

《数据库系统概论》（王珊.第五版）

★第一章 绪论

❖ 本章内容：数据库系统概述、数据模型、数据库系统结构

❖ 本章主要考点：数据、数据库、★数据库管理系统的概念、概念模型、★ER 图、三种数据模型、数据库的三级模式结构

1、试述数据、数据库、数据库系统、数据库管理系统的概念

- (1) 数据 (Data)：描述事物的符号记录称为数据。
- (2) 数据库 (Data Base，简称 DB)：数据库是长期储存在计算机内的、有组织的、可共享的数据集合。
- (3) 数据库系统 (Data Base System，简称 DBS)：数据库系统是指在计算机系统中引入数据库后的系统构成，一般由数据库、数据库管理系统（及其开发工具）、应用系统、数据库管理员构成。
- (4) 数据库管理系统 (DataBase Management System，简称 DBMS)：数据库管理系统是位于用户与操作系统之间的一层数据管理软件，用于科学地组织和存储数据、高效地获取和维护数据。

2、对于 DB、DBS、DBMS 的关系，下列哪一种描述是正确的 (B)

- A、DB 包括 DBS、DBMS
- B、DBS 包括 DB、DBMS
- C、DBMS 包括 DBS、DB
- D、以上都不对

3、概念模型中的术语：实体、属性、码、域、实体型、实体集

实体：客观存在并可相互区分的事物叫实体。

属性：实体所具有的某一特性，一个实体可由若干个属性来刻画。

码：惟一标识实体的属性集称为码。

域：域是一组具有相同数据类型的值的集合。

实体型：具有相同属性的实体具有相同的特征和性质，用实体名及其属性名集合来抽象和刻画同类实体，称为实体型。

实体集：同型实体的集合称为实体集。

4、目前数据库领域中最常用的逻辑数据模型有 层次模型、网状模型、关系模型 等。

5、数据库管理系统的主要功能有哪些？（冷叶顷）

- (1) 数据库定义功能 (2) 数据存取功能 (3) 数据库运行管理 (4) 数据库的建立和维护功能。

★6、实体联系图（E—R 图）

实体型：用矩形表示，矩形框内写明实体名；

属性：用椭圆形表示，并用无向边将其与相应的实体连接起来；

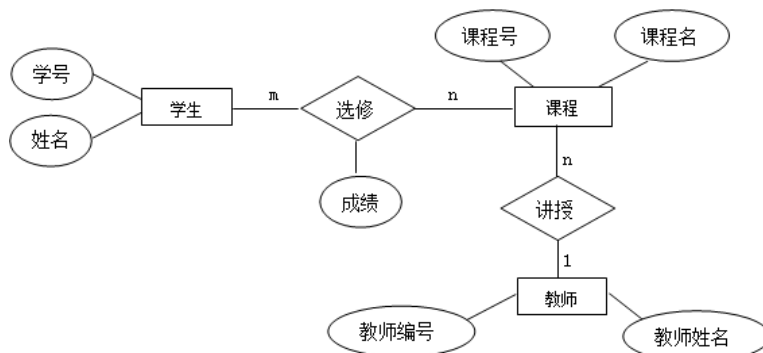
联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标上联系的类型（1: 1 或 1: n 或 m:n）。

▲例：假设教学管理规定：

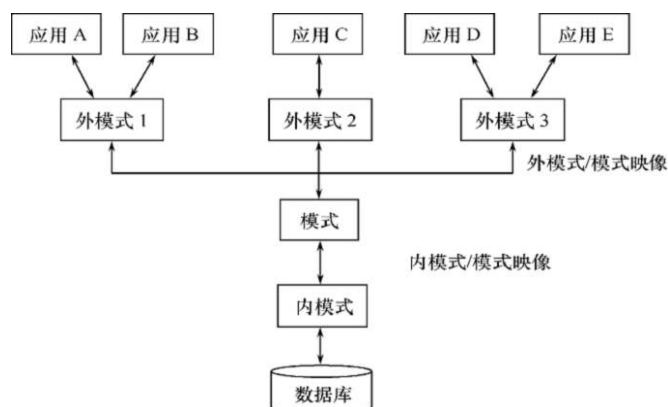
- ①一个学生可选修多门课，一门课有若干学生选修；
- ②一个教师可讲授多门课，一门课只有一个教师讲授；
- ③一个学生选修一门课，仅有一个成绩。

学生的属性有学号、学生姓名；教师的属性有教师编号，教师姓名；课程的属性有课程号、课程名。要求：根据上述语义画出 ER 图，要求在图中画出实体的属性并注明联系的类型；

解：根据以上规则画出对于的 E—R 图如下



7、数据库系统的三级模式结构：



外模式：亦称子模式或用户模式，是数据库用户（包括应用程序员和最终用户）能够看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。

模式：亦称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。模式描述的是数据的全局逻辑结构。外模式涉及的是数据的局部逻辑结构，通常是模式的子集。

内模式：亦称存储模式，是数据在数据库系统内部的表示，即对数据的物理结构和存储方式的描述。

数据库系统在这三级模式之间提供了两层映像：外模式 / 模式映像和模式 / 内模式映像。这两层映像保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

8、对数据库中进行增加记录类型或增加数据项，通过调整下列哪一级映像，可以使应用程序尽可能保持不变，便达到了哪一种数据的独立性 (A)

- A、外模式 / 模式映像，逻辑数据独立性
- B、外模式 / 模式映像，物理数据独立性
- C、模式 / 内模式映像，逻辑数据独立性
- D、模式 / 内模式映像，物理数据独立性

9、数据库的三级模式结构中，描述数据库中全体数据的全局逻辑结构和特征的是 (D)

