

## 10 | 常用的SQL标准有哪些，在SQL92中是如何使用连接的？

今天我主要讲解连接表的操作。在讲解之前，我想先给你介绍下连接（JOIN）在 SQL 中的重要性。

我们知道 SQL 的英文全称叫做 Structured Query Language，它有一个很强大的功能，就是能在各个数据表之间进行连接查询（Query）。这是因为 SQL 是建立在关系型数据库基础上的一种语言。关系型数据库的典型数据结构就是数据表，这些数据表的组成都是结构化的（Structured）。你可以把关系模型理解成一个二维表格模型，这个二维表格是由行（row）和列（column）组成的。每一个行（row）就是一条数据，每一列（column）就是数据在某一维度的属性。

正是因为在数据库中，表的组成是基于关系模型的，所以一个表就是一个关系。一个数据库中可以包括多个表，也就是存在多种数据之间的关系。而我们之所以能使用 SQL 语言对各个数据表进行复杂查询，核心就在于连接，它可以用一条 SELECT 语句在多张表之间进行查询。你也可以理解为，关系型数据库的核心之一就是连接。

既然连接在 SQL 中这么重要，那么针对今天的内容，需要你从以下几个方面进行掌握：

1. SQL 实际上存在不同的标准，不同标准下的连接定义也有不同。你首先需要了解常用的 SQL 标准有哪些；
2. 了解了 SQL 的标准之后，我们从 SQL92 标准入门，来看下连接表的种类有哪些；
3. 针对一个实际的数据库表，如果你想要做数据统计，需要学会使用跨表的连接进行操作。

### 常用的 SQL 标准有哪些

在正式开始讲连接表的种类时，我们首先需要知道 SQL 存在不同版本的标准规范，因为不同规范下的表连接操作是有区别的。

SQL 有两个主要的标准，分别是 SQL92 和 SQL99。92 和 99 代表了标准提出的时间，SQL92 就是 92 年提出的标准规范。当然除了 SQL92 和 SQL99 以外，还存在 SQL-86、SQL-89、SQL:2003、SQL:2008、SQL:2011 和 SQL:2016 等其他的标准。

这么多标准，到底该学习哪个呢？实际上最重要的 SQL 标准就是 SQL92 和 SQL99。一般来说 SQL92 的形式更简单，但是写的 SQL 语句会比较长，可读性较差。而 SQL99 相比于 SQL92 来说，语法更加复杂，但可读性更强。我们从这两个标准发布的页数也能看出，SQL92 的标准有 500 页，而 SQL99 标准超过了 1000 页。实际上你不用担心要学习这么多内容，基本上从 SQL99 之后，很少有人能掌握所有内容，因为确实太多了。就好比我们使用 Windows、Linux 和 Office 的时候，很少有人能掌握全部内容一样。我们只需要掌握一些核心的功能，满足日常工作的需求即可。

## 在 SQL92 中是如何使用连接的

相比于 SQL99，SQL92 规则更简单，更适合入门。在这篇文章中，我会先讲 SQL92 是如何对连接表进行操作的，下一篇文章再讲 SQL99，到时候你可以对比下这两者之间有什么区别。

在进行连接之前，我们需要用数据表做举例。这里我创建了 NBA 球员和球队两张表，SQL 文件你可以从[GitHub](#)上下载。

其中 player 表为球员表，一共有 37 个球员，如下所示：

player_id	team_id	player_name	height
10001	1001	韦恩·艾灵顿	1.93
10002	1001	雷吉·杰克逊	1.91
10003	1001	安德烈·德拉蒙德	2.11
10004	1001	索恩·马克	2.16
.....	.....	.....	.....
10037	1002	伊凯·阿尼博古	2.08

team 表为球队表，一共有 3 支球队，如下所示：

team_id	team_name
1001	底特律活塞
1002	印第安纳步行者
1003	亚特兰大老鹰

有了这两个数据表之后，我们再来看下 SQL92 中的 5 种连接方式，它们分别是笛卡尔积、等值连接、非等值连接、外连接（左连接、右连接）和自连接。

## 笛卡尔积

笛卡尔乘积是一个数学运算。假设我有两个集合 X 和 Y，那么 X 和 Y 的笛卡尔积就是 X 和 Y 的所有可能组合，也就是第一个对象来自于 X，第二个对象来自于 Y 的所有可能。

