Database System for Platinum Online Clothing Shop

1. Project Description

The title of the project is “Database System for Platinum Online Clothing Shop”. The main task of the system is to record the sales list of Platinum Online Clothing Shop daily. However, the system also stores the information of the salesperson, the details of the selling items and the details of the orders made by the customers. Moreover, the system also keeps the information of customers and delivery.

The main purpose of the project is to computerize a part of the work of a chief executive officer of the Platinum Online Clothing Shop so that the chief executive officer is able to eliminate paperwork. The system helps the chief executive officer avoid mistakes that are likely to be made and errors that are likely to be come out.

The system has many benefits for the chief executive officer and other staffs of the Platinum Online Clothing Shop. One of the best advantages is timesaving. The chief executive officer is able to save much time by using the system instead of using traditional bookkeeping as the system is efficient. One amazing advantages of the system is that the chief executive officer and other staffs are able to access the particular information anytime. However, they need to have appropriate technical knowledge in order to use the system.

One of the advantages of the system is the information is able to access easily at any time. Moreover, the system is efficient and time-saving. The system helps eliminate paperwork, avoid mistakes and errors. As a result of its advantages, it can make the chief executive officer of the Platinum Online Clothing Shop use the system.

1. Objectives of the Project

The objectives of the project are as follows.

* To help the chief executive officer in bookkeeping for the Platinum Online Clothing Shop
* To keep the record of the sales list daily
* To store the details of the items available and the information of the salesperson
* To eliminate paperwork in bookkeeping of the sales list for the chief executive officer of the Platinum Online Clothing Shop
* To avoid time-consuming
* To eliminate mistakes that are likely to be made in bookkeeping
* To eliminate errors that are likely to be come out in bookkeeping

1. Implementation of the Project

This project is implemented by MySQL 5.5.

Implementation includes

* Creating Tables
* Inserting Data
  1. Creating Tables

mysql> CREATE TABLE BRAND(

-> BRAND\_ID VARCHAR(10),

-> BRAND\_NAME VARCHAR(20),

-> PRIMARY KEY(BRAND\_ID));

mysql> CREATE TABLE ITEM\_TYPE(

-> TYPE\_ID VARCHAR(10),

-> TYPE VARCHAR(15),

-> PRIMARY KEY(TYPE\_ID));

mysql> CREATE TABLE ITEM(

-> ITEM\_ID VARCHAR(10),

-> TYPE\_ID VARCHAR(10),

-> TITLE VARCHAR(20),

-> BRAND\_ID VARCHAR(10),

-> PRIMARY KEY(ITEM\_ID),

-> FOREIGN KEY(BRAND\_ID) REFERENCES BRAND(BRAND\_ID)

-> ON DELETE RESTRICT ON UPDATE RESTRICT,

-> FOREIGN KEY(TYPE\_ID) REFERENCES ITEM\_TYPE(TYPE\_ID)

-> ON DELETE RESTRICT ON UPDATE RESTRICT);

mysql> CREATE TABLE ITEM\_DETAILS(

-> ITEM\_DETAILS\_ID VARCHAR(15),

-> ITEM\_ID VARCHAR(10),

-> SIZE VARCHAR(5),

-> COLOR VARCHAR(15),

-> AVAILABLE\_QTY INTEGER,

-> PURCHASE\_COST DOUBLE,

-> SELLING\_PRICE DOUBLE,

-> PRIMARY KEY(ITEM\_DETAILS\_ID),

-> FOREIGN KEY(ITEM\_ID) REFERENCES ITEM(ITEM\_ID)