- A. 外模式 B. 内模式 C. 存储模式 D. 模式

10、层次模型不能直接表示 (C)。

- A. 1 : 1 关系 B. 1 : m 关系 C. m : n 关系 D. 1 : 1 和 1 : m 关系

★第二章 关系数据库（重点）

❖ **本章内容：**关系数据结构及形式化定义、关系操作、关系的完整性、关系代数

❖ **本章主要考点：**概念、关系的完整性、★★关系代数

1、关系数据库管理系统应能实现的专门关系运算包括 (B)。

- A. 排序、索引、统 B. 选择、投影、连接 C. 关联、更新、排序 D. 显示、打印、制表

2、自然连接是构成新关系的有效方法。一般情况下，当对关系 R 和 S 使用自然连接时，要求 R 和 S 含有一个或多个共有的 (D)。

- A. 元组 B. 行 C. 记录 D. 属性

3、关系运算中花费时间可能最长的运算是 (C)。

- A. 投影 B. 选择 C. 笛卡尔积 D. 除

4、关系模式的任何属性 (A)。

- A. 不可再分 B. 可再分 C. 命名在该关系模式中可以不惟一 D. 以上都不是

5、在关系代数运算中，五种基本运算为 (C)。

- A. 并、差、选择、投影、自然连接 B. 并、差、交、选择、投影

- C. 并、差、选择、投影、乘积 D. 并、差、交、选择、乘积

6、要求学生的年龄在 15 至 40 岁之间，是关系模型的哪一种完整性规则 (C)

- A. 参照完整性 B. 实体完整性 C. 用户定义的完整性规则 D. 以上都不是

7、关系 R 的元数为 7，关系 S 的元数是 4，则 $R \bowtie S$ 可以等价于下列哪一种 (B)

$$2 > 3$$

- A. $\sigma_{2>3} (R \times S)$ B. $\sigma_{2>7} (R \times S)$ C. $R \bowtie S$ D. $\sigma_{6<2} (R \times S)$

8、关系的三类完整性约束：

实体完整性规则：若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值

参照完整性规则：若属性（或属性组）F 是基本关系 R 的外码它与基本关系 S 的主码 Ks 相对应（基本关系 R 和 S 不一定是不同的关系），则对于 R 中每个元组在 F 上的值必须为：

- 或者取空值（F 的每个属性值均为空值）
- 或者等于 S 中某个元组的主码值

用户定义的完整性：针对某一具体关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求

★★9、关系代数（冷叶顷）

运算符	含义	运算符	含义
集合运算符	\cup 并 $-$ 差 \cap 交 \times 笛卡尔积	比较运算符	$>$ 大于 \geq 大于等于 $<$ 小于 \leq 小于等于 $=$ 等于 \neq 不等于

运算符	含义	运算符	含义
专门的关系运算符	σ 选择 π 投影 \bowtie 连接 \div 除	逻辑运算符	\neg 非 \wedge 与 \vee 或

(1) $R \cup S$ 仍为 n 目关系，由属于 R 或属于 S 的元组组成

R		
A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S		
A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \cup S$		
A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
a_1	b_3	c_2

(2) $R - S$ 仍为 n 目关系，由属于 R 而不属于 S 的所有元组组成

R		
A	B	C
a ₁	b ₁	c ₁
a ₁	b ₂	c ₂
a ₂	b ₂	c ₁

S		
A	B	C
a ₁	b ₂	c ₂
a ₁	b ₃	c ₂
a ₂	b ₂	c ₁

R-S		
A	B	C
a ₁	b ₁	c ₁

(3) $R \cap S$ 仍为 n 目关系，由既属于 R 又属于 S 的元组组成

R		
A	B	C
a ₁	b ₁	c ₁
a ₁	b ₂	c ₂
a ₂	b ₂	c ₁

S		
A	B	C
a ₁	b ₂	c ₂
a ₁	b ₃	c ₂
a ₂	b ₂	c ₁

$R \cap S$		
A	B	C
a ₁	b ₂	c ₂
a ₂	b ₂	c ₁

(4) 笛卡尔积: $R \times S$ 列: $(n+m)$ 列元组的集合, 元组的前 n 列是关系 R 的一个元组, 后 m 列是关系 S 的一个元组, 行: $k_1 \times k_2$ 个元组

R		
A	B	C
a ₁	b ₁	c ₁
a ₁	b ₂	c ₂
a ₂	b ₂	c ₁

S		
A	B	C
a ₁	b ₂	c ₂
a ₁	b ₃	c ₂
a ₂	b ₂	c ₁

$R \times S$					
R.A	R.B	R.C	S.A	S.B	S.C
a ₁	b ₁	c ₁	a ₁	b ₂	c ₂
a ₁	b ₁	c ₁	a ₁	b ₃	c ₂
a ₁	b ₁	c ₁	a ₂	b ₂	c ₁
a ₁	b ₂	c ₂	a ₁	b ₂	c ₂
a ₁	b ₂	c ₂	a ₁	b ₃	c ₂
a ₁	b ₂	c ₂	a ₂	b ₂	c ₁
a ₂	b ₂	c ₁	a ₁	b ₂	c ₂
a ₂	b ₂	c ₁	a ₁	b ₃	c ₂
a ₂	b ₂	c ₁	a ₂	b ₂	c ₁

(5) 选择:

选择运算符的含义: 在关系 R 中选择满足给定条件的诸元组

$$\sigma_F(R) = \{t | t \in R \wedge F(t) = \text{'真'}\}$$

F : 选择条件, 是一个逻辑表达式

例如: 查询信息系 (IS 系) 全体学生 $\sigma_{\text{Dept} = \text{'IS'}}(\text{Student})$

(6) 投影:

投影运算符的含义: 从 R 中选择出若干属性列组成新的关系

$$\pi_A(R) = \{t[A] | t \in R\} \quad A \text{ 是 } R \text{ 中的属性列}$$

例如: 查询学生的姓名和所在系, 即求 Student 关系上学生姓名和所在系两个属性上的投影

$$\pi_{\text{Sname, Dept}}(\text{Student})$$

(7) 连接:

连接运算的含义: 从两个关系的笛卡尔积中选取属性间满足一定条件的元组

$$R \bowtie_{A \theta B} S = \{ \overset{\frown}{t_r} \overset{\frown}{t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A] \theta t_s[B] \}$$
 分为等值连接、一般连接、自然连接

