**Name:** Vishal Shashikant Salvi.

**UID:** 2019230069

**Class:** SE Comps

**Batch:** C

**Aim**:Practicing DDL and DML commands.

**Theory**:

**RDBMS**

RDBMS stands for "Relational Database Management System." An RDBMS is a DBMS designed specifically for relational databases. Therefore, RDBMSes are a subset of DBMSes.

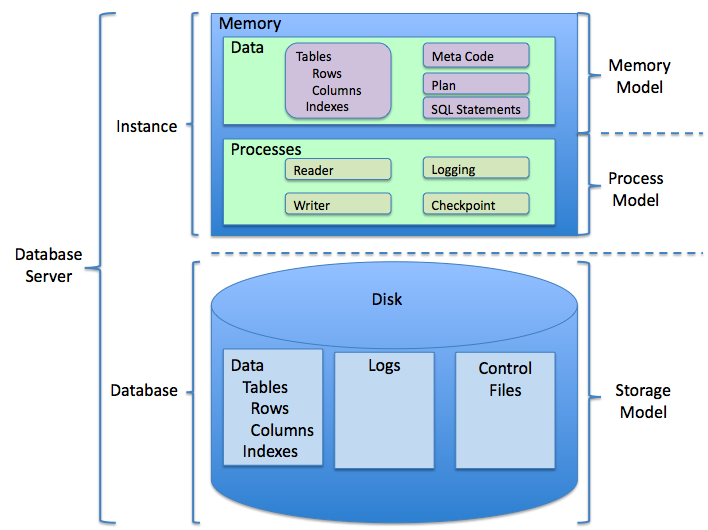
A relational database refers to a database that stores data in a structured format, using rows and columns. This makes it easy to locate and access specific values within the database. It is "relational" because the values within each table are related to each other. Tables may also be related to other tables. The relational structure makes it possible to run queries across multiple tables at once.

While a relational database describes the type of database an RDMBS manages, the RDBMS refers to the database program itself. It is the software that executes queries on the data, including adding, updating, and searching for values. An RDBMS may also provide a visual representation of the data. For example, it may display data in a tables like a spreadsheet, allowing you to view and even edit individual values in the table. Some RDMBS programs allow you to create forms that can streamline entering, editing, and deleting data.

Most well-known DBMS applications fall into the RDBMS category. Examples include Oracle Database, MySQL, Microsoft SQL Server, and IBM DB2. Some of these programs support non-relational databases, but they are primarily used for relational database management.

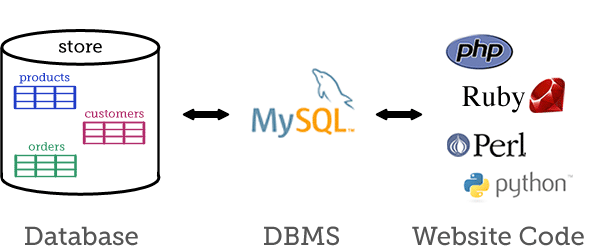
Examples of non-relational databases include Apache HBase, IBM Domino, and Oracle NoSQL Database. These type of databases are managed by other DMBS programs that support NoSQL, which do not fall into the RDBMS category.

**General Structure of Relational Database**



The term "relational database" was invented by E. F. Codd at IBM in 1970. Codd introduced the term in his research paper "A Relational Model of Data for Large Shared Data Banks". In this paper and later papers, he defined what he meant by "relational". One well-known definition of what constitutes a relational database system is composed of Codd's 12 rules. However, no commercial implementations of the relational model conform to all of Codd's rules, so the term has gradually come to describe a broader class of database systems, which at a minimum.

**MySQL**



MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into separate applications.

MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer.

MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results.

**DDL**

DDL is short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

* **CREATE** - to create a database and its objects like (table, index, views, store procedure, function, and triggers)

Syntax:

CREATE TABLE *table\_name*(

*column\_name datatype(size)*

) ;

* **ALTER** - alters the structure of the existing database

Syntax:

ALTER TABLE *table\_name*  
ADD *column\_name datatype*;

* **DROP** - delete objects from the database

Syntax:

DROP TABLE *table\_name*;

* **TRUNCATE** - remove all records from a table, including all spaces allocated for the records are removed

Syntax:

TRUNCATE TABLE *table\_name*;

* **RENAME** - rename an object

Syntax:

RENAME TABLE *table\_name1* TO *table\_name2*;

**DML**

A data manipulation language (DML) is a family of computer languages including commands permitting users to manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases.

DML resembles simple English language and enhances efficient user interaction with the system. The functional capability of DML is organized in manipulation commands like SELECT, UPDATE, INSERT INTO and DELETE FROM, as described below:

* **SELECT:**

This command is used to retrieve rows from a table.

Syntax: SELECT [column name(s)] from [table name] where [conditions].

* **UPDATE:**

This command modifies data of one or more records.

Syntax: UPDATE [table name] SET [column name = value] where [condition].

* **INSERT:**

This command adds one or more records to a database table.

Syntax: INSERT INTO [table name] [column(s)] VALUES [value(s)].

* **DELETE:**

This command removes one or more records from a table according to specified conditions.

Syntax: DELETE FROM [table name] where [condition].

**NOT NULL**

MySQL The MySQL IS NOT NULL condition is used to test for a NOT NULL value in a [SELECT](https://www.techonthenet.com/mysql/select.php), [INSERT](https://www.techonthenet.com/mysql/insert.php), UPDATE, or DELETE statement.

