



System for Material Management

Advanced Database Management System

12/3/17

Group 6:

Abhishek

Akash Gandhi

Ravali Kotha

Preksha

Contents

Introduction:	2
Database Design:	3
Database Provisioning:	21
Data Generation & Loading:	24
Performance Tuning:	28
Visualization:	42
Application Design:	45
Challenges:	48
Future Scope:	49
References:	49

Introduction:

The report is for a Material Management system for a small- scale Clothing company, it captures the purchase and sales details of the company. This system optimizes the production, sales and distribution channels for the company.

The purpose of the database is to replace the book keeping mechanism for the company to maintain data generated by customers and employees and facilitate smooth running of the business. It helps in improving the operating efficiency for the company.

The flow of operations is triggered by the stakeholders Customers, Employees, Management. The system is designed to capture the steps involved in ‘Procure to pay’ business process.

When the transaction is triggered by an Employee in the system, below are the steps carried out:

1. Once an employee joins the company, Login credentials are created for the employee.
2. The employee gathers the requirement of materials in the plants and creates a Purchase order for a Vendor.
3. The Purchase order created requires to be approved by the manager.
4. If the PO is rejected, a new PO maybe required with new specifications or the requirement was not met. If the PO is approved, the vendor supplies the materials.

When the transaction is initiated by the customer, below are the steps carried out:

1. Once the customer creates an account. If a customer is a member special discounts are provided based on the membership type-Silver, Gold or Platinum.
2. The customer creates a Sales Order from the listed finished product, Shipping details and details of any insurance plan for the product.
3. The customer can choose from multiple insurance providers and can enter unique shipping details for each transaction.



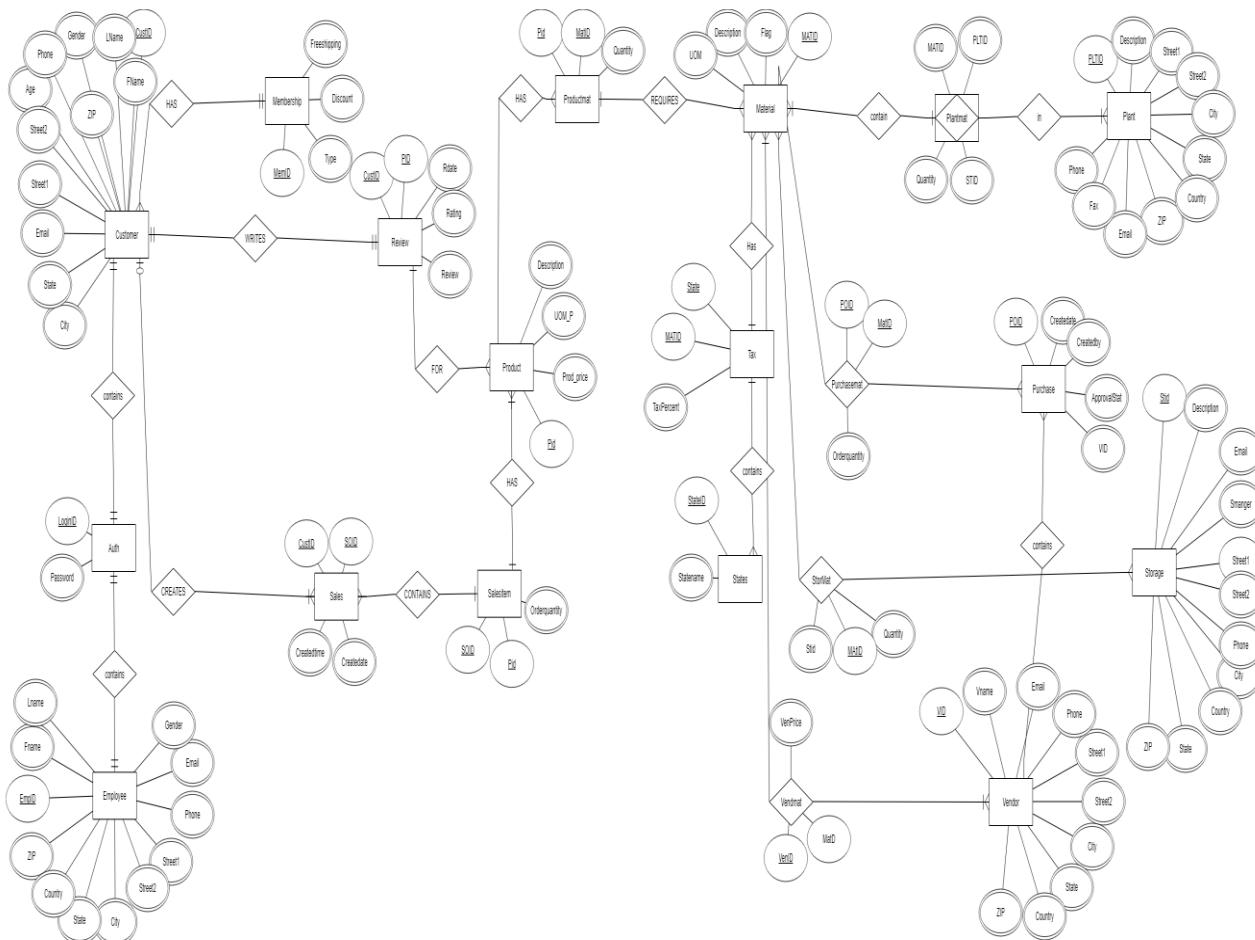
Requirements:

1. To store details of Purchase orders created by employees and Sales orders created by customers.
2. To record all the inventory available in each of the store locations and plants.
3. To capture the details of the vendors and insurance providers in the system. Improve the ease of access to update details of any of the stakeholders in the system.
4. To be able to create visualization on the data for leadership of the company.

Database Design:

Conceptual Design:

The conceptual design consists of identifying the requirements and creating the ER- diagram for the database. ER diagram is a visual representation of data using conventions to describe the relationship between the data. Below is the ER diagram for the database.

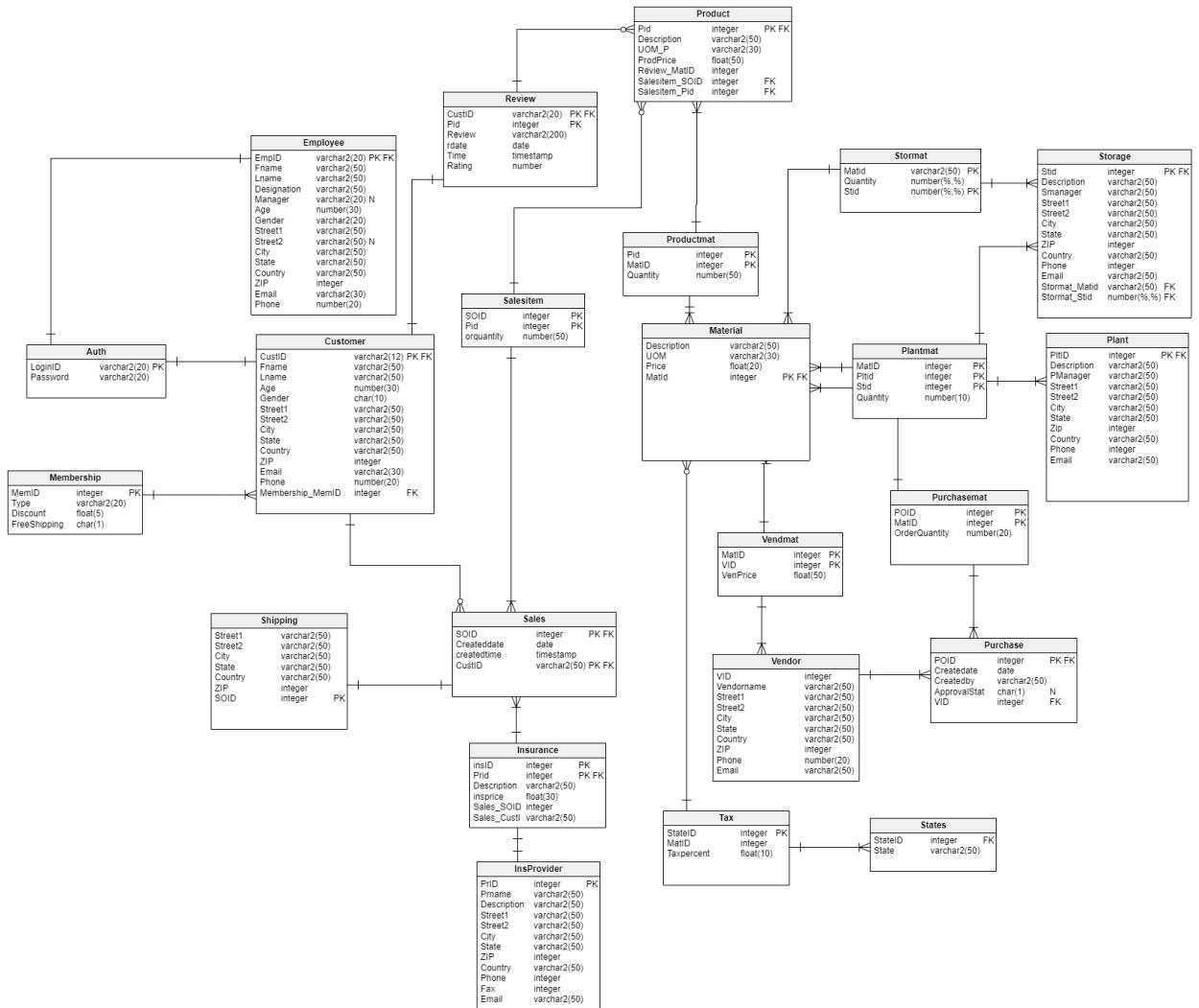


Logical Design:

The logical design is the translation of the ER diagram into tables. The database contains 23 tables that have designed to capture activities in managing inventory and retail activities for the business.

Assumptions for database:

1. One plant has many storage locations, but one location is for one plant.
2. The requisitions and quotations are pre-determined, and Purchase Order created needs approval.
3. One customer can have many Sales Order, but one Sales Orders will belong to one customer.
4. Prices will be different in different states.
5. Shipping details can be given for each Sales Orders. There is one to one mapping between Shipping and Sales Orders.
6. Membership is of three types Silver, Gold, Platinum. People with membership get free shipping during the tenure and special discounts.
7. Insurance
8. The database doesn't support returns and replacements.
9. Taxes differ from state to state and are uniform for all the materials within the state.



Physical Design:

The physical design involves creating the relational database in the system. We have used MySQL for the database as it is a stable, powerful and reliable solution. Below is a brief description of tables created in the database along with the attributes.

TABLES

1. **AUTH**: It is the authentication table which contains the Login ID and Password columns. The Login ID's are generated for each of the employees and customers.

```

CREATE TABLE Auth (
    LoginID varchar(20) NOT NULL,
    Password varchar(20) NOT NULL,
    CONSTRAINT Auth_pk PRIMARY KEY (LoginID)
);

```

The screenshot shows a database table creation interface. At the top, it says "Table Name: Auth" and "Schema: matmgmt". Below this is a table structure with columns for "Column Name", "Datatype", and various constraints like PK, NN, UQ, B, UN, ZF, AI, G, and "Default/Expression". Two rows are present: one for "LoginID" (datatype VARCHAR(20), PK, NN checked) and one for "Password" (datatype VARCHAR(20), NULL). Below the table structure are input fields for adding new columns: "Column Name", "Data Type", "Default", and checkboxes for "Storage" (Virtual, Stored), "Primary Key", "Not Null", "Unique", "Binary", "Unsigned", "Zero Fill", "Auto Increment", and "Generated".

