在本次课程中，我只重点记录自己的感悟，每一个感悟为一个次级标题。

# 第一讲：数据库概述

## 小知识点：

1. 对于SQL有很多路劲，要选择一个执行代价最低的[查询处理曾做的事情]
2. 数据库要查询时间+执行时间最短才可以
3. 数据库最核心的技术：数据页面的随意读写转成日志的连续读写：日志
4. 异构：不同系统不同语言协同使用
5. 异步：可以理解为两个事件同步进行
6. Datagrip是一个管理数据库的软件，数据库有很多，比如SQLite、MySQL、PostgreSQL、Gauss DB。不同的仓库有着不同的应用场景，也有着不同的内核。

## 专题：

### 数据库系统：

数据库系统可以分成四个方面：

数据（data）

数据库（database,DB）

数据库管理系统（database management system，DBMS）【我要学的核心】

核心内容有【查询处理层，查询执行层，存储管理层】

数据库系统（Database System,DBS）

### 数据库类型：

结构化数据（关系数据：由多张表格构成的数据）

半结构化数据（键值对，SML，JSON，图，向量）

非结构化数据（文本文档，电子邮件，图像，音频，视频）

### 数据库基本特征：

1. 数据按照一定的数据模型组织，描述和存储
2. 支持数据的增删改查
3. 支持并发查询处理
4. 通用性和复制性

### 查询接口（用户接口）：

JDBC（Java）

ODBC（open支持多种语言）

嵌入式SQL（内置函数）

### 数据库管理系统主要功能

1. 数据定义功能 （DDL data definition language）对表格进行处理
2. 数据操作功能 （DML data manipulation language）支持增删查改
3. 数据组织，存储，查询处理

### 不同的算子：

**1，选择σ（过滤 SC）​**

TypeScript取消自动换行复制

for t in SC:​

if t.Cno == '1':​

emit(t) ​

**2，连接⋈（关联 SC 过滤结果与 Student）​**

TypeScript取消自动换行复制

StudentHash = {t.Sno: t for t in Student} ​

for sc\_t in 选择算子输出流:​

s\_no = sc\_t.Sno​

if s\_no in StudentHash:​

student\_t = StudentHash[s\_no]​

joined\_t = (sc\_t.Sno, sc\_t.Cno, sc\_t.Grade, student\_t.Sname, ...) ​

emit(joined\_t) ​

**3，投影⋈（提取指定列）​**

TypeScript取消自动换行复制

for joined\_t in 连接算子输出流:​

projected\_t = (joined\_t.Sno, joined\_t.Sname, joined\_t.Grade) ​

emit(projected\_t) ​

​

# 第二讲：关系模型

## 小知识点：

1. 华为Gauss DB是分布式关系型数据库
2. 数据库操作为增删查改四个操作
3. 三种语言：关系代数语言（第一步，第二步，……）关系演算语言（我需要……），其中SQL是二者之间的一种，是非过程化语言（过程化语言具有关系代数语言的特性）

## 专题：

### 关系型数据库的专业术语：

|  |  |  |
| --- | --- | --- |
| 关系模型术语 | 对应 Excel 里的东西 | 例子（学生表） |
| 关系模式 | 表头 | [年龄，成绩]表格列名集合 |
| 关系名 | 表明 | Student学生表 |
| 关系 | 整个表格（表格其实代表的就是关系） | 学生表（含所有学生信息） |
| 元组 | 一行数据（录入一条信息就叫做元组） | 李博（学号 2021310721 那一行） |
| 属性 | 一列数据（年龄，科目都是属性） | “学号”“姓名” 这一列 |
| 属性名 | 列标题（属性的抬头） | “Sno”（学号）、“Sname”（姓名） |
| 分量 | 一个具体的单元格值（小明的年龄） | 李博的年龄 “17” |
| 超键 | 能唯一区分行的列组合（线性空间） | “学号 + 姓名”（可唯一确定一个学生） |
| 候选键 | 精简的唯一标识列（极大线性无关组） | “学号”（单独一列即可唯一确定学生） |
| 复合键 | 多个属性决定一个一个元组 | [名字]+[身份证后四位] |
| 主属性 | 极大线性无关组的属性 | [学号][身份证号] |
| 主键 | 官方指定的唯一标识列（基向量） | “学号”（被选定为主要标识） |
| 外键 | 关联其他表的列（同态映射），这个属性在别的表格中也有，就叫做外键。 | 选课表中的 “选 课 ”（关联学生表的学号） |

### 数据完整性约束

1. 实体完整性：主键唯一非空
2. 参照完整性：外键必须对应的上，只要引用，必须存在。
3. 用户定义完整性：成绩不能负数，年龄0~120之类的用户自定义数据
4. 其他约束[自增长约束：元组ID自增长][默认值约束：属性取值默认为null或者0][检查约束：属性需要满足特定属性，比如年龄＞0之类]
5. 关系演算：找到满足属性的所有元组，最后拼在一起。

### 6大关系运算

| **关系代数运算** | **符号** | **核心作用** | **举例（符号表示）** | **举例说明（言简意赅）** |
| --- | --- | --- | --- | --- |
| 选择 | σ | 挑出符合条件的行 |  | 从学生表中选出年龄为18岁的行 |
| 投影 | Π | 只保留需要的列，将表格降维 |  | 从学生表中只保留学号（Sno）和姓名（Sname）列 |
| 并 | ∪ | 合并两表的行（结构需相同） |  | 合并17岁学生表和18岁学生表，结果为17或18岁学生 |
| 差 | − | 从一个表去掉另一个表的行 |  | 从计算机系学生表中去掉18岁及以上的行，结果为计算机系未满18岁的学生 |
| 笛卡尔积 | × | 两表所有行两两配对 | Student × Course | 学生表的每行与课程表的每行配对，若学生表3行、课程表2行，结果为6行 |
| 重命名 | ρ | 给表或列改名字 |  | 把选课表（SC）改名为StudentCourse，成绩列（Grade）改名为Score |

### 更高级的算子

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **组合/扩展工具** | **符号** | **核心作用** | **举例（符号表示）** | **举例说明（言简意赅）** |
| 交 | ∩ | 取两表的共同行 | σSdept="CS"(Student)∩σSage=18(Student) | 计算机系且18岁的学生 |
| 连接 | ⋈ | 按条件合并两表（只保留匹配的配对） | Student ⋈Student.Sno = SC.Sno SC | 每个学生与其选课记录一一对应 |
| 赋值 |  | 右侧结果赋值给左边 |  | 将相等记录赋值给resule |
| 除 | ÷ | 找“包含所有目标”的行 | SC ÷ ΠCno(Course) | 选了所有课程的学生 |
| **拓展关系代数运算** | **符号** | **核心作用** | **举例（符号表示）** | **举例说明（言简意赅）** |
| 去重 | δ | 删除重复行 | δ(Student) | 去掉学生表中重复的行 |
| 广义投影 |  |  |  |  |
| 聚集 | 𝒢 | 计算统计量 | 𝒢AVG(Sage)(Student) | 求所有学生的平均年龄 |
| 分组 | 𝒢 | 先分组再统计 | 𝒢Sdept,COUNT(\*)(σSsex="男"(Student)) | 按系分组，统计每个系的男生人数 |

# 第四讲：SQL

## 小知识点：

## 专题：

# 第十讲：查询处理

## 小知识点：

## 专题：

# 第十一讲：数据库优化器

## 小知识点：

## 专题：