

SQL(Structured Query Language)

作者: [468336329Zc](#)

原文链接: <https://ld246.com/article/1572359188457>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

<h2 id="SQL">SQL</h2>
<h2 id="SQL分类">SQL 分类</h2>
<table>
<thead>
<tr>
<th>SQL 分类</th>
<th>操作对象</th>
<th>操作</th>
<th>关键字</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDL（数据定义语言）</td>
<td>数据库，表，视图和索引</td>
<td>创建，删除，修改</td>
<td>CREATE, DROP, ALTER</td>
</tr>
<tr>
<td>DCL（数据控制语言）</td>
<td>数据库</td>
<td>数据库的事务管理</td>
<td>grant</td>
</tr>
<tr>
<td>DML（数据操纵语言）</td>
<td>数据内容</td>
<td>增删改</td>
<td>INSERT INTO, DELETE, UPDATE</td>
</tr>
<tr>
<td>DQL（数据查询语言）</td>
<td>数据内容</td>
<td>查询</td>
<td>SELECT</td>
</tr>
</tbody>
</table>
<h2 id="DDL-数据定义-">DDL（数据定义）</h2>
<p>数据定义，数据库的创建修改删除</p>
<h2 id="DCL-数据控制-">DCL(数据控制)</h2>
<p>事务管理</p>
<p>授权：</p>
<h2 id="DML-数据操纵-">DML（数据操纵）</h2>
<p>数据操纵语言，对数据进行增删改</p>
<h2 id="DQL-数据查询-">DQL（数据查询）</h2>
<p>□ SQL 注释和注意事项</p>
<h3 id="1-简单查询">1.简单查询</h3>
<h4 id="1-基本的查询语句">1) 基本的查询语句</h4>
<h5 id="where条件查询">where 条件查询</h5>
<p>select name from student where id=1; </p>
<p>(查询 id 等于 1 的学生的名字)</p>
<hr>
<p>组合 where 子句</p>

```

<p>操作符 and</p>
<p>select name from student where id>2 and id<6 and age=18;</p>
<p>(and 操作符将条件组合在一起，每增加一个过滤条件就要添加一个 and)</p>
<hr>
<p>操作符 or</p>
<p>select * from student where age>8 or name like "zhang%";</p>
<p>(查询年龄大于 8 岁或者名字是 zhang 开头的 所有人的 学生)</p>
<hr>
<p>计算次序 当 or 与 and 一起出现的时候 and 优先级更高</p>
<p>select * from student where age>8 or age<18 and name like "zhang%";</p>
<p>(查询年龄小于 18 而且名字是 zhang 开头的 学生 或者年龄只需要大于 8 岁的学生)</p>
<p>不过最好用括号()来明确计算次序，括号括起来的先计算</p>
<hr>
<p>in 操作符</p>
<p>select *from student where id in(1,4,2,7);</p>
<p>(in 表示指定了条件范围) </p>
<h5 id="过滤查询-关键字distinct-不能有重复的">过滤查询 关键字 distinct(不能有重复的)</h5>
<p>SELECT DISTINCT name FROM student; (查询 student 表中所有不重复的名字)</p>
<h5 id="limit限制查询">limit 限制查询</h5>
<pre><code class="highlight-chroma">select *from student order by id desc limit 1,2;
(根据id倒序排序，
下标为1开始查，查2条数据)
</code></pre>
<h5 id="查询多个字段">查询多个字段</h5>
<p>SELECT id,name from student ; (从 student 表中查询所有的 id 和 name 列，用逗号作为字
之间的分隔符)</p>
<h5 id="别名--as">别名 as</h5>
<p>select id as sid from student;</p>
<p>(查询出来的视图中 id 字段名变为 sid，只是修改视图中的而不会修改原来表中的 id 名)</p>
<h5 id="连接-concat---分隔符--字段1-字段2->连接 concat(" 分隔符", 字段 1, 字段 2)</h5>
<p>select concat(' ' id,name) from student;</p>
<p>结果比如就是：2018214491=张三</p>
<h4 id="2-模糊查询">2)模糊查询</h4>
<h5 id="like操作符与-通配符">like 操作符与 % 通配符</h5>
<p>% 通配符(匹配任何字符任意长度,但 null 不能匹配)</p>
<p>select name from student where name like "张 %"; </p>
<p>(查询 “张” 开头的所有 student 表中的名字; “%” 是一个通配符，表示任何字符他都匹配
</p>
<p>应用场景:栈内搜索（效率比较低）</p>
<hr>
<h5 id="通配符--">通配符 _</h5>
<p>_只能匹配任意单个字符，不能多个字符</p>
<hr>
<h5 id="正则匹配">正则匹配</h5>
<p>关键字 regexp</p>
<p>select * from student where name regexp "[z,c]c" ;</p>
<p>regexp 表示后面接的是一个正则表达式; </p>
<h4 id="3-排序以及聚合函数">3) 排序以及聚合函数</h4>
<h5 id="排序关键字order-by">排序关键字 order by</h5>
<p>select *from student order bt age; 默认是升序</p>
<p>select *from student order bt age desc; 降序</p>
<h5 id="聚合函数">聚合函数</h5>
<p>count() 计算数量</p>

<p>sum() 计算和 select sum(age) from student; 求 student 表格中的年龄之和</p>

<p>avg() 求平均 select avg(age)from student; 求年龄的平均</p>

<h5 id="分组-group-by">分组 group by</h5>

<p>select count(*)as total ,sex from student group by sex;</p>

<p>(按照 sex 分组，查出男女各自的数量)</p>

<p>group by having</p>

<p>select count(*),sex from student group by sex having sex="女";</p>

<p>按照 sex 分组查询 sex= "女" 的数量；</p>

<p>4) 字符串模糊查询</p>

<h3 id="2-复杂查询">2.复杂查询</h3>

<h4 id="1-内连接">1) 内连接</h4>

<p>内连接查询列出与连接条件匹配的数据行，它使用比较运算符来比较被连接的值。</p>

<hr>

<h5 id="等值连接">等值连接</h5>

