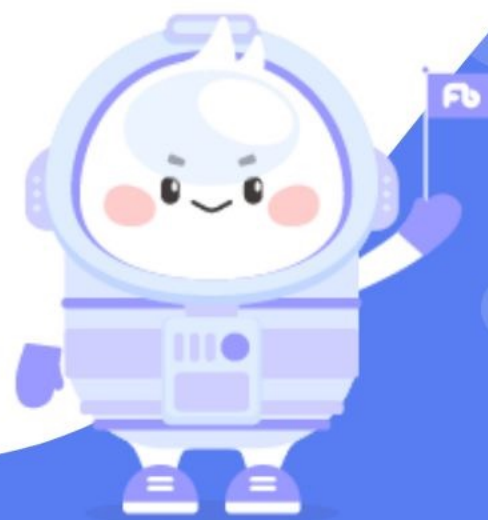


《信息技术》
数据库技术 3/3

► 讲师：孙珍珍

更多干货关注  粉笔教师教育  粉笔教师



✿ 复习一下



本页目的：了解出题方式即可

(2019下·初中) 在Access中，与图所示查询执行结果相同的SQL语句是 ()。



A.SELECT 作品名称 FROM hj

B.SELECT 作品名称,评审得分 FROM hj

C.SELECT 作品名称 FROM hj WHERE 评审得分 >=60

D.SELECT 作品名称,评审得分 FROM hj WHERE 评审得分 >=60



本页目的：了解出题方式即可

(2019上·高中) 根据关系模型Stu (学号 , 姓名 , 性别 , 出生年月) 。统计学生平均年龄应使用的SQL语句是 () 。

A.SELECT AVG(YEAR(DATE())) AS 平均年龄 FROM Stu

B.SELECT AVG(YEAR(出生年月())) AS 平均年龄 FROM Stu

C.SELECT AVG(YEAR(DATE()) + YEAR(出生年月)) AS 平均年龄 FROM Stu

D.SELECT AVG(YEAR(DATE()) - YEAR(出生年月)) AS 平均年龄 FROM Stu



本页目的：了解出题方式即可

(2021 下· 高中) SQL 的查询语句 “SELECT 员工姓名, 所属部门, 工资 FROM 员工数据库 ORDER BY 所属部门 COMPUTE SUM (工资) BY 所属部门” 的含义为 ()。

- A. 把员工数据库按照员工姓名排序后，计算每个部门的工资总和
- B. 把员工数据库按照所属部门排序后，计算每个部门的工资总和
- C. 把员工数据库按员工姓名排序后，计算每个部门的平均工资
- D. 把员工数据库按照所属部门排序后，计算每个部门的平均工资



SNo (学号)	SN (姓名)	Sex (性别)	Dept (系别)	Age (年龄)
11001	冯明	男	计算机	18
11002	陈月	女	通信	19
12001	褚共	男	计算机	18
12002	卫潮	男	自动化	20
12003	蒋生	女	通信	20

学生表S

选课表SC

SNo (学号)	CNo (课程编号)	Grade (成绩)
11001	C0201	85
11001	C0204	80
11002	C0302	65
12001	C0201	70
12001	D0101	76
12002	C0203	80
12003	C0201	85
12003	D0101	95

CNo (课程编号)	CN (课程名称)	Credit (学分)
C0201	数据库原理及应用	5
P0101	高等数学	6
C0203	数据结构	6
C0204	计算机网络	5
C0302	算法设计与分析	4
D0101	信息安全基础	4
C0103	计算机组成原理	4

课程表C

【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]  
FROM <表名>  
[WHERE <条件1>]  
[GROUP BY <列名1> [HAVING <条件2>]]  
[ORDER BY <列名2> [ASC|DESC]]
```

【说明】

- ①DISTINCT 代表在结果中，去除重复值
- ②若查询全部列，<列名>可以用*代替
- ③可以通过AS给列名和表起别名
- ④GROUP BY进行分组，HAVING根据条件进行分组
- ⑤ORDER BY进行排序；ASC升序（可省）、DESC降序

1.无条件查询【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]  
FROM <表名>
```

说明：

- ①DISTINCT 代表在结果中，去除重复值
- ②若查询全部列，<列名>可以用*代替
- ③可以通过AS给列名和表起别名

【实例】

【例7-1】从学生表s中查询所有学生的学号、姓名和性别。

```
SELECT SNo,SN,Sex  
FROM S
```

【例7-1】		
SNo	SN	Sex
11001	冯明	男
11002	陈月	女
12001	褚共	男
12002	卫潮	男
12003	蒋生	女

1.无条件查询【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]  
FROM <表名>
```

说明：

- ①DISTINCT 代表在结果中，去除重复值
- ②若查询全部列，<列名>可以用*代替
- ③可以通过AS给列名和表起别名

【实例】

【例7-2】从学生表s中查询所有学生信息。

```
SELECT *  
FROM S
```

【例7-2】

SNo	SN	Sex	Dept	Age
11001	冯明	男	计算机	18
11002	陈月	女	通信	19
12001	褚共	男	计算机	18
12002	卫潮	男	自动化	20
12003	蒋生	女	通信	20

2.条件查询【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]
FROM <表名>
WHERE <条件1>
```

说明：

①若查询全部列，<列名>可以用*代替

②查询条件及运算符对应表格

查询条件	运算符
比较	> , < , = , >= , <= , <>
多重条件	AND , OR , NOT
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

