**Title Of Your Project – SQL commands**

**Name – Shiva Chembeti**

**Git Hub Link-https://github.com/Shiv-Lewis/Week3-Project4-a.git**

**Current Data- 11/15/2022**

**Project Description**

The retail company’s aims to produce toys by focusing on underwater-themed action figures portrayed in the popular cartoon and animated toys. Kids are the target customers of the products developed by the retail sales company. The raw materials used for producing the animated or cartoon are produced outside of the country and assembled and shipped to USA. The major stores in USA such as Walmart, Costco and Target are used for distribution of the products. The retail sales company deals with large amount of data necessary to successful perform its daily operations. Manual storing of data poses various challenges due to human error and time consumption. The project focus on building a database for retail sales company. The database developed for the retail company incorporates necessary entities and attributes to successfully store, retrieve and access the data.

**Network Architecture**

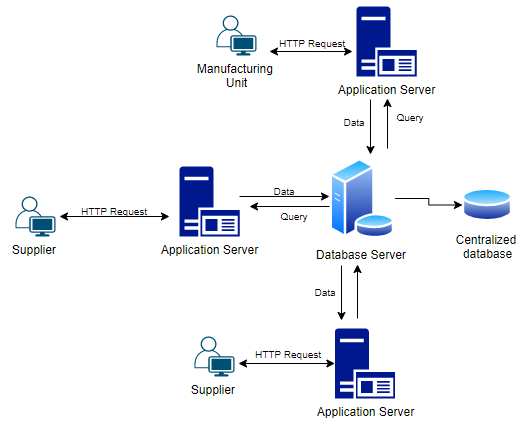


Figure 1: Network architecture of Retail Sales Company

The retail network architecture contains users such as client, supplier and manufacturing unit of retail sales company. To retrieve data stored in the database, the client / supplier initiated HTTP request for performing operation on the database server. The application server allows the users to connect to database server. Application server acts as an interface between the user and database server. The database server provides the data based on the request. For example if the manufacturing unit wants to access the raw material data, then it request a query through application server. The database server retrieves information and display it to a manufacturing unit.

**Entities and Attributes**

Entity is an object or event that stores data in the database. The database contains following entities.

**Entity 1: Raw\_Materials**

This entity helps to store the details of raw materials produced in different countries to produce the toy. Raw\_Materials enity contains attributes such as Raw-Material\_ID, Supplier\_ID, Material\_type and Unit\_cost. The data types of attrbutes are as follows

|  |  |
| --- | --- |
| **Attributes** | **Data Types** |
| Raw-Material\_ID | Integer |
| Supplier\_ID | Varchar |
| Material\_type | Integer |
| Unit\_cost | Varchar |

The Raw\_Material\_ID is the primary key which helps to uniquely identify other attributes in the Raw\_Materials entity. Supplier\_ID is the foreign key that helps to access supplier details.

**Entity 2: Supplier**

The suppliers from different countries supplies the raw materials to retail sales company. Supplier’s information is usually stored in supplier entity. The entity incorporates attributes such as Supplier\_ID, Supplier\_Name, Contact\_Number and Address. In this entity,

|  |  |
| --- | --- |
| **Attributes** | **Data Types** |
| Supplier\_ID | Integer |
| Supplier\_Name | Varchar |
| Contact\_Number | Integer |
| Address | Varchar |

Supplier\_ID is unique for all the suppliers which helps to uniquely identify each supplier. Supplier\_ID is also used to retrieve other attributes of the entity.

**Entity 3: Toys\_Production**

Toys\_production entity is the primary entity that stores the information of toys including name, description, weight and quality of the toys. The quality attribute helps to identify the quality level of each toy. The attributes included in the Toys\_production entity are Toy\_production\_ID, Lot\_ID, Toy\_Name, Toy\_Description, Toy\_weight and IsQualityTested. The data types of these attributes are given below

|  |  |
| --- | --- |
| **Attributes** | **Data Types** |
| Toy\_Production\_ID | Integer |
| Lot\_ID | Integer |
| Toy\_Name | Varchar |
| Toy\_Description | Varchar |
| Toy\_Weight | Integer |
| IsQualityTested | Varchar |

In this entity, Toy\_Production\_ID is the unique value that helps to retrieve other attributes stored under the Toy\_Production\_ID entity in the database.

**Entity 4: Lot**

Lot entity helps to store the cost related information of material. The entity incorporates attributes such as Lot\_ID, cost and created\_Date. The data types for these entities are

|  |  |
| --- | --- |
| **Attributes** | **Data type** |
| Lot\_ID | Integer |
| Cost | Decimal (5,2) |
| Created\_Date | Date |

Lot\_ID is the primary key which helps to uniquely identify other attributes in the Lot entity. 4

**Entity 5: Raw\_Material\_Lot**

The entity links raw material information and the cost of materials. It contains attributes such as Raw-material\_ID, Lot\_ID and Totla\_Unit. The data types of these three attributes are given below

|  |  |
| --- | --- |
| **Attributes** | **Data type** |
| Raw\_Material\_ID | Integer |
| Lot\_ID | Integer |
| Total\_Unit | Integer |

To retrieve the information from Raw\_material Entity and Lot entity, the Raw-Material\_Lot entity sets Raw\_material\_ID and Lot\_ID as a foreign key.

