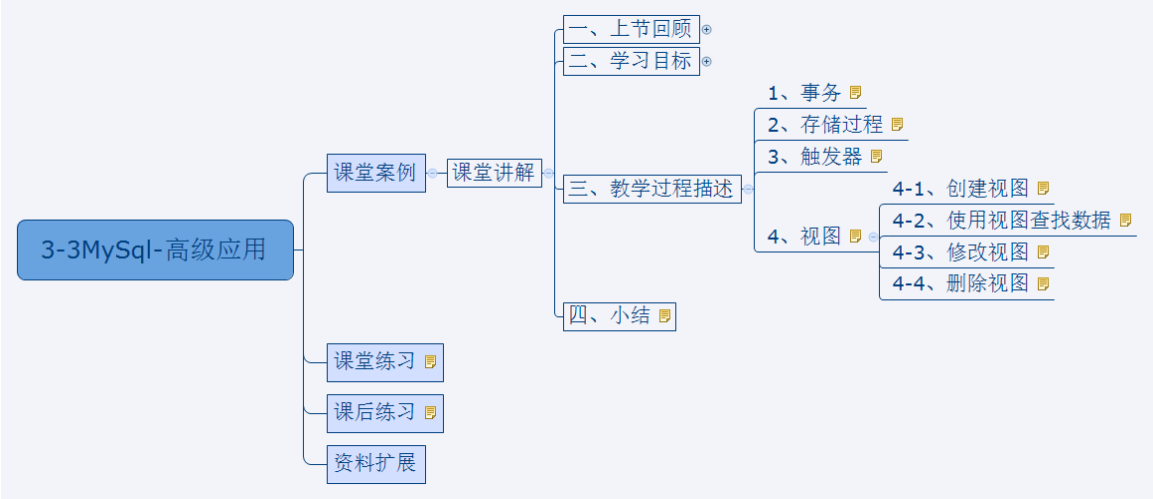
3-3MySql-高级应用



1. 课堂案例

课堂讲解

一、上节回顾

MySql-安全与引擎

二、学习目标

了解mysql高级知识

三、教学过程描述

1、事务

事务就是一组原子性的SQL查询，或者说一个独立的单元。如果数据库引擎能够成功地对数据库应用该组查询的全部语句，那么就执行该组查询。如果其中有任何一条语句因为崩溃或者其他原因无法执行，那么所有原句都不会执行。也就是说，事务内的语句，要么全部执行成功，要么全部执行失败。

典型例子：

假设一个银行的数据库有两张表：支票(checking)和储蓄表(savings)。现在要从用户Jane（id123）的支票账户转移200元到她的储蓄账户，那么需要至少三个步骤：

1.检查支票账户的余额高于200元

2.从支票账户余额减去200元

3.在储蓄账户余额增加200元

上述三个步骤的操作必须打包在一个事务中，任何一个步骤失败，则必须回滚所有步骤。

可以用START

TRANSACTION语句开始一个事务，然后要么使用COMMIT提交事务将修改的数据持久保留，要么使用ROLLBACK撤销所有的修改。事务SQL的样本如下：

```
START TRANSACTION;
SELECT balance FROM checking WHERE customer_id = 123;
UPDATE checking SET balance = balance – 200.00 WHERE
customer_id = 123;
UPDATE savings SET balance = balance + 200.00 WHERE
customer_id = 123;
COMMIT;
```

ACID：

一个运行良好的事务处理系统，必须具有ACID特性（原子性（atomicity）

一个事务必须被视为一个不可分割的最小工作单元，整个事务中的所有操作要么全部提交成功，要么全部失败回滚，对于一个事务来说，不可能只执行其中的一部分操作，这就是事务的原子性。

（一致性（consistency）

数据库总是从一个一致性的状态转换到另一个一致性的状态。在前面的例子中，一致性确保了，即使在执行第三、四条语句之前系统奔溃了，支票账户中也不会损失200元，因为事务最终没有提交，所有事务中所做的修改也不会保存到数据库中。