(1) 比较查询【实例】

【例8-1】从选课表 SC 中查询成绩大于80的学生选课信息。

```
SELECT *
FROM SC
WHERE Grade>80
```

【例8-1】		
SNo	CNo	Grade
11001	C0201	85
12003	C0201	85
12003	D0101	95

2. 条件查询【格式】

```
SELECT [ALL|DISTINCT] <列名> [AS <列别名>]
FROM <表名>
WHERE <条件1>
```

说明：

①若查询全部列，<列名>可以用*代替

②查询条件及运算符对应表格

查询条件	运算符
比较	> , < , = , >= , <= , <>
多重条件	AND , OR , NOT
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

(2) 多重条件查询【实例】

【例8-2】从选课表 SC 中查询选修了课程 “C0201”并且成绩高于80分的学生的学号与成绩。

```
SELECT SNo, Grade
```

```
FROM SC
```

```
WHERE CNo='C0201' AND Grade>80
```

【例8-2】	
SNo	Grade
11001	85
12003	85

2. 条件查询【格式】

```
SELECT [ALL|DISTINCT] <列名> [AS <列别名>]
FROM <表名>
WHERE <条件1>
```

说明：

①若查询全部列，<列名>可以用*代替

②查询条件及运算符对应表格

查询条件	运算符
比较	> , < , = , >= , <= , <>
多重条件	AND , OR , NOT
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

(3) 确定范围查询【实例】

【例8-3】从学生表 S 中查询年龄在[16,19]之间的学生学号、姓名和年龄。

```
SELECT SNo, SN, Age
```

```
FROM S
```

```
WHERE Age BETWEEN 16 AND 19
```

【例8-3】		
SNo	SN	Age
11001	冯明	18
11002	陈月	19
12001	褚共	18

2. 条件查询【格式】

```
SELECT [ALL|DISTINCT] <列名> [AS <列别名>]
FROM <表名>
WHERE <条件1>
```

说明：

①若查询全部列，<列名>可以用*代替

②查询条件及运算符对应表格

查询条件	运算符
比较	> , < , = , >= , <= , <>
多重条件	AND , OR , NOT
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

(4) 确定集合查询【实例】

【例8-4】从选课表 SC 中查询选修了课程 “C0204”或 “D0101”的学生的选课信息。

```
SELECT *
FROM SC
WHERE CNo IN ('C0204','D0101')
```

【例8-4】		
SNo	CNo	Grade
11001	C0204	80
12001	D0101	76
12003	D0101	95

2.条件查询【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]
FROM <表名>
WHERE <条件1>
```

说明：

- ①若查询全部列，<列名>可以用*代替
- ②查询条件及运算符对应表格

查询条件	运算符
比较	> , < , = , >= , <= , <>
多重条件	AND , OR , NOT
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

(5) 字符匹配查询【实例】

【例8-5】从课程表C中查询课号以“c”开头的的所有课号及课程名。

```
SELECT Cno, CN
FROM C
WHERE CNo like 'C*'
```

【例8-5】	
Cno	CN
C0103	计算机组成原理
C0201	数据库原理及应用
C0203	数据结构
C0204	计算机网络
C0302	算法设计及分析

2.条件查询-【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]
FROM <表名>
WHERE <条件1>
```

说明：

①若查询全部列，<列名>可以用*代替

②查询条件及运算符对应表格

查询条件	运算符
比较	> , < , = , >= , <= , <>
多重条件	AND , OR , NOT
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

(6) 空值查询【实例】

【例8-6】从表 SC中查询考试成绩有效的学生选课信息。

```
SELECT *
FROM SC
WHERE Grade IS NOT NULL
```

【例8-6】		
SNo	CNo	Grade
11001	C0201	85
11001	C0204	80
11002	C0302	65
12001	C0201	70
12001	D0101	76
12002	C0203	80
12003	C0201	85
12003	D0101	95

(2019下·初中) 在Access中 , 与图所示查询执行结果相同的SQL语句是 ()。



A.SELECT 作品名称 FROM hj

B.SELECT 作品名称,评审得分 FROM hj

C.SELECT 作品名称 FROM hj WHERE 评审得分 >=60

D.SELECT 作品名称,评审得分 FROM hj WHERE 评审得分 >=60



3. 聚集函数【格式】

```
SELECT [ALL|DISTINCT] <列名> [AS <列别名>]  
FROM <表名>  
WHERE <条件1>
```

说明：

①可以通过AS给列名和表起别名

②函数及功能对应表格

函数名称	功 能
AVG	计算列值平均值
SUM	计算列值的总和
MAX	求列值中的最大值
MIN	求列值中的最小值
COUNT(*)	统计元组个数

【实例】

【例9】从学生表s中查询计算机系学生的人数，以及这些学生年龄的总和及平均值。

```
SELECT COUNT(*), SUM(Age), AVG(Age)
```

```
FROM S
```

```
WHERE Dept='计算机';
```

【例9】		
Expr10	Expr10	Expr10
2	36	18

【例9】		
个数	年龄和	平均年龄
2	36	18

(2019上·高中) 根据关系模型Stu (学号 , 姓名 , 性别 , 出生年月) 。统计学生平均年龄应使用的SQL语句是 () 。

A.SELECT AVG(YEAR(DATE())) AS 平均年龄 FROM Stu

B.SELECT AVG(YEAR(出生年月())) AS 平均年龄 FROM Stu

C.SELECT AVG(YEAR(DATE()) + YEAR(出生年月)) AS 平均年龄 FROM Stu

D.SELECT AVG(YEAR(DATE()) - YEAR(出生年月)) AS 平均年龄 FROM Stu



4.分组查询【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]  
FROM <表名>  
[ WHERE <条件1> ]  
GROUP BY <列名1>  
[ HAVING <条件2> ]
```

说明：

- ①GROUP BY进行分组
- ②分组之前有条件进行过滤，用WHERE
- ③分组之后有条件进行过滤，用HAVING

【实例】

【例-书上无】统计不同性别的学生人数。

