Database Systems
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

What is SRA?

A specification of the Software Requirements

Not a design document.. Leaves all design decisions to the design stage
Database Systems
Data Model - Entity Relationship Diagram

User Requirements Spec (URS)
Database Systems
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

SRA?

Business Analyst

S/W Analyst

Architect / Design team
S/W Requirements Analysis (SRA)

Who conducts SRA?

Experienced Software Analysts

Why SRA?

Why not just do it as first step?
Skills required are different from URS
Provides to two different audiences
Gives an opportunity to evaluate
S/W Requirements Analysis (SRA)
What do we do during SRA?
Analyze the URS
Validate the URS
Model the Requirements SSAD (Functional Model - DFD (Data Flow Diagram), Data Model – ER Diagram (Entity Rel. Diagram), Behavior Model - State Transition Diagram)
S/W Requirements Analysis (SRA)
What do we do during SRA?

Model the Requirements OOAD
Use Cases
Class Diagrams
Object Interaction
Identify unstated Problems
Identify, Synthesize Candidate Solutions

STATE
TRANSITION
DIAGRAM
S/W Requirements Analysis (SRA)

What do we do during SRA?

- Identify Conflicting requirements
- Identify potential problems with
  - Requirements
    - Performance
    - Scalability
    - Usability, Maintainability
S/W Requirements Analysis (SRA)

What do we do during SRA?

• Identify deliverable functions
• Trace Requirements to functions
• Validate
• Draft SRS (Software Requirement Specifications)
S/W Requirements Analysis (SRA)

- Forms the bridge between Domain Experts (the person with Market knowledge) and the Software Development Team (or more specifically software systems analyst)
- More precise software estimation
- Identify missing requirements
Validate Requirement

Documentation

Refactoring/Transformation

Static analysis

Rapid prototyping

Code generation

Automated testing

Model
Why Model? Summary

- Are the requirements valid?
- Visualize the finished product
- Better communication with Users
- A precise specification
- Early Defect detection
- Missing, Conflicting requirements
Database Systems
Data Model - Entity Relationship Diagram

Why Model? Requirement Problems

- Not specific
- Changing requirements
- Missing Requirement
- Conflicting Requirements
- Early Defect detection
Why Model?

- Road map to design
- Transformations promote traceability
- Documentation
Why Model?

- Code generation
- Test Case and Test generation
- Prototyping
Database Systems
Data Model - Entity Relationship Diagram

Why Model?

• Analysis – vanishing skill

  ➔ requirements cannot be validated
  ➔ Design cannot be derived / traced

Break down of Life Cycle processes
Why Model?

Without Models

• Design can only be done
  • Step by Step only
  • Full system vision is not there
  • Evolutionary Design
    ➔ Incremental delivery ➔ Iterative
    ➔ Agile
Database Systems

Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

Data Model

What is a Data model?

Logical

Physical
Database Systems
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

Data Model – E-R Diagram

Data is the perspective

• Entities
• Attributes
• Relationships

Similar to Class diagram of OOAD

What is an Entity? Anything about which our system cares and needs to process, persist and protect
S/W Requirements Analysis (SRA)

Data Model – E-R Diagram

Examples of entities
Person
Book
Account

Attributes of Person entity
Name
Age, Gender
Data Model – E-R Diagram – Misconceptions

- ER is a design technique
- ER is only for projects based on Relational DB

ER is a technique to analyze the proposed system requirements from the perspective of data. It enables the analyst to detect defects in requirements, whether complete or not.

Though Relational DB projects typically use ER to create database objects, the technique applies to all projects.
Database Systems
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

Data Model – E-R Diagram

What identifies it

Relationships How each entity is related to another
Represented by Diamonds or just a string
Person READS Book
Database Systems
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

Data Model – E-R Diagram

Cardinality

Person READS 1 to N Book
Can be 1 To 1
Many to Many (should not permit .. must resolve)

Modality

Mandatory (must Read)
Optional (may Read)
Database Systems
Data Model – ERD Tools

Enterprise Architect
http://www.sparxsystems.com/products/ea/

MySQL Workbench
http://dev.mysql.com/downloads/workbench/5.1.html

ORACLE SQL Developer Data Modeler

MS Entity Framework

ER Win
Database Systems
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

Data Model – Entity Rules

Not a single thing but a generalization
Meaningful Name
  Communicative
Not refer to a specific instance
Expresses a single concept
Why is it important to business
Avoid Single instance entities

CS Department
English Department

These are actually instances of the same entity called Department
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

Data Model – Entity Rules
Avoid Single instance entities

Why?

