

数据库技术与应用

—— 索引与视图

讲师：孙煦雪

浙江传媒学院
COMMUNICATION
UNIVERSITY
OF ZHEJIANG



课程回顾

数据查询

◆ 基本查询

投影查询，选择查询，order by 子句的排序查询，聚集函数和 group by 子句的分组查询

◆ 嵌套查询

带 IN 谓词的子查询、带比较运算符的子查询、带 ANY 或 ALL 的子查询、带 EXISTS 谓词的子查询

◆ 连接查询

等值与非等值连接查询、自身连接、外连接、复合条件连接

◆ 集合查询

内容提要

➤ 索引

- ◆ 概述

- ◆ 操作

➤ 视图

- ◆ 概述

- ◆ 操作及应用

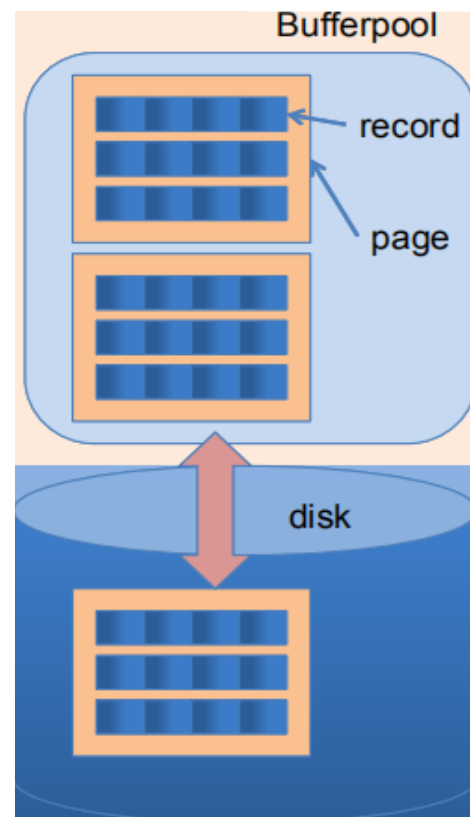
现象

- 数据库以文件方式存储数据到磁盘
- 在程序中使用时，需把磁盘文件中的数据读到内存中
- 磁盘 I/O 操作非常昂贵 <https://www.bilibili.com/video/BV1p7411k7nU>