```
SELECT Sex, COUNT(*) AS 人数  
FROM S  
GROUP BY Sex
```

Sex	人数
男	3
女	2

4.分组查询【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]
FROM <表名>
[ WHERE <条件1> ]
GROUP BY <列名1>
[ HAVING <条件2> ]
```

说明：

- ①GROUP BY进行分组
- ②分组之前有条件进行过滤，用WHERE
- ③分组之后有条件进行过滤，用HAVING

【实例】

【例-书上无】 查询至少有2名男同学所在的系名。

```
SELECT Dept
FROM S
WHERE Sex='男'
GROUP BY Dept
HAVING Count(*)>=2
```

Dept
计算机

5.排序查询【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]  
FROM <表名>  
[WHERE <条件1>]  
ORDER BY <列名2> [ASC|DESC]
```

说明：

- ORDER BY进行排序；
- ASC升序（可省）、DESC降序

【实例】

【例11】从学生表s中查询学生的学号、姓名和年龄，并按年龄降序排列。

```
SELECT SNo, SN, Age  
FROM S  
ORDER BY Age DESC;
```

【例11】		
SNo	SN	Age
12003	蒋生	20
12002	卫潮	20
11002	陈月	19
12001	褚共	18
11001	冯明	18

(2021 下·高中) SQL 的查询语句 “SELECT 员工姓名, 所属部门, 工资 FROM 员工数据库 ORDER BY 所属部门 COMPUTE SUM (工资) BY 所属部门” 的含义为 ()。

- A. 把员工数据库按照员工姓名排序后，计算每个部门的工资总和
- B. 把员工数据库按照所属部门排序后，计算每个部门的工资总和
- C. 把员工数据库按员工姓名排序后，计算每个部门的平均工资
- D. 把员工数据库按照所属部门排序后，计算每个部门的平均工资



1.使用单表查询的SQL命令-【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]
FROM <表名1> ,<表名2>
[WHERE <条件1>]
[GROUP BY <列名1> [HAVING <条件2>]]
[ORDER BY <列名2> [ASC|DESC]]
```

说明：

- ①FROM子句，需要连接多个表
- ②SELECT后列名，如果两个表中均有，需要指明表名
- ③WHERE子句，需要指明两表建立联系的条件

查询【实例】

【例12-1】 查询学生冯明的学号、姓名、课号和成绩。

```
SELECT  S.SNo, SN, CNo, Grade
FROM    S, SC
WHERE   S.SNO=SC.SNO AND SN='冯明'
```

【例12-1】			
SNo	SN	CNo	Grade
11001	冯明	C0201	85
11001	冯明	C0204	80

2.使用JOIN子句-【格式】

```
SELECT [ALL|DISTINCT] <列名> [ AS <列别名>]
FROM <表名1> INNER JOIN <表名2>
ON <连接条件>
[WHERE <条件1>]
[GROUP BY <列名1> [HAVING <条件2>]]
[ORDER BY <列名2> [ASC|DESC]]
```