```
<pre> <code class="language-go highlight-chroma"> <span class="highlight-line"> <span cla
s="highlight-cl"> <span class="highlight-k">select</span> <span class="highlight-o">*</sp
n> <span class="highlight-nx">from</span> <span class="highlight-nx">stu</span> <span
class="highlight-nx">as</span> <span class="highlight-nx">s</span><span class="highligh
-p">,</span><span class="highlight-nx">mark</span> <span class="highlight-nx">as</sp
n> <span class="highlight-nx">m</span> <span class="highlight-nx">where</span> <span
class="highlight-nx">s</span><span class="highlight-p">.</span><span class="highlight-n
">id</span><span class="highlight-p">=</span><span class="highlight-nx">m</span><span class="highlig
t-p">.</span><span class="highlight-nx">id</span><span class="highlight-nx">id</span><span class="highlig
t-p">.</span></span></pre>
```


<span class="high
ight-err">或者

<span class="high
ight-k">select * fr
m stu as
span class="highlight-nx">s,<span class="highli
ht-nx">mark as m
/span> inner join
on s<span class="
ighlight-p">.id=</
pan>m.<span class="
highlight-nx">id.id.

<span class="high
ight-p">(查询mar
表的id <s
an class="highlight-err">与stu<span class="hi
hlight-err">表的id<span class="highlight-err"
相等的所有的列值)
nner join 连接了
个表

<span class="high
ight-nx">ON条件：是过滤两个链接表笛卡尔积形成的</sp
n> 中间表的 约束条件。</s
an>

<span class="high
ight-err">意思是上方查询的是两个表的所有列，形成了一个笛卡尔积，以此基础上的约束条件是</s
an>s.<span class="

```

highlight-nx">id</span><span class="highlight-p">=</span><span class="highlight-nx">m
/span><span class="highlight-p">.</span><span class="highlight-nx">id</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-nx">where</span><span class="highlight-err">条件: </span><span class="highlight-
i">1</span><span class="highlight-err">在没有</span><span class="highlight-nx">on</sp
n><span class="highlight-err">条件的单表查询中, </span><span class="highlight-err">是
制物理表或者</span><span class="highlight-err">中间记录查询结果的约束</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span clas
="highlight-mf">2.</span><span class="highlight-err">在有</span><span class="highlight
nx">on</span><span class="highlight-err">的时候是两表形成的中间表的查询条件</span>
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-nx">on</span><span class="highlight-err">只进行连接操作, </span><span class="high
ight-nx">where</span><span class="highlight-err">只过滤物理表</span><span class="highl
ght-o">/</span><span class="highlight-err">中间表的记录</span>
</span></span><span class="highlight-line"><span class="highlight-cl"><span class="high
ight-nx">having</span><span class="highlight-err">用在分组中</span><span class="high
ight-nx">group</span><span class="highlight-nx">by</span><span class="highlight-nx"
having</span>
</span></span></code></pre>


```