！ 数据库千万级数据

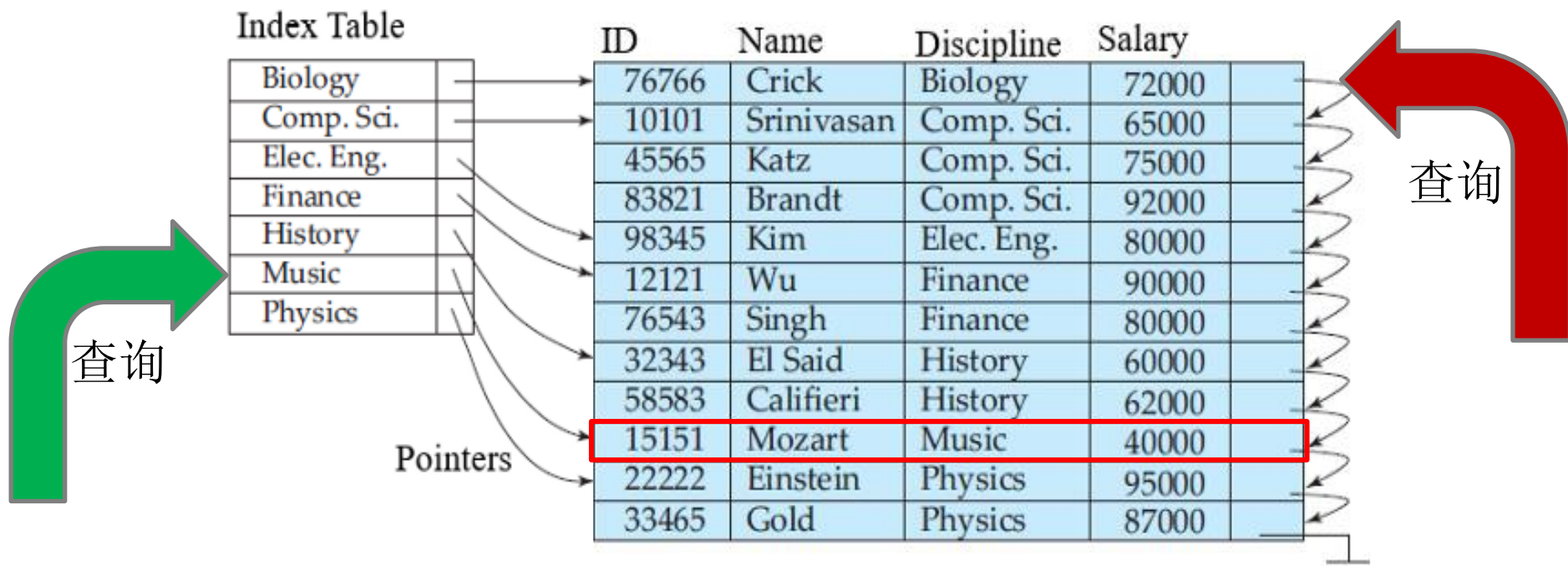


每次查找数据时把
磁盘IO次数控制在一
个很小的数量级



索引的概念

- 索引是一个单独的、物理的数据库结构，是对数据库表中一列或多列值进行排序的结构
- 索引依赖于表建立
- 属于内模式



索引的分类

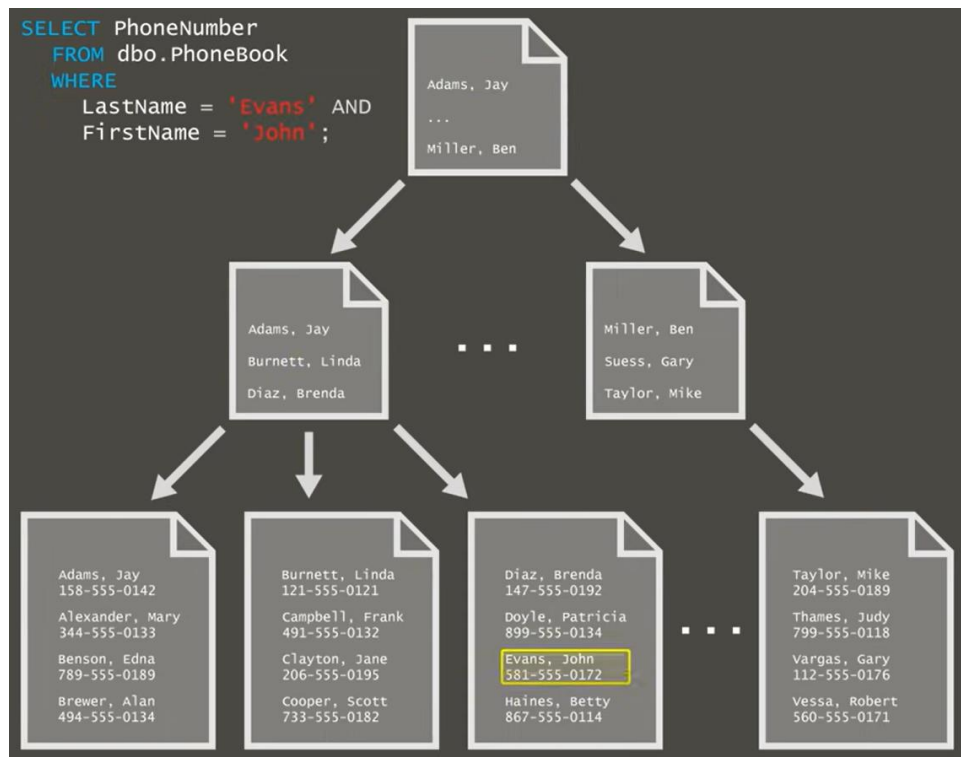
■ 聚集索引 (clustered index)