说明：

- ①两表之间用INNER JOIN进行连接
- ②两表建立连接的条件，使用ON子句

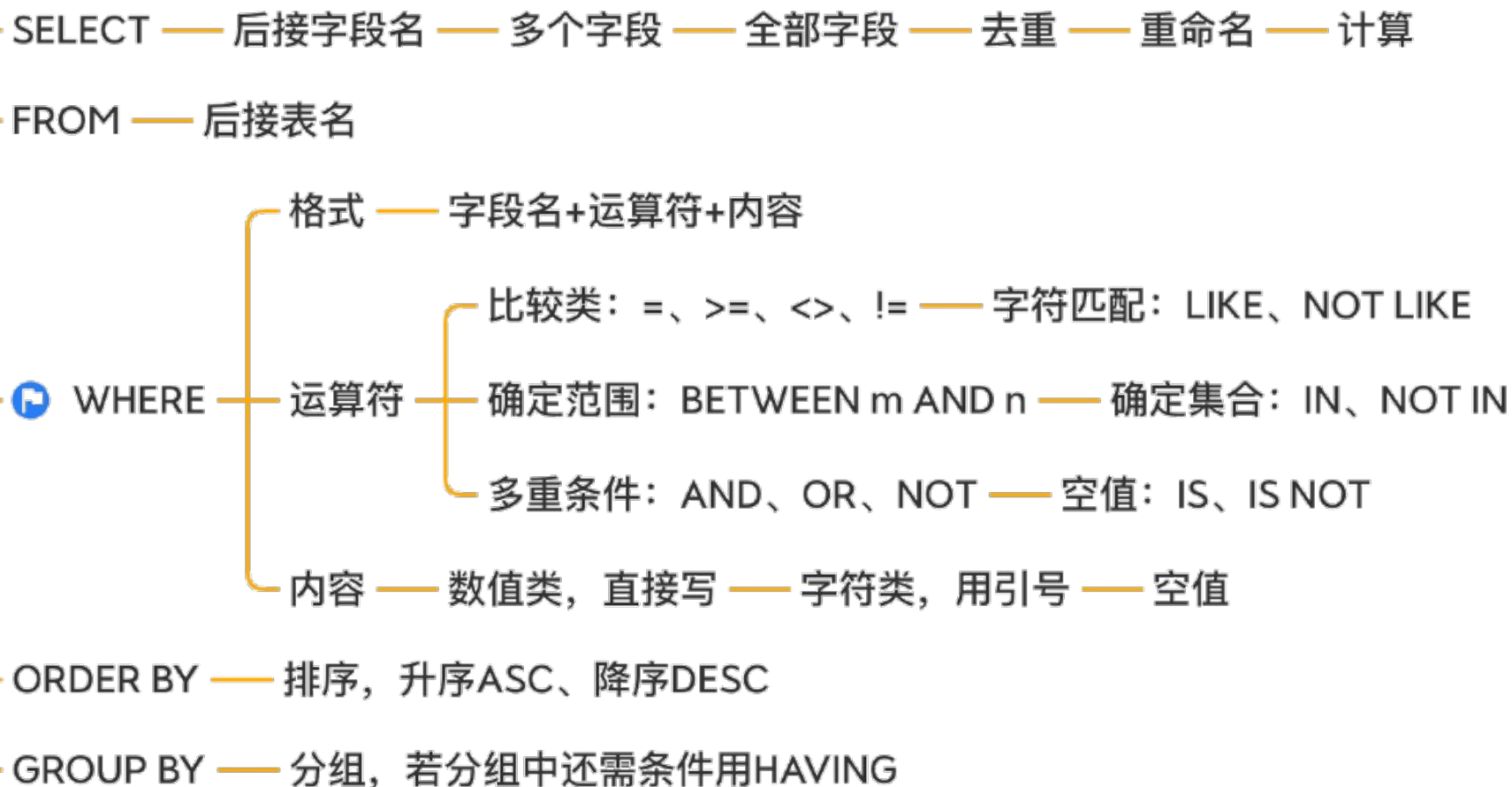
查询【实例】

【例12-2】 查询学生冯明的学号、姓名、课号和成绩。

```
SELECT S.SNo, SN, CNo, Grade
FROM S INNER JOIN SC
ON S.SNO=SC.SNO
WHERE SN='冯明'
```

【例12-2】			
SNo	SN	CNo	Grade
11001	冯明	C0201	85
11001	冯明	C0204	80

★ 查询数据

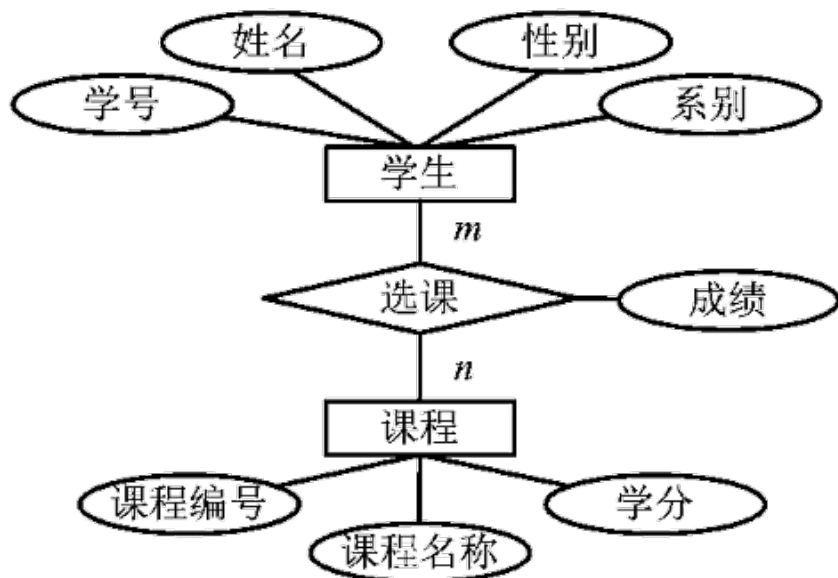




第六节 关系数据的设计

设计步骤	主要功能
需求分析	准确了解和分析用户的需求，是基础环节
概念结构设计	对用户需求进行归纳与抽象，并形成概念模型
逻辑结构设计	将概念模型转化成数据库管理系统所支持的逻辑模型
物理结构设计	为逻辑模型选取合适的物理结构
数据库实施	用数据库语言和宿主语言，建立数据库，编写与调试应用程序
数据库运行和维护	投入运行并不断对其进行评估、调整

1. 基于E-R模型的数据库设计方法



学生表

学号	姓名	性别	系别
--	--	--	--

课程表

课程编号	课程名称	学分
--	--	--

选课表

学号	课程编号	成绩
--	--	--

2. 基于3NF的数据库设计方法

➤ 单一关系模式 → 投影 → 若干关系模式

学号	姓名	性别	系别	课程编号	课程名称	学分	成绩
--	--	--	--	--	--	--	--

学生表

学号	姓名	性别	系别
--	--	--	--

课程表

课程编号	课程名称	学分
--	--	--

选课表

学号	课程编号	成绩
--	--	--

◆ 某教学信息管理数据库的**关系模式 SI** (SNo , SN , Sex , Age , CNo , CN , Grade)

➤ 根据实际情况，这些属性之间具有如下**语义设定**。

(1) 一个学号对应一名学生，一个课程号对应一门课程。

(2) 一名学生可以选修多门课程，每门课程可被若干名学生选修。

(3) 每名学生学习的每门课程都有一个成绩，但不一定立即给出。



主键：

SNo	SN	Sex	Age	CNo	CN	Grade
11001	冯明	男	18	C0201	数据库原理及应用	85
11001	冯明	男	18	C0204	计算机网络	80
11002	陈月	女	19	C0302	算法设计与分析	65
12001	褚共	男	18	C0201	数据库原理及应用	70
12001	褚共	男	18	D0101	信息安全基础	76

