**Using Triggers**

1. Create a trigger to record some information after a user inserts a new category. These information should include username, current time, command name and the new row id.

delimiter //

**create** **table** if **not** **exists** user\_info(

id **int** **not** **null** auto\_increment **primary** **key**,

uname **varchar**(40), utime datetime,

cmdname **varchar**(120), rowid **int**

)//

**drop** trigger if **exists** userLog//

**create** trigger userLog after **insert** **on** category

for each row

**begin**

**insert** **into** user\_info(uname, utime, cmdname, rowid)

**values**(current\_user(), now(), "INSERT", New.catid);

**end**//

1. Create a trigger to back up the category before a user delete it from table “category”.

delimiter //

**create** **table** if **not** **exists** category\_backup **like** category//

**drop** trigger if **exists** backupCat//

**create** trigger backupCat before **delete** **on** category

for each row

**begin**

**insert** **into** category\_backup **select** \* **from** category

**where** catid = Old.catid;

**end**//

1. Create a trigger to back up the product before a user updates it in table “product”.

delimiter //

**create** **table** if **not** **exists** product\_backup **like** product//

**drop** trigger if **exists** proBackup//

**create** trigger proBackup before **update** **on** product

for each row

**begin**

**insert** **into** product\_backup **select** \* **from** product

**where** pid = Old.pid;

**end**//

1. Create a table “seller\_report” that contains two column, the seller name and cross sale for each seller then populate data into that table.

**create** **table** if **not** **exists** seller\_report(

seller **varchar**(30) **not** **null**,

totalSale **float**(12,3)

);

**insert** **into** seller\_report

**select** sales.seller, sum(sales.quantity \* product.unitprice)

**from** sales **inner** **join** product **on** sales.pid = product.pid

**group** **by** sales.seller;

1. Create a trigger to recalculate data in table “seller\_report” every time after a user update data in table product.

delimiter //

**drop** trigger if **exists** popSeller//

**create** trigger popSeller after **update** **on** sales

for each row

**begin**

**delete** **from** seller\_report;

**insert** **into** seller\_report

**select** sales.seller, sum(sales.quantity \* product.unitprice)

**from** sales **inner** **join** product **on** sales.pid = product.pid

**group** **by** sales.seller;

**end**//