

数据库原理与技术



数据库系统概论

An Introduction to Database System

第三章 关系数据库标准语言SQL

浙江农林大学信息工程学院

第三章 关系数据库标准语言SQL

3.1 SQL概述

- 3.2 学生-课程数据库
- 3.3 数据定义
- 3.4 数据查询
- 3.5 数据更新
- 3.6 空值的处理
- 3.7 视图
- 3.8 小结



3.1 SQL概述

SQL (Structured Query Language)

结构化查询语言,是关系数据库的标准语言

▶ SQL是一个通用的、功能极强的关系数据库语言

SQL标准的进展过程

标准	大致页数	发布日期
SQL/86		1986.10
SQL/89 (FIPS 127-1)	120页	1989年
SQL/92	622页	1992年
SQL99 (SQL3)	1700页	1999年
SQL2003	3600页	2003年
SQL2008	3777页	2006年
SQL2011		2010年

目前,没有一个数据库系统能够支持SQL标准的所有概念和特性

SQL的特点

- 1.综合统一
 - 集数据定义语言(DDL),数据操纵语言(DML)数据控制语言(DCL)功能于一体。
 - ▶可以独立完成数据库生命周期中的全部活动。
 - ▶用户数据库投入运行后,可根据需要随时逐步修改模式,不影响数据库的运行。
 - ▶数据操作符统一

2. 高度非过程化

- 非关系数据模型的数据操纵语言"面向过程"必须指定存取路径。
- ▶ SQL只要提出"做什么",无须了解存取路径。
- ▶ 存取路径的选择以及SQL的操作过程由系统自动 完成。

- 3. 面向集合的操作方式
 - 非关系数据模型采用面向记录的操作方式,操作对象是一条记录
 - ▶ SQL采用集合操作方式
 - ▶ 操作对象、查找结果可以是元组的集合
 - 一次插入、删除、更新操作的对象可以是元组的集合

4. 以同一种语法结构提供多种使用方式

SQL是独立的语言

能够独立地用于联机交互的使用方式

▶ SQL又是嵌入式语言

SQL能够嵌入到高级语言(例如C#, C++, Java)程

序中, 供程序员设计程序时使用



5.语言简洁,易学易用

SQL功能极强,完成核心功能只用了9个动词。

表 3.2 SQL 的动词

			1000		
SQL	功能	动词		- 100 M	
	查询		37		
			DROP,	INTER !	
		SERT, UI	PDATE, D	ELEVE	
	控制		EVOKE	Albert 1	

SQL的基本概念(续)

SQL支持关系数据库三级模式结构





- 本身独立存在的表
- SQL中一个关系就对应一个基本表
- ▶一个(或多个)基本表对应一个存储文件
- ▶一个表可以带若干索引



视图

- 从一个或几个基本表导出的表
- 数据库中只存放视图的定义而不存放视图对应
 - 的数据
- ▶视图是一个虚表
- ▶用户可以在视图上再定义视图

第三章 关系数据库标准语言SQL

- 3.1 SQL概述
- 3.2 学生-课程数据库
- 3.3 数据定义
- 3.4 数据查询
- 3.5 数据更新
- 3.6 空值的处理
- 3.7 视图
- 3.8 小结



3.2 数据库

- → 学院表coll(collnum,collname,tnum)
- ▼ 系表d(dnum,dname,collnum)
- ◆ 学生表s(<u>snum</u>,sname,ssex,sbirth,spolity,snation,sorigin, enterdate,sphone,dnum)
- ◆ 教师表t(tnum,tname,tsex,tbirth,tpolity,tnation,torigin,degree,ttitle,tphone,tsalary,collnum)
- ◆ 课程表c(cnum,cname,credit)
- ◆ 选修表sc(snum,cnum,score)
- ◆ 任课表tc(tnum,cnum)

第三章 关系数据库标准语言SQL

- 3.1 SQL概述
- 3.2 学生-课程数据库
- 3.3 数据定义
- 3.4 数据查询
- 3.5 数据更新
- 3.6 空值的处理
- 3.7 视图
- 3.8 小结