SNo	SN	Sex	Age	CNo	CN	Grade
11001	冯明	男	18	C0201	数据库原理及应用	85
11001	冯明	男	18	C0204	计算机网络	80
11002	陈月	女	19	C0302	算法设计与分析	65
12001	褚共	男	18	C0201	数据库原理及应用	70
12001	褚共	男	18	D0101	信息安全基础	76
12002	卫潮	男	20	C0203	数据结构	80
12003	蒋生	女	20	C0201	数据库原理及应用	85
12003	蒋生	女	20	D0101	信息安全基础	95

数据冗余高

插入异常

更新异常

删除异常

◆ 将关系模式 SI 分解为3个关系模式

- 学生关系S (SNo , SN , Sex , Age)
- 课程关系 C (CNo , CN)
- 选课关系 SC (SNo , CNo , Grade)

SNo	SN	Sex	Age
11001	冯明	男	18
11002	陈月	女	19
12001	褚共	男	18
12002	卫潮	男	20
12003	蒋生	女	20

CNo	CN
C0201	数据库原理及应用
C0204	计算机网络
C0302	算法设计与分析
D0101	信息安全基础
C0203	数据结构

SNo	CNo	Grade
11001	C0201	85
11001	C0204	80
11002	C0302	65
12001	C0201	70
12001	D0101	76
12002	C0203	80
12003	C0201	85
12003	D0101	95

本页目的：了解出题方式即可

(2021下·高中) 关系模式 $R(A, B, C, D, E)$ ，根据语义有如下函数依赖集：
 $F = \{A \rightarrow C, BC \rightarrow D, CD \rightarrow A, AB \rightarrow E\}$ 。关系模式 R 的规范化程度最高达到()。

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



(一) 定义

学生 (学号、身份证号、姓名、身高、年龄)

- 设关系模式 $R(U)$, U 是属性集, X 和 Y 是 U 的子集。
- X 确定 Y , Y 依赖于 X , 记作 $X \rightarrow Y$ 。 X 为决定因子, Y 为依赖因子。
- 当 Y 不依赖于 X 时, 记作 $X \nrightarrow Y$;
- 当 $X \rightarrow Y$ 且 $Y \rightarrow X$ 时, 则记作 $X \leftrightarrow Y$ 。

(二) 推理规则

IR1 (自反规则): 如果 $X \supseteq Y$, 那么 $X \rightarrow Y$ 。

IR2 (传递规则): $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$ 。

1. 完全函数依赖

➤ $X \rightarrow Y$ ，且对于 X 的任何一个真子集 X' ，都有 $X' \nrightarrow Y$ ，记为 $X \xrightarrow{f} Y$

【例】关系模式 $S1(SNo, SN, Sex, Age, CNo, CN, Grade)$ 中

$(SNo, CNo) \xrightarrow{f} Grade$

SNo	SN	Sex	Age	CNo	CN	Grade
11001	冯明	男	18	C0201	数据库原理及应用	85
11001	冯明	男	18	C0204	计算机网络	80
11002	陈月	女	19	C0302	算法设计与分析	65
12001	褚共	男	18	C0201	数据库原理及应用	70
12001	褚共	男	18	D0101	信息安全基础	76

2.部分函数依赖

➤ $X \rightarrow Y$ ，且对于 X 的某一个真子集 X' ，有 $X' \rightarrow Y$ ，记为 $X \xrightarrow{p} Y$

【例】关系模式 $S1(SNo, SN, Sex, Age, CNo, CN, Grade)$ 中

$(SNo, CNo) \xrightarrow{p} SN$

SNo	SN	Sex	Age	CNo	CN	Grade
11001	冯明	男	18	C0201	数据库原理及应用	85
11001	冯明	男	18	C0204	计算机网络	80
11002	陈月	女	19	C0302	算法设计与分析	65
12001	褚共	男	18	C0201	数据库原理及应用	70
12001	褚共	男	18	D0101	信息安全基础	76

3. 传递函数依赖

➤ $X \rightarrow Y$, $Y \rightarrow Z$, 且 X 不包含 Y , Y 不包含 Z , $Y \not\rightarrow X$ 则 $X \rightarrow Z$, 记为 $X \xrightarrow{t} Z$

【例】关系模式 学生 (学号, 姓名, 系号, 系主任) 中

学号 \xrightarrow{t} 系主任

SNo	SN	系号	系主任
11001	冯明	JSJ01	张三
11002	陈月	JSJ01	张三
12001	褚共	RJ02	李四

◆ F 中所有 $X \rightarrow A$ 的 A 的集合称为 X 的闭包, 记为 X^+

【例】设有关系模式 $R(M, N, X, Y, Z)$, 其 $F = \{M \rightarrow X, X \rightarrow Z, Y \rightarrow Z, N \rightarrow Y, Z \rightarrow M\}$ 。计算 M 和 N 属性的闭包。

