National School of Business Management

CS105.3 - Database Management System

Group Assignment 16.2

(Group of 4 students)

Batch : MIS/CS/SE/PLY 16.2

Declaration

Student ID	Student Name	Signature
BSC-PLY-COM-16.2-**244**	Renuka, P.H. Heshan	
BSC-PLY-COM-16.2-**218**	Pramuditha, P.G. shan	
BSC-PLY-COM-16.2-**157**	Lekamge, W.L. Dhananja Y.S.	
BSC-PLY-COM-16.2-**304**	Wickramasinghe, W.A. Chethana D.	

Date of Submission : ……..

Table of contents

01.Scenario

National School of Business Management

CS105.3 Database Management Systems

Group Assignment (group of 4 students)

Batch: MIS/CS/SE/PLY 16.2

Hand out date: week of 12th December 2016

This assessment contributes to 25% of the module mark

You are required to analyse the scenario given below and design and develop a DB solution to answer the queries of interest.

Scenario : Online Order Processing System

A merchandiser maintains a catalog of products online. Customers can browse throw the catalog and place orders for the items. Website needs customers to be registered before placing any orders. Submitted orders are processed daily which involves updating the merchandiser's items inventory and raising the invoices for the customers. Once the customers make the payments, shipment details (delivery notes) are generated. Successful delivery of goods updates the shipment details and completes the transaction.

Incase ordered items (Customer orders) are not available in the inventory, merchandiser raises a back order (Supplier order) for its registered set of suppliers to get down the unavailable items to merchandizer warehouse. Payments are settled for the registered suppliers on a monthly basis for the goods they supplied during the month. Once the goods are received from the suppliers, customer order processing proceeds as explained before.
A system is required to support the above operations of the merchandiser and generate necessary reports for monitoring and management of the business operations.

The system should facilitate the following reports/queries or SQLs.

a) Items report with the details of all items in the catalog.
b) Query to list down the items that has been orders during the last 3 months.
c) Query to list down the **details of the items** that has been orders during the last 3 months.
d) Query to list down the items that has **not been orders** during the last 3 months.
e) Daily report to list down the item details of customer orders placed during the day.
f) Report to List down all items in the catalog but not available in the stores.
g) Query to list down all items in the pending (not processed yet) customer orders but not available in the stores.
h) Query to update the Customer Order 'CO0012' to increase the item quantity of 'IT004' to 16.
i) Write down the SQLs required to enter a new customer order (first insert the header record and then insert the item detail records).
j) Write SQLS to delete the Customer order 'CO0023'.
Please clearly specify any assumptions you make.

Deliverables:

1. Report including,
- The problem scenario with any assumptions you made
- ER diagram
- Relational mapping
- Normalized tables
- Conclusion with a justification of your solution and limitations.

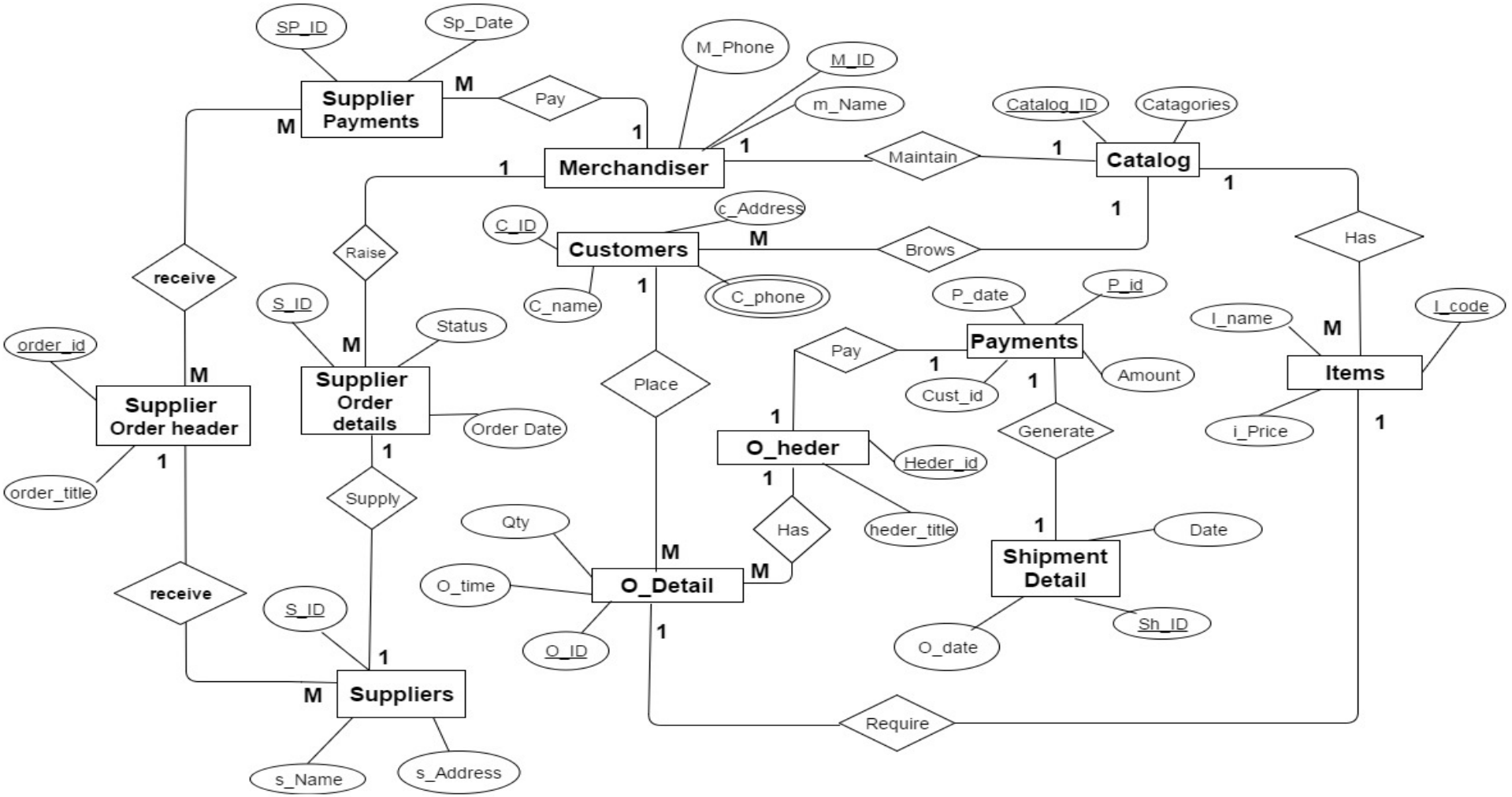
2. CD that includes
- a soft copy of your report
- an SQL script to generate the database with any constraints
- An SQL script to insert sample data for the tables
- An SQL script carrying the SQLs to generate all the required reports

