

● zdc 内容及重点总结

基本内容

1. 为什么要学习数据库系统?
2. 什么是数据库?
3. 什么是数据库系统?
4. 什么是数据库管理系统?
5. 本课程学什么以及学到什么程度?

重点与难点

- 一组概念的区分：数据库、数据库系统和数据库管理系统?
- 熟悉“表”的相关要素及术语
- 熟悉数据库系统的构成(工作环境)
- 了解数据库管理系统的功能：从用户角度和从系统角度

基本内容

1. 数据库系统的标准结构?
2. 数据模型?
3. 数据库系统的演变与发展?

重点与难点

- 一组概念的区分：三级模式两层映像，物理独立性和逻辑独立性
- 一组概念的区分：数据→模式→数据模型
- 几种数据模型的差异：网状/层次模型→关系模型→OO数据模型

基本内容

1. 关系模型概述?
2. 什么是关系?
3. 关系模型中的完整性约束

重点与难点

- 一组概念的区分：围绕关系的相关概念，如域、笛卡尔积，关系，关系模式，关键字/键/码，外码/外键，主码/主键，主属性与非主属性。
- 三个完整性：实体完整性，参照完整性和用户自定义的完整性；

基本内容

1. 关系代数之基本操作
2. 关系代数之扩展操作
3. 关系代数之组合与应用训练
4. 关系代数之复杂扩展操作(选学)

重点与难点

- 关系代数基本操作：并、差、积、选择、投影、(更名)。
- 关系代数扩展操作：交、 θ -连接、自然连接。
- 关系代数复杂扩展操作：除、外连接
- 书写关系代数的基本思维训练：“一个集合，施加一个操作得到一个集合，依次施加关系代数操作，进而得到所需结果”“以集合为中心”

基本内容

1. 关系演算之关系元组演算
2. 关系演算之关系域演算
3. 关系演算之安全性
4. 关于三种关系运算的一些观点

重点与难点

- 关系元组演算公式的递归定义；关系域演算公式的递归定义
- 关系元组演算公式：与 \wedge 、或 \vee 、非 \neg 、存在量词 \exists 、全称量词 \forall
- 用关系元组演算公式表达查询的思维训练
- 用QBE语言表达查询的思维训练
- 关系元组演算、域演算和关系代数在表达查询方面的思维差异

基本内容

1. SQL语言概述?
2. SQL语言之DDL-定义数据库
3. SQL语言之DML-操纵数据库

重点与难点

- SQL-DDL的基本语句：CREATE DATABASE, CREATE TABLE
- SQL-DML的基本语句：INSERT, DELETE, UPDATE, SELECT
- SQL-SELECT语句的训练：正确表达各种查询需求

基本内容

1. SQL语言之子查询运用
2. SQL语言之结果计算与聚集函数
3. SQL语言之分组查询与分组过滤
4. 利用SQL语言实现关系代数操作
5. SQL语言之视图及其应用

重点与难点

- SQL-SELECT : **IN | NOT IN, θ some, θ all, Exists | NOT Exists**
- SQL-SELECT : **聚集函数, GROUP BY, HAVING**
- 视图及其应用

基本内容

1. 数据库完整性的概念及分类
2. SQL语言之列约束与表约束—静态约束
3. SQL语言之触发器—动态约束
4. 数据库安全性的概念及分类
5. SQL语言之安全性实现

重点与难点

- 数据库完整性的概念, 完整性规则, 静态约束, 动态约束(触发器)
- 数据库安全性的概念, 安全性访问规则, 权利与授权

基本内容

1. 嵌入式SQL语言概述
2. 变量声明与数据库连接
3. 数据集与游标
4. 可滚动游标与数据库的增删改
5. 状态捕获及错误处理机制

重点与难点

- 数据库语言嵌入到高级语言中使用需要解决的问题—过程及其思维
- 怎样在高级语言中处理数据集—游标的使用技巧
- 错误捕获机制—设置错误陷阱与SQLCA的作用与使用
- 事务的概念—保证数据正确性的机制

基本内容

1. 动态SQL的概念和作用
2. SQL语句的动态构造
3. 动态SQL语句的执行方式
4. 数据字典与SQLDA
5. ODBC/JDBC简介?