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
LoginID	VARCHAR(20)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"					
Password	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name:	Data Type:
Collation:	Default:
Comments:	Storage: <input type="radio"/> Virtual <input checked="" type="radio"/> Stored <input type="checkbox"/> Primary Key <input type="checkbox"/> Not Null <input type="checkbox"/> Unique <input type="checkbox"/> Binary <input type="checkbox"/> Unsigned <input type="checkbox"/> Zero Fill <input type="checkbox"/> Auto Increment <input type="checkbox"/> Generated

- EMPLOYEE: The employee table contains the Employee ID, Name, Designation, Manager and Address fields for the employee.

```
CREATE TABLE Employee (
```

```

  EmpID varchar(20) NOT NULL,
  Fname varchar(50) NOT NULL,
  Lname varchar(50) NOT NULL,
  Designation varchar(50) NOT NULL,
  Manager varchar(20) NULL,
  Age numeric(30) NOT NULL,
  Gender varchar(20) NOT NULL,
  Street1 varchar(50) NOT NULL,
  Street2 varchar(50) NULL,
  City varchar(50) NOT NULL,
  State varchar(50) NOT NULL,
  Country varchar(50) NOT NULL,
  ZIP integer NOT NULL,
  Email varchar(30) NOT NULL,
  Phone numeric(20) NOT NULL,
  CONSTRAINT Phone UNIQUE (Phone),

```

```

CONSTRAINT Employee_pk PRIMARY KEY (EmpID)
);

```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
EmpID	VARCHAR(20)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Fname	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Lname	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Designation	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Manager	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Age	DECIMAL(30,0)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Gender	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL					

Column Name: Gender Data Type: VARCHAR(20)
 Collation: Table Default Default:
 Comments:
 Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

- CUSTOMER: the customer table contains the Customer ID, Name , Address fields for the customer.

```

CREATE TABLE Customer (
  CustID varchar(12) NOT NULL,
  Fname varchar(50) NOT NULL,
  Lname varchar(50) NOT NULL,
  Age numeric(30) NOT NULL,
  Gender char(10) NOT NULL,
  Street1 varchar(50) NOT NULL,
  Street2 varchar(50) NOT NULL,
  City varchar(50) NOT NULL,
  State varchar(50) NOT NULL,
  Country varchar(50) NOT NULL,
  ZIP integer NOT NULL,
  Email varchar(30) NOT NULL,
  Phone numeric(20) NOT NULL,
  Membership_MemID integer NOT NULL,
  CONSTRAINT Phone UNIQUE (Phone),
  CONSTRAINT Customer_pk PRIMARY KEY (CustID)
);

```

Table Name: Customer Schema: matmgmt

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
CustID	VARCHAR(12)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Fname	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Lname	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Age	DECIMAL(30,0)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Gender	CHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Street1	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Street2	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL					

Column Name: Data Type: Default: Storage:

Collation: Virtual Stored
Comments: Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

4. REVIEW: The review table captures the product reviews by each of the customers using the columns Customer ID, Product ID, Descriptive review and Rating.

```
CREATE TABLE Review (
    CustID varchar(20) NOT NULL,
    Pid integer NOT NULL,
    Review varchar(200) NOT NULL,
    rdate date NOT NULL,
    Time timestamp NOT NULL,
    Rating numeric NOT NULL,
    CONSTRAINT Review_pk PRIMARY KEY (CustID,Pid)
);
```

Table Name: Review Schema: matmgmt

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
CustID	VARCHAR(20)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
MatID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Review	VARCHAR(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
reviewdate	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
reviewtime	TIMESTAMP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Rating	DECIMAL(10,0)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Column Name: Data Type: Default: Storage:

Collation: Virtual Stored
Comments: Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

5. MATERIAL: The material table captures all the material ID, description and unit of measurement of the material along with price details.

```
CREATE TABLE Material (
    Description varchar(50) NOT NULL,
    UOM varchar(30) NOT NULL,
    Price float(20) NOT NULL
);
```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MatID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Description	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
UOM	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Price	FLOAT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: Data Type:
 Collation: Default:
 Comments: Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

6. PRODUCT:

The table provides information about the products manufactured along with the details like Product Id ,Name, Price, Materials used in manufacturing.

```
CREATE TABLE Product (
    Pid integer NOT NULL,
    Description varchar(50) NOT NULL,
    UOM_P varchar(30) NOT NULL,
    ProdPrice float(50) NOT NULL,
    MatID integer NOT NULL,
    CONSTRAINT Product_pk PRIMARY KEY (Pid)
);
```

Table Name: Product Schema: matmgmt

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Pid	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Description	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
UOM_P	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
ProdPrice	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: Data Type: Default: Storage:

Collation: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

Comments:

7. PRODUCTMAT:

This table contains the relationship between the products and the materials used to manufacture each of the products.

```
CREATE TABLE Productmat (
    Pid integer NOT NULL,
    MatID integer NOT NULL,
    Quantity numeric(50) NOT NULL,
    CONSTRAINT Productmat_pk PRIMARY KEY (Pid,MatID)
);
```

Table Name: Productmat Schema: matmgmt

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Pid	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
MatID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Quantity	DECIMAL(50,0)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: Data Type: Default: Storage:

Collation: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

Comments:

8. STORAGE:

This contains the details of each of the storage locations like Storage ID, Name, Address, Storage Manager.

```

CREATE TABLE Storage (
    Stid integer NOT NULL,
    Description varchar(50) NOT NULL,
    Smanager varchar(50) NOT NULL,
    Street1 varchar(50) NOT NULL,
    Street2 varchar(50) NOT NULL,
    City varchar(50) NOT NULL,
    State varchar(50) NOT NULL,
    ZIP integer NOT NULL,
    Country varchar(50) NOT NULL,
    Phone integer NOT NULL,
    Email varchar(50) NOT NULL,
    CONSTRAINT Storage_pk PRIMARY KEY (Stid)
);

```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Stid	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Description	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Smanager	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Street1	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Street2	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
City	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
State	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Table Name: Storage Schema: matmgmt

Column Name: Data Type: Default:

Collation: Storage:

Comments: Primary Key: Not Null: Unique:

Binary: Unsigned: Zero Fill:

Auto Increment: Generated:

9. STORAGEMAT: The table contains the mapping of storage locations to material.

```

CREATE TABLE Stomat (
    Matid varchar(50) NOT NULL,
    Quantity numeric NOT NULL,
    Stid integer NOT NULL,
    CONSTRAINT Stomat_pk PRIMARY KEY (Matid,Stid)
);

```

The screenshot shows a database table creation interface. At the top, it displays "Table Name: Stomat" and "Schema: matmgmt". The main area shows a table structure with three columns: MatID (INT(11)), Stid (INT(11)), and Quantity (DECIMAL(10,0)). Below this, there are fields for defining new columns: "Column Name", "Data Type", "Default", and checkboxes for "Storage" options like Virtual, Primary Key, Not Null, Unique, Binary, Unsigned, Zero Fill, Auto Increment, and Generated.

10. PLANT:The table contains the details of the Plant like Plant ID, Description, Manger for the Plant.

```

CREATE TABLE Plant (
    PltID integer NOT NULL,
    Description varchar(50) NOT NULL,
    PManager varchar(50) NOT NULL,
    Street1 varchar(50) NOT NULL,
    Street2 varchar(50) NOT NULL,
    City varchar(50) NOT NULL,
    State varchar(50) NOT NULL,
    Zip integer NOT NULL,
    Country varchar(50) NOT NULL,
    Phone integer NOT NULL,
    Email varchar(50) NOT NULL,
    CONSTRAINT Plant_pk PRIMARY KEY (PltID)
);

```

Table Name: Plant Schema: matmgmt

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
PltID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Description	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
PManager	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Street1	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Street2	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
City	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
State	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Column Name: Data Type: Default: Storage:

Collation: Primary Key Not Null Unique
Comments: Binary Unsigned Zero Fill

Auto Increment Generated

11. PLANTMAT:

It contains the relationship between the Plants, Storage and the materials.

```
CREATE TABLE Plantmat (
    MatID integer NOT NULL,
    Pltid integer NOT NULL,
    Stid integer NOT NULL,
    Quantity numeric(10) NOT NULL,
    CONSTRAINT Plantmat_ID PRIMARY KEY (MatID,Pltid,Stid)
);
```

Table Name: Plantmat Schema: matmgmt

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MatID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Pltid	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Stid	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Quantity	DECIMAL(10,0)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Column Name: Data Type: Default: Storage:

Collation: Primary Key Not Null Unique
Comments: Binary Unsigned Zero Fill

Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

12. PURCHASE:

This table captures the details of the Purchase orders created by the Employee's, created date and the approval status (Approved, Rejected, Pending Approval).

```
CREATE TABLE Purchase (
    POID integer NOT NULL,
    Createdate date NOT NULL,
    Createdby varchar(50) NOT NULL,
    ApprovalStat char(1) NULL,
    CONSTRAINT Purchase_pk PRIMARY KEY (POID)
);
```

The screenshot shows a table creation dialog with the following details:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
POID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Createdate	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Createdby	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
ApprovalStat	CHAR(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
VID	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Below the table structure, there are input fields for Column Name, Data Type, Default, Storage, and various checkboxes for Primary Key, Not Null, Unique, Binary, Unsigned, Zero Fill, Auto Increment, and Generated.

13. PURCHASEMAT:

The Purchasemat table contains details related to each of the purchase orders like the MatID , Ordered quantity.

```
CREATE TABLE Purchasemat (
    POID integer NOT NULL,
    MatID integer NOT NULL,
    OrderQuantity numeric(20) NOT NULL,
    CONSTRAINT Purchasemat_pk PRIMARY KEY (POID,MatID)
);
```

The screenshot shows the 'Create Table' dialog in MySQL Workbench. The table name is 'Purchasemal', schema is 'matmgmt', engine is 'InnoDB', and collation is 'latin1 - default collation'. The table has one column 'POID' of type INT(11), which is defined as a primary key (PK) and not null (NN). Other options like UQ, B, UN, ZF, AI, G, and Default/Expression are not selected. The storage section includes options for Virtual, Stored, Primary Key, Not Null, Unique, Binary, Unsigned, Zero Fill, Auto Increment, and Generated.

14. VENDOR:

The vendor table contains the details of the vendor like Vendor ID, name, Address of the vendor.

```
CREATE TABLE Vendor (
    VID integer NOT NULL,
    Vendorname varchar(50) NOT NULL,
    Street1 varchar(50) NOT NULL,
    Street2 varchar(50) NOT NULL,
    City varchar(50) NOT NULL,
    State varchar(50) NOT NULL,
    Country varchar(50) NOT NULL,
    ZIP integer NOT NULL,
    Phone numeric(20) NOT NULL,
    Email varchar(50) NOT NULL,
    CONSTRAINT Vendor_pk PRIMARY KEY (VID)
);
```

Table Name: **Vendor** Schema: **matmgmt**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
VID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Vendorname	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Street1	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Street2	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
City	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
State	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Country	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Column Name: Data Type:
 Collation: Default:
 Comments:
 Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

15. VENDORMAT:

The vendomat table contains the materials offered by each of the vendors.

```
CREATE TABLE Vendmat (
  MatID integer NOT NULL,
  VID integer NOT NULL,
  VenPrice float(50) NOT NULL,
  CONSTRAINT Vendmat_pk PRIMARY KEY (MatID,VID)
);
```

Table Name: **Vendmat** Schema: **matmgmt**

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
MatID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
VID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
VenPrice	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Column Name: Data Type:
 Collation: Default:
 Comments:
 Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

16. TAX:

The tax table contains the tax percent levied on the materials based on the State.

```
CREATE TABLE Tax (
  StateID integer NOT NULL,
  MatID integer NOT NULL,
  Taxpercent float(10) NOT NULL,
```

```

CONSTRAINT Tax_pk PRIMARY KEY (StateID)
);

```

The screenshot shows a table creation dialog for a table named 'Tax'. The schema is 'matmgmt'. The table has three columns: 'StateID' (INT(11), PK, NN, UQ), 'MatID' (INT(11), PK, NN, UQ), and 'Taxpercent' (FLOAT). Below the table definition, there are fields for adding a new column: 'Column Name', 'Data Type', 'Default', 'Collation', 'Comments', and storage options (Virtual, Stored, Primary Key, Not Null, Unique, Binary, Unsigned, Zero Fill, Auto Increment, Generated).

17. STATES:

The State table contains the State ID and Name of the State.

```

CREATE TABLE States (
    StateID integer NOT NULL,
    State varchar(50) NOT NULL
);

```

The screenshot shows a table creation dialog for a table named 'States'. The schema is 'matmgmt'. The table has two columns: 'StateID' (INT(11), NN, UQ) and 'State' (VARCHAR(50), NN, UQ). Below the table definition, there are fields for adding a new column: 'Column Name', 'Data Type', 'Default', 'Collation', 'Comments', and storage options (Virtual, Stored, Primary Key, Not Null, Unique, Binary, Unsigned, Zero Fill, Auto Increment, Generated).