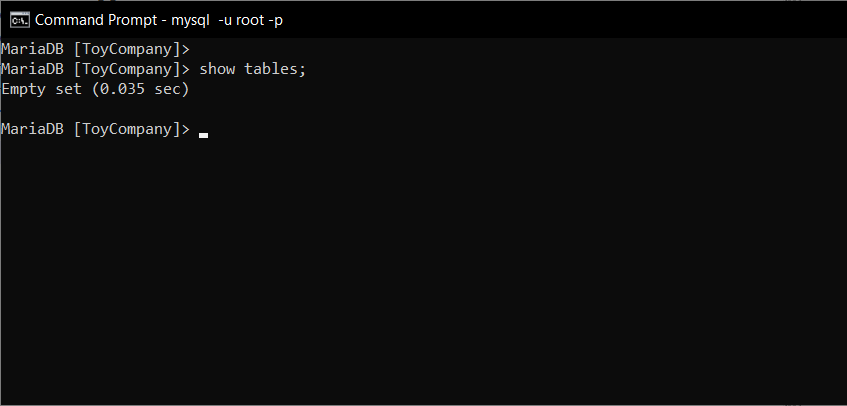
Each entity in the building database stores various attributes and each attributes stores n number of values.

The below sections of the project will demonstrate use of SQL commands for creating a database then loading the files into database for performing various operations.

