计算机导论

第七章 数据库系统概论

河北师范大学软件学院
Software College of Hebei Normal University

前言

- 为什么要开设该课程？
  - 其重要性以及应用的广泛性在日常生活中的体现
    - 学分制系统
    - 医院的挂号等系统
    - 银行的各种业务系统
    - 火车票的查询和订票系统
- 学会这么课程之后你能做什么？
  - DBA
  - 开发动态网站
  - 其他与数据库有关的应用
  - ……

内容

1. 基本概念
2. 关系数据理论
3. 数据库设计
4. 关系数据库标准语言SQL

数据

- 数据(Data)是数据库中存储的基本对象
- 数据的定义
  - 描述事物的符号记录
- 数据的种类
  - 文字、图形、图象、声音
- 数据的特点
  - 数据与其语义是不可分的

数据举例

- 学生档案中的学生记录
  - （李明，男，1972，江苏，计算机系，1990）
- 数据的形式不能完全表达其内容
- 数据的解释
  - 语义：学生姓名，性别，出生年月，籍贯，所在系别，入学时间
  - 解释：李明是个大学生，1972年出生，江苏人，1990年考入计算机系
- 请给出另一个解释和语义

- 数据库的定义
  - 数据库(Database,简称DB)是长期储存在计算机内、有组织的、可共享的大量数据的集合。
- 数据库的基本特征
  - 数据按一定的数据模型组织、描述和储存
  - 可为各种用户共享
  - 冗余度较小
  - 数据独立性较高
  - 易扩展

数据库管理系统

- 什么是DBMS
  - 位于用户与操作系统之间的一层数据管理软件。
  - 是基础软件，是一个大型复杂的软件系统

典型的DBMS系统

- ORACLE
- SQL SERVER
- SYBASE
- INFORMIX
- DB/2
- COBASE
- MySQL
- PBASE
- EasyBase
- OpenBase

数据库系统

- 什么是数据库系统（Database System，DBS）
  - 在计算机系统中引入数据库后的系统
  - 是由数据库、数据库管理系统应用程序和数据库管理员组成的存储、管理、处理和维护数据的系统。
- 数据库系统的构成
  - 硬件平台及数据库
  - 软件
  - 人员

域

- 域是一组具有相同数据类型的值的集合
- 例:
  - 整数
  - 实数
  - 介于某个取值范围的整数
  - 指定长度的字符串集合
  - {‘男’，‘女’}
  - 介于某个取值范围的日期

关系

- 关系

  一个关系对应通常说的一张表

  关系的表示：

  R：关系名　$R(D_1, D_2, \ldots, D_n)$

  - n：关系的目或度（Degree）

- **候选码**

  若关系中的某一属性组的值能**唯一**地表示一个元组，而其子集不能，则称该属性组为**候选码**

- **主码**

  若一个关系有多个候选码，则选定其中一个为**主码**

- **主属性**

  候选码的诸属性称为**主属性**

- **非主属性**

  不包含在任何候选码中的属性称为**非主属性**

- **全码**：在最极端的情况下，关系模式的所有属性是这个关系模式的候选码

关系术语

- 表(table)、列(column)、行(row)
- 关系(relation)、元组(tuple)、属性(attribute)

关系或表

属性或列

元组或行

学号 Sno	姓　名 Sname	性　别 Ssex	年　龄 Sage	所 在 系 Sdept
201215121	李勇	男	20	CS
201215122	刘晨	女	19	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

关系术语

- 候选码（Candidate key） 全码（All-key）

- 主码 主属性 非主属性

SC（Sno, Cno, Grade）

主属性 主属性 主码 非主属性

SC

学 号 Sno	课程号 Cno	成绩 Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

关系分类

- 三类关系
  - **基本关系**（基本表或基表）

    实际存在的表，是实际存储数据的逻辑表示
  - **查询表**

    查询结果对应的表
  - **视图表**

    由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据

关系模式

关系

学号 Sno	姓　名 Sname	性　别 Ssex	年　龄 Sage	所 在 系 Sdept
201215121	李勇	男	20	CS
201215122	刘晨	女	19	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

关系模式

- 关系模式是对关系的描述，是静态的、稳定的
- 关系是关系模式在某一时刻的状态或内容，是动态的、随时间不断变化的
- 关系模式和关系往往统称为关系，通过上下文加以区别

基本关系操作

- 常用的关系操作
  - 查询：选择、投影、连接、除、并、交、差
  - 数据更新：插入、删除、修改
  - 查询的表达能力是其中最主要的部分
- **关系操作的特点**
  - 集合操作方式：操作的对象和结果都是集合，一次一集合的方式

