INFO 6210

Database Management and Database Design

PROJECT:

MANUFACTURING ENTERPRISE DIGITIZATION DATABASE

Team Members :

Manya Raman
Kedar Chaudhari
Pratik Huddar
Pravin Dubey

Introduction:

This project deals with designing and implementing a system for a small-scale manufacturing firm. This system covers all aspects of the functioning of the enterprise and aims to conveniently give access to all employees according to their rights. With this project, we aim to organize the small scale firm in a more efficient way and ensure nothing slips through the cracks.

Problem Statement:

Despite having the expertise and workforce to work on the manufacturing processes and daily functioning, small businesses are often capped on the growth due to a brightly visible yet ignored fact of digitization of the business. Digitization in this context means creating and maintaining an ERP or a management system to track the daily functioning of the small business. The businesses typically have a workforce of not more than 100 laborers, owners, and floor/workshop managers collectively. While many off-the-shelf ERP software which is available in the market cater to the needs of a large corporation, for small businesses these softwares can either clog systems with unneeded options or can fail to cover certain concerns. The small enterprise that this project focuses on, rely heavily on the pen and paper format for tracking their work which is often inefficient. Sometimes there are high priority tasks that must be undertaken and completed on an immediate basis pushing many ongoing tasks down the pipeline. Due to the lack of a proper tracking management system of these daily tasks, there are high chances of them being skipped completely resulting in delays and revenue loss, along with
man-hours lost in the process.

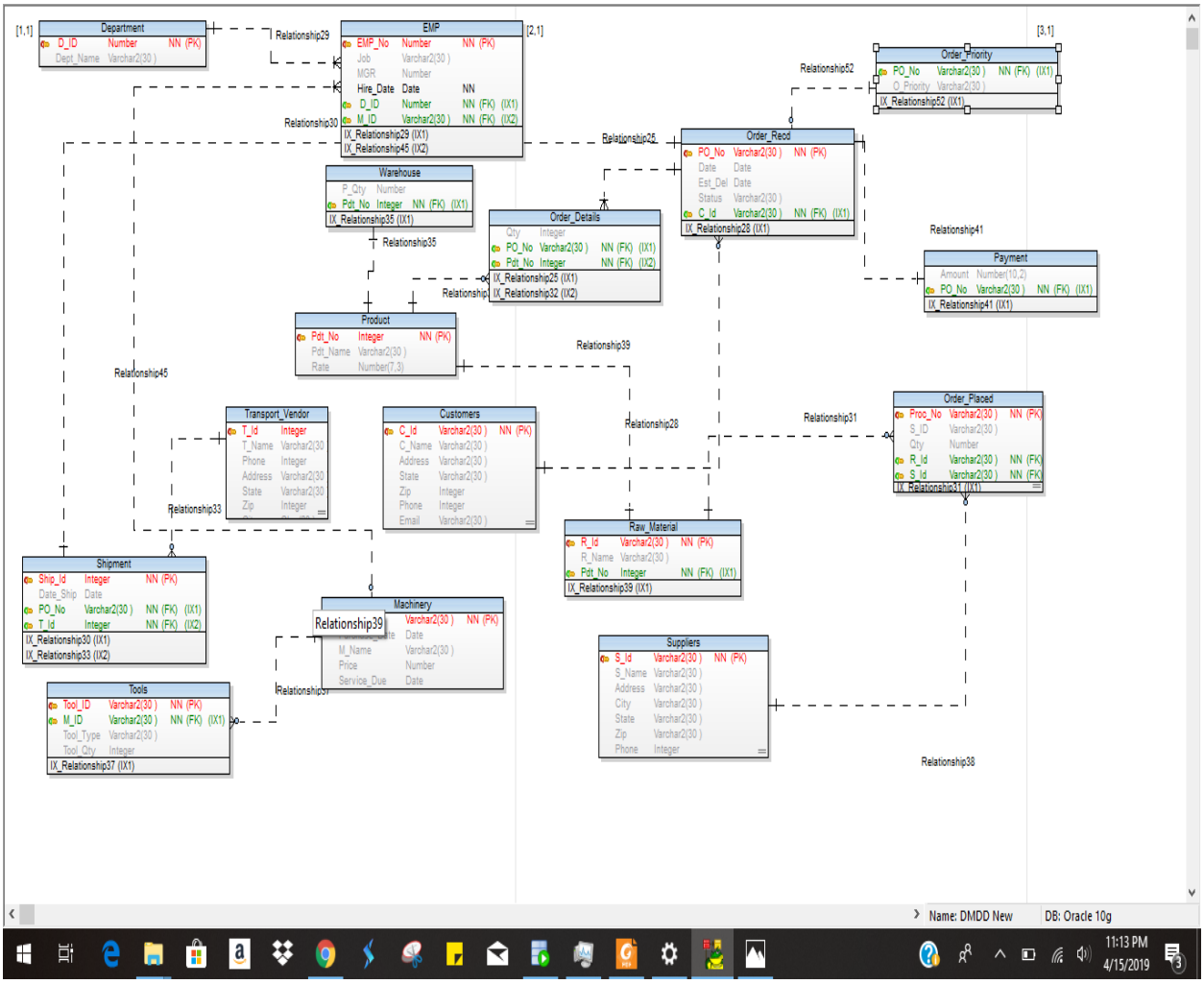
Entities in the Database:

- Department
- Employees
- Customers
- Machinery
- Tools
- Transport Vendors
- Products
- Orders Received
- Order Details
- Orders Placed
- Warehouse
- Shipment
- Suppliers
- Raw Materials
- Payment
- Order Priority

BUSINESS RULES

- The System has entities in product, process, resources, employees, customer, Transport vendors.
- Employees are divided in multiple sections based on departments.
- Each department can have one or more employees associated with it.
- Each department can only have one manager.
- Managers can have many employees under them but every employee can have only one manager.
- Each employee must belong to only one department.
- The President is the only head of all the managers and all the managers report to him.
- Each employee is associated to only one machine.
- Each purchase order is associated with only one customer
- Each purchase order (PO) can have one or more products in it
- Amount equal to quantity * rate in Payment table
- Each Ordered placed can have only one raw material in it
- Each raw material can be in one or many placed orders
- One or many raw materials can be ordered from a single supplier.
- Hourly wage of employee must be less than his/her manager
- Hourly wage of manager must be less than the president
- The delivery date can't be prior to ordered received date
- All the Shipments ID can only have one Transport Vendor associated to it.
- Every Transport Vendor can have one or more Shipment IDs associated to it.
- Each shipment is associated with just one Purchase Order.
- Payment table can only be updated once the status of the product in 'order received' changes to **completed**
- All the tools are associated with one or more machines.
- All the machines may/may not have tools associated to them.
- Every order placed having the status is **In Process** has a rank assigned based on the delivery date; if a new order has a delivery date earlier than the top ranked order, existing orders are pushed down the pipeline by 1 rank point

Updated E-R Diagram



Queries

1. Department, Department Name and Number of Employees

```
select d.d_id, d.d_name, count(*) as emp_count
from employee e
left join dept d on e.d_id=d.d_id
Where d.D_ID is not null
group by d.d_id, d.d_name
order by emp_count desc;
```

Script Output x | Query Result x

SQL | All Rows Fetched: 6 in 0.001 seconds

	D_ID	D_NAME	EMP_COUNT
1	D04	Production	15
2	D02	Inventory Management	5
3	D01	Quality	5
4	D05	Human Resources	5
5	D06	Delivery & Logistics	5
6	D03	Accounts	5

2. Total Amount Paid by the customer to our company

```
select * from(select c.c_id,c.C_Name,sum(payment) as total_amount
from payment p
left join order_recieved o on p.po_no= o.po_no
left join  customer c on o.c_id= c.c_id
group by c.c_id,c.C_Name
order by total_amount desc)
where rownum <=10;
```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x | Query Result

