Project Report On

Database Management System Of

Store Order Handling

Project By

Sushant S

Tushar Nirkulwar

Saurabh Deshmane

TABLE OF CONTENTS

ABSTRACT

To investigate a real-life problem in an organization and take the necessary steps to provide them with a database solution. We are provided with sample case study that we will use to build our solution. We had made the necessary assumptions and enrich the case environment using details that we have researched about current database implementations in organizations.

CASE STUDY

A store selling computer accessories handles orders of computer accessories like colour monitors, printers, chargers etc. The store has multiple vendors who provide specific product(s). The name and the selling price of every product are mentioned on the packing. The store needs to keep the information about the quantity on hand for every product.

The store has sales consultants and managers to interact with the customers. Information about the base location of the employee needs to be stored. Employees get commission for every product that they sell. Per product commission is fixed for an employee. For the same product, sales consultants get marginally more commission than the managers.

The store keeps details of every customer for various types of communications. The customer can purchase the goods on credit also. The bill provided to the customer includes order number, order date, name of the employee who catered to the order, customer name, city of the customer, city in which the order is placed and the order details. Order details contain the product name, class of the product and quantity ordered.

Section 1

PROJECT PLAN

Problem Statement:

The AZ Electronics solutions is one of biggest computer hardware and accessories store in our city. Store has multiple branches across the city and also have pan India presence in different states. They cater to huge customer base because of the reliable customer service and quality of products. Even though they provide services in computer industry they are facing many issues in manging their huge customer and vendor base requirements on time. Few of their branches have negative sales because of order delivery and handling constraints in backend operations across multiple locations.

It is observed that they are using Flat file system to store their customer data regarding purchase and delivery of items. Similar case is with vendors' data as they are not being able to keep track of availability of different products among different vendors and manage its supply chain efficiently. This has created issues like delayed products delivery, how mange huge customer database, security issue of huge data, lot people in upper management don't know how tackle this situation database management since there is no centralized system in place from which we can draw meaning full conclusions which will fast-track business growth.

We are required to collect vendor communication data, customer data, employ data and tracking of that data in one centralized database so that by manipulating it we can generate some good results

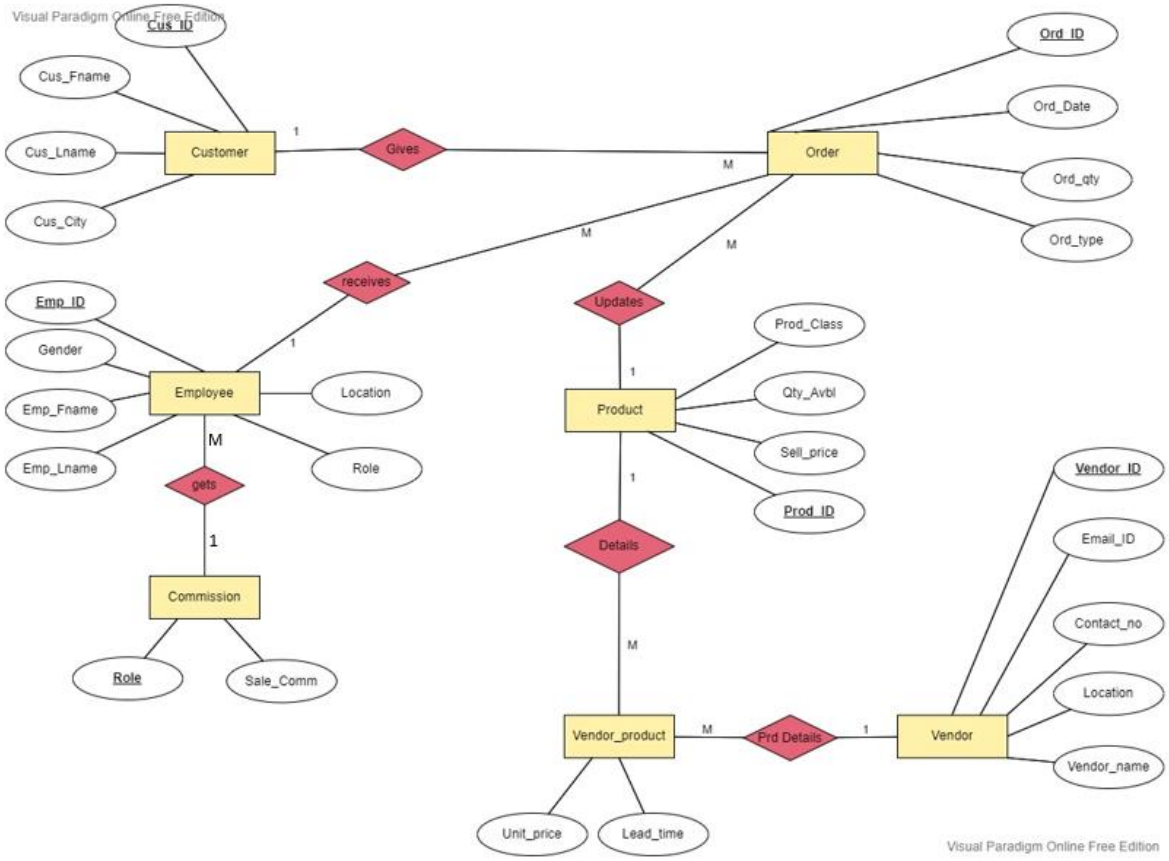
with single click. This will also ensure safe and organize data access for different people in business which will enhance productivity further.

