

第三章 关系数据库标准语言

— SQL（结构化查询语言）

- **Structured Query Language**

第一节 SQL概述

第二节 SQL数据定义功能

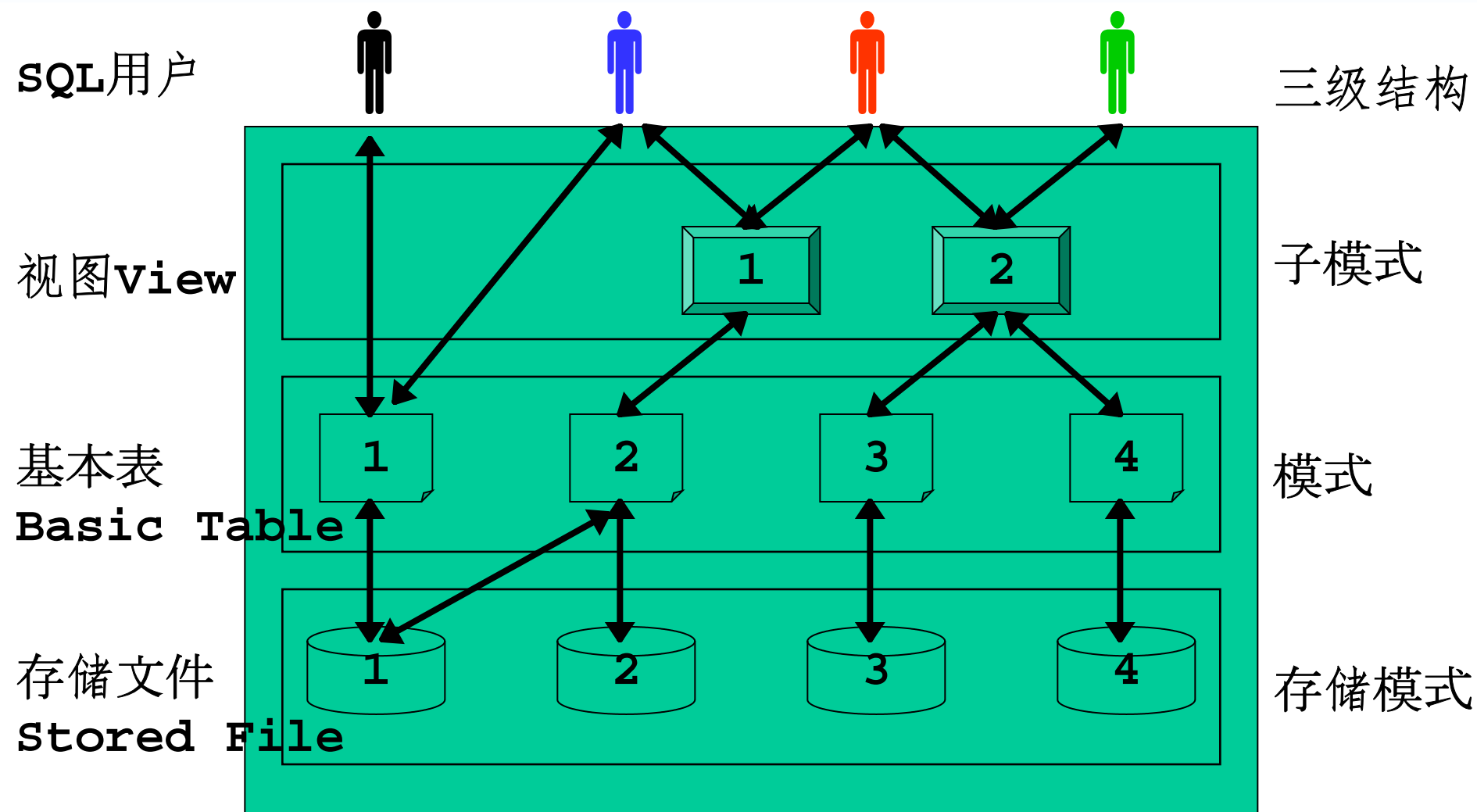
第三节 SQL数据查询功能

第四节 SQL数据更新功能

