

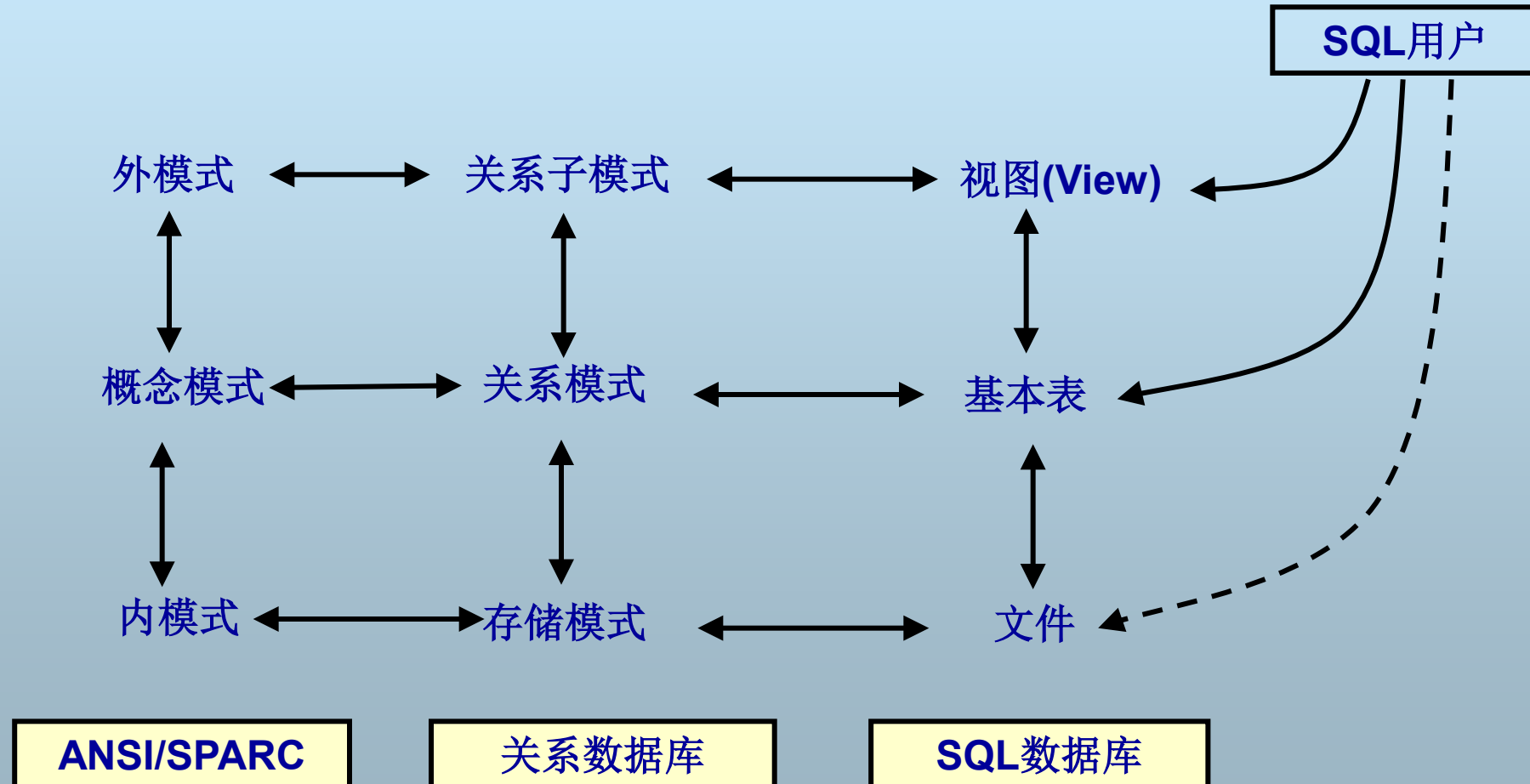
第4章 关系数据库语言SQL

Where are we?