对表的物理数据页的数据
按列进行排序，然后再重新
存储到磁盘上

➤ 优点：查找速度快

➤ 缺点：

完全重排，占用空间大



◆ 一个表只能有一个聚集索引

◆ SQL Server为主键字段自动创建一个聚集索引

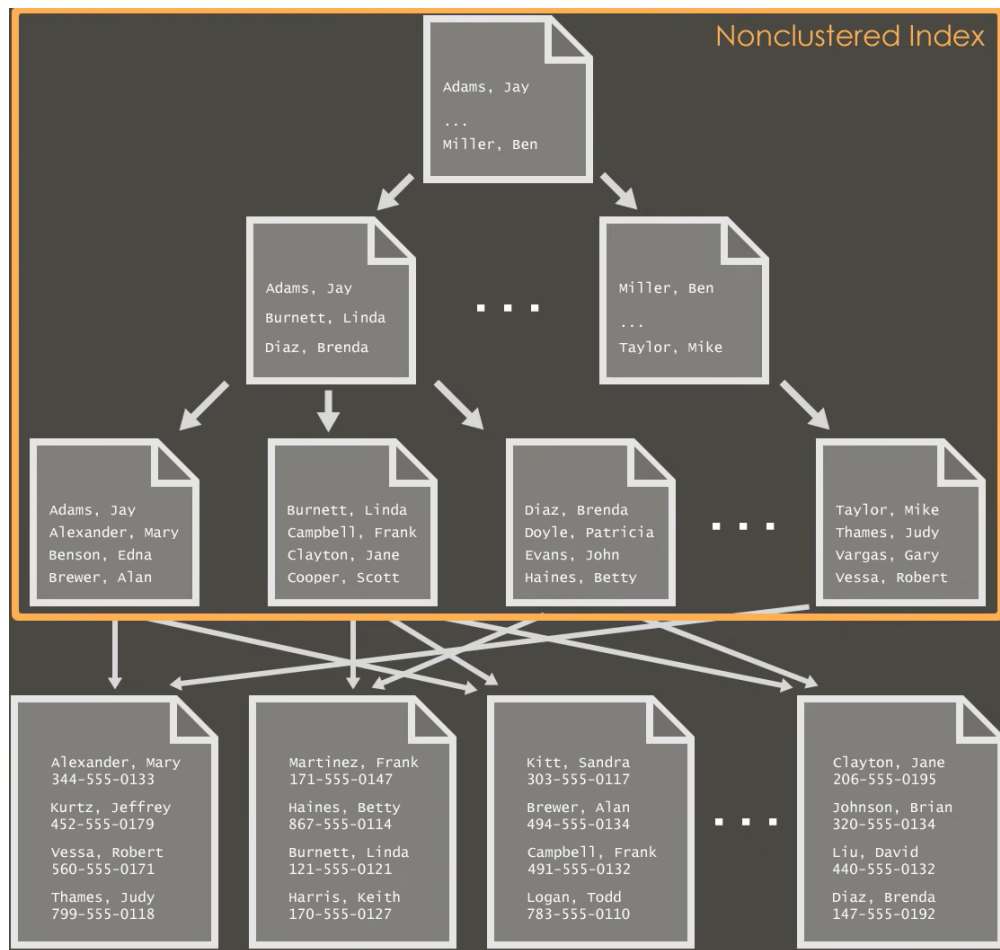
索引的分类

❑ 非聚集索引 (nonclustered index)

具有完全独立于数据行的结构，不会影响数据表中记录的实际存储顺序

- 优点：节省空间
- 缺点：检索效率较低

◆ 一个表中可以创建多个非聚集索引



索引的分类

□ 唯一索引（unique index）

- 不同行关键码不允许存在重复值
- 各种操作受唯一索引的约束

◆ 保证索引列中全部数据是**唯一**的，确保**数据完整性**

◆ 系统自动创建

- 创建 unique 字段，系统则为该字段创建一个唯一索引
- 数据库会自动将**主键**创建成为**唯一索引**

索引的特点



- ✓ 提高查询速度
- ✓ 确保表数据的唯一性
- ✓ 加速表与表之间的连接
- ✓ 提高系统性能



- 占用存储空间
- 在表中插入或更新数据时，将有额外的操作来维护索引

合理地设计和创建索引

性能的提高程度



存储空间、处理资源的代价

索引的使用

□ 下列情况，需要建立索引

- 在 **WHERE** 或连接条件中频繁使用的列
- 列取值范围较大时
- 列包含大量的非空值
- 查询少于2% ~ 4%行的大表

□ 下列条件不要创建索引

- 小表
- 列并不频繁用在查询条件中
- 查询超过2% ~ 4%的行
- 表频繁被更新（如操作日志表、事件表）

内容提要

➤ 索引

- ◆ 概述

- ◆ 操作

➤ 视图

- ◆ 概述

- ◆ 操作及应用

创建索引

- 使用 SSMS 创建索引
- 使用 T-SQL 语句创建索引

CREATE ... INDEX ... ON ...