-> ON DELETE CASCADE ON UPDATE CASCADE);

mysql> CREATE TABLE STAFF(

-> STAFF\_ID VARCHAR(5),

-> NRC VARCHAR(20),

-> NAME VARCHAR(30),

-> PHONE VARCHAR(15),

-> ADDRESS VARCHAR(30),

-> POSITION VARCHAR(20),

-> SALARY DOUBLE,

-> PRIMARY KEY(STAFF\_ID));

mysql> CREATE TABLE CUSTOMER(

-> CUSTOMER\_ID VARCHAR(10),

-> PHONE VARCHAR(15),

-> NAME VARCHAR(30),

-> ADDRESS VARCHAR(30),

-> PRIMARY KEY(CUSTOMER\_ID));

mysql> CREATE TABLE ORDER\_ITEM(

-> ORDER\_ITEM\_ID VARCHAR(10),

-> ORDER\_ID VARCHAR(10),

-> ITEM\_DETAILS\_ID VARCHAR(10),

-> QTY INTEGER,

-> PRIMARY KEY(ORDER\_ITEM\_ID),

-> FOREIGN KEY(ITEM\_DETAILS\_ID) REFERENCES

ITEM\_DETAILS(ITEM\_DETAILS\_ID)

-> ON DELETE RESTRICT ON UPDATE RESTRICT,

-> FOREIGN KEY(ORDER\_ID) REFERENCES ORDER\_TABLE(ORDER\_ID)

-> ON DELETE CASCADE ON UPDATE RESTRICT);

;

mysql> CREATE TABLE ORDER\_TABLE(

-> ORDER\_ID VARCHAR(10),

-> CUSTOMER\_ID VARCHAR(10),

-> DATE DATE,

-> PRIMARY KEY(ORDER\_ID),

-> FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMER(CUSTOMER\_ID)

-> ON DELETE RESTRICT ON UPDATE RESTRICT);

mysql> CREATE TABLE DELIVERY(

-> STAFF\_ID VARCHAR(5),

-> ORDER\_ID VARCHAR(10),

-> DATE DATE,

-> PRIMARY KEY(STAFF\_ID,ORDER\_ID,DATE),

-> FOREIGN KEY(ORDER\_ID) REFERENCES ORDER\_TABLE(ORDER\_ID)

-> ON DELETE CASCADE ON UPDATE RESTRICT,

-> FOREIGN KEY(STAFF\_ID) REFERENCES

STAFF(STAFF\_ID)

-> ON DELETE RESTRICT ON UPDATE RESTRICT);

mysql> CREATE TABLE SALES(

-> SALES\_ID VARCHAR(30),

-> STAFF\_ID VARCHAR(5),

-> VOUCHER\_ID VARCHAR(30),

-> ITEM\_DETAILS\_ID VARCHAR(15),

-> QTY INTEGER,

-> DATE DATE,

-> PRIMARY KEY(SALES\_ID),

-> FOREIGN KEY(STAFF\_ID) REFERENCES STAFF(STAFF\_ID)

-> ON DELETE RESTRICT ON UPDATE RESTRICT,

-> FOREIGN KEY(ITEM\_DETAILS\_ID) REFERENCES

ITEM\_DETAILS(ITEM\_DETAILS\_ID)

-> ON DELETE RESTRICT ON UPDATE RESTRICT);

mysql> CREATE TABLE SOLD\_ITEMS(

-> ITEM\_DETAILS\_ID VARCHAR(15),

-> QTY INT,

-> ORDER\_ID VARCHAR(10));

* 1. Inserting Data

The data for Item table is inserted as follows:

mysql> INSERT INTO BRAND VALUES

-> ('b0001','AA');

+----------+------------+

| BRAND\_ID | BRAND\_NAME |

+----------+------------+

| b0001 | AA |

| b0002 | Volume |

| b0003 | J2J |

| b0004 | Crown |

| b0005 | Nobody |

| b0006 | Apple |

+----------+------------+

mysql> INSERT INTO ITEM\_TYPE VALUES

-> ('type0001','jeans');

+----------+---------+

| TYPE\_ID | TYPE |

+----------+---------+

| type0001 | jeans |

| type0002 | jackets |

+----------+---------+

mysql> INSERT INTO ITEM VALUES

-> ('item0001','type0001','AA jeans','b0001');

