



链滴

# 数据库多表设计

作者: [chenyuan995](#)

原文链接: <https://ld246.com/article/1587212989747>

来源网站: [链滴](#)

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

<p> </p>

<h2 id="多表之间的关系">多表之间的关系</h2>

<h4 id="分类-">分类: </h4>

<ul>

<li>一对一(了解):

<ul>

<li>如: 人和身份证</li>

<li>分析: 一个人只有一个身份证, 一个身份证只能对应一个人</li>

</ul>

</li>

<li>一对多(多对一):

<ul>

<li>如: 部门和员工</li>

<li>分析: 一个部门有多个员工, 一个员工只能对应一个部门</li>

</ul>

</li>

<li>多对多:

<ul>

<li>如: 学生和课程</li>

<li>分析: 一个学生可以选择很多门课程, 一个课程也可以被很多学生选择</li>

</ul>

</li>

</ul>

<h4 id="实现关系-">实现关系: </h4>

<ul>

<li>一对多(多对一):

<ul>

<li>如: 部门和员工</li>

<li>实现方式: 在多的一方建立外键, 指向一的一方的主键。 </li>

</ul>

</li>

</ul>

<p> </p>

<ul>

<li>多对多:

<ul>

<li>如: 学生和课程</li>

<li>实现方式: 多对多关系实现需要借助第三张中间表。中间表至少包含两个字段, 这两个字段作为三张表的外键, 分别指向两张表的主键</li>

</ul>

</li>

</ul>

<p> </p>

<ul>

<li>一对一(了解):

<ul>

<li>如: 人和身份证</li>

<li>实现方式: 一对一关系实现, 可以在任意一方添加唯一外键指向另一方的主键。 </li>

</ul>

</li>

</ul>

<p>  <a href="https://b3logfile.com/file/2020/04/20200312194125301-c7c4283c.png?imageView2/2/interlace/1/format/jpg" data-bbox="118 88 919 115"> </a> </p>