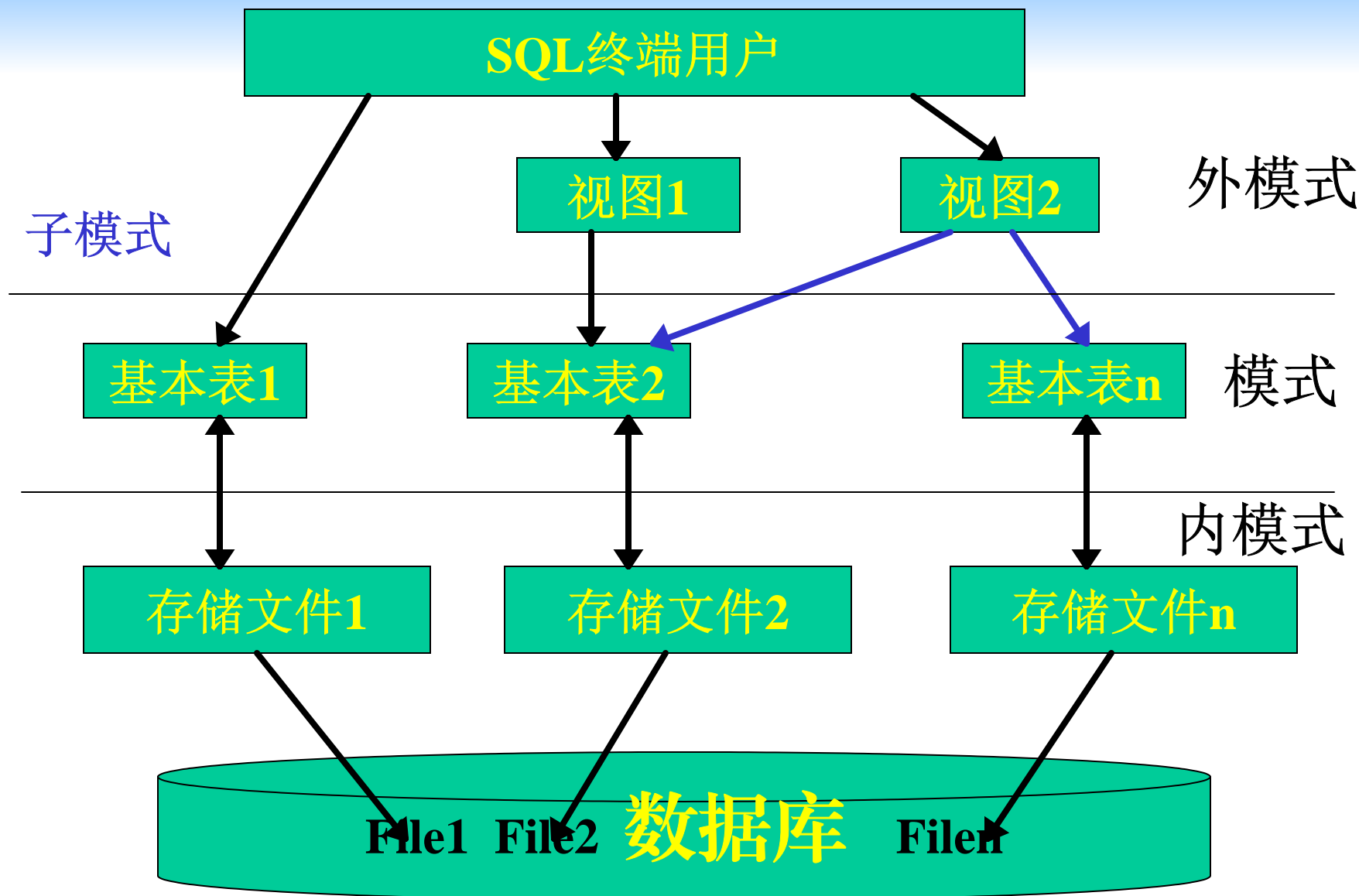
第五节 建立和使用视图

第六节 索引的建立与删除

数据库的体系结构



SQL的三级模式结构





数据库对象

- 基本表 (**BASE TABLE**)

一个关系对应一个基本表

一个或多个基本表对应一个存储文件。

- 视图 (**VIEW**)