+----------+----------+--------------+----------+

| ITEM\_ID | TYPE\_ID | TITLE | BRAND\_ID |

+----------+----------+--------------+----------+

| item0001 | type0001 | AA jeans | b0001 |

| item0002 | type0001 | Volume jeans | b0002 |

| item0003 | type0001 | J2J jeans | b0003 |

| item0004 | type0001 | Crown jeans | b0004 |

| item0005 | type0001 | Nobody jeans | b0005 |

| item0006 | type0002 | Apple jacket | b0006 |

| item0007 | type0002 | J2J jacket | b0003 |

+----------+----------+--------------+----------+

The data for Item\_details table is inserted as follows:

mysql> INSERT INTO ITEM\_DETAILS VALUES

-> ('id0001','item0001','s','blue',3,5000,7000);

+-----------------+----------+------+-------+---------------+---------------+---------------+

| ITEM\_DETAILS\_ID | ITEM\_ID | SIZE | COLOR | AVAILABLE\_QTY | PURCHASE\_COST | SELLING\_PRICE |

+-----------------+----------+------+-------+---------------+---------------+---------------+

| id0001 | item0001 | s | blue | 3 | 5000 | 7000 |

| id0002 | item0001 | m | blue | 2 | 5000 | 7000 |

| id0003 | item0001 | m | black | 4 | 5000 | 7000 |

| id0004 | item0002 | s | black | 5 | 5500 | 7000 |

| id0005 | item0003 | s | black | 5 | 6000 | 7500 |

| id0006 | item0003 | m | black | 4 | 6000 | 7500 |

| id0007 | item0003 | l | black | 3 | 6500 | 8000 |

| id0008 | item0004 | l | black | 3 | 6000 | 8000 |

| id0009 | item0005 | m | white | 3 | 6500 | 8000 |

| id0010 | item0006 | free | red | 3 | 5000 | 7000 |

| id0011 | item0006 | free | black | 3 | 5000 | 7000 |

| id0012 | item0007 | free | white | 2 | 4000 | 6500 |

+-----------------+----------+------+-------+---------------+---------------+---------------+

The data for Staff table is inserted as follows:

mysql> INSERT INTO STAFF VALUES

-> ('s01','14/PaThaNa(C)252825','Kyaw Lin','09691354898','No.111 KonThal Street,Pathein','salesperson',80000);

+----------+---------------------+------------+-------------+-------------------------------+------------------+--------+

| STAFF\_ID | NRC | NAME | PHONE | ADDRESS | POSITION | SALARY |

+----------+---------------------+------------+-------------+-------------------------------+------------------+--------+

| s01 | 14/PaThaNa(C)252825 | Kyaw Lin | 09691354898 | No.111 KonThal Street,Pathein | delievery person | 80000 |

| s02 | 14/PaThaNa(C)256464 | Myat Thu | 09402154644 | No.2 Akyaw Street,Pathein | delievery person | 80000 |

| s03 | 14/LaPaTa(C)005561 | Kay Zin | 09254588604 | No.10 Pat Street,Pathein | salesperson | 60000 |

| s04 | 14/LaPaTa(C)002256 | Thida Aye | 09454396054 | No.10 Pat Street,Pathein | salesperson | 60000 |

| s05 | 14/PaThaNa(C)252525 | Khaing Lay | 09691345747 | No.6, Min Gyi Street,Pathein | cashier | 100000 |

+----------+---------------------+------------+-------------+-------------------------------+------------------+--------+

The data for Customer table is inserted as follows:

mysql> INSERT INTO CUSTOMER VALUES

-> ('c0001','09254788159','Daw Yadana','No.5, 28 Pagodas Road');