重点与难点

- 熟练掌握SQL语句的动态构造技巧
- 了解数据字典的作用，掌握其使用技巧
- 理解ODBC/JDBC的工作原理

基本内容

1. 为什么要数据建模和数据库设计?
2. E-R模型--数据建模之基本思想
3. E-R模型--表达方法之Chen方法
4. E-R模型--表达方法之Crow's foot方法
5. 数据建模之案例讲解
6. 数据库设计中的抽象

重点与难点

- 理解E-R模型进行数据建模的基本思想，熟练掌握E-R模型的Crow's foot表达方法
- 熟练运用Crow's foot方法进行数据建模，即建模训练

基本内容

1. IDEF1x两种实体的区分
2. IDEF1x的标定联系与非标定联系
3. IDEF1x的不确定联系
4. IDEF1x的分类联系
5. IDEF1x建模之案例讲解
6. IDEF1x建模之案例作业点评

重点与难点

- 理解E-R模型进行数据建模的思想，熟练掌握E-R模型的IDEF1x表达方法
- 熟练运用IDEF1x方法进行数据建模，即建模训练
- 强化案例的学习与理解

基本内容

1. 数据库设计过程与设计方法
2. E-R图/IDEF1X向关系模式的转换
3. 不正确数据库设计引发的问题及其解决

重点与难点

- 理解数据库设计的四个过程
- 理解不正确数据库设计引发的问题，为数据库理论的学习奠定问题基础
- 理解不正确数据库设计引发的问题，提升数据建模与数据库设计能力

基本内容

1. 函数依赖
2. 完全函数依赖与传递函数依赖
3. 关于函数依赖的公理和定理
4. 函数依赖集的最小覆盖

重点与难点

- 一组概念：函数依赖、部分函数依赖和完全函数依赖、传递函数依赖、候选键、非主属性、逻辑蕴涵、闭包、属性闭包、覆盖、最小覆盖等
- 关于函数依赖的公理和定理，相关的证明
- 求属性闭包的算法、求最小覆盖的算法

基本内容

1. 关系的第1NF和第2NF
2. 关系的第3NF和Boyce-Codd NF
3. 多值依赖及其公理定理
4. 关系的第4NF

重点与难点

- 一组概念：1NF, 2NF, 3NF, BCNF, 4NF；多值依赖
- 熟练应用数据库设计的规范化形式，判断数据库设计的正确性及可能存在的问题

基本内容

1. 模式分解存在什么问题
2. 无损连接分解及其检验算法
3. 保持依赖分解及其检验算法
4. 关系模式无损连接或保持依赖的分解算法
5. 数据库设计需要知道的?

重点与难点

- 理解模式分解存在的问题是什么，怎样解决
- 两个概念：无损连接分解和保持依赖分解
- 关系模式如何进行无损连接或保持依赖地分解
- 五个算法：2个检验算法，3个分解算法(无损连接分解，保持依赖分解，既无损连接又保持依赖分解)

基本内容

1. 基础回顾-计算机系统的存储体系
2. 磁盘的结构与特性
3. DBMS数据存储与查询实现的基本思想
4. 数据库之表和记录与磁盘块的映射
5. 数据库之文件组织方法?

