

数据库原理(一)

复习课

李晓强
计算机学院



目录

- 1周： 第1章-数据库发展史
- 2周： 第2章-数据库系统结构
- 3周： 第4章-结构化查询语言：
 - SQL概貌及特点; SQL的数据定义; SQL的数据查询
- 4周： 第4章-结构化查询语言：
 - SQL 的数据查询; SQL的数据更新
- 5周： 第4章-结构化查询语言
 - 视图操作;嵌入式SQL
- 6周： 第3章-关系运算： 关系数据模型;关系代数(上)
- 7周： 第3章-关系运算： 关系代数(下);关系演算
- 8周： 第3章-关系运算： 查询优化
- 9周： 第6章-实体联系模型

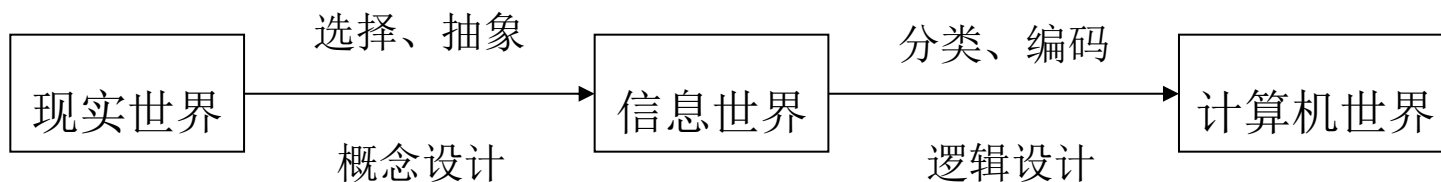


第1章 数据库发展史

数据、信息和知识

数据处理的三个领域

数据表示信息，信息反映事物，从事物到数据经历了三个领域。



第1章 数据库发展史

- 数据管理
 - **人工管理阶段;文件系统阶段;数据库系统阶段**
- 数据库技术
- 数据库体系结构
 - 三级模式：模式、内模式、外模式
 - 二级映象与数据独立性;
- DBMS
 - 工作流程
- 数据库系统结构的分类



第2章 数据库系统结构

- 数据描述

- 概念设计中的数据描述
- 逻辑设计中的数据描述
- 物理设计中的数据描述
- 数据联系的描述

- 数据模型

- 概念数据模型：实体联系模型
- 逻辑（结构）数据模型：层次、网状、关系、对象



第2章 数据库系统结构

- 数据库的体系结构
 - 三级模式结构
 - 两级映像
 - 两级数据独立性



第3章 关系代数

- 基本术语

- 用**二维表格**表示实体集；**外键**表示实体间联系的模型称为关系模型
- **关系**：对应二维表格
- **元组**：表中的行
- **属性**：表中的列
- **域**：属性的取值范围

