

1

1.1 SHOP.GET_INFO_CLIENT_GOODS

```

-----
--      :
--
--      /
--      : 27/01/2007
--      :
-----
CREATE OR REPLACE PROCEDURE SHOP.GET_INFO_CLIENT_GOODS
(
  P_CLIENT OUT NUMBER,
  P_GOODS OUT NUMBER
)
IS
BEGIN
  COMMIT;
  SET TRANSACTION READ ONLY NAME 'SHOP.TABLE_USER,SHOP.TABLE_GOODS';
  SELECT COUNT(*) INTO P_CLIENT FROM SHOP.TABLE_USER;
  SELECT COUNT(*) INTO P_GOODS FROM SHOP.TABLE_GOODS;
  COMMIT;
END;
/

```

1.2 SYSTEM.UPDATE_USER_LAST

```

-----
--      :
--
--      : 27/01/2007
--      :
-----
CREATE OR REPLACE PROCEDURE SYSTEM.UPDATE_USER_LAST
(
  P_USER_W_ID IN SYSTEM.TABLE_USER_LAST.USER_W_ID%TYPE
)
IS
  E      EXCEPTION;
  PRAGMA EXCEPTION_INIT(E, -08177);
  D DATE;
BEGIN
  COMMIT;
  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE NAME 'SYSTEM.TABLE_USER_LAST';
  SAVEPOINT A;
  SELECT SYSDATE INTO D FROM DUAL;
  UPDATE SYSTEM.TABLE_USER_LAST SET
    LASTTIMES=D
    WHERE USER_W_ID=p_USER_W_ID;
  COMMIT;
EXCEPTION
  WHEN E THEN
    ROLLBACK TO A;
  WHEN OTHERS THEN
    ROLLBACK TO A;
END;
/

```

2

2.1 SHOP.REGISTRATION

```

-----
--      :
--
--      : 27/01/2007
--      :
-----

```

```

-----
CREATE OR REPLACE FUNCTION SHOP.REGISTRATION
(
  PLOGIN          IN SHOP.TABLE_USER.LOGIN%TYPE,
  PPASSWORD       IN SHOP.TABLE_USER.PASSWORD%TYPE,
  PMAIL           IN SHOP.TABLE_USER.MAIL%TYPE,
  PPHONE          IN SHOP.TABLE_USER.PHONE%TYPE,
  PSNP            IN SHOP.TABLE_USER.SNP%TYPE,
  PINDEXX         IN SHOP.TABLE_ADDRESS_U.INDEXX%TYPE,
  PCOUNTRY        IN SHOP.TABLE_ADDRESS_U.COUNTRY%TYPE,
  PAREA           IN SHOP.TABLE_ADDRESS_U.AREA%TYPE,
  PCITY           IN SHOP.TABLE_ADDRESS_U.CITY%TYPE,
  PSTREET         IN SHOP.TABLE_ADDRESS_U.STREET%TYPE,
  PHOUSE          IN SHOP.TABLE_ADDRESS_U.HOUSE%TYPE,
  PCASES          IN SHOP.TABLE_ADDRESS_U.CASES%TYPE,
  PAPARTMENT      IN SHOP.TABLE_ADDRESS_U.APARTMENT%TYPE
) RETURN NUMBER
IS
  NUM SHOP.TABLE_USER.ID%TYPE;
BEGIN
  COMMIT;
  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE NAME
    'SHOP.TABLE_USER, SHOP.TABLE_ADDRESS_U';
  SAVEPOINT A;
  INSERT INTO SHOP.TABLE_ADDRESS_U (INDEXX,COUNTRY,AREA,CITY,STREET,HOUSE,CASES,APARTMENT)
    VALUES (PINDEXX,PCOUNTRY,PAREA,PCITY,PSTREET,PHOUSE,PCASES,PAPARTMENT);
  SELECT MAX(ID) INTO NUM FROM TABLE_ADDRESS_U;
  INSERT INTO SHOP.TABLE_USER (LOGIN, PASSWORD, MAIL, ADDRESS_U_ID, PHONE, SNP)
    VALUES (PLOGIN, PPASSWORD, PMAIL, NUM, PPHONE, PSNP);
  SELECT MAX(ID) INTO NUM FROM TABLE_USER;
  COMMIT;
  RETURN NUM;
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK TO A;
    RETURN 0;
END;
/