<ul>

<li>案例</li>

</ul>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> -- 创建旅游线路分类表 tab_category
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> -- cid 旅游线路分
主键, 自动增长
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> -- cname 旅游线
分类名称非空, 唯一, 字符串 100
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> CREATE TABLE tab
category (
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   cid INT PRIMA
Y KEY AUTO_INCREMENT,
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   cname VARCH
R(100) NOT NULL UNIQUE
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> );
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> -- 创建旅游线路表
ab_route
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> /*
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> rid 旅游线路主键
自动增长
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> rname 旅游线路
称非空, 唯一, 字符串 100
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> price 价格
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> rdate 上架时间,
期类型
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> cid 外键, 所属分类
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> */
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> CREATE TABLE tab
route(
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   rid INT PRIMAR
KEY AUTO_INCREMENT,
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   rname VARCH
R(100) NOT NULL UNIQUE,
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   price DOUBLE,
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   rdate DATE,
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   cid INT,
</span> </span> <span class="highlight-line"> <span class="highlight-cl">   FOREIGN KEY (c
id) REFERENCES tab_category(cid)
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> );
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> /*
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> */
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> 创建用户表 tab_us
er
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> uid 用户主键, 自
长
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> username 用户名
度 100, 唯一, 非空
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">password 密码长度
30, 非空
</span></span><span class="highlight-line"><span class="highlight-cl">name 真实姓名长度
100
</span></span><span class="highlight-line"><span class="highlight-cl">birthday 生日
</span></span><span class="highlight-line"><span class="highlight-cl">sex 性别, 定长字
串 1
</span></span><span class="highlight-line"><span class="highlight-cl">telephone 手机号
字符串 11
</span></span><span class="highlight-line"><span class="highlight-cl">email 邮箱, 字符
长度 100
</span></span><span class="highlight-line"><span class="highlight-cl">*/
</span></span><span class="highlight-line"><span class="highlight-cl">CREATE TABLE tab
user (
</span></span><span class="highlight-line"><span class="highlight-cl">  uid INT PRIMA
Y KEY AUTO_INCREMENT,
</span></span><span class="highlight-line"><span class="highlight-cl">  username VAR
CHAR(100) UNIQUE NOT NULL,
</span></span><span class="highlight-line"><span class="highlight-cl">  PASSWORD VA
CHAR(30) NOT NULL,
</span></span><span class="highlight-line"><span class="highlight-cl">  NAME VARCH
R(100),
</span></span><span class="highlight-line"><span class="highlight-cl">  birthday DATE,
</span></span><span class="highlight-line"><span class="highlight-cl">  sex CHAR(1) DE
AULT '男',
</span></span><span class="highlight-line"><span class="highlight-cl">  telephone VAR
CHAR(11),
</span></span><span class="highlight-line"><span class="highlight-cl">  email VARCHAR
100)
</span></span><span class="highlight-line"><span class="highlight-cl">);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">*/
</span></span><span class="highlight-line"><span class="highlight-cl">创建收藏表 tab_fa
orite
</span></span><span class="highlight-line"><span class="highlight-cl">rid 旅游线路 id,
键
</span></span><span class="highlight-line"><span class="highlight-cl">date 收藏时间
</span></span><span class="highlight-line"><span class="highlight-cl">uid 用户 id, 外键
</span></span><span class="highlight-line"><span class="highlight-cl">rid 和 uid 不能重
, 设置复合主键, 同一个用户不能收藏同一个线路两次
</span></span><span class="highlight-line"><span class="highlight-cl">*/
</span></span><span class="highlight-line"><span class="highlight-cl">CREATE TABLE tab
favorite (
</span></span><span class="highlight-line"><span class="highlight-cl">  rid INT, -- 线路i
</span></span><span class="highlight-line"><span class="highlight-cl">  DATE DATETIME
</span></span><span class="highlight-line"><span class="highlight-cl">  uid INT, -- 用户i
</span></span><span class="highlight-line"><span class="highlight-cl">  -- 创建复合主键
PRIMARY KEY(r
d,uid), -- 联合主键
</span></span><span class="highlight-line"><span class="highlight-cl">  FOREIGN KEY (r
d) REFERENCES tab_route(rid),

```