- 数据库语言
- SQL概述
- SQL DDL
- SQL DML
- 视图 

六、视图 (View)

- 视图 (View) 给出了SQL数据库的外模式定义

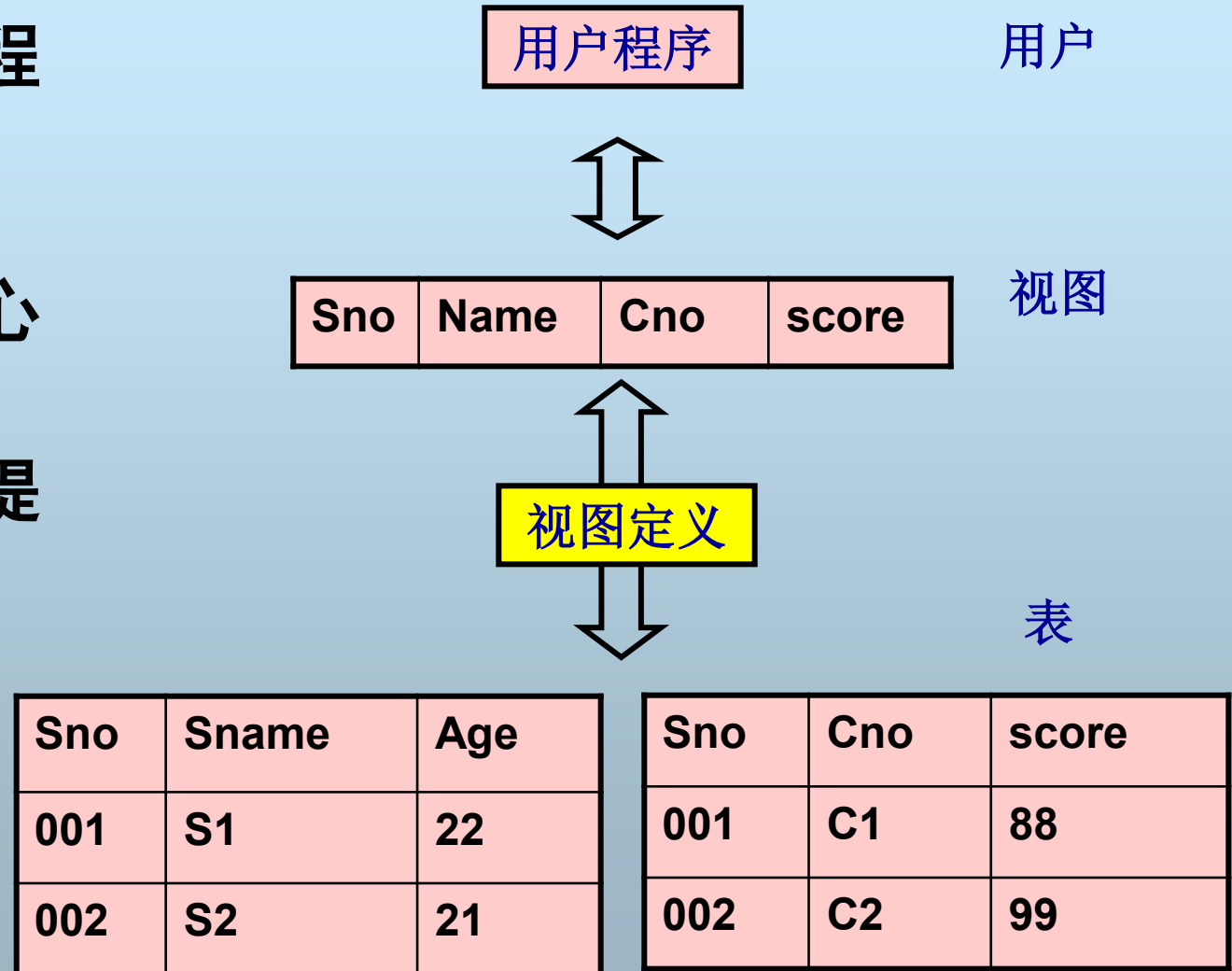


1、视图的概念

- 视图是从一个或几个基本表中导出的虚拟表，其数据没有实际存储，但可以和表一样操作
 - 视图具有和表一样的逻辑结构定义
 - 但视图没有相应的存储文件，而每个表都有相应的存储文件

