



# 第13章 Qt 5数据库

## 13.1 数据库基本概念

## 13.2 常用SQL命令





# 13.1 数据库基本概念

## 1. 数据和数据库 (DB)

利用计算机进行数据处理，首先需要将信息以数据形式存储到计算机中，因为数据是可以被计算机接收和处理的符号。

## 2. 数据库管理系统 (DBMS)

数据库管理系统即DBMS (DataBase Management System)，它是位于用户应用程序和操作系统之间的数据库管理系统软件，其主要功能是组织、存储和管理数据，高效地访问和维护数据，即提供数据定义、数据操纵、数据控制和数据维护等功能。常用的数据库管理系统有Oracle、Microsoft SQL Server、Sybase和DB2等。

## 3. 结构化查询语言SQL

SQL语言由以下三部分组成。

(1) 数据定义语言 (Data Description Language, DDL)，用于执行数据库定义的任务，对数据库及数据库中的各种对象进行创建、删除和修改等操作。数据库对象主要包括表、默认约束、规则、视图、触发器和存储过程等。

(2) 数据操纵语言 (Data Manipulation Language, DML)，用于操纵数据库中各种对象，检索和修改数据。

(3) 数据控制语言 (Data Control Language, DCL)，用于安全管理，确定哪些用户可以查看或修改数据库中的数据。



# 13.1 数据库基本概念

## 4. 表和视图

### (1) 表 (Table)。

表是在日常工作和生活中经常使用的一种表示数据及其关系的形式，如表 13.1 为一个学生表。

学 号	姓 名	专 业 名	性 别	出 生 时 间
130201	王 一	计算机	男	1994/10/01
130202	王 巍	计算机	女	1995/02/08
130302	林 滔	电子工程	男	1994/04/06
130303	江为中	电子工程	男	1997/12/08

### (2) 视图 (View)。

视图是从一个或多个表（或视图）导出的表。



## 13.2 常用SQL命令

### 13.2.1 数据查询

#### 1. SELECT语句

完备的SELECT语句很复杂，它主要的子句如下：

```
SELECT [DISTINCT] [别名.]字段名或表达式 [AS 列标题] /* 指定要选择的列或行及其限定 */  
FROM table_source /* FROM子句，指定表或视图 */  
[ WHERE search_condition ] /* WHERE子句，指定查询条件 */  
[ GROUP BY group_by_expression ] /* GROUP BY子句，指定分组表达式 */  
[ ORDER BY order_expression [ ASC | DESC ] /* ORDER BY子句，指定排序表达式和顺序 */
```



## 13.2.1 数据查询

下面用示例说明使用SQL语句对Student数据库进行的各种查询。

(1) 查询Student数据库。查询students表中各个同学的姓名和总学分。

```
USE Student SELECT name,totalscore FROM students
```

(2) 查询表中所有记录。查询students表中各个同学的所有信息。

```
SELECT * FROM students
```

(3) 条件查询。查询students表中总学分大于等于120的同学的情况。

```
SELECT * FROM students WHERE totalscore >= 120
```

(4) 多重条件查询。查询students表中所在系为“计算机”且总学分大于等于120的同学的情况。

```
SELECT * FROM students WHERE department='计算机' AND totalscore >= 120
```





## 13.2.1 数据查询

(5) 使用LIKE谓词进行模式匹配。查询students表中姓“王”且单名的学生情况。

```
SELECT * FROM students WHERE name LIKE '王_'
```

(6) 用BETWEEN...AND指定查询范围。查询students表中不在1997年出生的学生情况。

```
SELECT * FROM students  
WHERE birthday NOT BETWEEN '1997-1-1' and '1997-12-31'
```

(7) 空值比较。查询总学分尚不确定的学生情况。

```
SELECT * FROM students  
WHERE totalscore IS NULL
```

(8) 自然连接查询。查找计算机系学生姓名及其“C程序设计”课程的考试分数情况。

```
SLELCT name,grade  
FROM students, courses,grades,  
WHERE department = '计算机' AND coursename= 'C程序设计' AND  
students.studentid = grades.studentid AND courses.courseid =  
grades.coursesid
```



## 13.2.1 数据查询

(9) IN子查询。查找选修了课程号为101的学生情况。

```
SELECT * FROM students  
WHERE studentid IN  
    ( SELECT studentid FROM courses WHERE courseid = '101' )
```