```

2.2 SHOP.AUTH_USER

```

-----
--          :
--
--      : 27/01/2007
--          :
-----
CREATE OR REPLACE FUNCTION SHOP.AUTH_USER
(
  P_LOGIN      IN SHOP.TABLE_USER.LOGIN%TYPE,
  P_PASSWORD   IN SHOP.TABLE_USER.PASSWORD%TYPE
) RETURN NUMBER
IS
  I NUMBER(9);
  COUNTT NUMBER(9);
BEGIN
  COMMIT;
  SET TRANSACTION READ ONLY NAME 'SHOP.TABLE_USER';
  SELECT COUNT(*) INTO COUNTT FROM SHOP.TABLE_USER
    WHERE (LOGIN=P_LOGIN)AND(PASSWORD=P_PASSWORD);
  IF (COUNTT=0) THEN RETURN 0; END IF;
  SELECT ID INTO I FROM SHOP.TABLE_USER
    WHERE (LOGIN=P_LOGIN);
  COMMIT;
  RETURN I;
EXCEPTION
  WHEN OTHERS THEN
    COMMIT;
    RETURN 0;
END;
/

```

2.3 SHOP.REGISTRATION_NEW_GOODS

```
-- :  
-- : 27/01/2007  
-- :  
-----  
CREATE OR REPLACE FUNCTION SHOP.REGISTRATION_NEW_GOODS  
(  
    P_NAME                IN SHOP.TABLE_GOODS.NAME%TYPE,  
    P_CATEGORY_ID         IN SHOP.TABLE_GOODS.CATEGORY_ID%TYPE,  
    P_CHARACTERISTICS      IN SHOP.TABLE_GOODS.CHARACTERISTICS%TYPE,  
    P_BREND               IN SHOP.TABLE_GOODS.BREND%TYPE,  
    P_DESCRIPTION          IN SHOP.TABLE_GOODS.DESCRPTION%TYPE,  
    P_WARRANTY_PERIOD      IN SHOP.TABLE_GOODS.WARRANTY_PERIOD%TYPE,  
    P_SUPPLIER_ID          IN SHOP.TABLE_GOODS.SUPPLIER_ID%TYPE,  
    P_COST                 VARCHAR2 -- JDBC INTEGER(20,2)  
) RETURN NUMBER -- IMAGE  
IS  
    COST     SHOP.TABLE_GOODS.COST%TYPE;  
    NUM       SHOP.TABLE_GOODS.ID%TYPE; -- BLOB  
    NUMG      SHOP.TABLE_GOODS.ID%TYPE; -- GOODS  
    V_BLOB    BLOB;  
    CURSOR CURW IS SELECT ID FROM SHOP.TABLE_WAREHOUSES;  
    NUMW      SHOP.TABLE_WAREHOUSES.ID%TYPE; -- TABLE_WAREHOUSES.ID  
BEGIN  
    COMMIT;  
    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE NAME  
        'SHOP.TABLE_GOODS, SHOP.TABLE_IMAGE,  
        'SHOP.TABLE_INVENTORIES, SHOP.TABLE_WAREHOUSES';  
    SAVEPOINT A;  
-----  
    COST:=TO_NUMBER(P_COST,'9999999999999999D99'); -- 20 : 18 2  
    INSERT INTO SHOP.TABLE_IMAGE (BLOBS)  
        VALUES (EMPTY_BLOB()) RETURNING BLOBS INTO V_BLOB;  
    SELECT MAX(ID) INTO NUM FROM SHOP.TABLE_IMAGE; --NUM ,  
        -- . .  
-----  
    INSERT INTO SHOP.TABLE_GOODS  
        (NAME, CATEGORY_ID, CHARACTERISTICS, BREND, DESCRIPTION, WARRANTY_PERIOD,  
        SUPPLIER_ID, COST, IMAGE_ID)  
    VALUES  
        (P_NAME, P_CATEGORY_ID, P_CHARACTERISTICS, P_BREND, P_DESCRIPTION,  
        P_WARRANTY_PERIOD, P_SUPPLIER_ID, P_COST, NUM);  
-----  
    SELECT MAX(ID) INTO NUMG FROM SHOP.TABLE_GOODS;  
    OPEN CURW;  
    LOOP  
        FETCH CURW INTO NUMW;  
        EXIT WHEN CURW%NOTFOUND;  
        INSERT INTO SHOP.TABLE_INVENTORIES (WAREHOUSES_ID,GOODS_ID,AMOUNT)  
            VALUES (NUMW,NUMG,0);  
    END LOOP;  
    CLOSE CURW;  
-----  
    COMMIT;  
    RETURN NUM;  
EXCEPTION  
    WHEN OTHERS THEN  
        ROLLBACK TO A;  
        RETURN 0;  
END;  
/  