18. SALES:

This table contains the details of the sales orders created by customers.

```

CREATE TABLE Sales (
    SOID integer NOT NULL,
    Createddate date NOT NULL,
    createdtime timestamp NOT NULL,
    CustID varchar(50) NOT NULL,
    CONSTRAINT Sales_pk PRIMARY KEY (SOID,CustID)
)

```

);

The screenshot shows a table creation dialog for a table named "Sales". The schema is "matmgmt". The table has three columns: SOID (INT(11)), Createddate (DATE), and CustID (VARCHAR(50)). The SOID column is defined as the primary key (PK) and NOT NULL (NN). The Createddate column is NOT NULL (NN). The CustID column is NOT NULL (NN). The dialog also includes fields for Column Name, Data Type, Default, Storage, and various check boxes for constraints like Primary Key, Not Null, Unique, Binary, Unsigned, Zero Fill, Auto Increment, and Generated.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
SOID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Createddate	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
CustID	VARCHAR(50)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Column Name:
Collation:
Comments:
Data Type:
Default:
Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

19. SALESITEM:

The Sales Item table contains the

```
CREATE TABLE Salesitem (
    SOID integer NOT NULL,
    Pid integer NOT NULL,
    orquantity number(50) NOT NULL,
    CONSTRAINT Salesitem_pk PRIMARY KEY (SOID,Pid)
);
```

The screenshot shows a table creation dialog for a table named "Salesitem". The schema is "matmgmt". The table has three columns: SOID (INT(11)), MatID (INT(11)), and orquantity (DECIMAL(50,0)). The SOID column is defined as the primary key (PK) and NOT NULL (NN). The MatID column is NOT NULL (NN). The orquantity column is NOT NULL (NN). The dialog also includes fields for Column Name, Data Type, Default, Storage, and various check boxes for constraints like Primary Key, Not Null, Unique, Binary, Unsigned, Zero Fill, Auto Increment, and Generated.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
SOID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
MatID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
orquantity	DECIMAL(50,0)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Column Name:
Collation:
Comments:
Data Type:
Default:
Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

20. SHIPPING:

This table captures the shipping details of each of the Sales orders.

```
CREATE TABLE Shipping (
    Street1 varchar(50) NOT NULL,
    Street2 varchar(50) NOT NULL,
```

```

City varchar(50) NOT NULL,
State varchar(50) NOT NULL,
Country varchar(50) NOT NULL,
ZIP integer NOT NULL,
SOID integer NOT NULL,
CONSTRAINT Shipping_pk PRIMARY KEY (SOID)
);

```

The screenshot shows the MySQL Workbench interface for creating a table named 'Shipping'. The table is defined with the following schema:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
SOID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Street1	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Street2	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
City	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
State	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
Country	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
ZIP	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Below the table definition, there is a detailed configuration panel for the 'SOID' column:

- Column Name: SOID
- Data Type: INT(11)
- Collation: Table Default
- Comments: (empty)
- Storage: Primary Key (checked), Not Null (checked), Unsigned (unchecked), Auto Increment (unchecked)
- Default: NULL
- Unique (unchecked), Zero Fill (unchecked), Generated (unchecked)

21. INSURANCE:

This contains the details of the insurance plan selected for a Sales Order.

```

CREATE TABLE Insurance (
insID integer NOT NULL,
Prid integer NOT NULL,
Description varchar(50) NOT NULL,
insprice float(30) NOT NULL,
Sales_SOID integer NOT NULL,
CONSTRAINT Insurance_pk PRIMARY KEY (insID,Prid)
);

```

The screenshot shows a table creation interface with the following details:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
insID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
PrId	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Description	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
insprice	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Sales_SOID	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Below the table definition, there are fields for adding new columns:

- Column Name:
- Data Type:
- Collation: Table Default
- Comments:
- Default:
- Storage:
 - Virtual
 - Stored
- Primary Key:
- Not Null:
- Unique:
- Binary:
- Unsigned:
- Zero Fill:
- Auto Increment:
- Generated:

22. INSPROVIDER:

This contains the details of the insurance provider with details like Provider ID, provider Name, Address.

```
CREATE TABLE InsProvider (
    PrID integer NOT NULL,
    Prname varchar(50) NOT NULL,
    Description varchar(50) NOT NULL,
    Street1 varchar(50) NOT NULL,
    Street2 varchar(50) NOT NULL,
    City varchar(50) NOT NULL,
    State varchar(50) NOT NULL,
    ZIP integer NOT NULL,
    Country varchar(50) NOT NULL,
    Phone integer NOT NULL,
    Fax integer NOT NULL,
    Email varchar(50) NOT NULL,
    CONSTRAINT InsProvider_pk PRIMARY KEY (PrID)
);
```

Table Name: InsProvider Schema: matmgmt

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
PRID	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Pname	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Description	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Street1	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Street2	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
City	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
State	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL					

Column Name:	Data Type:
Collation:	Default:
Comments:	Storage: <input type="radio"/> Virtual <input checked="" type="radio"/> Stored <input type="checkbox"/> Primary Key <input type="checkbox"/> Not Null <input type="checkbox"/> Unique <input type="checkbox"/> Binary <input type="checkbox"/> Unsigned <input type="checkbox"/> Zero Fill <input type="checkbox"/> Auto Increment <input type="checkbox"/> Generated

List of tables:

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	TABLE_NAME	TABLE_ROWS		
	Auth	1428		
	Customer	1987		
	Discount	0		
	Dummv_Sales	9603		
	Employee	459		
	InsProvider	701		
	Insurance	460		
	Material	11		
	Membership	4		
	Plant	2		
	Plantmat	0		
	Product	10		
	Productmat	46		
	Purchase	10157		
	Purchasemat	14934		
	Review	199		
	Sales	10000		
	Salesitem	10000		

Database Provisioning:

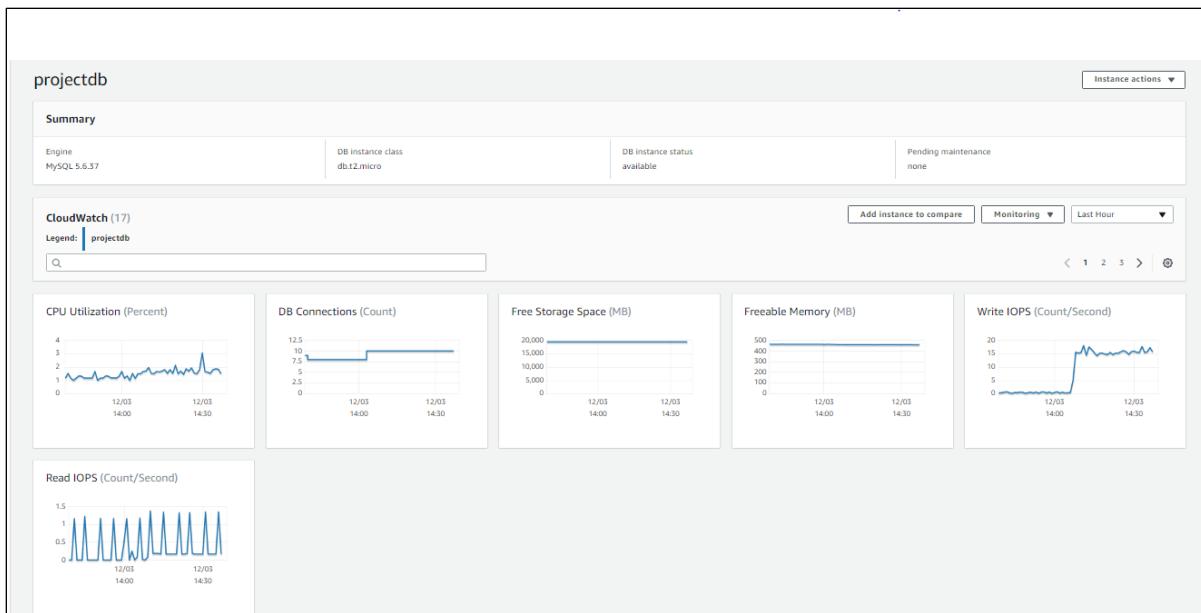
For this project we have created an MySQL database on cloud using Amazon's RDS management console.

Below are the snapshot details of cloud database in AWS

The screenshot shows the Amazon RDS Home page. On the left, a sidebar lists various RDS management options like Dashboard, Instances, Clusters, and Performance Insights. The main content area displays resource usage statistics: DB instances (1/40), Allocated storage (20.00 GB/100.00 TB), Reserved instances (0/40), Snapshots (10/40), and Subnet groups (2/50). It also lists Parameter groups (2), Option groups (1), and External licenses (0). Below this is a 'Create instance' section with 'Launch a DB Instance' and 'Restore from S3' buttons. To the right, there's an 'Additional information' panel with links to RDS documentation and forums, and a 'Related services' panel featuring Amazon ElastiCache.

This screenshot shows the 'Instances' page under the RDS service. It displays a single MySQL database instance named 'projectdb'. The instance is currently available with 1.50% CPU usage and 10 connections. The page includes standard navigation controls like 'Filter instances', 'Instance actions', 'Launch DB instance', and 'Restore from S3'.

This screenshot is identical to the one above, but the 'Instance actions' dropdown menu for the 'projectdb' instance is open, revealing options such as 'See details', 'Create read replica', 'Create Aurora read replica', 'Promote read replica', 'Take snapshot', 'Restore to point in time', 'Migrate latest snapshot', 'Modify', 'Stop', 'Reboot', and 'Delete'.



Connect

Endpoint projectdb.cq5wbnrwdbg1.us-east-1.rds.amazonaws.com	Port 3306	Publicly accessible Yes
--------------------------------------------------------------------------------	--------------	----------------------------

Security group rules (4)

Filter security group rules

Security group	Type	Rule
rds-launch-wizard-2	CIDR/IP - Inbound	0.0.0.0/0
rds-launch-wizard-2	CIDR/IP - Inbound	97.96.15.190/32
rds-launch-wizard-2	CIDR/IP - Inbound	47.198.31.183/32
rds-launch-wizard-2	CIDR/IP - Inbound	68.207.232.223/32

Details			
Configurations	Security and network	Instance and IOPS	Maintenance details
ARN arn:aws:rds:us-east-1:020983758190:db:projectdb	Availability zone us-east-1e VPC vpc-09a83771	Instance Class db.t2.micro Storage Type General Purpose (SSD) Storage 20 GB	Auto minor version upgrade Yes Maintenance window sun:09:29-sun:09:59 Backup window 03:07-03:37 Pending Modifications None Pending maintenance none Encryption details Encryption enabled No
Engine MySQL 5.6.37	Subnet group default	Availability and durability	
License Model General Public License	Subnets subnet-4756ed78 subnet-c4af328f subnet-cf665392 subnet-0870106c subnet-e28151cd subnet-d4f2fc08	DB instance status available Multi AZ No Automated backups Enabled (15 Days)	
Created Time Mon Nov 27 12:03:58 GMT-500 2017	Security groups rds-launch-wizard-2 (sg-33f97846) (active)	Latest restore time December 3, 2017 at 2:40:00 UTC-5	
DB Name matmgmt	Publicly accessible Yes		
Username group6users	Endpoint projectdb.cq5wbnnwdbg1.us-east-1.rds.amazonaws.com		
Option Group defaultmysql-5-6	Certificate authority rds-ca-2015 (Mar 5, 2020)		
Parameter group group6 (in-sync)			
Copy tags to snapshots false			
Resource ID db-SI2NHYLWQTXBDDR2FGYYSZKQPY			
IAM DB Authentication Enabled No			

Data Generation & Loading:

Data Generation and Loading is an important step in building any database. To maintain the relationship between entities the data population needs to be a carefully planned process. For instance in our database the Sales orders are created by the customers and Purchase orders are created by the employees. While populating the data in each of the tables the base table of Employees and Customers was populated first then we moved on to populating the data in the created by column of Purchase and Sales tables.

For generating the data in tables, the use of MS Excel and online tools has been made. We have specifically used the ‘Mockaroo’ to populate data in most of the tables, alongside excel to create necessary functions which weren’t supported by the online tools.

