数据库笔记

学习建模，sql语言，编程，数据库实现

group project要求：

DB related

检查（15周）

proposal

DB design

final(presentation)

1、实现可增删改查的数据库系统

2、实现小型数据库系统，基于C-store

http://db.csail.mit.edu/projects/cstore

3、其他（需要与老师商量）

需要交：ppt，视频，文档，代码，小组互评表

lecture 2：E/R 模型（实体/联系 模型）

数据库开发流程：

了解世界（需求分析）-> 实体/联系设计 -> 关系数据模型（变成计算机可以理解的形式）-> 创建数据库

E/R 模型是实现需求分析-> 实体/联系设计的工具

先将问题图形化，再表格化

包含四个元素：实体(entity)、实体集(entity set)（里面的实体要有相同属性，对应实实在在的对象）、属性(attritibute)、联系(relationships)（连接两个或多个实体集）

关键属性：每个实体集里面都有至少一个关键属性（可能是多个属性组成的集合），在此属性下每个实体各不相同（例如：ID）

因为DB的核心功能是找到数据

主键(primary key)：可以作为关键属性的最简形态称为候选键（缺一不可）(candidate key)，而在里面有一个最主要的关键属性（一般选里面比较简单的），称为主键

超码/关键字(super key/key)：所有能成为关键属性的属性及其集合（例如：编号+…，编号+姓名+地址+…）（可能删掉其中一两个属性还是能成为关键属性）

super key > candidate key -> primary key

外键(foreign key)：另外一个实体集的主键

集合论：集合(set)：没有结构，没有顺序，没有冗余

subset，superset，proper subset

cross product(Cartesian product)：笛卡尔积（X x Y = {(x, y), x in X, y in Y}，注意是小括号，说明有顺序，里面包含了X中元素和Y中元素的所有关系）

设计准则：如实正确，避免冗余，简单性，联系选择，实体选择

如实正确：如实反映现实特征

避免冗余：是否有相同属性，是否有某一属性的数据严重重复，是否有成环的多对多联系，是否有由相互独立的属性而构成的多对一实体集...

弱实体集：依赖于强实体集才可唯一确定的实体集（关键属性依赖于其他实体集）

弱实体集和强实体集之间的支撑联系必须是多对唯一的

约束：必须保证的规则

种类：

关键字约束：具备唯一性

单值约束：具备唯一性且可能为空值

参照完整性：不允许出现空值

lecture 3：RM（关系模型）

元组：属性的有序集合

关系模式：关系名称 + 属性名称 + 属性类型

数据库：关系模式组成的集合

从ER图到关系模型

1、应用半机械化的规则把ER转为关系模型

2、通过组合一些关系来优化设计

3、规范化设计（符合一定的范式）

强实体集转换为表格

实体集的关键字即为表格关键字

实体集里的属性即为表格中的属性

弱实体集转换为表格

弱实体集所依赖的强实体集的关键字和弱实体集的关键字的组合即为表格关键字

弱实体集里的属性即为表格中属性

关系转为表格

由相互关联的实体集的特点从他们的关键字中确定关键字

关系里的属性即为表格中的属性

弱实体集和对应唯一确定的强实体集间的联系（即弱联系）不需要变成表格

而弱实体集和强实体集间的强联系变成表格时还需要加上此弱实体集对应唯一确定的强实体集的关键字

组合关系以优化设计

将相同关键字的表格合并在一起

多对多的关系一般不合并，否则表格中可能出现重复冗余

lecture 4：FD（函数依赖）

函数决定（函数依赖）：属性A函数决定属性B（记为：A -> B）说明相同的A必定对应相同的B（一个自变量只能对应一个因变量）

函数决定具备下列性质（Armstrong公理）

传递性（A -> B, B -> C 可推出 A -> C）

结合性（A -> {B1, B2} 可推出 A -> B1 && A -> B2）

自反性（A可以函数决定其本身和它的子集）

增广率（A -> B 可推出 {A, C} -> {B, C}）

平凡（trivial）：B为A的子集

非平凡（nontrivial）：B中有属性不为A的子集

完全非平凡：B中全部属性均不为A的子集

通过函数决定定义超码（关键字）：超码是可以函数决定所有属性的（因为超码具有唯一性）

通过函数决定定义候选键：候选键是可以函数决定所有属性的，但去掉里面任一属性就不可函数决定所有属性

如何得到一个更好的函数决定?

1、增广去除　　2、尽量完全非平凡

方法：闭包算法

闭包：A的闭包为A所有可以决定的属性的集合

算法：反复将A可以决定的属性放进A集合里面直至A集合不变，此时的A集合即为A的闭包

如果一个集合A的闭包为所有属性的集合，那么集合A就是超码（关键字）

如果一个集合A的闭包为所有属性的最小集合，那么集合A就是候选码

lecture 5：BCNF（BC范式） 与 3NF（3范式）

冗余(redundancy)：数据不必要的重复，会导致以下异常(anomalies)

更新异常：只更新了一部分的信息使得信息不同步

删除异常：删除一些信息导致其他信息的丢失

插入异常：插入时必须先插其他位置的信息

解决冗余的方法

分解法：将表格中可由某些子属性所决定的属性拆出来成为新表

分解方法：投影法（将子属性拆分）+ 去重

要求：属性不丢失，要有公共的属性（不能是独立的两张表）

分解后通过连接操作如果可以还原回原来的表格即可说明分解有效

要求：分解过后两张表格的公共属性要是关键字

高级设计流程

实际问题 -> ER模型 -> 关系模型（表格）-> 查看函数决定 -> 去除冗余（异常）

范式

第一范式：所有属性原子性

BCNF：如果A -> B是非平凡的函数决定，那么A就是关键字

如果表格不符合BC范式，那么就要将表格进行分解

利用函数决定的结合性进行分解

分解过后的表格如果仍不符合BC范式则继续分解