

数据库系统原理

第3章 关系数据库语言1



分级通关平台,交流



□ 基础篇

第1章 绪论

第2章 关系数据库*3

第3章 标准语言SQL*3

第4章 数据库安全性

第5章 数据库完整性

实验1

□ 设计与应用开发篇

第6章 关系数据理论*2

第7章 数据库设计

实验2

第8章 数据库编程

Ⅲ 系统篇

第9章 *关系查询处理和优化 实验3

第10章 数据库恢复技术 第11章 并发控制 实验4







内容提要

关系数据库标准语言SQL-1

- 1 SQL概述
- 2 学生-课程数据库
- 3 数据定义
- 4数据查询一单表

SQL-2

4 数据查询一多表

SQL-3

- 5 数据更新
- 6 空值的处理
- 7 视图







1. SQL概述

- SQL(Structured Query Language)结构化查询语言, 是一个通用的、功能极强的关系数据库的标准语言
 - ➤ SQL只要提出"做什么",操作过程由系统自动完成。
 - ◆ 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,能够<mark>嵌入到高级语言</mark>(例如C, C++, Java)程序中,供程序员设计程序时使用。





- 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
 - ◆ 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

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

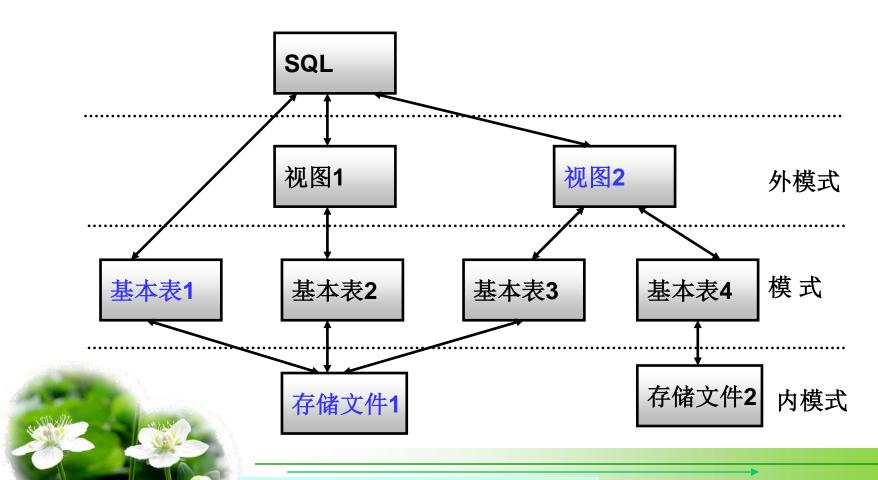




- 集数据定义语言(DDL),数据操纵语言(DML),数据控制语言(DCL)功能于一体。
- 可以独立完成数据库生命周期中的全部活动:
 - ●定义和<mark>修改</mark>、删除关系模式,定义和删除视图,插 入数据,建立数据库
 - 对数据库中的数据进行查询和更新
 - 数据库重构和维护
 - 数据库安全性、完整性控制,以及事务控制
 - ●嵌入式SQL和动态SQL定义
- 用户数据库投入运行后,可根据需要随时逐步修改模式,不影响数据库的运行。



SQL支持关系数据库三级模式结构



华中科技大学网络空间安全学院



- 基本表
 - 本身独立存在的表,SQL中一个关系就对应一个基本表
 - 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
- 存储文件
 - 逻辑结构组成内模式,物理结构对用户是<mark>隐蔽</mark>的
 - 一个(或多个)基本表对应一个存储文件







■ SQL功能极强,完成<mark>核心功能</mark>只用了9个动词。

表 3.2 SQL 的动词

SQL 功 能	动词
数据查询	SELECT
数据定义	CREATE, DROP, ALTER
数据操纵	INSERT, UPDATE, DELETE
数据控制	GRANT, REVOKE



2. 学生-课程 数据库

学生-课程模式 S-T: 学生关系Student、课程关系Course和选修关系SC

Student

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

Course

\# 40 D	\H.10 &	4. 4. 7田	314 V
课程号	课程名	先行课	学分
Cno	Cname	Cpno	Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	PASCAL语言	6	4