**Syntax:**

expression IS NOT NULL

**Note:**

* If expression is NOT a NULL value, the condition evaluates to TRUE.
* If expression is a NULL value, the condition evaluates to FALSE.

**Example**:

**With SELECT Statement**

select \* from Contacts where last\_name IS NOT NULL;

This MySQL IS NOT NULL example will return all records from the contacts table where the last\_name does not contain a null value.

**With INSERT Statement**

insert into Contacts(contact\_id, contact\_name)

Select account\_no, supplier\_name from Suppliers where category IS NOT NULL;

This MySQL IS NOT NULL example will insert records into the contacts table where the category does not contain a null value.

**With UPDATE Statement**

update Contacts set status=’completed’ where last\_name IS NOT NULL;

This MySQL IS NOT NULL example will update records in the contacts table where the last\_name does not contain a null value.

**With DELETE Statement**

delete from Contacts where last\_name IS NOT NULL;

This MySQL IS NOT NULL example will delete all records from the contacts table where the last\_name does not contain a null value.

**Primary Key**

* The PRIMARY KEY constraint uniquely identifies each record in a table.
* Primary keys must contain UNIQUE values, and cannot contain NULL values.
* A table can have only one primary key, which may consist of single or multiple fields.

**SQL PRIMARY KEY on CREATE TABLE**

CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int,

PRIMARY KEY (ID)  
);

**SQL PRIMARY KEY on ALTER TABLE**

ALTER TABLE Persons  
ADD PRIMARY KEY (ID);

**DROP PRIMARY KEY Constraint**

ALTER TABLE Persons  
DROP PRIMARY KEY;

**Unique Key**

* The UNIQUE constraint ensures that all values in a column are different.
* Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.
* A PRIMARY KEY constraint automatically has a UNIQUE constraint.
* However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

**SQL UNIQUE Constraint on CREATE TABLE**

CREATE TABLE Persons (

ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

Age int,

UNIQUE (ID)  
);

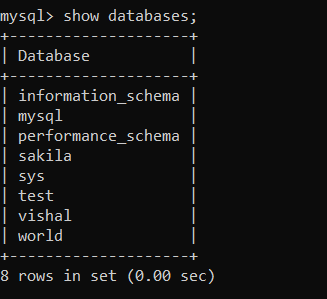
**SQL UNIQUE Constraint on ALTER TABLE**

ALTER TABLE Persons  
ADD UNIQUE (ID);

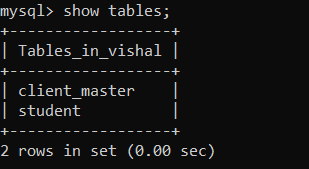
**DROP a UNIQUE Constraint**

ALTER TABLE Persons  
DROP INDEX UC\_Person;

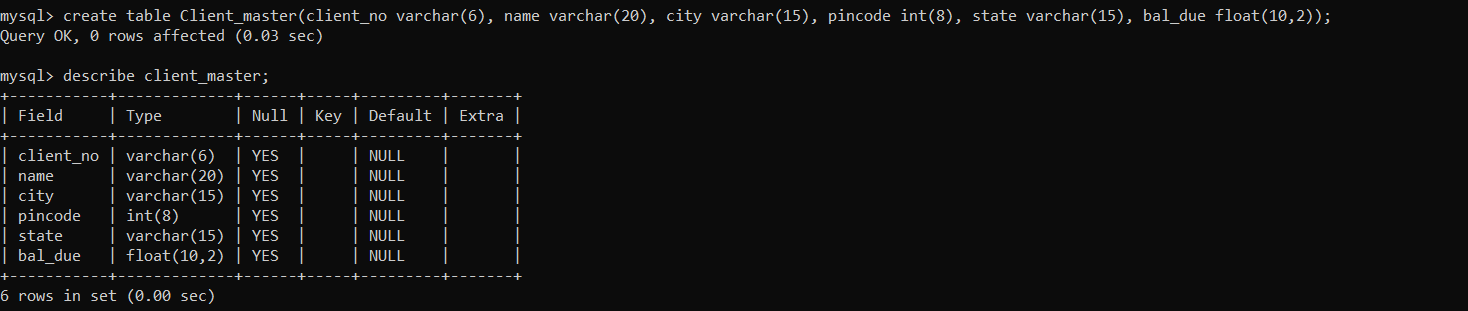
**Output**:

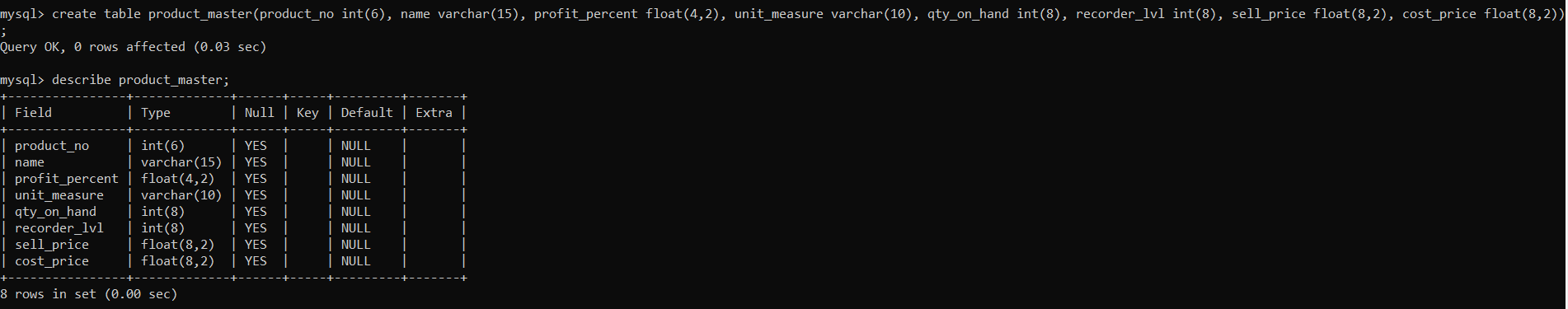






**1]Create table Client master:**





**Create table product master:**

mysql> create table product\_master(product\_no int(6), name varchar(15), profit\_percent float(4,2), unit\_measure varchar(10), qty\_on\_hand int(8), recorder\_lvl int(8), sell\_price float(8,2), cost\_price float(8,2));

Query OK, 0 rows affected (0.03 sec)

mysql> describe product\_master;

+----------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+-------------+------+-----+---------+-------+

| product\_no | int(6) | YES | | NULL | |

| name | varchar(15) | YES | | NULL | |

| profit\_percent | float(4,2) | YES | | NULL | |

| unit\_measure | varchar(10) | YES | | NULL | |

| qty\_on\_hand | int(8) | YES | | NULL | |

| recorder\_lvl | int(8) | YES | | NULL | |

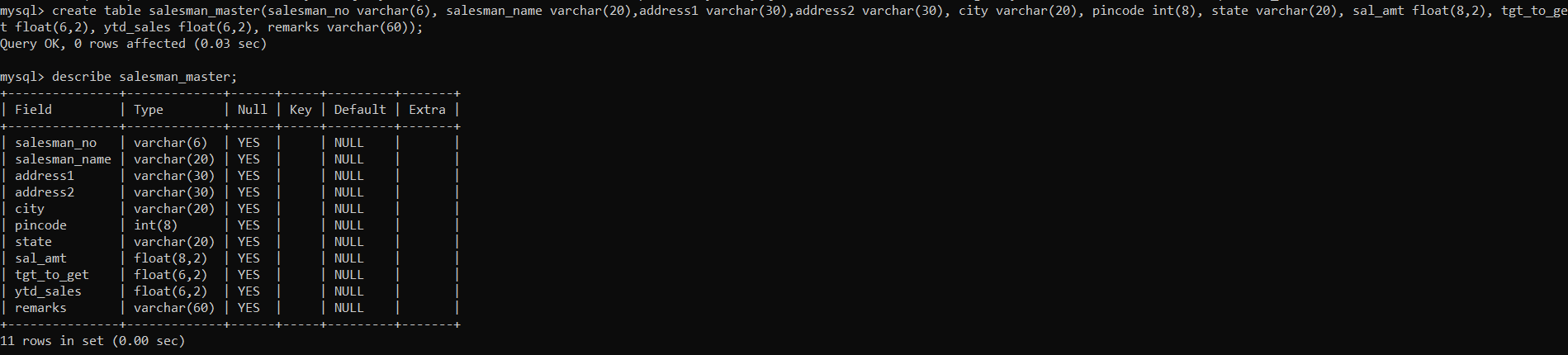
| sell\_price | float(8,2) | YES | | NULL | |

| cost\_price | float(8,2) | YES | | NULL | |

+----------------+-------------+------+-----+---------+-------+

8 rows in set (0.00 sec)

**Create table salesman master:**



mysql> create table salesman\_master(salesman\_no varchar(6), salesman\_name varchar(20),address1 varchar(30),address2 varchar(30), city varchar(20), pincode int(8), state varchar(20), sal\_amt float(8,2), tgt\_to\_get float(6,2), ytd\_sales float(6,2), remarks varchar(60));

Query OK, 0 rows affected (0.03 sec)

mysql> describe salesman\_master;

+---------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------------+-------------+------+-----+---------+-------+

| salesman\_no | varchar(6) | YES | | NULL | |

| salesman\_name | varchar(20) | YES | | NULL | |

| address1 | varchar(30) | YES | | NULL | |

| address2 | varchar(30) | YES | | NULL | |

| city | varchar(20) | YES | | NULL | |

| pincode | int(8) | YES | | NULL | |

| state | varchar(20) | YES | | NULL | |

| sal\_amt | float(8,2) | YES | | NULL | |

| tgt\_to\_get | float(6,2) | YES | | NULL | |

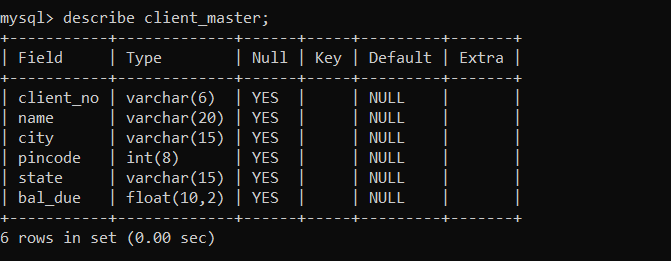
| ytd\_sales | float(6,2) | YES | | NULL | |

