SQL语言

1. SQL语言
2. 什么是sql(Structured Query Language)语言:

数据库的结构化查询语言.

注意:sql语言是数据库的通用语言.

1. Sql语言组成:

DML数据操作语言:

用来操作数据库中所包含数据.(insert,update,delete)

DQL数据查询语言:

用来对数据库中的数据进行查询.(select)

DDL数据定义语言:

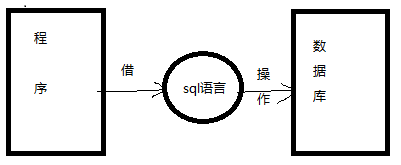
用于创建和删除数据库对象等操作.如建库建表(create、drop、alter)

DCL数据控制语言:

用来控制数据库组件的存取许可、存取权限.(grant、commit、rollback)

1. Sql语言的作用

通过sql语言操作数据库



1. sql文件的扩展名是:.sql
2. 使用DML和 DQL操作表中的数据

1.查询语句

SELECT \* FROM 表名;

注意:\*表示所有列

示例:

SELECT \* FROM students;

2.插入数据(添加数据)语法:

INSERT INTO 表名(列1,列2...) VALUES(值1,值2);

注意:

a.值为文本类型或者手动日期类型需加单引号,数值类型不加

b.添加可选列:为空的列、默认值列、自动增长列可不加

c.值为default表示默认值

示例:添加学生

-- 添加所有列

INSERT INTO students(xh,NAME,age,sex,birthday,address,tel,state,gid)

VALUES(1007,'李四',21,'男','2019-1-1','湖北宜昌','12345',1,2);

INSERT INTO students(xh,NAME,age,sex,birthday,address,tel,state,gid)

VALUES(1011,'冷血',21,'男',NULL,DEFAULT,'12345',1,2); -- 采用默认

INSERT INTO students(xh,NAME,age,sex,birthday,address,tel,state,gid)

VALUES(1012,'追命',21,'男',NOW(),DEFAULT,'12345',1,2); -- 获得系统日期

-- 添加部分列

INSERT INTO students(xh,NAME,state,gid)

VALUES(1009,'无情',1,2);

-- 同时添加多行

INSERT INTO `students`(`xh`,`name`,`age`,`sex`,`birthday`,`address`,`tel`,`state`,`gid`)

VALUES (1002,'杨过',22,'男','2019-11-04','北京','123456654',1,1)

,(1003,'小龙女',18,'男','2019-11-05','不详','564322345',0,1),

(1004,'王建兵',24,'男','2019-11-18','湖北省武汉市汉阳区某小区19栋1901','234565444',0,2),

(1005,'新任班长姜晖',41,'男','2019-11-17','大家投我，谢谢大家','+',1,1),

(1006,'马明伍副班长',40,'男','2019-11-17','不详',NULL,1,2),

(1007,'李四',34,'男','2019-11-18','不详','12345',1,2)

,(1009,'无情',NULL,'男',NULL,'不详',NULL,1,2);

3. 修改语句的语法

UPDATE 表名 SET 列1=新值,列2=新值... [WHERE 条件]

注意:加where条件就是修改满足条件的行，否则修改所有

示例:

-- 修改某列所有行

UPDATE students SET sex='女'

-- 修改学号为1007的性别为男，年龄22,生日为系统日期，地址为默认

UPDATE students SET sex='男',age=22,birthday=NOW(),address=DEFAULT

WHERE xh=1007;

-- 修改学号在1005到1007的学生性别为男,年龄在原有基础加3

-- update students set sex='男',age=age+3 where xh=1005 or xh=1007 or xh=1006

-- update students set sex='男',age=age+3 where xh>=1005 and xh<=1007

-- UPDATE students SET sex='男',age=age+3 WHERE xh between 1005 and 1007

-- UPDATE students SET sex='女',age=age+3 WHERE xh in(1005,1006,1007)

-- 修改年龄为null的学生性别为男

UPDATE students SET sex='男' WHERE age IS NULL; -- 空值比较

-- 修改年级编号不为空的所有学生性别改为男

UPDATE students SET sex='男' WHERE gid IS NOT NULL; -- 空值比较

4.删除语句语法

DELETE FROM 表名 [WHERE 条件]

注意:where条件不给则删除所有，否则删除满足条件的行

示例:

删除某学号的学生

DELETE FROM students WHERE xh=1013;

删除学号为1008,1011的学生

DELETE FROM students WHERE xh IN(1008,1011);

删除学号在20和25之间的

DELETE FROM students WHERE age BETWEEN 20 AND 21;

-- 删除所有

DELETE FROM students;

5.查询语句的语法

SELECT \* FROM 表名 [WHERE 条件] [ORDER BY 子句]

注意:

a. \*代表所有列

b. ORDER BY 子句用于排序:

语法: ORDER BY 列 排序规则:asc默认升序|desc降序

示例:

-- 查询所有行所有列

SELECT \* FROM students;

-- 查询部分列 查询学号，姓名，地址，性别

SELECT xh,NAME,address,sex FROM students;

-- 查询部分列 查询学号，姓名，地址，性别 并使用as关键取别名

SELECT xh 学号,NAME AS 姓名,address AS 地址,sex AS 性别 FROM students;

-- 查询部分行 所有性别为男的学生

-- select \* from students where sex='男';

查询学生年龄在20-25之间，性别为男的学生

SELECT \* FROM students WHERE age BETWEEN 20 AND 25 AND sex='男';

-- 查询年龄在20-25之间学生的学号，姓名，地址 取别名

SELECT xh AS 学号,NAME AS 姓名,address 地址 FROM

students WHERE age BETWEEN 20 AND 25;

-- 查询常量列和计算列

SELECT xh,'姓名',NAME,age\*xh FROM students;

-- 排序

示例:

查询所有学生 按年龄排序 升序

SELECT \* FROM students ORDER BY age ASC;

查询所有男生，按年龄降序排列

SELECT \* FROM students WHERE sex='男' ORDER BY age DESC

示例: 多列排序

查询所有的学生信息，先按级升序排序，再按年龄降序

SELECT \* FROM students ORDER BY gid ASC,age DESC;

-- 查询前多少条

示例:查询学生表的前三条信息

SELECT \* FROM students LIMIT 0,3

查询年龄最大的前三名学生

SELECT \* FROM students ORDER BY age DESC LIMIT 0,3

-- 分页查询

SELECT \* FROM students LIMIT (页码-1)\*页大小,页大小;

示例:每页5条，显示第一页

SELECT \* FROM students LIMIT 5,5;