例如：关系 R 和关系 S 如下所示：

R		
A	B	C
a ₁	b ₁	5
a ₁	b ₂	6
a ₂	b ₃	8
a ₂	b ₄	12

S	
B	E
b ₁	3
b ₂	7
b ₃	10
b ₃	2
b ₅	2

一般连接 $R \bowtie_{C<E} S$ 的结果如下：

$R \bowtie_{C<E} S$				
A	R.B	C	S.B	E
a ₁	b ₁	5	b ₂	7
a ₁	b ₁	5	b ₃	10
a ₁	b ₂	6	b ₂	7
a ₁	b ₂	6	b ₃	10
a ₂	b ₃	8	b ₃	10

等值连接 $R \bowtie_{R.B=S.B} S$ 的结果如下：

A	R.B	C	S.B	E
a ₁	b ₁	5	b ₁	3
a ₁	b ₂	6	b ₂	7
a ₂	b ₃	8	b ₃	10
a ₂	b ₃	8	b ₃	2

自然连接 $R \bowtie S$ 的结果如下：

A	B	C	E
a ₁	b ₁	5	3
a ₁	b ₂	6	7
a ₂	b ₃	8	10
a ₂	b ₃	8	2

(8) 除：

给定关系 R(X, Y) 和 S(Y, Z)，其中 X, Y, Z 为属性组。R 中的 Y 与 S 中的 Y 可以有不同的属性名，但必须出自相同的域集。R 与 S 的除运算得到一个新的关系 P(X)，P 是 R 中满足下列条件的元组在 X 属性列上的投影：元组在 X 上分量值 x 的象集 Y_x 包含 S 在 Y 上投影的集合，记作：

$$R \div S = \{t_r[X] \mid t_r \in R \wedge \pi_Y(S) \subseteq Y_x\}$$

例如：设关系 R、S 分别为下图的(a)和(b)，R÷S 的结果为图(c)

R			S		
A	B	C	B	C	D
a ₁	b ₁	c ₂	b ₁	c ₂	d ₁
a ₂	b ₃	c ₇	b ₂	c ₁	d ₁
a ₃	b ₄	c ₆	b ₂	c ₃	d ₂
a ₁	b ₂	c ₃	(b)		
a ₄	b ₆	c ₆	$R \div S$		
a ₂	b ₂	c ₃	A		
a ₁	b ₂	c ₁	a ₁		
(a)			(c)		

★例：设有一个 SPJ 数据库，包括 S，P，J，SPJ 四个关系模式：

S(SNO, SNAME, STATUS, CITY);

P(PNO, PNAME, COLOR, WEIGHT);

J(JNO, JNAME, CITY);

SPJ(SNO, PNO, JNO, QTY);

供应商表 S 由供应商号码 (SNO)、供应商姓名 (SNAME)、供应商状态 (STATUS)、供应商所在城市 (CITY) 组成；零件表 P 由零件代码 (PNO)、零件名 (PNAME)、颜色 (COLOR)、重量 (WEIGHT) 组成；工程项目表 J 由工程项目代码 (JNO)、工程项目名 (JNAME)、工程项目所在城市 (CITY) 组成；供应情况表 SPJ 由供应商代码 (SNO)、零件代码 (PNO)、工程项目代码 (JNO)、供应数量 (QTY) 组成，表示某供应商供应某种零件给某工程项目的数量为 QTY。

试用关系代数完成如下查询：

- (1) 求供应工程 J1 零件的供应商号码 SNO；
- (2) 求供应工程 J1 零件 P1 的供应商号码 SNO；
- (3) 求供应工程 J1 零件为红色的供应商号码 SNO；
- (4) 求没有使用天津供应商生产的红色零件的工程号 JNO；
- (5) 求至少用了供应商 S1 所供应的全部零件的工程号 JNO。

解：

- (1) 求供应工程 J1 零件的供应商号码 SNO：

$$\pi_{SNO}(\sigma_{JNO='J1'}(SPJ))$$

- (2) 求供应工程 J1 零件 P1 的供应商号码 SNO：

$$\pi_{SNO}(\sigma_{JNO='J1' \wedge PNO='P1'}(SPJ))$$

(3) 求供应工程 J1 零件为红色的供应商号码 SNO:

$\pi_{SNO}(\sigma_{JNO='J1'}(\sigma_{COLOR='红'}(P) \bowtie SPJ))$ 或

$\pi_{SNO}(\sigma_{JNO='J1'}(SPJ) \bowtie (\sigma_{COLOR='红'}(P)))$

(4) 求没有使用天津供应商生产的红色零件的工程号 JNO:

$\pi_{JNO}(SPJ) - \pi_{JNO}(\sigma_{CITY='天津' \wedge COLOR='红'}(S \bowtie SPJ \bowtie P))$ 或

$\pi_{JNO}(SPJ) - \pi_{JNO}(\sigma_{CITY='天津'}(S) \bowtie SPJ \bowtie \sigma_{COLOR='红'}(P))$

(5) 求至少用了供应商 S1 所供应的全部零件的工程号 JNO:

$\pi_{JNO, PNO}(SPJ) \div \pi_{PNO}(\sigma_{SNO='S1'}(SPJ))$

★第三章 关系数据库标准语言 SQL (重点)

❖ 本章内容: SQL 概述、数据定义、数据查询、数据更新、视图

❖ 本章主要考点: 基本概念、★★数据查询的基本语句操作、数据更新语句

1、SQL 语言是 (C) 语言。

A. 层次数据库 B. 网络数据库 C. 关系数据库 D. 非数据库

2、SQL 语言具有两种使用方式, 分别称为交互式 SQL 和 (C)。

A. 提示式 SQL B. 多用户 SQL C. 嵌入式 SQL D. 解释式 SQL

3、假定学生关系是 S(S#, SNAME, SEX, AGE), 课程关系是 C(C#, CNAME, TEACHER), 学生选课关系是 SC(S#, C#, GRADE)。要查找选修“COMPUTER”课程的“女”学生姓名, 将涉及到关系 (D)。

