作业6答案

第一题答案是：

**delimiter $$**

**set global log\_bin\_trust\_function\_creators=TRUE $$**

**DROP FUNCTION IF EXISTS func\_rand\_ID $$**

**CREATE FUNCTION func\_rand\_ID() RETURNS varchar(20)**

**BEGIN**

**DECLARE res VARCHAR(20) DEFAULT "";**

**DECLARE num\_alpha VARCHAR(11) DEFAULT "0123456789X";**

**DECLARE date VARCHAR(20) DEFAULT "";**

**DECLARE i INT DEFAULT 0;**

**SET @len1 = 6;**

**SET @len2 = 3;**

**WHILE i<@len1 DO**

**SET res = CONCAT(res, substring(num\_alpha, FLOOR(1+RAND()\*10), 1));**

**SET i = i+1;**

**END WHILE;**

**SELECT DATE\_ADD("1960-01-01",INTERVAL rand()\*21915 day) INTO date;**

**SET res = CONCAT(res, year(date));**

**SET res = CONCAT(res, substring(date, 6, 2));**

**SET res = CONCAT(res, substring(date, 9, 2));**

**SET i = 0;**

**WHILE i<@len2 DO**

**SET res = CONCAT(res, substring(num\_alpha, FLOOR(1+RAND()\*10), 1));**

**SET i = i+1;**

**END WHILE;**

**SET res = CONCAT(res, substring(num\_alpha, FLOOR(1+RAND()\*11), 1));**

**return res;**

**END $$**

**delimiter ;**

第二题答案是：

**DROP TABLE IF EXISTS `testuser`;**

**CREATE TABLE `testuser` (**

**`uid` int(11) NOT NULL AUTO\_INCREMENT,**

**`username` varchar(20) DEFAULT NULL,**

**`email` varchar(150) DEFAULT NULL,**

**`telephone` varchar(20) DEFAULT NULL,**

**`ucode` varchar(20) DEFAULT NULL,**

**PRIMARY KEY (`uid`)**

**) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;**

**DROP PROCEDURE IF EXISTS createTestCases;**

**CREATE PROCEDURE createTestCases()**

**BEGIN**

**DECLARE alpha varchar(80) DEFAULT "1234567890qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM";**

**DECLARE i INT DEFAULT 0;**

**DECLARE j INT DEFAULT 0;**

**DECLARE username varchar(10) DEFAULT "";**

**WHILE i<100 DO**

**SET username = "";**

**SET j=0;**

**WHILE j<8 DO**

**SET username = CONCAT(username, substring(alpha, FLOOR(RAND()\*62+1), 1));**

**SET j=j+1;**

**END WHILE;**

**SET i=i+1;**

**INSERT INTO testuser VALUES (NULL, username, func\_rand\_email(), func\_rand\_telnum(), func\_rand\_ID());**

**END WHILE;**

**END;**

第三题答案是：

**DROP PROCEDURE IF EXISTS getEmpStructure;**

**CREATE PROCEDURE getEmpStructure()**

**BEGIN**

**WITH RECURSIVE employee\_info AS**

**(**

**SELECT EMP\_ID, CONCAT(LAST\_NAME,FIRST\_NAME) as NAME, 1 as level, TITLE**

**FROM employee**

**WHERE ISNULL(SUPERIOR\_EMP\_ID)**

**UNION ALL**

**SELECT employee.EMP\_ID, CONCAT(employee.LAST\_NAME, employee.FIRST\_NAME) as NAME, employee\_info.level+1 as level, employee.TITLE**

**FROM employee INNER JOIN employee\_info**

**ON employee.SUPERIOR\_EMP\_ID = employee\_info.EMP\_ID**

**)**

**SELECT \* FROM employee\_info;**

**END;**

第四题答案是：

**DROP PROCEDURE IF EXISTS updateCloseDate;**

**CREATE PROCEDURE updateCloseDate()**

**BEGIN**

**DECLARE a int;**

**DECLARE b varchar(255);**

**DECLARE c date;**

**DECLARE d DECIMAL(12, 4);**

**DECLARE cur CURSOR FOR**

**SELECT ACCOUNT\_ID, PRODUCT\_TYPE\_CD, OPEN\_DATE, AVAIL\_BALANCE**

**FROM account NATURAL JOIN product;**

**DECLARE exit HANDLER FOR NOT FOUND CLOSE cur;**

**OPEN cur;**

**REPEAT**

**FETCH cur INTO a,b,c,d;**

**UPDATE account**

**SET account.CLOSE\_DATE = ADDDATE(c,INTERVAL 30 year)**

**WHERE b = "ACCOUNT" and a = account.ACCOUNT\_ID and UNIX\_TIMESTAMP(account.OPEN\_DATE) > UNIX\_TIMESTAMP("2015-01-01");**

**UPDATE account**

**SET account.CLOSE\_DATE = ADDDATE(c,INTERVAL 20 year)**

**WHERE b = "ACCOUNT" and a = account.ACCOUNT\_ID and UNIX\_TIMESTAMP(account.OPEN\_DATE) <= UNIX\_TIMESTAMP("2015-01-01");**

**UPDATE account**

**SET account.CLOSE\_DATE = ADDDATE(c,INTERVAL 30 year)**

**WHERE b = "LOAN" and a = account.ACCOUNT\_ID and d>100000;**

**UPDATE account**

**SET account.CLOSE\_DATE = ADDDATE(c,INTERVAL 20 year)**

**WHERE b = "LOAN" and a = account.ACCOUNT\_ID and d<=100000;**

**UPDATE account**

**SET account.CLOSE\_DATE = ADDDATE(c,INTERVAL 15 year)**

**WHERE b = "INSURANCE" and a = account.ACCOUNT\_ID;**

**UNTIL 0 END REPEAT;**

**CLOSE cur;**

**END;**

第五题答案是：

**DROP TRIGGER IF EXISTS t\_newTransaction;**

**delimiter $$**

**CREATE TRIGGER t\_newTransaction BEFORE INSERT ON acc\_transaction for each row**

**BEGIN**

**SET @msg = "余额不足Insufficient Balance";**

**IF (new.TXN\_TYPE\_CD in ("CD", "TT", "IC", "LI")) THEN**

**UPDATE account**

**SET account.AVAIL\_BALANCE = account.AVAIL\_BALANCE + new.AMOUNT**

**WHERE account.ACCOUNT\_ID = new.ACCOUNT\_ID;**

**UPDATE account**

**SET account.LAST\_ACTIVITY\_DATE = CURDATE()**

**WHERE account.ACCOUNT\_ID = new.ACCOUNT\_ID;**

**ELSEIF (new.TXN\_TYPE\_CD in ("CW", "TF")) THEN**

**IF(account.AVAIL\_BALANCE-new.AMOUNT < 0) THEN**

**SIGNAL SQLSTATE 'HY000' SET MESSAGE\_TEXT = @msg;**

**ELSE**

**UPDATE account**

**SET account.AVAIL\_BALANCE = account.AVAIL\_BALANCE - new.AMOUNT**

**WHERE account.ACCOUNT\_ID = new.ACCOUNT\_ID;**

**UPDATE account**

**SET account.LAST\_ACTIVITY\_DATE = CURDATE()**

**WHERE account.ACCOUNT\_ID = new.ACCOUNT\_ID;**

**END IF;**

**END IF;**

**END $$**

**delimiter ;**