| remarks | varchar(60) | YES | | NULL | |

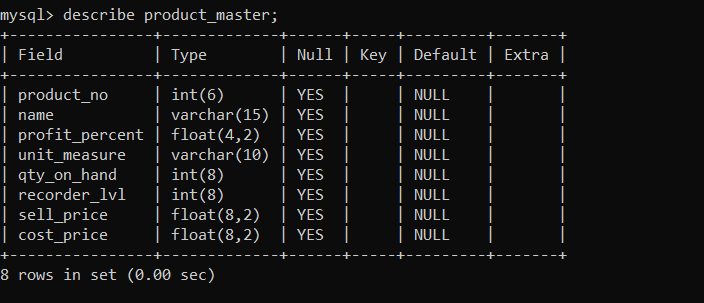
+---------------+-------------+------+-----+---------+-------+

11 rows in set (0.00 sec)

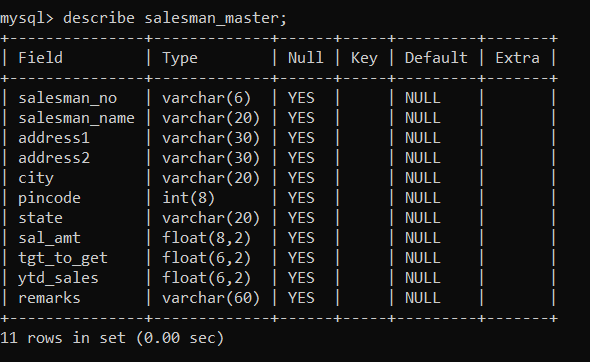
**2]Describe Client master:**



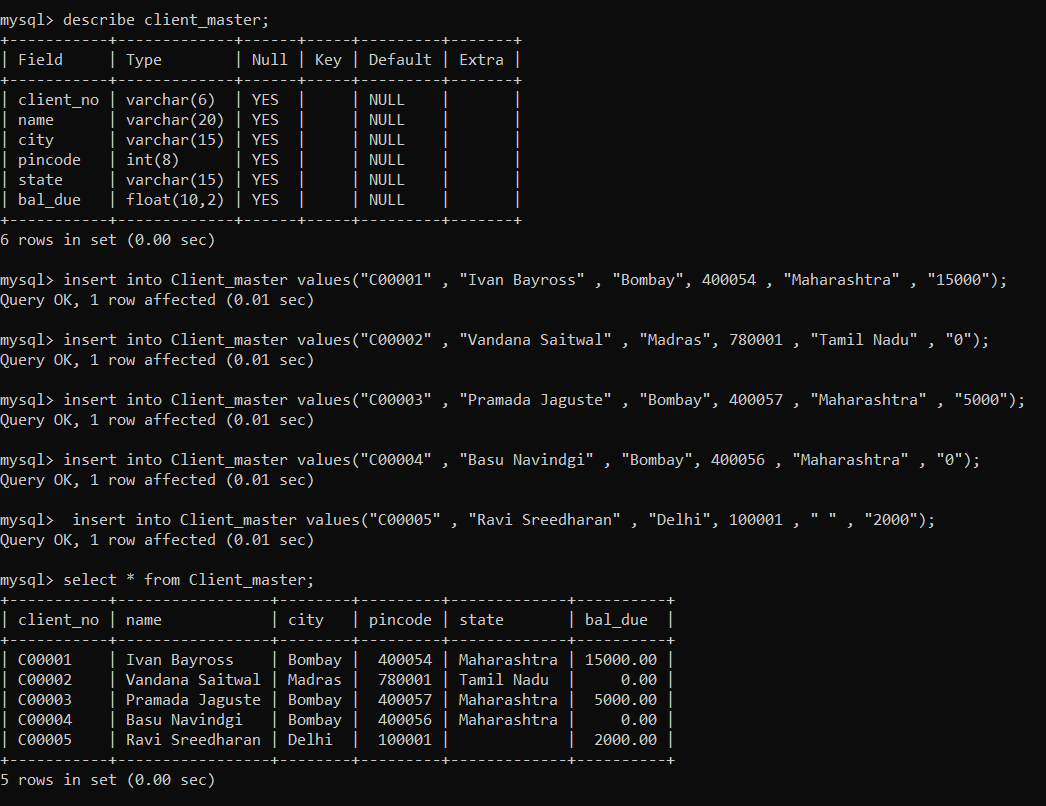
**Describe Product master:**

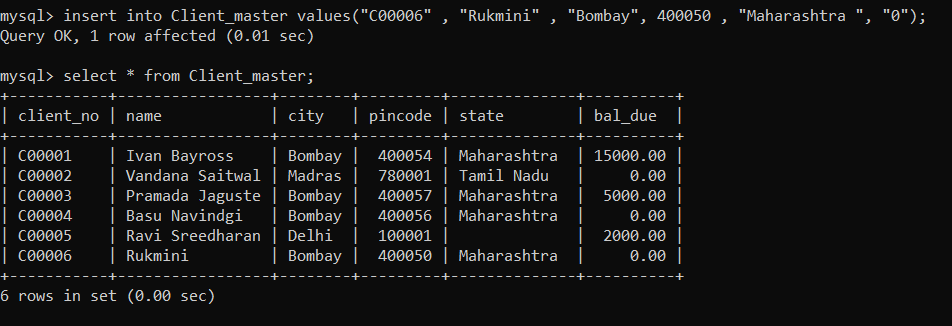