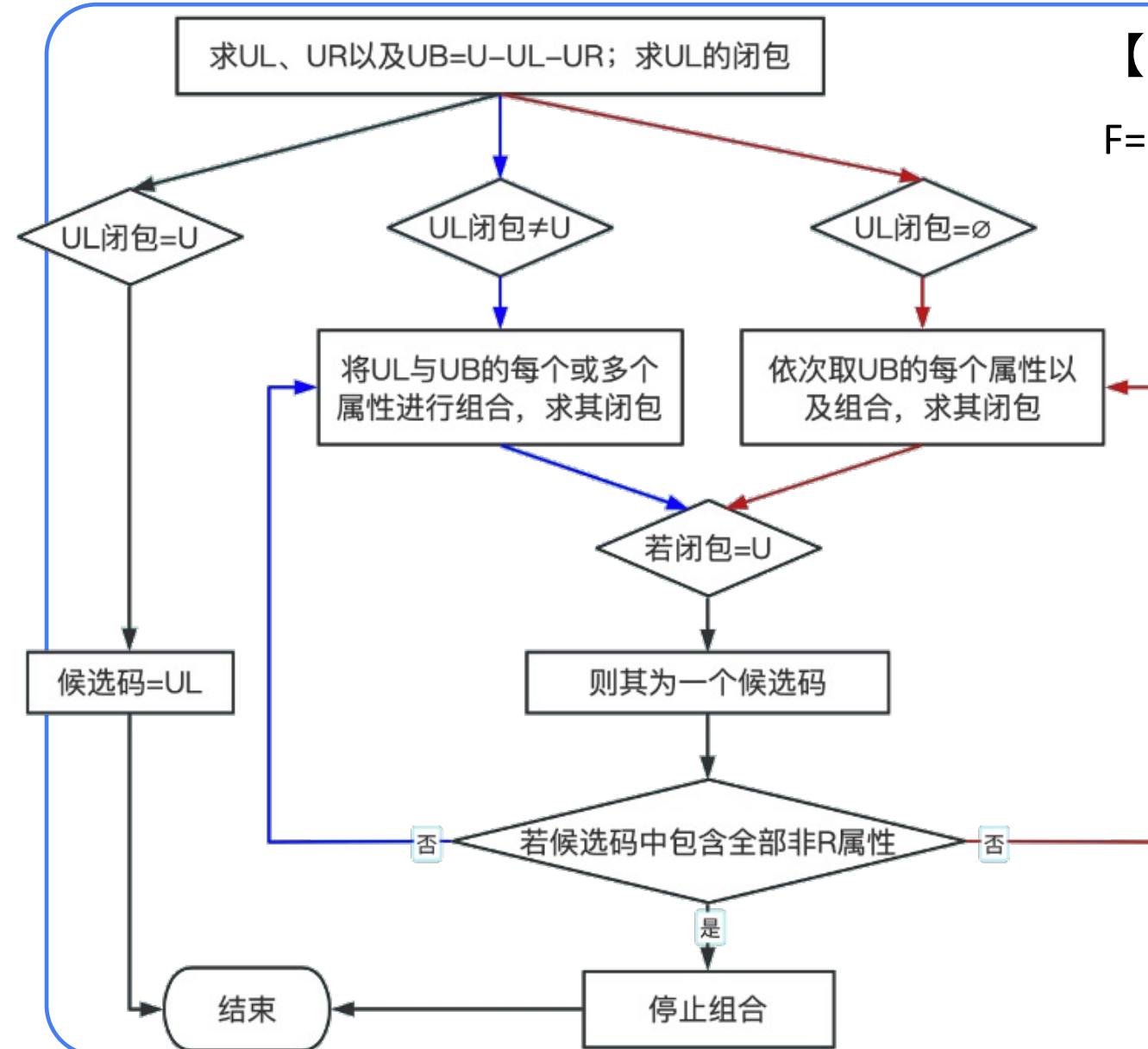
设有关系模式 $R(U)$ ， F 是函数依赖集，可将其中的属性分为以下四类。

- (1) L 类：仅出现在函数依赖左边的属性。
- (2) R 类：仅出现在函数依赖右边的属性。
- (3) N 类：在函数依赖左右两边均未出现的属性。
- (4) LR 类：在函数依赖左右两边均出现的属性。