重点与难点

- 理解利用磁盘组织大规模数据的基本思维
- 初步了解数据存储与查询实现的基本思想
- 理解三种文件组织方法及其特性：堆文件、顺序文件和散列文件
- 理解数据库重组的概念和作用

基本内容

1. 为什么需要索引与什么是索引
2. 索引的简单分类
3. B+树索引
4. 散列索引

重点与难点

- 理解索引的作用，掌握应用索引改进数据库查询性能的方法
- 理解不同类型索引的概念：稠密索引与稀疏索引，主索引与辅助索引，聚簇索引与非聚簇索引，倒排索引，多级索引等
- 理解B+树索引，怎样建立、维护和利用B+树索引(算法层面)
- 理解散列索引，包括静态散列索引与动态散列索引(算法层面)

基本内容

1. 数据库查询实现算法概述
2. 以连接操作为例看逻辑实现算法与物理实现算法
3. 利用迭代器构造查询实现算法
4. 几个关系操作的一趟扫描算法
5. 基于索引的查询实现算法?

重点与难点

- 理解数据库查询实现的基本思想--逻辑算法和物理算法
- 理解查询实现算法与内外存环境的关系--如何利用内存
- 从物理存储上理解关系运算：一趟扫描算法
- 掌握关系运算的几个一趟扫描算法及其应用条件与算法复杂性

基本内容

1. 为什么需要两趟算法暨两趟算法的基本思想
2. 两阶段多路归并排序算法
3. 基于排序的两趟扫描算法
4. 基于散列的两趟扫描算法

重点与难点

- 理解两趟算法的基本思想
- 理解两阶段多路归并排序算法，进一步理解基于排序的两趟扫描算法
- 理解散列算法的核心思想，进一步理解基于散列的两趟扫描算法
- 关系代数操作的两趟扫描算法实现

基本内容

1. 为什么要及什么是查询优化?
2. 查询优化的基本思路
3. 逻辑查询优化
4. 物理查询优化

重点与难点

- 理解查询优化的整体思路是什么?
- 理解并掌握基于关系代数进行逻辑查询优化的方法和原则
- 理解物理查询优化中的代价估算方法

基本内容

1. 为什么需要并发控制
2. 事务调度及可串行性
3. 基于封锁的并发控制方法
4. 基于时间戳的并发控制方法
5. 基于有效性确认的并发控制方法?

重点与难点

- 理解数据库并发操作的三种不一致性及其产生原因
- 理解一组概念：事务、事务调度、可串行性、时间戳等
- 掌握三种类型的并发控制方法：基于封锁的方法、基于时间戳的方法、基于有效性确认的方法
- 重点掌握：冲突可串行性判别算法，两段封锁法，基于时间戳的方法；

