

多表查询

读者的借阅信息，保存在借阅表 borrow 中。为了数据完整性，在设计数据库表的时候，该表保存的是读者编号、书籍编号等。对于用户而言，这些编号毫无意义。实际应用时，需要知道读者的借阅信息，例如读者的姓名，借阅的书籍名称等。这些数据分别保存在书籍表 book、读者表 reader 和借阅表 borrow 中，此时需要将相关的表连接起来进行查询。

如果知道一个读者的姓名，想知道该读者借阅了什么书。读者的书籍保存在读者表 reader 里；书籍表 book 里保存的书籍的书籍；借阅的记录保存在借阅表 borrow 里。例如，知道一个读者的姓名，例如“崔同贺”读者，需要把三个表连接起来，用读者表 reader 里的 readerNo 和借阅表 borrow 里的 readerNo、用借阅表 borrow 里的 bookNo 和书籍表 book 里的 bookNo 连接起来，形成一个“查询链”。

表和表之间连接有多种形式，一般有全连接、内连接和外连接。

全连接使之将每个表的每行都与其它表中的每行交叉，以产生所有可能的组合，列包含了所有表中出现的列，也就是笛卡尔积。

表 A 中的数据如下：

T1	T2
1	A
3	B
5	C

表 B 中的数据如下：

T3	T4	T5
1	X	M
5	Y	N

表 A 和 B 全连接后得到如下 6 行（3x2=6）的表：

T1	T2	T3	T4	T5
1	A	1	X	M
3	B	1	X	M
5	C	1	X	M
1	A	5	Y	N
3	B	5	Y	N
5	C	5	Y	N

全连接产生的结果大多数情况没有意义。通常需要设定连接条件来聚焦结果集合，这样的连接即为内连接。如果设定的条件是等值条件，也叫等值连接。

外连接包括左外连接和右外连接。左外连接，结果集中除了匹配行外，还包括左表有但右表中不匹配的行，对于这样的行，从右表被选择的列设置为 NULL。右外连接，与左外连接相反。

例如，上面的表 A 的 T1 列左外连接表 B 的 T3 列，将得到如下结果：

T1	T2	T3	T4	T5
1	A	1	X	M
3	B	NULL	NULL	NULL
5	C	5	Y	N

2.4.1 内连接查询

【例 2-19】查看“崔同贺”读者所有的借阅记录，要求显示借阅的书籍名称和借阅状态。在 MySQL 命令行窗口中执行如下 SQL 语句：

```
SELECT bookName, re_new, re_turn
```

```
FROM reader INNER JOIN borrow ON reader.readerNo=borrow.readerNo
      INNER JOIN book ON borrow.bookNo=book.bookNo
WHERE readerName='崔同贺';
```

内连接是在 FROM 子句产生的中间结果中应用 ON 条件后得到的结果。这里内连接 ON reader.readerNo=borrow.readerNo 和 borrow.bookNo=book.bookNo 是等值比较。内连接是系统默认的,可以省略 INNER 关键字。上述内连接语句是使用 ANSI 连接语法的 SELECT 语句。

连接条件还可以写在 WHERE 子句后面,如下:

```
SELECT bookName, re_new, re_turn
FROM reader, borrow, book
WHERE reader.readerNo=borrow.readerNo AND borrow.bookNo=book.bookNo
      AND readerName='崔同贺';
```

reader、borrow 和 book 表的连接条件是 “reader.readerNo=borrow.readerNo AND borrow.bookNo=book.bookNo”。

2.4.2 使用别名

别名主要针对的是表,给表起个别名。例如,在【例 2-19】中,如果把读者表 reader 起个别名 R,借阅表 borrow 起个别名为 B,书籍表 book 起个别名 K,则语句为:

```
SELECT bookName, re_new, re_turn
FROM reader AS R INNER JOIN borrow AS B ON R.readerNo=B.readerNo
      INNER JOIN book AS K ON B.bookNo=K.bookNo
WHERE readerName='崔同贺';
```

或者连接条件写在 WHERE 子句后面,如下:

```
SELECT bookName, re_new, re_turn
FROM reader R, borrow B, book K
WHERE R.readerNo=B.readerNo AND B.bookNo=K.bookNo
      AND readerName='崔同贺';
```

作为特例,可以将一根表与它自身进行连接,成为自连接。例如,若要在根表中查找具有相同列值的行,则可以使用自连接。使用自连接时,必须为该表指定两个别名。

【例 2-20】查询书籍表 book 中作者相同但出版社不同的书籍信息,要求显示书籍名、作者和出版社。在 MySQL 命令行窗口中执行如下 SQL 语句:

```
SELECT A.bookName, A.author, A.publishName
FROM book AS a JOIN book AS b ON A.author=B.author
      AND A.publishName<>B.publishName;
```

连接条件写在 WHERE 子句后面,则为:

```
SELECT A.bookName, A.author, A.publishName
FROM book AS a, book AS b
WHERE A.author=B.author AND A.publishName<>B.publishName;
```

2.4.3 左外连接与右外连接

【例 2-21】查找图书借阅信息,要求显示书籍名称和对应的读者号,没有被借阅的书也显示出来。在 MySQL 命令行窗口中执行如下 SQL 语句:

```
SELECT bookName, readerNo
FROM book LEFT OUTER join borrow
```

```
ON book.bookNo=borrow.bookNo
```

上述语句运行后，先显示借阅信息，在这之后，还把没有被借阅过的书籍名称显示出来。

使用右外连接，显示借阅信息，要求显示读者号和书籍号，没有借阅记录的读者号也显示出来。

```
SELECT reader.readerNo, borrow.bookNo
FROM borrow RIGHT OUTER JOIN reader ON reader.readerNo=borrow.readerNo;
```

2.4.4 组合查询

【例 2-22】显示借阅信息，要求显示未还的读者姓名、书籍名称，要求按应该还书的时间顺序排列。在 MySQL 命令行窗口中执行如下 SQL 语句：

```
SELECT reader.readerName, bookName
FROM reader JOIN borrow ON reader.readerNo=borrow.readerNo
      JOIN book ON borrow.bookNo=book.bookNo
WHERE re_turn='否'
ORDER BY returnDate;
连接条件写在 WHERE 子句后面，则为：
SELECT reader.readerName, bookName
FROM reader, borrow, book
WHERE re_turn='否'
      AND reader.readerNo=borrow.readerNo AND borrow.bookNo=book.bookNo
ORDER BY returnDate;
```

显示从未借过书的读者信息，要求显示读者号和姓名。

```
SELECT readerNo, readerName
FROM reader
WHERE NOT EXISTS (SELECT readerNo FROM borrow WHERE readerno=reader.readerNo);
```

2.4.5 使用 UNION 合并查询结果

UNION 子句可以将若干个查询结果合并。

【例 2-23】先列表显示书籍名称，之后再显示读者姓名。在 MySQL 命令行窗口中执行如下 SQL 语句：

```
SELECT bookName
FROM book
UNION
SELECT readerName
FROM reader;
```