Website: <https://www.mockaroo.com>

Example of one of the tables generated using the online tool

Shipping table:

The columns in the shipping table are the following

- SOID -Sales Order ID
- Fname- First Name
- Lname-Last Name
- Street1
- Street2
- City
- State
- Country
- ZIP

The schema of the table is created in the online tool using the graphical user interface and the data for each of the columns is generated. The data generated can be captured in Excel, CSV or JSON. The CSV format was used to export data from Mockaroo as it is easy to use the same option of file import in MySQL.

Step 1:

Field Name	Type	Options
SOID	Row Number	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
first_name	First Name	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
last_name	Last Name	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
Street1	Street Number	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
Street2	Street Name	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
City	City	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
State	State	<input checked="" type="checkbox"/> generate only US locations <input type="text" value="restrict states..."/> blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
Country	Country	<input type="text" value="restrict countries..."/> blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>
ZIP	Postal Code	blank: 0 % <input type="button" value="fx"/> <input type="button" value="x"/>

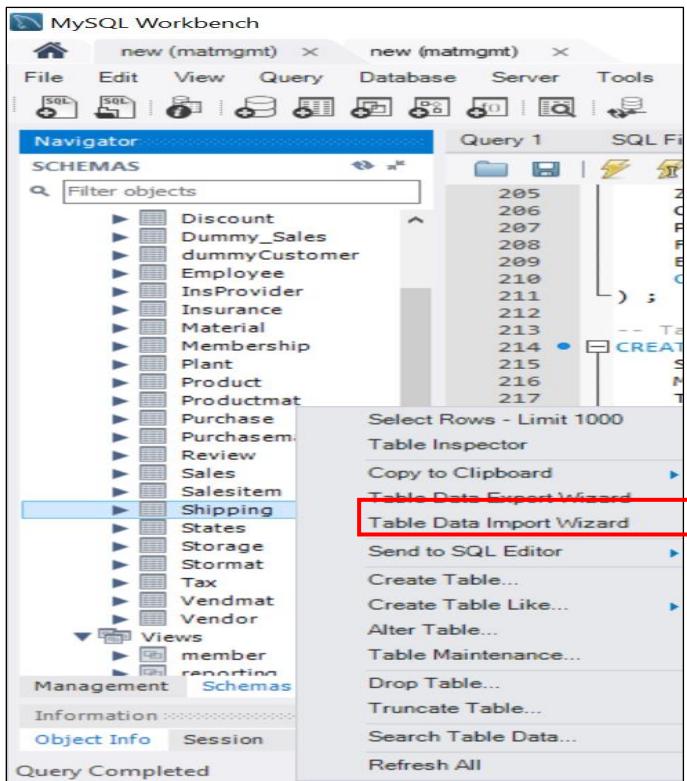
[Add another field](#)

The data generated each time has a limitation of 1000 rows. Below is the screenshot of the generated data.

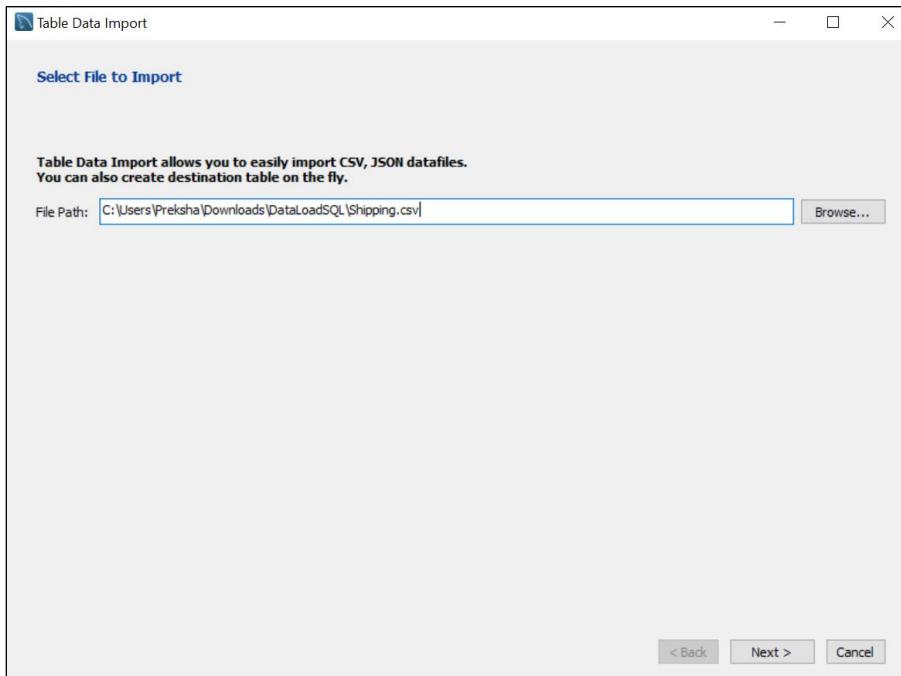
	A	B	C	D	E	F	G	H	I	J
1	SOID	first_name	last_name	Street1	Street2	City	State	Country	ZIP	
2	1 Amelia	Czaja		219 Homewo	Phoenix	Arizona	United States		85062	
3	2 Rafaelle	Yoakley		5 Havey	Austin	Texas	United States		78732	
4	3 Liz	Champe		87 Talmadge	Annapolis	Maryland	United States		21405	
5	4 Nicholas	Menego		8 Maple Wc	Milwaukee	Wisconsin	United States		53263	
6	5 Clem	Areles		6 Jana	New York	New York	United States		10165	
7	6 Aime	Albrooke		3 Pankratz	New Hyde	New York	United States		11044	
8	7 Hillary	Thumann		7 Lerdahl	Boise	Idaho	United States		83727	
9	8 Alexine	Squires		6382 Carpenter	Davenport	Iowa	United States		52804	
10	9 Cristionna	Graver		6968 Bonner	Apache Ju	Arizona	United States		85219	
11	10 Aimee	Keates		80 Anthes	San Diego	California	United States		92176	
12	11 Dulcia	Itzig		5378 Randy	Vero Beach	Florida	United States		32969	
13	12 Fallon	Dibble		109 Dayton	Fort Smith	Arkansas	United States		72916	
14	13 Rozalie	Purkis		58345 John Wall	Oklahoma	Oklahoma	United States		73135	
15	14 Lenci	Dace		99 Golf View	Milwaukee	Wisconsin	United States		53205	
16	15 Maegan	Adshad		5058 Ohio	West Palm	Florida	United States		33405	
17	16 Kale	Dearsley		7 Merry	New Orleans	Louisiana	United States		70187	
18	17 Lauritz	Lortz		950 Westend	Charlotte	North Carolina	United States		28242	
19	18 Brandice	Withers		2 Eastwood	Kansas City	Missouri	United States		64109	
20	19 Darbee	Bellerby		64448 Swallow	Dayton	Ohio	United States		45419	
21	20 Stanton	Ren		892 East	Atlanta	Georgia	United States		30323	
22	21 Marie-jeanne	Ivey		337 Colorado	Akron	Ohio	United States		44305	

Step 2:

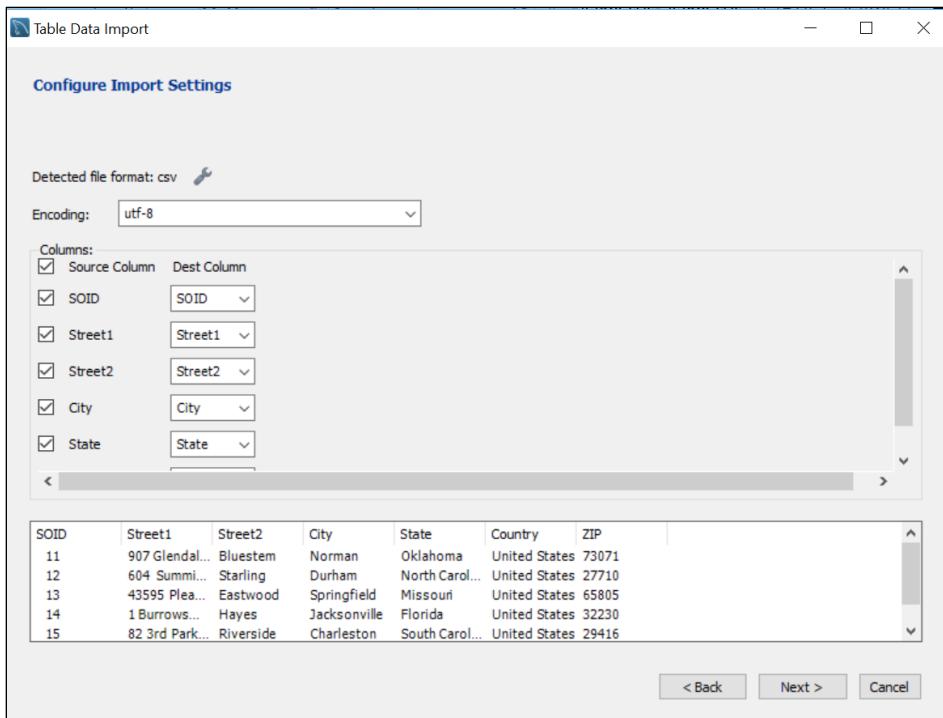
The data is now imported into the MySQL database using the “Table data Import Wizard”.



The next screen selection of the file that needs to be imported.

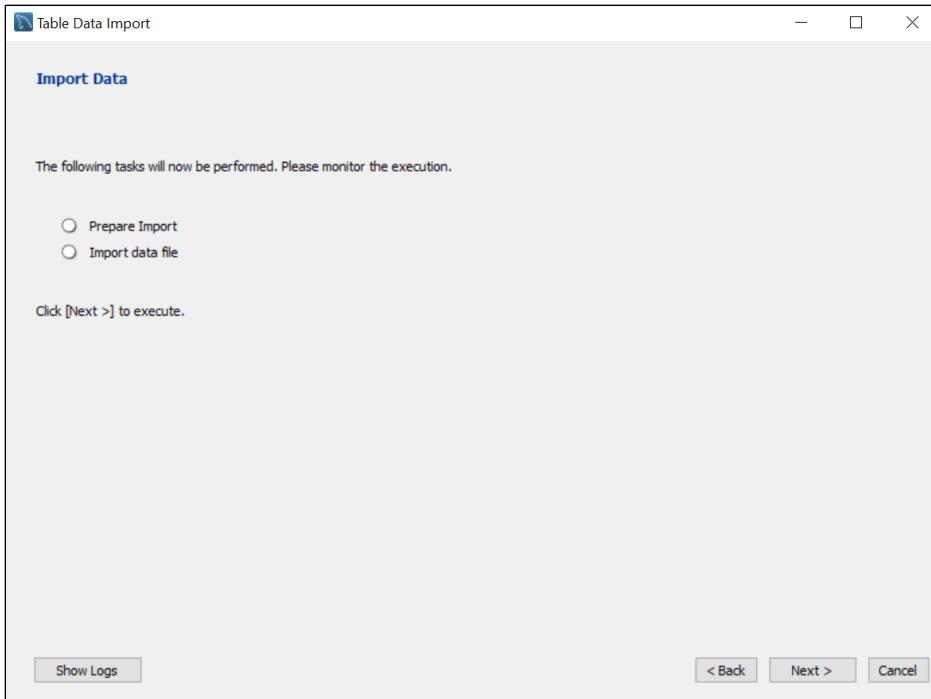


Step 3: Verify the columns created in the database match the file header section.



Step 4:

Import the data and verify the presence of the data in required format after it is imported.



Similarly, all other tables were populated either using online or excel.

Performance Tuning:

As relational database systems evolved in terms of volume and complexity, the focus has moved towards writing queries that get executed efficiently and other techniques. To get a result set that a user needs, different queries to be written which obtain the same results. But the one that is most efficient is the one that runs faster, consumes minimal resources and executes with lowest cost. There are various mechanisms in which performance of queries can be improved. Some of them are as discussed below.