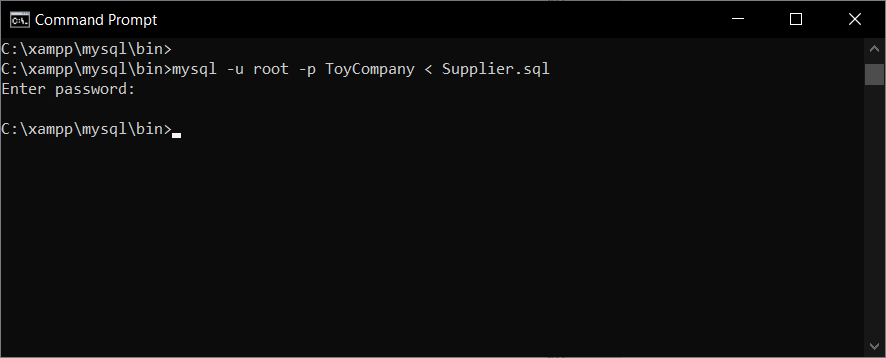
**Table creation**

1. **Supplier**

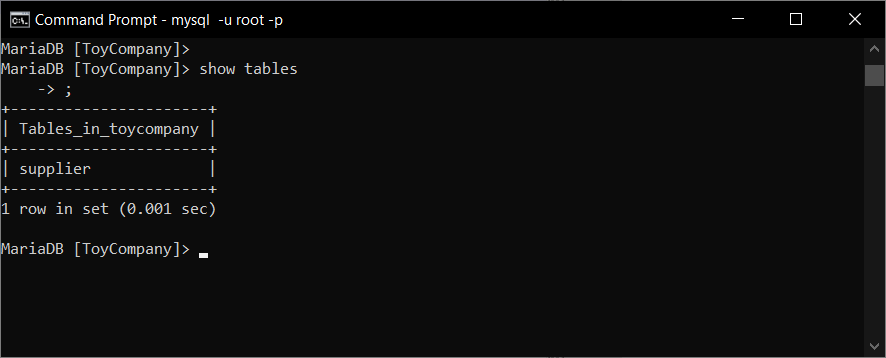
**Before create**



mysql -u root -p ToyCompany < Supplier.sql

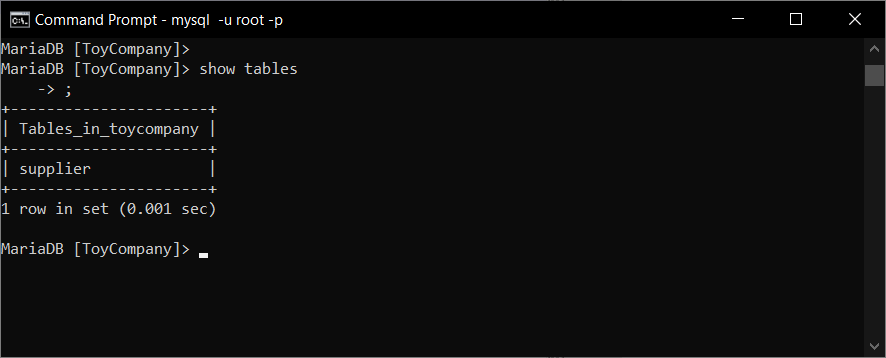


**After create**

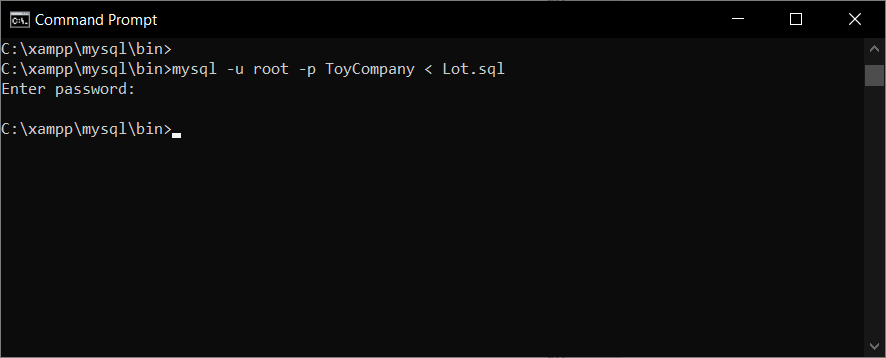


1. **Lot**

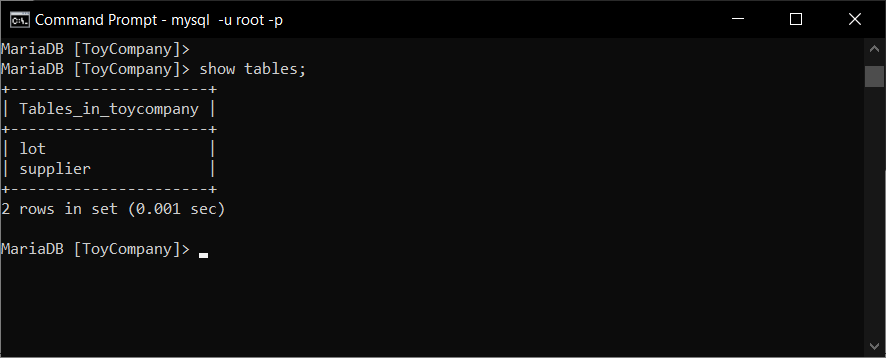
**Before create**



mysql -u root -p ToyCompany < Lot.sql

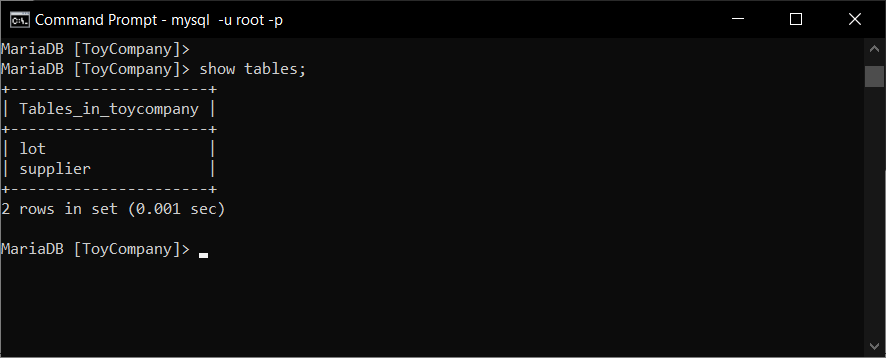


**After create**

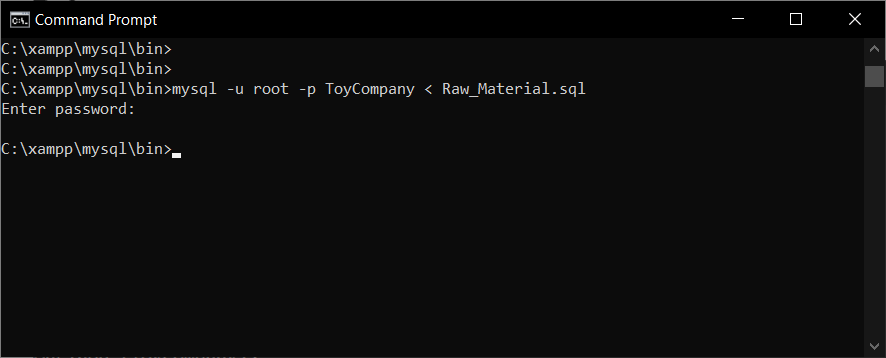