（隔离性（isolation）

通常来说，一个事务所做的修改在最终提交以前，对其他事务是不可见的。

（持久性（durability）

一旦事务提交，则其所做的修改就会永久保存到数据库中。此时即使系统崩溃，修改的数据也不会丢失

2、存储过程

1、创建存储过程：

```
create procedure 存储过程名字()
(
[in|out|inout] 参数 datatype
)
begin
MySQL 语句;
end;
```

MySQL存储过程参数如果不显式指定“in”、“out”、“inout”，则默认为“in”。习惯上，对于是“in”的参数，我们都不会显式指定。

2、调用存储过程

CALL 过程名 ([过程参数[,...]])

MySQL存储过程名字后面的“()”是必须的，即使没有一个参数，也需要“()”

MySQL存储过程参数，不能在参数名称前加“@”，如：“@a int”。下面的创建存储过程语法在MySQL中是错误的（在SQLServer中是正确的）。MySQL存储过程中的变量，不需要在变量名字前加“@”，虽然MySQL客户端用户变量要加个“@”。

MySQL存储过程的参数不能指定默认值。

如果 MySQL存储过程中包含多条 MySQL 语句，则需要 begin end 关键字。

MySQL存储过程中的每条语句的末尾，都要加上分号 “;”

MySQL存储过程中的注释：

```
/*
这是个
多行 MySQL 注释。
*/
```

```
declare c int; -- 这是单行 MySQL 注释 （注意 -- 后至少要有一个空格）
if a is null then # 这也是个单行 MySQL 注释
set a = 0;
end if;
...
end;
```

不能在 MySQL 存储过程中使用 “return” 关键字：

```
set c = a + b;
select c as sum;
/*
return c; -- 不能在 MySQL 存储过程中使用。return 只能出现在函数中。
*/
end;
```

调用 MySQL 存储过程时候，需要在过程名字后面加“()”，即使没有一个参数，也需要“()”

call pr_no_param();

3、触发器

1、概念

触发器(Trigger)是一个特殊的存储过程，它的执行不是由程序调用，也不是手工启动，而是由事件触发。

触发器经常用于加强数据的完整性约束和业务规则等。

2、语法

CREATE TRIGGER触发器名称 BEFORE|AFTER 触发事件

ON 表名 FOR EACH ROW

BEGIN

触发器程序体

END

-- 触发器名称，它和MySQL中其他对象的命名方式

-- {BEFORE|AFTER} 触发器触发的时机

-- {INSERT|UPDATE|DELETE} 触发器事件

-- FOR EACH ROW 子句通知触发器每隔一行执行一次动作，而不是对整个表执行一次。

-- (针对insert和delete语句，每一行都触发)

3. 查看触发器

通过show语句查看：show triggers\G

通过系统表查看：

use information_shcema;

select * from tirggers\G

select * from triggers where TRIGGER_NAME = '触发器名称'\G

4. 删除触发器

drop trigger 触发器名称

4、视图

1、语法：

CREATE [OR REPLACE] [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}] VIEW [db_name.]view_name [(column_list)] AS select_statement [WITH [CASCADED | LOCAL] CHECK OPTION]

通过该语句可以创建视图，若给定了[OR

REPLACE]，则表示当已具有同名的视图时，将覆盖原视图。select_statement是一个查询语句，这个查询语句可从表或其它的视图中查

询。视图属于数据库，因此需要指定数据库的名称，若未指定时，表示在当前的数据库创建新视图。

在创建视图前应先看看是否有权限：

select select_priv,create_view_priv from mysql.user where user='root'

Y表示有创建的权限

2、查看视图：

1）DESCRIBE 命令：describe v_view2

2）show table status 查看表状态

3）show create view命令 显示创建视图

3、修改视图

(1)create or replace命令

create or replace view v_view1(id, name, sex) as select id, name, sex from t_employee;