- **Indexing:**

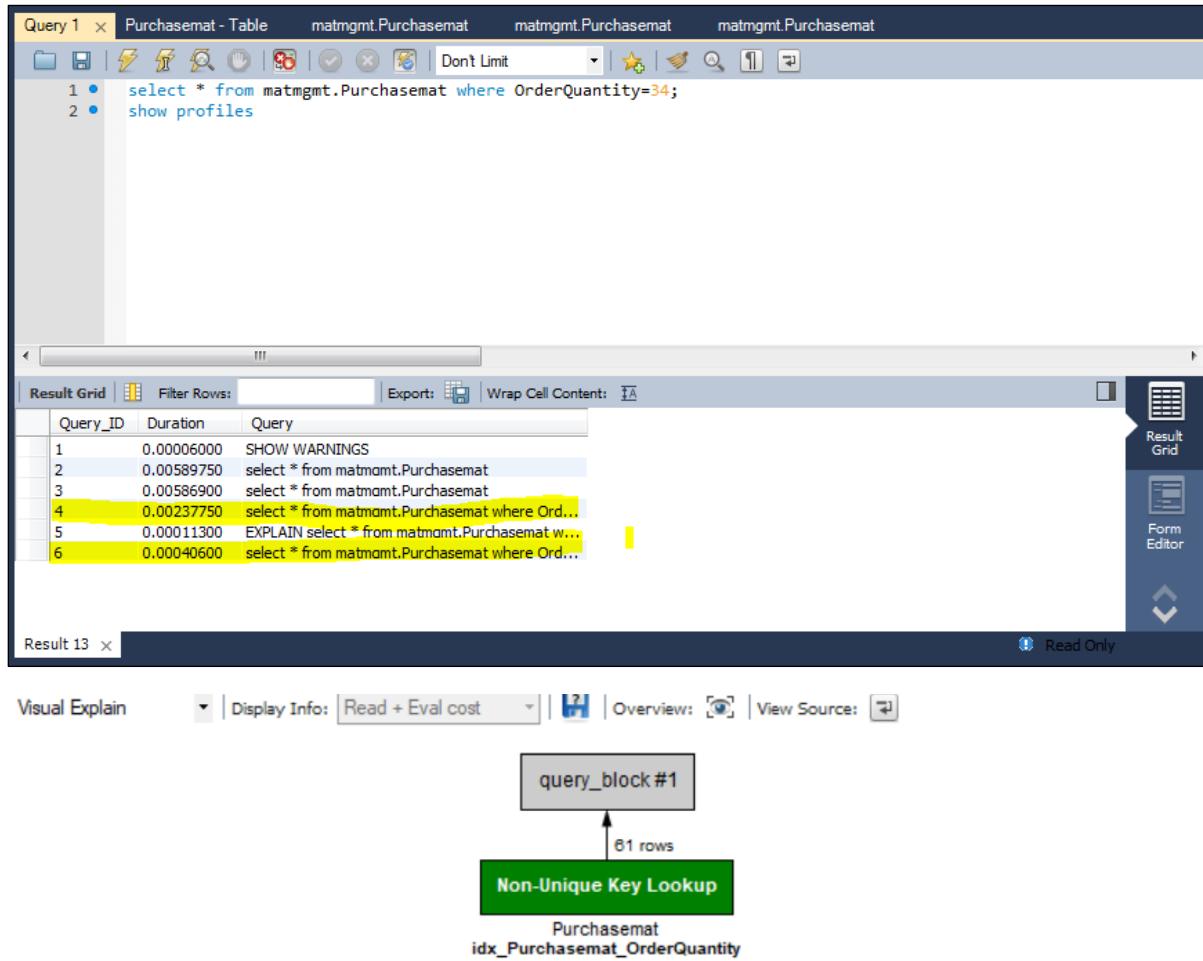
When there are no indexes, the database will scan the table and then sort the rows to process the query. However, the index will provide the database with already sorted list of table's columns. The database can simply scan the index from the first record to the last record and retrieve the rows in sorted order. Index is the best tool for the database to use to enforce uniqueness in data of a column. Each time an application adds or modifies a row in the table, the database needs to search all existing records to ensure none of values in the new data duplicate existing values. Indexes improves the search time.

Query 1: To view the purchase orders where ordered quantity is 34 with and without indexing on Purchasemat table.

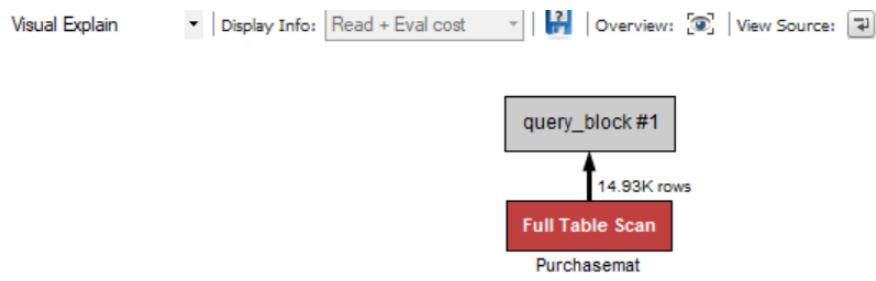
```

set profiling=1;
select * from matmgmt.Purchasemat where OrderQuantity=34;
show profiles

```



Without Index:

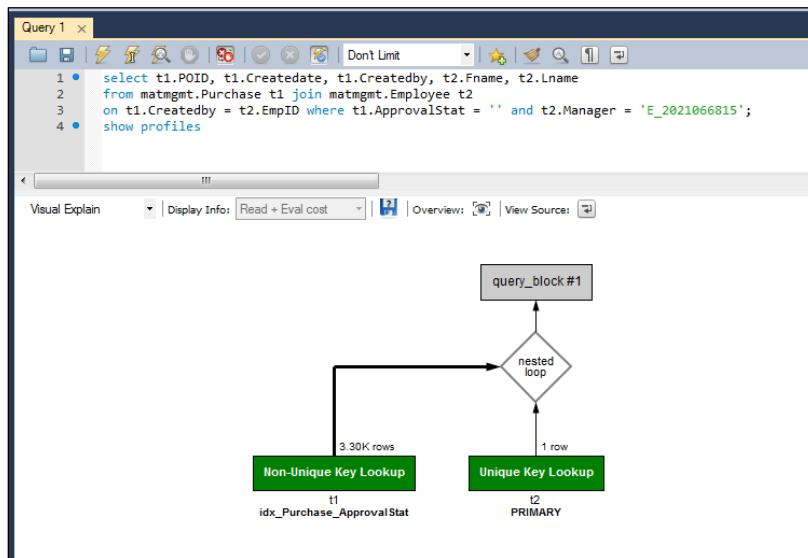
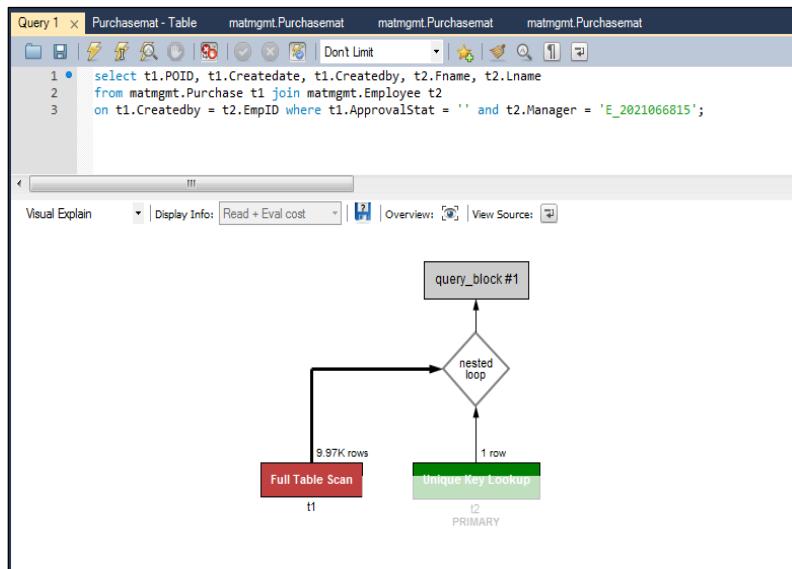


Query 2: To view the purchase orders of which are pending approval of the manager.

```

select t1.POIID, t1.Createdate, t1.Createdby, t2.Fname, t2.Lname
from matmgmt.Purchase t1 join matmgmt.Employee t2
on t1.Createdby = t2.EmpID where t1.ApprovalStat = '' and t2.Manager = 'E_2021066815';

```



Query 1

```

1 • select t1.POID, t1.Createdate, t1.Createdby, t2.Fname, t2.Lname
2   from matmgmt.Purchase t1 join matmgmt.Employee t2
3   on t1.Createdby = t2.EmpID where t1.ApprovalStat = '' and t2.Manager = 'E_2021066815';
4 • show profiles

```

Result Grid | Filter Rows: Export: Wrap Cell Content:

Query_ID	Duration	Query
8	0.00029450	EXPLAIN select t1.POID, t1.Createdate, t1.Cre...
9	0.00033300	EXPLAIN FORMAT=JSON select t1.POID, t1.Cr...
10	0.000806900	select t1.POID, t1.Createdate, t1.Createdby, t...
11	0.00033675	EXPLAIN select t1.POID, t1.Createdate, t1.Cre...
12	0.00035175	EXPLAIN FORMAT=JSON select t1.POID, t1.Cr...
13	0.20769925	select * from matmgmt.Customer
14	0.00377700	select * from matmgmt.Customer group by State
15	0.00697800	select t1.POID, t1.Createdate, t1.Createdby, t...
16	0.00722650	select t1.POID, t1.Createdate, t1.Createdby, t...
17	0.00705275	select t1.POID, t1.Createdate, t1.Createdby, t...
18	0.00707875	select t1.POID, t1.Createdate, t1.Createdby, t...
19	0.00719175	select t1.POID, t1.Createdate, t1.Createdby, t...

Query 3: To view the material review of customer with platinum membership.

```

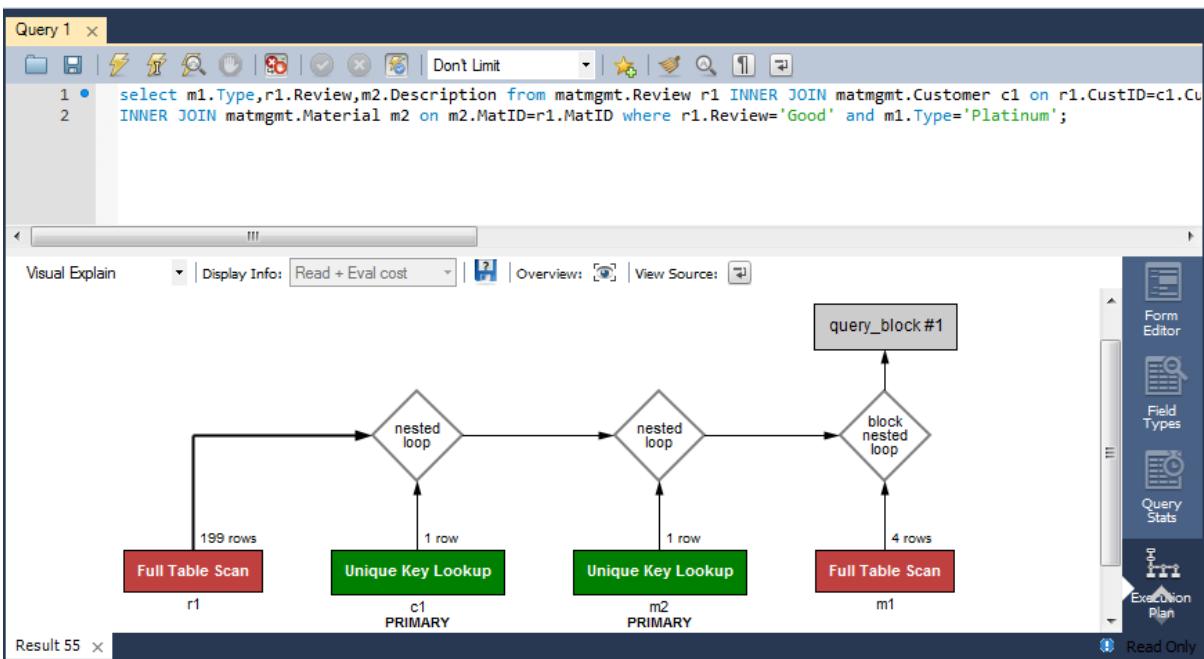
select m1.Type,r1.Review,m2.Description from matmgmt.Review r1 INNER JOIN
matmgmt.Customer c1 on r1.CustID=c1.CustID INNER JOIN matmgmt.Membership m1 on
m1.MemID=c1.Membership_MemID

```

```

INNER JOIN matmgmt.Material m2 on m2.MatID=r1.MatID where r1.Review='Good' and
m1.Type='Platinum';