```

2.4 SHOP.FIND_STRING

```
--          :
--          <ID1;...IDN>
--          : 27/01/2007
--          :
-----:
CREATE OR REPLACE FUNCTION SHOP.FIND_STRING
(
```

```

    S IN VARCHAR2
)
RETURN STRING
IS
    SID    VARCHAR2(32767);
    PID    SHOP.TABLE_GOODS.ID%TYPE;
    PNAME  SHOP.TABLE_GOODS.NAME%TYPE;
    CURSOR CUR1 IS SELECT ID,NAME FROM SHOP.TABLE_GOODS;
BEGIN
    SID:='';
    OPEN CUR1;
    LOOP
        FETCH CUR1 INTO PID, PNAME;
        EXIT WHEN CUR1%NOTFOUND;
        -----
        IF (INSTR(PNAME,S)>0) THEN
            SID:=CONCAT(SID,PID);
            SID:=CONCAT(SID,',' );
        END IF;
        -----
    END LOOP;
    CLOSE CUR1;
    RETURN SID;
END;
/

```

2.5 SYSTEM.GETLASTTIME

```

-----
--          :
--          -
--      : 27/01/2007
--          :
-----

CREATE OR REPLACE FUNCTION SYSTEM.GETLASTTIME
(
    P_USER_W_ID IN SYSTEM.TABLE_USER_W.ID%TYPE
) RETURN DATE
IS
    D DATE;
BEGIN
    SELECT LASTTIMES INTO D FROM SYSTEM.TABLE_USER_LAST
        WHERE USER_W_ID=P_USER_W_ID;
    RETURN D;
EXCEPTION
    WHEN OTHERS THEN
        RETURN NULL;
END;
/

```

2.6 SYSTEM.AUTH

```

-----
--          :
--          -
--      : 27/01/2007
--          :
-----

CREATE OR REPLACE FUNCTION SYSTEM.AUTH
(
    P_LOGIN      IN SYSTEM.TABLE_USER_W.LOGIN%TYPE,
    P_PASSWORD   IN SYSTEM.TABLE_USER_W.PASSWORD%TYPE,
    P_DATE       OUT SYSTEM.TABLE_USER_LAST.LASTTIMES%TYPE
) RETURN NUMBER
IS
    I SYSTEM.TABLE_USER_W.ID%TYPE;
    COUNTT SYSTEM.TABLE_USER_W.ID%TYPE;
BEGIN
    COMMIT;
    SET TRANSACTION READ ONLY NAME 'SYSTEM.TABLE_USER_LAST,SYSTEM.TABLE_USER_W';
    SELECT COUNT(*) INTO COUNTT FROM SYSTEM.TABLE_USER_W
        WHERE (LOGIN=P_LOGIN)AND(PASSWORD=P_PASSWORD);
    P_DATE:='1.1.1900'; --

```

```

IF (COUNT=0) THEN RETURN 0; END IF;
SELECT ID INTO I FROM SYSTEM.TABLE_USER_W
WHERE (LOGIN=P_LOGIN)AND(PASSWORD=P_PASSWORD);
P_DATE:=SYSTEM.GETLASTTIME(I);
SYSTEM.UPDATE_USER_LAST(I);
COMMIT;
RETURN 1;
EXCEPTION
WHEN OTHERS THEN
COMMIT;
RETURN 0;
END;
/