在执行包含子查询的SELECT语句时，系统首先执行子查询，产生一个结果表，再执行外查询。本例中，首先执行子查询：

```
SELECT studentid FROM courses, students, grades WHERE courseid = '101'  
AND students.studentid = grades.studentid AND courses.courseid =  
    grades.coursesid
```



## 13.2.1 数据查询

(10) 比较子查询。这种子查询可以认为是IN子查询的扩展，它是表达式的值与子查询的结果进行比较运算。查找课程号206的成绩不低于课程号101的最低成绩的学生学号。

```
SELECT studentid FROM grades
WHERE courseid = '206' AND grade !< ANY
  ( SELECT grade FROM grades
    WHERE courseid = '101'
  )
```

(11) EXISTS子查询。EXISTS谓词用于测试子查询的结果是否为空表，若子查询的结果集不为空，则EXISTS返回TRUE，否则返回FALSE。EXISTS还可与NOT结合使用，即NOT EXISTS，其返回值与EXISTS刚好相反。查找选修206号课程的学生姓名。

```
SELECT name FROM students
WHERE EXISTS
  ( SELECT * FROM grades
    WHERE studentid = students.studentid AND courseid = '206'
  )
```





## 13.2.1 数据查询

(12) 查找选修了全部课程的同学姓名（即查找没有一门功课不选修的学生）。

```
SELECT name FROM students
WHERE NOT EXISTS
  ( SELECT * FROM courses
    WHERE NOT EXISTS
      ( SELECT * FROM grades
        WHERE studentid= students.studentid
          AND courseid=courses.courseid
        )
    )
```



## 13.2.1 数据查询

(13) 查询结果分组。将各课程成绩按学号分组。

```
SELECT studentid,grade FROM grades  
GROUP BY studentid
```

(14) 查询结果排序。将计算机系的学生按出生时间先后排序。

```
SELECT * FROM students  
WHERE department = '计算机'  
ORDER BY birthday
```



## 13.2.1 数据查询

### 2. 常用聚合函数

常用的聚合函数列于表13.2中。

函 数 名	说 明
AVG	求组中值的平均值
COUNT	求组中项数，返回 int 类型整数
MAX	求最大值
MIN	求最小值
SUM	返回表达式中所有值的和
VAR	返回给定表达式中所有值的统计方差



## 13.2.1 数据查询

(1) 求选修101课程学生的平均成绩。

```
SELECT AVG(grade) AS '课程101平均成绩'  
FROM grades  
WHERE courseid = '101'
```

(2) 求选修101课程学生的最高分和最低分。

```
SELECT MAX(grade) AS '课程101最高分', MIN(grade) AS '课程101最低分'  
FROM grades  
WHERE courseid = '101'
```

(3) 求学生的总人数。

```
SELECT COUNT(*) AS '学生总数'  
FROM students
```



## 13.2.2 数据操作

### 1. 插入数据语句INSERT

INSERT可添加一条或多条记录至一个表中。

INSERT有两种语法形式。

语法1:

```
INSERT INTO target [IN externaldatabase] (fields_list)
{DEFAULT VALUES | VALUES (DEFAULT | expression_list) }
```

语法2:

```
INSERT INTO target [IN externaldatabase] fields_list
{SELECT... | EXECUTE...}
```





## 13.2.2 数据操作

### 2. 删除数据语句DELETE

DELETE用于从一个或多个表中删除记录。

DELETE语句的语法格式如下：

```
DELETE FROM table_names  
[WHERE...]
```

例如，用以下语句从students表中删除姓名为“罗亮”的记录：

```
DELETE FROM students  
WHERE name = '罗亮'
```



## 13.2.2 数据操作

### 3. 更新数据语句UPDATE

UPDATE语句用于更新表中的记录。

UPDATE语句的语法格式如下：

UPDATE table\_name

SET Field\_1=expression\_1[,Field\_2=expression\_2...]

[FROM table1\_name|view1\_name[,table2\_name|view2\_name...]]

[WHERE...]

例如，以下语句将计算机系学生的总分增加10：

UPDATE students

SET totalscore = totalscore +10

WHERE department = '计算机'