3.3 数据定义

- ❖SQL的数据定义功能:
 - ■数据库定义
 - ■表定义
 - ■视图和索引的定义

表 3.3 SQL 的数据定义语句

	操作方式
	创建删除深刻修改
数据库	CREATE DATABASE DROP DATABASE
	CREATE TABLE DROP TABLE ALTER
	TABLE
	CREATE VIEW DROP VIEW
	CREATE INDEX DROP INDEX INDEX



3.3.2 基本表的定义、删除与修改

3.3.3 索引的建立与删除



1. 定义数据库

CREATE DATABASE <数据库名>;

2. 删除数据库

DROP DATABASE <数据库名>;

3. 切换默认数据库

USE <数据库名>;

4. 展示所有数据库 SHOW DATABASES;

5. 显示当前使用的数据库 SELECT DATABASE();





SQL书写准则

- 在SQL语句语法格式中的一些约定符号:
 - 头括号 "<>"中的内容为实际语义。
 - ▶中括号"[]"中的内容为任选项。
 - ▶大括号 "{}"与竖线 "|"表明此处为选择项,在所列出的各项中仅需选择一项。

例如: { A | B | C | D }意思是A、B、C、D中取其一。

▶SQL中的数据项(包括列项、表和视图)分隔符为

",";其字符串常数的定界符用单引号""表示。

在编写SQL语句时,遵从某种准则以提高语句的可读性, 使其易于编辑,是很有好处的。

以下是一些通常的准则:

- > SQL语句对大小写不敏感 为了提高SQL语句的可读性,子句开头的关键字通常采用大写形式。
- ▶ SQL语句可写成一行或多行,习惯上每个子句占用一行。
- ▶ 关键字不能在行与行之间分开,并且很少采用缩写形式。

数据类型

- SQL中域的概念用数据类型来实现
- 定义表的属性时需要指明其数据类型及长度
- ▶选用哪种数据类型
 - ▶取值范围
 - ▶要做哪些运算



2. 数据类型

- ▶ 精度: 指数值中所存储的十进制数据的总位数。
- 小数位数:指数值数据中小数点右边可以有的数字位数的最大值。如3568.863的精度为7,小数位数为3。
- ▶ 长度: 指存储数据所使用的字节数

1.MySQL中的数据类型

类型	类型举例
整数类型	TINYINT、SMALLINT、MEDIUMINT、INT(或INTEGER)、BIGINT
浮点类型	FLOAT, DOUBLE
定点数类型	DECIMAL
位类型	BIT
日期时间类型	YEAR, TIME, DATE, DATETIME, TIMESTAMP
文本字符串类型	CHAR, VARCHAR, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT
枚举类型	ENUM
集合类型	SET
二进制字符串类 型	BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB
JSON类型	JSON对象、JSON数组
空间数据类型	单值类型: GEOMETRY、POINT、LINESTRING、POLYGON; 集合类型: MULTIPOINT、MULTILINESTRING、MULTIPOLYGON、 GEOMETRYCOLLECTION

常见数据类型的属性

MySQL关键字	含义
NULL	数据列可包含NULL值
NOT NULL	数据列不允许包含NULL值
DEFAULT	默认值
PRIMARY KEY	主键
AUTO_INCREMENT	自动递增,适用于整数类型
UNSIGNED	无符号
CHARACTER SET name	指定一个字符集

2.整数类型

整数类型一共有 5 种,包括 TINYINT、SMALLINT、MEDIUMINT、INT (INTEGER) 和 BIGINT

整数类型	字节	有符号数取值范围	无符号数取值范围
TINYINT	1	-128-127	0-255
SMALLINT	2	-32768-32767	0~65535
MEDIUMINT	3	-8388608~8388607	0-16777215
INT, INTEGER	4	-2147483648~2147483647	0~4294967295
BIGINT	8	-9223372036854775808~9223372036854775807	0-18446744073709551615

