

数据库相关基本概念

- 数据库：按照数据结构来组织、存储和管理数据的建立在计算机存储设备上的仓库。现在主流的是关系数据库。数据里面使用一张或以上的**表**（table）来按业务存储数据。
- 数据库表：以现实世界中的表格引申而来的概念。在数据库中，每个表都有自己的名字（table_name），用于区分它们各自的作用。一个数据库表含有1个或以上的数据库**列**（column），每个列都有自己的名字(column_name)来代表这一列的含义；数据库表中可以有**0行**(row)或以上的数据。如下例子：

水果	价格	数量
香蕉	\$1	5
苹果	\$1	6
草莓	\$1	7

这个表中有3列，列名分别是：水果，价格，数量，它们说明了一行数据中相应的列的数据元素表示的东西。

然后，这个表中有3行数据，分别记录了香蕉，苹果，草莓3中水果的价格和数据。

sql基础

假设当前有以下数据表

student:

id	name	age	class
1	Tom	22	计算机
2	Jim	21	经济
3	Ada	23	计算机

score:

id	student_id	course	score
1	1	C语言	90
2	1	数据库	85
3	2	宏观经济学	88
4	2	古典经济学	70

1. SELECT

- 作用：SELECT 语句用于从数据库中选取数据。结果被存储在一个结果表中，称为结果集。

- 语法:

```
SELECT column_name1,column_name2 FROM table_name;
```

- 示例:

查询student表: `select id, name, age, class from student`

结果:

id	name	age	class
1	Tom	22	计算机
2	Jim	21	经济
3	Ada	23	计算机

如果只想查询name和age这2列, 则是: `select name, age from student`

结果:

name	age
Tom	22
Jim	21
Ada	23

2. WHERE 子句

- 作用: WHERE 子句用于提取那些满足指定标准的记录, 也就是根据条件筛选需要的数据
- 语法:

```
SELECT column_name,column_name  
FROM table_name  
WHERE column_name operator(运算符, 后面有讲) value;
```

- 示例: 查询student表中计算机班的数据:

`select id, name, age, class from student where class = '计算机'`

结果:

id	name	age	class
1	Tom	22	计算机
3	Ada	23	计算机

- where中可以使用的基本的运算符：

运算符	描述	例子
=	等于，筛选与条件相等的数据	class = '计算机'
<>	不等于。作用与等于相反。 注释： 在 SQL 的一些版本中，该操作符可被写成 !=	class = '计算机'
>	大于，筛选比条件大的数据	age > 22
<	小于，筛选比条件小的数据	age < 22
>=	大于等于，筛选比条件大的数据或等于条件的数据	age >= 22
<=	小于等于，筛选比条件小的数据或等于条件的数据	age <= 22
BETWEEN	在某个范围内	age BETWEEN 21 and 23
LIKE	搜索某种模式，既模糊搜索	class like '%算%'（注释：查询class中含有'算'字的数据）
IN	指定针对某个列的多个可能值	age in (21, 23)（注释：查询age是21或23岁的数据，括号里面的条件的关系是“或”关系）

“

like中的条件注释：

<p>% 表示多个字值，_ 下划线表示一个字符；</p> <p>M% ：为能配符，正则表达式，表示的意思为模糊查询信息为 M 开头的。</p> <p>%M% ：表示查询包含M的所有内容。</p> <p>%M_ ：表示查询以M在倒数第二位的所有内容。</p>
--

3. AND & OR 运算符

- 作用：用于基于一个以上的条件对记录进行过滤。

如果第一个条件和第二个条件都成立，则 AND 运算符显示一条记录。

如果第一个条件和第二个条件中只要有一个成立，则 OR 运算符显示一条记录。

- 语法：

```
SELECT column_name,column_name
FROM table_name
WHERE column_name1 operator value
      and column_name2 operator value
      or column_name3 operator value ;
```

- 示例：

AND: 查询查询22岁并且是计算机班的学生：

```
select id, name, age, class from student where age = 22 and
class = '计算机'
```

结果：

id	name	age	class
1	Tom	22	计算机

OR: 查询22岁或经济班的学生

```
select id, name, age, class from student where age = 22 or
class = '经济'
```