我们假定 player 表的数据是集合 X，先进行 SQL 查询：

 复制代码

```
1 SELECT * FROM player
```

再假定 team 表的数据为集合 Y，同样需要进行 SQL 查询：

 复制代码

```
1 SELECT * FROM team
```

你会看到运行结果会显示出上面的两张表格。

接着我们再来看下两张表的笛卡尔积的结果，这是笛卡尔积的调用方式：

 复制代码

```
1 SQL: SELECT * FROM player, team
```

运行结果（一共  $37 \times 3 = 111$  条记录）：

player_id	team_id	player_name	height	team_id(1)	team_name
10001	1001	韦恩·艾灵顿	1.93	1001	底特律活塞
10001	1001	韦恩·艾灵顿	1.93	1002	印第安纳步行者
10001	1001	韦恩·艾灵顿	1.93	1003	亚特兰大老鹰
.....	.....	.....	.....	.....	.....
10037	1002	伊凯·阿尼博古	2.08	1003	亚特兰大老鹰

笛卡尔积也称为交叉连接，英文是 CROSS JOIN，它的作用就是可以把任意表进行连接，即使这两张表不相关。但我们通常进行连接还是需要筛选的，因此你需要在连接后面加上 WHERE 子句，也就是作为过滤条件对连接数据进行筛选。比如后面要讲到的等值连接。

## 等值连接

两张表的等值连接就是用两张表中都存在的列进行连接。我们也可以对多张表进行等值连接。

针对 player 表和 team 表都存在 team\_id 这一列，我们可以用等值连接进行查询。

 复制代码

```
1 SQL: SELECT player_id, player.team_id, player_name, height, team_name FROM player, team
```

运行结果（一共 37 条记录）：

player_id	team_id	player_name	height	team_name
10001	1001	韦恩·艾灵顿	1.93	底特律活塞
10002	1001	雷吉·杰克逊	1.91	底特律活塞
10003	1001	安德烈·德拉蒙德	2.11	底特律活塞
.....	.....	.....	.....	.....
10037	1002	Ike Anigbogu	2.08	印第安纳步行者

我们在进行等值连接的时候，可以使用表的别名，这样会让 SQL 语句更简洁：

[复制代码](#)

```
1 SELECT player_id, a.team_id, player_name, height, team_name FROM player AS a, team AS b
```

需要注意的是，如果我们使用了表的别名，在查询字段中就只能使用别名进行代替，不能使用原有的表名，比如下面的 SQL 查询就会报错：

[复制代码](#)

```
1 SELECT player_id, player.team_id, player_name, height, team_name FROM player AS a, team
```

## 非等值连接

当我们进行多表查询的时候，如果连接多个表的条件是等号时，就是等值连接，其他的运算符连接就是非等值查询。

这里我创建一个身高级别表 `height_grades`，如下所示：

height_level	height_lowest	height_highest
A	2.00	2.50
B	1.90	1.99
C	1.80	1.89
D	1.60	1.79

我们知道 `player` 表中有身高 `height` 字段，如果想要知道每个球员的身高的级别，可以采用非等值连接查询。

[复制代码](#)

```
1 SQL: SELECT p.player_name, p.height, h.height_level  
2 FROM player AS p, height_grades AS h  
3 WHERE p.height BETWEEN h.height_lowest AND h.height_highest
```

运行结果（37 条记录）：

player_name	height	height_level
韦恩·艾灵顿	1.93	B
雷吉·杰克逊	1.91	B
安德烈·德拉蒙德	2.11	A
.....	.....	.....
Ike Anigbogu	2.08	A