3.浮点类型

- FLOAT 表示单精度浮点数;
- DOUBLE 表示双精度浮点数;



4.定点类型

MySQL中的定点数类型只有 DECIMAL 一种类型。

数据类型	字节数	含义
DECIMAL(M,D),DEC,NUMERIC	M+2字节	有效范围由M和D决定

使用 DECIMAL(M,D) 的方式表示高精度小数。其中,M被称为精度,D被称为标度。0<=M<=65, 0<=D<=30,D<M。例如,定义DECIMAL (5,2) 的类型,表示该列取值范围是-999.99~999.99。

当DECIMAL类型不指定精度和标度时,其默认为DECIMAL(10,0)

- 浮点类型取值范围大,但是不精准,适用于需要取值范围大,又可以容忍微小误差的科学计算场景(比如计算化学、分子建模、流体动力学等)
- 定点数类型取值范围相对小,但是精准,没有误差,适合于对精度要求极高的场景 (比如涉及金额计算的场景)

5.日期与时间类型

类型	名称	字节	日期格式	最小值	最大值
YEAR	年	1	YYY T	1901	2155
TIME	时间	3	HH:MM:SS	-838:59:59	838:59:59
DATE	日期	3	YYYY-MM-DD	1000-01-01	9999-12-03
DATETIME	日期时间	8	YYYY-MM-DD HH:MM:SS	1000-01-01 00:00:00	9999-12-31 23:59:59
TIMESTAMP	日期时间	4	YYYY-MM-DD HH:MM:SS	1970-01-01 00:00:00 UTC	2038-01-19 03:14:07UTC

5.日期与时间类型

- · Year:4位字符串或数字格式表示YEAR类型,其格式为YYYY,最小值为1901,最大值为2155。
- Date:使用 CURRENT_DATE() 或者 NOW() 函数,会插入当前系统的日期。
- Time:可以使用"D HH:MM:SS"格式来表示TIME类型,使用CURRENT_TIME()或者 NOW()
- TIMESTAMP类型:使用TIMESTAMP存储的同一个时间值,在不同的时区查询时会显示不同的时间。

6.文本字符串类型

文本字符串类型	值 的 长 度	长 度 范 围	占用的存储空间
CHAR(M)	M	0 <= M <= 255	M个字节
VARCHAR(M)	M	0 <= M <= 65535	M+1个字节
TINYTEXT	L	0 <= L <= 255	L+2个字节
TEXT	L	0 <= L <= 65535	L+2个字节
MEDIUMTEXT	L	0 <= L <= 16777215	L+3个字节
LONGTEXT	L	0 <= L <= 4294967295	L+4个字节
ENUM	L	1 <= L <= 65535	1或2个字节
SET	L	0 <= L <= 64	1,2,3,4或8个字节

6.文本字符串类型

• ENUM类型也叫作枚举类型,ENUM类型的取值范围需要在 定义字段时进行指定。设置字段值时,ENUM类型只允许 从成员中选取单个值,不能一次选取多个值。ENUM类型 的成员个数的上限为65535个。

• SET表示一个字符串对象,可以包含0个或多个成员,但成员个数的上限为 64

总结

- · 如果确定是整数 , 就用 INT ;
- · 如果是小数 , 一定用定点数类型DECIMAL(M, D) ;
- 如果是日期与时间,就用 DATETIME 。

阿里开发手册

- · 任何字段如果为非负数,必须是 UNSIGNED
- · 小数类型为 DECIMAL,禁止使用 FLOAT 和 DOUBLE。
- · 如果存储的字符串长度几乎相等,使用 CHAR 定长字符串类型。
- VARCHAR 是可变长字符串,不预先分配存储空间,长度不要超过 5000。如果存储长度大于此值,定义字段类型为 TEXT,独 立出来一张表,用主键来对应,避免影响其它字段索引效率。

3.3.2 基本表的定义、删除与修改

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>]);