关系的完整性

- 实体完整性
- 参照完整性
- 用户定义的完整性

内容

1. 基本概念
2. 关系数据理论
3. 数据库设计
4. 关系数据库标准语言SQL

假设存在这样一个关系：

Student(Sno, Sdept, Mname, Cno, Grade)

Sno	Sdept	Mname	Cno	Grade
S1	计算机系	张明	C1	95
S2	计算机系	张明	C1	90
S3	计算机系	张明	C1	88
S4	计算机系	张明	C1	70
S5	计算机系	张明	C1	78
⋮	⋮	⋮	⋮	⋮

请问该关系模式好吗？

Sno	Sdept	Mname	Cno	Grade
S1	计算机系	李四	C1	95
S2	计算机系	李四	C1	90
S3	计算机系	李四	C1	88
S4	计算机系	李四	C1	70
S5	计算机系	李四	C1	78
⋮	⋮	⋮	⋮	⋮

- **系名、系主任名**重复出现
- “张明”退休，李四接替
- 一个新系刚成立，尚无学生
- 一个系的学生全部毕业

- **数据冗余太大**
- **更新异常**
- **插入异常**
- **删除异常**

结论：

- Student关系模式不是一个好的关系模式。
- “好”的关系模式：

不会发生插入异常、删除异常、更新异常，

数据冗余应尽可能少

原因：由存在于关系模式中的**某些数据依赖**引起的

解决方法：通过**分解**关系模式来消除其中不合适

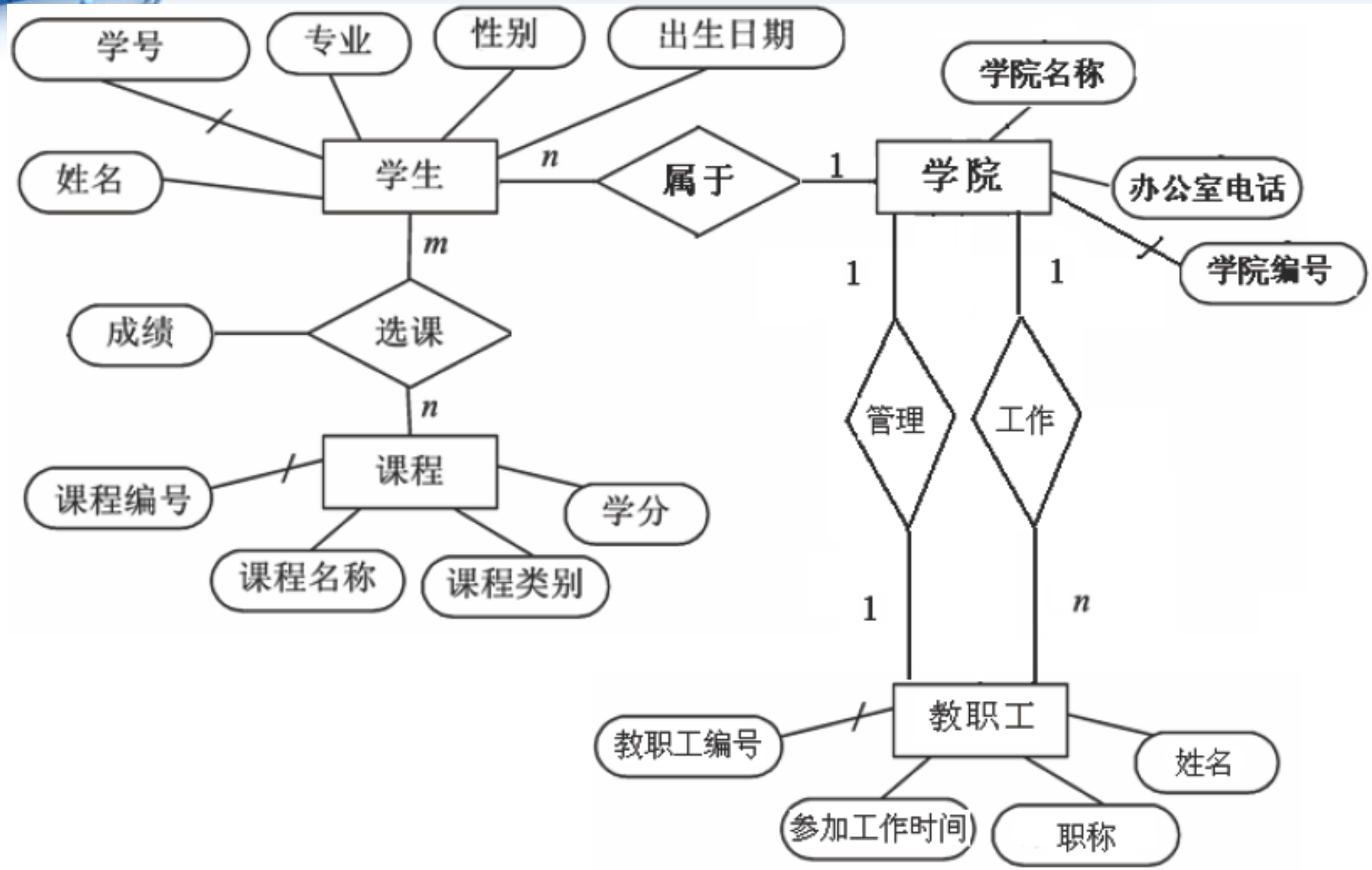
内容

1. 基本概念
2. 关系数据理论
3. 数据库设计
4. 关系数据库标准语言SQL

数据库设计

- 数据库设计概述
- 需求分析
- 概念结构设计
- 逻辑结构设计
- 数据库的物理设计
- 数据库实施和维护

概念结构设计


学号
专业
性别
出生日期
学院名称
姓名
学生
n
属于
1
学院
办公室电话
m
1
1
学院编号
成绩
选课
管理
工作
n
课程
课程编号
学分
课程名称
课程类别
1
n
教职工
教职工编号
姓名
参加工作时间
职称

转换原则

- 一个实体型转换为一个关系模式
  - 关系的属性：实体型的属性
  - 关系的码：实体型的码

关系模式：

学生（学号，姓名，性别，专业，出生日期）

课程（课程编号，课程名称，课程类别，学分）

选课（学号，课程编号，成绩）

内容

1. 基本概念
2. 关系数据理论
3. 数据库设计
4. 关系数据库标准语言SQL

- **数据定义**
- 数据查询
- 数据更新

创建数据库

- 语法

  CREATE DATABASE <database_name>

  例：CREATE DATABASE student

- 使用数据库

  use <database_name>

  例：use student

- 删除数据库

  drop database <database_name>

