1、绪论

**1.1 4个基本概念**：数据、数据库、数据库管理系统、数据库技术

数据：描述事物的符号记录，数据的含义成为数据的语义。

数据库：较小的冗余度、较高的数据独立性、易拓展性、共享性。

数据库管理系统，位于用户与操作系统之间的一层数据管理软件。

数据库系统：由数据库、数据库管理系统、应用程序、数据库管理员组成。

**1.2 数据库系统特点**

数据结构化

数据共享性高、冗余度低

数据独立性高（物理独立性、逻辑独立性）

**1.3 数据模型**

概念模型、逻辑模型、物理模型

逻辑模型（层次模型、网状模型、关系模型）

数据库模式：数据库全体数据的逻辑结构和特征的描述。

模式、外模式、内模式。

数据库二级映像功能和数据独立性。外模式-模式映像。模式-内模式映像。

2、关系数据库

**2.1 关系**

域：domain一组具有相同数据类型的集合

笛卡儿积 cartestian product：

关系relation

主键primary key

基本表、查询表、视图表

关系规范化normalization，规范化的关系称为范式。

**2.2 关系操作**

选择、投影、并、差、笛卡儿积

实体完整性、主键不能取空值。

参照完整性：外码要么取空值、要么等于主码值。

关系代数：选择selection：按条件选择行

投影projection：选择列

连接join：等值连接。自然连接相当于等值连接取消重复列

除division：

3、SQL语句structured query language

外模式：视图。模式：基本表。内模式，存储文件。

**数据定义**：create table

Create scheme创建模式

Create view 视图

Create index索引

修改基本表：alter table

删除：drop table

**数据查询：**select Sno from SC where grade < 20

Select … from … where sage between 20 and 23

字符串匹配：where sname like ‘欧阳\_’

排序(升序)：order by

聚集函数：count(\*)总数 sum avg max min

连接查询：where student.sno = sc.sno and sc.cno =’2’ and sc.grade > 90

嵌套查询：多表查询可用。复杂度高。

数据更新

插入数据：insert into student(…) values (…)

修改数据: update student set sage = 22 where sno = ‘…’

删除数据 delete from table where …

视图：数据库中只存放视图的定义，不存放视图对应的数据，这些数据仍存放在原来的基本表中。

视图一般不允许更新，按不同规定而来。

4、数据库安全性

TCSEC数据库安全D、C、B、A。c2是安全保护最低档；B1是真正意义的安全产品。

**存取控制**：包括定义用户权限和合法权限检查两部分。

合法权限检查：1、自主存取控制（用户对于不同的数据库对象有不同的存取权限。）2、强制存取控制（对于任意对象、只有具有合法许可证的用户才可以存取）。C2支持DAC(Discrationary access control),B1支持强制存取控制。

强制存取控制：（1）仅当主体的许可证级别大于等于客体密级，该主体才能读取相应客体。（2）仅当主体许可证级别小于等于客体密级，主体才能读取客体。

授权：grant 权限 to user

收回权限：revoke

**安全保护方法**

视图机制、用户身份鉴别、存取控制

审计，利用审计日志监控数据库行为。

数据加密。

5、数据库完整性

数据完整性指数据正确性和相容性。

1、提供定义完整性约束条件的机制

2、提供完整性检查的方法

3、进行违约处理。

实体完整性：

参照完整性：

用户定义的完整性：

断言：任何断言不为真值的操作都会被**拒绝执行**。

触发器：自动更改器，如工资不得低于4000元，如果低于4000元自动改为4000元。

完整性机制的实施会影响系统性能；注意一个触发器的动作可能激活另一个触发器。

6、关系数据理论

一个关系模式应当是一个五元组R(U,D,DOM,F)