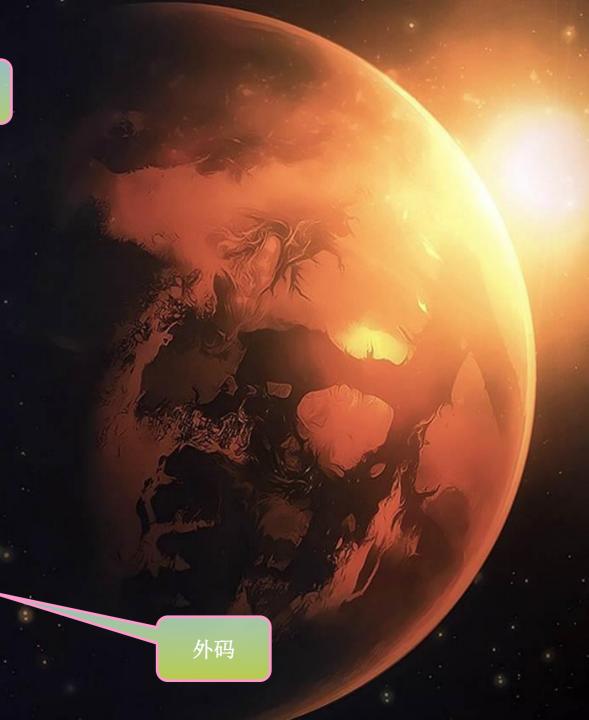
- ▶ <表名>: 所要定义的基本表的名字
- ▶ <列名>: 组成该表的各个属性(列)
- ▶ <列级完整性约束条件>:涉及相应属性列的完整性约束条件
- ▶<表级完整性约束条件>:涉及一个或多个属性列的完整性约束条件
- ▶如果完整性约束条件涉及到该表的多个属性列,则必须定义在表级上,否则既可以定义在列级也可以定义在表级。

[例] 建立"学生"表s。学号是主码,系编号是外码。

► CREATE TABLE s

主码

- snum CHAR(12) PRIMARY KEY,
- sname VARCHAR(50) NOT NULL,
- ▶ ssex ENUM ('男','女'),
- sbirth DATE,
- spolity VARCHAR(30),
- ▶ snation VARCHAR(20) DEFAULT '汉族',
- sorigin VARCHAR(50),
- enterdate DATE DEFAULT '2022-09-12',
- sphone VARCHAR(20),
- ▶ hobby SET('吃饭','睡觉','打游戏','写代码'),
- dnum CHAR(4),
- FOREIGN KEY(dnum)REFERENCES d(dnum)
- ▶)COMMENT '学生表';



列的属性

- 默认值 DEFAULT
- NOT NULL
- 主键 PRIMARY KEY
- 唯一约束 UNIQUE主键和UNIQUE的区别
 - (1)一张表只能定义一个主键,却可以定义多个UNIQUE 约束
 - (2)主键不允许NULL,UNIQUE 可以,并且可以重复
 - (3)如果没有定义主键,MySQL会将第一个声明为NOT NULL UNIQUE的 列或列组合自动定义为主键
- 自増 AUTO_INCREMENT
 - (1)一个表最多有一个
 - (2)具有auto_increment属性的列必须建立索引
 - (3)auto_increment—般作为主键的属性。 auto_increment primary key

修改表名

▶ 方式1: 1个

ALTER TABLE <旧表名> RENAME TO <新表名>;

▶ 方式2: n个

RENAME TABLE <旧表名1> TO <新表名1>,<旧表名2> TO <新表名2>,...< 旧表名n> TO <新表名n>

▶ 例:

ALTER TABLE t RENAME TO teacher;

RENAME TABLE teacher TO t;

修改基本表

ALTER TABLE <表名>

[ADD COLUMN<新列名> <数据类型> [完整性约束] [AFTER <指定列名>]]

[ADD <表级完整性约束>]

[DROP COLUMN]<列名>]

[DROP CONSTRAINT<完整性约束名>]

[MODIFY <列名> <新数据类型>]

[CHANGE < | 日列名 > < 新列名 > < 新数据类型 >];



修改基本表 (续)