创建索引 — T-SQL语句

CREATE UNIQUE CLUSTERED 或 NONCLUSTERED

INDEX 索引名

ON 表名(列名1 次序, 列名2 次序 ...)

- UNIQUE : 索引的每一个索引值只对应唯一的数据记录
- CLUSTERED : 要建立的索引是聚集索引

创建索引举例

为学生选课数据库中的student, course, sc三个表建立索引

➤ student 表按学号升序建唯一索引

```
CREATE UNIQUE INDEX idx_stusno ON studinfo(sno);
```

➤ course 表按课程号升序建唯一索引

```
CREATE UNIQUE INDEX idx_cno ON course(cno);
```

➤ sc 表按学号升序和课程号降序建唯一索引

```
CREATE UNIQUE INDEX idx_sc ON sc(sno ASC, cno DESC);
```

创建索引举例

例：

```
create table proj1
```

```
(
```

```
  projid int primary key nonclustered,
```

```
  projname char(4) unique clustered,
```

```
  projdeptno char(10) null
```

```
)
```

```
create unique nonclustered index IX_projdeptno
```

```
  on proj1(projdeptno)
```

删除索引

- 使用 SSMS 删除索引
- 使用 T-SQL 语句创建索引

`DROP INDEX` 索引名

例：删除studinfo表的idx_stusno索引

```
DROP INDEX idx_stusno
```

可删除当前数据库中一个或多个索引，但不适用于删除通过 primary key 或 unique 约束所创建的索引

▣ 索引提高查询速度

<https://jianwai.youdao.com/videocheck/24020401/32797554>

▣ 数据库索引——B树和B+树实现

<https://www.bilibili.com/video/BV1et4y117wc/>

<https://www.bilibili.com/video/BV1yB4y1v7Jy/>

<https://www.bilibili.com/video/BV1UC4y1p7zm/>

<https://www.bilibili.com/video/BV1p7411k7nU>

<https://www.bilibili.com/video/BV1n7411A7x3>

<https://www.bilibili.com/video/BV1DE411i77d/>

内容提要

➤ 索引

- ◆ 概述

- ◆ 操作

➤ 视图

- ◆ 概述

- ◆ 操作及应用