一个虚拟的表，是从一个或几个基本表导出的表。

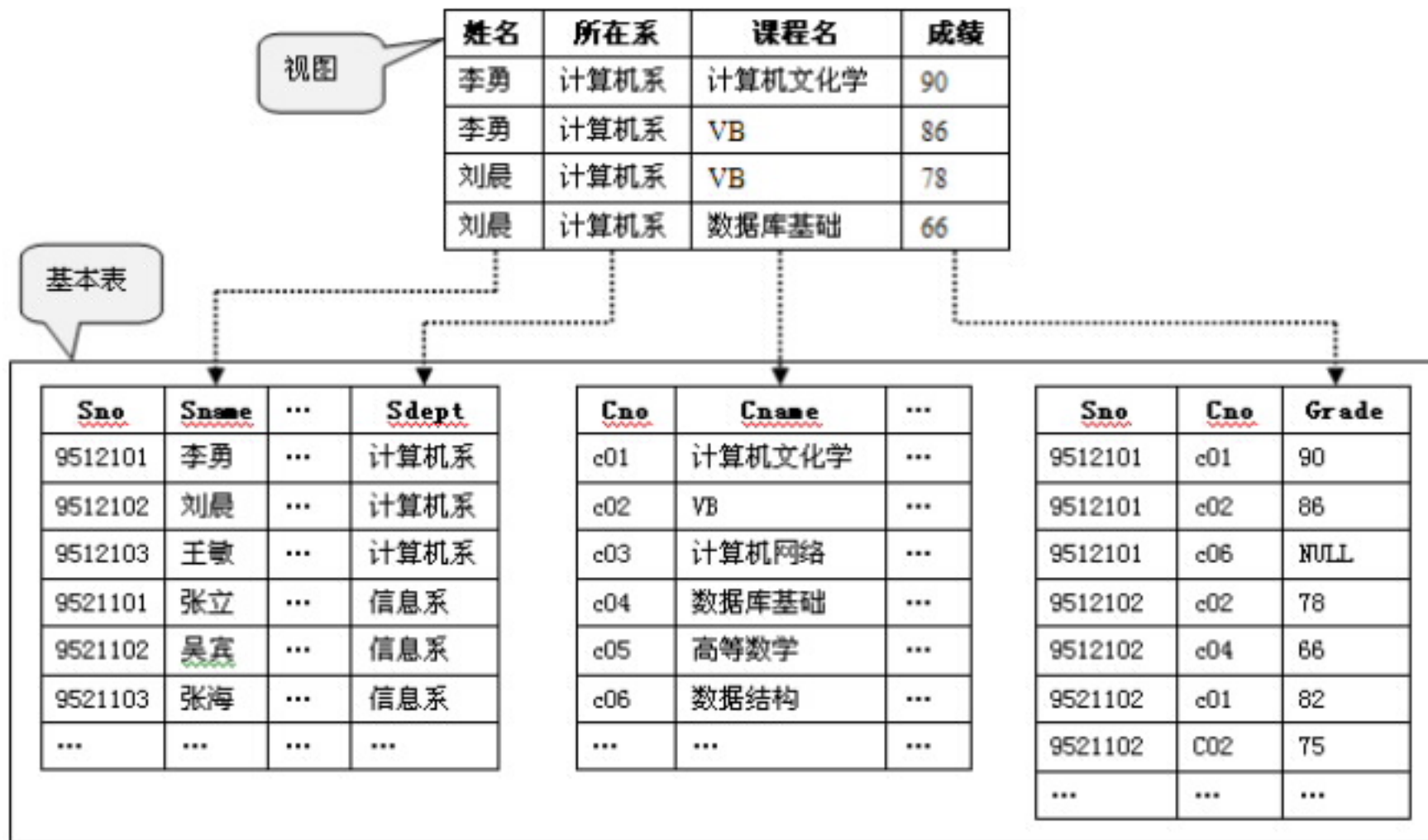
SQL 的数据定义语句

操 作 对 象	操 作 方 式		
	创 建	删 除	修 改
表 (模式)	CREATE TABLE	DROP TABLE	ALTER TABLE
视图 (外模式)	CREATE VIEW	DROP VIEW	删除+创建
索引 (内模式)	CREATE INDEX	DROP INDEX	删除+创建

视图 (VIEW)

- 视图是由从数据库的基本表中选取出来的数据组成的逻辑窗口，
- 是基本表的部分行和列数据的组合。
- 视图是一个虚表。
- 数据库中只存储视图的定义，而不存储视图所包含的数据。

视图与基本表关系





一 视图建立


CREATE VIEW <视图名>[(<视图列表>)]
AS <子查询>

教师表: **T (TNO, TN, TAGE, PROF, DEPT, SAL)**

例1 创建一个计算机系教师情况的视图SUB_T。

```
CREATE VIEW SUB_T  
AS SELECT TNO, TN , PROF  
FROM T WHERE DEPT ='计算机'
```

- ▶ 视图创建后，只在数据字典中存放视图的定义，而其中的子查询SELECT语句并不执行。
- ▶ 只有当用户对视图进行操作时，才按照视图的定义将数据从基本表中取出。



```
CREATE VIEW <视图名>[(<视图列表>)]  
AS <子查询>
```

视图列表不可省略：

- 视图由多个基本表连接得到，在不同的表中存在同名属性列，或定义新的属性列名时，则需指定列名；
- 当视图的列名为表达式或集函数的计算结果时，而不是单纯的属性列名时，则需指明列名。



例2 创建一学生成绩视图V1（包括姓名、课程名及成绩）。

```
CREATE VIEW V1(学生姓名, 课程名, 分数)  
AS SELECT SNAME, CNAME, GRADE  
FROM S, C, SC  
WHERE S.SNO = SC.SNO AND  
SC.CNO = C.CNO
```

视图定义

V1

学生姓名	课程名	分数

基本表

学生表	课程表	学生选课表
SNO	CNO	SNO
SNAME	CNAME	CNO
SAGE	学分	GRADE
SSEX		
SDEPT		



例3 建立学号从97001到97030的学生视图表，表中包括学号和成绩平均值。

```
CREATE VIEW V2 (SNO, GAVG)  
AS SELECT SNO , AVG(GRADE)  
FROM SC  
WHERE SNO between 97001 and 97030  
GROUP BY SNO;
```