1. **Raw\_Material**

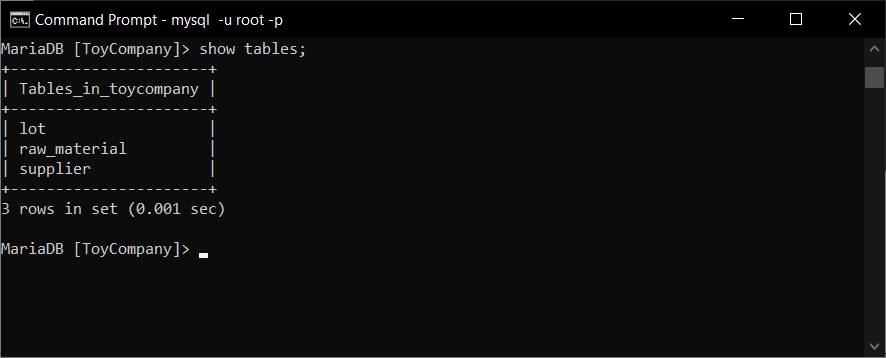
**Before create**



mysql -u root -p ToyCompany < Raw\_Material.sql

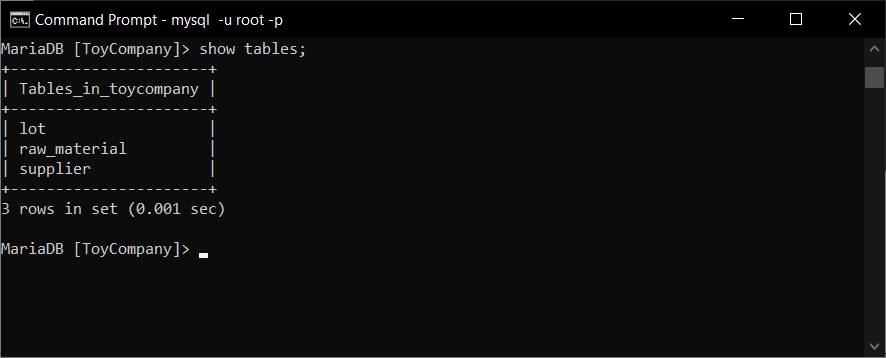


**After create**

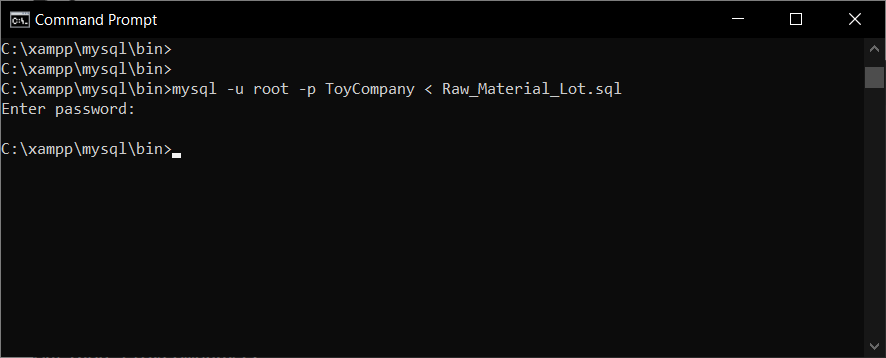


1. **Raw\_Material\_Lot**

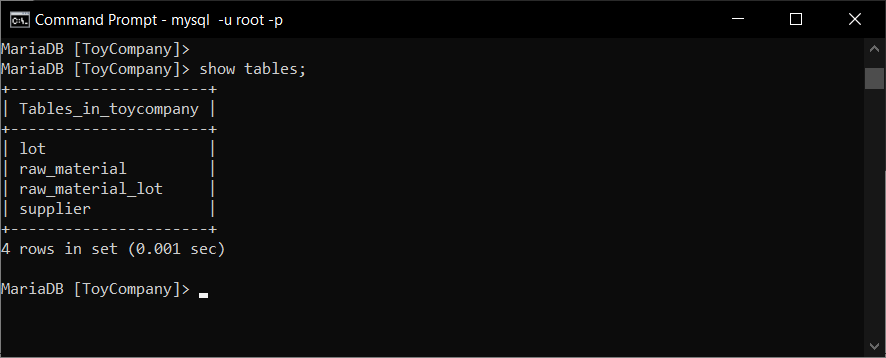
**Before create**



mysql -u root -p ToyCompany < Raw\_Material\_Lot.sql

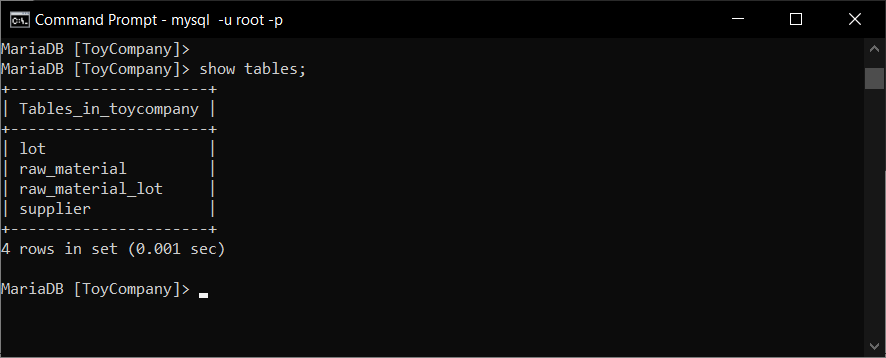


**After create**

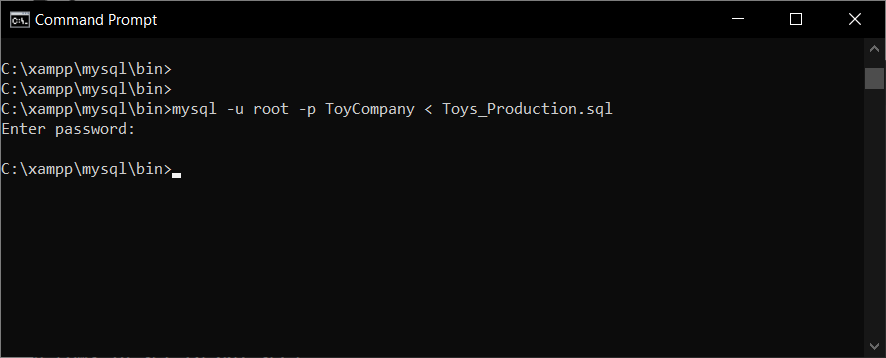


