# 数据库系统原理笔记

陈鸿峥

2019.09*

## 目录



本课程采用书目Avi Silberschatz, Henry F. Korth, S. Sudarshan, *Database System Concepts (6th ed)*[1]。

## 1 数据库系统概述

早期的数据库直接建立在文件系统上，但这会导致：

- 数据冗余与不一致
- 访问数据非常麻烦
- 完整性问题：难以添加限制（如年龄为非负整数）
- 更新的原子性
- 多用户的并发访问
- 安全性问题：权限

查询过程：解释编译+求值(evaluation)

*Build 20190904

[1] http://www.db-book.com

# 2 关系型数据库

纵向为属性(attributes/columns)，横向为元组(tuples/rows)
注意关系都是无序的，元组可以以任意顺序存储

- Schema：`instructor(ID,name,dept_name,salary)`
- Instance：局部数据
- 键值(keys)$R$
- 超键(superkey)$K \subset R$
- 候选键(candidate key)$K$为原子/不可分割/最小键

关系代数(relational algebra)

| | | |
|---|---|---|
| 选择 | $\sigma$ | 挑选出符合一定性质的元组<br>$\sigma_{\text{Sub="Phy"} \wedge \text{age>30}}(\text{teachers})$ |
| 投影 | $\Pi$ | 只选出对应属性<br>$\Pi_{\text{ID,name,salary}}(\text{teachers})$ |
| 笛卡尔积 | $\times$ | 将两个关系整合（简单并置，需要进一步筛选） |
| 合并 | $r \bowtie_\theta s = \sigma_\theta(r \times s)$ | |
| 并集 | $\cup$ | 数目应相同，属性可兼容 |
| 交集 | $\cap$ | |
| 差集 | $-$ | |
| 赋值 | $\leftarrow$ | |
| 重命名 | $\rho_x(E)$ | 给$E$的返回值赋名为$x$ |

# 3 SQL简介

结构化查询语言(Structured Query Language, SQL)
数据定义语言(Data Definition Language, DDL)

```
create table instructor (
    ID char(5),
    name varchar(20),
    dept_name varchar(20),
    salary numeric(8,2),
    primary key (ID),
    foreign key (dept_name) references department);
```

查询语言

```
select A1, A2, ..., An
from r1, r2, ..., rm
where P
```

其中$A_i$为属性(attribute)、$R_i$为一个关系(relation)/一个表、$P$是谓词(preicate)

```
select distinct dept_name
from instructor

select all dept_name
from instructor

select *
from instructor

select '437' as FOO

select ID, name, salary/12 as monthly_salary

select name
from instructor
where dept_name = 'Comp. Sci.' and salary > 80000

select *
from instructor, teaches -- Cartesian product

select distinct T.name
from instructor as T, instructor as S -- rename
where T.salary > S.salary and S.dept_name = 'Comp. Sci.'

/**
percent ( % ). The % character matches any substring.
underscore ( _ ). The _ character matches any character.
**/
select name
from instructor
where name like '%dar%' matches any string containing "dar" as a substring

select distinct name
from instructor
order by name

select name
from instructor
where salary between 90000 and 100000 -- both contain

(select course_id from section where sem = 'Fall' and year = 2017)
union -- intersect, except
```

```
(select course_id from section where sem = 'Spring' and year = 2018)

select name
from instructor
where salary is null

-- arg, min, max, sum, count
select avg (salary)
from instructor
where dept_name= 'Comp. Sci.';

select count (distinct ID)
from teaches
where semester = 'Spring' and year = 2018;
```

三值逻辑，添加了Unknown

| AND | OR | NOT |
|---|---|---|
| $T \wedge U = U$ | $T \vee U = T$ | |
| $F \wedge U = F$ | $F \vee U = U$ | $\neg U = U$ |
| $U \wedge U = U$ | $U \vee U = U$ | |