DB上

- **Introduction**
  - 关键概念
    - data：包含信息
    - database： 数据和元数据的集合
    - DBMS： 允许用户对数据库进行 定义，创建，维护，和访问控制的软件系统
    - data dependence：应用程序和数据相互独立
  - 文件系统的限制
    - 数据的分离和孤立
    - 数据的冗余
    - 程序数据依赖性
    - 文件格式不相容
    - 查询一成不变
  - 数据库方法的优势
    - 数据共享
    - 少量的冗余
      - 数据一致性
    - 程序数据独立性
    - 增强的数据完整性
      - 有效性
      - 一致性
    - 安全性
  - 数据库方法的劣势
    - 复杂
    - 费用高
    - 性能低
  - DBMS的环境
    - 硬件

- 软件
- 数据
- 过程
- 人

- **数据库环境**
  - ANSI-SPARC 三层体系结构/三级模式两级映射
    - 外部层：用户视角
      - 外部模式（schema？）
    - 概念层：逻辑表示
      - 概念模式
        - 实体
        - 属性
        - 联系
        - 完整性约束
    - 内部层：物理存储
      - 内部模式
    - 数据独立性
      - 逻辑数据独立性：外部模式不受概念模式变化的影响
      - 物理数据独立性：概念模式不受内部模式变化的影响
  - 数据模型（model）
    - 三个组件
      - 结构部分
      - 操纵部分
      - 一组完整性约束
    - 三类模型
      - 基于对象的
      - 基于记录的
      - 物理的
    - 定义：一组集成的概念，描述和操纵组织机构内的
      - 数据

- 数据之间的联系
- 数据的约束

- 系统目录
  - 描述数据库中数据的信息库，存储关于数据的数据（元数据）
  - 优点
    - 访问控制
    - 保持数据的完整性
    - 保持数据的可扩展性
- 文件服务器结构
  - 三个缺点
    - 网络堵塞
    - 内存浪费，每个工作站都要有一个DBMS副本
    - 并发、恢复的完整性控制复杂
- 客户服务器结构
  - 优点
    - 广泛支持数据库的访问
    - 性能高
    - 费用低，硬件/通信费用低
    - 增强一致性
  - 定义
    - 软组件相互作用形成系统的方式
    - 客户端请求资源，服务端提供资源

关系模型

- 结构
  - 关系
  - 属性
  - 域
  - 元组
  - 关系模式
- 性质

- 由关系名且不重复
- 单元格包含一个确切的原子值
- 属性有不同的名字
- 同一属性的各个值都取自相同的域
- 属性的顺序不重要
- 元组互不相同
- 元组的顺序不重要

- 关键字
  - 超关键字 - 唯一标识
  - 候选关键字 - 最小属性集合
  - 主关键字 - 候选关键字之一
  - 可替换关键字 - 不是PK的候选关键字
  - 外部关键字 - 与某个表的候选关键字匹配

- **关系代数**
  - 基本运算
    - 选择 - ∧ ∨ ~ 与或非
    - 投影 - 去掉了重复元组
    - 笛卡尔乘积
    - 集合并
    - 集合差
  - 其他运算
    - 交：R 交 S = R - （R - S)
    - 连接运算 = 选择加笛卡尔积
      - seita 连接 - 一般的连接
      - 等接 - 谓词全是等号
      - 自然连接 - 公共属性上的等接
      - 外连接 - 结果中：不封口的那一边 所有元组都保留，公共属性匹配的那些元组，来自另一个表的属性填空
      - 半连接 - 挑选三角形竖边的部分元组
    - 除法

- R ÷ S： 与S每个元组的组合都能在R中找到 -> 笛卡尔乘积的逆运算
- 如：列出查看过三居室房产的客户

SQL

DML

SELECT：

SELECT [DISTINCT | ALL]attrs or funcs
FROM tables
WHERE conditions
GRUOP BY attr [HAVING condition]
ORDER BY attr [DESC]

- where
  - = > < <>(不等于)
  - AND OR NOT
  - (NOT)BETWEEN ... AND ...
  - (NOT)IN(...,..)
  - (NOT)LIKE
    - % 通配符
    - _ 单个字符
    - 转义字符 ESCAPE
  - IS (NOT) NULL
- order by
  - 默认升序
  - DESC降序
  - DESC关键字只能管前面的一个列
  - 是select语句的最后一个子句
- 聚集函数
  - COUNT / SUM / AVG / MIN / MAX
  - 限制
    - 单列操作，返回单值
    - COUNT / MAX / MIN 可用于数值和非数值，SUM/AVG 只能用于数值
    - 除去COUNT（*），除掉空值，计算非空值