SC

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





3. 数据定义

- **❖SQL**的数据<mark>定义</mark>功能:
 - ■模式定义
 - 表定义
 - 视图和索引的定义





表 3.3 SQL 的数据定义语句

操作对象		操作方式	
	创 建	删除	修改
模式	CREATE SCHEMA	DROP SCHEMA	
表	CREATE TABLE	DROP TABLE	ALTER TABLE
视图	CREATE VIEW	DROP VIEW	
索引	CREATE INDEX	DROP INDEX	ALTER INDEX





模式





表以及视图、索引等

- 现代关系数据库管理系统提供了一个层次化的数据库对象**命名机制**
 - 一个关系数据库管理系统的实例(Instance)中可以建立多个数据库
 - 一个数据库中可以建立多个模式
 - 一个模式下通常包括多个表、视图和索引等<mark>数据</mark> 库对象







模式的定义

- 定义模式实际上定义了一个命名空间。
 - 在这个空间中可以定义该模式包含的数据库对象,例如基本表、视图、索引等。
- 在CREATE SCHEMA中可以接受CREATE TABLE,
 CREATE VIEW和GRANT子句。

CREATE SCHEMA <模式名> AUTHORIZATION <用户名>[<表定义子句 > | <视图定义子句> | <授权定义子句>];



[] 可选参数:



阅读

全民



模式的定义

[例3.1] 为用户WANG定义一个学生-课程模式S_T CREATE SCHEMA S_T AUTHORIZATION WANG;

[例3.2] CREATE SCHEMA AUTHORIZATION WANG;

◆ 该语句没有指定<模式名>, <模式名>隐含为<用户名>





模式的定义

[例3.3]为用户ZHANG创建了一个模式TEST,并且在其中定义一个表TAB1

CREATE SCHEMA TEST AUTHORIZATION ZHANG

CREATE TABLE TAB1 (COL1 SMALLINT,

COL2 INT,

COL3 CHAR(20),

COL4 NUMERIC(10,3),

COL5 DECIMAL(5,2)

);





模式的删除

- DROP SCHEMA <模式名> < CASCADE | RESTRICT>;
 - CASCADE (级联)
 - 删除模式的同时把该模式中所有的数据库对象全部删除
 - RESTRICT (限制)
 - 如果该模式中定义了下属的数据库对象(如表、视图等),则 拒绝该删除语句的执行。
 - 仅当该模式中没有任何下属的对象时才能执行。







模式的删除

[例3.4] DROP SCHEMA ZHANG CASCADE;

删除模式ZHANG

同时该模式中定义的表TAB1也被删除

DROP SCHEMA <模式名> < CASCADE | RESTRICT>;



■ 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3.5] 建立"学生"表Student。 学号是主码,姓名取值唯一。

 学号
 姓名
 性别
 年龄
 所在系

 Sno
 Sname
 Ssex
 Sage
 Sdept

 201215121
 李勇
 男
 20
 CS

主码

CREATE TABLE Student

(Sno CHAR(9) PRIMARY KEY,

/* 列级完整性约束条件,Sno是主码*/

Sname CHAR(20) UNIQUE,

Ssex CHAR(2),

Sage SMALLINT,

Sdept CHAR(20)

/* Sname取唯一值*/

UNIQUE 约束

表定义的注释





[例3.6] 建立一个"课程"表Course

课程号	课程名	先行课	学分
Cno	Cname	Cpno	Ccredit
1	数据库	5	4

CREATE TABLE Course

(Cno CHAR(4) PRIMARY KEY,

Cname CHAR(40),

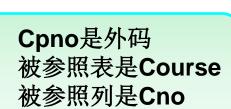
Cpno CHAR(4),

Ccredit SMALLINT,

FOREIGN KEY (Cpno) REFERENCES Course(Cno)

);









[例3.7] 建立一个学生选课表SC

CREATE TABLE SC

(Sno CHAR(9),

Cno CHAR(4),

Grade SMALLINT,

PRIMARY KEY (Sno,Cno),

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

/* 主码由两个属性构成,必须作为表级完整性进行定义*/

FOREIGN KEY (Sno) REFERENCES Student(Sno),

/* 表级完整性约束条件,Sno是外码,被参照表是Student */

FOREIGN KEY (Cno)REFERENCES Course(Cno)