**Describe Salesman master:**

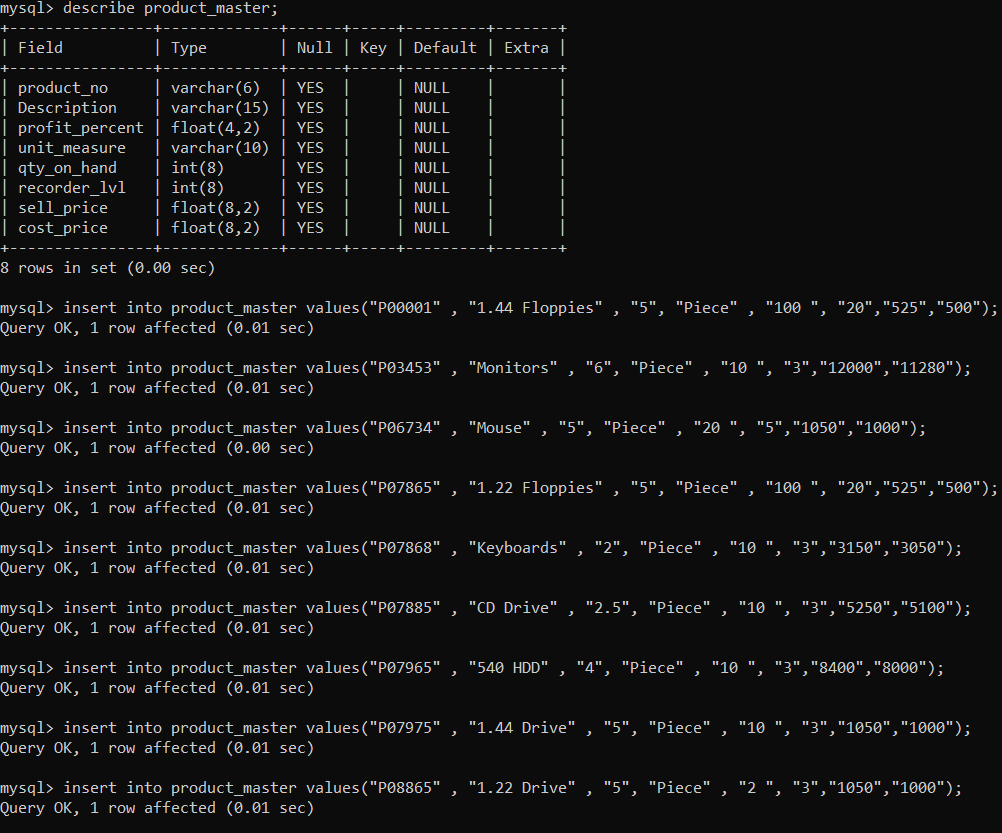


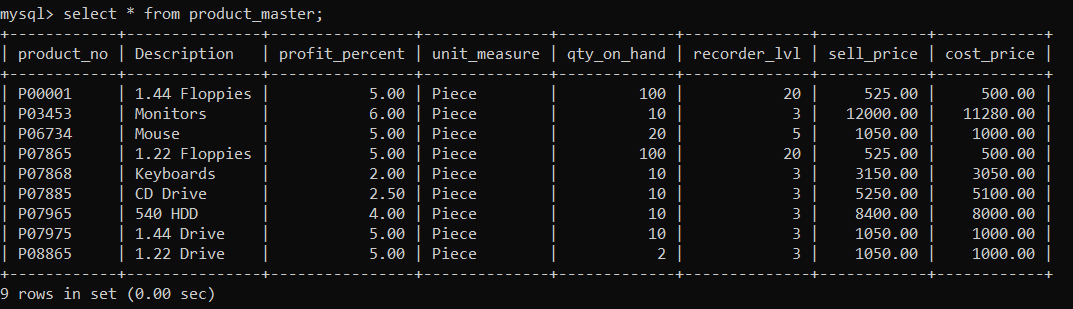
**Insert data in client master:**



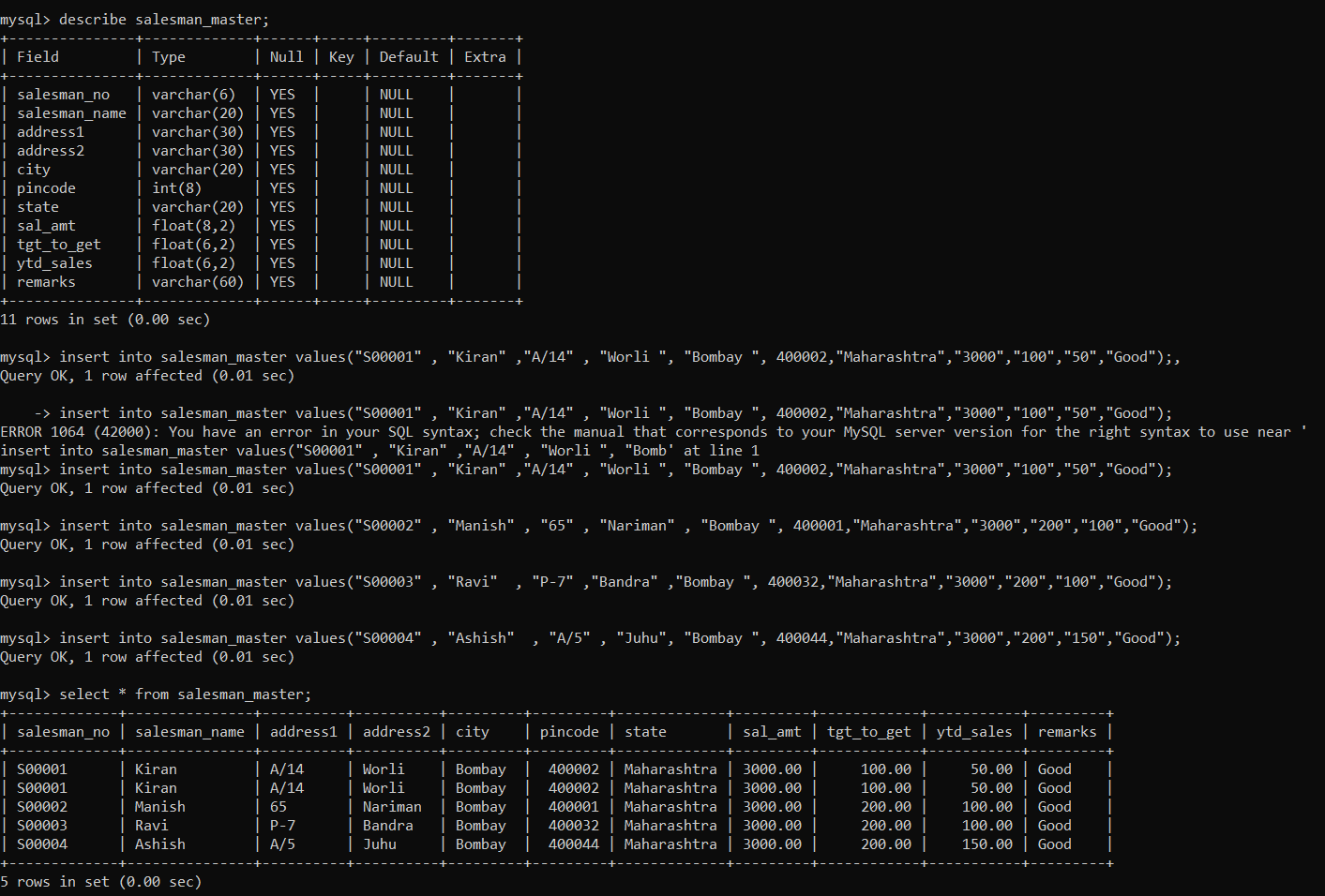


**Insert data in product master:**



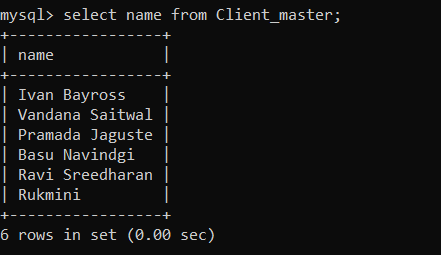


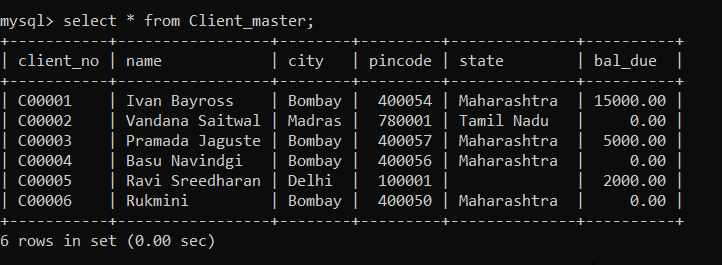
**Insert data in salesman master:**



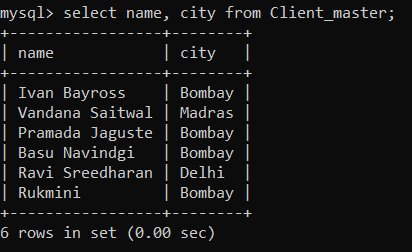