STUDY

SNAME	COURSE
张三	离散数学
李四	计算机原理
李四	数据结构



第3章 关系代数

- 关系代数语言
 - 传统的集合运算：并、交、差、笛卡尔积
 - 专门的关系运算：选择、投影、联接、除法

关系代数的八种运算中，并、差、积、选择、投影是五种基本运算；其他三种为组合运算。

其中前四个从水平方向看待关系，投影则从垂直方向考虑关系。

外联接操作



第3章 关系代数

- 关系代数 运算的有限次复合，形成 关系代数表达式
- 可以利用 关系代数表达式 表示对数据库的各种查询
- 查询的一般形式：

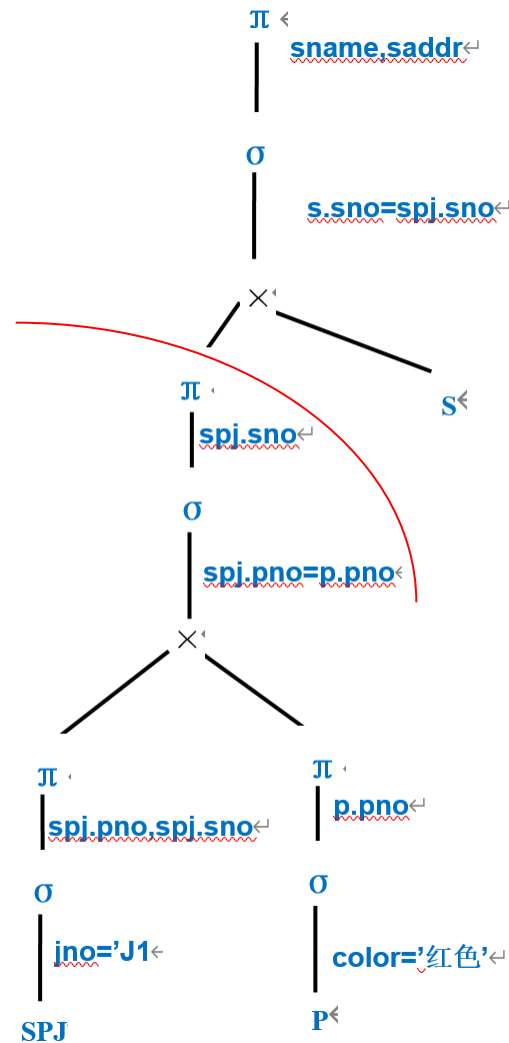
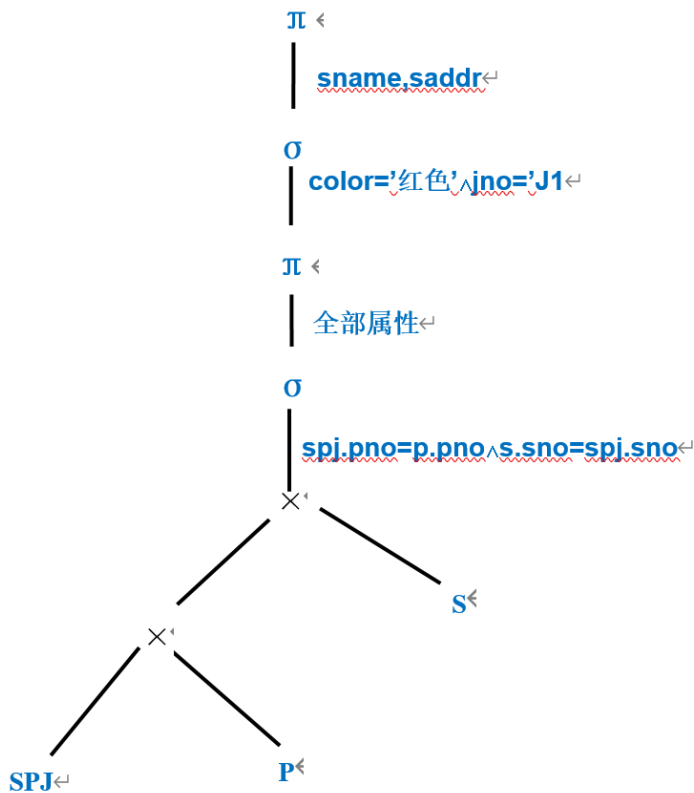
$$\pi_{\dots} (\sigma_{\dots} (R \times S)) \text{ 或 } \pi_{\dots} (\sigma_{\dots} (R \bowtie S))$$



第3章 关系代数

- 查询优化 例：检索供应给工程J1零件红色的供应商名和地址

- 关系代数表达式
- 语法树
- 优化树



第4章

- 数据库是长期存储在计算机内，有组织的，可共享的数据集合
- SQL的特点
 - SQL语言集数据查询、数据操纵、数据定义和数据控制功能于一体
 - 面向集合的语言
 - 非过程语言
 - 类似自然语言，简洁易用
 - 是自含式语言，又是嵌入式语言
- 常见的关系型数据库和非关系型数据库