- <表名>是要修改的基本表
- ADD COLUMN子句用于增加新列
- DROP COLUMN子句用于删除表中的列
- DROP CONSTRAINT子句用于删除指定的完整性约束条件
- MODIFY子句用于修改字段数据类型与位置
- CHANGE子句用于修改字段名

修改基本表 (续)

[例]在s表的sphone列后增加semail列

ALTER TABLE s

ADD COLUMN semail VARCHAR(50) AFTER sphone;

不管基本表中原来是否已有数据,新增加的列一律为空值

修改基本表 (续)

[例] 将生日的数据类型改为DATETIME。

ALTER TABLE S MODIFY sbirth DATETIME;

[例] 将性别的字段名称改为gender。

ALTER TABLE s CHANGE ssex gender ENUM('男',女');

[例] 删除学生表的外键。

ALTER TABLE s DROP CONSTRAINT s_ibfk_1;

[例] 增加外键。

ALTER TABLE s ADD CONSTRAINT s_ibfk_1 FOREIGN KEY (dnum) REFERENCES d(dnum);

删除基本表

4. 删除基本表

► 格式 DROP TABLE <表名> ► DANGER

需要注意的是,一旦这个命令发出以后就没有办法可以取消它

删除学生表S

DROP TABLE S



查看基本表

▶查看表结构

DESC<表名>;

SHOW COLUMNS FROM <表名>;

▶ 查看建表语句 SHOW CREATE TABLE <表名>;



3.3 数据定义

3.3.1 模式的定义与删除

3.3.2 基本表的定义、删除与修改

3.3.3 索引的建立与删除

3.3.4 数据字典



3.3.3 索引的建立与删除

- 建立索引的目的: 加快查询速度
- 关系数据库管理系统中常见索引:
 - ▶顺序文件上的索引
 - ▶ B+树索引(参见爱课程网3.2节动画《B+树的增删改》
 - ▶ 散列 (hash) 索引
 - ▶位图索引
- ▶ 特点:
 - ▶ B+树索引具有动态平衡的优点
 - ▶ HASH索引具有查找速度快的特点

索引

- 谁可以建立索引
 - ▶数据库管理员 或 表的属主 (即建立表的人)
- ▶谁维护索引
 - ▶ 关系数据库管理系统自动完成
- ▶ 使用索引
 - ▶ 关系数据库管理系统自动选择合适的索引作为存取 路径,用户不必也不能显式地选择索引

1. 建立索引

语句格式

CREATE [UNIQUE] [CLUSTER] INDEX <索引名>

ON <表名>(<列名>[<次序>][,<列名>[<次序>]]...);

- ▶ <表名>: 要建索引的基本表的名字
- ▶索引:可以建立在该表的一列或多列上,各列名之间 用逗号分隔
- ▶<次序>: 指定索引值的排列次序, 升序: ASC, 降序: DESC。缺省值: ASC
- ▶UNIQUE:此索引的每一个索引值只对应唯一的数据记录
- ▶ CLUSTER: 表示要建立的索引是聚簇索引

建立索引 (续)

[例3.13] 为学生-课程数据库中的Student, Course, SC 三个表建立索引。Student表按学号升序建唯一索引, Course表按课程号升序建唯一索引, SC表按学号升序和课程号降序建唯一索引

CREATE UNIQUE INDEX Stusno ON Student(Sno);
CREATE UNIQUE INDEX Coucno ON Course(Cno);
CREATE UNIQUE INDEX SCno ON SC(Sno ASC,Cno DESC);

2. 修改索引

▶ ALTER INDEX <旧索引名> RENAME TO <新索引名>

▶ [例3.14] 将SC表的SCno索引名改为SCSno ALTER INDEX SCno RENAME TO SCSno;

3. 删除索引

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3.15] 删除Student表的Stusname索引

DROP INDEX Stusname;

