

- we have to map the relationship between the MALL entity and the STORE entity

- This is a binary 1:N relationship

- Include the key of the entity on the 1 side of the relationship to the N side as a foreign key

The relationship owns

- From OWNER to STORE
  - Owners, which are recorded in the database, must own one or more stores
  - One owner must own at least one store and may own many stores
  - 1(full):M
- From STORE to OWNER
  - Stores, which are recorded in the database, must have one and only one owner
  - Many stores can have one owner
  - M(full):1

Mapping

- Include the key of the entity on the 1 side of the relationship to the N side as a foreign key

The relationship manages

- From STORE to STORE MANAGER
  - Stores, which are recorded in the database, must have one store manager
  - Stores must have one store manager and can only have one and only one store manager
  - 1(full):1
- From STORE MANAGER to STORE
  - Store managers, which are recorded in the database, must manage one and only one store
  - Store managers must manage at least one store and can manage only one store
  - 1(full):1

Mapping

- Take the key from STORE MANAGER and include it in STORE as the foreign key

Relations

MALL

name	address

Store

sloc	sname	snum	mall_name	so_ssn	sm_ssn

Store_dept

snum	depts

Owner

so_ssn	so_name	so_off_phone	so_address

Store_manager

sm_ssn	sm_name	sm_salary

Case study

Part four

Additional input from the user

- We need additional information for departments
- Each department has at least one employee working for it
- We have to record information about the employees in the store
- Now we have two new entities
  - Departments
  - Employee

The Entity

- For each DEPARTMENT in the database, we record :
    - department name (dname)
    - department number (dnum)

The Attributes for DEPARTMENT

- For each DEPARTMENT there will be one and only one :
  - Dname
  - Dnum

The Keys

- For each DEPARTMENT , we do not assume that any attribute will be unique enough to identify individual entities without the accompanying reference to STORE, the owner entity.

The Entity

- For each EMPLOYEE in the database, we record :
  - employee name (ename)
  - employee Social Security number (essn)

The Attributes for EMPLOYEE

- For each EMPLOYEE, there will be one and only one :
  - Ename
  - Essn

The Keys

- For each EMPLOYEE, we will assume that the essn will be unique

structural constraints

- the relationship dept_of :
    - From STORE to DEPARTMENT
        - Stores, which are recorded in the database, must have many (one or more) departments
    - From DEPARTMENT to STORE
        - Many departments (one or more) must be in one store
        - 1(full):N
        - M(full):1

structural constraints

- the relationship works_for :
    - Departments, which are recorded in the database, must have one or more employees working for it.
        - 1(full):N

Relations

MALL

name	address

Store

sloc	sname	snum	mall_name	so_ssn	sm_ssn

Owner

so_ssn	so_name	so_off_phone	so_address

Store_manager

sm_ssn	sm_name	sm_salary

Department

dname	dnum	snum

Employee

ename	essn	dnum	snum

Case study

Part five

Additional input from the user

- An employee can also be a department manager
- A department manager can manage at most one department
- A department manager supervises at least one employee and may manage several employees
- we have a (recursive relationship) developing since an employee can be a department manager supervising other employees

Relation

Employee

ename	essn	snum	dm_ssn	dnum

Case study

Part six

New Entity

- A PERSON may be an owner, employee, or manager

- For each PERSON, we will record the name, Social Security number address, and phone number.

The Entity

- For each PERSON in the database, we record :
  - person’s name (pname)
  - person’s Social Security number (pssn)
  - person’s phone (pphone)
  - person’s address (padd)

The Attributes for PERSON

- For each PERSON, there will be one and only one :
  - Pname
  - Pssn
  - Pphone
  - padd

The Keys

- For each PERSON, we will assume that the pssn will be unique

structural constraints

- there is a disjoint relationship between PERSON and STORE_ MANAGER,

  OWNER, and EMPLOYEE
- This means that a person may be an owner, store manager, or an employee
- (a disjoint generalization/specialization relationship)

Relations

MALL

name	address

Store

sloc	sname	snum	mall_name	so_ssn	sm_ssn

Owner

so_ssn	so_off_phone

Store_manager

sm_ssn	sm_salary

Department

dname	dnum	snum

Employee

essn	snum	dm_ssn	dnum

Person

pssn	pname	padd	pphone

ER Design Methodology

Step one :

Select one, primary entity from the database requirements description and show attributes to be recorded for that entity, Label keys if appropriate.

Step two:

Use structured English for entities, attributes, and keys to describe the database that has been elicited.

Step three:

Examine attributes in the existing entities (possibly with user assistance) to find out if information about one of the entities is to be recorded.

Step three (A):

If information about an attribute is needed, then make the attribute an entity.

Step three (B):

Define the relationship back to the original entity.

Step four:

If another entity is appropriate, draw the second entity with its attributes. Repeat steps 2 and 3 to see if this entity should be further split into more entities.

Step five:

Connect entities with relationships (one or more) if relationships exist.

Step six:

State the exact nature of the relationships in structured English from all sides, for example, if a relationship is A:B::1:M, then there is a relationship from A(1) to B(M) and from B(M) back to A(1).

Step seven:

Show some sample data.

Step eight:

Present the “as designed” database to the user complete with the English for entities, attributes, keys, and relationships. Refine the diagram as necessary.