- 只能用于SELET和HAVING中
- SELECT中用了聚集函数但没用group by，那只能应用聚集函数中的参数列

- group by
  - 按指定列进行分组，每组产生一个查询结果
  - 先使用where子句
  - 空值被认为是相等的
- having和where
  - where对单个行进行过滤，having对分组进行过滤
  - having子句的列名必须出现在 group by中或在聚集函数里
  - having可以出现聚集函数 where不可以
- 子查询
  - where 中可使用子查询，查询结果返回一组数据
  - 将聚集函数放在子查询里，可以破除where不能包含聚集函数的限制
  - 子查询必须出现在比较表达式的右边
  - SOME 至少一个 ALL 任何
- 多表查询
  - from 列的多个表=表的笛卡尔积
  - 给每个表起别名，从select开始就要体现。
    select c.cNO from Client c, Viewing v
  - from 多个表的后面可以用 X join Y on C表示连接条件 X join Y using A 表示等接 X natural join Y 表示自然连接
  - (NOT)EXISTS 表示子查询的结构存在吗

- INSERT
  - INSERT INTO table [(.....)]
    VALUES(.....)
- UPDATE
  - UPDATE table
    SET col1 = val1,.....
    WHERE condition
- DELETE

- DELETE FROM table
  where condition

- DDL
  - 完整性增强特性(Integrity Enhancement Feature)
    - 必须有值的数据：某些列不允许为空
      - NOT NULL
    - 域约束：合法值的集合
      - CHECK (condition)
      - CREATE DOMAIN name AS datatype
        DEFAULT default_val
        CHECK (condition)
      - DROP DOMAIN name [RESTRICT | CASCADE]
    - 实体完整性：主关键字 唯一非空
      - PRIMARY KEY(attrs)
      - UNIQUE(attrs)
    - 引用完整性：外键 是 父表 已存在的有效元组 或 空值
      - FOREIGN KEY (attrs) REFFERENCES table
      - 删除父表的对应元组 ON DELETE/UPDATE CASCADE/.....
        - CASCADE：自动删除子表匹配的行
        - SET NULL : 设置子表的外部关键字为NULL
        - SET DEFAULT ： 设置子表的外部关键字为默认值
        - NO ACTION :拒绝父表的删除操作
    - 一般性约束
      - 企业规则的约束
  - 数据定义
    - 创建表
      - CREATE TABLE name (
        col_name   domain   NOT NULL,
        ......,
        PRIMARY KEY(),
        FOREIGN KEY () REFFERENCES table
        );

- 视图
  - 视图的定义
    - 对一个或多个基关系进行关系操作得到的 动态结果
    - 无需存储在数据库中，用户使用时动态生成的虚关系
  - 创建视图
    - CREATE VIEW name [(cols...)]
      AS subselect
      [WITH CHECK OPTION]
  - 视图分解：处理对视图进行的查询
    - 包括聚集函数都不行，不管是查询语句里还是视图定义里
  - 视图的可更新性
    - 没有指定DISTINCT，即重复元组未从查询结果中消除
    - SELECT列表的每个元素均为列名且列名出现的次数不多于一次
    - FROM子句只能指定一个表即视图只有一个源表，且用户对该视图有要求的权限
    - WHERE子句不能包括引用了from子句中的表的子查询
    - 查询中不能有 GROUP BY 或 HAVING 子句
  - 视图的优缺点
    - 优点
      - 数据独立性
      - 数据的完整性
      - 提高安全性
      - 方便，用户化
    - 缺点
      - 更新局限性
      - 结构局限性
      - 性能开销
  - 视图物化：
    把第一次访问视图的结果存储为数据库的临时表
- SQL的访问控制机制
  - SQL创建的对象有一个所有者

- 所有者可以指定或撤销用户的访问权限

数据库开发的生命周期

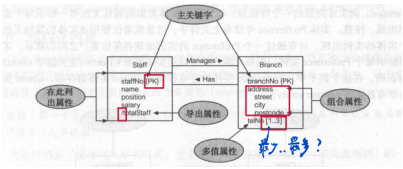
- 生命周期的主要阶段
  - 数据库规划
  - 系统定义
  - 需求收集与分析
  - 数据库设计
    - 概念数据库设计
      - 建立概念数据模型
    - 逻辑数据库设计
      - 建立逻辑数据模型
    - 物理数据库设计
      - 描述在辅助存储器上如何实现数据库
  - 应用程序设计
  - 实现
  - 数据转换与加载
  - 测试和运行维护