  例：drop database student

不能删除当前数据库

基本表

- 定义基本表

CREATE TABLE <表名>

(<列名> <数据类型> [<列级完整性约束条件>]

[，<列名> <数据类型> [<列级完整性约束条件>]] …

[，<表级完整性约束条件>] ）；

DROP TABLE <表名>;

索引

- 语句格式

  CREATE [UNIQUE] [CLUSTER] INDEX <索引名> ON

  <表名>(<列名>[<次序>][,<列名>[<次序>]]…);

- DROP INDEX <索引名>
  - 删除索引时，系统会从数据字典中删去有关该索引的描述
- [例] 删除Student表的Stusname索引。
  - DROP INDEX Student.Stusname

- 数据定义
- **数据更新**
- 数据查询

数据更新

- 数据的插入
  - 插入元组
  - 插入子查询结果
- 数据的修改
- 数据的删除

插入元组

- 语句格式

  INSERT

  INTO <表名> [(<属性列1>[，<属性列2 >…)]

  VALUES (<常量1> [，<常量2>] …)

- 功能

  将新的元组插入到指定表

插入子查询结果

- 语句格式

  INSERT INTO <表名> [(<属性列1>[，<属性列2 >…)]

  子查询

- 功能
  - 将子查询结果插入指定表中
- 注意
  - 子查询的结果必须包含和insert的字段列表一样多的字段，并且数据类型兼容

数据更新

- 数据的插入
- 数据的修改
  - 修改某元组的值
  - 修改多个元组的值
  - 带子查询的修改语句
- 数据的删除

修改数据

- 语句格式

  UPDATE <表名>

  SET <列名>=<表达式>[，<列名>=<表达式>]…

  [WHERE <条件>];

  - SET子句
    - 指定修改方式、要修改的列、修改后取值
  - WHERE子句
    - 指定要修改的元组，缺省表示要修改表中的所有元组
- 功能
  - 修改指定表中满足WHERE子句条件的元组

数据更新

- 数据的插入
- 数据的修改
- 数据的删除
  - 删除某一个元组的值
  - 删除多个元组的值
  - 带子查询的删除语句

删除语句

- 定义

  DELETE

  FROM <表名>

  [WHERE <条件>];

  - WHERE子句
    - 指定要删除的元组，缺省表示要删除表中的所有元组

- 功能

  - 删除指定表中满足WHERE子句条件的元组

- 数据定义
- 数据更新
- **数据查询**

基本语法

- 基本语法

SELECT [ALL|DISTINCT]〈目标列表达式〉[,〈目标列表达式>] …

FROM 〈表名或视图名〉[，〈表名或视图名〉] …

[WHERE <条件表达式>]