/* 表级完整性约束条件, Cno是外码,被参照表是Course*/



属性的数据类型

- 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

- ✓ Sno CHAR(9)
- ✓ Cno CHAR(4)
- ✓ Grade SMALLINT







数据类型	含义
CHAR(n),CHARACTER(n)	长度为n的 <mark>定长字符串</mark>
VARCHAR(n), CHARACTERVARYING(n)	最大长度为n的 <mark>变长字符串</mark>
CLOB	字符串大对象
BLOB	二进制大对象
INT, INTEGER	长整数(4字节)
SMALLINT	短整数(2字节)
BIGINT	大整数(8字节)
NUMERIC(p, d)	定点数,由 p 位数字(不包括符号、小数点)组成,小数后面有 d 位数字
DECIMAL(p, d), DEC(p, d)	同NUMERIC
REAL	取决于机器精度的单精度浮点数
DOUBLE PRECISION	取决于机器精度的双精度浮点数
FLOAT(n)	可选精度的浮点数,精度至少为 n 位数字
BOOLEAN	逻辑布尔量
DATE	日期,包含年、月、日,格式为 YYYY-MM-DD
TIME	时间,包含一日的时、分、秒,格式为 HH:MM:SS
TIMESTAMP	时间戳类型
INTERVAL	时间间隔类型



索引的建立与删除

- 建立索引的目的: 加快查询速度
 - 顺序文件上的索引
 - B+树索引
 - 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









阅读

全民

索引的建立与删除

■ 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<mark>聚簇索引</mark>





索引的建立

[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);





索引的管理

- 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





索引的修改与删除

■ 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;

■ 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;



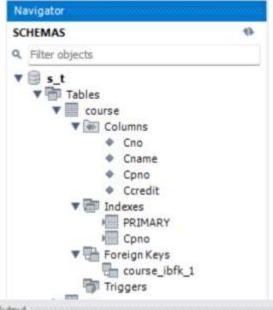


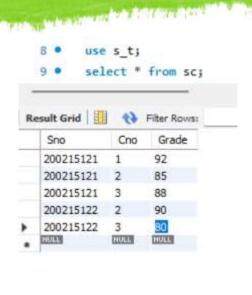
数据字典

- 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。









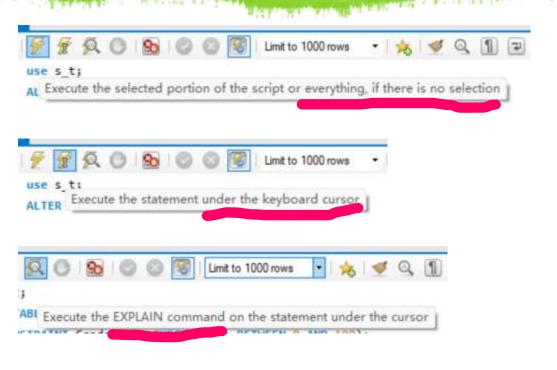


đ	Actio	on Output	•		
	*	Time	Action	Message	Duration / Fetch
9	4	08:37:37	use s_t	0 row(s) affected	0.000 sec
3	5	08:37:43	EXPLAIN explain select * from sc	Error Code: 1064 You have an error in your SQL syntax; check the	
9	6	08:38:53	explain select * from sc	1 row(s) returned	0.016 sec / 0.000 sec
9	7	08:39:24	explain select grade from sc limit 0,1	1 row(s) returned	0.000 sec / 0.000 sec
9	8	08:46:11	select * from sc LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

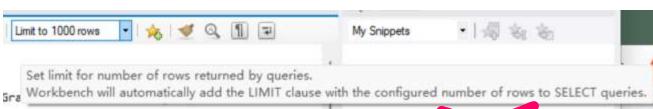


MySQL环境











MySQL环境

华中科技大学网络空间安全学院















学习行为







4. 数据查询一单表

■ 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]];





select * from 表名; 然后加条件;



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



select * from T where 1=1 order by b1;



数据查询一单表

- 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





数据查询一单表(选择表中的若干列)