实体联系建模

- 实体类型
  - 定义：被认可的，独立存在的 一组具有相同属性的对象
  - 实体存在的分类
    - 物理存在
    - 概念存在
  - 分类
    - 强实体类型：不依赖于其他实体类型
    - 弱实体类型：依赖其他实体类型
- 联系类型
  - 定义： 一组有意义的关联
  - 联系类型的度/分类 一元二元三元四元
  - 联系的属性

- 联系也有属性
- 用虚线表示

- 属性



- 定义：实体或联系类型的某一特性
- 属性域
- 分类
  - 简单属性
  - 组合属性
  - 单值属性
  - 多值属性
  - 导出属性derived
  - 关键字
    - 候选关键字
    - 主关键字
    - 合成关键字： 包含两个或以上属性的候选关键字

- 结构化约束
  - 多重性multiplicity constraint 在联系中的类型
    - 一对一的联系
    - 多对多的联系
    - 一对多的联系
    - 靠近实体的两个数字 x..y
      - 左边的代表最小值，对面拿出一个，这边最少要拿出几个?
      - 右边的代表最大值，对面拿出一个，这边最多能拿出几个
      - 几对几看的是右边的数字
    - 定义
      - 一个实体类型
      - 通过某一联系

- 与另一实体类型的某个（1个）出现发生关联
- 的出现的数目或范围

- 企业约束
  - 定义：数据库中数据遵守的规则
  - 多重性描述：多重性限制了实体的关联方式

- ER图转化为表
  - Strong entity： Create relation that includes all simple attributes.

    Weak entity： Create relation that includes all simple attributes （primary key still has to be identified after the relationship with each owner entity has been mapped).

    1:* binary relationship： Post primary key of entity on one side to act as foreign key in relation representing entity on many side.Any attributes of relationship are also posted to many side.

    1:1 binary relationship:
    Mandatory participation on both sides： Combine entities into one relation.
    Mandatory participation on one side： Post primary key of entity on optional side to act as foreign key in relation representing entity on mandatory side.
    Optional participation on both sides：Arbitrary without further information.

    ： binary relationship, complex relationship：Create a relation to represent the relationship and include any attributes of the relationship.Post a copy of the primary keys from each of the owner entities into the new relation to act as foreign keys.

    Multi-valued attribute：Create a relation to represent the multi-valued attribute and post a copy of the primary key of the owner entity into the new relation to act as a foreign key.

规范化

- 数据的冗余会带来更新异常
  - 插入异常
  - 删除异常
  - 修改异常

- 函数依赖
  - 定义：描述属性之间的联系
  - 类似于函数的映射关系 x->y，一个x只有一个y与之对应

- 完全函数依赖： 被依赖（决定方）的是最小集
- 部分函数依赖
- 传递函数依赖 ： a推b，b推c则a推c
- 识别函数依赖
  - 逻辑推理
  - 当某一列的取值相同的时，看看其他列的取值情况是否也是如此（同样的x对应一个y/但同样的y不应顶对应同样的x）
- 利用函数依赖确定主关键字
  - 所有不属于主关键字的属性都依赖于主关键字
- 最小函数依赖集
  - 右边只包含单个属性
  - 完全函数依赖
  - 每个函数依赖都不重复
- 规范化的过程
  - 先画出所有的函数依赖再说
  - 非范式 -> 1NF
    - 行和列有且仅有一个值
    - 新建行：确认主关键字和重复数据，复制非重复数据数据
  - 1NF -> 2NF
    - 非主关键字属性 完全函数依赖与 主关键字
  - 2NF -> 3NF
    - 非主关键字属性 直接依赖(不传递依赖)于主关键字
  - 3NF->BCNF
    - 每个函数依赖的决定方都是候选关键字

事务管理

- 事务支持
  - 定义
    - 一个或一系列动作
    - 单个用户或程序发出的
    - 从数据库中读或者修改内容

- 意义
  - 逻辑单元
  - 从一个一致性状态转移到另一个一致性状态，但在整个过程中有不一致的情况发生
- 状态
  - 成功 -提交
  - 失败 - 撤销或者回滚
  - 事务一旦提交就不可回滚，只能够进行一个相逆的事务来抵消
- 性质
  - 原子性
  - 一致性
  - 隔离性
  - 持久性

