1) Full backup():

CREATE DEFINER=`root`@`localhost` PROCEDURE `take\_Backup`()

block1: BEGIN

DECLARE tab\_name char(50);

DECLARE q varchar(1500);

DECLARE done INTEGER DEFAULT 0;

DECLARE cursorBackupTable CURSOR FOR SELECT TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES WHERE table\_schema='mydb' and table\_type='BASE TABLE';

DROP DATABASE IF EXISTS mydb\_backup;

CREATE DATABASE mydb\_backup;

open cursorBackupTable;

block2: begin

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

cur\_loop:Loop

FETCH cursorBackupTable into tab\_name;

IF done= 1 THEN LEAVE cur\_loop;

END IF;

SET @q=CONCAT('DROP TABLE IF EXISTS mydb\_backup.',tab\_name);

PREPARE stmt FROM @q;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

SET @q=CONCAT('CREATE TABLE mydb\_backup.',tab\_name,' AS SELECT \* FROM mydb.',tab\_name,' WHERE 1=1');

PREPARE stmt FROM @q;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

END LOOP cur\_loop;

END block2;

close cursorBackupTable;

END block1

**Triggers:**

1)

Create trigger onlinessale\_AINS, after update on onlineorderitems

for each row

Begin

update inventory

SET quantity = quantity - NEW.quantity

WHERE productid = NEW.productid,

end

2)

Create trigge rsale\_AINS, after update on orderitems

for each row

Begin

update inventory

SET quantity = quantity - NEW.quantity

WHERE productid = NEW.productid

end

3)

create trigger cancelorder, after delete on orderitems,

For each row

Begin

Update inventory

set quantity = quantity+ old.quantity

where productid=old.productid;

end,

4)

create trigger cancelonlineorder, after delete on onlineorderitems,

For each row

Begin

Update inventory

set quantity = quantity+ old.quantity

where productid=old.productid;

end

Stored Procedures

1)

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `addItem`( prodid int, quant int, ordid int)

Begin

if (quant> (select quantity from inventory where productid = prodid)) then

Select concat('Sorry. Available quantity for selected Product is only ',

(select quantity from inventory where productid = prodid)) as 'ERROR';

else

Insert into onlineorderitems(productID,quantity,onlineorderid, costForProduct)

values((select productid from product where productid = prodid), quant, ordid, ((select price from product where productid = prodid) \* quant));

End if;

end$$

DELIMITER ;

2)

show triggers;

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `addOrderItem`(prodid int , quant int, ordid int)

Begin

if (quant> (select quantity from inventory where productid = prodid)) then

Select concat('Sorry. Available quantity for selected Product is only ',

(select quantity from inventory where productid = prodid)) as 'ERROR';

else

Insert into orderitems(productID,quantity,orderid, amount) values((select productid from product where productid = prodid), quant, ordid, ((select price from product where productid = prodid) \* quant));

End if;

end$$

DELIMITER ;

3)

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `cancelOnlineOrder`(oid int)

BEGIN

declare message varchar(100);

select shippingstatus from onlineorders where OnlineOrderID = oid into message;

if message = 'Delivered'

then

Select 'Sorry. Too Late! Order has been Deliverd' as 'Error';

else

delete from onlineorderitems where onlineorderid = oid;

UPDATE onlineorders

SET orderstatus = 'Canceled', Amount =0 where onlineorderid=oid;

END if;

END$$

DELIMITER ;

4)

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `cancelOrder`(oid int)

BEGIN

delete from orderitems where orderid = oid;

UPDATE orders

SET orderstatus = 'Canceled', OrderAmount =0 where orderid=oid;

END$$

DELIMITER ;

5)

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `concludeonlineOrder`(in oid int)

Begin

update onlineorders

set amount =(select sum(costForProduct) from onlineorderitems where onlineorderid = oid),

orderstatus = 'completed', ShippingStatus='Ships next day of order' where onlineorderid=oid;

insert into shipment (OnlineOrderId, ShipmentStatus, LocationID, shipmentname)

values( oid , "Order Issued",

(select locationid from person where personid = (select personid from onlineorders where onlineorderid=oid)),

concat('Shipment - Order no-',oid));

END$$

DELIMITER ;

6)

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `concludeorder`(oid int)

Begin

update orders

set Orderamount =(select sum(amount) from orderitems where orderid = oid) , Orderstatus ='Completed' where orderid=oid;

END$$

DELIMITER ;

7)

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `createNewOrder`( pid int, out oid int)

begin

insert into orders(personid) values (pid);

select max(orderid) from orders into oid;

end$$

DELIMITER ;

8)

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `createOnlineOrder`(persID int, out oid int)

begin

declare message varchar(100);

select checkage(persID) into message;

if message = 'Cannot buy'

then

Select 'Customer underage or unverified';

else

insert into onlineorders(personid) values (persid);

select max(OnlineOrderID) from onlineorders into oid;

end if;

end$$

DELIMITER ;

9)

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `Shippingdone`(in oid int)

BEGIN

update shipment

Set Shipmentstatus = 'Shipment Completed' where OnlineOrderId = oid;

update onlineorders

set ShippingStatus = 'Delivered' where OnlineOrderID = oid;

End$$

DELIMITER ;

10)

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `supplyproducts`(in sid int, in pid int, in quant int)

begin

insert into product\_supplies (productid,supplierID, Quantity) values( pid,sid,quant);

update inventory

set quantity = (Quantity + quant) where productid=pid;

end$$

DELIMITER ;

11)

CREATE DEFINER=`root`@`localhost` FUNCTION `checkage`( pid int) RETURNS varchar(80) CHARSET utf8

begin

declare message varchar(80);

declare age float;

declare veri varchar (20);

set age:= (Select TIMESTAMPDIFF(YEAR, (select dateofbirth from person where PersonID=12), CURDATE()));

set veri:= (select eligible from person where personid=pid);

if (pid > (select max(personid) from person))

then set message = 'Person doesnot exist';

elseif ((age<21) || (veri = 'Unverified')) THEN

set message = 'Cannot Buy';

ELSE

set message ='Person is elgibile';

end if;

return message;

END$$

DELIMITER ;