3.3 数据定义

3.3.1 模式的定义与删除

3.3.2 基本表的定义、删除与修改

3.3.3 索引的建立与删除

3.3.4 数据字典



数据字典

- 数据字典是关系数据库管理系统内部的一组系统表,它记录了 数据库中所有定义信息:
 - ▶关系模式定义
 - ▶视图定义
 - ▶索引定义
 - ▶完整性约束定义
 - ▶各类用户对数据库的操作权限
 - ▶统计信息等
- ▶ 关系数据库管理系统在执行SQL的数据定义语句时,实际上就是在更新数据字典表中的相应信息。

第三章 关系数据库标准语言SQL

- 3.1 SQL概述
- 3.2 学生-课程数据库
- 3.3 数据定义
- 3.4 数据查询
- 3.5 数据更新
- 3.6 空值的处理
- 3.7 视图
- 3.8 小结



数据查询

语句格式

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>]...|(SELECT 语句)

[AS]<别名>

[WHERE <条件表达式>]

[GROUP BY <列名1>[HAVING <条件表达式>]]

[ORDER BY <列名2> [ASC|DESC]];

数据查询

- ▶ SELECT子句:指定要显示的属性列
- ▶ FROM子句: 指定查询对象 (基本表或视图)
- ▶ WHERE子句: 指定查询条件
- ▶ GROUP BY子句:对查询结果按指定列的值分组,该属性列值相等的元组为一个组。通常会在每组中作用聚集函数。
- ▶ HAVING短语: 只有满足指定条件的组才予以输出
- ▶ ORDER BY子句:对查询结果表按指定列值的升序或降序排序

3.4 数据查询

- 3.4.1 单表查询
- 3.4.2 连接查询
- 3.4.3 嵌套查询
- 3.4.4 集合查询
- 3.4.5基于派生表的查询
- 3.4.6 Select语句的一般形式



- 3.4.1 单表查询
 - 查询仅涉及一个表
 - 1.选择表中的若干列
 - 2.选择表中的若干元组
 - 3.ORDER BY子句
 - 4.聚集函数
 - 5.GROUP BY子句



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。



选择表中的若干列(续)

- 查询全部列
 - ▶ 选出所有属性列:
 - ●在SELECT关键字后面列出所有列名
 - ●将<目标列表达式>指定为 *

[例] 查询全体学生的详细记录



查询经过计算的值(续)

查询经过计算的值

SELECT子句的<目标列表达式>为表达式

- 算术表达式
- ▶字符串常量
- ▶函数
- ▶列别名

[例] 查全体学生的姓名及其出生年份。

[例] 显示每一个教师的姓名月收入,若每个教师每月补贴1000元。



使用LIMIT限制查询结果的数量

- ▶ LIMIT [位置偏移量,] 行数
- 第一条记录的位置偏移量是0
- ▶ 行数指返回记录的条数



3.4.1 单表查询

- 查询仅涉及一个表:
 - 1.选择表中的若干列
 - 2.选择表中的若干元组
 - 3.ORDER BY子句
 - 4.聚集函数
 - 5.GROUP BY子句





▶消除取值重复的行

如果没有指定DISTINCT关键词,则缺省为ALL

[例] 查询选修了课程的学生学号。



查询满足条件的元组

表3.6 常用的查询条件

查询条件	谓 词
比较	=, >, <, >=, <=, !=, <>, !>, !<
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

[例] 查询所有教授的信息。

[例]查询考试成绩有不及格的学生的学号。



确定范围

谓词: BETWEEN ... AND ...

NOT BETWEEN ... AND ...