## 外连接

除了查询满足条件的记录以外，外连接还可以查询某一方不满足条件的记录。两张表的外连接，会有一张是主表，另一张是从表。如果是多张表的外连接，那么第一张表是主表，即显示全部的行，而第剩下的表则显示对应连接的信息。在 SQL92 中采用 (+) 代表从表所在的位置，而且在 SQL92 中，只有左外连接和右外连接，没有全外连接。

什么是左外连接，什么是右外连接呢？

左外连接，就是指左边的表是主表，需要显示左边表的全部行，而右侧的表是从表， (+) 表示哪个是从表。

 复制代码

```
1 SQL: SELECT * FROM player, team where player.team_id = team.team_id(+) 
```

相当于 SQL99 中的：

 复制代码

```
1 SQL: SELECT * FROM player LEFT JOIN team on player.team_id = team.team_id 
```

右外连接，指的就是右边的表是主表，需要显示右边表的全部行，而左侧的表是从表。

 复制代码

```
1 SQL: SELECT * FROM player, team where player.team_id(+) = team.team_id 
```

相当于 SQL99 中的：

 复制代码

```
1 SQL: SELECT * FROM player RIGHT JOIN team on player.team_id = team.team_id
```

需要注意的是，LEFT JOIN 和 RIGHT JOIN 只存在于 SQL99 及以后的标准中，在 SQL92 中不存在，只能用 (+) 表示。

## 自连接

自连接可以对多个表进行操作，也可以对同一个表进行操作。也就是说查询条件使用了当前表的字段。

比如我们想要查看比布雷克·格里芬高的球员都有谁，以及他们的对应身高：

 复制代码

```
1 SQL: SELECT b.player_name, b.height FROM player as a , player as b WHERE a.player_name <
```

运行结果（6 条记录）：

player_name	height
安德烈-德拉蒙德	2.11
索恩-马克	2.16
扎扎-帕楚里亚	2.11
亨利-埃伦森	2.11
多曼塔斯-萨博尼斯	2.11
迈尔斯-特纳	2.11

如果不用自连接的话，需要采用两次 SQL 查询。首先需要查询布雷克·格里芬的身高。

 复制代码

```
1 SQL: SELECT height FROM player WHERE player_name = '布雷克 - 格里芬'
```

运行结果为 2.08。

然后再查询比 2.08 高的球员都有谁，以及他们的对应身高：

 复制代码

```
1 SQL: SELECT player_name, height FROM player WHERE height > 2.08
```

运行结果和采用自连接的运行结果是一致的。

## 总结

今天我讲解了常用的 SQL 标准以及 SQL92 中的连接操作。SQL92 和 SQL99 是经典的 SQL 标准，也分别叫做 SQL-2 和 SQL-3 标准。也正是在这两个标准发布之后，SQL 影响力越来越大，甚至超越了数据库领域。现如今 SQL 已经不仅仅是数据库领域的主流语言，还是信息领域中信息处理的主流语言。在图形检索、图像检索以及语音检索中都能看到 SQL 语言的使用。

除此以外，我们使用的主流 RDBMS，比如 MySQL、Oracle、SQL Sever、DB2、PostgreSQL 等都支持 SQL 语言，也就是说它们的使用符合大部分 SQL 标准，但很难完全符合，因为这些数据库管理系统都在 SQL 语言的基础上，根据自身产品的特点进行了扩充。即使这样，SQL 语言也是目前所有语言中半衰期最长的，在 1992 年，Windows3.1 发布，SQL92 标准也同时发布，如今我们早已不使用 Windows3.1 操作系统，而 SQL92 标准却一直持续至今。

当然我们也要注意到 SQL 标准的变化，以及不同数据库管理系统使用时的差别，比如 Oracle 对 SQL92 支持较好，而 MySQL 则不支持 SQL92 的外连接。



我今天讲解了 SQL 的连接操作，你能说说内连接、外连接和自连接指的是什么吗？另外，你不妨拿案例中的 team 表做一道动手题，表格中一共有 3 支球队，现在这 3 支球队需要进行比赛，请用一条 SQL 语句显示出所有可能的比赛组合。