```

2.7 SHOP.ADD_WAREHOUSES

```

-----
--      :
--
--      : 27/01/2007
--      :
-----

CREATE OR REPLACE FUNCTION SHOP.ADD_WAREHOUSES
(
P_NAME          IN SHOP.TABLE_WAREHOUSES.NAME%TYPE,
P_COUNTRY       IN SHOP.TABLE_ADDRESS_U.COUNTRY%TYPE,
P_AREA          IN SHOP.TABLE_ADDRESS_U.AREA%TYPE,
P_CITY          IN SHOP.TABLE_ADDRESS_U.CITY%TYPE,
P_STREET        IN SHOP.TABLE_ADDRESS_U.STREET%TYPE,
P_HOUSE         IN SHOP.TABLE_ADDRESS_U.HOUSE%TYPE
) RETURN NUMBER
IS
NUMW  SHOP.TABLE_WAREHOUSES.ID%TYPE; --      TABLE_WAREHOUSES.ID;
NUMG  SHOP.TABLE_GOODS.ID%TYPE;      --      TABLE_GOODS.ID;
NUMA  SHOP.TABLE_ADDRESS_W.ID%TYPE;  --      TABLE_ADDRESS_W.ID
CURSOR CURG IS SELECT ID FROM SHOP.TABLE_GOODS;
BEGIN
COMMIT;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE NAME 'SHOP.TABLE_ADDRESS_W,
SHOP.TABLE_GOODS,
SHOP.TABLE_WAREHOUSES,
SHOP.TABLE_INVENTORIES';

SAVEPOINT A;
-----
INSERT INTO SHOP.TABLE_ADDRESS_W (COUNTRY,AREA,CITY,STREET,HOUSE)
VALUES (P_COUNTRY,P_AREA,P_CITY,P_STREET,P_HOUSE);
SELECT MAX(ID) INTO NUMA FROM SHOP.TABLE_ADDRESS_W;
INSERT INTO SHOP.TABLE_WAREHOUSES (NAME, ADDRESS_W_ID)
VALUES (P_NAME, NUMA);
-----
SELECT MAX(ID) INTO NUMW FROM SHOP.TABLE_WAREHOUSES;
OPEN CURG;
LOOP
FETCH CURG INTO NUMG;
EXIT WHEN CURG%NOTFOUND;
INSERT INTO SHOP.TABLE_INVENTORIES (WAREHOUSES_ID,GOODS_ID,AMOUNT)
VALUES (NUMW,NUMG,0);
END LOOP;
CLOSE CURG;
-----
COMMIT;
RETURN 1;
EXCEPTION
WHEN OTHERS THEN
ROLLBACK TO A;
RETURN 0;
END;
/

```

2.8 SHOP.ADD_DELIVERY

```

-----
--          :
--
--      : 11/02/2007
--          :
-----
CREATE OR REPLACE FUNCTION SHOP.ADD_DELIVERY
(
  P_WAY  IN SHOP.TABLE_DELIVERY.WAY%TYPE,
  P_COST VARCHAR2
) RETURN NUMBER
IS
  COST  SHOP.TABLE_GOODS.COST%TYPE;
BEGIN
  COMMIT;
  SAVEPOINT A;
  COST:=TO_NUMBER(P_COST, '9999999999999999D99' );
  INSERT INTO SHOP.TABLE_DELIVERY (WAY,COST)
    VALUES(P_WAY,P_COST);
  COMMIT;
  RETURN 1;
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK TO A;
    RETURN 0;
END;
/

```

2.9 SHOP.ADD_PAYMENT

```

-----
--          :
--
--      : 11/02/2007
--          :
-----
CREATE OR REPLACE FUNCTION SHOP.ADD_PAYMENT
(
  P_WAY  IN SHOP.TABLE_PAYMENT.WAY%TYPE,
  P_COST VARCHAR2
) RETURN NUMBER
IS
  COST  SHOP.TABLE_PAYMENT.COST%TYPE;
BEGIN
  COMMIT;
  SAVEPOINT A;
  COST:=TO_NUMBER(P_COST, '9999999999999999D99' );
  INSERT INTO SHOP.TABLE_PAYMENT (WAY,COST)
    VALUES(P_WAY,P_COST);
  COMMIT;
  RETURN 1;
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK TO A;
    RETURN 0;
END;
/

```

2.10 SHOP.ADD_SUPPLIER

```

-----
--          :
--
--      : 27/01/2007
--          :
-----
CREATE OR REPLACE FUNCTION SHOP.ADD_SUPPLIER
(
  P_NAME IN SHOP.TABLE_SUPPLIER.NAME%TYPE
) RETURN NUMBER

```

```

IS
BEGIN
  COMMIT;
  SAVEPOINT A;
  INSERT INTO SHOP.TABLE_SUPPLIER (NAME)
    VALUES(P_NAME);
  COMMIT;
  RETURN 1;
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK TO A;
    RETURN 0;
END;
/