1. **Toys\_Production**

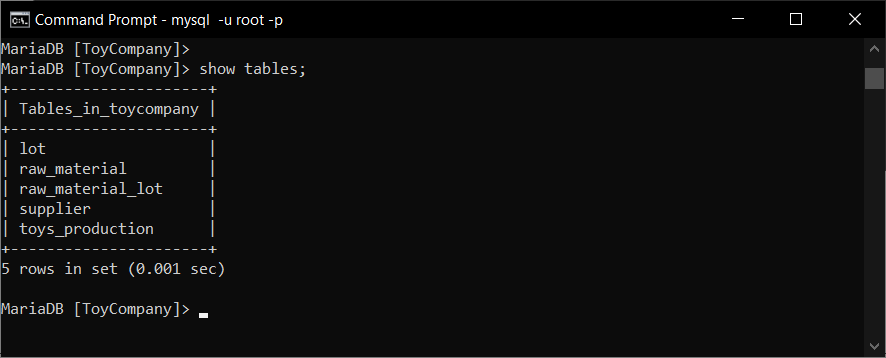
**Before create**



mysql -u root -p ToyCompany < Toys\_Production.sql



**After create**

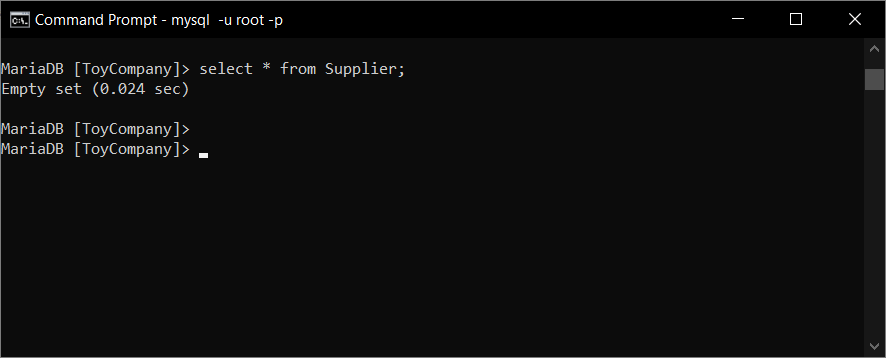


**Data insertion**

1. **Supplier**

**Before insert**

Select \* from Supplier;



LOAD DATA INFILE 'F:/CSV files/Suppliers.csv'

INTO TABLE Supplier

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

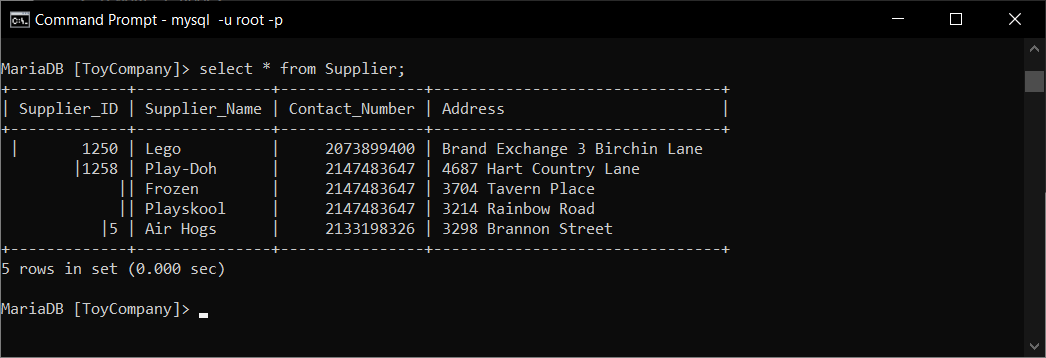
LINES TERMINATED BY '\n'

IGNORE 1 ROWS;