Section 2

LOGICAL DATA MODEL

Logical data model has been shown in following entity relationship diagram. There various are components of ER diagram which explained below:

1. **Entity Identification:** We have 7 entities in ER diagram. Which includes Customer, order, Product, Vendor, Vendor products, Employee, commission.
2. **Relationship Identification:** Here Customer entity is giving the order which gets updated in product table. Order is received by employee and he gets commission based on order count he handles. Product is dispatched from vendor with product details.
3. **Cardinality Identification:** Customer has one to many relationships with order which means one customer can give multiple orders, order has many to one relationship with employee; which means multiple order can be taken one employee. Similarly, product has one to many relationships with vendor product.



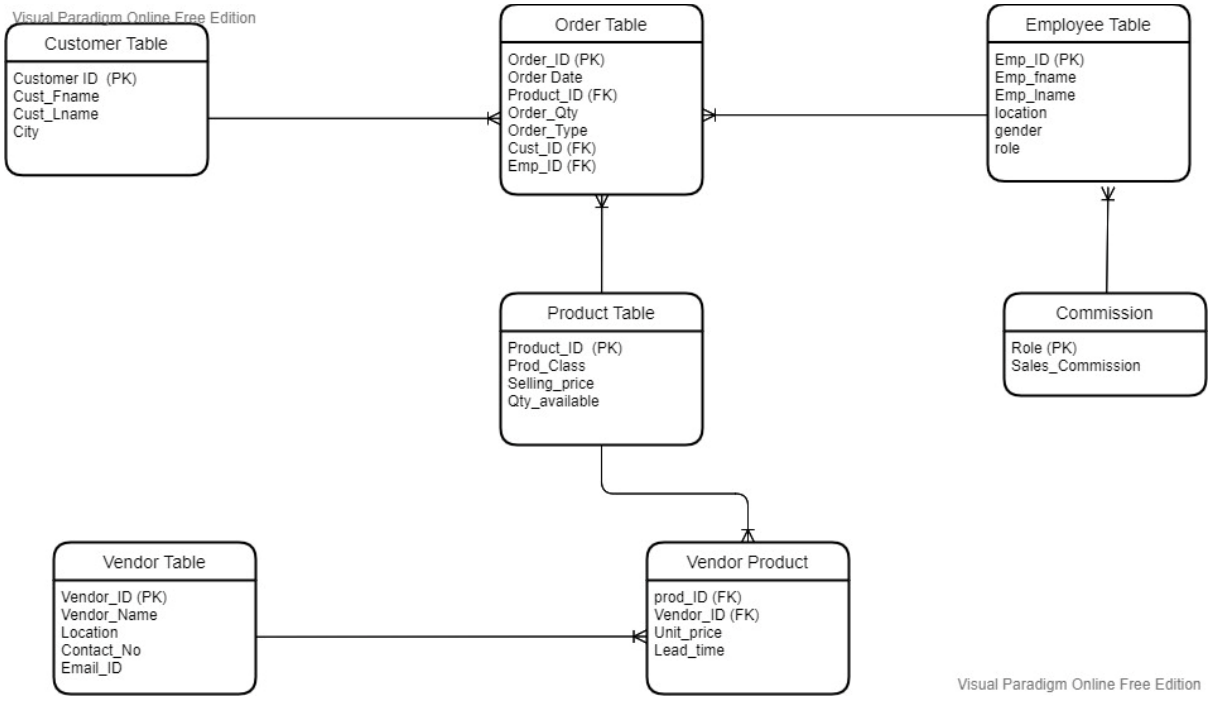
ER diagram Relationship Identification

Section 3

PHYSICAL DATA MODEL

This model explains attributes related to all 7 entities along with their cardinality relationships. Here we can see Customer table has one primary key assigned to customer id. Order table has one primary key as order id and three foreign keys for product ID, customer ID, Employee ID.