Model instability .. Need to create new entities
Poor translation to database design
Database Systems

Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

Data Model – Entity Rules

Entity without Relationship

Must have a relationship with at least one other entity … Incomplete modeling..
Data is never isolated

Many to Many Mandatory

Duplicate Entities Same entity named differently in different departments
S/W Requirements Analysis (SRA)

Data Model – Entity Rules

Entity without Relationship

Must have a relationship with at least one other entity … Incomplete modeling.. Data is never isolated

Many to Many Mandatory

Student must register 1-N Courses
A course is attended by 8-N to students
RESOLVE it using associative entity
S/W Requirements Analysis (SRA)

Data Model – Attribute Rules

**Key attributes** – originate in one entity migrate to other to support relationships

**Non Key attributes** – Never migrate

Primitive, Derived
Database Systems
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

Data Model – Attribute Rules

Must be Atomic represents a single fact about the entity

Optional attributes – Every instance must eventually have a value for each attribute

If you have only keys it is wrong
Database Systems
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)

Data Model – Attribute Rules

Primary Keys must be

Stable

Stock Symbol may change

Minimal – No Fat keys – Employee Id and SSN

Cannot be Inaccessible – bound by security
S/W Requirements Analysis (SRA)

Data Model – Attribute Rules

Primary Keys must be

Must be Definitive – Must always have a value SSN fails this check

Must not be optional

A primary key will never a non key attribute in another entity
Database Systems
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)
Data Model – Attribute Rules
Artificial Keys are permitted when

- No single natural attribute to satisfy PK rules
- Composite Candidate Key is very large
Database Systems
Data Model - Entity Relationship Diagram

S/W Requirements Analysis (SRA)
Data Model – Relationship rules

Must be supported by migration of PK
Eliminate Balanced 1 To 1 relationship

Usually the same thing; cannot exist without the other

Resolve Many to Many mandatory
ER Relationships

Strong Entity

Not dependent on other entities for existence.
Uniquely identifiable by Primary Key
Example – Depends on how the business views the data

Person
Department
Course
Term
ER Relationships

Strong Entity

Not dependent on other entities for existence.

Uniquely identifiable by Primary Key

Example – Depends on how the business views the data

Person
Department
Course
Term
ER Relationships

Strong Entity

<table>
<thead>
<tr>
<th>TERM</th>
<th>DEPARTMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP2015 (PK)</td>
<td>Code (22C)</td>
</tr>
<tr>
<td>Start Date</td>
<td>Name (CS)</td>
</tr>
</tbody>
</table>
Database Systems
Data Model - Entity Relationship Relationship Diagram

ER Relationships

Weak Entity
Dependent on other entities for existence.
Uniquely identifiable only by its association another entity.

Grades
Course
Student
Database Systems
Data Model - Entity Relationship Diagram

ER Relationships

Weak Entity

TERM
SP2015 (PK)
Start Date ….

DEPARTMENT
Dept Code (22C)
Name (CS)

SECTION
Code (EXV)
Name (WWW)

COURSE
Dept Code
Code (144)
Name (DB Systems)
ER Relationships

Department offers 1 to many Courses in a term
Student takes 1 to many classes

Multiplicity Constraints

Supported by PK migration

1 to 1
1 to Many
Many to Many
Database Systems
Data Model - Entity Relationship Diagram

ER Relationships

Relationships

Multiplicity Constraints

A Student may take a course in 0 to 1 section.

A Faculty may teach 0 to many courses

FT Student may take 9 to 18 credits
ER Relationships

Multiplicity Constraints - Cardinality

The maximum occurrence of an entity in a relation

In a 1 to 1 – Cardinality is 1
ER Relationships

Department may offer 1 to many Courses in a term

Student must take 1 to many classes each term

Multiplicity Constraints – Modality or Participation

Mandatory or Optional
MAY or MUST
ER Relationships

Balanced 1 to 1 Relationships

Same Cardinality / Optionality at both ends
Too closely related; One cannot exist without another

Perhaps they are one and the same entity

Example: Sales Manager has many sales reps and Sales territories. One and Only one Rep will be assigned to a Territory. Each rep will be assigned to only 1 Territory and Each territory will have only one assigned Sales rep.
Database Systems
Data Model - Entity Relationship Diagram