视图定义

V2

学号	平均成绩

基本表

学生表	课程表	学生选课表
SNO	CNO	SNO
SNAME	CNAME	CNO
SAGE	学分	GRADE
SSEX		
SDEPT		



二 查询视图

视图定义后，用户就可以像对基本表一样对视图进行查询操作。

例4 查询计算机系讲师的信息。

```
SELECT * FROM SUB_T  
WHERE PROF='讲师'
```

→ **SELECT TNO, TN , PROF FROM T
WHERE PROF='讲师' and DEPT ='计算机'**



例5 查询平均成绩高于80分的学生学号和成绩平均值。

```
SELECT * FROM V2  
WHERE GAVG >80
```




➤ 使用视图优点:

利于数据保密，对不同的用户定义不同的视图，使用户只能看到与自己有关的数据。

简化查询操作，为复杂的查询建立一个视图，用户不必键入复杂的查询语句，只需针对此视图做简单的查询即可。

保证数据的逻辑独立性。对于视图的操作，比如查询，只依赖于视图的定义。当构成视图的基本表要修改时，只需修改视图定义中的子查询部分。

第三章 关系数据库标准语言

— SQL（结构化查询语言）

- **Structured Query Language**

第一节 SQL概述

第二节 SQL数据定义功能

第三节 SQL数据查询功能

第四节 SQL数据更新功能

第五节 建立和使用视图

第六节 索引的建立与删除

一 索引的概念

➤ 创建索引优点：提高查询速度

索引使对数据的查找不需要对整个表进行扫描，就可以在其中找到所需数据。

➤ 建立索引

- DBA或表的属主建立，根据需要建立

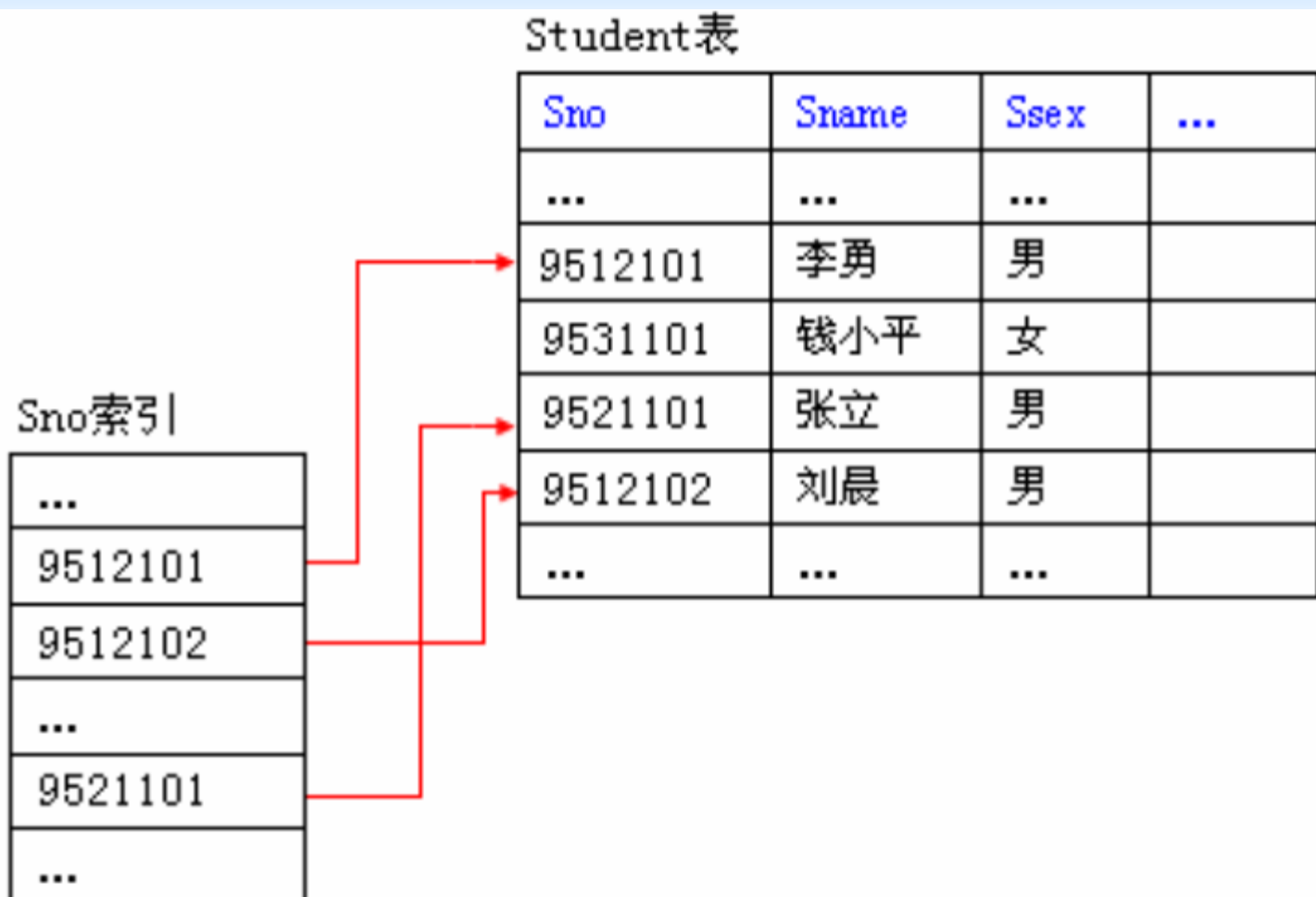
- DBMS自动建立以下列上的索引

 - **PRIMARY KEY**

➤ 使用索引

- DBMS自动选择是否使用索引

索引及数据间的对应关系示意图





– 创建索引的原则

- 在经常用来检索的列上创建索引（如经常在**where**子句中出现的列）。
- 在表的主键、外键上创建索引。
- 在经常用于表间连接的列上建立索引。
- 一般而言，如下情况的列不考虑在其上创建索引：
 - 在查询中几乎不涉及的列。
 - 很少有唯一值的列（即包含太多重复值的列，如性别字段）。
 - 数据类型为**text**、**ntext**或**image**的列。
 - 只有较少行数的表没有必要创建索引。