Product table has one Primary key as Product ID. Vendor table and vendor product table has been joined with foreign keys as product ID, vendor ID. Employee table is joined with commissions table with Role as primary key in commission table and Employee ID as primary key in employee table.



ER Model with Identified Attributes

Section 4

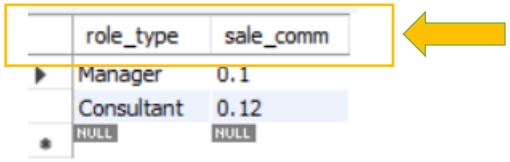
DATABASE SYSTEM

A) List of tables created in database and their purpose

1. Commission Table

   Query:
   CREATE TABLE commission ***(role_type*** *ENUM ('Manager', 'Consultant'),* ***sale_comm*** *float, PRIMARY KEY (role_type));*

   Result:

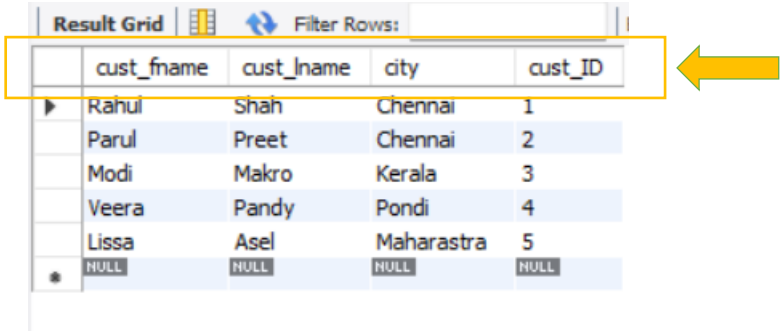


role_type	sale_comm
Manager	0.1
Consultant	0.12
NULL	NULL

2. Customer Table

   Query:
   *CREATE TABLE customer (****cust_fname*** *VARCHAR(30) NOT NULL,* ***cust_lname*** *VARCHAR(30) NOT NULL****, city*** *VARCHAR(30),* ***cust_ID*** *INT* ***NOT NULL UNIQUE****, PRIMARY KEY (cust_ID), INDEX (cust_ID));*

   Result:



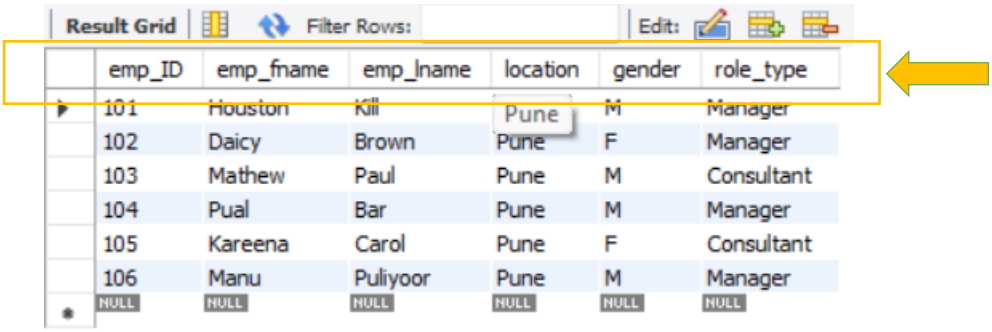
Result Grid | Filter Rows:

cust_fname	cust_lname	city	cust_ID
Rahul	Shah	Chennai	1
Parul	Preet	Chennai	2
Modi	Makro	Kerala	3
Veera	Pandy	Pondi	4
Lissa	Asel	Maharastra	5
NULL	NULL	NULL	NULL

3. Employee Table

   Query:
   *CREATE TABLE employee (****emp_ID*** *INT UNIQUE****, emp_fname*** *VARCHAR(50),*
   emp_lname *VARCHAR(50),* ***location*** *VARCHAR(30),*
   gender *ENUM ('M','F'),*
   role_type *ENUM ('Manager', 'Consultant'),*
   PRIMARY KEY (emp_ID));

Result:

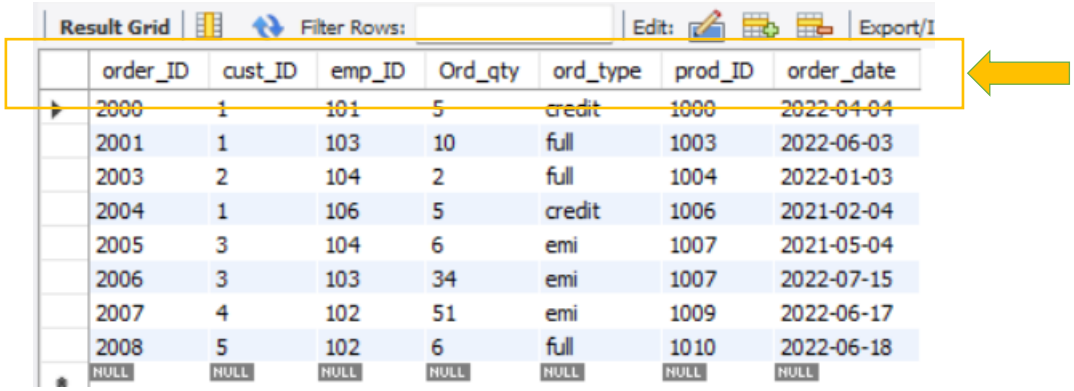


emp_ID	emp_fname	emp_lname	location	gender	role_type
101	Houston	Kill	Pune	M	Manager
102	Daicy	Brown	Pune	F	Manager
103	Mathew	Paul	Pune	M	Consultant
104	Pual	Bar	Pune	M	Manager
105	Kareena	Carol	Pune	F	Consultant
106	Manu	Puliyoor	Pune	M	Manager
NULL	NULL	NULL	NULL	NULL	NULL

4. Order Table

Query:
*CREATE TABLE order_t(**order_ID** INT **NOT NULL UNIQUE**, **cust_ID** int **NOT NULL**, **emp_ID** int NOT NULL, **Ord_qty** INT **NOT NULL**, **ord_type** ENUM ('credit','full','emi') NOT NULL, **prod_ID** INT NOT NULL, **order_date** DATE NOT NULL, PRIMARY KEY (order_ID), FOREIGN KEY (prod_ID) REFERENCES product (prod_ID), FOREIGN KEY (cust_ID) REFERENCES customer (cust_ID), FOREIGN KEY (emp_ID) REFERENCES employee(emp_ID));*

Result:

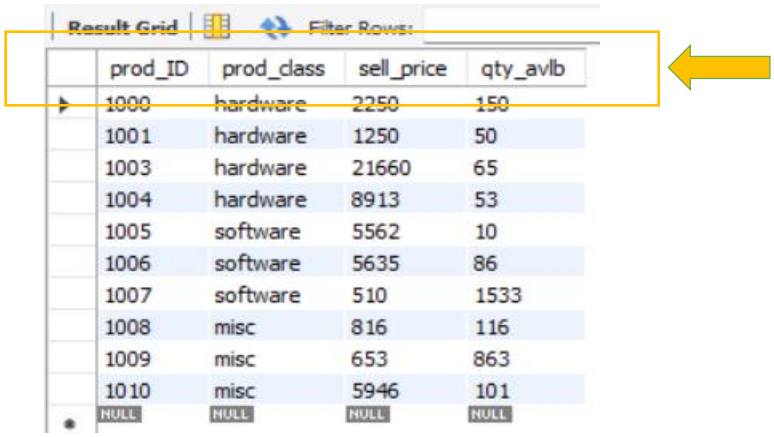


order_ID	cust_ID	emp_ID	Ord_qty	ord_type	prod_ID	order_date
2000	1	101	5	credit	1000	2022-04-04
2001	1	103	10	full	1003	2022-06-03
2003	2	104	2	full	1004	2022-01-03
2004	1	106	5	credit	1006	2021-02-04
2005	3	104	6	emi	1007	2021-05-04
2006	3	103	34	emi	1007	2022-07-15
2007	4	102	51	emi	1009	2022-06-17
2008	5	102	6	full	1010	2022-06-18
NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. Product Table

Query:
*CREATE TABLE product (**prod_ID** INT **UNIQUE**,*
***prod_class**, ENUM('hardware','software','misc'),*
***sell_price** float,*
***qty_avlb** int,*
PRIMARY KEY (prod_ID));

Result:



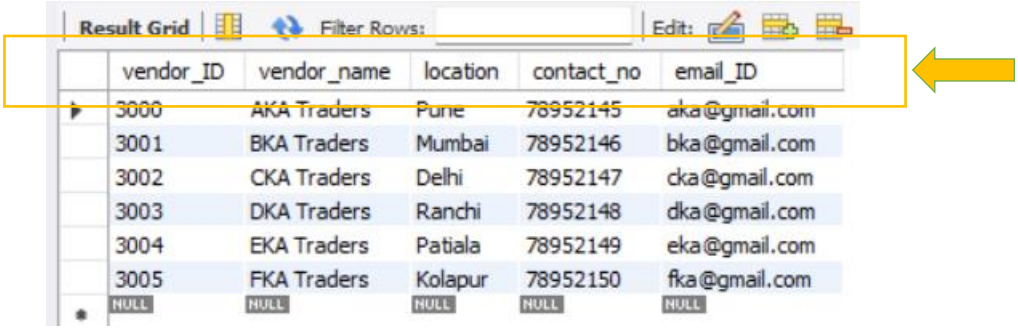
prod_ID	prod_class	sell_price	qty_avlb
1000	hardware	2250	150
1001	hardware	1250	50
1003	hardware	21660	65
1004	hardware	8913	53
1005	software	5562	10
1006	software	5635	86
1007	software	510	1533
1008	misc	816	116
1009	misc	653	863
1010	misc	5946	101
NULL	NULL	NULL	NULL