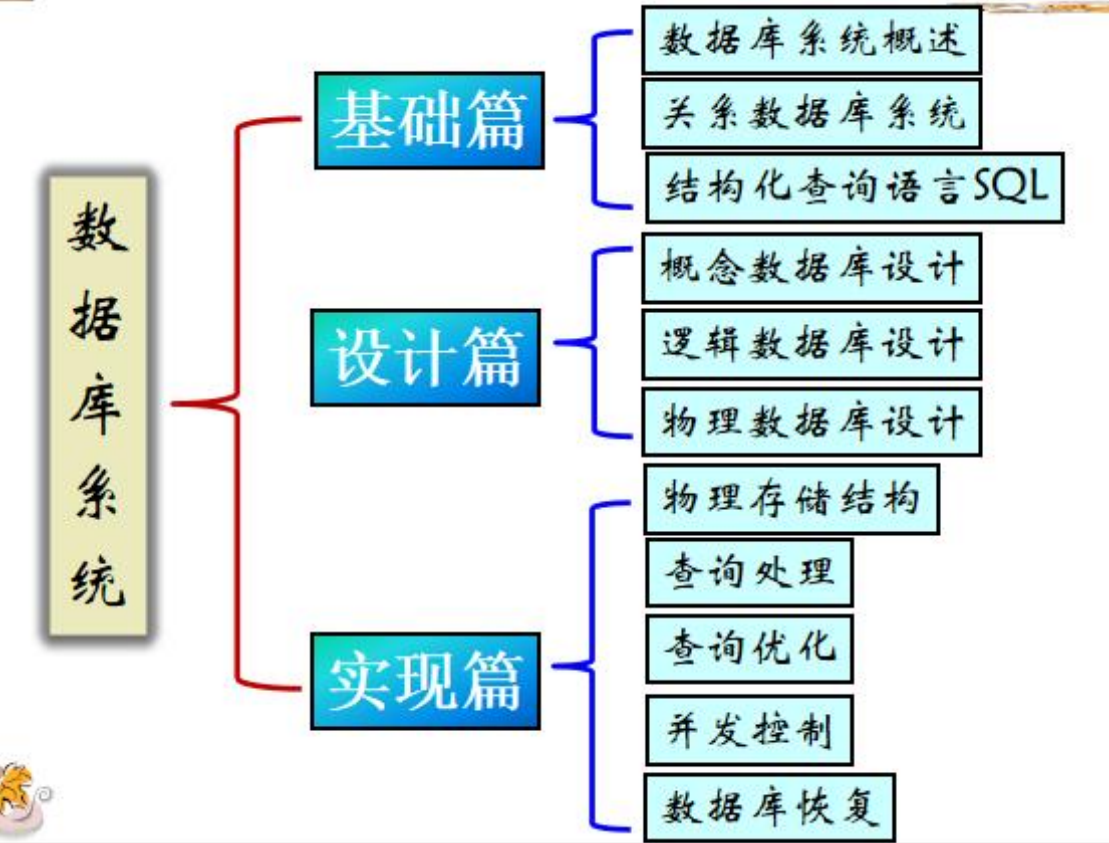
基本内容

1. 数据库故障恢复的宏观思路
2. 运行日志及其检查点
3. 三种类型的运行日志
4. 利用运行日志进行故障恢复

重点与难点

- 理解三种类型的故障：事务故障、系统故障和介质故障
- 三种类型故障的恢复手段：运行日志和副本
- 理解检查点的作用
- 理解三种类型的运行日志及其故障恢复的操作方法：Undo型日志，Redo型日志，Undo/Redo型日志

● zh+csy 内容及重点总结



章节	csy_zh 重点	zzn 重点
数据库系统概述	<p>数据、数据库、数据库管理系统、数据抽象与数据模型，层次模型、网状模型</p> <p>重点：</p> <p>DBS 定义与特点,文件系统与数据库系统的区别</p> <p>三级抽象(视图抽象、逻辑抽象、物理抽象)</p> <p>二级映射</p> <p>数据独立性(物理数据独立性、逻辑数据独立性)</p>	<ul style="list-style-type: none">① 什么是数据管理？② 数据库系统的基本概念③ 数据库系统的宝贵知识财富<ul style="list-style-type: none">▶ 数据独立性▶ 数据库语言▶ 索引结构▶ 事务处理④ 数据库管理系统的组成