二 索引的分类

1) 单列索引

- 对基本表的某一单独的列建立索引。

2) 复合索引

- 是将多个属性列组合起来建立的索引。

3) 唯一索引

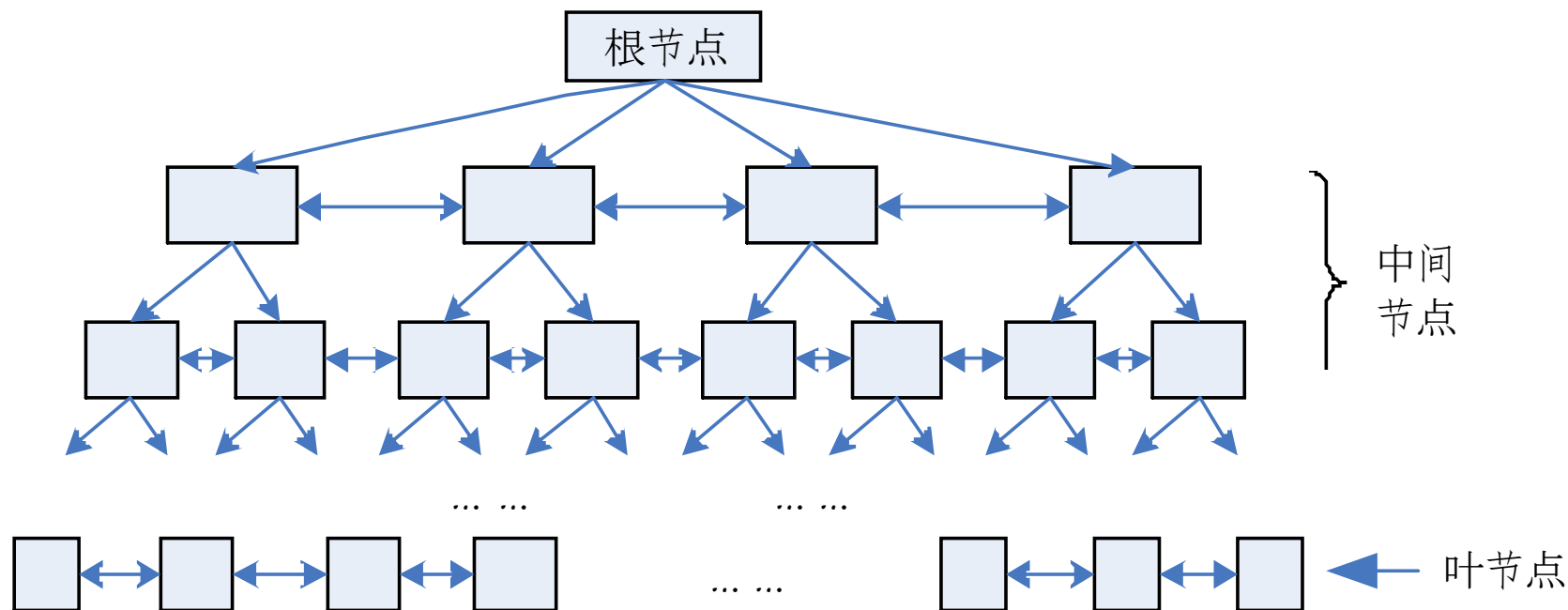
- 表中每一个索引值只对应唯一的数据记录。

4) 聚簇索引与非聚簇索引

将表中的记录在物理数据页中的位置按索引字段值重新排序，再将重排后的结果写回到磁盘上。

- **聚簇索引**：对数据按索引关键字进行物理排序
- **非聚簇索引**：不对数据进行物理排序
- 索引一般采用**B**树结构。

B-树结构

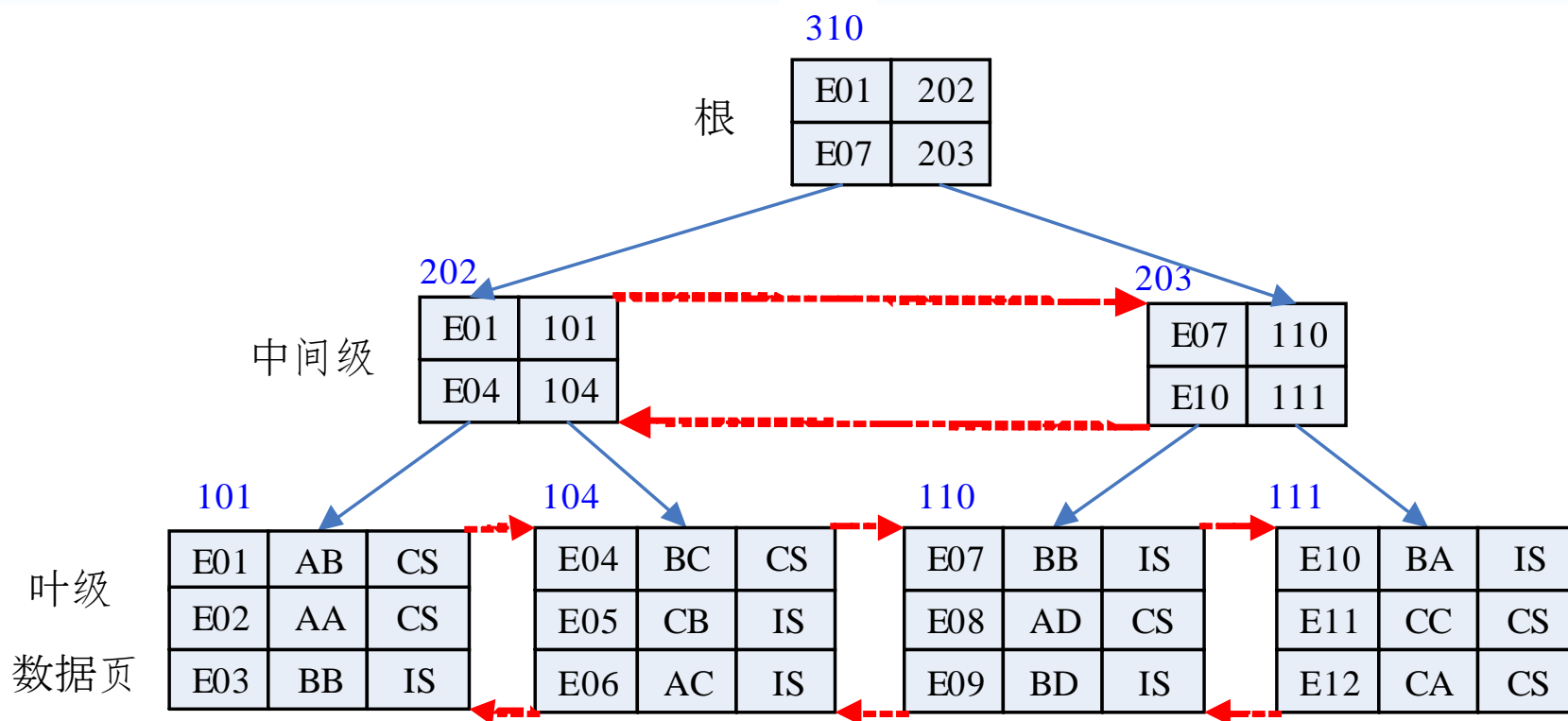


- 聚簇索引的B-树按自下而上建立，最下层的叶级节点存放一条记录，它同时也是数据页。
- 多个数据页生成一个中间层节点的索引页，然后再由数个中间层的节点的索引页合成更上层的索引页，
- 如此上推，直到生成顶层的根节点的索引页。

EMP表数据

eno	ename	dept
E01	AB	CS
E02	AA	CS
E03	BB	IS
E04	BC	CS
E05	CB	IS
E06	AS	IS
E07	BB	IS
E08	AD	CS
E09	BD	IS
E10	BA	IS
E11	CC	CS
E12	CA	CS