```

2.11 SHOP.ADD_CATEGORY

```

-----
--                                     :
--                                     :
--      : 27/01/2007
--                                     :
-----
CREATE OR REPLACE FUNCTION SHOP.ADD_CATEGORY
(
  P_NAME  IN SHOP.TABLE_CATEGORY.NAME%TYPE
) RETURN NUMBER
IS
BEGIN
  COMMIT;
  SAVEPOINT A;
  INSERT INTO SHOP.TABLE_CATEGORY (NAME)
    VALUES (P_NAME);
  COMMIT;
  RETURN 1;
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK TO A;
    RETURN 0;
END;
/

```

2.12 SHOP.ISSUE_WAYBILL

```

-----
--                                     :
--                                     :
--      : 27/01/2007
--                                     :
-----
CREATE OR REPLACE FUNCTION SHOP.ISSUE_WAYBILL
(
  PWAREHOUSES_ID  IN SHOP.TABLE_WAYBILL.WAREHOUSES_ID%TYPE,
  PGOODS_ID       IN SHOP.TABLE_WAYBILL.GOODS_ID%TYPE,
  PAMOUNT         IN SHOP.TABLE_WAYBILL.AMOUNT%TYPE
) RETURN NUMBER
IS
  E      EXCEPTION;
  PRAGMA EXCEPTION_INIT(E, -00054);
  D      DATE;
  N      SHOP.TABLE_INVENTORIES.ID%TYPE;
  N_ALL  SHOP.TABLE_INVENTORIES.AMOUNT%TYPE; --
BEGIN
  COMMIT;
  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE NAME 'SHOP.TABLE_INVENTORIES,
    SHOP.TABLE_WAYBILL';

  SAVEPOINT A;
  SELECT SYSDATE INTO D FROM DUAL;
  INSERT INTO TABLE_WAYBILL(TIMES, WAREHOUSES_ID, GOODS_ID, AMOUNT)
    VALUES (D, PWAREHOUSES_ID, PGOODS_ID, PAMOUNT);
  -----
  -----TABLE_INVENTORIES-----
  SELECT ID, AMOUNT INTO N, N_ALL FROM TABLE_INVENTORIES
    WHERE (GOODS_ID=PGOODS_ID) AND (WAREHOUSES_ID=PWAREHOUSES_ID) FOR UPDATE NOWAIT;

```

```

UPDATE TABLE_INVENTORIES SET
    AMOUNT = N_ALL + PAMOUNT
    WHERE ID = N;
COMMIT;
RETURN 1;
EXCEPTION
    WHEN E THEN
        ROLLBACK TO A;
        RETURN -1;
    WHEN OTHERS THEN
        ROLLBACK TO A;
        RETURN 0;
END;
/

```

2.13 SYSTEM.CANCEL_REG_PERS_SHOP

```

-----
--          :
--
--      : 27/01/2007
--          :
-----
CREATE OR REPLACE FUNCTION SYSTEM.CANCEL_REG_PERS_SHOP
(
    P_LOGIN      IN SYSTEM.TABLE_USER_W.LOGIN%TYPE
) RETURN NUMBER
IS
    NUM SYSTEM.TABLE_USER_W.ID%TYPE;
BEGIN
    COMMIT;
    SAVEPOINT A;
    SELECT ID INTO NUM FROM SYSTEM.TABLE_USER_W
        WHERE LOGIN=P_LOGIN;
    DELETE FROM SYSTEM.TABLE_USER_LAST
        WHERE USER_W_ID=NUM;
    DELETE FROM SYSTEM.TABLE_USER_W
        WHERE ID=NUM;
    COMMIT;
    RETURN 1;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK TO A;
        RETURN 0;
END;
/

```

2.14 SYSTEM.REGISTRATION_PERSONNEL_SHOP

```

-----
--          :
--
--      : 27/01/2007
--          :
-----
CREATE OR REPLACE FUNCTION SYSTEM.REGISTRATION_PERSONNEL_SHOP
(
    P_LOGIN      IN SYSTEM.TABLE_USER_W.LOGIN%TYPE,
    P_PASSWORD    IN SYSTEM.TABLE_USER_W.PASSWORD%TYPE
) RETURN NUMBER
IS
    NUM SYSTEM.TABLE_USER_W.ID%TYPE;
    D    DATE;
BEGIN
    COMMIT;
    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE NAME
        'SYSTEM.TABLE_USER_W, SYSTEM.TABLE_USER_LAST';
    SAVEPOINT A;
    -----
    SELECT SYSDATE INTO D FROM DUAL;
    INSERT INTO SYSTEM.TABLE_USER_W (LOGIN,PASSWORD)
        VALUES(P_LOGIN,P_PASSWORD);
    SELECT MAX(ID) INTO NUM FROM SYSTEM.TABLE_USER_W;