6. Vendor Table

   Query:

   *CREATE TABLE vendor (**vendor_ID** INT **UNIQUE**, **vendor_name** varchar(20), location varchar(20), **contact_no** int, **email_ID** varchar(20), PRIMARY KEY (vendor_ID));*

   Result:



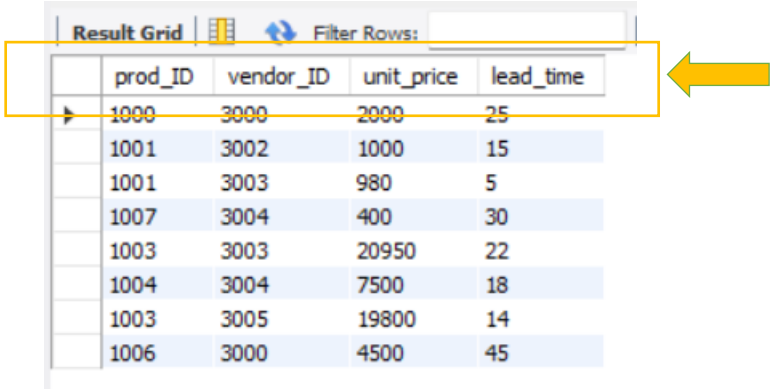
vendor_ID	vendor_name	location	contact_no	email_ID
3000	AKA Traders	Pune	78952145	aka@gmail.com
3001	BKA Traders	Mumbai	78952146	bka@gmail.com
3002	CKA Traders	Delhi	78952147	cka@gmail.com
3003	DKA Traders	Ranchi	78952148	dka@gmail.com
3004	EKA Traders	Patiala	78952149	eka@gmail.com
3005	FKA Traders	Kolapur	78952150	fka@gmail.com
NULL	NULL	NULL	NULL	NULL

7. Vendor Product Information Table

   Query:

   *CREATE TABLE vendor_prod (**prod_ID** int, **vendor_ID** INT, **unit_price** float, **lead_time** INT, FOREIGN KEY (prod_ID) REFERENCES product (prod_ID),FOREIGN KEY (vendor_ID) REFERENCES vendor(vendor_ID));*

   Result:



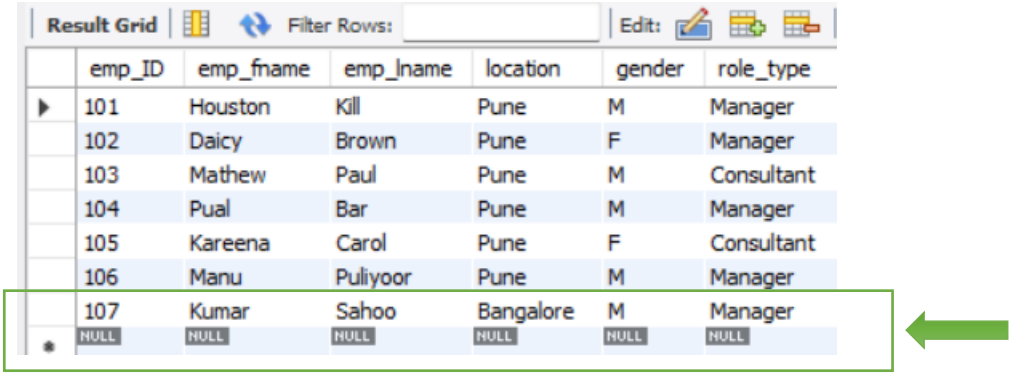
prod_ID	vendor_ID	unit_price	lead_time
1000	3000	2000	25
1001	3002	1000	15
1001	3003	980	5
1007	3004	400	30
1003	3003	20950	22
1004	3004	7500	18
1003	3005	19800	14
1006	3000	4500	45

B) List of queries to address strategic management requirements

1. Query 1: Insert a record in employee table

   Query: insert into employee values (107, 'Kumar','Sahoo', 'Bangalore', 'M','Manager');

   Result:

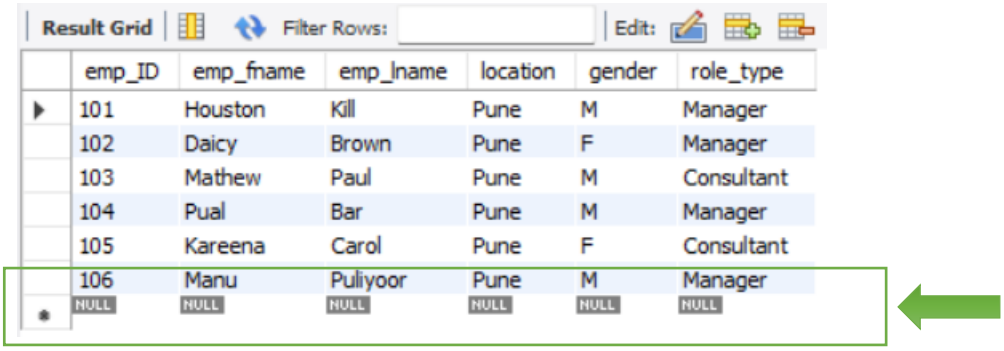


emp_ID	emp_fname	emp_lname	location	gender	role_type
101	Houston	Kill	Pune	M	Manager
102	Daicy	Brown	Pune	F	Manager
103	Mathew	Paul	Pune	M	Consultant
104	Pual	Bar	Pune	M	Manager
105	Kareena	Carol	Pune	F	Consultant
106	Manu	Puliyoor	Pune	M	Manager
107	Kumar	Sahoo	Bangalore	M	Manager

2. Query 2: Delete the new record inserted into employee table

   Query: DELETE FROM employee WHERE emp_ID = 107;

   Result:

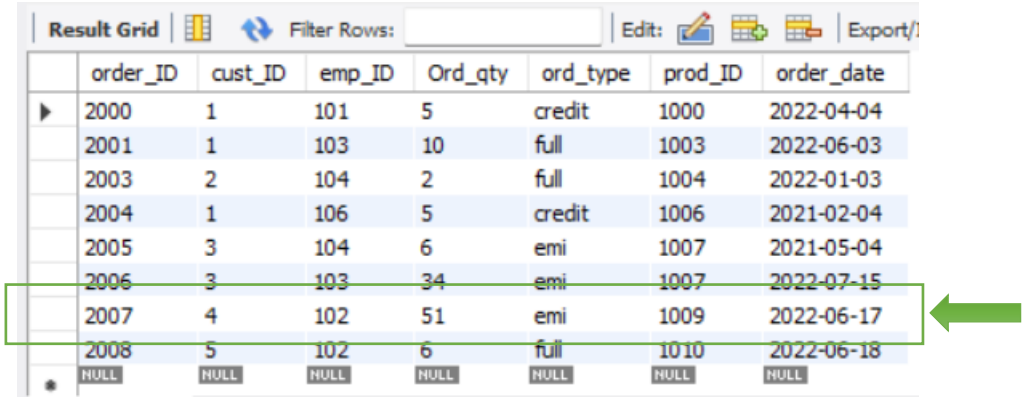


emp_ID	emp_fname	emp_lname	location	gender	role_type
101	Houston	Kill	Pune	M	Manager
102	Daicy	Brown	Pune	F	Manager
103	Mathew	Paul	Pune	M	Consultant
104	Pual	Bar	Pune	M	Manager
105	Kareena	Carol	Pune	F	Consultant
106	Manu	Puliyoor	Pune	M	Manager

3. Query 3: Update order date for any record in the order table

   Query: UPDATE order_t SET order_date = '2022-06-17' WHERE order_ID = 2007;

   Result:



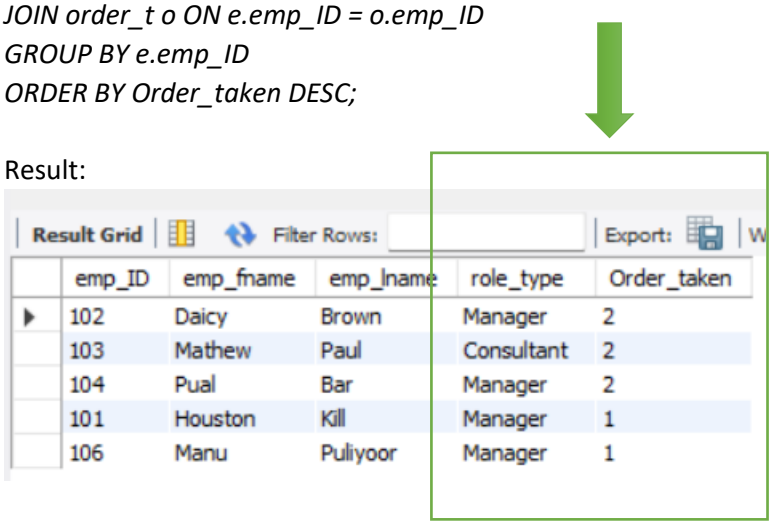
order_ID	cust_ID	emp_ID	Ord_qty	ord_type	prod_ID	order_date
2000	1	101	5	credit	1000	2022-04-04
2001	1	103	10	full	1003	2022-06-03
2003	2	104	2	full	1004	2022-01-03
2004	1	106	5	credit	1006	2021-02-04
2005	3	104	6	emi	1007	2021-05-04
2006	3	103	34	emi	1007	2022-07-15
2007	4	102	51	emi	1009	2022-06-17
2008	5	102	6	full	1010	2022-06-18