SQL | All Rows Fetched: 10 in 0.004 seconds

	C_ID	C_NAME	TOTAL_AMOUNT
1	C01	Fairfield Industries	8000
2	C11	Gilchrist Industries	7000
3	C18	Clarke Industries	6375
4	C15	Lee Industries	6300
5	C05	Phoenix Industries	6200
6	C17	Waugh Industries	6175
7	C09	Ronaldinho Industries	6100
8	C10	Hayden Group of Companies	6075
9	C14	Dr. Brand Industries	6050
10	C16	Warne Industries	5700

3. Salary expense in each department can be fetched

```
/*Salary expense in each department can be fetched using:*/
Select emp.D_ID, dept.D_name, sum(salary) as Total_Salary_Expense
From employee emp
Left join dept on dept.D_ID=emp.D_ID
Where dept.D_ID is not null
Group by emp.D_ID, dept.D_name
Order by Total_Salary_Expense DESC;
```

Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.002 seconds

	D_ID	D_NAME	TOTAL_SALARY_EXPENSE
1	D04	Production	103000
2	D05	Human Resources	33000
3	D02	Inventory Management	32500
4	D03	Accounts	31500
5	D06	Delivery & Logistics	31000
6	D01	Quality	30000

4. The number of customers according to the state can be retrieved

```
/*The number of customers according to the state can be retrieved using:*/
Select a_state, count(*)as Customer_in_region
From customer
Group by a_state
order by  Customer_in_region desc;
```

Script Output x | Query Result x | Query Result 1 x

SQL | All Rows Fetched: 14 in 0.001 seconds

	A_STATE	CUSTOMER_IN_REGION
1	TX	4
2	AZ	2
3	OH	2
4	CA	2
5	NC	1
6	IL	1
7	ID	1
8	IN	1
9	FL	1
10	MD	1
11	AR	1
12	AL	1
13	AK	1
14	NE	1

5. Most ordered Raw Material can be found out

```
/*Most ordered Raw Material can be found out using the query given below:*/
Select r.R_Name, o.Raw_Id, sum(Qty) as Qty_ordered
From order_placed o
Left join Raw_Material r on r.R_Id=o.Raw_Id
Group by o.Raw_Id, r.R_Name
order by Qty_ordered desc;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 15 in 0.003 seconds

	R_NAME	RAW_ID	QTY_ORDERED
1	Alloy Steel	R02	320
2	Austenitic	R09	290
3	Iridium	R13	280
4	Low Carbon Steel	R06	280
5	Nickel	R15	240
6	Stainless Steel	R08	240
7	Tungsten	R12	230
8	Cobalt	R14	220
9	Ferritic	R10	215
10	Titanium	R04	205
11	High Carbon Steel	R03	140
12	Medium Carbon Steel	R07	100
13	Martensitic	R11	100
14	Cast Iron	R01	100
15	Pig Iron	R05	80

6. Transport Vendors who delivers frequently

```
Select s.T_ID, t.T_Name, count(*)  as No_of_deliveries
From shipment s
left join transp_vendor t on t.T_Id=s.T_Id
Group by s.T_Id, t.T_Name
ORDER BY s.T_Id ;
```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x

SQL | All Rows Fetched: 10 in 0.002 seconds

	T_ID	T_NAME	NO_OF_DELIVERIES
1	TV01	YRC Worldwide	4
2	TV02	Swift Transportation	4
3	TV03	Schneider International	5
4	TV04	JW Transportation	4
5	TV05	Ruben Worldwide	3
6	TV06	Conor Transportation	2
7	TV07	Cameron Worldwide	2
8	TV08	Diogo Transportation	1
9	TV09	Giles International	2
10	TV10	Elliot Transportation	3

7. Most frequent ordered products of the company