[GROUP BY 〈列名〉[，〈列名〉]…

[HAVING <内部函数表达式>]]

[ORDER BY 〈列名〉 [ASC | DESC] [，〈列名〉[ASC | DESC]]…]

子句功能

- 子句功能
  - **SELECT**子句与**FROM**子句是**必选子句**
  - SELECT ---- 列出查询的结果
  - FROM ---- 指明所访问的对象
  - WHERE ---- 指定查询的条件
  - GROUP BY ---- 将查询结果按指定字段的取值分组
  - HAVING ----筛选出满足指定条件的组
  - ORDER BY ---- 按指定的字段的值，以升序或降序排列查询结果

单表查询

- 单表查询
  - 投影查询
  - 选择查询
  - order by子句
  - 聚集函数
  - group by子句

SELECT <目标列表达式> FROM <表名或视图名>

目标表达式可以是：属性名、算术表达式、字符串常量、函数等

- 属性名

［例1］ 查询全体学生的学号、姓名、所在系。

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

［例2］ 查询全体学生的详细记录。

```
SELECT *
FROM Student；
```

- 查询满足条件的元组（where子句）

WHERE子句常用的查询条件

表3.4　常用的查询条件

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

比较大小

- 在WHERE子句的<比较条件>中使用比较运算符
  - =，>，<，>=，<=，!= 或 <>，!>，!<
  - 逻辑运算符NOT+比较运算符

［例3］　查询所有年龄在20岁以下的学生姓名及其年龄。

```
SELECT Sname，Sage
FROM    Student
WHERE Sage < 20；
```

```
SELECT Sname，Sage
FROM    Student
WHERE NOT Sage >= 20；
```

确定范围

- 使用谓词 BETWEEN … AND …

  NOT BETWEEN … AND …

[例4] 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。

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

确定集合

- 使用谓词IN <值表>,NOT IN <值表>
  - <值表>：用逗号分隔的一组取值

[例5]　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。

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

字符串匹配

- [NOT] LIKE ‘<匹配串>’ [ESCAPE ‘ <换码字符>’]
  - <匹配串>：指定匹配模板，可以是固定字符串或含通配符的字符串
  - 当匹配模板为固定字符串时，可以用 = 运算符取代 LIKE 谓词，用 != 或 < >运算符取代 NOT LIKE 谓词
  - 通配符
    - % (百分号) 代表任意长度（长度可以为0）的字符串
    - _ (下横线) 代表任意单个字符

［例6］　查询所有姓刘学生的姓名、学号和性别。

```
SELECT  Sname，Sno，Ssex
FROM  Student
WHERE  Sname LIKE ‘刘%’;
```

［例7］　查询姓“欧阳”且全名为三个汉字的学生的姓名。

```
SELECT  Sname
FROM    Student
WHERE   Sname LIKE '欧阳_ _';
```

ORDER BY子句

- 使用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
- 当排序列含空值时
  - ASC：排序列为空值的元组最先显示
  - DESC：排序列为空值的元组最后显示
- 当按多个属性排序时
  - 首先根据第一个属性排序，如果在该属性上有多个相同的值时，则按第二个属性排序，以此类推

［例8］　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。

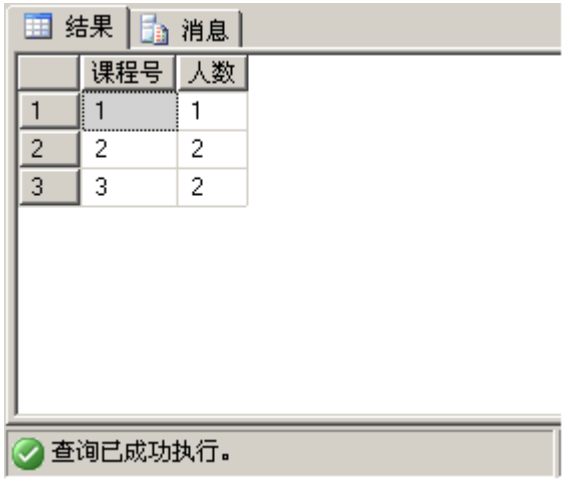
```
SELECT Sno，Grade
FROM  SC
WHERE  Cno = ' 3 '
ORDER BY Grade  DESC；
```

GROUP BY子句

- 使用GROUP BY子句分组
- 细化聚集函数的作用对象
  - 未对查询结果分组，聚集函数将作用于整个查询结果
  - 对查询结果分组后，聚集函数将分别作用于每个组

［例9］　求各个课程号及相应的选课人数。

```
SELECT     Cno  课程号，COUNT(Sno)
人数
FROM    SC
GROUP BY Cno;
```



Questions?