**HOMEWORK CREATE STORE PROCEDURE**

1. Create a simple stored procedure to calculate your age by input born year.

DELIMITER //

**CREATE** **PROCEDURE** findAge (IN bornYear **int**)

**BEGIN**

**SELECT** **YEAR** (CURDATE ())- bornYear **as** Age;

**END** //

2. Create a simple stored procedure to calculate the sum, product, division, power and average of two floating point numbers dynamically.

DELIMITER //

**CREATE** **PROCEDURE** Call2Number (IN n1 **FLOAT** (11), **IN** n2 **FLOAT** (11))

**BEGIN**

**SELECT** **SUM** (n1+n2) **as** "Sum",

**CONCAT** (n1\*n2) **as** "Product",

**CONCAT** (n1-n2) **as** "Minus",

**CONCAT** (n1/n2) **as** "Division",

**POWER** (n1,n2) **as** "Power",

**CONCAT** (n1+n2)/2 **as** "Average";

**END** //

3.Create a stored procedure to select all category from table category.

DELIMITER //

**CREATE** **PROCEDURE** getCatList ()

**BEGIN**

**SELECT**

category.catid **as** "Cat ID",

category.catname **as** "CateName",

category.description **as** "Description"

**FROM** category;

**END** //

4. Create a stored procedure to find the product that has minimum quantity.

DELIMITER //

**CREATE** **PROCEDURE** minQuantity()

**BEGIN**

**SELECT** product.pid,product.pname,product.supplier,

product.unitprice,product.quantity,product.catid,

product.storeid,product.description

**FROM** product **WHERE**

product.quantity = (**SELECT** **MIN**(product.quantity)**FROM** product)

**GROUP** **BY** catid;

**END** //

6. Create a stored procedure to delete product dynamically by its id.

DELIMITER //

**CREATE** **PROCEDURE** deleteProduct(**IN** proid **int**(11))

**BEGIN**

**DELETE** **FROM** product **WHERE** pid = proid;

**END** //

7. Create a stored procedure to show all products that don’t have category or have wrong category.

DELIMITER //

**CREATE** **PROCEDURE** getwrongCategory()

**BEGIN**

**SELECT** p.pid,p.pname,p.supplier,p.unitprice,

p.quantity,p.catid,p.storeid,p.description

**FROM** product p **WHERE** p.catid

**NOT** **IN** (**SELECT** category.catid **FROM** category)

**OR** p.catid = **NULL**;

**END** //

8. Create a stored procedure to show all products that don’t have category or have wrong category.

DELIMITER //

**CREATE** **PROCEDURE** AddCategory (**IN** catName **varchar**(40),**IN** descrip **varchar** (40))

**BEGIN**

**IF**(catName **IN**( **SELECT** category.catname **FROM** category)) **THEN**

**SELECT** **CONCAT**(catName," has been already");

**ELSE**

**INSERT** **INTO** category(catname,description) **VALUES** (catName,descrip);

**END** **IF**;

**END/**/

9. Create a stored procedure to search product by its name.

DELIMITER //

**CREATE** **PROCEDURE** SearchProduct (**IN** search **varchar** (200))

**BEGIN**

**SELECT** p.pid,p.pname,p.supplier,p.unitprice,p.quantity,p.catid,

p.storeid,p.description

**FROM** product p **WHERE** p.pname **LIKE** (**CONCAT**("%",search,"%"));

**END** //

10. Create a stored procedure to search product by its name.

DELIMITER //

**CREATE** **PROCEDURE** SaleReportByYear()

**BEGIN**

**SELECT** **YEAR**(sales.salesdate) **as** **YEAR** ,

**MIN**(product.unitprice\*sales.quantity) **as** "Min",

**MAX**(product.unitprice\*sales.quantity) **as** "Max",

**SUM**(sales.quantity)/**COUNT**(sales.quantity) **as** "Average Sale",

**SUM**(sales.quantity\*product.unitprice) **as** "Total"

**FROM** product **INNER** **JOIN**

sales **ON** product.pid=sales.pid **GROUP** **BY** **year**(salesdate);

**END/**/