```
Select o.pdt_no, p.pdt_name,count(*) as times_ordered
    from order_details o
    left join product p
    on o.pdt_no=p.pdt_no
    group by o.pdt_no,p.pdt_name, p.rate
    order by times_ordered desc;
```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x | Query Re

SQL | All Rows Fetched: 15 in 0.003 seconds

	PDT_NO	PDT_NAME	TIMES_ORDERED
1	PR10	Internal Gear	6
2	PR15	Spline Shafts	4
3	PR08	Spiral Bevel Gear	4
4	PR13	Pawl	3
5	PR07	Worm Drive	3
6	PR04	Connecting Rod	3
7	PR01	Crank Shaft	3
8	PR11	Ratchet	3
9	PR14	Screw Gear	3
10	PR12	Miter Gear	3
11	PR06	Bevel Gear	2
12	PR03	Stub Axle	2
13	PR02	Gear Shift Levers	2
14	PR09	Helical Gear	2
15	PR05	Pinion	2

8. The product which generates maximum revenue for the company

```
select o.pdt_no,p.pdt_name, sum(o.qty*p.rate) as total_revenue_generate
    from order_details o
    left join product p
    on o.pdt_no=p.pdt_no
    group by o.pdt_no,p.pdt_name
    order by total_revenue_generate desc;
```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x

SQL | All Rows Fetched: 15 in 0.003 seconds

	PDT_NO	PDT_NAME	TOTAL_REVENUE_GENERATE
1	PR10	Internal Gear	21000
2	PR08	Spiral Bevel Gear	11475
3	PR15	Spline Shafts	11025
4	PR07	Worm Drive	10075
5	PR14	Screw Gear	9750
6	PR09	Helical Gear	7750
7	PR04	Connecting Rod	7500
8	PR13	Pawl	7425
9	PR01	Crank Shaft	7000
10	PR11	Ratchet	6300
11	PR03	Stub Axle	6000
12	PR12	Miter Gear	5700
13	PR05	Pinion	4550
14	PR06	Bevel Gear	4500
15	PR02	Gear Shift Levers	4500

9. List of top 10 customers according to number of orders placed by them

```
select * from(Select o.C_ID,c.C_Name,count(*) as no_orders
    FROM order_recieved o
    left join customer c on c.C_ID = o.C_ID
    group by o.C_id,c.C_Name
    order by no_orders desc)
where rownum <=10;
```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x | Quer

SQL | All Rows Fetched: 10 in 0.003 seconds

	C_ID	C_NAME	NO_ORDERS
1	C19	Fleming Industries	3
2	C09	Ronaldinho Industries	2
3	C20	Taibu Industries	2
4	C17	Waugh Industries	2
5	C05	Phoenix Industries	2
6	C01	Fairfield Industries	2
7	C15	Lee Industries	2
8	C08	Zidane Industries	2
9	C03	Capricorn Tech	2
10	C11	Gilchrist Industries	2

10. Database: Facts & Figures

	TABLE_NAME	NUM_ROWS	STATUS	COLUMNS	COMMENTS	INDEXED_COLUMNS	AVG_ROW_LEN	TABLESPACE_NAME	ESTIMATED_SIZE	LAST_DDL_TIME	CREATED	REFERENCED_OBJECTS	TRIGS
1	CUSTOMER	20	VALID	8	(null)	3	80	USERS	1600	15-APR-19	15-APR-19	0	0
2	DEPT	6	VALID	2	(null)	2	18	USERS	108	15-APR-19	15-APR-19	0	0
3	EMPLOYEE	41	VALID	12	(null)	2	81	USERS	3321	15-APR-19	15-APR-19	0	0
4	MACHINERY	15	VALID	5	(null)	1	39	USERS	585	15-APR-19	15-APR-19	1	1
5	ORDER_DETAILS	45	VALID	3	(null)	1	13	USERS	585	15-APR-19	15-APR-19	1	0
6	ORDER_PLACED	25	VALID	5	(null)	1	27	USERS	675	15-APR-19	15-APR-19	0	0
7	ORDER_PRIORITY	9	VALID	2	(null)	0	7	USERS	63	15-APR-19	15-APR-19	1	0
8	ORDER_RECIEVED	30	VALID	5	(null)	1	35	USERS	1050	15-APR-19	15-APR-19	3	3
9	PAYMENT	(null)	VALID	2	(null)	0	(null)	USERS	(null)	15-APR-19	15-APR-19	1	0
10	PRODUCT	15	VALID	3	(null)	2	20	USERS	300	15-APR-19	15-APR-19	1	0
11	RAW_MATERIAL	15	VALID	3	(null)	1	21	USERS	315	15-APR-19	15-APR-19	0	0
12	SHIPMENT	30	VALID	4	(null)	1	23	USERS	690	15-APR-19	15-APR-19	0	0
13	SUPPLIERS	10	VALID	7	(null)	2	62	USERS	620	15-APR-19	15-APR-19	0	0
14	TOOLS	15	VALID	4	(null)	1	26	USERS	390	15-APR-19	15-APR-19	0	0
15	TRANSP_VENDOR	10	VALID	7	(null)	2	66	USERS	660	15-APR-19	15-APR-19	0	0
16	WAREHOUSE	15	VALID	2	(null)	1	9	USERS	135	15-APR-19	15-APR-19	0	0

DDL