关系数据库系统	<p>关系模型、关系运算</p> <p>重点:</p> <p>关系数据库的数据结构(元组、属性、域、码(超码、候选码、外码))</p> <p>完整性约束规则(实体完整性、参照完整性、用户自定义完整性)、</p> <p>关系代数(6 种基本操作, 关系代数书写查询)</p>	<p>关系数据模型</p> <ul style="list-style-type: none">关系数据结构: 关系、属性、键关系操作: 查询操作、更新操作(插入、修改、删除)、查询语言(关系代数、关系演算、SQL)关系完整性约束: 实体完整性、参照完整性、用户定义完整性 <p>关系代数</p> <ul style="list-style-type: none">基本关系代数操作: 选择σ、投影Π、笛卡尔积\times、并\cup、差$-$、重命名ρ派生关系代数操作: 交\cap、内连接\bowtie_{θ}、自然连接\bowtie、外连接(左外连接\bowtie_{L}、右外连接\bowtie_{R}、全外连接\bowtie_{ALL})、除\div扩展关系代数操作: 分组操作γ、赋值操作$=$ <p>关系演算</p> <ul style="list-style-type: none">元组关系演算域关系演算
结构化查询语言 SQL	<p>SQL-数据定义、查询、数据库修改、事物、触发器, 嵌入式 SQL 简介</p> <p>重点:</p> <p>SQL 数据定义(关系定义、视图定义、完整性约束的 SQL 语言表示),</p> <p>SQL 查询(单表、连接、分组、聚集)、</p> <p>SQL 数据修改</p>	<p>SQL 数据定义</p> <ul style="list-style-type: none">基本数据类型: 数值型、日期时间型、字符串型、枚举型创建关系模式: CREATE TABLE修改关系模式: ALTER TABLE删除关系模式: DROP TABLE定义视图: CREATE VIEW, ALTER VIEW, DROP VIEW <p>SQL 数据更新</p> <ul style="list-style-type: none">插入数据: INSERT修改数据: UPDATE删除数据: DELETE <p>SQL 数据查询</p> <ul style="list-style-type: none">单关系查询: SELECT ... FROM ... WHERE集合查询: UNION, INTERSECT, MINUS/EXCEPT连接查询: INNER JOIN, NATURAL JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN嵌套查询: IN, 比较运算符, EXISTS, 派生表
概念数据库设计	<p>数据库设计概述、需求分析、概念数据库设计方法</p> <p>重点:</p> <p>ER 图 实体 属性 码</p> <p>联系:映射基数、参与约束(全域关联、部分关联)</p>	<p>数据库设计的过程: 概念数据库设计→ 逻辑数据库设计→ 物理数据库设计</p> <p>实体-联系模型(ER模型)</p> <ul style="list-style-type: none">与实体相关的概念<ul style="list-style-type: none">实体属性: 简单属性、复合属性、多值属性、派生属性键属性实体型、实体集弱实体型: 标识实体型、标识联系型、部分键与联系相关的概念<ul style="list-style-type: none">联系联系型、联系集联系型的约束: 基数比(1:1、N:1、M:N)、参与度约束(全部参与、部分参与)联系型的属性多元联系 <p>实体-联系图(ER图)²</p> <p>增强实体-联系图(EER图)</p>
逻辑	<p>初始关系模式形成, 关系模式规范化、优化, 定义完整性和安全性约束, 定义子模</p>	

数据库设计	<p>式，性能估计</p> <p>重点：</p> <p>ER 图—>关系表，</p> <p>关系数据库的规范化：函数依赖、完全函数依赖、部分函数依赖、传递函数依赖、</p> <p>Armstrong 公理系统、求属性闭包、求候选码、求极小函数依赖集，</p> <p>关系数据规范形式:1NF、2NF、3NF、BCNF、无损连接性、函数依赖保持性、判别方法、关系模式的分解算法</p>	<p>ER模型转换为关系数据库模式</p> <ul style="list-style-type: none">▶ 实体相关的概念→关系模式: 实体型、复合属性、多值属性、弱实体型的转换▶ 联系相关的概念→关系模式: M:N联系型、N:1联系型、二元自联系型的转换 <p>函数依赖</p> <ul style="list-style-type: none">▶ 函数依赖的类型: 完全函数依赖、部分函数依赖、传递函数依赖▶ Armstrong公理系统: 自反律、增广律、传递律▶ 属性集的闭包: 概念、计算算法、如何用于推理▶ 等价函数依赖集: 概念、证明方法▶ 函数依赖集的最小覆盖: 概念、计算算法 <p>关系模式的范式</p> <ul style="list-style-type: none">▶ 范式的种类: $4NF \subseteq BCNF \subseteq 3NF \subseteq 2NF \subseteq 1NF$▶ 不规范关系模式存在的问题、成因及解决方法 <p>关系模式分解</p> <ul style="list-style-type: none">▶ 关系模式分解的准则: 无损连接性、函数依赖保持性▶ 无损连接性的判定▶ 函数依赖保持性的判定▶ 关系模式分解算法
物理数据库设计	<p>影响物理数据库设计的因素，关系模式选择存取方法，设计关系、索引等数据库文件的物理存储结构</p> <p>重点：</p> <p>影响物理数据库设计的因素、聚集存储方法</p>	<p>① 物理数据库设计的步骤</p> <p>② 索引的设计</p> <ul style="list-style-type: none">▶ 索引的构成▶ 索引的分类▶ 索引数据结构▶ 索引设计技巧▶ 查询改写 <p>③ 物理存储结构的设计</p> <ul style="list-style-type: none">▶ 数据类型的选择▶ 数据库的划分
物理存储结构	<p>数据库存储设备，磁盘文件，Hash 文件，索引文件 B+树文件索引，B 树文件索引</p> <p>重点：</p> <p>磁盘缓冲区处理技术、</p> <p>磁盘容错技术(RAID1,2,4,5)</p> <p>磁盘文件(文件，文件记录、文件存储方式(连续、链式、索引)</p> <p>Hash 文件(动态 Hash 技术)</p>	<p>1.存储介质</p> <p>2.数据库在磁盘上的表示</p> <p>值表示 元组布局 页面布局</p> <p>面向元组的页面布局</p> <p>日志结构的页面布局</p> <p>文件组织</p> <p>3.系统目录</p> <p>4 .Buffer 管理</p>