**After insert**

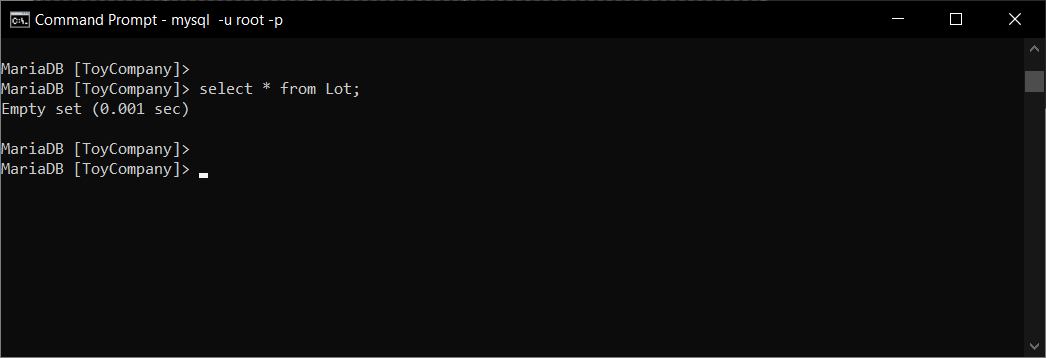
Select \* from Supplier;



1. **Lot**

**Before insert**

select \* from Lot;



LOAD DATA INFILE 'F:/CSV files/Lot.csv'

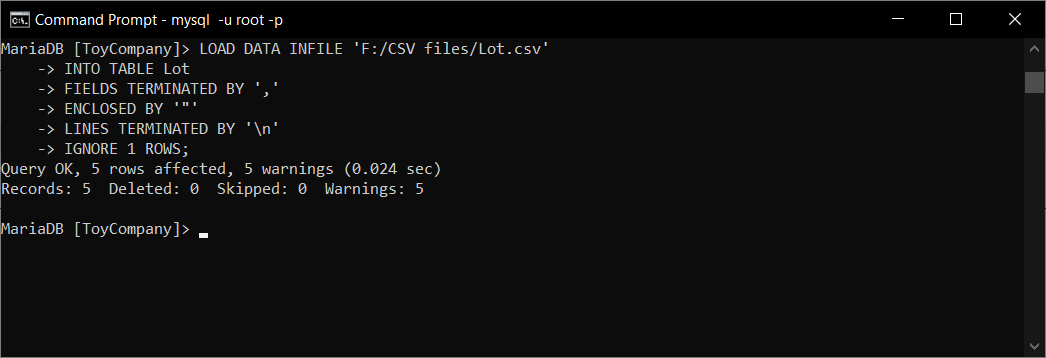
INTO TABLE Lot

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

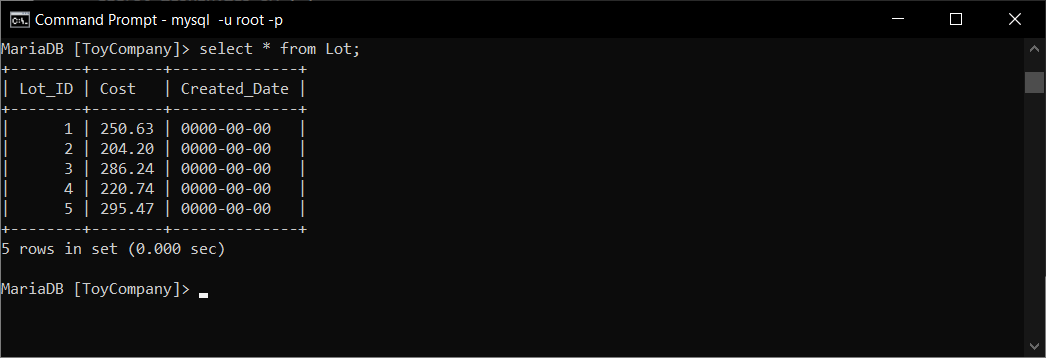
LINES TERMINATED BY '\n'

IGNORE 1 ROWS;



**After insert**

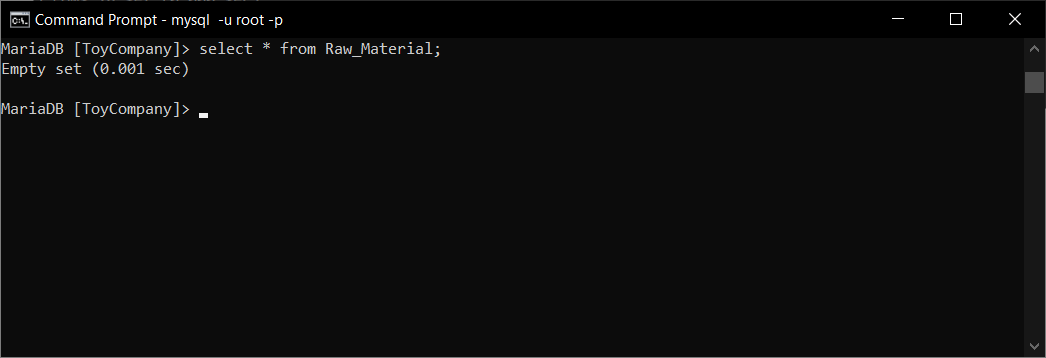
select \* from Lot;