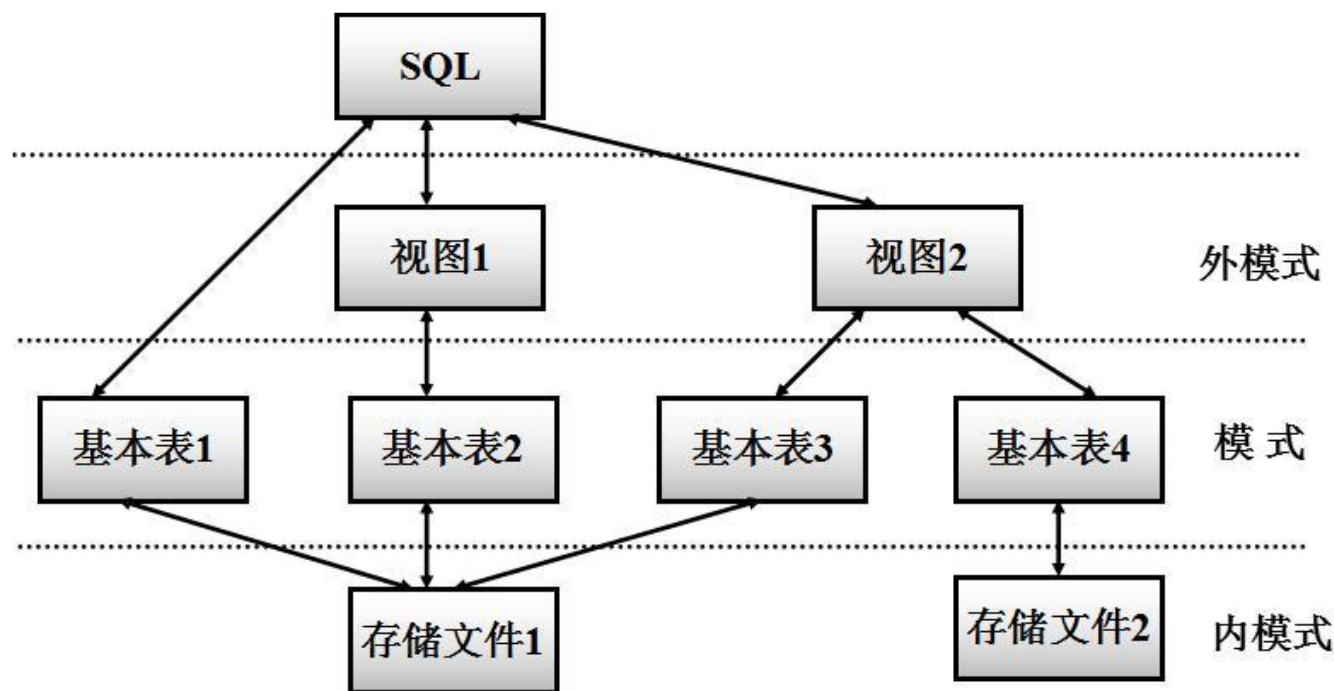
现象

- ❑ 数据表设计上无法为每种关系都创建数据表
 - 能够封装复杂查询，存储查询定义
- ❑ 对数据表中不同的列，不同角色的用户，可能有不同的访问权限
 - 灵活控制数据表的安全性



- ✓ 方便用户
- ✓ 加强数据库安全

视图的特点



- ◆ 是从一个或几个基本表中导出的**虚拟**的表
- ◆ **只存放视图的定义**，不会出现数据冗余
- ◆ 基本表中的数据**变化**，视图中查询出的数据也**随之改变**

视图的作用

- ✓ 视图能够简化用户的操作
- ✓ 视图使用户能以多种角度看待同一数据
- ✓ 视图对重构数据库提供了一定程度的逻辑独立性
- ✓ 视图能够对机密数据提供安全保护

<https://jianwai.youdao.com/videocheck/24020432/32797552>

内容提要

➤ 索引

- ◆ 概述

- ◆ 操作

➤ 视图

- ◆ 概述

- ◆ 操作及应用

 - ◆ 定义视图

 - ◆ 修改和删除视图

 - ◆ 查询视图

 - ◆ 更新视图

定义视图

- 使用 SSMS 创建视图
- 使用 T-SQL 语句创建视图

CREATE ... VIEW... AS...

定义视图

```
CREATE VIEW 视图名(列名1 ...)  
AS 子查询  
WITH CHECK OPTION
```

- ◆ 子查询**不允许**含有ORDER BY子句和DISTINCT短语
- ◆ 组成视图的属性列名**全部省略**或者**全部指定**
- ◆ DBMS 执行 CREATE VIEW语句时**只是把视图的定义存入数据字典**，对视图查询时，按视图的定义从基本表中将数据查出
- ◆ WITH 子句表示透过视图进行增删改操作时，**必须满足**视图定义中的谓词条件

定义视图举例

```
CREATE VIEW DEPT02
AS SELECT emp_no EMPLOYEE_NUMBER,
        emp_name NAME, deptno department
FROM employees
WHERE deptno = 'D002'
```

```
SELECT *
FROM DEPT02
```

 表“DEPT02”中的数据，位置是“empl

	EMPLOYEE_NUMBER	NAME	department
▶	1002	李四	D002
	1003	王五	D002
✱			

减少冗余，简化查询

视图的分类

常见的视图形式

- ◆ 行列子集视图
- ◆ 带表达式视图
- ◆ 分组视图
- ◆ 多表视图
- ◆ 基于视图的视图

行列子集视图

- 从单个表中导出
- 只从基本表中去掉了某些行和某些列，保留了主码

例：建立信息系学生的视图

```
CREATE VIEW V_ISstu AS
SELECT sno, sname, sage
FROM studinfo
WHERE sdept= 'IS'
WITH CHECK OPTION
```