+-------------+-------------+-------------------+--------------------------------+

| CUSTOMER\_ID | PHONE | NAME | ADDRESS |

+-------------+-------------+-------------------+--------------------------------+

| c0001 | 09254788159 | Daw Yadana | No.5, 28 Pagodas Road |

| c0002 | 09691253698 | Thuya Aung Htet | No.24,Ngawon Kyoon Tha(1)Stree |

| c0003 | 09402158766 | Lamin Htain Htain | No.100,Kan Na Street |

| c0004 | 09794529866 | Hay Mar | Doctors Hall, in front of BEHS |

| c0005 | 09976889542 | Kyaw Min | No.40,Sayyoetan Street,Pathein |

+-------------+-------------+-------------------+--------------------------------+

The data for Order\_table table is inserted as follows:

mysql> INSERT INTO ORDER\_TABLE VALUES

-> ('order0001','c0001', '2019-2-1');

+-----------+-------------+------------+

| ORDER\_ID | CUSTOMER\_ID | DATE |

+-----------+-------------+------------+

| order0001 | c0001 | 2019-02-01 |

| order0002 | c0001 | 2019-02-01 |

| order0003 | c0002 | 2019-02-01 |

| order0004 | c0003 | 2019-02-02 |

| order0005 | c0004 | 2019-02-02 |

| order0006 | c0005 | 2019-02-03 |

+-----------+-------------+------------+

The data for Order\_item table is inserted as follows:

mysql> INSERT INTO ORDER\_ITEM VALUES

-> ('oid0001','order0001','id0007',1);

+---------------+-----------+-----------------+------+

| ORDER\_ITEM\_ID | ORDER\_ID | ITEM\_DETAILS\_ID | QTY |

+---------------+-----------+-----------------+------+

| oid0001 | order0001 | id0007 | 1 |

| oid0002 | order0001 | id0001 | 2 |

| oid0003 | order0002 | id0001 | 1 |

| oid0004 | order0003 | id0006 | 2 |

| oid0005 | order0004 | id0005 | 1 |

| oid0006 | order0005 | id0002 | 2 |

| oid0007 | order0006 | id0001 | 2 |

| oid0008 | order0006 | id0002 | 1 |

+---------------+-----------+-----------------+------+

The data for Delivery table is inserted as follows:

mysql> INSERT INTO DELIVERY VALUES

-> ('s01','order0001','2019-2-4');

+----------------+-----------+------------+

| STAFF\_ID | ORDER\_ID | DATE |

+----------------+-----------+------------+

| s01 | order0001 | 2019-02-04 |

| s02 | order0002 | 2019-02-04 |

+----------------+-----------+------------+

The data for Sales table is inserted as follows:

mysql> INSERT INTO SALES VALUES

-> ('sale00001','s04','v00001','id0010','2','2019-2-18');

+-----------+----------+------------+-----------------+------+------------+

| SALES\_ID | STAFF\_ID | VOUCHER\_ID | ITEM\_DETAILS\_ID | qty | DATE |

+-----------+----------+------------+-----------------+------+------------+

| sale00001 | s04 | v00001 | id0010 | 2 | 2019-02-18 |

| sale00002 | s04 | v00001 | id0009 | 1 | 2019-02-18 |

| sale00003 | s04 | v00001 | id0005 | 1 | 2019-02-18 |

| sale00004 | s04 | v00002 | id0004 | 1 | 2019-02-19 |

| sale00005 | s03 | v00003 | id0007 | 1 | 2019-02-19 |

| sale00006 | s03 | v00003 | id0008 | 1 | 2019-02-19 |

+-----------+----------+------------+-----------------+------+------------+

* 1. Table Description

mysql> DESC BRAND;

+------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+-------------+------+-----+---------+-------+

| BRAND\_ID | varchar(10) | NO | PRI | NULL | |

| BRAND\_NAME | varchar(20) | YES | | NULL | |

+------------+-------------+------+-----+---------+-------+

mysql> DESC ITEM\_TYPE;