[例]列出月工资在6000元和8000元之间的教师信息。

[例]列出月工资不在在6000元和8000元之间的教师信息。

确定集合

▶ 谓词: IN (值表), NOT IN (值表)

[例]列出职称是副教授或教授或工程师的教师信息。

[例]列出职称不是副教授或教授或工程师的教师信息。

字符匹配

▶ 谓词: [NOT] LIKE '<匹配串>' [ESCAPE '<换码字符>']

<匹配串>可以是一个完整的字符串,也可以含有通配符%和

- ▶ % (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。



匹配串为固定字符串

[例] 查询姓名为'李晓'的学生的详细情况。





四配串为含通配符的字符串 [例] 查询所有姓刘学生的姓名、学号和性别。

[例] 查询姓"欧阳"且全名为三个汉字的学生的姓名。

字符匹配 (续)

使用换码字符将通配符转义为普通字符

[例]

SELECT * FROM s

WHERE sphone LIKE '%#_%' escape '#' ESCAPE '#' 表示 "#" 为换码字符

正则表达式: MySQL中使用REGEXP关键字

选项	说明	例子	匹配值示例
^	匹配文本的开始字符	'^b' 匹配以字母 b 开头 的字符串	book, big, banana, bike
\$	匹配文本的结束字符	'st\$' 匹配以 st 结尾的字 符串	test、resist、persist
	匹配任何单个字符	'b.t' 匹配任何 b 和 t 之间有一个字符	bit、bat、but、bite
*	匹配零个或多个在它前面的字 符	'f*n' 匹配字符 n 前面有 任意个字符 f	fn、fan、faan、abcn
+	匹配前面的字符 1 次或多次	'ba+' 匹配以 b 开头,后 面至少紧跟一个 a	ba、bay、bare、battle
<字符串>	匹配包含指定字符的文本	'fa'	fan、afa、faad
[字符集合]	匹配字符集合中的任何一个字 符	'[xz]'匹配 x 或者 z	dizzy、zebra、x-ray、 extra
[^]	匹配不在括号中的任何字符	'[^abc]' 匹配任何不包 含 a、b 或 c 的字符串	desk、fox、f8ke
字符串{n,}	匹配前面的字符串至少 n 次	b{2} 匹配 2 个或更多 的 b	ppp′ pppppp
字符串 {n,m}	匹配前面的字符串至少 n 次, 至多 m 次	b{2,4} 匹配最少 2 个,最多 4 个 b	ppp′ pppp

涉及空值的查询

→ 谓词: IS NULL 或 IS NOT NULL

"IS"不能用 "="代替

[例]查询职称为空的教师信息

SELECT tnum,tname,tsex,ttitle-- IFNULL(ttitle,'讲师')

FROM t

WHERE ttitle IS NULL



多重条件查询

- 逻辑运算符: AND和 OR来连接多个查询条件
 - ■AND的优先级高于OR
 - 可以用括号改变优先级

[例]列出1970年1月1日前出生,目前工资低于6000元或高于8000元的教师信息

3.4.1 单表查询

- ▶ 查询仅涉及一个表:
 - 1.选择表中的若干列
 - 2.选择表中的若干元组
 - 3.ORDER BY子句
 - 4.聚集函数
 - 5.GROUP BY子句



3.ORDER BY子句

- ► ORDER BY子句
 - 一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
- ▶ 对于空值,排序时显示的次序由具体系统实现来决定
- ▶ ORDER BY 列名1[ASC|DESC],列名2[ASC|DESC], ...

首先排列按列名1的值,若列名1的值相等则按列名2的值排列, 依次类推。

ORDER BY子句 (续)

[例]查询选修了'c01'号课程的学生的学号及其成绩,查询结果按分数降序排列。

[例]查询全体学生情况,查询结果按所在系的系号升序排列,同一系中的学生按年龄降序排列。

?? 先按职称的升序, 然后按工资的降序, 列出所有教师的信息。

ORDER BY子句 (续)

- -- 查询选修了'c01'这门课的学生的学号和成绩,
- -- 并用rank()和dense_rank()函数分别查看成绩的位次情况,
- >-- 并观察这两个函数的区别。

SELECT snum, score, rank() over (ORDER BY score DESC)

FROM sc

WHERE cnum='c01'

3.4.1 单表查询

- ▶ 查询仅涉及一个表:
 - 1.选择表中的若干列
 - 2.选择表中的若干元组
 - 3.ORDER BY子句
 - 4.聚集函数
 - 5.GROUP BY子句



