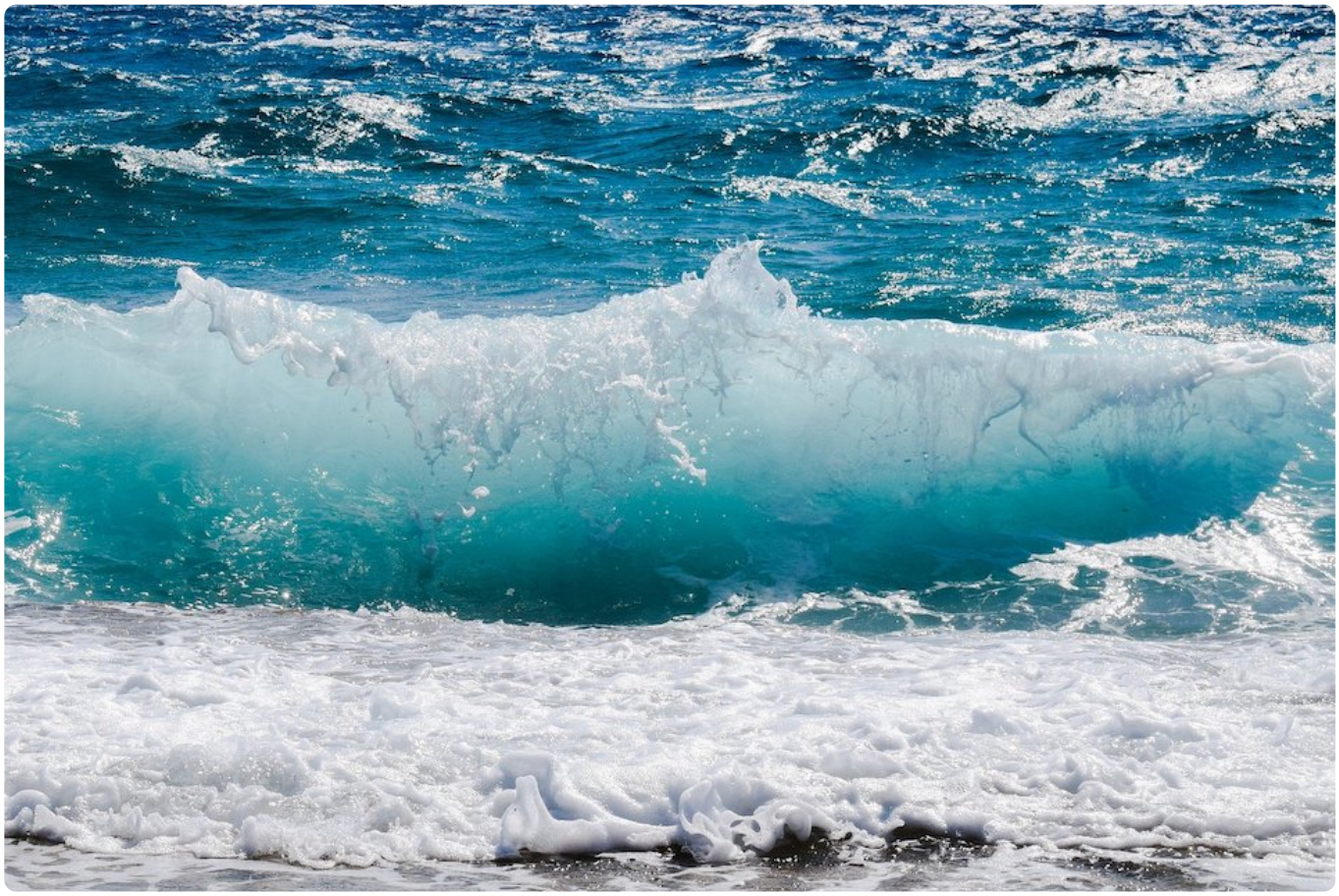
18 | 权限管理：如何控制数据库访问，消除安全隐患？

2021-04-20 朱晓峰

MySQL 必知必会 进入课程 >



讲述：朱晓峰
时长 08:49 大小 8.09M

你好，我是朱晓峰，今天，我来和你聊一聊权限管理。

我们在开发应用的时候，经常会遇到一种需求，就是要根据用户的不同，对数据进行横向和纵向的分组。

所谓横向的分组，就是指用户可以接触到的数据的范围，比如可以看到哪些表的数据；所谓纵向的分组，就是指用户对接触到的数据能访问到什么程度，比如能看、能改，甚至是删除。