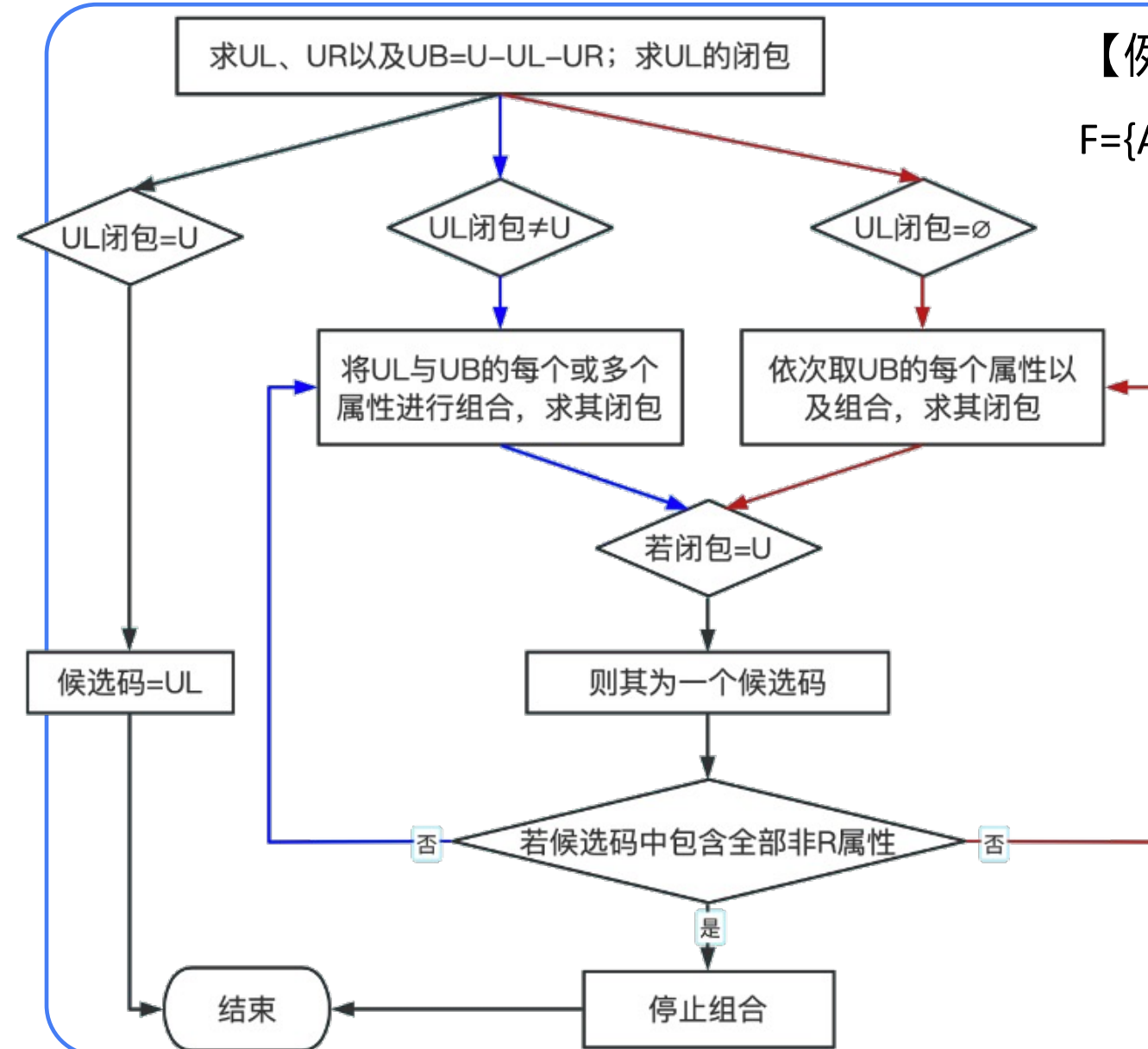
【例】 关系模式 $R(M, N, X, Y, Z)$ ， $F = \{M \rightarrow X, X \rightarrow Z, Y \rightarrow Z, N \rightarrow Y, Z \rightarrow M\}$ 。求分类。

- ① L 类 - UL
- ② R 类 - UR
- ③ N 类
- ④ LR 类

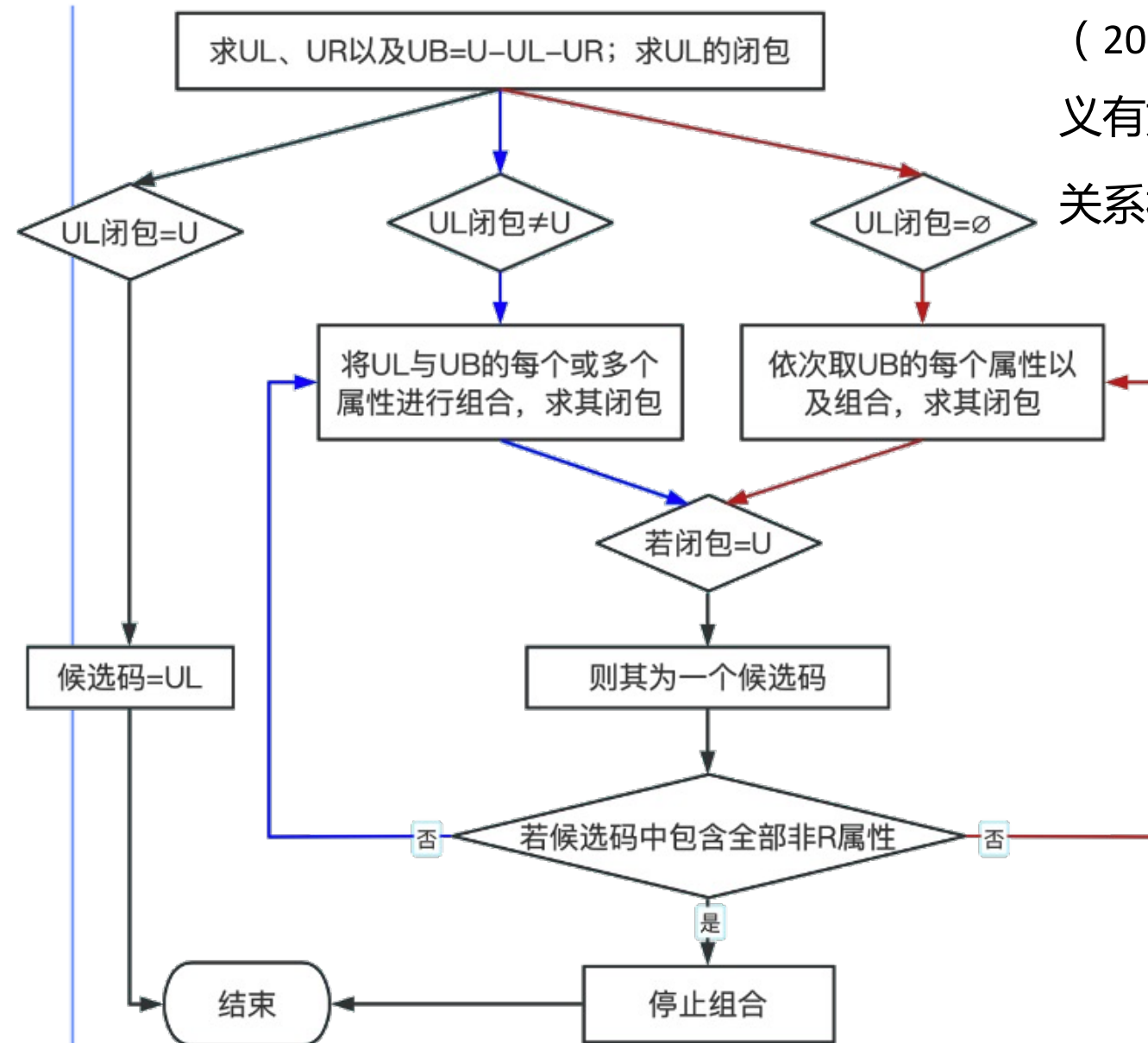
【例1】设有 $R(A, B, C, D)$ ，它的函数依赖集 $F = \{D \rightarrow B, B \rightarrow D, AD \rightarrow B, AC \rightarrow D\}$ ，求 R 的候选码。