(2) alter 命令

alter view v_view1(id, name) as select id, name from t_employee;

select * from v_view1

4、视图删除：

DROP VIEW [IF EXISTS] view_name [,view_name…]

[RESTRICT|CASEADE]

5、视图创建特点：

1）视图涉及到相关字段名，不能有重复

2）视图名字不能与任何表名重复

3）创建视图的sql查询语句，语法与普通查询一样

4）视图查询与普通sql查询一致，如视图名为news_view，查询语句：select * from news_view

6、视图类型：

1）merge：会将引用视图的语句的文本与视图定义合并起来，使得视图定义的某一部分取代语句的对应部分，效率较高，推荐使用。

2）temptable：视图的结果将被置于临时表中，然后使用它执行语句，方便字查询使用，实用性较低

3）undefined：MySQL将选择所要使用的算法。如果可能，它倾向于MERGE而不是TEMPTABLE，这是因为MERGE通常更有效，而且如果使用了临时表，视图是不可更新的

7、视图的作用：

1）安全性：隐藏真实的数据表，防止数据表泄漏。

2）开发速度提升：例如多表查询，以往要写复杂的语句，现在多个表关联到一个视图，只查询一个视图，语句变得简单□

3）查询速度提升：以往多表查询，需从硬盘中获取多次，而视图只获取一次

8、更新视图：

在MySQL中，更新视图是指通过视图来插入（INSERT）、更新（UPDATE）和删除（DELETE）表中的数据。

因为视图是一个虚拟表，其中没有数据，所以通过视图更新时，都是转换到基本表来更新。

更新视图时，只能更新权限范围内的数据。超出了范围，就不能更新。

UPDATE V_VIEW2 SET POS='高级工程师' WHERE NAME='天天'。

以下类型的视图是不可更新的：

1、包含一下关键字的SQL语句：聚合函数（SUM，MIN，MAX，COUNT等）、DISTINCT、GROUP BY、HAVING、UNION或者UNION ALL□

2、常量视图

1）SELECT中包含子查询

2）JOIN

3） FROM一个不可更新的视图。

4）WHERE子句的子查询引用了FROM子句中的表

5）LOCAL只要满足本视图的条件就可以更新。

6）CASCADED则必须满足所有针对该视图的所有视图的条件才可以更新。

4-1、创建视图

视图包含行和列，就像一个真实的表。视图中的字段就是来自一个或多个数据库中的真实的表中的字段。

创建视图的语法：

```
CREATE VIEW 视图名 AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

```
--创建视图

CREATE VIEW v_user_view AS SELECT t_name,t_job FROM e_user;
```

4-2、使用视图查找数据

查询视图和查询表的语法是一样的。

```
SELECT * FROM v_user_view;
```

4-3、修改视图

使用CREATE OR REPLACE VIEW...语句修改视图。

语法（可以创建要可以更新视图操作）

```
CREATE OR REPLACE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

-- 更新视图
CREATE OR REPLACE VIEW v_user_view AS select id,t_name,t_job from e_user;

4-4、删除视图

--删除视图
DROP VIEW 视图名;

注意:删掉视图不会导致数据的丢失,因为视图是基于数据库的表之上的一个查询定义。

四、小结

1、事务主要用于处理操作量大的，复杂度高的数据
2、存储过程简单来说,就是为以后的使用而保存的一条或多条MySQL语句的集合。存储过程用来做一些日常维护,如数据历史迁移等等
3、触发器,从字面来理解,一触即发的一个器,简称触发器(哈哈,个人理解),举个例子吧,好比天黑了,你开灯了,你看到东西了。你放炮仗,点燃了,一会就炸了。
4、视图的作用就是不必每次都重新编写查询的SQL代码,而是通过视图直接查询即可视图是虚拟表,本身不存储数据,而是按照指定的方式进行查询。

2. 课堂练习

掌握事务、存储过程、触发器、视图

3. 课后练习

1、写一个新闻表的是视图
2、写一个新闻查询的存储过程

4. 资料扩展