我们把具有相同数据访问范围和程度的用户归成不同的类别，这种类别就叫角色。通过角色，管理用户对数据库访问的范围和程度就更加方便了。这也就是对用户的数据访问权限的管理。

恰当的权限设定，可以确保数据的安全性，这是至关重要的。

那么，怎么进行权限管理呢？这节课，我就结合超市项目的实际案例，给你讲一下权限管理的具体操作，包括怎么操作角色和用户，怎么通过角色给用户授权，怎么直接给用户授权，从而帮助你管理好用户的权限，提升数据库的安全性。

下面我就来先讲讲角色。我们可以通过角色对相同权限的用户进行分组管理，这样可以使权限管理更加简单高效。

角色的作用

角色是在 MySQL 8.0 中引入的新功能，相当于一个权限的集合。引入角色的目的是方便管理拥有相同权限的用户。

下面我举个超市项目中的例子，来具体说明一下如何通过角色管理用户权限。

超市项目中有库管、营运和财务等不同的模块，它们各自对应不同的数据表。比如库存模块中的盘点表（demo.invcount）、营运模块中的商品信息表（demo.goodsmaster），还有财务模块中的应付账款表（demo.settlement）。下面是这些表的具体信息。

盘点表：

listnumber（盘点单号）	stockid（仓库编号）	itemnumber（商品编号）	accquant（结存数量）	invquant（盘存数量）	plquant（盈亏数量）
1234	1	1	10	5	-5

商品信息表：

itemnumber（商品编号）	barcode（条码）	goodsname（名称）	unit（单位）	salesprice（售价）
1	0001	书	本	89

应付账款表：

listnumber（单号）	supplierid（供货商编号）	topay（应付金额）	paid（已付金额）	balance（余额）	recordingdate（记录日期）
4578	1	1000	900	100	2020-12-02

在超市项目中，员工的职责不同，包括库管、营运和财务等，不同的职责有不同的数据访问权限。比如：

张三是库管，他就可以查询商品信息表，对盘点表有增删改查的权限，但无权访问应付账款表；

李四是营运，他就拥有对商品信息表有增删改查的权限，而对库存表和应付账款表，只有查看的权限；

王五是财务，他就有对应付账款表有增删改查的权限，对商品信息表和库存表，只有查看的权限。

所以，我们需要为每一个职责创建一个对应的角色，为每个员工创建一个对应的数据库用户。然后通过给角色赋予相关的权限，再把角色赋予用户，实现对超市员工访问数据权限的管理，从而保证数据的安全性。

这样说有点抽象，下面我们具体操作一下角色和用户。

如何操作角色？

首先，我们要创建一个角色，为后面的授权做好准备。

如何创建角色？

MySQL 中的角色名称由角色名称加主机名称组成。创建角色的语法结构如下：

```
CREATE ROLE 角色名;
```

复制代码

假设我们现在需要创建一个经理的角色，就可以用下面的代码：

```
mysql> CREATE ROLE 'manager'@'localhost';
Query OK, 0 rows affected (0.06 sec)
```

复制代码

这里的意思是，创建一个角色，角色名称是“manager”，角色可以登录的主机是“localhost”，意思是只能从数据库服务器运行的这台计算机登录这个账号。你也可以不写主机名，直接创建角色“manager”：

```
mysql> CREATE ROLE 'manager';
Query OK, 0 rows affected (0.01 sec)
```

复制代码

如果不写主机名，MySQL 默认是通配符“%”，意思是这个账号可以从任何一台主机上登录数据库。

同样道理，如果我们要创建库管的角色，就可以用下面的代码：

```
mysql> CREATE ROLE 'stocker';
Query OK, 0 rows affected (0.02 sec)
```

复制代码

创建角色之后，默认这个角色是没有任何权限的，我们需要给角色授权。

怎么给角色赋予权限?