+---------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------+-------------+------+-----+---------+-------+

| TYPE\_ID | varchar(10) | NO | PRI | NULL | |

| TYPE | varchar(15) | YES | | NULL | |

+---------+-------------+------+-----+---------+-------+

mysql> DESC ITEM;

+----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+-------------+------+-----+---------+-------+

| ITEM\_ID | varchar(10) | NO | PRI | NULL | |

| TYPE\_ID | varchar(10) | YES | MUL | NULL | |

| TITLE | varchar(20) | YES | | NULL | |

| BRAND\_ID | varchar(10) | YES | MUL | NULL | |

+----------+-------------+------+-----+---------+-------+

mysql> DESC ITEM\_DETAILS;

+-----------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+-------------+------+-----+---------+-------+

| ITEM\_DETAILS\_ID | varchar(15) | NO | PRI | NULL | |

| ITEM\_ID | varchar(10) | YES | MUL | NULL | |

| SIZE | varchar(5) | YES | | NULL | |

| COLOR | varchar(15) | YES | | NULL | |

| AVAILABLE\_QTY | int(11) | YES | | NULL | |

| PURCHASE\_COST | double | YES | | NULL | |

| SELLING\_PRICE | double | YES | | NULL | |

+-----------------+-------------+------+-----+---------+-------+

mysql> DESC STAFF;

+----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+-------------+------+-----+---------+-------+

| STAFF\_ID | varchar(5) | NO | PRI | NULL | |

| NRC | varchar(20) | YES | | NULL | |

| NAME | varchar(30) | YES | | NULL | |

| PHONE | varchar(15) | YES | | NULL | |

| ADDRESS | varchar(30) | YES | | NULL | |

| POSITION | varchar(20) | YES | | NULL | |

| SALARY | double | YES | | NULL | |

+----------+-------------+------+-----+---------+-------+

mysql> DESC CUSTOMER;

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| CUSTOMER\_ID | varchar(10) | NO | PRI | NULL | |

| PHONE | varchar(15) | YES | | NULL | |

| NAME | varchar(30) | YES | | NULL | |

| ADDRESS | varchar(30) | YES | | NULL | |

+-------------+-------------+------+-----+---------+-------+

mysql> DESC ORDER\_TABLE;

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| ORDER\_ID | varchar(10) | NO | PRI | NULL | |

| CUSTOMER\_ID | varchar(10) | YES | MUL | NULL | |

| DATE | date | YES | | NULL | |

+-------------+-------------+------+-----+---------+-------+

mysql> DESC ORDER\_ITEM;

+-----------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+-------------+------+-----+---------+-------+

| ORDER\_ITEM\_ID | varchar(10) | NO | PRI | NULL | |

| ORDER\_ID | varchar(10) | YES | MUL | NULL | |

| ITEM\_DETAILS\_ID | varchar(10) | YES | MUL | NULL | |

| QTY | int(11) | YES | | NULL | |

+-----------------+-------------+------+-----+---------+-------+

mysql> DESC DELIVERY;

+----------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+-------------+------+-----+---------+-------+

| STAFF\_ID | varchar(5) | NO | PRI | NULL | |

| ORDER\_ID | varchar(10) | NO | PRI | NULL | |

| DATE | date | NO | PRI | NULL | |

+----------------+-------------+------+-----+---------+-------+

mysql> DESC SALES;

+-----------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+-------------+------+-----+---------+-------+

| SALES\_ID | varchar(30) | NO | PRI | NULL | |

| STAFF\_ID | varchar(5) | YES | MUL | NULL | |

| VOUCHER\_ID | varchar(30) | YES | | NULL | |

| ITEM\_DETAILS\_ID | varchar(15) | YES | MUL | NULL | |

| QTY | int(11) | YES | | NULL | |

| DATE | date | YES | | NULL | |

+-----------------+-------------+------+-----+---------+-------+

mysql> DESC SOLD\_ITEMS;

+-----------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+-------------+------+-----+---------+-------+

| ITEM\_DETAILS\_ID | varchar(15) | YES | | NULL | |

| QTY | int(11) | YES | | NULL | |

| ORDER\_ID | varchar(10) | YES | | NULL | |

+-----------------+-------------+------+-----+---------+-------+

* 1. Integrity Constraint

The term integrity refers to the accuracy or correctness of data in the database.