- 并发控制
  - 定义
    - 同时发生
    - 互不打扰
  - 问题
    - 丢失更新
    - 未提交依赖 / 脏读
    - 不一致分析 / 不可重复读
  - 可串行性与可恢复性
    - 调度- 一系列操作
    - 串行调度 - 没有重叠
    - 非串行调度 - 有重叠
    - 可串行性：非串行调度的小工和串行调度的效果一样
    - 可恢复性：某事务在另一个事务写之后进行读，那么该事务的提交应该在另一个事务之后
  - 并发控制的方法
    - 锁方法

- 两段锁协议2PL
  - 所有的上锁在解锁之后
  - 会有级联回滚的问题 -> 严格2PL： 在事务结束时才释放锁
- 死锁
  - 持有且等待
  - 只有撤销某个/些事务，三种方法
    - 超时
    - 死锁预防： 使用时间戳方法
    - 死锁检测：wait-for图（WFG）
- 共享锁
- 互斥锁

- 时间戳方法
  - 时间戳：记录事务的相对开始时间
  - 读或写操作只有在该数据项的最新一次修改是更老的事务的时候才被允许，否则就回滚并设置新的时间戳
  - 写时间戳和读时间戳
    - 写操作要和读写时间戳比较
    - 读操作要和写时间戳比较
  - 托马斯写规则/忽略过时写：
    不用回滚老的写操作

- 数据项的粒度
  - 被选作未保护单位的大小
  - 文件-页-记录-字段
  - 平衡
    - 更大的粒度：更小的并行性
    - 更小的粒度：更多的锁信息
  - 层次结构
  - 意向锁（intention）
    - 意向互斥锁：有一部分上了互斥锁
    - 意向共享锁：有一部分上了共享锁

- 共享意向互斥锁：剩下全部上共享锁，除了有一部分上了互斥锁（只能与IS共存）

- 数据库恢复
  - 定义
    - 从故障中恢复到正确状态
  - 可靠性
    - 定义
      - 应对故障的适应能力
      - 能够从失败中恢复
    - 失败的原因
      - 系统崩溃
      - 介质故障
      - 软件错误
      - 自然灾害
      - 疏忽与蓄意破坏
  - 事务时恢复的基本单位
  - 两种应对措施
    - 重做/前滚：当事务已经提交时
    - 撤销/回滚：当事务未提交时
  - 恢复机制
    - 备份
    - 日志
    - 检查点
    - 回复管理器
  - 恢复技术 （数据库并未物理损坏，只是处于不一致状态）
    - 延迟修改
      - 事务提交以后修改结果才写到数据库中
      - 重做已提交的事务即可
    - 立即修改
      - 更新一旦发生就更新到数据库中，不需要等待提交

- 撤销未完成事务的影响
- 影像页
  - 两张页表，当前页用来修改，影像页用来备份。一旦修改完成，当前页变成影像页
  - 定期回收无用的单元

查询处理

- 定义：从数据库中检索数据
- 目标
  - 将高级语言正确有效的转换为低级语言
  - 执行计划
- 查询优化
  - 目标
    - 选择一个最有效的执行计划
    - 最少的资源利用
- 查询处理的四个阶段
  - 解析
    - 分析
      - 关系或属性是否存在
      - 操作是否正确
      - 查询树：从下往上到查询目标
        - 叶子节点是原表
        - 节点代表关系代数操作
    - 规范化
    - 语义分析
    - 化简
    - 查询重构
  - 优化
    - 两个方法
      - 启发式方法
        - 策略

- 尽可能早的使用选择（先）和投影
- 笛卡尔积和选择合并为连接
- 利用结合律对节点进行重新排序，产生较小结果的优先执行
- 只计算一次公共表达式，如视图

- 比较并选择

- 代码生成
- 执行

- 查询估算所需的统计信息
  - 关系
    - 元组数nTuples
    - 块因子bFactor
    - 块数nBlocks
  - 属性
    - 不同取值的个数 nDistinct
    - 最值 min max
    - 选择基数SC ： 满足某个等值条件的平均元组数
  - 索引
    - 索引的级数
    - 索引中叶的块数

- **数据库安全**
  - 目标
    - 保护数据库抵御有意或无意的威胁
  - 范围（scope）
    - 盗用和冒用
    - 破坏机密性：对组织而言
    - 破坏隐私：个人隐私
    - 破坏完整性：无效或毁坏的数据
    - 破坏可用性：数据或系统无法访问
  - 安全控制
    - 授权：授予一个主体合法访问的权力

- 视图：一种虚关系，保护数据库
- 备份和恢复：周期性的将数据库和日志文件复制到脱机的存储介质中
- 完整性:完整性约束，防止非法数据的生成
- 加密：没有解密密钥的程序无法读取数据
- RAID技术：多个独立磁盘构成的大型磁盘阵列