给角色授权的语法结构是：

```
GRANT 权限 ON 表名 TO 角色名;
```

假设我们现在想给经理角色授予商品信息表、盘点表和应付账款表的只读权限，就可以用下面的代码来实现：

```
mysql> GRANT SELECT ON demo.settlement TO 'manager';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT SELECT ON demo.goodsmaster TO 'manager';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT SELECT ON demo.invcount TO 'manager';
Query OK, 0 rows affected (0.01 sec)
```

如果我们需要赋予库管角色盘点表的增删改查权限、商品信息表的只读权限，对应付账款表没有权限，就可以这样：

```
mysql> GRANT SELECT,INSERT,DELETE,UPDATE ON demo.invcount TO 'stocker';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT SELECT ON demo.goodsmaster TO 'stocker';
Query OK, 0 rows affected (0.02 sec)
```

查看角色权限

赋予角色权限之后，我们可以通过 SHOW GRANTS 语句，来查看权限是否创建成功了：

```
mysql> SHOW GRANTS FOR 'manager';
```

```
+-----------------------------------------------------+
| Grants for manager@% |
+-----------------------------------------------------+
| GRANT USAGE ON *.* TO `manager`@`%` |
| GRANT SELECT ON `demo`.`goodsmaster` TO `manager`@`%` |
| GRANT SELECT ON `demo`.`invcount` TO `manager`@`%` |
| GRANT SELECT ON `demo`.`settlement` TO `manager`@`%` |
+-----------------------------------------------------+
4 rows in set (0.00 sec)
```

只要你创建了一个角色，系统就会自动给你一个“USAGE”权限，意思是连接登录数据库的权限。代码的最后三行代表了我们给角色“manager”赋予的权限，也就是对商品信息表、盘点表和应付账款表的只读权限。

再来看看库管角色的权限:

复制代码

```
mysql> SHOW GRANTS FOR 'stocker';
+-----------------------------------------------------------------------------+
| Grants for stocker@% |
+-----------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `stocker`@`%` |
| GRANT SELECT ON `demo`.`goodsmaster` TO `stocker`@`%` |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `demo`.`invcount` TO `stocker`@`%` |
+-----------------------------------------------------------------------------+
3 rows in set (0.00 sec)
```

结果显示，库管角色拥有商品信息表的只读权限和盘点表的增删改查权限。

当我们需要对业务重新整合的时候，可能就需要对之前创建的角色进行清理，删除一些不会再使用的角色。

删除角色

删除角色的操作很简单，你只要掌握语法结构就行了。

复制代码

```
DROP ROLE 角色名称;
```

这个操作十分简单，我就不多说了。

到这里，关于角色的操作，我就介绍完了，下面我们来看看关于用户的操作。

如何操作用户？

我们可以把数据库中的角色看作是一个权限集。角色本身不能操作数据，这个任务还是要靠用户来完成。在操作用户前，我们先要创建一个用户。

创建用户

创建用户的语法结构是这样的：

复制代码

```
CREATE USER 用户名 [IDENTIFIED BY 密码];
```

“[ ]”表示可选，也就是说，可以指定用户登录时需要密码验证，也可以不指定密码验证，这样用户可以直接登录。不过，不指定密码的方式不安全，不推荐使用。

举个例子，假设我们要给张三创建一个用户，用户名是“zhangsan”，密码是“mysql”，可以通过下面的代码来实现：

复制代码

```
mysql> CREATE USER 'zhangsan' IDENTIFIED BY 'mysql';
Query OK, 0 rows affected (0.02 sec)
```

这样，张三的用户就创建成功了。

给用户授权

给用户授权的方式有 2 种，分别是通过把角色赋予用户给用户授权，和直接给用户授权。

通过把角色赋予用户给用户授权的语法结构如下：

复制代码

```
1
```

举个小例子，我们想要给张三赋予库管的角色，可以通过下面的代码实现：

复制代码

```
1 mysql> GRANT 'stocker' TO 'zhangsan';
2 Query OK, 0 rows affected (0.01 sec)
```

我们也可以直接给用户授权，语法结构如下：

复制代码

```
1 GRANT 权限 ON 表名 TO 用户名;
```

这种方式简单直接，我就不多说了。下面我们来查看一下这个用户的权限有哪些。

查看用户权限

查看用户权限的语法结构是：

复制代码

```
1 SHOW GRANTS FOR 用户名;
```

我们可以通过下面的代码来查看张三的权限：

复制代码

```
1 mysql> SHOW GRANTS FOR 'zhangsan';
2 +-------------------------------------+
3 | Grants for zhangsan@% |
4 +-------------------------------------+
5 | GRANT USAGE ON *.* TO `zhangsan`@`%` |
6 | GRANT `stocker`@`%` TO `zhangsan`@`%` |
7 +-------------------------------------+
8 2 rows in set (0.00 sec)
```