4. Query 4: Give total orders taken by each employee in descending order

   Query:

   *SELECT * FROM employee;*
   *SELECT * FROM order_t;*
   SELECT e.emp_ID, e.emp_fname, e.emp_lname, e.role_type,
   COUNT(o.order_ID) Order_taken
   FROM employee e
   JOIN order_t o ON e.emp_ID = o.emp_ID
   GROUP BY e.emp_ID
   ORDER BY Order_taken DESC;

   Result:



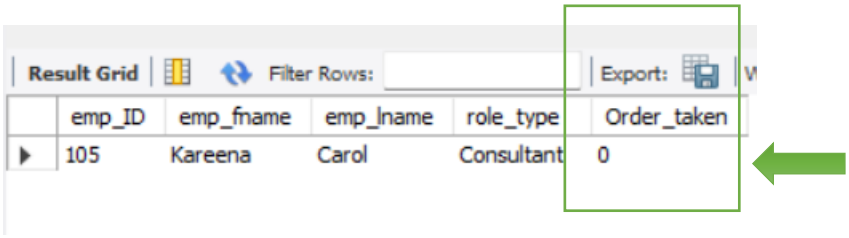
Result Grid | Filter Rows: | Export:

emp_ID	emp_fname	emp_lname	role_type	Order_taken
102	Daicy	Brown	Manager	2
103	Mathew	Paul	Consultant	2
104	Pual	Bar	Manager	2
101	Houston	Kill	Manager	1
106	Manu	Puliyoor	Manager	1

5. Query 5: Get the name of employee who has not taken any order till date

   Query:

   SELECT e.emp_ID, e.emp_fname, e.emp_lname, e.role_type,
   COUNT(o.order_ID) as Order_taken
   FROM employee e
   LEFT JOIN order_t o ON e.emp_ID = o.emp_ID
   GROUP BY e.emp_ID
   HAVING Order_taken = 0
   ORDER BY Order_taken DESC;



Result Grid | Filter Rows: | Export:

emp_ID	emp_fname	emp_lname	role_type	Order_taken
105	Kareena	Carol	Consultant	0

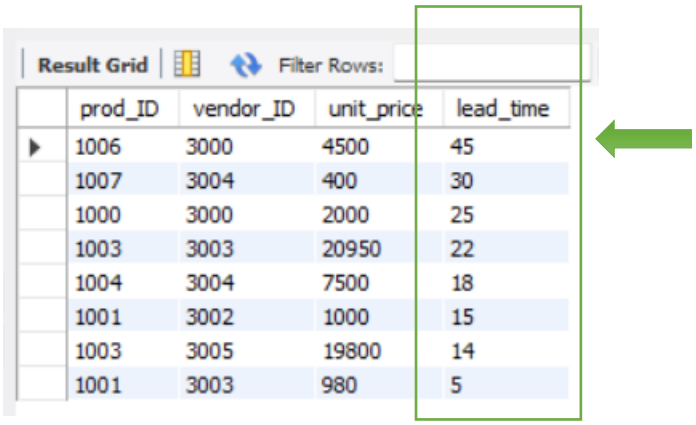
6. Query 6: Arrange the product in descending order of lead time to delivery

   Query:

   *SELECT * FROM vendor_prod*

   ORDER BY lead_time DESC;

Result:



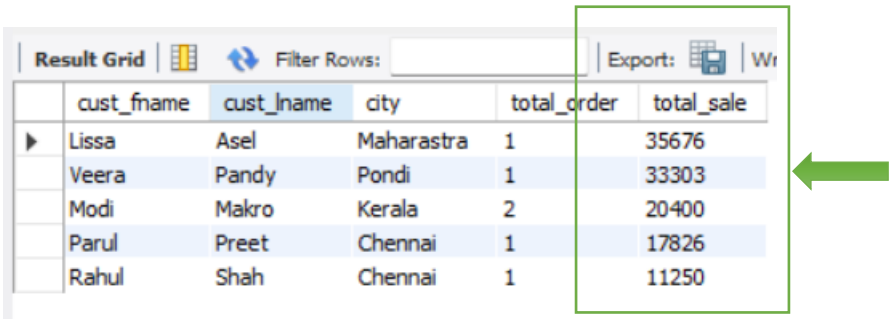
prod_ID	vendor_ID	unit_price	lead_time
1006	3000	4500	45
1007	3004	400	30
1000	3000	2000	25
1003	3003	20950	22
1004	3004	7500	18
1001	3002	1000	15
1003	3005	19800	14
1001	3003	980	5