Marking Criteria:

Following items carry a group mark.
1. Quality of the documentation – 10 Marks
2. Introduction to the problem and assumptions – 10 marks

Following items are assessed on individual student basis
3. ER Diagram - 20 Marks
4. Normalization - 10 Marks
5. SQLs to create the database and constraints - 20 Marks
6. SQLs to load the sample data – 10 Marks
7. SQLs to generate the reports – 20 Mark

2. Conceptual Design- ER Diagram



3.Relational Mapping

Supplier_payment (SS_id , Date , M_id)

Merchandiser (m_id , M_name , m_place)

Catalog (catalog_id , catagories , m_id)

Customer (c_id , c_name, c_address , catlog_id)

Payments (p_id , p_date , amount , cust_id)

Items (i_code , i_name ,i_ price , cost_id)

Supplier order detail (s_id , status , order_date , m_id)

Supplier order heder (order_id , order_title , s_id)

Order header (heder_id , heder_title , p_id)

Order detail (o_id , o_time , qty , i_code , c_id , h_id)

Suppliers (s_id , s_name , s_address , s_id , order_id)

Shipment detail (sh_id , date , o_date , p_id)

C_phone (cust_id , c_phone)

Receive (so_id , order_id)

4.Assumptions

General assumptions

- Assuming that one supplier can supply one order detail list,
- Assuming that customer can browse only one catalog,
- Assuming that one order header can have more than one details,
- Assuming that supplies can receive only one order header,
- Assuming that merchandiser only can maintain one catalog,

5.Normalization

- Supplier_payment (SS_id , Date , M_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key 'SS_id'.

  * Therefore the relation 'Supplier_payment' is in 3NF.

- Merchandiser (m_id , M_name , m_place)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key 'm_id'.

  * Therefore the relation 'Merchandiser' is in 3NF.

- Catalog (<u>catalog_id</u> , catagories , m_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key ‘<u>catalog_id</u>’.

  * Therefore the relation ‘Catalog’ is in 3NF.

- Customer (<u>c_id</u> , c_name, c_address , catlog_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key ‘<u>c_id</u>’.

  * Therefore the relation ‘Customer’ is in 3NF.

- Payments (<u>p_id</u> , p_date , amount , cust_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key ‘<u>p_id</u>.

  * Therefore the relation ‘Payments’ is in 3NF.

- Items (i_code , i_name ,i_ price , cost_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key 'i_code'.

  * Therefore the relation 'Items' is in 3NF.

- Supplier order detail (s_id , status , order_date , m_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key 's_id'.

  * Therefore the relation 'Supplier order detail' is in 3NF.

- Supplier order heder (order_id , order_title , s_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key 'order_id'.

  * Therefore the relation 'Supplier order heder' is in 3NF.

- Order header (heder_id , heder_title , p_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key 'heder_id'.

  * Therefore the relation 'Order header' is in 3NF.

- Order detail (o_id , o_time , qty , i_code , c_id , h_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key 'o_id'.

  * Therefore the relation 'Order detail' is in 3NF.

- Suppliers (s_id , s_name , s_address , s_id , order_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key 's_id'.

  * Therefore the relation 'Suppliers' is in 3NF.

- Shipment detail (sh_id , date , o_date , p_id)

  * Every attribute in this relation is atomic.

  * Every non-primary key attribute is fully dependent on the primary key 'sh_id'.

  * Therefore the relation 'Shipment detail ' is in 3NF.