2、视图的用途

- 逻辑数据独立性：用户程序与数据库结构
- 简化了用户眼中的数据，使用户可以集中于所关心的数据上
- 同一数据库对不同用户提供不同的数据呈现方式
- 安全保护



3、视图的定义

■ **Create View** <视图名> (列名1, 列名2, ...)
AS <查询>

- <查询>是一个**Select**语句，指明视图定义在哪些基本表上，定义了什么内容的数据
- <列名表>定义了视图的逻辑结构，与<查询>中返回的数据相对应

3、视图的定义

■ 例1：定义计算机系的学生视图

- **Create View cs_view (sno, name, age)**
As Select Sno, sname, age
From student
Where Dept='计算机系'

cs_view(sno,name,age)

- **Create View cs_view**
As Select Sno, sname, age
From student
Where Dept='计算机系'

cs_view(Sno,sname,age)

- 若省略视图的列名表，则自动获得Select查询返回的列名

3、视图的定义

■ 例2：把每门课程的课程号和平均成绩定义为视图

- **Create View c_view**
As Select cno, AVG(score) as avg_score
From sc
Group By Cno
- **Create View c_view (cno, avg_score)**
As Select cno, AVG(score)
From sc
Group By cno

■ 在查询中使用了函数时

- 若省略列名表，则必须为函数指定别名
- 若使用了列名表，则可以不指定函数的别名

4、视图的查询

- 与基本表的查询相同

- 例：查询平均成绩在80分以上的课程名

- 不使用视图

- ◆ Select a. cname

- From Course a,

- (Select cno, avg(score) as avg_score From sc Group By cno) SC2

- Where a.Cno=SC2.Cno and SC2.avg_score>80

- 使用前面定义的视图 c_view

- ◆ Select a.cname From course a, c_view b

- Where a.Cno=b.Cno and b.avg_score>80

5、视图的更新

- 与表的更新类似
- 例：将计算机系学号为'001'的学生的姓名改为'Rose'
 - **Update cs_view**
Set name='Rose'
Where Sno='001'
 - 执行时先转换为student上的更新语句再执行
- 不是所有视图都是可更新的
 - 基于连接查询的视图不可更新
 - 使用了函数、表达式、**Distinct**的视图不可更新
 - 使用了分组聚集操作的视图不可更新
- 只有**建立在单个表上，而且只是去掉了基本表的某些行和列，但保留了主键的视图**才是可更新的