7. Query 7: Display the total sales for each customer in descending order - SUB Query

Query:

```
SELECT c.cust_fname, c.cust_lname, c.city, o.total_order, o.total_sale
FROM (SELECT o.cust_ID, o.prod_ID,
COUNT(o.order_ID) total_order,(SUM(o.Ord_qty) * p.sell_price) total_sale
FROM Order_t O
JOIN product p ON o.prod_ID = p.prod_ID
GROUP BY prod_ID) o
JOIN customer c  ON o.cust_ID = c.cust_ID
GROUP BY o.cust_ID
ORDER BY o.total_sale DESC;
```

Result:



cust_fname	cust_lname	city	total_order	total_sale
Lissa	Asel	Maharastra	1	35676
Veera	Pandy	Pondi	1	33303
Modi	Makro	Kerala	2	20400
Parul	Preet	Chennai	1	17826
Rahul	Shah	Chennai	1	11250

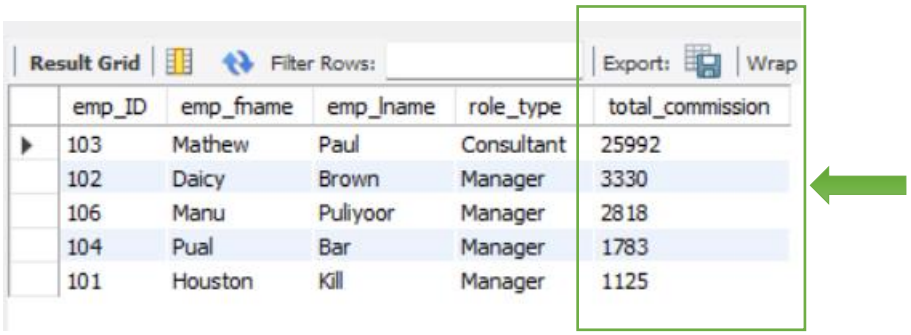
8. Query 8: Display commission earned by each employee in descending order

Query:

```
SELECT em.emp_ID, em.emp_fname, em.emp_lname, em. role_type, round ((om. total_sale *
em. sale_comm)) total_commission
FROM (SELECT o.emp_ID, o.prod_ID,
COUNT(o.order_ID) total_order, (SUM(o.Ord_qty) * p.sell_price) total_sale
FROM Order_t O
```

JOIN product p ON o.prod_ID = p.prod_ID
GROUP BY o.prod_ID) om JOIN (SELECT e.emp_ID, e.emp_fname, e.emp_lname, e.location, e.role_type, c.sale_comm
FROM employee e
JOIN commission c ON e.role_type = c.role_type) em ON om.emp_ID = em.emp_ID
group by om.emp_ID
ORDER BY total_commission DESC;

Result:



Result Grid | Filter Rows: | Export: | Wrap

emp_ID	emp_fname	emp_lname	role_type	total_commission
103	Mathew	Paul	Consultant	25992
102	Daicy	Brown	Manager	3330
106	Manu	Puliyoor	Manager	2818
104	Pual	Bar	Manager	1783
101	Houston	Kill	Manager	1125

CONCLUSION

- This project consolidates the idea that how database management system creates more integrated picture of your operations by easily illustrating how processes in one segment of the organization affect other segments. What once was done completely manually now can be fully automated and more accurate.
- We have successfully demonstrated that how one centralized database system in place can help businesses to increase their productivity and efficiency in longer run. It empowers people to spend more time on high-value activities and strategic initiatives, and less time cleaning data and manually scrubbing lists like flat file system.
- This also explains the fact that by manipulating data by various angles we can minimize our mistakes while doing any tasks in business environment to boost healthy work culture. A database management system helps provide a framework to facilitate data quality initiatives.
- In this unified database system store chain not only being able keep track of customer requirements but they are also managing their supply chain efficiently by analysing key hiccups from database. Database management systems also provided a better framework for the enforcement of privacy and security policies.
- We also showed that in datacentric world how important it is to store customer data in an organized manner. By providing quick solutions to database queries, a data management system enables faster access to more accurate data. End users, like salespeople, are able to speed up sales cycles and get more accurate in their sales prospecting.