```



Query 1

```

1 • select m1.Type,r1.Review,m2.Description from matmgmt.Review r1 INNER JOIN matmgmt.Customer c1 on r1.CustID=c1.CustID
2 • INNER JOIN matmgmt.Material m2 on m2.MatID=r1.MatID where r1.Review='Good' and m1.Type='Platinum';
3 • show profiles
4

```

Result Grid | Filter Rows: Export: Wrap Cell Content: ⌂

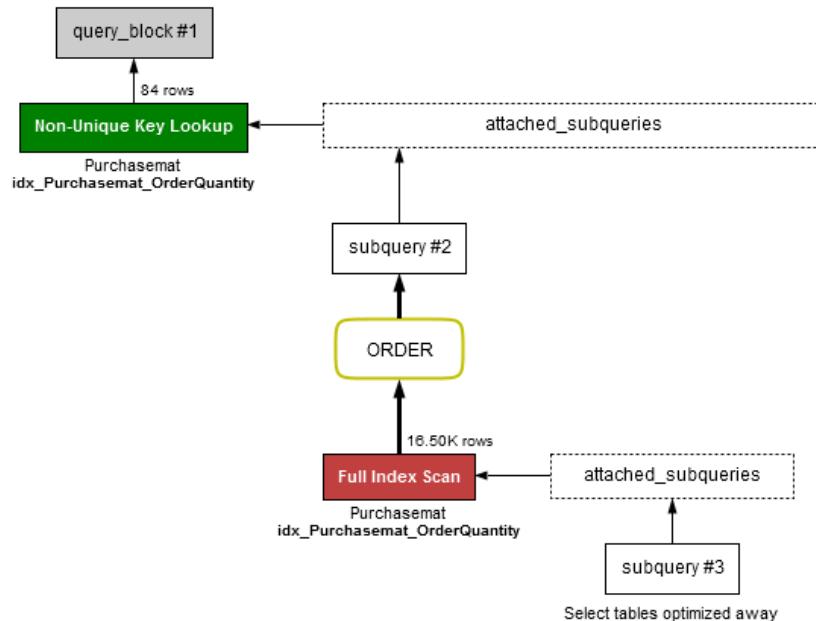
Query_ID	Duration	Query
64	0.00090800	select m1.MemID,r1.Review,m2.Description fro...
65	0.00032825	EXPLAIN select m1.MemID,r1.Review,m2.Descri...
66	0.00039425	EXPLAIN FORMAT=JSON select m1.MemID,r1.R...
67	0.00081225	select m1.MemID,r1.Review,m2.Description fro...
68	0.00091800	select m1.Tvoe,r1.Review,m2.Description from ...
69	0.00067600	select m1.Tvoe,r1.Review,m2.Description from ...
70	0.00064025	select m1.Tvoe,r1.Review,m2.Description from ...
71	0.00035425	EXPLAIN select m1.Tvoe,r1.Review,m2.Descri...
72	0.00039400	EXPLAIN FORMAT=JSON select m1.Tvoe,r1.Re...
73	0.01678625	ALTER TABLE `matmgmt`.`Review` ADD INDEX...
74	0.00061425	select m1.Tvoe,r1.Review,m2.Description from ...
75	0.00037200	EXPLAIN select m1.Tvoe,r1.Review,m2.Descri...
76	0.00042525	EXPLAIN FORMAT=JSON select m1.Tvoe,r1.Re...
77	0.00058325	select m1.Tvoe,r1.Review,m2.Description from ...
78	0.00063925	select m1.Tvoe,r1.Review,m2.Description from ...

Query 4:

```

select MatID,POID,OrderQuantity from matmgmt.Purchasemat where OrderQuantity=(select
max(OrderQuantity) from matmgmt.Purchasemat where OrderQuantity not in (select
max(OrderQuantity) from matmgmt.Purchasemat) order by MatID limit 1)

```



{

"query_block": {

```

"select_id": 1,
"table": {
    "table_name": "Purchasemat",
    "access_type": "ref",
    "possible_keys": [
        "idx_Purchasemat_OrderQuantity"
    ],
    "key": "idx_Purchasemat_OrderQuantity",
    "used_key_parts": [
        "OrderQuantity"
    ],
    "key_length": "9",
    "ref": [
        "const"
    ],
    "rows": 84,
    "filtered": 100,
    "using_index": true,
    "attached_condition": "(`matmgmt`.`Purchasemat`.`OrderQuantity` = /* select#2 */ select max(`matmgmt`.`Purchasemat`.`OrderQuantity`) from `matmgmt`.`Purchasemat` where (not(<in_optimizer>(`matmgmt`.`Purchasemat`.`OrderQuantity`),<exists> /* select#3 */ select max(`matmgmt`.`Purchasemat`.`OrderQuantity`) from `matmgmt`.`Purchasemat` having (<cache>(`matmgmt`.`Purchasemat`.`OrderQuantity`)) = <ref_null_helper>(max(`matmgmt`.`Purchasemat`.`OrderQuantity`)))))) order by `matmgmt`.`Purchasemat`.`MatID` limit 1))",
    "attached_subqueries": [
        {
            "dependent": false,
            "cacheable": true,
            "query_block": {

```

```

"select_id": 2,
"ordering_operation": {
    "using_filesort": false,
    "table": {
        "table_name": "Purchasemat",
        "access_type": "index",
        "key": "idx_Purchasemat_OrderQuantity",
        "used_key_parts": [
            "OrderQuantity"
        ],
        "key_length": "9",
        "rows": 16504,
        "filtered": 100,
        "using_index": true,
        "attached_condition":
"(not(<in_optimizer>(`matmgmt`.`Purchasemat`.`OrderQuantity`,<exists>(* select#3 */ select
max(`matmgmt`.`Purchasemat`.`OrderQuantity`) from `matmgmt`.`Purchasemat` having
(<cache>(`matmgmt`.`Purchasemat`.`OrderQuantity`) =
<ref_null_helper>(max(`matmgmt`.`Purchasemat`.`OrderQuantity`))))))",
        "attached_subqueries": [
            {
                "dependent": true,
                "cacheable": false,
                "query_block": {
                    "select_id": 3,
                    "table": {
                        "message": "Select tables optimized away"
                    }
                }
            }
        ]
    }
}

```

```

        ]
    }
}
}
]
}
}
}
}

```

- **Using Join instead of subquery:**

Query using subquery:

```
select POID, Createdate, Createdby from matmgmt.Purchase where ApprovalStat = " and
Createdby in (select EmpID from matmgmt.Employee where Manager = 'E_2021066815');
```

Using a join instead of subquery:

```
select t1.POID, t1.Createdate, t1.Createdby, t2.Fname, t2.Lname
from matmgmt.Purchase t1 join matmgmt.Employee t2
on t1.Createdby = t2.EmpID where t1.ApprovalStat = " and t2.Manager = 'E_2021066815';
```

- **Stored Procedure:**

1. **To identify Matid and POID for which OrderQuantity is maximum**

```
CREATE DEFINER=`group6users`@`%` PROCEDURE `displaymax`(OUT param1 int,OUT
param2 int)
```

```
BEGIN
```

```
select MatID,POID INTO param1,param2 from matmgmt.Purchasemat where
OrderQuantity=(select max(OrderQuantity) from matmgmt.Purchasemat) order by MatID limit
1;
```

```
END
```

```
call displaymax (@a,@b);
```

```
select @a as 'Material',@b as 'Purchase Order';
```

```

Query 1 SQL File 3* x displaymax - Routine
1 • call displaymax (@a,@b);
2
3 • select @a as 'Material',@b as 'Purchase Order';

Result Grid Filter Rows: Export: Wrap Cell Content: □
Material Purchase Order
1 730

```

2. To find count for second most ordered quantity in Purchasemat table

```

CREATE DEFINER=`group6users`@`%` PROCEDURE `display2ndmax`(OUT param1
int,OUT param2 int)
BEGIN
    select OrderQuantity,count(*) As 'Frequency' INTO param1,param2 from
matmgmt.Purchasemat where OrderQuantity=(select max(OrderQuantity) from
matmgmt.Purchasemat where OrderQuantity not in (select max(OrderQuantity) from
matmgmt.Purchasemat) order by MatID limit 1);
END

call display2ndmax (@a,@b);

select @a as 'QTY',@b as 'Frequency';

```

3.To review counts

```

USE `matmgmt`;

DROP procedure IF EXISTS `reviewcounts`;

```

DELIMITER \$\$

USE `matmgmt`\$\$

```

CREATE DEFINER=`group6users`@`%` PROCEDURE `reviewcounts`(OUT param1 int,OUT
param2 varchar(10))

```

BEGIN

```

    select count(*),Review from matmgmt.Review group by Review;
END$$

```

DELIMITER ;

```
call reviewcounts(@a,@b);
```

The screenshot shows the MySQL Workbench interface. In the top tab bar, 'Query 1' is active, followed by 'SQL File 3*', 'SQL File 4*', 'Review - Table', and 'reviewcounts - Routine'. Below the tabs, there's a toolbar with various icons. The main area contains three lines of code: '1 call reviewcounts(@a,@b)', '2', and '3'. Underneath the code, the 'Result Grid' tab is selected, showing a table with two columns: 'count(*)' and 'Review'. The data is as follows:

count(*)	Review
28	Bad
100	Good
28	Moderate
43	Verv Good

4. Procedure to extract details of latest review from customer

```
USE `matmgmt`;
```

```
DROP procedure IF EXISTS `latestreview`;
```

```
DELIMITER $$
```

```
USE `matmgmt`$$
```

```
CREATE
```

```
DEFINER=`group6users`@`%`
```

```
PROCEDURE `latestreview`(OUT param1 varchar(20),OUT param2 varchar(20),OUT param3  
varchar(20),OUT param4 varchar(20),OUT param5 varchar(10),OUT param6 int)
```

```
BEGIN
```

```
select r1.CustID,c1.Fname,c1.Lname,m1.Type,r1.Review,r1.Rating from matmgmt.Review r1  
INNER JOIN matmgmt.Customer c1 on r1.CustID=c1.CustID  
INNER JOIN matmgmt.Membership m1 on m1.MemID=c1.Membership_MemID order by  
r1.reviewdate limit 1;
```

```
END$$
```

```

DELIMITER ;

call latestreview(@a,@b,@c,@d,@e,@f)

```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Query 1' is selected. Below the bar, there are tabs for 'SQL File 3*', 'SQL File 4*', 'Review - Table', and 'latestreview - Routine'. The main area contains two lines of code:

```

1  call latestreview(@a,@b,@c,@d,@e,@f)
2

```

Below the code, a 'Result Grid' is displayed with the following data:

CustID	Fname	Lname	Type	Review	Rating
C 2039411820	Ryan	Scottini	Gold	Good	9

- **TRIGGERS:**

1.Trigger calling a stored procedure to display number of reviews in each category after each insert in reviews table

```

DROP TRIGGER IF EXISTS `matmgmt`.`Review_AFTER_INSERT`;

DELIMITER $$

USE `matmgmt`$$

CREATE DEFINER=`group6users`@`%` TRIGGER `matmgmt`.`Review_AFTER_INSERT`
AFTER INSERT ON `Review`
FOR EACH ROW
BEGIN
call reviewcounts(@a,@b);
END$$

DELIMITER ;

```

2.Trigger calling a stored procedure to display the latest review details before insertion on reviews table:

```
DROP TRIGGER IF EXISTS `matmgmt`.`Review_BEFORE_INSERT`;
```

```

DELIMITER $$

USE `matmgmt`$$

CREATE

DEFINER = CURRENT_USER

TRIGGER `matmgmt`.`Review_BEFORE_INSERT`

BEFORE INSERT ON `Review`

FOR EACH ROW

BEGIN

call latestreview(@a,@b,@c,@d,@e,@f);

END$$

DELIMITER ;

```

- **Views:**

A database view is a virtual table or logical table. Views can be used in query optimization as the time and required to execute the query is reduced when we can create views for the queries run frequently.