	索引文件(主索引、聚集索引、辅助索引、多级索引、B+树索引结构)	<div>1.基于散列的索引结构</div> <div>可扩展的哈希表 线性散列表</div> <div>2.基于树的索引结构</div> <div>B +树</div> <div>3.日志结构合并树(Ism - tree)</div>
查询处理	关系代数操作实现算法 重点: 选择、投影、连接(Nest-Loop-Join(优化), Sort-Merge-Join、Hash Join , 共 5 种)、集合操作	<div>1.概述</div> <div>2.外部排序 外部归并排序</div> <div>3.执行关系代数的操作</div> <div>执行选择操作 执行投影操作</div> <div>执行重复消除操作 执行聚合操作</div> <div>执行集合操作 执行连接操作</div> <div>4.执行表达式</div>
查询优化	关系表达式的等价转换规则, 表达式结果大小的估计, 启发式关系代数优化算法, 复杂性估计方法 重点: 关系代数的等价转换规则 表达式结果大小的估计(均匀分布)、处理时间的估计 启发式关系代数优化算法(查询树、优化树)	<div>1.概述</div> <div>2.改进逻辑查询计划</div> <div>关系代数表达式的转换</div> <div>查询计划代价估算</div> <div>连接顺序优化</div> <div>3.改进物理查询计划</div>
并发	事务概念, 事务的并发执行和调度, 并发控制协议(基于锁的协议、基于时间戳的	<div>1.交易</div> <div>2.并发控制</div>

控制	<p>协议、多版本机制快照隔离)</p> <p>重点:</p> <p>事务的概念(ACID)、数据库的一致性</p> <p>串行调度、可串行调度、冲突可串行化的判定方法(定义, 冲突图)</p> <p>基于锁的并发控制协议(锁的概念与种类, 锁的相容关系、两段锁协议、死锁的判定)</p> <p>基于时间戳的并发控制协议, 两者的比较</p>	<p>日程安排 隔离级别 可串行性</p> <p>3.基于锁的并发控制</p> <p>锁 两相锁(2PL) 强严格两阶段锁(SS2PL)</p> <p>死锁 多粒度锁 幻影</p> <p>4.时间戳排序(T/O)并发控制</p> <p>基本的 T / O 乐观并发控制(OCC)</p> <p>多版本并发控制(MVCC)</p>
数据库恢复	<p>数据库恢复必要性, 使用日志的数据库恢复技术使用检查点的数据库恢复技术, 恢复算法, 缓冲技术</p> <p>重点:</p> <p>数据库故障分类</p> <p>使用日志的数据库恢复技术(日志的内容, redo、undo、推迟更新技术、即时更新技术)</p> <p>使用检查点的数据库恢复技术(日志的内容, 哪些事物需要 redo、哪些需要 undo)、恢复算法(*先写日志, 后更新数据库)</p>	