```



```

INSERT INTO SYSTEM.TABLE_USER_LAST(USER_W_ID, LASTTIMES)
VALUES(NUM,D);

-----

COMMIT;
RETURN 1;
EXCEPTION
WHEN OTHERS THEN
    ROLLBACK TO A;
    RETURN 0;
END;
/

```

2.15 SHOP.REGISTRATION_ORDER

```

-----
--          :
--          '          '
--          ID
--          : 27/01/2007
--          :
-----

CREATE OR REPLACE FUNCTION SHOP.REGISTRATION_ORDER
(
    P_LIST          IN LONG,          --
    P_USER_ID       IN SHOP.TABLE_WAYBILL_ORDER.USER_ID%TYPE,
    P_PAYMENT_ID    IN SHOP.TABLE_WAYBILL_ORDER.PAYMENT_ID%TYPE,
    P_DELIVERY_ID   IN SHOP.TABLE_WAYBILL_ORDER.DELIVERY_ID%TYPE
) RETURN NUMBER --
IS
    E              EXCEPTION;
    PRAGMA         EXCEPTION_INIT(E, -00054);
    -----1-----
    LIST           LONG;             --
    LIST_ONE       LONG;             --
    S1             LONG;             --
    S2             LONG;             --
    NUM1           NUMBER(5);        --          " : "
    NUM2           NUMBER(5);        --          " : "
    FLAG           BOOLEAN;          --
    P_DATE         DATE;             --
    -----2-----
    SUMS           NUMBER(18);        --
    P_ID_ORDER     NUMBER(9);
    REAL_SUMM      NUMBER(18);        --
    P_ID           NUMBER(9);        --ID
    P_GOODS_ID     NUMBER(9);        --
    P_AMOUNT       NUMBER(18);        --
    K              NUMBER(18);        --
    ONE_ORDER      SHOP.TABLE_GOODS_ORDER%ROWTYPE;
    ONE_GOODS      SHOP.TABLE_INVENTORIES%ROWTYPE;
    -----
    CURSOR CUR1 IS
        SELECT * FROM SHOP.TABLE_GOODS_ORDER
            WHERE WAYBILL_ORDER_ID = P_ID_ORDER FOR UPDATE NOWAIT;
    -----
    CURSOR CUR2(PARID NUMBER) IS
        SELECT * FROM SHOP.TABLE_INVENTORIES
            WHERE GOODS_ID = PARID FOR UPDATE NOWAIT;
BEGIN
    COMMIT;
    SET TRANSACTION ISOLATION LEVEL SERIALIZABLE NAME
        'SHOP.TABLE_WAYBILL_ORDER,SHOP.TABLE_GOODS_ORDER,
        SHOP.TABLE_PACKING,SHOP.TABLE_INVENTORIES';
    SAVEPOINT A;
    -----SHOP.TABLE_WAYBILL_ORDER-----
    SELECT SYSDATE INTO P_DATE FROM DUAL; --
    INSERT INTO TABLE_WAYBILL_ORDER
        (TIMES, STATUS, USER_ID, PAYMENT_ID, DELIVERY_ID)
        VALUES (P_DATE, 0, P_USER_ID, P_PAYMENT_ID, P_DELIVERY_ID); --0 =>
    SELECT MAX(ID) INTO P_ID_ORDER
        FROM TABLE_WAYBILL_ORDER; --ID
    -----SHOP.TABLE_GOODS_ORDER-----
    LIST:=P_LIST;
    FLAG:=FALSE;
    LOOP
        NUM1:=INSTR(LIST, ';');