**3) Exercise on retrieving records from a table:**

**1] Find out the names of all the clients.**

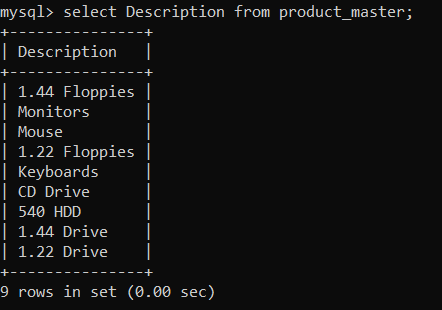


**2] Retrieve the entire contents of the client\_master table.**

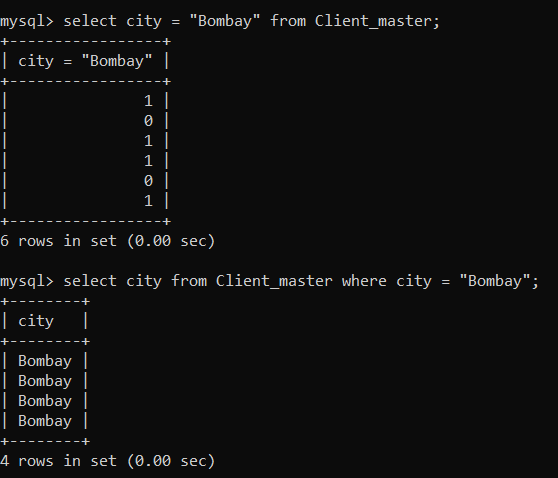
**3] Retrieve the list of names and the cities of all the clients.**