```
</span></span><span class="highlight-line"><span class="highlight-cl"> FOREIGN KEY(u
d) REFERENCES tab_user(uid)
</span></span><span class="highlight-line"><span class="highlight-cl">);
</span></span></code></pre>
<h2 id="数据库设计的范式">数据库设计的范式</h2>
<h4 id="概念-">概念: </h4>
<p>设计数据库时, 需要遵循的一些规范。要遵循后边的范式要求, 必须先遵循前边的所有范式要求<
p>
<p>设计关系数据库时, 遵从不同的规范要求, 设计出合理的关系型数据库, 这些不同的规范要求被
为不同的范式, 各种范式呈递次规范, 越高的范式数据库冗余越小。</p>
<p>目前关系数据库有六种范式: 第一范式 (1NF)、第二范式 (2NF)、第三范式 (3NF)、巴斯-
德范式 (BCNF)、第四范式(4NF) 和第五范式 (5NF, 又称完美范式)。</p>
<h4 id="分类--">分类: </h4>
<ul>
<li>第一范式 (1NF) :
<ul>
<li>每一列都是不可分割的原子数据项</li>
</ul>
</li>
</ul>
<table>
<thead>
<tr>
<th>学号</th>
<th>姓名</th>
<th>班级</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>张三</td>
<td>一年三班</td>
</tr>
<tr>
<td>2</td>
<td>李四</td>
<td>一年二级</td>
</tr>
<tr>
<td>3</td>
<td>王五</td>
<td>二年三班</td>
</tr>
</tbody>
</table>
<ul>
<li>第二范式 (2NF) :
<ul>
<li>在 1NF 的基础上, 非码属性必须完全依赖于码 (在 1NF 基础上消除非主属性对主码的部分函数
赖) </li>
</ul>
</li>
<li>示例: </li>
```

</ul>

<p>1、借书证表: </p>

<table>

<thead>

<tr>

<th>学生证号</th>

<th>学生证名称</th>

<th>学生证办理时间</th>

<th>借书证号</th>

<th>借书证名称</th>

<th>借书证办理时间</th>

</tr>

</thead>

<tbody>

<tr>

<td>2、分成两张表</td>

<td></td>

<td></td>

<td></td>

<td></td>

<td></td>

</tr>

</tbody>

</table>

<table>

<thead>

<tr>

<th>学生证号</th>

<th>学生证名称</th>

<th>学生证办理时间</th>

</tr>

</thead>

<tbody>

<tr>

<td>借书证号</td>

<td>借书证名称</td>

<td>借书证办理时间</td>

</tr>

</tbody>

</table>

<ul>

<li>几个概念:

<ul>

<li>函数依赖: A--&gt;B,如果通过 A 属性(属性组)的值,可以确定唯一 B 属性的值。则称 B 依赖于 A

<ul>

<li>例如: 学号--&gt; 姓名。 (学号, 课程名称) --&gt; 分数</li>

</ul>

</li>

<li>完全函数依赖: A--&gt;B, 如果 A 是一个属性组, 则 B 属性值得确定需要依赖于 A 属性组中有的属性值。

<ul>

<li>例如: (学号, 课程名称) --&gt; 分数</li>

</ul>

</li>

<li>部分函数依赖: A--&gt;B, 如果 A 是一个属性组, 则 B 属性值得确定只需要依赖于 A 属性组某一些值即可。

<ul>

<li>例如: (学号, 课程名称) -- &gt; 姓名</li>

</ul>

</li>

<li>传递函数依赖: A--&gt;B, B -- &gt;C . 如果通过 A 属性(属性组)的值, 可以确定唯一 B 属性的, 在通过 B 属性 (属性组) 的值可以确定唯一 C 属性的值, 则称 C 传递函数依赖于 A

<ul>

<li>例如: 学号--&gt; 系名, 系名--&gt; 系主任</li>

</ul>

</li>

<li>码: 如果在一张表中, 一个属性或属性组, 被其他所有属性所完全依赖, 则称这个属性(属性组)该表的码

<ul>

<li>例如: 该表中码为: (学号, 课程名称) </li>

<li>主属性: 码属性组中的所有属性</li>

<li>非主属性: 除过码属性组的属性</li>

</ul>

</li>

</ul>

</li>

<li>第三范式 (3NF) :

<ul>

<li>在 2NF 基础上, 任何非主属性不依赖于其它非主属性 (在 2NF 基础上消除传递依赖) </li>

</ul>

</li>

<li>示例: 学生信息表</li>

</ul>

<table>

<thead>

<tr>

<th>学号</th>

<th>姓名</th>

<th>年龄</th>

<th>所在学院</th>

<th>学院地点</th>

</tr>

</thead>

</table>

<ul>

<li>

<p>存在传递的决定关系: <br>

学号 -&gt; 所在学院 -&gt; 学院地点</p>

</li>

<li>

<p>拆分成两张表</p>

</li>

</ul>

<table>

<thead>

<tr>

<th>学号</th>

<th>姓名</th>

<th>年龄</th>  
<th>所在学院的编号(外键)</th>  
</tr>  
</thead>  
</table>

<table>  
<thead>  
<tr>  
<th>学院编号</th>  
<th>所在学院</th>  
<th>学院地点</th>  
</tr>  
</thead>  
</table>

## 拓展--数据库的备份和还原">拓展——数据库的备份和还原</h2>

<ul>

<li>命令行:

<ul>

<li>语法:

<ul>

<li>备份: mysqldump -u 用户名 -p 密码 数据库名称 &gt; 保存的路径</li>

<li>还原:

<ul>

<li>登录数据库</li>

<li>创建数据库</li>

<li>使用数据库</li>

<li>执行文件。source 文件路径</li>

</ul>

</li>

</ul>

</li>

</ul>

</li>

<li>图形化工具</li>

</ul>