```
/* create department table*/

create table dept(
D_ID varchar2(3) CHECK (regexp_LIKE(D_ID, 'D[0-9]+?$')),
D_Name varchar2(30) unique,
constraint d_dept primary key(D_ID)
);
-----------------------

/* create Machinery table*/


create table machinery(
M_ID varchar2(4) CHECK (regexp_LIKE(M_ID, 'M[0-9]+?$')),
M_Name varchar2(40) not null,
purchase_date date,
price number(10,2),
constraint m_machinery primary key(M_ID),
service_due date
);
-----------------------
/*create tools*/

create table tools(
T_ID varchar2(4) CHECK (regexp_LIKE(T_Id, 'T[0-9]+?$'))constraint pk_tool_id primary key,
Tool_Name varchar2(100) not null,
Qty integer CHECK(Qty>0),
M_ID NOT NULL CONSTRAINT machine_id references machinery(M_ID)
);
-----------------------

/* create employee table*/

create table employee(
empno varchar2(10)CHECK (regexp_LIKE(empno, 'E[0-9]+?$')),
f_name varchar2(30) not null,
L_name varchar2(30),
D_ID varchar2(3),
Manager_id varchar2(5),
street_address varchar(100),
Zip varchar2(5) NOT NULL check (regexp_like(ZIP,'^[0-9]{5}')),
Phone varchar2(10) NOT NULL UNIQUE CHECK (regexp_like(Phone,'^[0-9]{10}')),
M_ID varchar2(3),
```

```
Designation varchar2(30),
Salary number(10,2),
hiredate date,
constraint emp primary key(empno),
CONSTRAINT machine_fk foreign key (M_ID) references machinery(M_ID),
constraint dept_id foreign key (D_ID) REFERENCES dept(D_ID),
CONSTRAINT manager_employee foreign key (Manager_id) references employee(empno)
);

----------

/* create customer table*/

Create table Customer(
C_ID varchar2(4) CHECK (regexp_like(C_ID,'^C[0-9]+?$')),
C_Name varchar2(30) NOT NULL,
Address varchar2(30),
A_City varchar2(30),
A_State Varchar2(30),
Zip varchar2(5) NOT NULL check (regexp_like(ZIP,'^[0-9]{5}')),
Phone varchar2(10) NOT NULL UNIQUE CHECK (regexp_like(Phone,'^[0-9]{10}')),
Email varchar2(30) NOT NULL UNIQUE ,
CONSTRAINT zip_customer CHECK(length(Zip)=5),
CONSTRAINT email_customer check ( Email LIKE '%@%.%' AND email NOT LIKE '@%' AND email NOT
LIKE '%@%@%' ),
CONSTRAINT pk_customer PRIMARY KEY (C_ID)
);
----------

/* create Product table*/


Create table Product(
Pdt_no varchar(4) CHECK (regexp_LIKE(Pdt_No, 'PR[0-9]+?$')) CONSTRAINT  pk_pdt primary key,
Pdt_name varchar2(30) NOT NULL UNIQUE,
Rate number(7,3));