第4章

- SQL的组成：
 - 数据定义 CREATE、DROP、ALTER
 - 数据操纵 SELECT、INSERT、UPDATE、DELETE
- SQL 数据库的体系结构
 - 一个表可以是一个基本表，也可以是一个视图。基本表是实际存储在数据库中的表，视图是从基本表或其他视图中导出的表，它本身不独立存储在数据库中，即数据库中只存放视图的定义
- 建表
 - 基本表的创建、修改、撤销



第4章

```
CREATE TABLE SPJ
( SNO  CHAR(4)      NOT NULL
  PNO  CHAR(4)      NOT NULL
  JNO  CHAR(4)      NOT NULL
  PRICE NUMERIC(7,2) CHECK(PRICE>0),
  QTY   INT          CHECK(QTY BETWEEN 0 AND 10000),
  PRIMARY KEY(SNO,PNO,JNO))
FOREIGN KEY(SNO) REFERENCES S(SNO),
FOREIGN KEY(PNO) REFERENCES P(PNO),
FOREIGN KEY(JNO) REFERENCES J(JNO),
```



第4章

SELECT [DISTINCT] 列名 表达式序列	--- 投影
FROM 表 视图	--- 关系
[WHERE 行条件表达式]	--- 元组选择
[<GROUP BY 列名序列>	--- 分组
[HAVING 组条件表达式]]	--- 分组选择
[ORDER BY 列名 [ASC DESC], ...]	--- 排序

提示：整个语句执行过程：

FROM→WHERE→GROUP→HAVING→SELECT→ORDER



第4章

- **查询工程的名称和该工程使用的零件名称**

```
select jname,pname  
from j join spj on j.jno=spj.jno join p on spj.pno=p.pno
```

- **查询至少使用了P3和P5零件的工程编号**

```
select jno from spj as a, spj as b,j  
where a.jno=b.jno and a.jno=j.jno and a.pno='P1' and b.pno='P5'
```

- **查询供应红色零件给工程J1的供应商和工程名称**

```
select * from spj,j,p,s  
where spj.jno=j.jno and spj.pno=p.pno and spj.sno=s.sno  
and j.jno='J1' and p.color='红'
```

- **查询使用了P3零件的工程名称**

```
select distinct spj.jno,j.jname from spj,j  
where j.jno=spj.jno and spj.pno='P3'
```



第4章除法、聚合函数、分组、集合操作

如果查询“全部”内容，需用到全称量词

```
SELECT <查询内容> FROM 表1 WHERE  
NOT EXISTS(SELECT * FROM 表2 WHERE  
NOT EXISTS(SELECT * FROM 表3 WHERE  
表3与表1联接条件 AND 表3与表2联接条件))
```

- 表1是查询结果所需要的表
- 表2是全部内容所在的表
- 表3是将联接上述两表的表

例：1.查询使用了全部零件的工程名称(理解为没有零件是不用的)
2.查询至少用了供应商S1所供应的的全部零件的工程号

聚合函数、分组和排序、集合查询

例：1.查询提供两种或两种以上商品的供应商号
2.查询所有单价都超过22元的商品号和商品名称

提示：1.聚合函数不能单独出现在WHERE子句中
2.聚合函数在比较运算符中必是嵌套形式



第4章

1.查询使用了全部零件的工程名称(理解为没有零件是不用的)

```
select jno,jname
from j
where not exists (select * from p where
                  not exists (select * from spj where spj.jno=j.jno and
                  spj.pno=p.pno))
```

2.查询至少用了供应商S1所供应的的全部零件的工程号

```
select jno,pno,sno
from spj as a
where not exists (select * from spj b where b.sno='S1' and
                  not exists (select * from spj as c where c.jno=a.jno and
                  c.pno=b.pno))
```



第4章

- 例： 1.查询提供两种或两种以上商品的供应商号
2.查询所有单价都超过22元的商品号和商品名称

```
select sno  
rom spj  
group by sno  
having COUNT(pno)>=2
```

```
select spj.pno,p.pname  
from spj,p  
where spj.pno=p.pno  
group by spj.pno,p.pname  
having MIN(price)>22
```



第4章 数据更新、视图

- **单记录插入：没有列名表表示插入的是表的所有属性**

INSERT INTO <表名(列名表)> VALUES(元组值)