Balanced 1 to 1 Relationships

A
  key a (PK)
  key b (FK)
  A1
  A2

B
  key b (PK)
  key a (FK)
  B1
  B2

Mandatory on both ends 1:1 ➔ analysis / modeling error

Could be justified during design in very rare cases
ER Relationships

Balanced 1 to 1 Relationships

Same Cardinality / Optionality at both ends
- Might mask the actual relationship
- Introduces redundancy which can be avoided
- Hard to establish data dependency path
ER Relationships
Mandatory 1 to Optional 1

-Similar problems as in Balanced 1 to 1

-Also check if the optional side is a subtype of mandatory side

Emp
ER Relationships

Mandatory 1 to  Optional 1

Employee
id (PK)
name
Hire date
Mandatory end

Ex Employee
Id (PK)
Term date
(Optional end)

Analysis / modeling error
Resolve it as Employee and Emp Status
Database Systems
Data Model - Entity Relationship Diagram

ER Relationships
Many to Many $\rightarrow$ Non Specific
Hard to implement in data structures
Hard to detect data dependencies
Unstable Model

Resolve early into Associative Entity
Database Systems
Data Model - Entity Relationship Diagram

ER Relationships
Many to Many ➔ Non Specific

Example
Resolve early into Associative Entity
STUDENT (student id name.....)
COURSE (course id, title, desc, ....)
Roster of students?
So we need an Associative entity CLASS
ER Relationships

Associative Entities

Resolves Many to many

CLASS (student Id, course id, Section, Grade )
Research
Second Grade option
Withdrawal without grade penalty
ER Relationships

Associative Entities

Student

Course

Class
Section (EXV)
Student id, Course Id ……
ER Relationships

Circular relationship is not allowed

Cyclical relationship

Key migration problem.

Examples?

Organization (org id (pk), market id (fk))
Offering (offering id (pk), org id (fk))
Market (marketid (pk), offering id (fk))

Resolve it
Database Systems
Data Model - Entity Relationship Diagram

ER Relationships

Circular relationship resolved

Organization (org id (pk))
Offering (offering id (pk))
Market (marketid (pk))
Organization Offering (org id (pk), offering id (pk))
Market Offering (market id (pk), offering id (pk))
Org Market (marketid (pk), org id (pk))
ER Relationships

Triad relationship is not allowed

Redundant

Two dependency paths between same entities

Biz process problems are indicated

Examples?
ER Relationships

Recursive relationship

But that is real.

Key migration problem.

RDBMS extensions support it.

Hierarchical DB supports it naturally

Examples?

Bill of materials (cid1, cid2, level, .....)

Raman Aravamudhan
ER Relationships

Special Relationships

SuperType

All attributes of Super applies to every one of its subtypes but not vice versa.

Example: TransportVehicle

With sub types like

- Automobiles
- Aircrafts
- OnWaterTransport
ER Relationships

Special Relationships

SubType

Mutually exclusive with each other but share a common relationship with the supertype

Can have direct relationship with other entities which are not subtypes of the same supertype

Must have at least one non key attribute or a distinct relationship
ER Relationships
Special Relationships
Supertype SubType

Practice question

Write the attributes of Transport Vehicle and its subtypes’ attributes
Database Systems
Data Model - Entity Relationship Diagram
S/W Requirements Analysis (SRA)
Special Relationships

Inheritance

From OO

Class Model
Aggregation
“Has a“
“Is Part of“
Database Systems
Data Model - Entity Relationship Diagram

Extended ER Relationships

Special Relationships

Composition Special form of aggregation
WHOLE – PART
To be continued.

ER To Database Design
Problems with ER
Extended ER
S/W Requirements Analysis (SRA)

Data Dictionary
Organized Collection of all Data Items and their qualities

Case Tools
Round Trip engineering
S/W Requirements Analysis (SRA)

Benefits of Modeling

- Identify Missing requirements
- Identify redundant requirements
- Identify incompatible requirements
- Identify big processes and analyze them
- Mark potential component reuse
S/W Requirements Analysis (SRA)

Benefits of Modeling

- Process Flow
- Identify Events which cause no response
- Resolve many to many relationships
- Multiple views to the system
- Graphical representation; users like it
- Good documentation
Database Systems
Structured Query Language SQL

Textbook Notes

Connolly – Chapters 10, 11, 12, 13

Ullman – Chapter 4