----------

/* create Raw Material table*/


Create table Raw_material(
R_Id varchar2(4) CHECK (regexp_LIKE(R_ID, 'R[0-9]+?$')) CONSTRAINT pk_raw_id primary key,
R_Name varchar2(30) NOT NULL,
Pdt_No varchar2(5) references Product (Pdt_No)
);
```

/* create Order Received table*/

```
Create table Order_recieved(
PO_No varchar2(5) CHECK (regexp_LIKE(PO_No, 'PO[0-9]+?$')),
C_id varchar2(30),
O_Date date ,
Delivery_date date,
Order_status varchar2(30) ,
CONSTRAINT status_chk check (Order_status in ('In Process','Completed', 'Received')),
CONSTRAINT pk_order_recieved PRIMARY KEY (PO_NO),
CONSTRAINT fk_c_ID FOREIGN KEY (c_id) references Customer(c_id)
);
```

/* create Order Details table*/

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

/* create Shipment table*/

```
Create table Shipment(
SH_Id varchar(5) CHECK (regexp_LIKE(SH_Id, '^SH[0-9]+?$')) 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)
);
```

/* create Transport Vendor table*/

```
Create table Transp_Vendor(
T_Id varchar2(4) CHECK (regexp_LIKE(T_Id, '^TV[0-9]+?$')),
```

```
T_Name varchar2(30) NOT NULL,
T_Address varchar2(30),
T_City varchar2(30),
T_State Varchar2(30),
CONSTRAINT Trans_vend primary key (T_Id),
Zip varchar2(5) NOT NULL check (regexp_like(ZIP,'^[0-9]{5}')),
Phone varchar2(10) NOT NULL UNIQUE CHECK (regexp_like(Phone,'^[0-9]{10}')),
Email varchar2(30) NOT NULL UNIQUE check ( Email LIKE '%@%.%' AND email NOT LIKE '@%' AND email
NOT LIKE '%@%@%' )
);
```

```
/* alter supplier table*/

alter table Transp_Vendor drop column email
drop table transp_vendor


----------

/* create supplier table*/


Create table Suppliers(
S_Id varchar2(4) CHECK (regexp_LIKE(S_ID, '^S[0-9]+?$')) 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 varchar2(5) NOT NULL check (regexp_like(ZIP,'^[0-9]{5}')),
Phone varchar2(10) NOT NULL UNIQUE CHECK (regexp_like(Phone,'^[0-9]{10}'))
);

----------

/* create Orders placed table*/

Create table Order_placed(
OP_Id varchar2(8) CHECK (regexp_LIKE(OP_ID, '^PRID[0-9]+?$')) ,
CONSTRAINT pk_procurement_id primary key(OP_Id),
Raw_Id varchar2(4) references Raw_material (R_Id),
Order_date date,
Sup_Id varchar2(4) ,
Constraint Supp_ID foreign key (Sup_Id)references Suppliers(S_Id),
Qty integer NOT NULL CHECK(Qty>0)
);
----------
```

```
/* create Warehouse table*/

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 Payment table*/


create table Payment(
PO_NO varchar2(5),
Payment number(10,2),
constraint PO_NO_FK foreign key (PO_NO) references order_recieved(po_no)
);

----------

/* create Order Priority table*/


create table order_priority(
PO_NO varchar2(5),
O_priority varchar2(10),
constraint PO_NO_Foreign foreign key(PO_NO) references order_recieved(Po_NO)
);
```

DML

INSERT STATEMENTS :

CUSTOMER:

```
--Row 1
INSERT INTO PRAVEEN.CUSTOMER (C_ID, C_NAME, ADDRESS, A_CITY, A_STATE, EMAIL, ZIP, PHONE)
VALUES ('C01','Fairfield Industries','Hemenway
St','Montgomery','AL','fairfield@gmail.com','48887','7778889998');
--Row 2
INSERT INTO PRAVEEN.CUSTOMER (C_ID, C_NAME, ADDRESS, A_CITY, A_STATE, EMAIL, ZIP, PHONE)
VALUES ('C02','Atlas Group of Companies','Germain
St','Juneau','AK','atlas@gmail.com','54475','4545645444');
--Row 3
INSERT INTO PRAVEEN.CUSTOMER (C_ID, C_NAME, ADDRESS, A_CITY, A_STATE, EMAIL, ZIP, PHONE)
VALUES ('C03','Capricorn Tech','Belvidere
St','Phoenix','AZ','capricorn@gmail.com','55479','5465464888');
--Row 4
INSERT INTO PRAVEEN.CUSTOMER (C_ID, C_NAME, ADDRESS, A_CITY, A_STATE, EMAIL, ZIP, PHONE)
VALUES ('C04','Pragmatic Industries','Munro St','Little
Rock','AR','pragmatic@gmail.com','23312','2121212121');
--Row 5
INSERT INTO PRAVEEN.CUSTOMER (C_ID, C_NAME, ADDRESS, A_CITY, A_STATE, EMAIL, ZIP, PHONE)
VALUES ('C05','Phoenix Industries','Ponting St','Boise','ID','phoenix@gmail.com','14636','7889213545');
--Row 6
INSERT INTO PRAVEEN.CUSTOMER (C_ID, C_NAME, ADDRESS, A_CITY, A_STATE, EMAIL, ZIP, PHONE)
VALUES ('C06','Delena Industries','Gilchrist St','Chicago','IL','delena@gmail.com','47895','5345456455');
--Row 7
```

DEPT:

```
INSERT INTO PRAVEEN.DEPT (D_ID, D_NAME) VALUES ('D01','Quality');
--Row 2
INSERT INTO PRAVEEN.DEPT (D_ID, D_NAME) VALUES ('D02','Inventory Management');
--Row 3
INSERT INTO PRAVEEN.DEPT (D_ID, D_NAME) VALUES ('D03','Accounts');
--Row 4
INSERT INTO PRAVEEN.DEPT (D_ID, D_NAME) VALUES ('D04','Production');
--Row 5
INSERT INTO PRAVEEN.DEPT (D_ID, D_NAME) VALUES ('D05','Human Resources');
--Row 6
INSERT INTO PRAVEEN.DEPT (D_ID, D_NAME) VALUES ('D06','Delivery & Logistics');
```

TRIGGERS:

Payment Table Trigger :
/*When status of order received changes to Completed the trigger is executed and fills the payment table */

```
Create or replace trigger table_a_update
after insert or update on Order_recieved for each row
DECLARE Pay NUMBER(10,2);
begin
if :new.Order_status='Completed' then

  SELECT SUM(RATE *QTY) AS PAYMENT
  INTO PAY
  FROM (select * from order_details
       left join product ON
       product.PDT_NO = order_details.PDT_NO
      )
  where PO_NO = :new.PO_No
  GROUP BY PO_NO;

 insert into payment
  (PO_no, payment)
 values
  (:new.PO_No, pay);

end if;

end;
```

Order_Priority table trigger

/*When the status of received order changes to 'In Process' then the trigger executes and compare the expected date of all the In Progress orders and ranks them in order of priority to fill Order_Priority table*/

```
create or replace trigger table_priority after insert or update on Order_recieved for each row
begin

if :new.Order_status ='In Process' then

  insert into order_priority
   (PO_no)
 values
   (:new.PO_No);

elsif :new.Order_status ='Completed' then

   delete from order_priority
   where PO_no = :new.PO_No;

end if;
end;
```

Update:

Update payment table:

```
update  Order_recieved
set order_status='Completed' where
po_no='PO1'

update  Order_recieved
set order_status='Completed' where
po_no='PO2'


/*update priority to fill the rank*/
update order_priority a
set a.o_Priority = (select b.rank from (select PO_no, RANK() OVER (ORDER BY Delivery_date) as rank
                    from Order_recieved
                    where Order_status ='In Process') b
                    where b.PO_no = a.PO_no)
```