1. **Raw\_Material**

**Before insert**

select \* from Raw\_Material;



LOAD DATA INFILE 'F:/CSV files/Raw\_Material.csv'

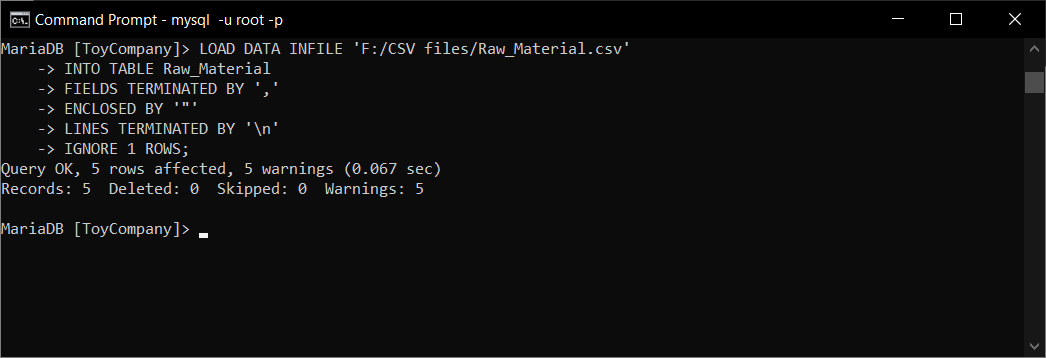
INTO TABLE Raw\_Material

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

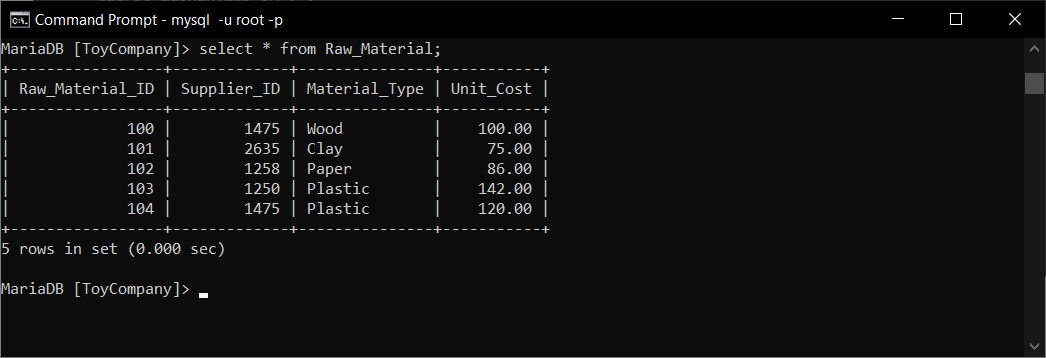
LINES TERMINATED BY '\n'

IGNORE 1 ROWS;



**After insert**

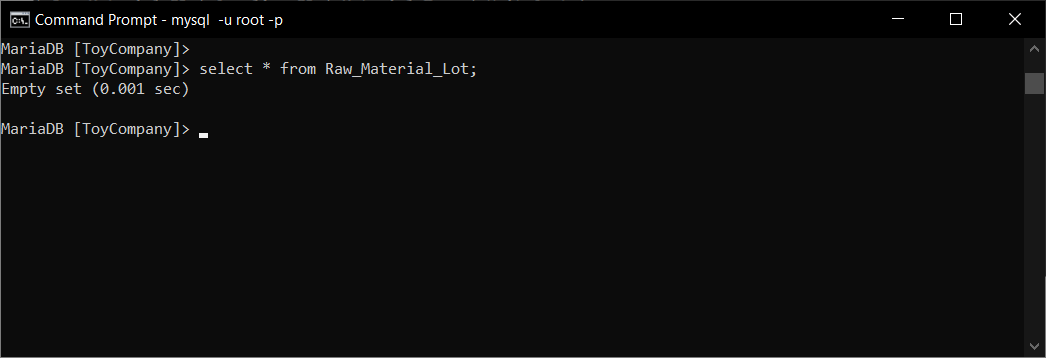
select \* from Raw\_Material;



1. **Raw\_Material\_Lot**

**Before insert**

select \* from Raw\_Material\_Lot;



LOAD DATA INFILE 'F:/CSV files/Raw\_Material\_Lot.csv'