例：在J表中插入一个元组('J8','地铁两号线','上海')

```
insert into j(jno,jname,jcity) values('J8','地铁两号线','上海')
```

- **多记录插入**

INSERT INTO <表名(列名表)> SELECT子查询

例：插入工程号为J7的供应零件记录，供应给J7的零件和供应给J1的零件号及供应商相同，但每种零件的供应量改为60

```
insert into spj(sno,pno,jno,qty)
select sno,pno,'J7',60 from spj where jno='J1'
```



第4章

➤ 修改

**UPDATE <表名> SET 列名 = 值表达式[, 列名 = 值表达式 ...]
[WHERE 条件表达式]**

例：当S3提供的P5零件的单价低于该零件的平均单价时，将其提高8%

`select AVG(price) from spj where pno='P5'`

`update spj set price=price*1.08
where sno='S3' and pno='P5' and
price<(select AVG(price) from spj where pno='P5')`

➤ 删除

DELETE FROM <表名> [WHERE 条件表达式]

例：删除红色零件



第4章

- 视图

- 视图是从若干基本表或其他视图采用SELECT语句构造出来的表，并不存储视图对应的数据，称为“虚表”
- **CREATE VIEW <视图名(列名表)> AS SELECT 子查询 [WITH CHECK OPTION]**
- 视图的更新
 - 如果在导出视图的过程中，使用了分组和聚合操作，那么不容许对这个视图执行更新操作



第4章

- 设计一个提供黄色零件的供应商名和地址

```
CREATE VIEW V2 AS  
SELECT SNAME,SADDR  
FROM S,SPJ,P  
WHERE S.SNO=SPJ.SNO AND P.PNO=SPJ.PNO AND  
COLOR='黄'
```



第6章

- 属性、联系和实体

- 实体

- 弱实体；子类和超类

- 属性:基本属性(单值)、复合属性、多值属性和导出属性

- 联系:联系的元数;联系的类型;联系基数

联系基数:

有两个实体集E1和E2，E1中每个实体与E2中有联系实体数目的最小MIN最大值MAX，称为E1的基数，记为(MIN,MAX)

- ER模型设计步骤

- 设计实体类型

- 设计联系类型

- 添加属性



第6章

- 数据模型：**表示实体及实体的联系**
 - 概念数据模型、逻辑数据模型
 - 概念数据模型：独立于计算机系统的数据模型，如**E-R模型**
 - 逻辑(结构)数据模型：与DBMS有关的数据模型，如关系模型
 - 层次、网状、关系
 - 数据模型三要素：数据结构；数据操作；数据完整性约束
- 关系模型
 - 关系数据结构、关系操作和关系完整性约束
- 关系完整性约束
 - 实体完整性、参照完整性、用户定义的完整性



第6章

E-R模型到关系模型的转换

- 1) 每个实体集、弱实体、超类、子类各转为一个关系
- 2) 1:m 联系: n端实体集所对应的关系中新增1端实体集的主键作为外键
- 3) 1:1 联系: 实体数少的实体集所对应关系中增加另1端实体集的主键作为外键;或者任意一个加入另一个。
- 4) n:m 联系: 引入一个新的关系, 新关系主键由相关实体集的主键组合而成, 而这些新增属性又是新关系的外键
- 5) 在弱实体集对应的关系的主键中加入父实体集的主键, 新增属性又是弱实体的外键
- 6) 超类和子类: 在子类对应的关系中加入超类的主键作为子表的主键。该主键也是外键



题型

- 选择题
- 判断题
- 填空题
- 应用题
- 简答题
- 综合题



Q & A

练习题

1. 检索至少使用了编号为**P3**和**P5**零件的工程号
2. 检索至少使用了编号为**P3**或**P5**零件的工程号
3. 检索不使用编号为**P3**零件的工程编号和名称
4. 检索使用了全部零件的工程名
5. 检索使用包含编号**S1**的供应商所供应的全部零件的工程号