1. The only legal staff numbers are ones that can be represented by a character string of at least three characters, of which the first is an "S" and the following two characters denote integer in the range 0 to 9.

TYPE STAFF\_ID POSSREP (CHAR) CONSTRAINT

SUBSTR ( THE\_STAFF\_ID (STAFF\_ID), 1,1)='S' AND

CAST\_AS\_INTEGER ( SUBSTR (THE\_STAFF\_ID (STAFF\_ID),2))>=0 AND

CAST\_AS\_INTEGER ( SUBSTR (THE\_STAFF\_ID (STAFF\_ID),2))<=99;

2. The number of type\_id in type table is not greater than the number of type\_id in item table.

TYPE {TYPE\_ID} < ITEM{TYPE\_ID};

3. No purchasing price is greater than selling price.

IS\_EMPTY(ITEM\_DETAILS WHERE PURCHASE\_COST>SELLING\_COST);

* 1. Security Constraint

Security refers to the protection of data against unauthorized users.

There are two types of securities.

* Discretionary control
* Mandatory control

**Syntax:**

AUTHORITY <authority name>

GRANT <privilege commalist>

ON <relvar name>

TO <user ID commalist>

**Create Users**

mysql> CREATE USER 'ceo'@localhost identified by 'owner';

mysql> CREATE USER 'salesperson1'@localhost identified by 'kayzin';

mysql> CREATE USER 'salesperson2'@localhost identified by 'thidaaye';

mysql> CREATE USER 'deliveryperson1'@localhost identified by 'kyawlin';

mysql> CREATE USER 'deliveryperson2'@localhost identified by 'myatthu';

mysql> CREATE USER 'cashier'@localhost identified by 'khainglay';

5.1 Security Constraints

1. CEO manages the entire database and grants other users.

mysql> GRANT ALL

-> ON PLATINUM\_SHOP.\*

-> TO ceo@localhost WITH GRANT OPTION;

1. The cashier manages sales data.

mysql> GRANT ALL

-> ON SALES

-> TO cashier@localhost;

1. Both cashier and salespersons retrieves data associated with item except purchasing price.

mysql> GRANT SELECT

-> ON BRAND

-> TO

cashier@localhost,salesperson1@localhost,salesperson2@localhost;

mysql> GRANT SELECT

-> ON ITEM\_TYPE

-> TO cashier@localhost,salesperson1@localhost,salesperson2@localhost;

mysql> GRANT SELECT

-> ON ITEM

-> TO cashier@localhost,salesperson1@localhost,salesperson2@localhost;

mysql> GRANT

SELECT(ITEM\_DETAILS\_ID,ITEM\_ID,SIZE,COLOR,SELLING\_PRICE)

-> ON ITEM\_DETAILS

-> TO cashier@localhost,salesperson1@localhost,salesperson2@localhost;

1. All staff retrieves their data.

mysql> GRANT SELECT

-> ON STAFF

-> TO cashier@localhost,salesperson1@localhost,salesperson2@localhost,deliveryperson1@localhost,deliveryperson2@localhost;

1. The salespersons manage order data, customer data and retrieve delivery information and order that are left to deliver.