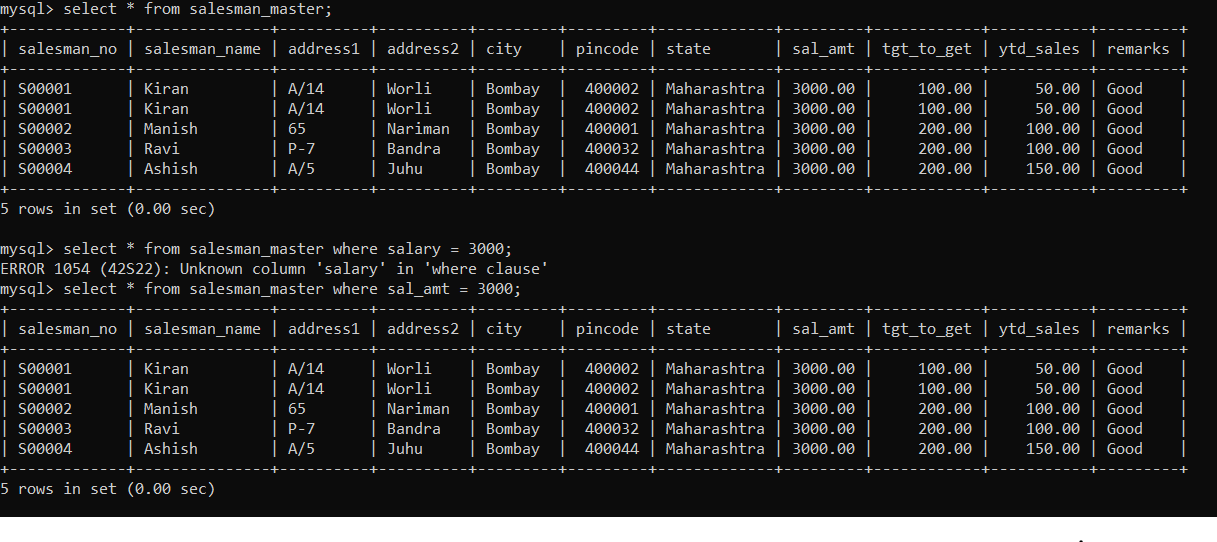
**4] List the various products available from the product\_master table**