■ 查询指定列

[例3.16] 查询全体学生的学号与姓名。 SELECT Sno,Sname FROM Student;

[例3.17] 查询全体学生的姓名、学号、所在系。 SELECT Sname, Sno, Sdept FROM Student;





- 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3.18] 查询全体学生的详细记录

SELECT Sno, Sname, Ssex, Sage, Sdept

FROM Student;

或

SELECT *

FROM Student;





- 查询经过计算的值
 - SELECT子句的<目标列表达式>不仅可以为表中的属性列, 也可以是表达式

[例3.19] 查全体学生的姓名及其出生年份。

SELECT Sname, 2014-Sage

/*假设当时为2014年*/

FROM Student;

输出结果:

Sname	2014-Sage
李勇	1994
刘晨	1995
王敏	1996
张立	1995



[例**3.20**] 查询全体学生的姓名、出生年份和所在的院系,要求用 小写字母表示系名。

SELECT Sname, 'Year of Birth: ',2014-Sage, LOWER (Sdept) FROM Student;

输出结果:

Sname 'Year of Birth:' 2014-Sage LOWER(Sdept)

李勇 Year of Birth: 1994 cs

刘晨 Year of Birth: 1995 cs

王敏 Year of Birth: 1996 ma

张立 Year of Birth: 1995 is





■ 使用列别名改变查询结果的列标题

SELECT Sname NAME,'Year of Birth:' BIRTH,
2014-Sage BIRTHDAY,LOWER(Sdept) DEPARTMENT
FROM Student;

输出结果:

NAME	BIRTH	BIRTHDAY	DEPARTMENT	
李勇	Year of Birth	n: 1994	CS	
刘晨	Year of Birth	n: 1995	CS	
王敏	Year of Birth	n: 1996	ma	
张立	Year of Birth	n: 1995	is	



数据查询一单表(选择表中的若干元组)

■ 如何消除取值重复的行 如果没有指定DISTINCT关键词,则缺省为ALL

[例3.21] 查询选修了课程的学生学号。

SELECT Sno FROM SC;

等价于:

SELECT ALL Sno FROM SC;

执行上面的SELECT语句后,结果为:

Sno

SELECT DISTINCT Sno FROM SC;

执行结果:

Sno

201215121 201215122





数据查询一单表(查询满足条件的元组)



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3.22] 查询计算机科学系全体学生的名单。

SELECT Sname

FROM Student

WHERE Sdept='CS';

[例3.23]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;

[例3.24]查询考试成绩有不及格的学生的学号。

SELECT DISTINCT Sn

FROM SC

WHERE Grade < 60;



■ 谓词: BETWEEN ... AND ...

NOT BETWEEN ... AND ...

[例3.25] 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;

[例3.26] 查询年龄不在20~23岁之间的学生姓名、系别和年龄

SELECT Sname, Sdept, Sage

FROM Student

WHERE Sage NOT BETWEEN 20 AND 23;



■ 谓词: IN <值表>, NOT IN <值表>

[例3.27]查询计算机科学系(CS)、数学系(MA)和信息系(IS) 学生的姓名和性别。

SELECT Sname, Ssex

FROM Student

WHERE Sdept IN ('CS','MA','IS');

[例3.28]查询既不是计算机科学系、数学系,也不是信息系的学生的姓名和性别。

SELECT Sname, Ssex

FROM Student

WHERE Sdept NOT IN ('IS','MA','CS');





- 谓词: [NOT] LIKE `<匹配串>' [ESCAPE`<换码字符>']
 - % 代表任意长度(长度可以为0) 字符串, _代表任意单个字符

[例3.29] 查询学号为201215121的学生的详细情况。

SELECT *

FROM Student

WHERE Sno LIKE '201215121';

例3.30] 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3.31] 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_';

华中科技大学网络空间安全学院



■ 使用换码字符将通配符转义为普通字符

[例3.34] 查询DB_Design课程的课程号和学分。

SELECT Cno, Ccredit

FROM Course

WHERE Cname LIKE 'DB_Design' ESCAPE '\';

[例3.35] 查询以"DB_"开头,且倒数第3个字符为 i的课程的详细情况。

SELECT *

FROM Course

WHERE Cname LIKE 'DB_%i_ _' ESCAPE '\ ';