带表达式视图

- 根据应用的需要，设置一些派生属性列（虚拟列）
- 带表达式视图必须明确定义组成视图的各个属性列名

例：建立一个反映产品折扣价的视图

```
CREATE VIEW V_prodprice (prodtid, unitprice, disprice)
AS SELECT productid, unitprice, unitprice*0.88
FROM products
```

例：定义一个反映学生出生年份的视图

```
CREATE VIEW V_BTS (sno,sname,sbirthyear) AS
SELECT sno, sname, 2023-sage
FROM studinfo
```

分组视图

- 用带**聚集函数**和 **GROUP BY 子句**的查询来定义的视图
- 分组视图必须**明确定义**组成视图的各个属性列名

例：建立一个反映每个学生所选课程平均成绩的视图

```
CREATE VIEW averagescore( sno, avgscore ) AS  
SELECT sno, avg(score)  
FROM sc  
GROUP BY sno
```



多表视图

- 基于**多个**基本表的视图
- 若有**同名列**，必须**明确说明**视图的各个属性列

例：建立一个反映每个学生所选课程信息的视图

```
CREATE VIEW stud_detail( sno,sname,cname,score )  
AS SELECT studinfo.sno, sname, cname, score  
FROM studinfo,course,sc  
WHERE studinfo.sno=sc.sno AND  
course.cno=sc.cno
```

基于视图的视图

建立选修1号课程的网络工程系学生视图

```
CREATE VIEW V_S1(sno,sname,score) AS
SELECT studinfo.sno, studinfo.sname, sc.score
FROM studinfo,sc
WHERE sdept= '网络工程' AND
      studinfo.sno=sc.sno AND
      sc.cno= 1
```

建立选修1号课程且成绩在90分以上的网络工程系学生的视图

```
CREATE VIEW V_S2 AS
SELECT sno, sname, score
FROM V_S1
WHERE score>=90
```

内容提要

➤ 索引

- ◆ 概述

- ◆ 操作

➤ 视图

- ◆ 概述

- ◆ 操作及应用

 - ◆ 定义视图

 - ◆ 修改和删除视图

 - ◆ 查询视图

 - ◆ 更新视图

修改视图

- 使用 SSMS 修改视图
- 使用 T-SQL 语句修改视图

ALTER VIEW 视图名

AS 修改的子查询

可加 WITH CHECK OPTION

删除视图

DROP VIEW 视图名 **CASCADE**

- ◆ 从数据字典中**删除**指定的视图**定义**
- ◆ 由该视图导出的其他视图定义仍在数据字典中，但不能使用，必须**显式**删除；使用 **CASCADE** 级联删除语句，把视图和由它导出的所有视图一起删除
- ◆ 删除基本表时，由该表导出的所有视图定义必须**显式**删除

例：

```
DROP VIEW V_prodprice
DROP VIEW stud_detail
```

内容提要

➤ 索引

- ◆ 概述

- ◆ 操作

➤ 视图

- ◆ 概述

- ◆ 操作及应用

 - ◆ 定义视图

 - ◆ 修改和删除视图

 - ◆ 查询视图

 - ◆ 更新视图

查询视图

- 从用户角度：查询视图与查询基本表相同
- DBMS 实现视图查询的方法
 - ◆ 实体化视图（临时表）
 - ◆ 视图消解法
 - 进行有效性检查
 - 把视图定义中的子查询与用户的查询结合起来，
转换成**等价的对基本表的查询**
 - 执行修正后的查询

查询视图

在信息系学生的视图中找出年龄小于20岁的学生

```
SELECT  sno,sage  
FROM    V_ISstu  
WHERE   sage<20
```

视图消解法，转换后的查询语句为：

```
SELECT  sno,sage  
FROM    studinfo  
WHERE   sdept= 'IS' AND sage<20
```