结果显示，张三拥有库管角色的权限。

说到这里，我必须要提醒你一个常见的坑。

如果现在你用张三的这个用户去登录，你会发现，这个账号是没有任何权限的。你是不是觉得很奇怪，我不是把角色“stocker”赋予用户“zhangsan”了吗？那用户“zhangsan”应该有角色“stocker”的权限啊。其实，这是因为，**MySQL 中创建了角色之后，默认都是没有被激活的**，也就是不能用，必须要用下面的语句激活：

复制代码

```
SET global activate_all_roles_on_login=ON;
```

这条 SQL 语句的意思是，对所有角色永久激活。运行这条语句之后，用户“zhangsan”才真正拥有了角色“stocker”的所有权限。

下面我们就用张三的账号登录，确认一下他有没有相应的权限：

复制代码

```
H:\>mysql -u zhangsan -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 8.0.23 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from demo.goodsmaster;
+------------+---------+-----------+---------------+------+------------+------
| itemnumber | barcode | goodsname | specification | unit | salesprice | avgim
+------------+---------+-----------+---------------+------+------------+------
| 1 | 0001 | 书 | 16开 | 本 | 89.00 | 31.00 |
| 2 | 0002 | 笔 | NULL | 包 | 5.00 | 2.87 |
+------------+---------+-----------+---------------+------+------------+------
2 rows in set (0.02 sec)
```

结果显示，我们可以正常登录，并且可以查询商品信息表的内容。

删除用户

当用户不再使用的时候，我们也可以删除用户。操作起来很简单，你只要知道语法结构就行了：

```
1 DROP USER 用户名;
```

复制代码

总结

今天这节课，我们学习了权限管理的方法，包括如何操作角色和用户，如何通过角色给用户授权，如何直接给用户授权等。

角色是权限的集合。你可以直接给用户授予访问数据库的权限，也可以通过把角色授予用户，从而把角色对数据库的访问权限全部授予给用户。而用户是数据库的使用者，我们可以通过给用户授予访问数据库中资源的权限，来控制使用者对数据库的访问，消除安全隐患。

需要注意的是，角色在刚刚创建出来的时候，默认是没有激活的，需要手动激活，才可以使用。如果你把角色赋予了用户，那么用户就拥有了角色的全部权限。但是，如果你删除了角色，那么用户也就失去了通过这个角色所获得的所有权限。

我知道，有一些程序员喜欢使用 Root 超级用户来访问数据库，完全把权限控制放在应用层面实现。这样当然也是可以的。不过我建议你，尽量使用数据库自己的角色和用户机制来控制访问权限，不要轻易用 Root 账号。因为 Root 账号密码放在代码里面不安全，一旦泄露，数据库就会完全失去保护。而且，MySQL 的权限控制功能十分完善，应该尽量利用，可以提高效率，而且安全可靠。

思考题

在今天的课里，我举了一个例子，提到超市运营中的一个职责“财务”。财务可以对商品信息表、盘点表有只读的权限，对应付账款表有增删改查的权限。请你设计一个角色：财务“accountant”具备这些权限。给会计“李四”创建一个用户账号“lisi”，使李四通过财务的角色获得对应付账款表增删改查的权限，和对商品信息表、盘点表有只读的权限。

欢迎在留言区写下你的思考和答案，我们一起交流讨论。如果你觉得今天的内容对你有所帮助，也欢迎你分享给你的朋友或同事，我们下节课见。

提建议

更多课程推荐

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 17 | 触发器：如何让数据修改自动触发关联操作，确保数据一致性？

下一篇 19 | 日志（上）：系统出现问题，如何及时发现？

精选留言 (4)

写留言

floating
2021-04-21

```
-- 创建财务角色
CREATE ROLE 'accountant' ;

-- 授予财务角色对商品信息表、盘点表有只读的权限
GRANT SELECT ON demo.goodsmaster TO 'accountant';...
```

展开

Harry
2021-04-21

有一点要注意，在使用 Windows + SQL Server 技术栈时，不要把操作系统的用户等同于数据库的用户。

展开

Harry
2021-04-20

工作中负责的一个项目，对数据库的访问控制非常严格，屡受其害。
MySQL 的权限控制感觉非常灵活，下个版本安排上。

展开

lesserror
2021-04-20

既然老师指出了这个「坑」，那我必须要记录下来了。

权限这块儿的知识点，之前只是了解，实际开发中没太把权限的控制放在数据库用户上面去做。今天这节，补足了我这块之前模糊的理解。

...

展开