ESCAPE'\'表示"\"为换码字符



- ☀ 谓词: IS NULL 或 IS NOT NULL
 - "IS" 不能用 "=" 代替

[例3.36] 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。

SELECT Sno, Cno

FROM SC

WHERE Grade IS NULL;

[例3.37] 查所有有成绩的学生学号和课程号。

SELECT Sno, Cno

FROM SC

WHERE Grade IS NOT NULL;







- 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

[例3.38] 查询计算机系年龄在20岁以下的学生姓名。

SELECT Sname

FROM Student

WHERE Sdept= 'CS' AND Sage<20;

[例3.27] SELECT Sname, Ssex

FROM Student

WHERE Sdept IN ('CS ','MA ','IS')

可改写为:

SELECT Sname, Ssex

FROM Student

WHERE Sdept= 'CS' OR Sdept= 'MA' OR Sdept= 'IS';



- ORDER BY子句
 - ▶ 可以按一个或多个属性列排序;升序ASC;降序DESC;缺省升序
 - » 对于<mark>空值</mark>,排序时显示的次序由具体系统实现来决定

[例3.39]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;

[例3.40]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。

SELECT *

FROM Student

ORDER BY Sdept, Sage DESC;



華中科技大學

数据查询一单表

- 聚集函数:
 - 统计元组个数COUNT(*)
 - 统计一列中<mark>值的个数</mark>
 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|<u>ALL</u>] <列名>)
 - 求一列中的最大值和最小值
 MAX([DISTINCT|ALL] <列名>)
 MIN([DISTINCT|ALL] <列名>)







```
[例3.41] 查询学生总人数。
```

SELECT COUNT(*)

FROM Student;

[例3.42] 查询选修了课程的学生人数。

SELECT COUNT(DISTINCT Sno)

FROM SC;

[例3.43] 计算1号课程的学生平均成绩。

SELECT AVG(Grade)

FROM SC

WHERE Cno= '1';





[例3.44] 查询选修1号课程的学生最高分数。

SELECT MAX(Grade)

FROM SC

WHERE Cno='1';



[例3.45] 查询学生201215012选修课程的总学分数。

SELECT SUM(Ccredit)

FROM SC, Course

WHERE Sno='201215012' AND SC.Cno=Course.Cno;





■ GROUP BY子句分组:

<mark>细化聚集函数的作用对象</mark>,按指定的一列或多列值分组,值相等的为一组

- 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

[例3.46] 求<mark>各个</mark>课程号及相应的选课人数。

SELECT Cno, COUNT(Sno) FROM SC

GROUP BY Cno;

查询结果可能为:

Cno	COUNT(Sno)	
1	22	
2	34	
3	44	
4	33	
5	48	







[例3.48]查询平均成绩大于等于90分的学生学号和平均成绩

下面的语句是不对的

SELECT Sno, AVG(Grade)

FROM SC

WHERE AVG(Grade)>=90

GROUP BY Sno;

因为WHERE子句中是不能用聚集函数作为条件表达式

正确的查询语句应该是:

SELECT Sno, AVG(Grade)

FROM SC

GROUP BY Sno

HAVING AVG(Grade)>=90;





- 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3.47] 查询选修了3门以上课程的学生学号。

SELECT Sno

FROM SC

GROUP BY Sno

HAVING COUNT(*) >3;





小结

- ➤ **SQL**概述
- > 关系与关系模式

模式的定义与删除 基本表的定义、删除与修改 索引的建立与删除



▶ 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







■ 选择管理员: 命令提示符 - mysql -u root -p

C:\WINDOWS\system32>net start mysql

MySQL 服务正在启动 MySQL 服务已经启动成功。

C:\WINDOWS\system32>mysq1 -u root -p

Enter password:

Welcome to the MySQL monitor. Commands end with; or \g .

Your MySQL connection id is 2

Server version: 5.7.27 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input

mysql>

d	name	age	gender	is_del
1 2 3 4 5 6 7	张李小王王王王王王王五三四五	18 22 16 32 23 24 25	男男女男男男男	0 0 0 0 0

7 rows in set (0.00 sec)

mysql>