**Creating required views**

mysql> CREATE VIEW LEFT\_ORDER AS(

-> SELECT ORDER\_ID

-> FROM ORDER\_TABLE

-> WHERE ORDER\_ID NOT IN(

-> SELECT ORDER\_ID

-> FROM DELIEVERY));

**Granting views to user**

mysql> GRANT ALL

-> ON ORDER\_TABLE

-> TO salesperson1@localhost,salesperson2@localhost;

mysql> GRANT ALL

-> ON CUSTOMER

-> TO salesperson1@localhost,salesperson2@localhost;

mysql> GRANT SELECT

-> ON DELIEVERY

-> TO salesperson1@localhost,salesperson2@localhost;

mysql> GRANT SELECT

-> ON LEFT\_ORDER

-> TO salesperson1@localhost,salesperson2@localhost;

1. The delivery-persons retrieve order data, customer data and manage delivery information.

mysql> GRANT SELECT

-> ON ORDER\_TABLE

-> TO deliveryperson1@localhost,deliveryperson2@localhost;

mysql> GRANT SELECT

-> ON CUSTOMER

-> TO delieveryperson1@localhost,delieveryperson2@localhost;

mysql> GRANT ALL

-> ON DELIEVERY

-> TO delieveryperson1@localhost,delieveryperson2@localhost;

1. The salespersons retrieve sold items and left items details.

**Creating required views**

mysql> CREATE VIEW AVAILABLE\_ITEMS AS(

-> SELECT ITEM\_DETAILS.ITEM\_DETAILS\_ID,AVAILABLE\_QTY-TOTAL\_Q AS

LEFT\_QTY

-> FROM SOLD\_OUT,ITEM\_DETAILS

-> WHERE SOLD\_OUT.ITEM\_DETAILS\_ID=ITEM\_DETAILS.ITEM\_DETAILS\_ID);

mysql> CREATE VIEW SOLD\_ITEMS AS(

-> SELECT ITEM\_DETAILS\_ID,SUM(QTY) AS TOTAL\_Q

-> FROM SALES

-> GROUP BY ITEM\_DETAILS\_ID);

mysql> CREATE VIEW SOLD\_OUT AS(

-> SELECT ITEM\_DETAILS\_ID,SUM(QTY) AS TOTAL\_Q

-> FROM SOLD\_ITEMS

-> GROUP BY ITEM\_DETAILS\_ID);

**Granting views to user**

mysql> GRANT SELECT

-> ON SOLD\_OUT

-> TO salesperson1@localhost,salesperson2@localhost;

mysql> GRANT SELECT

-> ON AVAILABLE\_ITEMS

-> TO salesperson1@localhost,salesperson2@localhost;

* 1. Triggers Procedures

A trigger is special procedural program that is automatically executed in response to specified events such as INSERT, UPDATE and DELETE on a particular table in a database. The trigger can be used for database event logging or data modification audit purposes and it can be set to activate either BEFORE or AFTER the triggering statement.

Trigger Event and Keyword

|  |  |
| --- | --- |
| Event | Keyword |
| INSERT | NEW |
| UPDATE | NEW,OLD |
| DELETE | OLD |

Trigger Timing with Event Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Timing | INSERT | UPDATE | DELETE |
| BEFORE | NEW(editable) | NEW(editable),OLD | OLD |
| AFTER | NEW | NEW,OLD | OLD |

1. The chief executive officer and salespersons require the details of the items that are already sold out or ordered. Therefore, it is designed to record the sold or ordered items every time a row is inserted into sales table or order table.

mysql> DELIMITER $$

mysql> CREATE TRIGGER SOLD\_OUT\_ORDER\_ITEM

-> AFTER INSERT

-> ON ORDER\_ITEM

-> FOR EACH ROW

-> BEGIN

-> INSERT INTO SOLD\_ITEMS

-> SET ITEM\_DETAILS\_ID=NEW.ITEM\_DETAILS\_ID,

-> QTY=NEW.QTY,

-> ORDER\_ID=NEW.ORDER\_ID;

-> END$$

mysql> CREATE TRIGGER SOLD\_OUT\_SALES

-> AFTER INSERT

-> ON SALES

-> FOR EACH ROW

-> BEGIN

-> INSERT INTO SOLD\_ITEMS

-> SET ITEM\_DETAILS\_ID=NEW.ITEM\_DETAILS\_ID,

-> QTY=NEW.QTY;