6、视图的删除

■ **Drop View** <视图名>

补充：再论Group By

■ Select语句后能否包含不在Group By中出现的列？

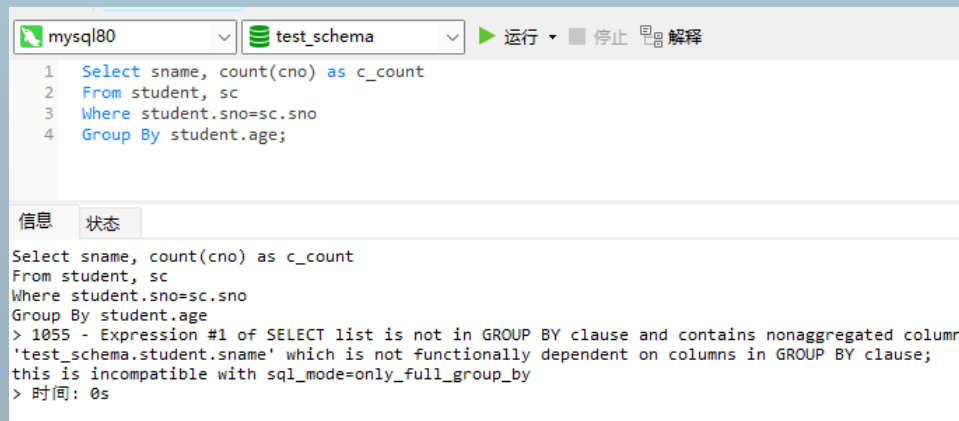
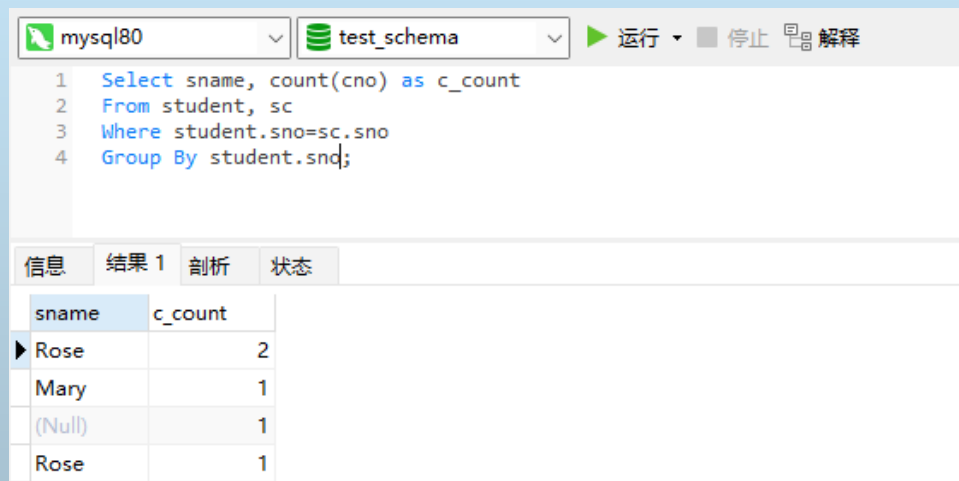
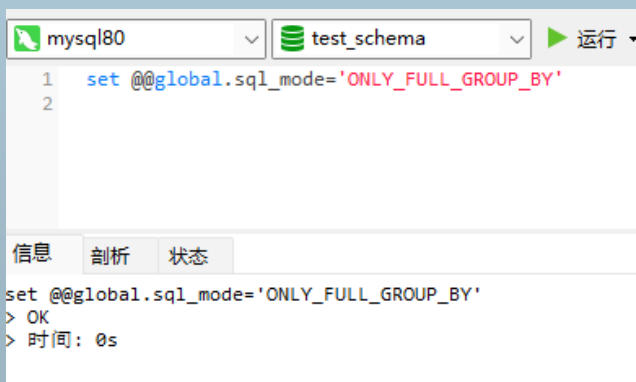
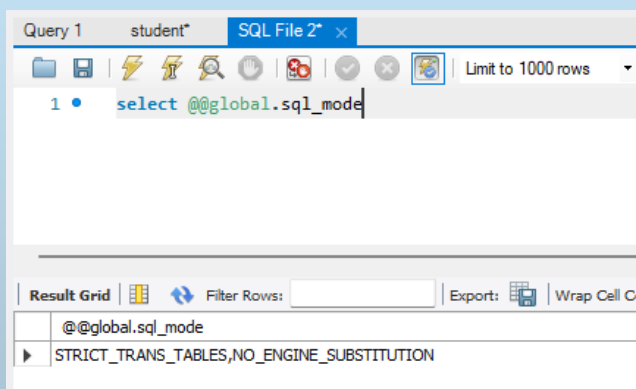
```
Select sname, count(cno) as c_count  
From student, sc  
Where student.sno=sc.sno  
Group By student.sno;
```

```
Select sname, count(cno) as c_count  
From student, sc  
Where student.sno=sc.sno  
Group By student.age;
```

- Oracle和Microsoft SQL Server: **NO!**

补充：再论Group By

- **Select**语句后能否包含不在**Group By**中出现的列？
 - **MySQL**：取决于**sql_mode**是否配置了“**ONLY_FULL_GROUP_BY**”
 - 同时还要满足：**select**后的分组属性函数依赖于**group by**后的属性