A. S B. SC, C C. S, SC D. S, C, SC

4、如下面的数据库的表中, 若职工表的主关键字是职工号, 部门表的主关键字是部门号, SQL 操作 (B) 不能执行。

A. 从职工表中删除行('025', '王芳', '03', 720)

B. 将行('005', ' ', '乔兴', '04', 750)插入到职工表中

C. 将职工号为, '001'的工资改为 700

D. 将职工号为, '038'的部门号改为'03'

5、在 SQL 语言中, 用来测试一个集合里是否有重复元组存在, 使用下列哪一个关键字 (A)

A. UNIQUE B. DISTINCT C. EXISTS D. NOT IN

★★6、SQL 数据查询语句 (冷叶顷)

SELECT 语句格式:

```
SELECT [ALL|DISTINCT] <目标列表表达式> [, <目标列表表达式>] ...  
FROM <表名或视图名>[, <表名或视图名>] ...  
[ WHERE <条件表达式> ]  
[ GROUP BY <列名 1> [ HAVING <条件表达式> ] ]  
[ ORDER BY <列名 2> [ ASC|DESC ] ];    (注: SQL 语句中不区分大小写)
```

①单表查询

(1) 选择表中的若干列:

查询指定列:

例: 查询全体学生的学号与姓名

```
SELECT Sno, Sname  
FROM Student;
```

查询所有列: (将<目标列表表达式>指定为 *)

例: 查询全体学生的详细记录。

```
SELECT *  
FROM Student;
```

查询结果计算的值:

例: 查全体学生的姓名及其出生年份。

```
SELECT Sname, 2012-Sage    /*假定当年的年份为 2012 年, 减去年龄即得出生年份*/  
FROM Student;
```

(2) 选择表中的若干元组

取消取值重复的行: (指定 DISTINCT 关键词, 去掉表中重复的行)

例: 查询选修了课程的学生号码

```
SELECT DISTINCT Sno    (使用 DISTINCT 关键词取消了重复学号的行)  
FROM SC;
```

查询满足条件的行: (WHERE 语句)

查 询 条 件	谓 词
比 较	=, >, <, >=, <=, !=, <>, !>, !<; NOT+上述比较运算符
确定范围	BETWEEN AND, NOT BETWEEN AND
确定集合	IN, NOT IN
字符匹配	LIKE, NOT LIKE
空 值	IS NULL, IS NOT NULL
多重条件 (逻辑运算)	AND, OR, NOT

例: 查询计算机科学系全体学生的名单。 (比较)

```
SELECT Sname
FROM Student
WHERE Sdept='CS';
```

例：查询年龄在 20~23 岁（包括 20 岁和 23 岁）之间的学生的姓名、系别和年龄（*确定范围*）

```
SELECT Sname, Sdept, Sage
FROM Student
WHERE Sage BETWEEN 20 AND 23;
```

例：查询信息系（IS）、数学系（MA）和计算机科学系（CS）学生的姓名和性别。（*确定集合*）

```
SELECT Sname, Ssex
FROM Student
WHERE Sdept IN ('IS', 'MA', 'CS');
```

例：查询学号为 200215121 的学生的详细情况。（*字符匹配*）

```
SELECT *
FROM Student
WHERE Sno LIKE '200215121';
```

例：某些学生选修课程后没有参加考试，所以有选课记录，但没有考试成绩。查询缺少成绩的学生的学号和相应的课程号。（*空值*）

```
SELECT Sno, Cno
FROM SC
WHERE Grade IS NULL /*不能用 '=' 代替 IS*/
```

例：查询计算机系年龄在 20 岁以下的学生姓名。（*多重条件*）

```
SELECT Sname
FROM Student
WHERE Sdept='CS' AND Sage<20;
```

(3) ORDER BY 子句

可以按一个或多个属性列排序。升序：ASC；降序：DESC；缺省值为升序

例：查询选修了 3 号课程的学生的学号及其成绩，查询结果按分数降序排列。

```
SELECT Sno, Grade
FROM SC
WHERE Cno='3'
ORDER BY Grade DESC;
```

(4) 聚集函数

COUNT ([DISTINCT ALL] *)	统计元组个数
COUNT ([DISTINCT ALL] <列名>)	统计一列中值的个数
SUM ([DISTINCT ALL] <列名>)	计算一列值的总和
AVG ([DISTINCT ALL] <列名>)	计算一列值的平均值
MAX ([DISTINCT ALL] <列名>)	求一列中的最大值

MIN ([DISTINCT|ALL] <列名>) 求一列中的最小值

例：计算 1 号课程的学生平均成绩。

```
SELECT AVG(Grade)
FROM SC
WHERE Cno= '1 ';
```

②多表查询

(1) 等值查询

例：查询每个学生及其选修课程的情况

```
SELECT Student.*, SC.*
FROM Student, SC
WHERE Student.Sno = SC.Sno;
```

(2) 嵌套查询

带有 IN 谓词的子查询：

例：查询与“刘晨”在同一个系学习的学生

```
SELECT Sno, Sname, Sdept
FROM Student
WHERE Sdept IN
      (SELECT Sdept
       FROM Student
       WHERE Sname=' 刘晨 ');
```

带有比较运算符的子查询：（当内查询的结果是一个值时，可以用=代替 IN）

例：查询与“刘晨”在同一个系学习的学生

```
SELECT Sno, Sname, Sdept
FROM Student
WHERE Sdept =
      (SELECT Sdept
       FROM Student
       WHERE Sname=' 刘晨 ');
```

