```
Create or Replace Procedure CheckDate(
  date_chk in date)
is
begin
  if( date_chk > SYSDATE) then

    raise_application_error(-20001,
      'Date cannot be after todays date'
    );
  end if;
end;

CREATE OR REPLACE TRIGGER trg_check_dates
  BEFORE INSERT OR UPDATE ON APPLE
  FOR EACH ROW
BEGIN
  CheckDate(:new.A_DATE);
END;

CREATE OR REPLACE TRIGGER trg_check_order_received_dates
  BEFORE INSERT OR UPDATE ON Order_recieved
  FOR EACH ROW
BEGIN
   if( :new.O_DATE > SYSDATE) then

    raise_application_error(-20001,
      'Date cannot be after todays date'
    );
  end if;
END;

CREATE OR REPLACE TRIGGER check_dt
AFTER INSERT OR UPDATE ON SHIPMENT
FOR EACH ROW
DECLARE
 po_date Order_recieved.O_Date%type;
BEGIN
 SELECT O_Date
 INTO po_date
 FROM Order_recieved
 WHERE PO_No = :NEW.PO_no;

IF(:NEW.SH_Date<po_date)THEN
 RAISE_APPLICATION_ERROR(-20250,'Shipment cannot be done before receiving order!');
```

```
END IF;
END;
```

```
Create or Replace Procedure CheckDate(
  date_chk in date)
is
begin
  if( date_chk > SYSDATE) then

    raise_application_error(-20001,
      'Date cannot be after todays date'
    );
  end if;
end;
```

```
Create table Customer(
C_ID varchar2(4) CHECK (C_ID LIKE 'C%'),
C_Name varchar2(30) NOT NULL,
Address varchar2(30),
A_City varchar2(30),
A_State Varchar2(30),
Zip integer NOT NULL CONSTRAINT zip_constraint CHECK(length(Zip)=5),
Phone integer NOT NULL UNIQUE CHECK(length(Phone)=10),
Email varchar2(30) NOT NULL UNIQUE CONSTRAINT email_constraint check ( Email LIKE
'%@%.%' AND email NOT LIKE '@%' AND email NOT LIKE '%@%@%' ),
CONSTRAINT pk_customers PRIMARY KEY (C_ID)
)
```

```
Create table Order_recieved(
PO_No varchar2(5) CHECK (PO_No LIKE 'PO%'),
C_id varchar2(30),
O_Date date ,
Delivery_date date,
Order_status varchar2(30) CONSTRAINT status_chk check (Order_status in ('Cancelled','In
Progress','Completed', 'Received')),
CONSTRAINT pk_order_recieved PRIMARY KEY (PO_NO),
CONSTRAINT fk_c_ID FOREIGN KEY (c_id) references Customer(c_id)
);



CREATE OR REPLACE TRIGGER trg_check_order_received_dates
  BEFORE INSERT OR UPDATE ON Order_recieved
  FOR EACH ROW
BEGIN
   if( :new.O_DATE > SYSDATE) then

    raise_application_error(-20001,
      'Date cannot be after todays date'
    );
  end if;
END;

Create table Product(
Pdt_no varchar(4) CHECK (Pdt_No LIKE 'P%') CONSTRAINT  pk_pdt primary key,
Pdt_name varchar2(30) NOT NULL UNIQUE,
Rate number(7,3));

Create table Warehouse(
Pdt_no varchar2(4) UNIQUE NOT NULL CONSTRAINT pdt_no references product(pdt_no),
Pdt_qty integer NOT NULL CHECK(Pdt_qty>0)
);

Create table Order_details(
Pdt_no varchar2(4) references Product (Pdt_No),
PO_no varchar2(4) references Order_recieved (PO_No),
Qty integer NOT NULL CHECK(Qty>0),
PRIMARY KEY (pdt_no, PO_no)
)


Create table Shipment(
SH_Id varchar(5) CHECK (SH_Id LIKE 'SH%') CONSTRAINT  pk_shp_id primary key,
SH_Date date,
PO_No varchar(5) references Order_recieved (PO_No),
T_Id varchar(4) references Transp_Vendor(T_Id)
)

insert into Customer
```

```
values('C01','Fairfield Industries','Hemenway St','Montgomery','AL', 048887, 7778889998,
'fairfield@gmail.com');

insert into Order_recieved
values('PO5','C01', to_date('10-18-2018','mm-dd-yyyy'), to_date('12-18-2018','mm-dd-
yyyy'),'Completed');

insert into Shipment
values('SH05', to_date('10-10-2018','mm-dd-yyyy'),'PO5','TV03');

insert into Transp_Vendor
values('TV04','HDEHDf','ERFDGf','RGTRf','RDGsf',21156,9897296899,'R.MANYAGUPddTA@GMA
IL.COM');


CREATE OR REPLACE TRIGGER check_dt
AFTER INSERT OR UPDATE ON SHIPMENT
FOR EACH ROW
DECLARE
 po_date Order_recieved.O_Date%type;
BEGIN
 SELECT O_Date
 INTO po_date
 FROM Order_recieved
 WHERE PO_No = :NEW.PO_no;

IF(:NEW.SH_Date<po_date)THEN
 RAISE_APPLICATION_ERROR(-20250,'Shipment cannot be done before receiving order!');
END IF;
END;
```

```
Create table Transp_Vendor(
T_Id varchar2(4) CHECK (T_ID LIKE 'T%') CONSTRAINT pk_trans_id primary key,
T_Name varchar2(30) NOT NULL,
T_Address varchar2(30),
T_City varchar2(30),
T_State Varchar2(30),
Zip integer NOT NULL CHECK(length(Zip)=4),
Phone integer NOT NULL UNIQUE CHECK(length(Phone)=10),
Email varchar2(30) NOT NULL UNIQUE check ( Email LIKE '%@%.%' AND email NOT LIKE '@%'
AND email NOT LIKE '%@%@%' )
)
select * from Transp_Vendor

Create table Raw_material(
R_Id varchar2(4) CHECK (R_ID LIKE 'R%') CONSTRAINT pk_raw_id primary key,
R_Name varchar2(30) NOT NULL UNIQUE,
Pdt_No varchar(5) references Order_recieved (PO_No) ON DELETE CASCADE
)

Create table Suppliers(
S_Id varchar2(4) CHECK (S_ID LIKE 'S%') CONSTRAINT pk_suppliers_id primary key,
S_Name varchar2(30) NOT NULL,
S_Address varchar2(30),
S_City varchar2(30),
S_State Varchar2(30),
Zip integer NOT NULL CHECK(length(Zip)=5),
Phone integer NOT NULL UNIQUE CHECK(length(Phone)=10),
Email varchar2(30) NOT NULL UNIQUE check ( Email LIKE '%@%.%' AND email NOT LIKE '@%'
AND email NOT LIKE '%@%@%' )
)

Create table Order_placed(
PR_Id varchar2(5) CHECK (PR_Id LIKE 'PR%') CONSTRAINT pk_procurement_id primary key,
Raw_Id varchar2(4) references Raw_material (R_Id),
Order_date date,
Sup_Id varchar2(4) references Suppliers(S_Id),
Qty integer NOT NULL CHECK(Qty>0)
)

CREATE OR REPLACE TRIGGER order_placed_check_dates
  BEFORE INSERT OR UPDATE ON Order_placed
  FOR EACH ROW
BEGIN
  CheckDate(:new.Order_date);
END;
```