- C_phone (cust_id , c_phone)

  * Every attribute in this relation is atomic.

- Receive (so_id , order_id)

  * Every attribute in this relation is atomic.

6.SQL

SQL Statements for database creation & constrains

(01)

```
CREATE TABLE Customer
  (
  C_Id  INT NOT NULL PRIMARY KEY,
  C_name VARCHAR (15) NOT NULL,
  Address VARCHAR (50) NOT NUL,
  C_phone CHAR (10), NOT NUL,
  Catalog_Id INT,
  CONSTRAINT fk_CatCustomer FOREIGN KEY (Catalog_Id)
  REFERENCES Catalog(Catalog_Id),
  )
```

(02)

```
CREATE TABLE O_details
  (
  O_Id  int  NOT NULL PRIMARY KEY,
  O_time  TIME,
  Qty INT,
  C_Id  int,
  CONSTRAINT fk_CusOrders FOREIGN KEY (C_Id)
  REFERENCES Customer (C_Id),
  )
```

(03) CREATE TABLE Merchandiser
(
M_Id int NOT NULL PRIMARY KEY,
M_Phone CHAR (10),
M_Name VARCHAR (30),
)

(04) CREATE TABLE Payment
(
P_Id int NOT NULL PRIMARY KEY,
P_Date DATE,
Cust_Id INT ,
Amount INT,
Sh_Id int ,
CONSTRAINT fk_Shipment_Payment FOREIGN KEY (Sh_Id)
REFERENCES Shipment_Details (Sh_Id),
)

(05) CREATE TABLE Supplier
(
S_Id int NOT NULL PRIMARY KEY,
S_name VARCHAR (30) NOT NULL,
S_Address VARCHAR (50) NOT NUL,
)

(06) CREATE TABLE Supplier_Order_details
(
S_Id int NOT NULL PRIMARY KEY,
Order_Date DATE,
Status VARCHAR(30),
S_Id int ,
m_Id int,
CONSTRAINT fk_SupOrders FOREIGN KEY (S_Id)
REFERENCES Suppliers (S_Id),
CONSTRAINT fk_Merch_SupOrders FOREIGN KEY (m_Id)
REFERENCES Merchandiser (m_Id),
)

(07) CREATE TABLE Supplier_Payment
(
SP_Id int NOT NULL PRIMARY KEY,
Sp_Date DATE ,
order_Id int ,
M_Id int,
CONSTRAINT fk_SupOrders FOREIGN KEY (order_Id)
Supplier_order_heder(order_Id),
CONSTRAINT fk_Merch_ SupPayment FOREIGN KEY (M_Id)
REFERENCES Merchandiser (M_Id)
)

(08) CREATE TABLE Catalog
(
Catalog_Id int NOT NULL PRIMARY KEY,
Categories VARCHAR (15) NOT NULL,
M_Id int,
CONSTRAINT fk_Merch_Catelog FOREIGN KEY (m_Id)
REFERENCES Merchandiser (m_Id)
)

(09) CREATE TABLE Item
(
I_Code int NOT NULL PRIMARY KEY,
I_Name VARCHAR (15) NOT NULL,
I_Price MONEY,
Cat_Id int,
O_ID int,
CONSTRAINT fk_Item_Catelog FOREIGN KEY (Catalog_Id)
REFERENCES Catalog (Catalog_Id),
CONSTRAINT fk_OrderDetails_Items FOREIGN KEY (O_ID)
REFERENCES O_Detail(O_ID),
)

(10) CREATE TABLE Shipment_Details
(
Sh ID INT NOT NULL PRIMARY KEY,
Ship_Date DATE NOT NULL,
O Date DATE,
Item_Code int ,
CONSTRAINT fk_Payments_Shipmentdetail FOREIGN KEY (P_ID)
REFERENCES Payments (P_ID),
)

(11) CREATE TABLE O_Heder
(
Heder_Id int NOT NULL PRIMARY KEY,
Heder Title VARCHAR (15) NOT NULL,
P_Id int,
O_Id int,
CONSTRAINT fk_payments_Oheder FOREIGN KEY (P_ID)
REFERENCES payments (P_ID),
CONSTRAINT fk_ODetails _Oheder FOREIGN KEY (O_ID)
REFERENCES O_Detail(O_ID),
)

(12) CREATE TABLE Supplier_Order_Heder
(
Order_Id int NOT NULL PRIMARY KEY,
Order_Title VARCHAR (15) NOT NULL,
SP_Id int,
O_Id int,
CONSTRAINT fk_Suppliers_SupOrderHeder FOREIGN KEY (S_ID)
REFERENCES Suppliers (S_ID),
CONSTRAINT fk_ SupplierPayements _ SupOrderHeder FOREIGN KEY (SP_ID)
REFERENCES Suppliers_Payements (SP_ID),
)

7. Contribution of group members ,

Lekamge , W.L. Dhananja Y.S. & Wickramasinghe, W.A. Chethana D. -:

- Conceptual Design - ER diagram
- Relational mapping
- Assumptions

Renuka, P.H. Heshan & Pramuditha, P.G. shan -:

- Normalization
- SQL