带有 ANY 或 ALL 谓词的子查询

> ANY	大于子查询结果中的某个值
> ALL	大于子查询结果中的所有值
< ANY	小于子查询结果中的某个值
< ALL	小于子查询结果中的所有值
>= ANY	大于等于子查询结果中的某个值
>= ALL	大于等于子查询结果中的所有值
<= ANY	小于等于子查询结果中的某个值
<= ALL	小于等于子查询结果中的所有值
= ANY	等于子查询结果中的某个值
= ALL	等于子查询结果中的所有值（通常没有实际意义）
!= (或<>) ANY	不等于子查询结果中的某个值
!= (或<>) ALL	不等于子查询结果中的任何一个值

例：查询其他系中比计算机科学某一学生年龄小的学生姓名和年龄

```
SELECT Sname, Sage
FROM Student
WHERE Sage < ANY (SELECT Sage
                  FROM Student
                  WHERE Sdept= 'CS ')
AND Sdept <> 'CS ';          /*这句是父查询块中的条件 */
```

带有 EXISTS 谓词的子查询：

相当于存在量词 \exists ，带有 EXISTS 谓词子查询不返回任何数据，只产生逻辑真值“true”或逻辑假值“false”。

例：查询所有选修了 1 号课程的学生姓名。

```
SELECT Sname
FROM Student
WHERE EXISTS
  (SELECT *
   FROM SC
   WHERE Sno=Student.Sno AND Cno= '1 ');
```

(3) 集合查询（冷叶顷）

集合操作的种类：并操作 UNION，交操作 INTERSECT，差操作 EXCEPT

例：查询计算机科学系的学生及年龄不大于 19 岁的学生。（并）

```
SELECT *
FROM Student
WHERE Sdept= 'CS'
UNION
SELECT *
FROM Student
WHERE Sage<=19;
```

例：查询计算机科学系的学生与年龄不大于 19 岁的学生（交）

```

SELECT *
FROM Student
WHERE Sdept='CS'
INTERSECT
SELECT *
FROM Student
WHERE Sage<=19

```

例：查询计算机科学系的学生与年龄不大于 19 岁的学生。（差）

```

SELECT *
FROM Student
WHERE Sdept='CS'
EXCEPT
SELECT *
FROM Student
WHERE Sage <=19;

```

★例题：设学生课程数据库中有三个关系：

学生关系 S (S#, SNAME, AGE, SEX)

学习关系 SC (S#, C#, GRADE)

课程关系 C (C#, CNAME)

其中 S# (学号)、C# (课程号)、SNAME (姓名)、AGE (年龄)、SEX (性别)、GRADE (成绩)、CNAME (课程名)

用 SQL 语句表达下列操作

- (1) 检索选修课程名称为“MATHS”的学生的学号与姓名
- (2) 检索至少学习了课程号为“C1”和“C2”的学生的学号
- (3) 检索年龄在 18 到 20 之间 (含 18 和 20) 的女生的学号、姓名和年龄
- (4) 检索平均成绩超过 80 分的学生学号和平均成绩
- (5) 检索选修了全部课程的学生姓名
- (6) 检索选修了三门课以上的学生的姓名

解： (1) SELECT SNAME,S#

```

FROM S,SC,C

```

```

WHERE S.S#=SC.S#

```

```

AND C.C#=SC.C#

```

```

AND CNAME=' MATHS'

```

```

(2) SELECT S#
      FROM SC
      WHERE CNO='C1' AND S# IN( SELECT S#
                                   FROM SC
                                   WHERE CNO='C2')

```

```

(3) SELECT S#,SNAME,AGE
      FROM S
      WHERE AGE BETWEEN 18 AND 20

```

```

(4) SELECT S# ,AVG(GRADE)
      FROM SC
      GROUP BY S#
      HAVING AVG(GRADE)>80

```

```

(5) SELECT  SNAME
      FROM   S
      WHERE  NOT EXISTS
            (SELECT  *
             FROM    C
             WHERE NOT EXISTS
                  (SELECT  *
                   FROM    SC
                   WHERE S#=S.S#  AND C#=C.C# )
            )

```

```

(6) SELECT SNAME
      FROM S,SC
      WHERE S.S#=SC.S#
      GROUP BY SNAME
      HAVING COUNT(*)>3

```

◎第四章 数据库安全性

❖ **本章内容：**计算机安全性概述、数据库安全性控制、审计、数据加密

❖ **本章主要考点：**基本概念、数据库安全性控制

1、下面哪个不是数据库系统必须提供的数据库控制功能 (B)。

- A. 安全性 B. 可移植性 C. 完整性 D. 并发控制

2、数据库的 (B) 是指数据的正确性和相容性。

- A. 安全性 B. 完整性 C. 并发控制 D. 恢复

3、在数据系统中，对存取权限的定义称为 (B)。

- A. 命令 B. 授权 C. 定义 D. 审计

4、数据库管理系统通常提供授权功能来控制不同用户访问数据的权限，这主要是为了实现数据库的 (D)。

- A. 可靠性 B. 一致性 C. 完整性 D. 安全性

5、保护数据安全性的一般方法是 设置用户标识和存取权限控制。

6、安全性控制的一般方法有 用户标识鉴定、存取控制、审计、数据加密和视图的保护 五级安全措施。

★第六章 关系数据理论（冷叶顷）

❖ **本章内容：**规范化、数据依赖的公理系统、模式分解

❖ **本章主要考点：**函数依赖、多值依赖的概念、★范式判定、Armstrong 公理系统

1、设计性能较优的关系模式称为规范化，规范化主要的理论依据是 (A)。

- A. 关系规范化理论 B. 关系运算理论 C. 关系代数理论 D. 数理逻辑

2、规范化过程主要为克服数据库逻辑结构中的插入异常，删除异常以及 (C) 的缺陷。

- A. 数据的不一致性 B. 结构不合理 C. 冗余度大 D. 数据丢失

3、当关系模式 $R(A, B)$ 已属于 3NF，下列说法中 (B) 是正确的。

- A. 它一定消除了插入和删除异常 B. 仍存在一定的插入和删除异常
C. 一定属于 BCNF D. A 和 B 都是

4、在关系 DB 中，任何二元关系模式的最高范式必定是 (D)。

A. 1NF B. 2NF C. 3NF D. BCNF

5、在关系模式 R 中，若其函数依赖集中所有候选关键字都是决定因素，则 R 最高范式是 (C)。

A. 2NF B. 3NF C. 4NF D. BCNF

6、根据关系数据库规范化理论，关系数据库中的关系要满足第一范式。下面“部门”关系中，因哪个属性而使它不满足第一范式? (B)。

部门(部门号，部门名，部门成员，部门总经理)