【例2】设有 $R(A, B, C, D)$ ，它的函数依赖集 $F = \{AB \rightarrow C, D \rightarrow B, C \rightarrow A, C \rightarrow D\}$ ，求 R 的候选码。



(2021下·高中) 关系模式 $R(A, B, C, D, E)$ ，根据语义有如下函数依赖集： $F = \{A \rightarrow C, BC \rightarrow D, CD \rightarrow A, AB \rightarrow E\}$ 。
关系模式 R 的规范化程度最高达到 ()。



◆5种范式的包含关系： $5NF \subset 4NF \subset BCNF \subset 3NF \subset 2NF \subset 1NF$

(一) 第一范式 1NF

➤如果关系模式 R 的所有属性均为**原子属性**（不可再分），则称 R 属于第一范式。**最起码条件**

TNo	TN	Sex	Prof	Sal	
				Basic	Bonus
01001	赵乾	女	讲师	6000	500
01002	钱坤	男	讲师	6000	1000
01003	孙震	女	副教授	7000	1200
02011	李离	女	教授	8000	800
02013	周巽	男	教授	8000	2000



TNo	TN	Sex	Prof	Basic	Bonus
01001	赵乾	女	讲师	6000	500
01002	钱坤	男	讲师	6000	1000
01003	孙震	女	副教授	7000	1200
02011	李离	女	教授	8000	800
02013	周巽	男	教授	8000	2000

(二) 第二范式 2NF

➤ 如果关系模式 $R \in 1NF$ ，且每个非主属性都完全依赖于 R 的主码/候选码，则称 R 属于第二范式

【例】 将关系模式 选课 (学号, 课程号, 姓名, 成绩) 规范化为 2NF

学生 (学号, 姓名)

选课 (学号, 课程号, 成绩)

(三) 第三范式 3NF

➤ 如果关系模式 $R \in 2NF$ ，且每个非主属性都不传递依赖于 R 的主码/候选码，则称 R 属于第三范式

【例】 将关系模式学生（学号，姓名，系号，系主任）规范化为 3NF

学生（学号，姓名，系号）

系管理（系号，系主任）

(四) BC范式 BCNF

➤ 如果关系模式 $R \in 3NF$ ，且不存在主属性对非主属性的函数依赖，则称 R 属于 BC 范式，简称 BCNF

【例】 假设该学校中每名教师只讲授一门课程，但一门课程可由多个教师讲授；每名学生可以选修多门课程，每门课程可由多名学生选修，且每次选课都会有一条成绩记录。

要求：将关系模式 学生（学号，课号，教师，成绩）规范化为 BCNF

学生（学号，课号，成绩）

授课（教师，课号）

(2021下·高中) 关系模式 $R(A, B, C, D, E)$ ，根据语义有如下函数依赖集：
 $F = \{A \rightarrow C, BC \rightarrow D, CD \rightarrow A, AB \rightarrow E\}$ 。关系模式 R 的规范化程度最高达到()。

A. 1NF

第一范式-1NF 不存在合并列

B. 2NF

第二范式-2NF 非主属性不部分依赖于候选码

C. 3NF

第三范式-3NF 非主属性不传递依赖于候选码

D. BCNF

BC范式-BCNF 主属性不依赖于非主属性



关系数据库的设计

设计步骤

需求分析 — 概念/逻辑/物理结构设计 — 数据库实施/运行和维护

函数依赖

定义 — $X \rightarrow Y$ — 决定因素 — 依赖因素



分类

- 完全函数依赖 — $(A, B) \rightarrow C$, 不存在 $A \rightarrow C$ 或 $B \rightarrow C$
- 部分函数依赖 — $(A, B) \rightarrow C$, 存在 $A \rightarrow C$ 或 $B \rightarrow C$
- 传递函数依赖 — $A \rightarrow B, B \rightarrow C$, 则 $A \rightarrow C$



候选码求解 — 具体过程见流程图

会求解

基础

范式

- 第一范式-1NF — 不存在合并列
- 第二范式-2NF — 非主属性不部分依赖于候选码
- 第三范式-3NF — 非主属性不传递依赖于候选码
- BC范式-BCNF — 主属性不依赖于非主属性

会判断





有疑问没？等你吖

下节内容

第六章 数据结构与算法.....269

P269 ~ P279

第一节 算法基础 269



岸上等你

THE TEST

光芒万丈
不负理想

粉笔
教师



机读卡

姓名:

考号:

