

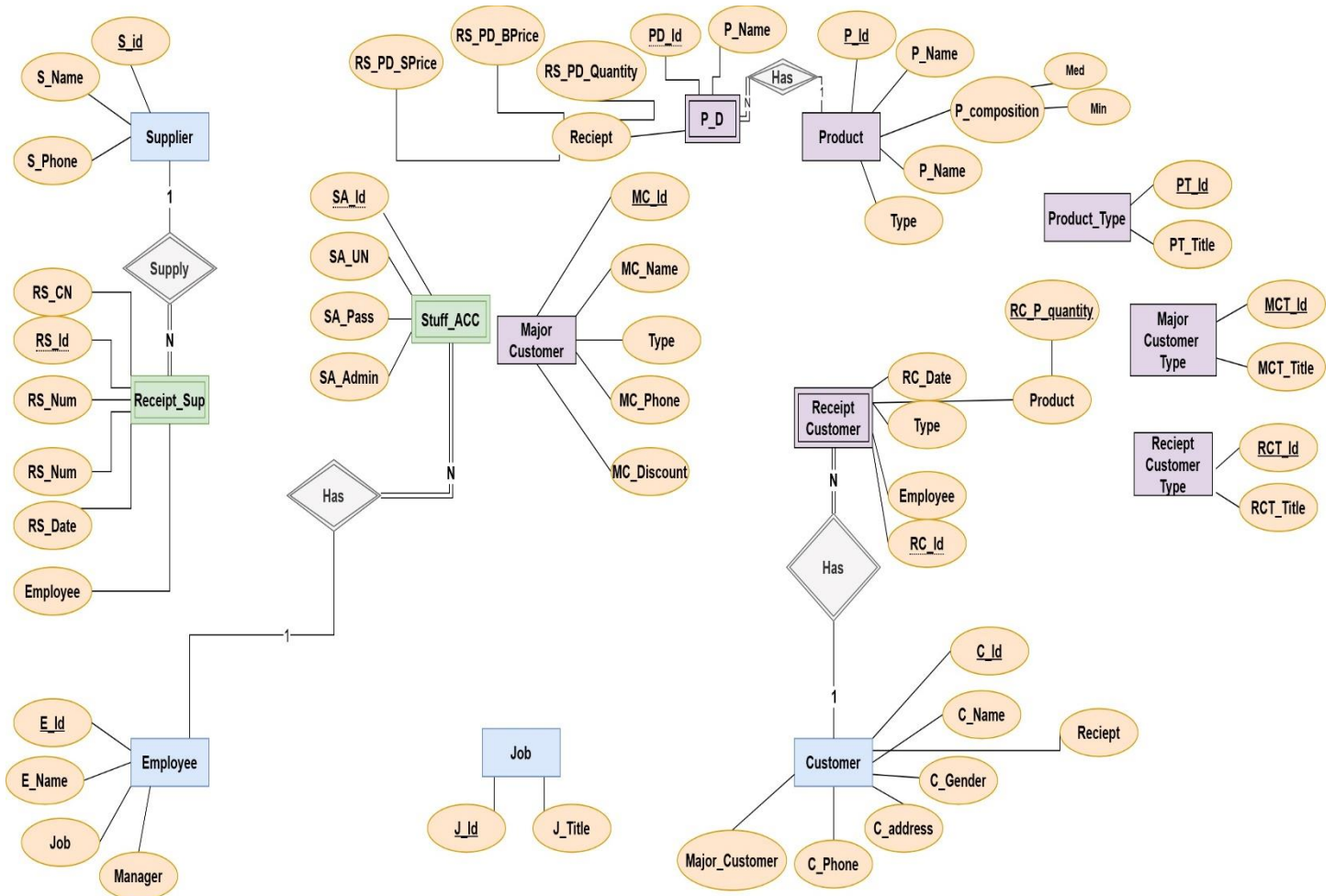
Data base project

Project Title: Pharmacy Data base

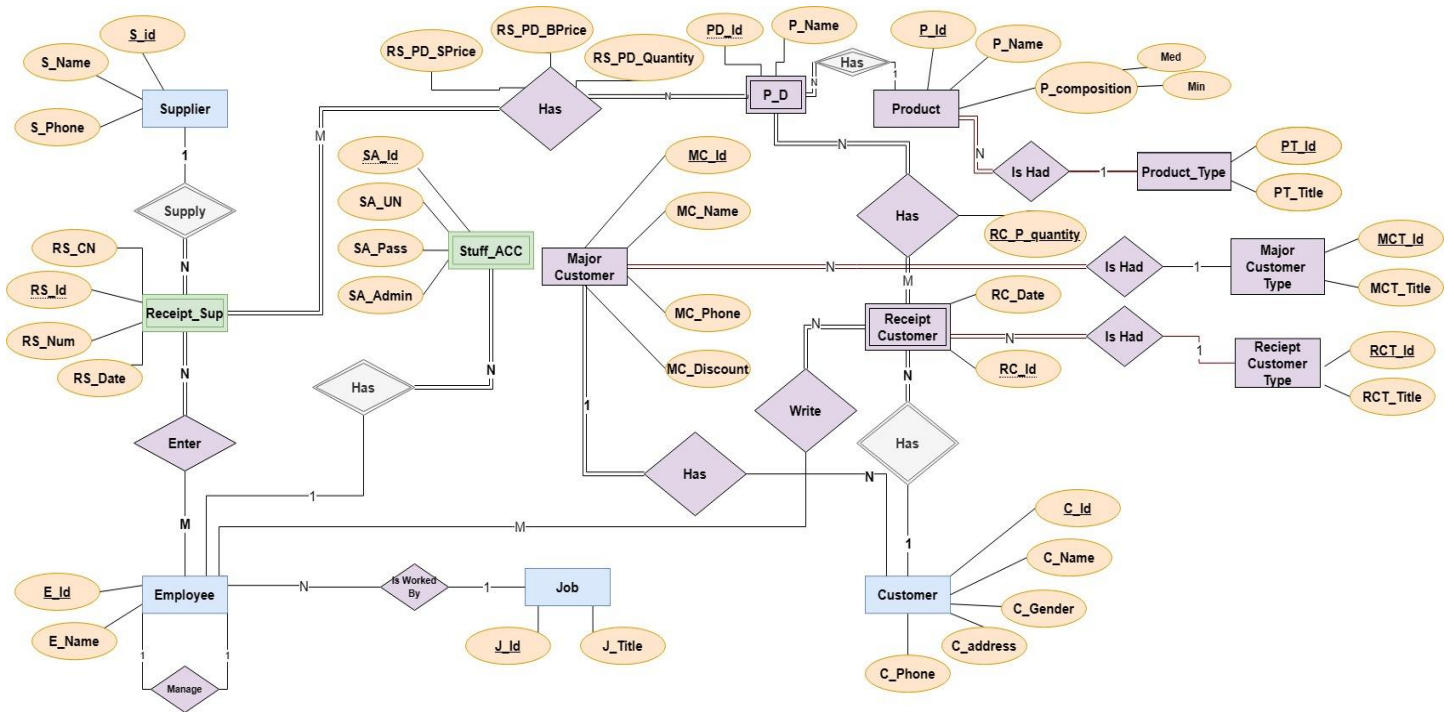
Pharmacy problems

Problems that a pharmacy can face without a database system, and their solutions: Tracking the drug's stock; The pharmacy holds all types of medicines which are difficult to track, each medicine has name, can be sold by min or mid or max, also each date of same medicine has different or same price comparing to previous. The pharmacist wouldn't be able to track the medicines that are received by the supplier and who received them from employee (who has name, job) , the ones out of stock, and the ones that are ordered by the customers. So, the pharmacist can mark the medicine, where it shows the amount of drug in stock, the drug that needs to be delivered from the company that hold a contract (supplier), the date & time of the drug that the customer ordered. Tracking Customers; Who served customer, it can be 1 or more employee Some customers either pay by cash and give the prescription, or they have a health insurance, where they submit the prescription to receive a form in order to give to the pharmacy, and the pharmacy pays the health insurance later. The pharmacist needs to organize the customers who paid directly or by their health insurance ID, along with the date & time the process took place. If the pharmacist wouldn't be able to trace the customer's processes, it'll lead to a de-organized failure, therefore, the profit will decrease. Expired Medicines; When a certain period of time passes, the pharmacy makes a yearly stocktaking of the medicines. For example, checking the stock, the expired medicines, the amount left from a certain category (painkiller, antibiotic, etc.) The pharmacist can trail the expired medicines that need to be shipped back to the company or supplier, and order a new stock. That way, the pharmacy is updated and the drug is available to be sold. If The pharmacist can't check the medicine and their expiry date along with the ones that aren't in stock, it will cause an unwanted disaster.

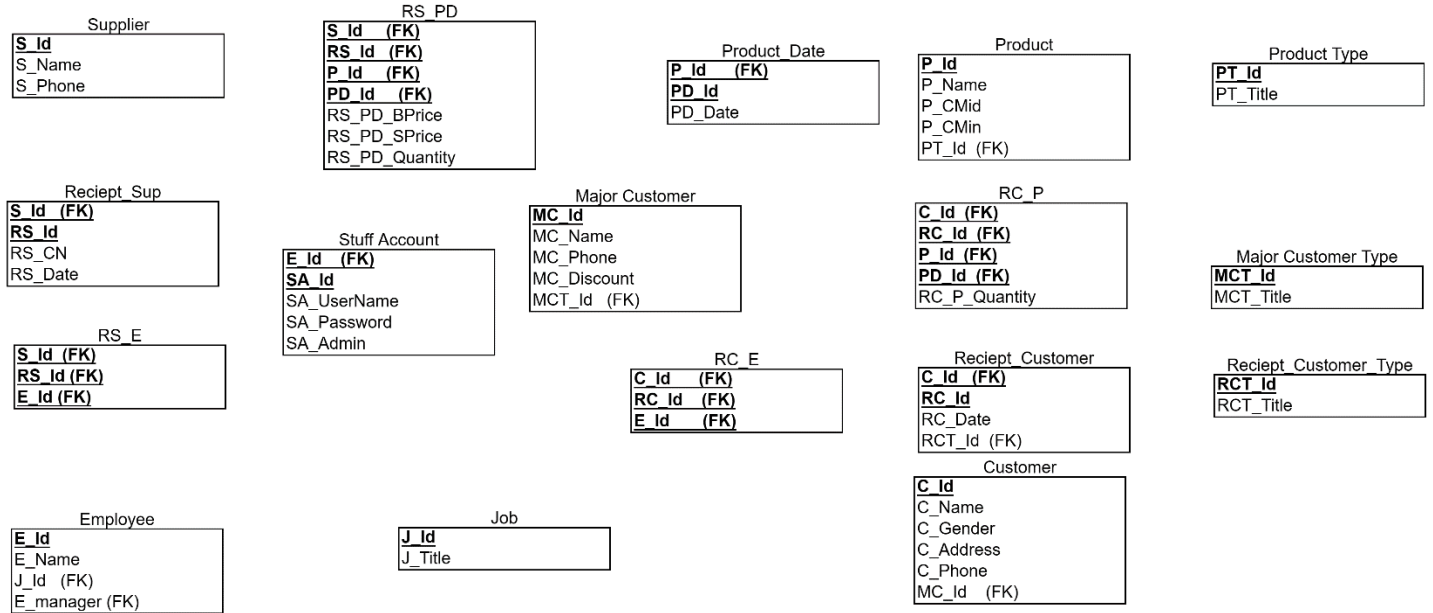
Rough ERD



Final Erd



Mapping



Complex Query

1	Get Total Cost of Receipt	<pre>select s_name as "Supplier",rs_id as "Reciept Number",rs_cn as "Company Reciept Number",rs_date as "Date" , sum(rs_pd.rs_pd_bprice*rs_pd.rs_pd_quantity) as "Total Price" from product inner join product_date using(P_Id) inner join RS_PD using(pd_id) inner join RECIEPT_SUPplier using(S_id,RS_id) inner join supplier using(S_id) group by s_name,rs_id,rs_cn,rs_date;</pre>
2	Get all products Before Determined Date With receipt supplier date And its expire date	<pre>select product.p_id as "id" , product_date.pd_id,P_Name as "Name",s_name as "Supplier",rs_date as "Reciept Date" ,pd_date as "Product expire Date",rs_pd_quantity as "Quantity" from product inner join product_date on product.p_id=product_date.P_Id inner join RS_PD on product_date.pd_id = RS_PD.pd_id inner join RECIEPT_SUPPLIER using(S_id,RS_id) inner join supplier using(S_id) Where to_char(rs_date,'YYYYMMDD') < to_char(to_date('2022-03-20','YYYY-MM-DD'),'YYYYMMYY') order by P_name;</pre>
3	Get products In Receipt Customer	<pre>select product.p_id as "id" ,product.p_name As "Name" , product_date.pd_date as "Expire Date" , rs_pd.rs_pd_sprice ,rc_pd.rc_pd_quantity from product inner join product_date on product.p_id=product_date.p_id inner join rs_pd on rs_pd.p_id=product_date.p_id and rs_pd.pd_id=product_date.pd_id inner join rc_pd on rc_pd.p_id=product_date.p_id and rc_pd.pd_id=product_date.pd_id where product.p_id in (select p_id from rc_pd where rc_pd.rc_id =8007);</pre>
4	Get Expired products	<pre>select product.p_id as "id",product.P_Name as "Name",product_date.pd_date as "Product expire Date" from product inner join product_date on product.p_id=product_date.P_Id Where abs(to_char(pd_date,'YYYYMM')-to_char(to_date('2022-03-20','YYYY-MM-DD'),'YYYYMM')) >= 3;</pre>
5	Get Stock	<pre>select p_id,p_name ,pd_date, sum(quantity) quantity from (select p_id , rs_pd_quantity as quantity from rs_pd union all select p_id, -rc_pd_quantity as quantity from rc_pd) dt inner join product using(p_id) inner join product_date using(p_id) group by p_id,p_name, pd_date</pre> <p>HINT: We Can use NVL or COALESCE Functions instead of all this.</p> <pre>select p_id ,p_name ,pd_date,COALESCE(rs_pd_quantity,0)-COALESCE(rc_pd_quantity,0) from rs_pd full outer join rc_pd using(p_id) natural join product natural join product_date;</pre>
6	Get Total Price of Receipt Customer	<pre>select sum(rs_pd.rs_pd_sprice*rc_pd.rc_pd_quantity) as "Total Price" from product inner join product_date on product.p_id=product_date.p_id inner join rs_pd on rs_pd.p_id=product_date.p_id and rs_pd.pd_id=product_date.pd_id inner join rc_pd on rc_pd.p_id=product_date.p_id and rc_pd.pd_id=product_date.pd_id where product.p_id in (select p_id from rc_pd where rc_pd.rc_id =8007);</pre>

DDL Statements

Supplier	<pre>CREATE TABLE Supplier(S_Id int GENERATED ALWAYS as IDENTITY(START with 1001 INCREMENT by 1) PRIMARY KEY, S_Name varchar(255) NOT NULL, S_Phone varchar(255) DEFAULT '+20' NOT NULL);</pre>
Receipt Supplier	<pre>CREATE TABLE Reciept_Supplier(S_Id int NOT NULL , RS_Id int GENERATED ALWAYS as IDENTITY(START with 3001 INCREMENT by 1) PRIMARY KEY, RS_CN int NOT NULL, RS_Date DATE NOT NULL, FOREIGN KEY (S_Id) REFERENCES supplier(S_Id)); Alter table RECIEPT_SUPPLIER Add constraint RS_PK Unique (S_Id,RS_Id);</pre>
Job	<pre>CREATE TABLE job(J_Id int GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1) PRIMARY KEY, J_Title varchar(255) NOT NULL);</pre>
Employee	<pre>CREATE TABLE employee(E_Id int GENERATED ALWAYS as IDENTITY(START with 201 INCREMENT by 1) PRIMARY KEY, E_Name varchar(255) NOT NULL, J_Id int, E_manager int, FOREIGN KEY (E_manager) REFERENCES employee(E_Id), FOREIGN KEY (J_Id) REFERENCES job(J_Id)); drop table employee;</pre>
Stuff Account	<pre>CREATE TABLE Stuff_Acc(SA_Id int GENERATED ALWAYS as IDENTITY(START with 31 INCREMENT by 1) PRIMARY KEY, SA_UN varchar(255) NOT NULL, SA_Pass varchar(255) NOT NULL, SA_Admin number(1) NOT NULL, E_Id int NOT NULL, FOREIGN KEY (E_Id) REFERENCES employee(E_Id)); Alter table Stuff_Acc Add constraint SAC_PK Unique (E_Id,SA_Id);</pre>
RS_E (Receipt Supllier And Employee)	<pre>CREATE TABLE RS_E(S_Id int NOT NULL, RS_Id int NOT NULL, E_Id int NOT NULL, FOREIGN KEY (S_Id,RS_Id) REFERENCES Reciept_Supplier(S_Id,RS_Id), FOREIGN KEY (E_Id) REFERENCES Employee(E_Id)); Alter table RS_E Add constraint RSE_PK Unique (S_Id,RS_Id,E_Id);</pre>

Product_Type	<pre> CREATE TABLE Product_Type(PT_Id int GENERATED ALWAYS as IDENTITY(START with 901 INCREMENT by 1) PRIMARY KEY, PT_Title varchar(255) NOT NULL); drop table product_type; </pre>
Product	<pre> CREATE TABLE Product(P_Id int GENERATED ALWAYS as IDENTITY(START with 10001 INCREMENT by 1) PRIMARY KEY, P_Name varchar(255) NOT NULL, P_CMax int NOT NULL, P_CMid int NOT NULL, P_CMin int NOT NULL, PT_Id int NOT NULL, FOREIGN KEY (PT_Id) REFERENCES Product_Type(PT_Id)); Alter table Product DROP column P_CMax ; drop table product; </pre>
PD Product Date	<pre> CREATE table product_date(P_Id int NOT NULL, PD_Id int GENERATED ALWAYS as IDENTITY(START with 6001 INCREMENT by 1) PRIMARY KEY, PD_Date date NOT NULL); Alter table PRODUCT_DATE Add constraint PD_PK Unique (P_Id,PD_Id); </pre>
RS_PD Receipt Supplier And Product	<pre> CREATE TABLE RS_PD(S_Id int NOT NULL, RS_Id int NOT NULL, P_Id int NOT NULL, PD_Id int NOT NULL, RS_PD_BPrice number NOT NULL, RS_PD_SPrice number NOT NULL, RS_PD_Quantity number NOT NULL, FOREIGN KEY (S_Id,RS_Id) REFERENCES RECIEPT_SUPPLIER(S_Id,RS_Id), FOREIGN KEY (P_Id,PD_Id) REFERENCES PRODUCT_DATE(P_Id,PD_Id)); Alter table RS_PD Add constraint RSPD_PK Unique (S_Id,RS_Id,P_Id,PD_Id); </pre>
Major Customer Type	<pre> CREATE TABLE MC_Type(MCT_Id int GENERATED ALWAYS as IDENTITY(START with 701 INCREMENT by 1) PRIMARY KEY, MCT_Title varchar(255) NOT NULL); </pre>
Major Customer	<pre> CREATE TABLE major_customer(MC_Id int GENERATED ALWAYS as IDENTITY(START with 401 INCREMENT by 1) PRIMARY KEY, MC_Name varchar(255) NOT NULL, MC_Phone varchar(255) DEFAULT '+20' NOT NULL, </pre>

	MC_Discount int NOT NULL, MCT_Id int NOT NULL, FOREIGN KEY (MCT_Id) REFERENCES MC_Type(MCT_Id));
Customer	CREATE TABLE customer(C_Id int GENERATED ALWAYS as IDENTITY(START with 501 INCREMENT by 1) PRIMARY KEY, C_Name varchar(255) NOT NULL, C_Gender NUMBER(1) NOT NULL, C_Address varchar(255), C_Phone varchar(255), MC_Id int, FOREIGN KEY (MC_Id) REFERENCES major_customer(MC_Id));
Reciept_Customer_Type	CREATE TABLE Reciept_Customer_Type(RCT_Id int GENERATED ALWAYS as IDENTITY(START with 9001 INCREMENT by 1) PRIMARY KEY, RCT_Title varchar(255) NOT NULL);
Reciept_Customer	CREATE TABLE Reciept_Customer(C_Id int NOT NULL, RC_Id int GENERATED ALWAYS as IDENTITY(START with 8001 INCREMENT by 1) PRIMARY KEY, RC_Date Date NOT NULL, RCT_Id int, FOREIGN KEY (C_Id) REFERENCES Customer(C_Id), FOREIGN KEY (RCT_Id) REFERENCES Reciept_Customer_Type(RCT_Id)); Alter table Reciept_Customer add constraint RCC_PK unique (C_Id,RC_Id);
RS_E	CREATE TABLE RS_E(S_Id int NOT NULL, RS_Id int NOT NULL, E_Id int NOT NULL, FOREIGN KEY (S_Id,RS_Id) REFERENCES Reciept_Supplier(S_Id,RS_Id), FOREIGN KEY (E_Id) REFERENCES Employee(E_Id)); Alter table RS_E Add constraint RSE_PK Unique (S_Id,RS_Id,E_Id);
RC_E	CREATE TABLE RC_E(C_Id int NOT NULL, RC_Id int NOT NULL, E_Id int NOT NULL, FOREIGN KEY (C_Id,RC_Id) REFERENCES Reciept_Customer(C_Id,RC_Id), FOREIGN KEY (E_Id) REFERENCES Employee(E_Id)); Alter table RC_E Add constraint RSE_PKe Unique (C_Id,RC_Id,E_Id);

Insert Sample

Supplier	INSERT INTO Supplier(S_Name,S_Phone) values('UCP','+201234567890');
Receipt Supplier	<pre> create or replace NONEDITIONABLE procedure Insert_ReceiptS(num_SID in number,num_RSCN in number,num_RSD in Date) is begin insert into RECIEPT_SUPPLIER(S_ID,RS_CN,RS_Date) values(num_SID,num_RSCN,num_RSD); end; exec Insert_ReceiptS(1001,4984611,TO_DATE('2020-03-18','YYYY-MM-DD'));</pre>
Product Type	INSERT INTO product_type (pt_title) values('cream');
Product	INSERT INTO product(p_name, p_cmid, p_cmin, pt_id) values('Abimol 125ml', 1, 1, 907);
Product Date & RS_PD	<pre> create or replace NONEDITIONABLE procedure Insert_Receipt_S(num_RS_Id in number ,num_P_ID in number,num_PD_DATE in Date,num_RS_PD_BPrice in number,num_RS_PD_SPrice in number,num_RS_PD_Quantity in number) IS numS_Id receipt_supplier.s_id%TYPE; numPD_Id product_date.pd_id%TYPE; begin select s_id into numS_Id from RECIEPT_SUPPLIER where RS_Id=num_RS_Id; insert into PRODUCT_DATE(P_ID,PD_DATE) values(num_P_ID,num_PD_DATE); select pd_id into numPD_Id from PRODUCT_DATE where pd_date like num_PD_DATE and P_Id = num_P_ID; insert into RS_PD(S_Id ,RS_Id ,P_Id ,PD_Id ,RS_PD_BPrice,RS_PD_SPrice ,RS_PD_Quantity) values(numS_Id,num_RS_Id,num_P_ID,numPD_Id,num_RS_PD_BPrice,num_RS_PD_SPrice ,num_RS_PD_Quantity); end; exec insert_receipt_s(3001,10006,TO_DATE('2020-07-01','YYYY-MM-DD'),262.5,350,3);</pre>
Job	Insert into JOB(J_Title) values('Pharmacist');
Employee	INSERT into employee(E_Name,J_Id) values('Kolood Diab',3);
RS_E	insert into RS_E(S_Id,RS_Id,E_Id) values(1001,3001,203);
Receipt Customer	insert into receipt_customer(c_id,rc_date,rct_id) values(501,to_date('2020-03-20','YYYY-MM-DD'),9001);

RC_PD	create or replace NONEDITIONABLE procedure Insert_Receipt_C(num_C_Id in number,num_RC_Date in Date ,num_P_ID in number,x in number,num_RC_PD_Quantity in number) IS numRC_Id receipt_customer.rc_id%TYPE; numPD_Id RC_PD.pd_id%TYPE := x; begin select rc_id into numRC_Id from receipt_customer where C_Id=num_C_Id; insert into RC_PD(C_Id ,RC_Id ,P_Id ,PD_Id ,RC_PD_Quantity) values(num_C_Id,numRC_Id,num_P_ID,numPD_Id,num_RC_PD_Quantity); end;
Stuff ACC	insert into Stuff_Acc(SA_UN,SA_Pass,SA_Admin,E_Id) values('Admin','I am admin',1,201);
Receipt Customer Type	Insert into Receipt_Customer_Type(RCT_Title) values('In Pharmacy');
Customer	INSERT INTO customer(C_NAME,C_Gender,C_Address,C_phone,MC_Id) values('Ali Yasser',1,'11 Abas Elakad','+2084564974808',403);
Major Customer Type	INSERT INTO MC_Type(MCT_ID,MCT_TITLE) values('701','HOSPITAL');
Major Customer	Insert major_customer(MC_Name,MC_Phone,MC_discount,MCT_Id) values('Ahl Almasreen','+02122334434',7,701)
RC_E	insert into RC_E(C_ID,RC_ID,E_ID) values(501,8007,202);