A. 部门总经理 B. 部门成员 C. 部门名 D. 部门号

7、函数依赖的概念

函数依赖的定义： 设 $R(U)$ 是一个属性集 U 上的关系模式， X 和 Y 是 U 的子集。若对于 $R(U)$ 的任意一个可能的关系 r ， r 中不可能存在两个元组在 X 上的属性值相等，而在 Y 上的属性值不等，则称“ X 函数确定 Y ”或“ Y 函数依赖于 X ”，记作 $X \rightarrow Y$ 。即只要 X 上的属性值相等， Y 上的值一定相等。

●平凡的函数依赖

在关系模式 $R(U)$ 中，对于 U 的子集 X 和 Y ，

如果 $X \rightarrow Y$ ，但 $Y \subseteq X$ ，则称 $X \rightarrow Y$ 是平凡的函数依赖

●非平凡的函数依赖

在关系模式 $R(U)$ 中，对于 U 的子集 X 和 Y ，

若 $X \rightarrow Y$ ，但 $Y \not\subseteq X$ ，则称 $X \rightarrow Y$ 是非平凡的函数依赖

例：在关系 $SC(\underline{Sno}, \underline{Cno}, Grade)$ 中，

非平凡函数依赖： $(Sno, Cno) \rightarrow Grade$

平凡函数依赖： $(Sno, Cno) \rightarrow Sno$

$(Sno, Cno) \rightarrow Cno$

●部分函数依赖：若 $X \rightarrow Y$ ，但 Y 不完全依赖于 X ，则称 Y 对 X 部分函数依赖。记作 $X \xrightarrow{p} Y$

●完全函数依赖：在 $R(U)$ 中，如果 $X \rightarrow Y$ ，并且对于 X 的任何一个真子集 X' ，都有 $X' \not\rightarrow Y$ ，则

称 Y 对 X 完全函数依赖。记作 $X \xrightarrow{f} Y$ 。

例：学校教务的数据库：学生的学号 (Sno)、所在系 ($Sdept$) 系主任姓名 ($Mname$)、课程号 (Cno) 成绩 ($Grade$)

$U = \{ Sno, Sdept, Mname, Cno, Grade \}$

$(Sno, Cno) \rightarrow Grade$ 是完全函数依赖,

$(Sno, Cno) \rightarrow Sdept$ 是部分函数依赖

因为 $Sno \rightarrow Sdept$ 成立, 且 Sno 是 $(Sno, Sanme)$ 的真子集

●传递函数依赖: 在 $R(U)$ 中, 如果 $X \rightarrow Y$, ($Y \subsetneq X$), $Y \not\rightarrow X$, $Y \rightarrow Z$, Z 不属于 Y , 则称 Z 对 X 传递函数依赖。

例: 在关系 $Std(Sno, Sdept, Mname)$ 中, 有:

$Sno \rightarrow Sdept, Sdept \rightarrow Mname$

$Mname$ 传递函数依赖于 Sno

$1NF \supset 2NF \supset 3NF \supset BCNF \supset 4NF \supset 5NF$

8、第一范式~BCNF

第一范式 (1NF): 若关系模式 R 的每一个分量是不可再分的数据项, 则关系模式 R 属于第一范式。

第二范式 (2NF): 若关系模式 $R \in 1NF$, 且每一个非主属性完全函数依赖于码, 则关系模式 $R \in 2NF$ 。(即 $1NF$ 消除了非主属性对码的部分函数依赖则成为 $2NF$)。

例: 关系模式 $S-L-C(Sno, Sdept, Sloc, Cno, Grade)$

$Sloc$ 为学生住处, 假设每个系的学生住在同一个地方

函数依赖包括 $(Sno, Cno) \xrightarrow{F} Grade$

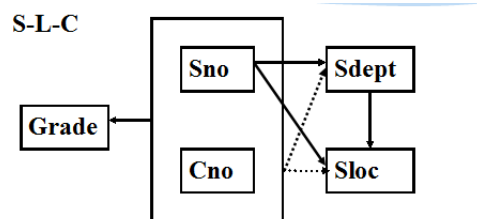
$Sno \rightarrow Sdept$

$(Sno, Cno) \xrightarrow{P} Sdept$

$Sno \rightarrow Sloc$

$(Sno, Cno) \xrightarrow{P} Sloc$

$Sdept \rightarrow Sloc$

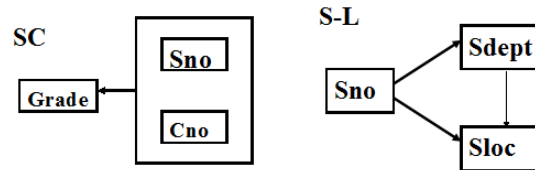


由于非主属性 $Sdept$ 和 $Sloc$ 并不完全函数依赖于码 (Sno, Cno) , 所以不是 $2NF$
解决方法:

$S-L-C$ 分解为两个关系模式, 以消除这些部分函数依赖, 这就属于 $2NF$ 了

SC $(Sno, Cno, Grade)$

S-L $(Sno, Sdept, Sloc)$



第三范式 (3NF)：关系模式 $R<U, F>$ 中若不存在这样的码 X 、属性组 Y 及非主属性 Z (Z 不是 Y 的子集)使得 $X \rightarrow Y, Y \not\rightarrow X, Y \rightarrow Z$ 成立, 则称 $R<U, F> \in 3NF$

(每一个非主属性既不部分依赖于码又不传递依赖于码)