- 4. 聚集函数
 - ▶ 聚集函数:
 - 统计元组个数 COUNT(*)
 - ► 统计一列中值的个数 COUNT([DISTINCT]ALL] <列名>)
 - ▶ 计算一列值的总和(此列必须为数值型) SUM([DISTINCT|ALL] <列名>)
 - ▶ 计算一列值的平均值(此列必须为数值型) AVG([DISTINCT|ALL] <列名>)
 - ▶ 求一列中的最大值和最小值 MAX([DISTINCT|<u>ALL</u>] <列名>) MIN([DISTINCT|<u>ALL</u>] <列名>)



4. 聚集函数(续)

- 需要传递一个参数
- ▶ 统计结果忽略NULL
- ▶不能嵌套



聚集函数 (续)

[例] 查询学生总人数。

[例] 查询选修了课程的学生人数。

[例] 计算'B3503021'号课程的学生平均成绩。

[例]查询学生年龄最大的是几岁



3.4.1 单表查询

- ▶ 查询仅涉及一个表:
 - 1.选择表中的若干列
 - 2.选择表中的若干元组
 - 3.ORDER BY子句
 - 4.聚集函数

5.GROUP BY子句



5. GROUP BY子句

GROUP BY子句分组:

细化聚集函数的作用对象

- 如果未对查询结果分组,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
- ▶按指定的一列或多列值分组,值相等的为一组

使用HAVING短语筛选最终输出结果

- 只有满足HAVING短语指定条件的组才输出
- ► HAVING短语与WHERE子句的区别: 作用对象不同
 - WHERE子句作用于基表或视图,从中选择满足条件的元组。
 - ▶ HAVING短语作用于组,从中选择满足条件的组。

[例] 求各个课程号及相应的选课人数,并按选课人数降序排列。

[例] 查询学生的学号和选修的课程门数。

[例] 查询选修了50门以上课程的学生学号。

注意: WHERE子句中是不能用聚集函数作为条件表达式 MySQL支持group_concat(), with rollup

[例]查询平均成绩大于等于90分的学生学号和平均成绩

[例]按平均工资从小到大的顺序列出各种职称教师的总工资、人数和平均工资。(职称非空,平均工资大于6000)

[例]分别列出男、女教师各种职称的人数和平均工资。

注意: 当按多个列值进行分组时, 所有的集合函数统计都是对最后的分组列进行的, 如上例中统计的是男或女教师某一种职称的人数和平均工资。

- ▶ HAVING短语与WHERE子句的区别:
 - 作用对象不同
 - ▶ WHERE子句作用于基表或视图,从中选择满足条件的元组
 - ▶ HAVING短语作用于组,从中选择满足条件的组。
- ▶ 参见爱课程网 数据库系统概论 数据查询节 动画《GROUP BY子句》

► GROUP_CONCAT()函数

SELECT ttitle,COUNT(*),group_concat(tname)

FROM t

GROUP BY ttitle

▶ With rollup显示本次查询所有记录的总和信息

SELECT ssex,count(*)

FROM s

GROUP BY ssex

WITH ROLLUP



注意:

- 1.GROUP BY子句的作用对象是查询的中间结果表.
- 2.分组方法:按指定的一列或多列值分组,值相等的为一组.
- 3.使用GROUP BY子句后,SELECT子句的列名列表中只能出现分组属性和集函数.

总结 (列)

- 选择所有列
- 选择若干列
- 算数表达式
- 函数
- 字符串



总结 (行)

查询条件	谓词
比较	=, >, <, >=, <=, !=, <>, !>, !<
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



- 对查询结果排序
- 对查询结果分组



难点解析

查询选修了50门以上课程是90分以上的学生的学号及(90分以上的)课程数,查询结果按课程数从多到少的顺序排列。

■ 按平均工资从高到低的顺序分别列出男、女教师各种职称的人数(任职人数多于100)和平均工资。 (筛选出工资非空信息)



■多重条件查询

列出1970年1月1日前出生,目前工资低于5000元或高于7000元的教师信息



数据库原理与技术

信息工程刘丽娟