View 1:

```

CREATE

ALGORITHM = UNDEFINED

DEFINER = `group6users`@``

SQL SECURITY DEFINER

VIEW `member` AS

SELECT

`C`.`CustID` AS `CustID`,

`C`.`Lname` AS `Lname`,

`M`.`Discount` AS `Discount`,

`M`.`Type` AS `Type`


FROM

`Customer` `C`

```

```
JOIN `Membership` `M` ON ((`C`.`Membership_MemID` = `M`.`MemID`)))  
WHERE
```

```
(`M`.`Type` IN ('Platinum', 'Gold', 'Silver'))
```

View2 : To view the manager name for each employee

```
CREATE
```

```
ALGORITHM = UNDEFINED
```

```
DEFINER = `group6users`@`%`
```

```
SQL SECURITY DEFINER
```

```
VIEW `reporting` AS
```

```
SELECT
```

```
`E`.`Fname` AS `Fname`,
```

```
`E`.`Lname` AS `Lname`,
```

```
`E`.`Designation` AS `Designation`,
```

```
CONCAT(`M`.`Fname`, ", ", `M`.`Lname`) AS `Manager`
```

```
FROM
```

```
(`Employee` `E`
```

```
JOIN `Employee` `M` ON ((`E`.`Manager` = `M`.`EmpID`)))
```

View 3: Fetch all customers with membership

```
CREATE
```

```
ALGORITHM = UNDEFINED
```

```
DEFINER = `group6users`@`%`
```

```
SQL SECURITY DEFINER
```

```
VIEW `salesordcust` AS
```

```
SELECT
```

```
`C`.`CustID` AS `CustID`,
```

```
`C`.`Fname` AS `Fname`,
```

```
`C`.`Lname` AS `Lname`,
```

```
`S`.`SOID` AS `SOID`,  
 `S`.`Createddate` AS `Createddate`  
FROM  
(`Customer` `C`  
JOIN `Membership` `M`  
JOIN `Sales` `S` ON ((`C`.`Membership_MemID` = `M`.`MemID`)  
AND (`C`.`CustID` = `S`.`CustID`)))  
WHERE  
(`M`.`Type` IN ('Platinum', 'Gold', 'Silver'))  
GROUP BY `S`.`Createddate`
```

View 4: Count of customers by state

CREATE

```
ALGORITHM = UNDEFINED  
DEFINER = `group6users`@`%`  
SQL SECURITY DEFINER  
VIEW `Statewise` AS  
SELECT  
 `Customer`.`State` AS `State`, COUNT(0) AS `count(*)`  
FROM  
 `Customer`  
GROUP BY `Customer`.`State`  
ORDER BY COUNT(0) DESC
```

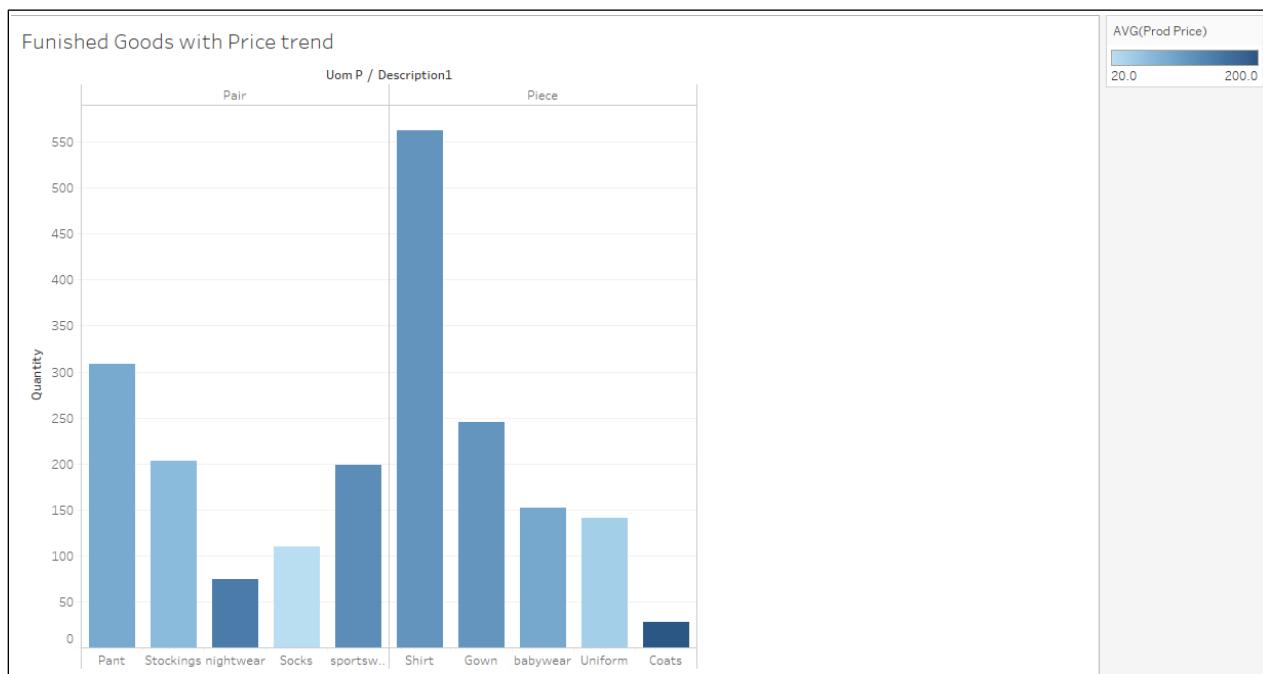
Visualization:

Visualization is helpful in analysing the information stored in the relational database. Specifically, in the e-commerce industry. It is extremely helpful in understanding the operating conditions, business performance, understanding customer trends and forecasting. It is helpful to find the correlation among the data available. It helps in understanding any emerging trends and provides the leadership with the opportunity to act faster by accommodating the emerging trends. Any company lagging behind in analytics has a risk of losing an edge in the market.

Hence, in today's highly competitive environment it is become necessary to keep a track of rapidly evolving business trends.

Below are few visualizations performed on the database:

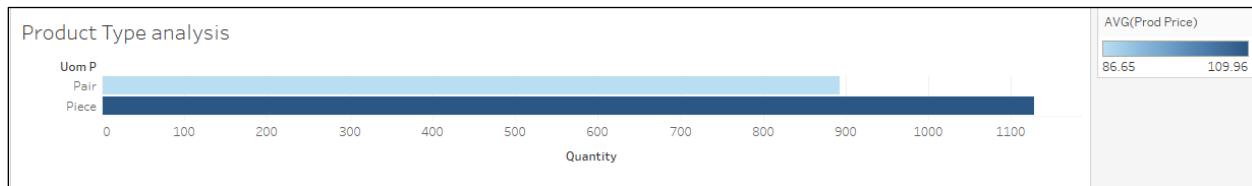
1. Trend in Finished goods sold in the market



Observation: Shirt and Pant in each category has maximum number of quantities produced and shipped despite more than average price.

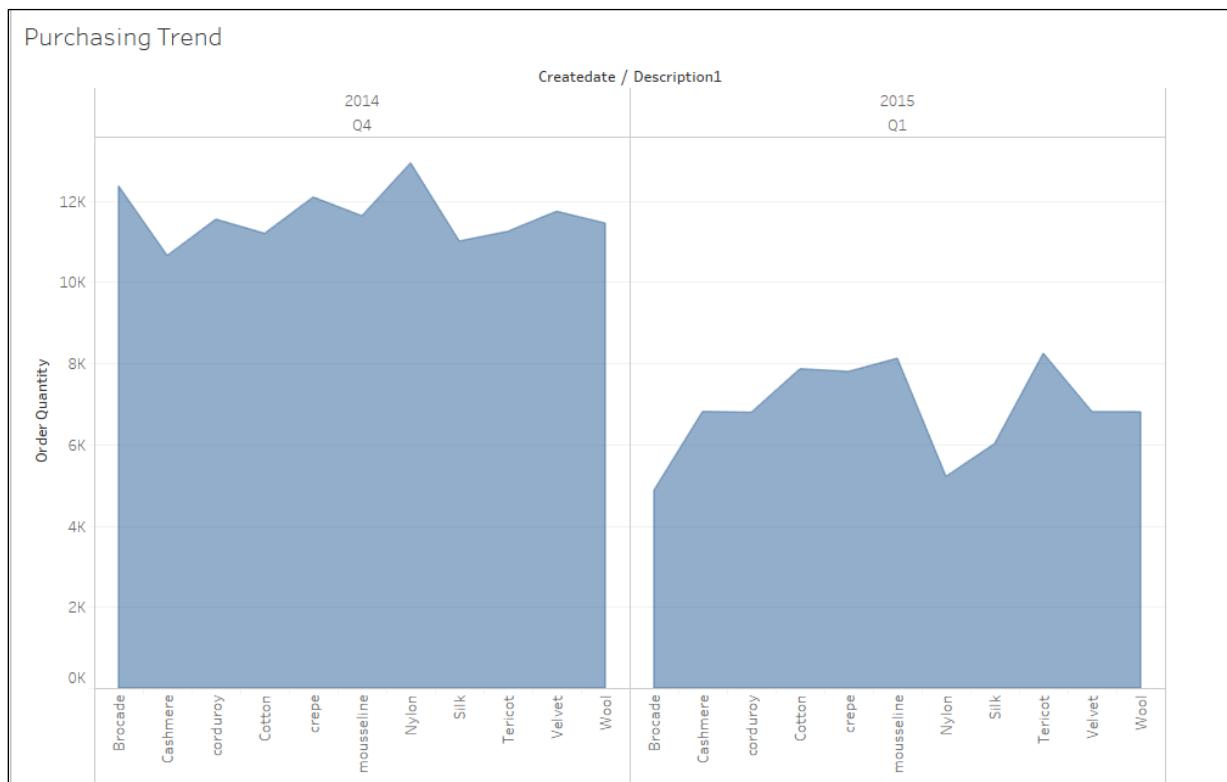
Since Coat being highly priced hence the demand of it is low followed by nightwear in Pair category.

2. Trend in sales of product type



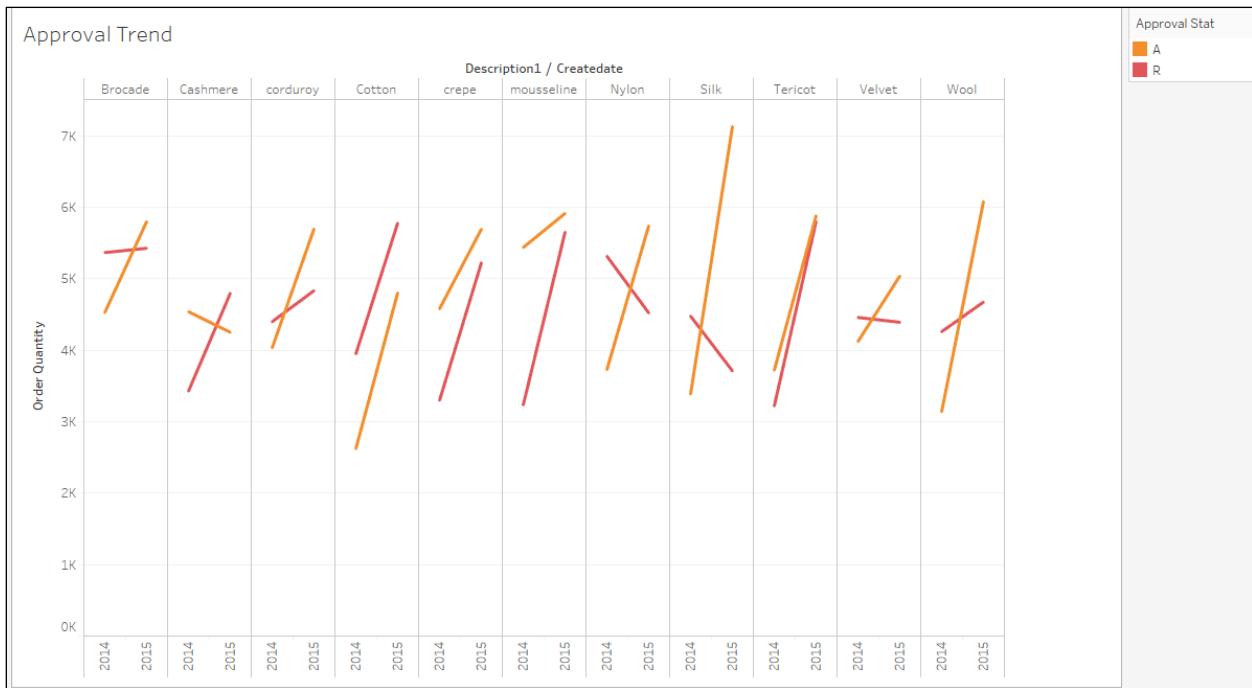
Observation: It clearly indicate the Apparel sold as a single Piece has a better price and also an increase demand

3. Trend in volume of products being sold over the year.



Observation: Q4 shows high quantities being sold because of the increase sales during the holiday season.