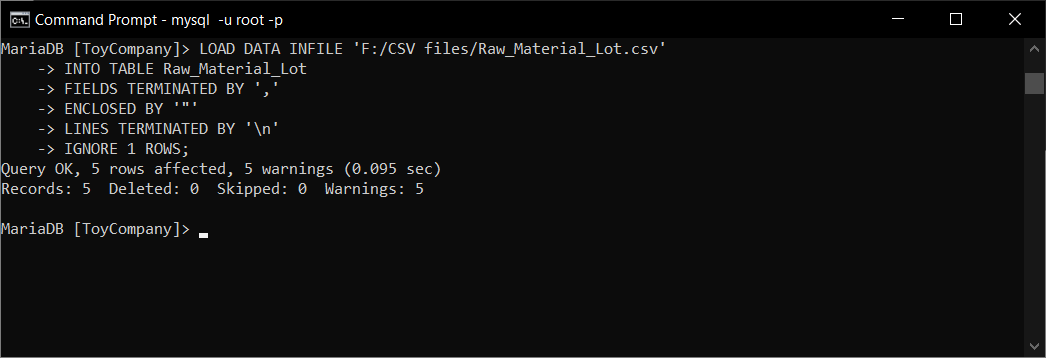
INTO TABLE Raw\_Material\_Lot

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

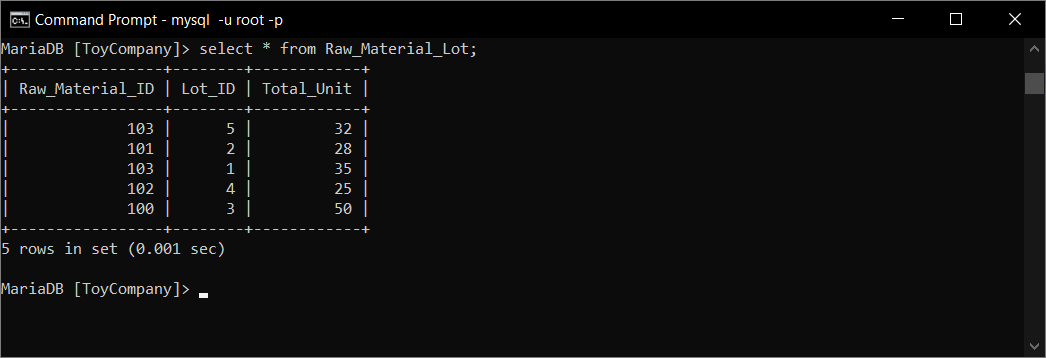
LINES TERMINATED BY '\n'

IGNORE 1 ROWS;



**After insert**

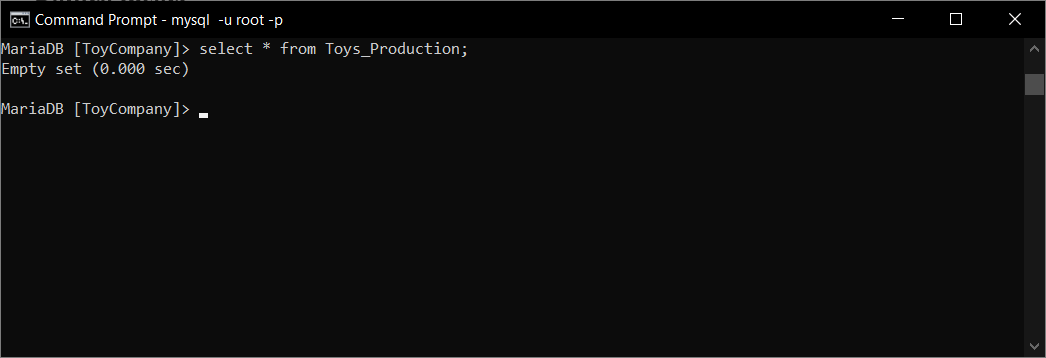
select \* from Raw\_Material\_Lot;



1. **Toys\_Production**

**Before insert**

select \* from Toys\_Production;



LOAD DATA INFILE 'F:/CSV files/Toys\_Production.csv'

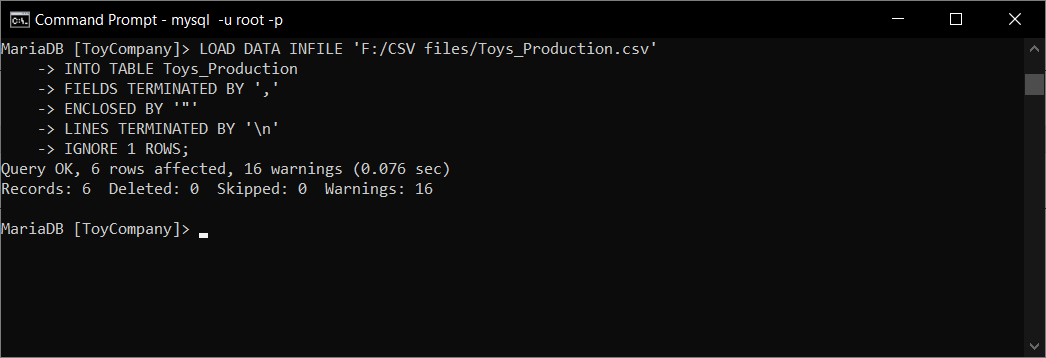
INTO TABLE Toys\_Production

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;



**After insert**

select \* from Toys\_Production;