补充：几个常用的特殊查询

- **Limit**
- **All, Some, Any**
- **Outer Join**

1、Limit

- **Limit:** 限制返回前多少行 (**MySQL Only**)
- 返回平均成绩排前**10**的学生学号和平均成绩

- **Select sno, avg(score) as avg_score**
From SC
Group By sno
Order By avg_score DESC
Limit 10

MS SQL Server: Select **TOP** ...
Oracle: Where rownum <10

- 返回平均成绩排第**5-10**名的学生学号和平均成绩

- **Select sno, avg(score) as avg_score**
From SC
Group By sno
Order By avg_score DESC
Limit 4,6 --从第5行开始, 返回6行 **offset, rows**

2、All, Some, Any

- 查询工资比sales部门所有人都高的finance部门的员工
 - **Select eno from employee where dept='finance' and salary > ALL (select salary from employee where dept='sales')**
- **All**: 要求子查询中的所有条件都满足
- **Some和Any**: 要求子查询中的某个条件满足即可
 - **Select eno from employee where dept='finance' and salary > SOME (select salary from employee where dept='sales')**

3、Outer Join

- 传统自然连接一般称为**Inner Join**，即连接时只返回匹配的结果。但实际中有时需要同时返回不匹配的结果
- “求供应商的供应商号以及该供应商供应的平均零件数”
 - **Select sno, avg(QTY) as avg_items**
From SPJ
Group By Sno
 - 如果要求没供应零件的供应商也需返回该怎么处理？
 - **Select S.Sno, avg(QTY) as avg_items**
From S LEFT OUTER JOIN SPJ on S.Sno=SPJ.Sno
Group By S.Sno

3、Outer Join

■ Left Outer Join

左外连接

- 左边关系的所有元组都出现在结果关系中
- 对于左边表的一条记录若与右边表有连接结果则直接输出连接结果；若没有连接结果，则除了左边表自己的字段外其余字段都为NULL

Sno	Sname	City
001	S1	HF
002	S2	BJ
003	S3	SH

S

Sno	P#	QTY
001	C1	100
002	C2	300
001	C3	200
002	C1	500

SPJ

- **Select S.sno, avg(QTY) as avg_items**
From S LEFT OUTER JOIN SPJ on S.sno=SPJ.sno
Group By S.sno

S.Sno	avg_items
001	150
002	400
003	

■ Right Outer Join

■ Full Outer Join (MySQL不支持)

补充：在select语句中使用if和case

- 输出每个同学的学号和选修课程数，要求如果没选课则输出选修课程数0

- **Select sno, count(cno) as c_count from SC group by sno**

- **问题：没选课的同学不会输出。解决方法：用left outer join**

- ◆ **Select student.sno, count(cno) as c_count From student
LEFT OUTER JOIN SC on student.sno=sc.sno Group by SC.sno**

- ◆ 下一个问题：没选课的同学输出的c_count是NULL

- ◆ 如果想让没选课的输出数字0，解决方法：用if

- ◆ **Select student.sno,
If(count(cno) is null, 0, count(cno)) as c_count
From student LEFT OUTER JOIN SC on student.sno=sc.sno
Group by SC.sno**

If(条件表达式, true时的值, false时的值)

补充：在select语句中使用if和case

- 输出每个同学的学号和选修课程号与成绩，要求成绩按等级制输出，从E到A+

- **Select sno, cno**

- (Case**

- when score >= 95 then 'A+'**

- when score >= 90 and score < 95 then 'A'**

- when score >= 85 and score < 90 then 'A-'**

- ...**

- Else 'E'**

-) as grade**

- From SC**

- 通过case语句可以定制多样化输出

本章小结

■ SQL数据库

■ DDL:

- **Create Table/Alter Table/Drop Table**

■ DML

- **Insert, Delete, Update**
- **Select:** 基本查询、连接查询、嵌套查询

■ View

- **作用与优点、Create View/Drop View**

本章小结

- **Select** <列名表> ——指定希望查看的列
- From** <表名列表> ——指定要查询的表
- Where** <条件> ——指定查询条件
- Group By** <分组列名表> ——指定要分组的列
- Having** <条件> ——指定分组的条件
- Order By** <排序列名表> ——指定如何排序