例：上例中, 在 S-L 中存在非主属性对码的传递函数依赖, 所以不属于 3NF

解决方法：采用投影分解法, 把 S-L 分解为两个关系模式, 以消除传递函数依赖:

S-D (Sno, Sdept) 码为 Sno D-L (Sdept, Sloc) 码为 Sdept

BCNF：关系模式 $R<U, F> \in 1NF$ 。若 $X \rightarrow Y$ 且 Y 不是 X 的子集时, X 必含有码, 则

$R<U, F> \in BCNF$ (即每一个决定属性因素都包含码)

BCNF 性质：若 $R \in BCNF$

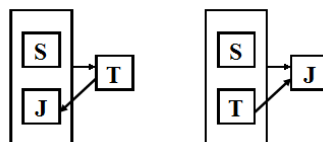
- ①所有非主属性对每一个码都是完全函数依赖
- ②所有的主属性对每一个不包含它的码, 也是完全函数依赖
- ③没有任何属性完全函数依赖于非码的任何一组属性

例：在关系模式 STJ (S, T, J) 中, S 表示学生, T 表示教师, J 表示课程。

每一教师只教一门课。每门课由若干教师教, 某一学生选定某门课, 就确定了一个固定的教师。

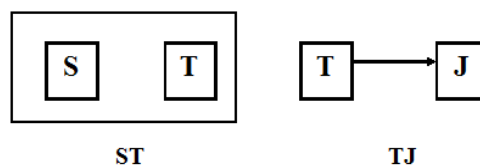
某个学生选修某个教师的课就确定了所选课的名称

函数依赖： $(S, J) \rightarrow T, (S, T) \rightarrow J, T \rightarrow J$ (S, J)和(S, T)都是候选码



$STJ \in 3NF$ (S, J)和(S, T)都可以作为候选码, S、T、J 都是主属性, 没有任何非主属性对码传递依赖或部分依赖。但 STJ 不是 BCNF, 因为 T 是决定因素, T 不包含码

解决方法：将 STJ 分解为二个关系模式: $ST(S, T) \in BCNF, TJ(T, J) \in BCNF$



关于第四范式和多值依赖的内容, 大家按书复习

★例题：设有如下关系 R

课程名	教师名	教师地址
C1	刘爽英	D1
C2	乔志伟	D2
C3	于得水	D2
C4	井超	D2

- (1) 它为第几范式? 为什么?
- (2) 是否存在删除操作异常?若存在, 则说明是在什么情况下发生的?
- (3) 将它分解为高一级范式, 分解后的关系是如何解决分解前可能存在的删除操作异常问题?

解: (1) 它是 2NF。因为 R 的候选关键字为“课程名”。依赖关系: 课程名 \rightarrow 教师名, 教师名 \rightarrow 课程名, 教师名 \rightarrow 教师地址, 所以课程名 \rightarrow 教师地址。即存在非主属性“教师地址”对候选关键字课程名的传递函数, 因此 R 不是 3NF。但: 因为不存在非主属性对候选关键字的部分函数依赖, 所以 R 是 2NF。

(2) 存在。当删除某门课程时, 会删除不该删除的教师的有关信息。

(3) 分解为高一级范式如图所示。

R1 (课程名, 教师名) R2 (教师名, 教师地址)

