6. Database Design*

6.1 Data Dependency and Normalization of Relational Schema

- Some dependent relations exist between attributes.
- Function dependency (FD): the most basic kind of data dependencies. The value of one or a group attributes can decide the value of other attributes.
  FD is the most important in general database design.
- Multi-valued Dependency (MVD): the value of some attribute can decide a group of values of some other attributes.
- Join Dependency (JD): the constraint of lossless join decomposition.

1NF

every attribute of a relation must be atomic.

| name | dept | address | | |
| | | prov | city | street |

Non 1NF

| name | dept | prov | city | street |

1NF

2NF

- R ∈ 1NF and no partially function dependency exists between attributes.

  S(S#, SNAME, AGE, ADDR, C#, GRADE)

  --- non 2NF

Problems of non 2NF:

- Insert abnormity: can not insert the students' information who have not selected course.
- Delete abnormity: if a student unselect all courses, his basic information is also lost.
- Hard to update: because of redundancy, it is hard to keep consistency when update.

Resolving：
  According to the rule of **"one fact in one place"** to decompose the relation into 2 new relations:
  S(S#, SNAME, AGE, ADDR)
  SC(S#, C#, GRADE)

3NF

- R ∈ 2NF and no transfer function dependency exists between attributes.

  EMP(EMP#, SAL_LEVEL, SALARY)

  --- non 3NF

1

Problems of non 3NF

- ✓ Insert abnormity: before the employees's sal_level are decided, the correspondence between sal_level and salary can not input.
- ✓ Delete abnormity: if some sal_level has only one man, the correspondence between sal_level and salary of this level will be lost when the man is deleted.
- ✓ Hard to update: because of redundancy, it is hard to keep consistency when update.

Resolving：

According to the rule of **"one fact in one place"** to decompose the relation into 2 new relations:

EMP(EMP#,SAL_LEVEL)

SAL(SAL_LEVEL,SALARY)

Material Card

How to define relations to express the information on this card?

Equipment name:		Type:		Code:				
Unit price:		Store place:		room	rack	layer	position	
date	voucher No.	coming/going place	take in	take out	balance	sum	remark	

5.2 ER Model and ER Diagram

- Concept model: entity—relation, be independent of practical DBMS.
- Legend:

	entity
	relation
	attribute

SNAME — student — S#

GRADE — elective

CNAME — course — C#
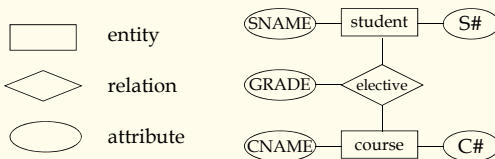
5.3 Database Design Method
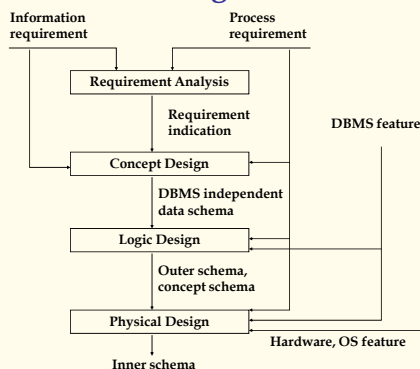
- **Procedure oriented method**
  This method takes business procedures as center, the database schema is designed basically in accordance directly with the vouchers, receipts, reports, etc. in business. Because of no detailed analysis on data and inner relationships between data, although it is fast at the beginning of the project, it is hard to ensure software quality and the system will be hard to fit future changes in requirement and environment. So this method is not suitable for the development of a large, complex system.
- **Data oriented method**
  This method design the database schema based on the detailed analysis on data and inner relationships between data which are involved in business procedures. It takes data as center, not procedures. It can not only fulfill the current requirements, but also some potential requirements. It is liable to fit future changes in requirement and environment. It is recommended in the development of large, complex systems.

Database Design Flow

Information requirement   Process requirement

Requirement Analysis

Requirement indication

DBMS feature

Concept Design

DBMS independent data schema

Logic Design

Outer schema, concept schema

Physical Design

Hardware, OS feature

Inner schema

Requirement Analysis

A very important part of system requirement analysis. In requirement analysis phase, the data dictionary and DFD (or UML) diagrams are the most important to database design.

- **Dictionary and DFD**
  - ➤ Name conflicts
    - ➤ Homonym(the same name with different meanings)
    - ➤ Synonym(the same meaning in different names)
  - ➤ Concept conflicts
  - ➤ Domain conflicts
- **About coding**
  - ➤ Standardization of information
  - ➤ Identifying entities
  - ➤ Compressing information
- **Through requirement analysis, all information must be with unique source and unique responsibility.**

Concept Design

Based on data dictionary and DFD, analyze and classify the data in data dictionary, and refer to the processing requirement reflected in DFD, identify entities, attributes, and relationships between entities. Then we can get concept schema of the database.

- Identify Entities
- Define the relationships between entities
- Draw ER diagram and discuss it with user
  - It is proposed to use ER design tools such as ERWin, Rose, etc.

Logic Design

According to the entities and relationships in ER diagram, define tables and views in target DBMS. Basic standard is 3NF.

- Translate entities and relationships in ER diagram to tables
- Naming rule of table and attribute
- Define the type and domain of every attribute
- Suitable denormalization
- Necessary view
- Consider the tables in legacy system
- Interface tables

Physical Design

For relational database, the main task in this phase is to consider creating necessary indexes according to the processing requirements, including single attribute indexes, multi attributes indexes, cluster indexes, etc. Generally, the attribute often as query conditions should have index.

Other problems:

- Partition design
- Stored procedure
- Trigger
- Integrity constraints

Remarks

- 仅仅在结构上达到3NF (BCNF) 是不够的。
- "一事一地"包括每项信息的唯一，要提取出问题的本质，识别出本质上同一概念的信息项。
- 对于表达类似信息，模式相似只是取值不同的表，应尽量合并。如学习经历、进修经历；奖励信息、惩处信息等。
- 考虑到效率、用途等因素，该分开的表还应分开。如本科生基本信息和研究生基本信息。
- 结合DBMS内部实现技术，合理设计索引和文件结构，为查询优化准备好存取路径。
- 在结构规范化、减少数据冗余和提高数据库访问性能之间仔细权衡，适当折中。
  数据库设计实例分析