聚簇索引示例

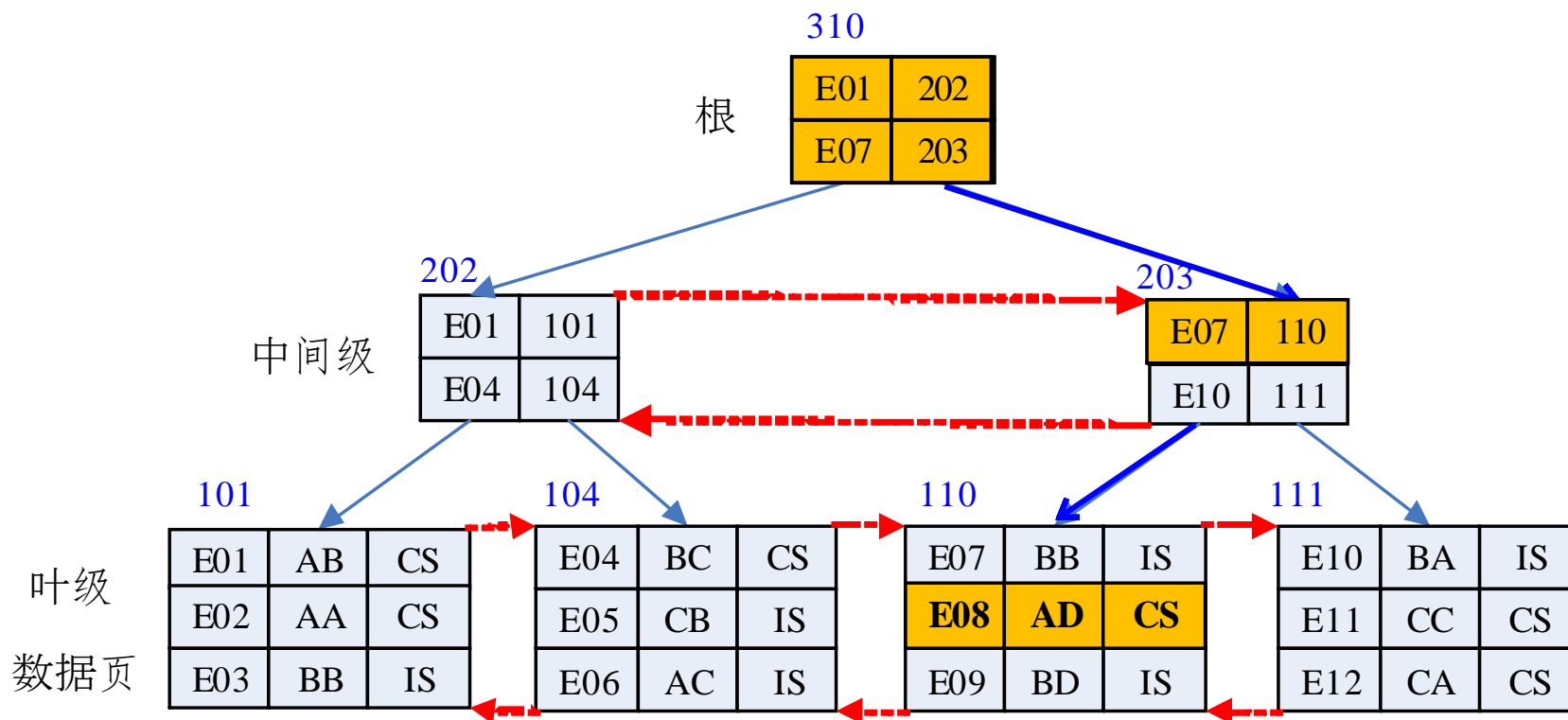


查找过程

- 当在建有聚簇索引的列上查找数据时，首先从聚簇索引树的入口（根节点）开始逐层向下查找
- 直到达到**B**-树索引的叶级，也就是达到了要找的数据所在的数据页
- 数据页中查找所需数据