课程号	教师名
C1	刘爽英
C2	乔志伟
C3	于得水
C4	井超

R1

教师名	教师地址
刘爽英	D1
乔志伟	D2
于得水	D2

R2

分解后, 若删除课程数据时, 仅对关系 R1 操作, 教师地址信息在关系 R2 中仍然保留, 不会丢失教师方面的信息。

★第七章 数据库设计（冷叶顷）

❖ 本章内容：数据库设计概述、数据库设计的 6 个步骤

❖ 本章主要考点：★数据库设计的 6 个步骤、ER 图合并出现的冲突与解决方法、★ER 图转换成关系模式

★1、简述数据库设计的主要步骤。

(1) 需求分析 (2) 概念结构设计 (3) 逻辑结构设计 (4) 数据库物理设计
(5) 数据库实施 (6) 数据库运行和维护。

2、ER 图合并汇总会出现什么冲突？怎么解决冲突？

①出现的冲突有：属性冲突、命名冲突、结构冲突
②属性冲突和命名冲突通常用讨论、协商等行政手段解决。结构冲突通常根据应用语义对实体联系的类型进行综合或调整。

3、在数据库设计中，用 E-R 图来描述信息结构但不涉及信息在计算机中的表示，它是数据库设计的 (B) 阶段。

A. 需求分析 B. 概念设计 C. 逻辑设计 D. 物理设计

4、在关系数据库设计中，设计关系模式是 (C) 的任务。

A. 需求分析阶段 B. 概念设计阶段 C. 逻辑设计阶段 D. 物理设计阶段

5、数据库物理设计完成后，进入数据库实施阶段，下列各项中不属于实施阶段的工作是 (B)。

A. 建立库结构 B. 扩充功能 C. 加载数据 D. 系统调试

6、从 E-R 模型关系向关系模型转换时，一个 M:N 联系转换为关系模型时，该关系模式的关键字是 (C)。

A. M 端实体的关键字 B. N 端实体的关键字
C. M 端实体关键字与 N 端实体关键字组合 D. 重新选取其他属性

7、当局部 E-R 图合并成全局 E-R 图时可能出现冲突，不属于合并冲突的是 (B)。

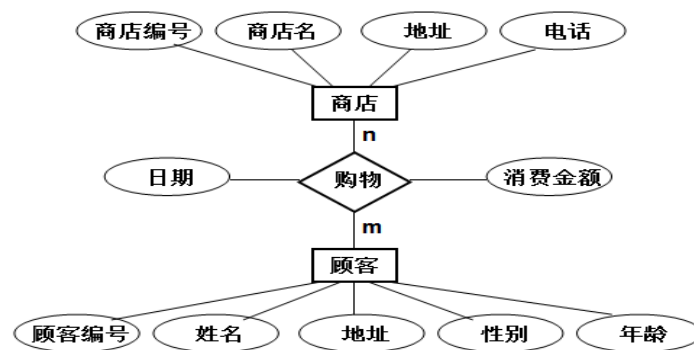
A. 属性冲突 B. 语法冲突 C. 结构冲突 D. 命名冲突

8、数据流程图（DFD）是用于描述结构化方法中 (C) 阶段的工具。

A. 可行性分析 B. 详细设计 C. 需求分析 D. 程序编码

★9、设有商店和顾客两个实体，“商店”有属性商店编号、商店名、地址、电话，“顾客”有属性顾客编号、姓名、地址、年龄、性别。假设一个商店有多个顾客购物，一个顾客可以到多个商店购物，顾客每次去商店购物有一个消费金额和日期。试画出 E-R 图，并注明属性和联系，并将其转换为关系模型。

解：E-R 图



关系模型：顾客（顾客编号，姓名，地址，年龄，性别）

商店（商店编号，商店名，地址，电话）

购物（顾客编号，商店名称，日期，消费金额）

◎第十章 数据库恢复技术（冷叶顷）

❖ 本章内容：事务的基本概念、故障的种类、恢复策略

❖ 本章主要考点：基本概念、恢复策略

1、(C) 是 DBMS 的基本单位，它是用户定义的一组逻辑一致的程序序列。

- A. 程序 B. 命令 C. 事务 D. 文件

2、事务的原子性是指 (A)。

- A. 事务中包括的所有操作要么都做，要么都不做
B. 事务一旦提交，对数据库的改变是永久的
C. 一个事务内部的操作及使用的数据对并发的其他事务是隔离的
D. 事务必须是使数据库从一个一致性状态变到另一个一致性状态

3、若数据库中只包含成功事务提交的结果，则此数据库就称为处于 (B) 状态。

- A. 安全 B. 一致 C. 不安全 D. 不一致

4、若系统在运行过程中，由于某种原因，造成系统停止运行，致使事务在执行过程中以非控制

方式终止，这时内存中的信息丢失，而存储在外存上的数据未受影响，这种情况称为(B)。

- A. 事务故障 B. 系统故障 C. 介质故障 D. 运行故障

5、(B)用来记录对数据库中数据进行的每一次更新操作。

- A. 后援副本 B. 日志文件 C. 数据库 D. 缓冲区

6、用于数据库恢复的重要文件是(C)。

- A. 数据库文件 B. 索引文件 C. 日志文件 D. 备注文件

7、数据库恢复的基础是利用转储的冗余数据。这些转储的冗余数据包括(C)。

- A. 数据字典、应用程序、审计档案、数据库后备副本
B. 数据字典、应用程序、日志文件、审计档案
C. 日志文件、数据库后备副本 D. 数据字典、应用程序、数据库后备副本

8、事务的特性为原子性、隔离性、一致性、持续性。

9、数据库系统中的故障包括事务内部的故障、系统故障、介质故障、计算机病毒。

◎第十一章 并发控制

❖ **本章内容：**并发控制的概述、封锁、活锁和死锁、两段锁协议

❖ **本章主要考点：**封锁中读锁和写锁的概念、活锁和死锁的概念、两段锁协议

1、设有两个事务 T1、T2，其并发操作如右图所示，下面评价正确的是(B)。

- A. 该操作不存在问题 B. 该操作丢失修改
C. 该操作不能重复读 D. 该操作读“脏”数据

2、解决并发操作带来的数据不一致性总是普遍采用(A)。

- A. 封锁 B. 恢复 C. 存取控制 D. 协商

3、若事务 T 对数据 R 已经加 X 锁，则其他事务对数据 R (D)。

- A. 可以加 S 锁不能加 X 锁 B. 不能加 S 锁可以加 X 锁
C. 可以加 S 锁也可以加 X 锁 D. 不能加任何锁

T1	T2
①读 A=10	读 A=10 A=A-8 写回
②	
③A=A-5 写回	
④	

4、并发操作会带来哪些数据不一致性 (D)。

A. 丢失修改、不可重复读、脏读、死锁 B. 不可重复读、脏读、死锁

C. 丢失修改、脏读、死锁 D. 丢失修改、不可重复读、脏读

5、什么是封锁？

封锁就是事务 T 在对某个数据对象操作之前，先向系统发出请求，对其加锁

6、什么是活锁？什么是死锁？

活锁：事务 T1 封锁了数据 R，事务 T2 又请求封锁 R，于是 T2 等待。T3 也请求封锁 R，当 T1 释放了 R 上的封锁之后系统首先批准了 T3 的请求，T2 仍然等待。T4 又请求封锁 R，当 T3 释放了 R 上的封锁之后系统又批准了 T4 的请求……T2 有可能永远等待，这就是活锁的情形。避免方法：采用先来先服务的策略。

死锁：事务 T1 封锁了数据 R1，T2 封锁了数据 R2，T1 又请求封锁 R2，因 T2 已封锁了 R2，于是 T1 等待 T2 释放 R2 上的锁，接着 T2 又申请封锁 R1，因 T1 已封锁了 R1，T2 也只能等待 T1 释放 R1 上的锁，这样 T1 在等待 T2，而 T2 又在等待 T1，T1 和 T2 两个事务永远不能结束，形成死锁。

7、预防死锁的方法有一次性封锁法、顺序封锁法。

8、死锁的诊断与解除方法有超时法、等待图法。

9、试述两段锁协议的概念

所谓两段锁协议是指所有事务必须分两个阶段对数据项加锁和解锁。

获得封锁：在对任何数据进行读、写操作之前，事务首先要获得对该数据的封锁

释放封锁：在释放一个封锁之后，事务不再申请和获得任何其他封锁

若并发事务都遵守两段锁协议，则对这些事务的任何并发调度策略都是可串行化的

若并发事务的一个调度是可串行化的，不一定所有事务都符合两段锁协议