结果：

id	name	age	class
1	Tom	22	计算机
2	Jim	21	经济

4. ORDER BY 关键字

- 作用：用于对结果集按照一个列或者多个列进行排序。默认按照升序对记录进行排序。如果需要按照降序对记录进行排序，您可以使用 DESC 关键字。
- 语法：

```
SELECT column_name1,column_name2
FROM table_name
ORDER BY column_name1,column_name2 ASC|DESC;
```

- 示例:

查询学生表，并按年龄排序: `select id, name, age, class from student order by age`

结果:

id	name	age	class
2	Jim	21	经济
1	Tom	22	计算机
3	Ada	23	计算机

5. 别名

- 作用: 为表名称或列名称指定别名。基本上，创建别名是为了让列名称的可读性更强。或在多表查询时，区分不同表间的相同列名。
- 语法:

列的 SQL 别名语法

```
SELECT column_name (AS) alias_name
FROM table_name;
```

表的 SQL 别名语法

```
SELECT column_name(s)
FROM table_name (AS) alias_name;
```

注释: 现在主流的数据库引擎都可以忽略AS关键字。

- 示例: 将查询结果中的列名变成中文: `select id, name '姓名', age '年龄', class '班级' from student`

结果:

id	姓名	年龄	班级
1	Tom	22	计算机
2	Jim	21	经济
3	Ada	23	计算机

使用表别名指定查询的列: `select s.id, s.name '姓名', s.age '年龄', s.class '班级' from student s`

结果:

id	姓名	年龄	班级
1	Tom	22	计算机
2	Jim	21	经济
3	Ada	23	计算机

6. 多表关联查询

- 作用: 将两个或以上的表按照特定条件筛选后返回结果集。
- 语法:

```
select t1.column_name c1, t2.column_name c2
from table1 t1, table2 t2
where t1.column operator t2.column
```

- 示例:

查询学生的成绩:

```
select stu.name , s.course, s.score from student stu, score s
where s.student_id = stu.id
```

结果:

name	course	score
Tom	C语言	90
Tom	数据库	85
Jim	宏观经济学	88
Jim	古典经济学	70

7. INSERT INTO 语句

- 作用: 用于向表中插入新记录。
- 语法: INSERT INTO 语句可以有两种编写形式。

第一种形式无需指定要插入数据的列名, 只需提供被插入的值即可:

```
INSERT INTO table_name
VALUES (value1,value2,value3,...);
```

第二种形式需要指定列名及被插入的值:

```
INSERT INTO table_name (column1,column2,column3,...)
VALUES (value1,value2,value3,...);
```

- 示例:

向student表中增加一条记录:

第一种形式:

```
INSERT INTO student
VALUES (4,'Jack',21, '通信工程');
```

第二种形式:

```
INSERT INTO student(id, name, age, class)
VALUES (4,'Jack',21, '通信工程');
```

8. UPDATE 语句

- 作用: 用于更新表中已存在的记录。
- 语法:

```
UPDATE table_name
SET column1=value1,column2=value2,...
WHERE some_column=some_value;
```

请注意 SQL UPDATE 语句中的 WHERE 子句!

WHERE 子句规定哪条记录或者哪些记录需要更新。如果您省略了 WHERE 子句, 所有的记录都将被更新!

- 示例:

将Tom的年龄改成25:

```
update student
  set age = 25
  where name = 'Tome';
```

9. DELETE 语句

- 作用：用于删除表中的记录。
- 语法：

```
DELETE FROM table_name
WHERE some_column=some_value;
```

- 示例：

删除Jim宏观经济学的成绩：

```
-- 知道主键时，可以直接通过主键删除
delete score where id = 3
```

如果不知道主键，则需要通过关联student表查询记录，就更复杂一点的

```
delete score s
  where s.course = '宏观经济学'
        and s.student_id = (select stu.id from student stu where
                             stu.name = 'Jim')
```

注意：执行delete语句前，可以先用select语句来执行一下delete的where子句，看一下得到的记录是不是想删除的。