查询视图

在 [averagescore 视图](#) 中查询平均成绩在90分以上的
学生学号和平均成绩

```
SELECT *  
FROM averagescore  
WHERE avgscore >=90
```

视图消解法，转换后的查询语句为

```
SELECT sno,AVG(score)  
FROM sc  
GROUP BY sno  
HAVING AVG(score)>=90;
```

内容提要

➤ 索引

- ◆ 概述

- ◆ 操作

➤ 视图

- ◆ 概述

- ◆ 操作及应用

 - ◆ 定义视图

 - ◆ 修改和删除视图

 - ◆ 查询视图

 - ◆ 更新视图

更新视图

- 从用户角度：更新视图与更新基本表相同
- DBMS 实现视图更新的方法
 - ◆ 视图实体化法
 - ◆ 视图消解法
- 指定 WITH CHECK OPTION 子句后，DBMS 在更新视图时会进行检查，防止用户通过视图对不属于视图范围内的基本表数据进行更新

更新视图的条件

❑ 不允许更新

- 视图从多个基本表使用连接操作导出
- 如果导出的视图使用了分组和聚集操作
- 由视图导出的视图
- 带表达式的视图

❑ 基本表的行列子集视图可以执行更新操作， 可以插入、删除元组

更新视图举例

视图 V_ISstu2	对视图的更新操作	转换成对基表的操作
<pre>CREATE VIEW V_ISstu2 AS SELECT sno,sname,ssex FROM studinfo WHERE sdept='IS' WITH CHECK OPTION</pre>	<pre>UPDATE V_ISstu2 SET sname='李四' WHERE sno='08007'</pre>	<pre>UPDATE studinfo SET sname='李四' WHERE sno='08007' AND sdept='IS'</pre>
	<pre>INSERT INTO V_ISstu2 VALUES('08006', '李林', '女')</pre>	<pre>INSERT INTO studinfo (sno,sname,ssex,sdept) VALUES('08006', '李林', '女', 'IS')</pre>
	<pre>DELETE FROM V_ISstu2 WHERE sno='08006'</pre>	<pre>DELETE FROM studinfo WHERE sno='08006' AND sdept='IS'</pre>

重构中利用视图举例

关系数据库的重构需求，要将一个表分解成多个表

例：

StudentInfo(sno,sname,ssex,sage,sdept,sdeptloca)

将转化成两个关系：

studinfo(sno,sname,ssex,sage,sdept)

departments(sdept,sdeptloca)

重构中利用视图举例

```
CREATE VIEW
```

```
V_student(sno,sname,ssex,  
sage,sdept,sdeptloca)
```

```
AS
```

```
SELECT sno,sname, ssex,  
sage, sdept,sdeptloca FROM  
StudentInfo
```

逻辑独立性
用户和应用不受影响

```
CREATE VIEW
```

```
V_student(sno,sname,ssex,  
sage,sdept,sdeptloca)
```

```
AS
```

```
SELECT sx.sno,sx.sname,  
sx.sage,sx.ssex,sy.sdept,  
sy.sdeptloca
```

```
FROM studinfo sx,  
departments sy
```

```
WHERE sx.sdept=sy.sdept
```

课堂练习 — 比较语句的执行

基本表 emp (id, name, address)

```
CREATE VIEW emp_view  
AS SELECT * FROM emp  
WHERE id=5  
WITH CHECK OPTION
```

```
CREATE VIEW emp_view2  
AS SELECT * FROM emp  
WHERE id=5
```

INSERT INTO

DELETE FROM

UPDATE

emp_view

?

emp_view2

VALUES(8,'李静','江苏')

WHERE id=5

SET id=6 WHERE name='张三'

是否成功执行，会对视图和基表产生什么作用？

视图小结

- ❑ 视图是构建于基本表上的虚表
- ❑ 视图有创建、修改和删除定义的操作
- ❑ 视图中的数据可以更新，但有一定的限制
- ❑ 视图的用途包括简化、多角度数据、重构数据、
数据保护

课程小结

□ 索引

- ◆ 机制、特点、分类和使用
- ◆ 操作：创建和删除

□ 视图

- ◆ 特点、作用和分类
- ◆ 操作：创建、删除、查询和更新

Q & A