4. Trend in Purchase Orders created for materials

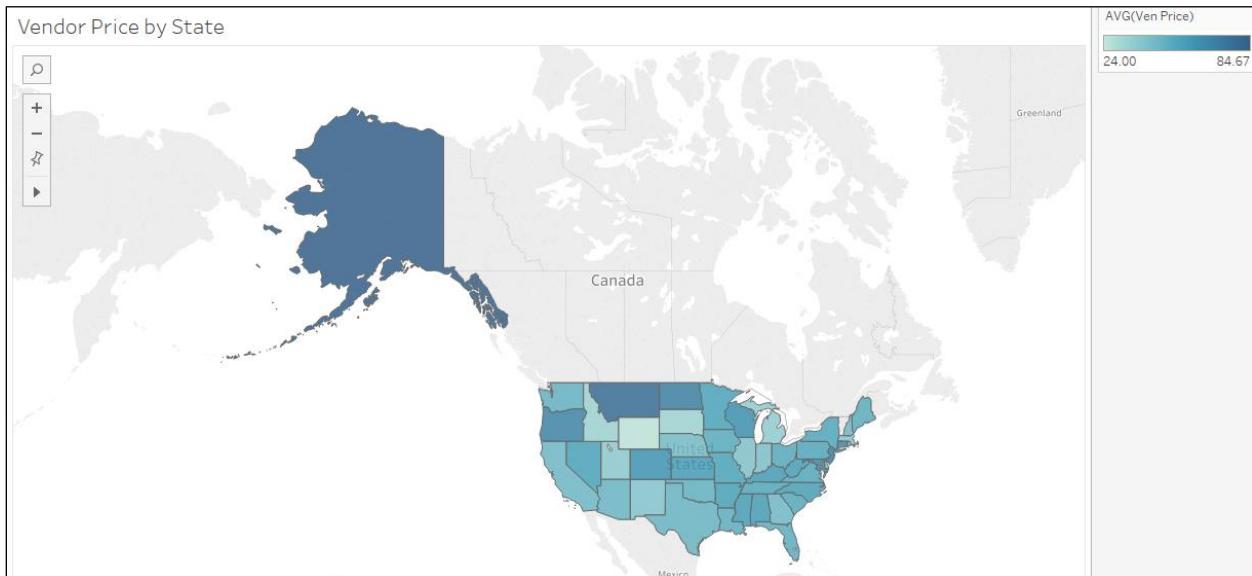


Observation:

Approval for PO trend has increased trend (due to high popularity) except item like Cashmere

Whereas the Rejected trend is observed with clothing material like Silk, Nylon , Velvet etc.

5. Trend in price by geographical region

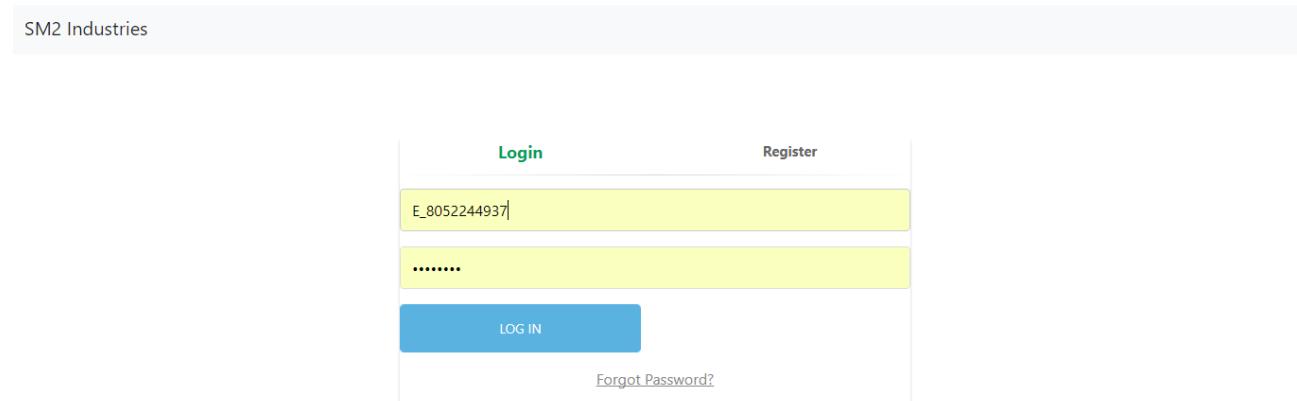


*Observation :*We observed a surprising trend for sparsely populated states like Alaska , Montana , Oregon etc high vendor price was recorded . Notifying the higher trend.

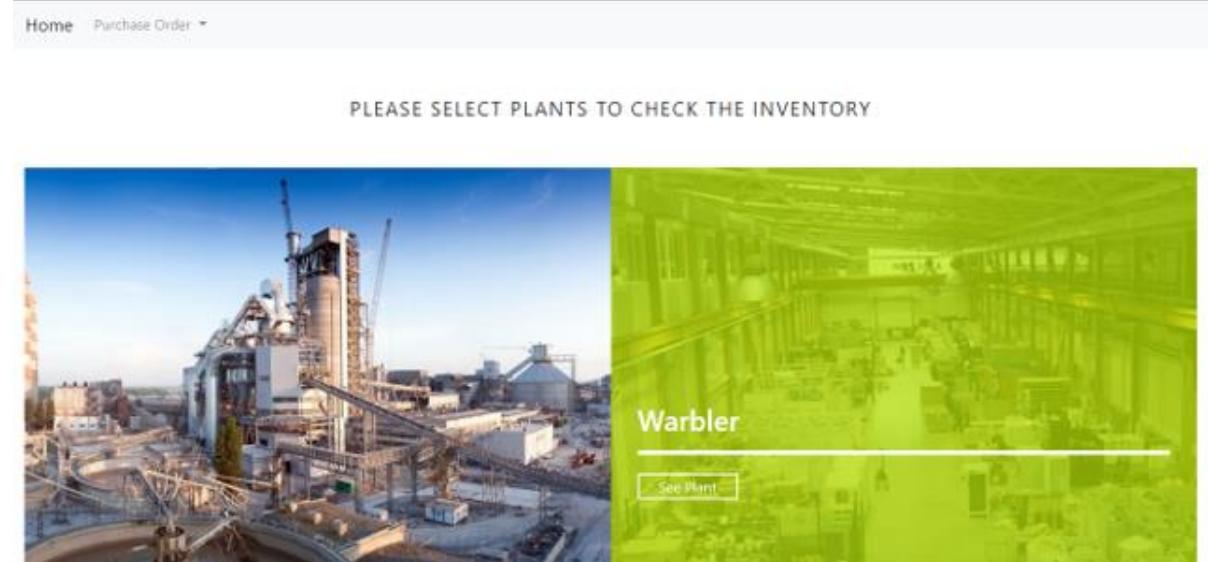
Application Design:

To complete the IT solution for the material management system, after the design of the database. The front-end of the application design was completed using PHP, HTML,CSS, Javascript and jquery.Below are few screenshots of pages in the application.

Login screen:



The screenshot shows the login interface for 'SM2 Industries'. At the top left is the company logo. On the right, there are 'Login' and 'Register' buttons. Below these are two input fields: the first contains the text 'E_8052244937', and the second contains several dots representing a password. A blue 'LOG IN' button is positioned below the inputs. At the bottom right of the form is a link 'Forgot Password?'. The background of the form is white, while the input fields and button are highlighted in light yellow.



The screenshot displays a user interface for managing inventory across multiple plants. At the top, a navigation bar includes links for 'Home' and 'Purchase Order'. Below this, a message reads 'PLEASE SELECT PLANTS TO CHECK THE INVENTORY'. Two large images are shown side-by-side: on the left, an aerial view of a modern industrial facility with various structures and pipes; on the right, a blurred image of an interior warehouse or factory floor. A green overlay on the right side features the word 'Warbler' and a 'See Plant' button.

Material details with respect to plant:

Home Purchase Order ▾

Material id	Material	Storage Location	Location id	Quantity
1	Cotton	Independence	1	211
5	Tericot	Independence	1	97
9	Cashmere	Independence	1	40
4	Nylon	Mallory	4	8
8	corduroy	Mallory	4	263

Create Purchase order:

Home Purchase Order ▾

<input type="text" value="Nylon"/> ▾	<input type="text" value="15"/>	<input type="button" value="Delete"/>
<input type="text" value="Tericot"/> ▾	<input type="text" value="45"/>	<input type="button" value="Delete"/>
<input type="text" value="Silk"/> ▾	<input type="text" value="26"/>	<input type="button" value="Delete"/>
<input type="button" value="Add"/>	<input type="button" value="Submit"/>	

View purchase orders Pending for approval:

PO Number	Created On	Created By	EmpID
13	2014-11-02	FrancescaAndrews	E_5206303796
19	2014-11-04	FrancescaAndrews	E_5206303796
29	2014-11-05	FrancescaAndrews	E_5206303796
39	2014-11-07	FrancescaAndrews	E_5206303796
64	2014-11-09	FrancescaAndrews	E_5206303796
70	2014-11-10	FrancescaAndrews	E_5206303796
78	2014-11-10	FrancescaAndrews	E_5206303796
81	2014-11-11	InezSears	E_2033700575
87	2014-11-11	InezSears	E_2033700575
90	2014-11-12	FrancescaAndrews	E_5206303796

Approve or reject purchase order (allowed by manager only):

Material	Quantity
Wool	104
Tericot	77

Approve **Reject**

Below is the link for the code :

<https://github.com/AakashGH123/ADBMSGroup6.git>

Challenges:

1. Data binding to front-end was a challenge.
2. Passing the parameters between the pages was difficult.
3. Identifying the relationship between the tables was complicated, scoping out and limiting the database design helped in resolution.
4. The setup of database on cloud was completely new and hence had challenges in connectivity
5. Access issues as not all team members were able to access the database while connecting with the client.
6. Privileges issues as default privileges is restricted for a user in cloud.

Privilege	Context	Comment
Alter	Tables	To alter the table
Alter routine	Functions,Procedures	To alter or drop stored functions/procedures
Create	Databases,Tables,Indexes	To create new databases and tables
Create routine	Databases	To use CREATE FUNCTION/PROCEDURE
Create temporary tables	Databases	To use CREATE TEMPORARY TABLE
Create view	Tables	To create new views
Create user	Server Admin	To create new users
Delete	Tables	To delete existing rows
Drop	Databases,Tables	To drop databases, tables, and views
Event	Server Admin	To create, alter, drop and execute events
Execute	Functions,Procedures	To execute stored routines
File	File access on server	To read and write files on the server
Grant option	Databases,Tables,Functions,Procedures	To give to other users those privileges you possess
Index	Tables	To create or drop indexes
Insert	Tables	To insert data into tables
Lock tables	Databases	To use LOCK TABLES (together with SELECT privilege)
Process	Server Admin	To view the plain text of currently executing queries
Proxy	Server Admin	To make proxy user possible
References	Databases,Tables	To have references on tables
Reload	Server Admin	To reload or refresh tables, logs and privileges
Replication client	Server Admin	To ask where the slave or master servers are
Replication slave	Server Admin	To read binary log events from the master
Select	Tables	To retrieve rows from table
Show databases	Server Admin	To see all databases with SHOW DATABASES
Show view	Tables	To see views with SHOW CREATE VIEW
Shutdown	Server Admin	To shut down the server
Super	Server Admin	To use KILL thread, SET GLOBAL, CHANGE MASTER, etc.
Trigger	Tables	To use triggers
Create tablespace	Server Admin	To create/alter/drop tablespaces
Update	Tables	To update existing rows
Usage	Server Admin	No privileges - allow connect only

Future Scope:

1. Quotations, Requisitions can be included in future.
2. Setting up public domain for the web application.
3. Expanding the web application towards retail.
4. Future scope to send notification to member for when the membership will expire.

References:

http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.html

<https://dev.mysql.com/doc/refman/5.5/en/optimization.html>