```

```

IF (NUM1=LENGTH(LIST)) THEN FLAG:=TRUE; END IF; --      ";" -
LIST_ONE:=SUBSTR(LIST,1,NUM1-1);
LIST:=SUBSTR(LIST,NUM1+1,LENGTH(LIST)); --
NUM2:=INSTR(LIST_ONE,':'); --
S1:=SUBSTR(LIST_ONE,1,NUM2-1);
S2:=SUBSTR(LIST_ONE,NUM2+1,LENGTH(LIST_ONE));
-----
INSERT INTO TABLE_GOODS_ORDER (GOODS_ID, AMOUNT, WAYBILL_ORDER_ID)
VALUES (TO_NUMBER(S1),TO_NUMBER(S2),P_ID_ORDER);
EXIT WHEN FLAG=TRUE;
END LOOP;
-----
OPEN CUR1;
LOOP
  FETCH CUR1 INTO ONE_ORDER;
  EXIT WHEN CUR1%NOTFOUND;
  -----
  P_ID:=ONE_ORDER.ID; --ID
  P_GOODS_ID:=ONE_ORDER.GOODS_ID; --      ,
  P_AMOUNT:=ONE_ORDER.AMOUNT; --
  SELECT SUM(AMOUNT) INTO REAL_SUMM FROM      --      ,
    TABLE_INVENTORIES
    WHERE GOODS_ID = P_GOODS_ID;
  IF (REAL_SUMM<P_AMOUNT) THEN --      ,
    BEGIN
      CLOSE CUR1;
      ROLLBACK TO A;
      RETURN P_GOODS_ID;
    END;
  END IF;
  -----
  SUMS:=0;
  OPEN CUR2(P_GOODS_ID);
  LOOP
    EXIT WHEN SUMS=P_AMOUNT; --      ,
    FETCH CUR2 INTO ONE_GOODS;
    IF (P_AMOUNT-SUMS>ONE_GOODS.AMOUNT) THEN --
      BEGIN
        SUMS:=SUMS+ONE_GOODS.AMOUNT; --
        UPDATE TABLE_INVENTORIES SET
          AMOUNT = 0
          WHERE ID=ONE_GOODS.ID;
        INSERT INTO TABLE_PACKING (WAREHOUSES_ID, AMOUNT, GOODS_ORDER_ID)
          VALUES (ONE_GOODS.WAREHOUSES_ID, ONE_GOODS.AMOUNT, P_ID);
      END;
    ELSE --      (
      BEGIN
        K:=P_AMOUNT-SUMS;
        SUMS:=P_AMOUNT; --
        UPDATE TABLE_INVENTORIES SET
          AMOUNT = AMOUNT - K
          WHERE ID=ONE_GOODS.ID;
        INSERT INTO TABLE_PACKING (WAREHOUSES_ID, AMOUNT, GOODS_ORDER_ID)
          VALUES (ONE_GOODS.WAREHOUSES_ID, K, P_ID);
      END;
    END IF;
  END LOOP;
  CLOSE CUR2;
  -----
END LOOP;
CLOSE CUR1;
-----
COMMIT;
RETURN 0; --
EXCEPTION
  WHEN E THEN --      -
    ROLLBACK TO A; --
    RETURN -1;
  WHEN OTHERS THEN
    ROLLBACK TO A;
    RETURN -2; --
END;
/

```

3

Shop

```
GRANT EXECUTE ON SHOP.GET_INFO_CLIENT_GOODS TO SHOPUSER;
GRANT EXECUTE ON SHOP.AUTH_USER TO SHOPUSER;
GRANT EXECUTE ON SHOP.REGISTRATION TO SHOPUSER;
GRANT EXECUTE ON SHOP.FIND_STRING TO SHOPUSER;
GRANT EXECUTE ON SHOP.REGISTRATION_ORDER TO SHOPUSER;
GRANT SELECT ON SHOP.TABLE_CATEGORY TO SHOPUSER;
GRANT SELECT ON SHOP.TABLE_GOODS TO SHOPUSER;
GRANT SELECT ON SHOP.TABLE_DELIVERY TO SHOPUSER;
GRANT SELECT ON SHOP.TABLE_PAYMENT TO SHOPUSER;
GRANT SELECT ON SHOP.TABLE_USER TO SHOPUSER;
GRANT SELECT ON SHOP.TABLE_ADDRESS_U TO SHOPUSER;
GRANT SELECT ON SHOP.TABLE_GOODS_ORDER TO SHOPUSER;
GRANT SELECT ON SHOP.TABLE_WAYBILL_ORDER TO SHOPUSER;
GRANT SELECT ON SHOP.TABLE_IMAGE TO SHOPUSER;
GRANT SELECT ON SHOP.TABLE_IMAGE TO SHOPMIN;
```

4

SYSTEM

```
GRANT EXECUTE ON SYSTEM.AUTH TO SHOP;
```