-> END$$

mysql> SELECT \* from SOLD\_ITEMS;

+-----------------+------+-----------+

| ITEM\_DETAILS\_ID | QTY | ORDER\_ID |

+-----------------+------+-----------+

| id0007 | 1 | order0001 |

| id0001 | 2 | order0001 |

| id0001 | 1 | order0002 |

| id0006 | 2 | order0003 |

| id0005 | 1 | order0004 |

| id0002 | 2 | order0005 |

| id0003 | 2 | order0006 |

| id0002 | 1 | order0006 |

| id0010 | 2 | NULL |

| id0009 | 1 | NULL |

| id0005 | 1 | NULL |

| id0004 | 1 | NULL |

| id0007 | 1 | NULL |

| id0008 | 1 | NULL |

+-----------------+------+-----------+

1. As the order is cancelable, the ordered items that are canceled will be back into the available items.

mysql> DELIMITER $$

mysql> CREATE TRIGGER ORDER\_CANCEL

-> AFTER DELETE

-> ON ORDER\_TABLE

-> FOR EACH ROW

-> BEGIN

-> DELETE FROM SOLD\_ITEMS WHERE

SOLD\_ITEMS.ORDER\_ID=OLD.ORDER\_ID;

-> END$$

mysql> CREATE TRIGGER ORDER\_CANCEL2

-> BEFORE DELETE

-> ON ORDER\_TABLE

-> FOR EACH ROW

-> BEGIN

-> DELETE FROM ORDER\_ITEM WHERE

ORDER\_ITEM.ORDER\_ID=OLD.ORDER\_ID;

-> END$$

1. A trigger procedure is taken place when new items are arrived in order to update the available items.

mysql> DELIMITER $$

mysql> CREATE TRIGGER ARRIVAL

-> AFTER INSERT

-> ON ITEM\_DETAILS

-> FOR EACH ROW

-> BEGIN

-> INSERT INTO SOLD\_ITEMS

-> SET ITEM\_DETAILS\_ID=NEW.ITEM\_DETAILS\_ID,

-> QTY=0;

-> END$$

mysql> DELIMITER ;

mysql> INSERT INTO ITEM\_DETAILS VALUES

-> ('id0011','item0006','free','black',3,5000,7000),

-> ('id0012','item0007','free','white',2,4000,6500);

mysql> SELECT \* FROM SOLD\_OUT;

+-----------------+---------+

| ITEM\_DETAILS\_ID | TOTAL\_Q |

+-----------------+---------+

| id0001 | 3 |

| id0002 | 3 |

| id0003 | 2 |

| id0004 | 1 |

| id0005 | 2 |

| id0006 | 2 |

| id0007 | 2 |

| id0008 | 1 |

| id0009 | 1 |

| id0010 | 2 |

| id0011 | 0 |

| id0012 | 0 |

+-----------------+---------+

mysql> SELECT \* FROM SOLD\_ITEMS;

+-----------------+------+

| ITEM\_DETAILS\_ID | QTY |

+-----------------+------+

| id0010 | 2 |

| id0009 | 1 |

| id0005 | 1 |

| id0004 | 1 |

| id0007 | 1 |

| id0008 | 1 |

| id0007 | 1 |

| id0001 | 2 |

| id0001 | 1 |

| id0006 | 2 |

| id0005 | 1 |

| id0002 | 2 |

| id0003 | 2 |

| id0002 | 1 |

| id0011 | 0 |

| id0012 | 0 |

+-----------------+------+

1. Conclusion

This database system is for Platinum Clothing Shop in Pathein. This database system can be used in any other appropriate application like Web Applications, Java Applications, C++ Applications, C# Applications, Android Applications and so on.

1. References

7th edition of “An introduction to Database Systems” by C.J.Date and the Internet.