U:一组属性;D代表属性域;DOM为属性到域的映射;F为属性U的多值依赖。

数据依赖是关系内部**属性与属性**之间的一种约束关系。函数依赖、多值依赖。

函数依赖：X->Y表示Y依赖于X，即X确定Y。

范式，关系数据库的关系是要满足一定要求的，满足不同程度要求的为不同范式。一个低一级范式的关系模式可以通过模式分解，可以转换为高一级范式的关系模式，称为规范化。

2NF:一个非主属性完全依赖一个候选码

3NF:每一个非主属性不传递依赖于码，也不部分依赖于码。

4NF:限制关系模式的属性之间不允许有非平凡且非函数依赖的多值依赖。

7、数据库设计

三分技术、七分管理、十二分基础数据

（1）需求分析阶段

（2）概念结构设计：E-R模型

E-R模型：实体型用矩形表示、属性用椭圆形表示、联系用菱形表示。

1对1联系、1对多联系、多对多联系

UML表示E-R图

（3）逻辑结构设计

（4）物理结构设计

B+树和hash索引是数据库经典的存取方法

聚簇存取：根据特点将一系列元组集中存放，每读一个物理块可以得到多个满足条件的元组。

（5）数据库实施和维护

9、关系查询处理和查询优化

查询优化可分为代数优化（逻辑优化）和物理优化（非代数优化）

代数优化是指关系代数表达式的优化、物理优化指通过存取路径和底层操作算法的选择进行的优化。

查询处理可分为四个阶段：查询分析、查询检查、查询优化和查询执行。

选择操作：

（1）全表扫描

（2）索引扫描（B+树和哈希索引）

连接操作：

（1）嵌套循环

（2）排序-合并

（3）索引连接

（3）hash join算法

查询优化：

总代价 = I/O代价 + CPU代价 + 内存代价 + 通信代价

计算查询代价一般使用查询处理读写的块数作为衡量单位。

（1）**选择运算应尽可能先做**

（2）投影运算和选择运算同时进行

（3）把笛卡尔积转换成一个连接运算。（连接的数量越少越好）

10、数据库恢复技术

**事务**：用户定义的一个数据库操作序列，这些操作要么全做要么全不做，是一个不可分割的工作单位。

事务通常以BEGIN TRANSACTION开始，以COMMIT或ROLLBACK结束，commit表示提交，rollback表示回滚，撤销。

事务四个特性：原子性、一致性、隔离性、持续性。

原子性：要么全做，要么不做；

一致性：数据库必须由一个一致性状态变到另一个一致性状态，尤其在分布式数据库中，分布式锁zookeeper。

隔离性：事务执行不能被其他事务干扰，分布式。

持久性：事务一旦提交，改变永久性。

事务ACID特性被破坏原因（1）多个事务并行运行时不同事务操作交叉执行；

（2）事务在运行过程中被强行停止。

故障的种类：事务内部的故障，系统故障，介质故障，计算机病毒。

**恢复实现**：**数据转储、日志文件**

**具有检查点的恢复技术**。在检查点之前提交、不要重做；

在检查点之前执行、故障点之前提交：REDO

检查点之前执行、故障点未完成：UNDO撤销

11、并发控制

防止并发操作破坏事务的隔离性，并发控制的主要技术有封锁、时间戳等。

封锁：排他锁（exclsive locks, X锁），又称为写锁，只允许上锁者读取和修改。

共享锁(share locks, S锁)，读锁；上锁者可以读，但期间其他操作也可以读，但不能修改，直到上锁者释放锁为止。

一级封锁协议：事务T在**修改数据**R之前必须先对其**加X锁**，直到事务结束才释放。但仅仅读数据，则不加锁。

二级封锁协议：在一级封锁协议基础上，如果读取数据R必须加S锁，读完后释放S锁。

三级封锁协议：在一级封锁协议基础上，如果读取数据R必须加S锁，直到事务结束后释放S锁。

死锁和活锁：

活锁，某些事务饥饿，永远等待的情况。

死锁：互斥，不剥夺，请求和保持，循环等待。

12、数据库管理系统

系统结构：应用层->语言处理层（SQL）->数据存取层（单个元组）->数据存储层（数据页和缓冲区）

语言处理层：解释、预编译方法

数据存取层：记录存取、事务管理子系统；日志登记子系统；

数据存储层：缓冲区管理、物理组织。

12、大数据

分布式数据库系统、并行数据库系统

非关系型数据库NoSQL

传统数据库在**高可扩展型**（满足数据量增长需要）、**高性能**（满足读写实时和查询处理）、**容错性**（保证分布式系统可用性，查询失败只需重做部分工作）、**可伸缩性**（按需分配资源）。

NoSQL指非关系型、分布式的、不保证满足ACID特性的一类数据管理系统。

（1）对数据进行划分，通过大量节点的并行处理获得高性能、采用横向扩展方式；

（2）放松对数据的ACID一致性约束，允许数据暂时不一致情况，接受最终一致性。

（2）对各个数据分区进行备份，应对节点可能的失败。

大数据应用：互联网文本大数据管理和挖掘

基于大数据分析的用户建模

数据仓库：为了构建新的分析处理环境而出现的一种数据存储和组织技术。

（1）数据仓库的数据是面向主题进行组织的

（2）数据仓库是集成的，数据从原始数据库抽取并加工集成而来

（3）数据仓库是不可更新的，主要设计查询操作。

（4）数据仓库随时间变化。

即HBase -> Hive -> HDFS -> MapReduce，Hive为数据仓库。

hive是基于Hadoop的一个[数据仓库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E4%BB%93%E5%BA%93/381916)工具，可以将结构化的数据文件映射为一张数据库表，并提供简单的sql查询功能，可以将sql语句转换为MapReduce任务进行运行。