查找示例

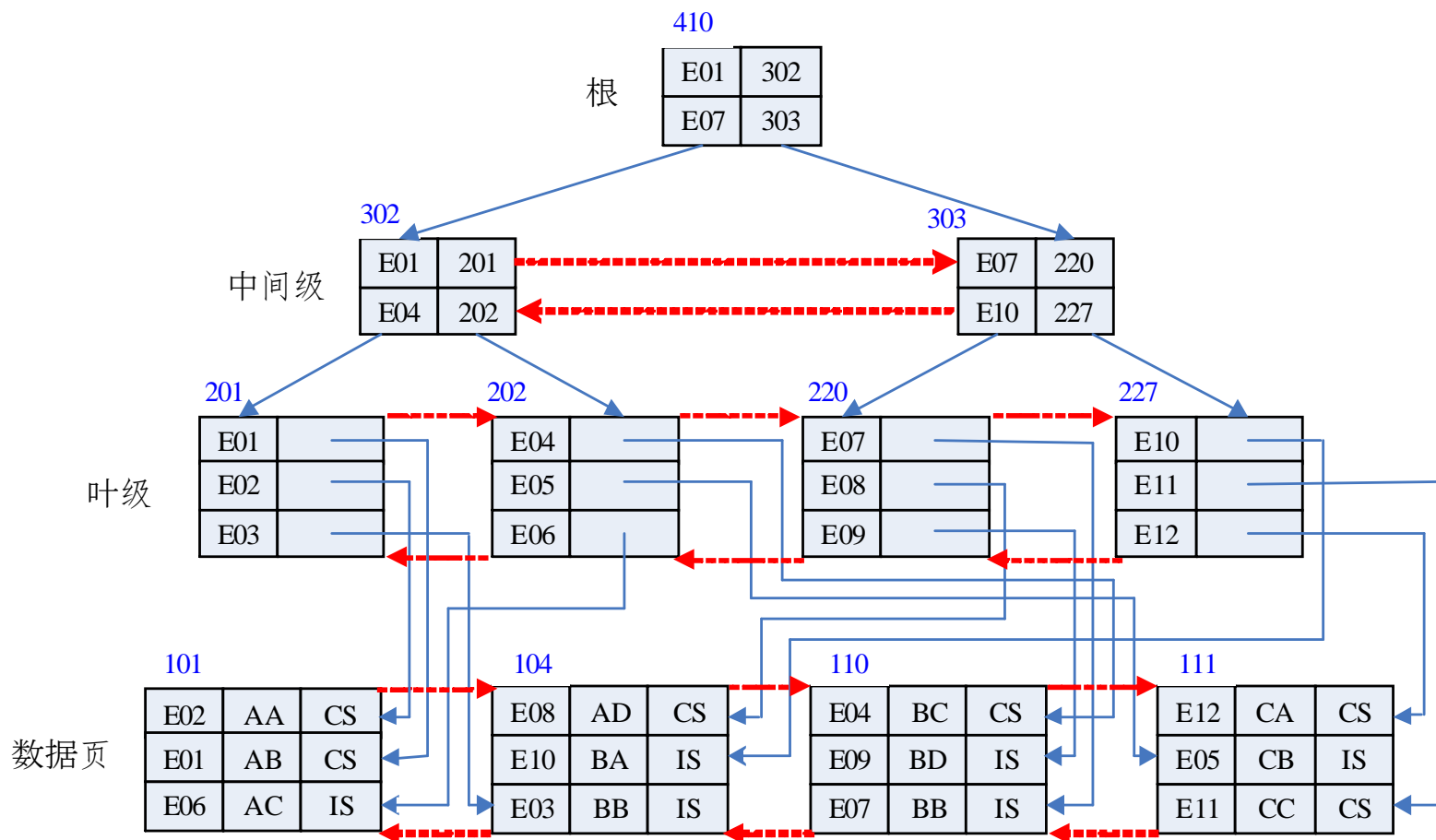
SELECT * FROM emp WHERE eno='E08'



非聚簇索引

- 数据不按非聚簇索引关键字值的顺序排序和存储。
- 叶级节点不是存放数据的数据页。
- 非聚簇索引B树的叶级节点是索引行。每个索引行包含非聚集索引关键字值以及一个或多个行定位器，这些行定位器指向该关键字值对应的数据行

在eno列上建有非聚集索引的情形





三 建立索引

CREATE [**UNIQUE**] [**CLUSTER**] **INDEX** <索引名> **ON**
<表名> (<列名> [次序] [,<列名>] [次序]...)

- **UNIQUE**表明建立唯一索引。
- **CLUSTER**表示建立聚簇索引。
- 次序: **ASC** (升序) 或**DESC** (降序)
- 缺省值为**ASC**



例6 为学生管理数据库中三个表建立索引。其中学生基本表按学号升序建唯一索引，课程表按课程号升序建唯一索引，考试基本表按学号降序和课程号升序建唯一的聚簇索引。

CREATE UNIQUE INDEX Stusno ON 学生(学号);

CREATE UNIQUE INDEX Coucno ON 课程(课程号);

**CREATE UNIQUE CLUSTER INDEX SCno ON 考试(学号
DESC, 课程号 ASC);**

2:表“学生”中的数据，位置是“学生管理”...

学号	姓名	性别	年龄	班级号
010125	曲波	男	19	01001
010126	高天	男	20	01002
010138	张新	女	21	01001
000234	关笑	男	22	00011
020308	王乐	女	18	02010
001025	李明	男	18	00005
020016	张铁	男	19	02001
001004	刘芳	女	20	00011

2:表“考试”中的数据，位置是“学...”

学号	课程号	成绩
020016	A001	80
010126	A002	70
001025	A004	92
010138	A002	85
001004	A004	80
001004	A003	90

2:表“考试”中的数据，位置是“学生管理”...

学号	课程号	成绩
020016	A001	80
010138	A002	85
010126	A002	70
001025	A004	92
001004	A003	90
001004	A004	80



四 删除索引

DROP INDEX <索引名>;

//系统会从数据字典中删除该索引的描述

例7 删除Student表的Stusname索引

DROP INDEX Stusname;