**5] List all the clients who are located in Bombay.**

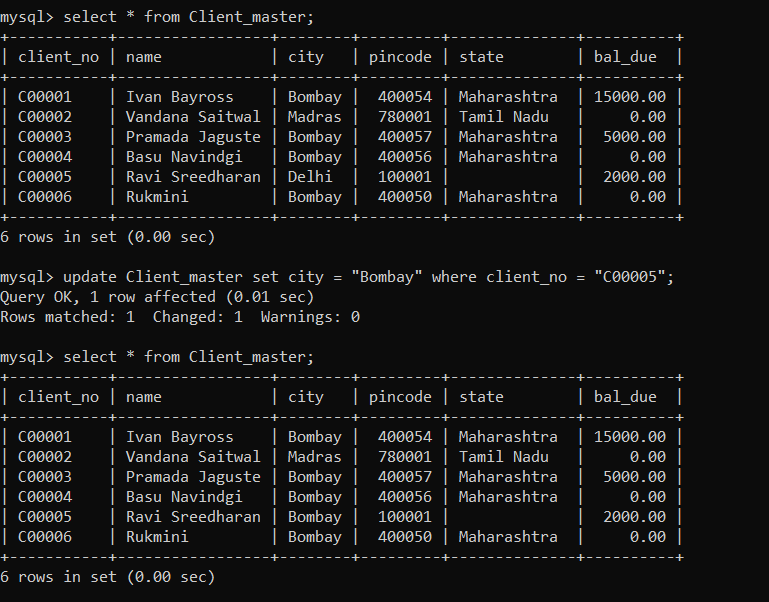


**6] Find the names of the salesman who have a salary equal to Rs.3000.**

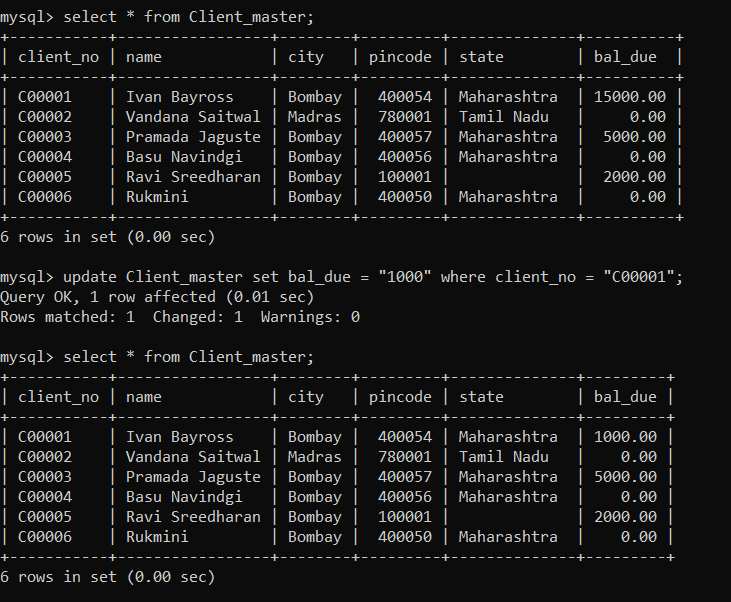


**4] Exercise on updating records in a table:**

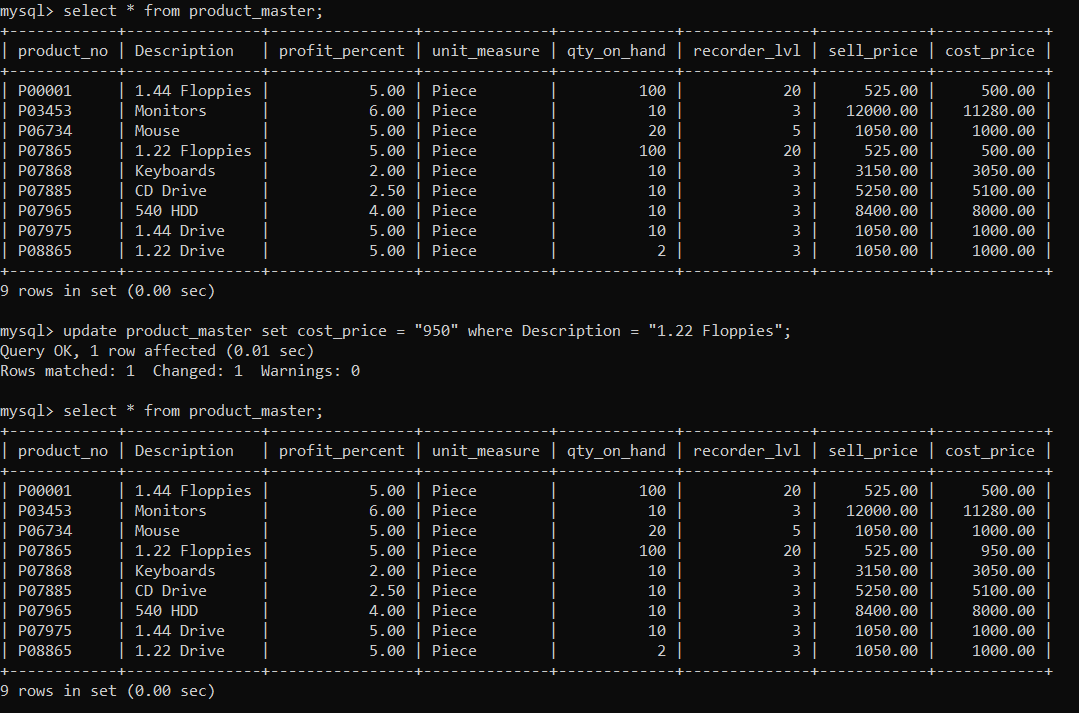
**1] Change the city of client\_no C0000S to Bombay**



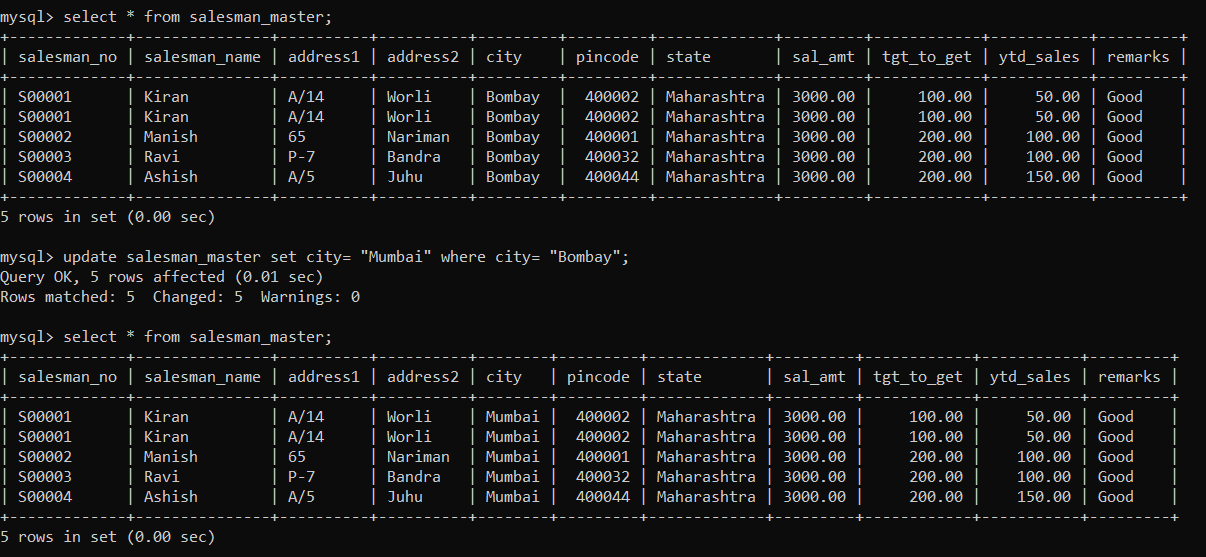
**2] Change the bal due of client no C00001 to Rs 1000.**



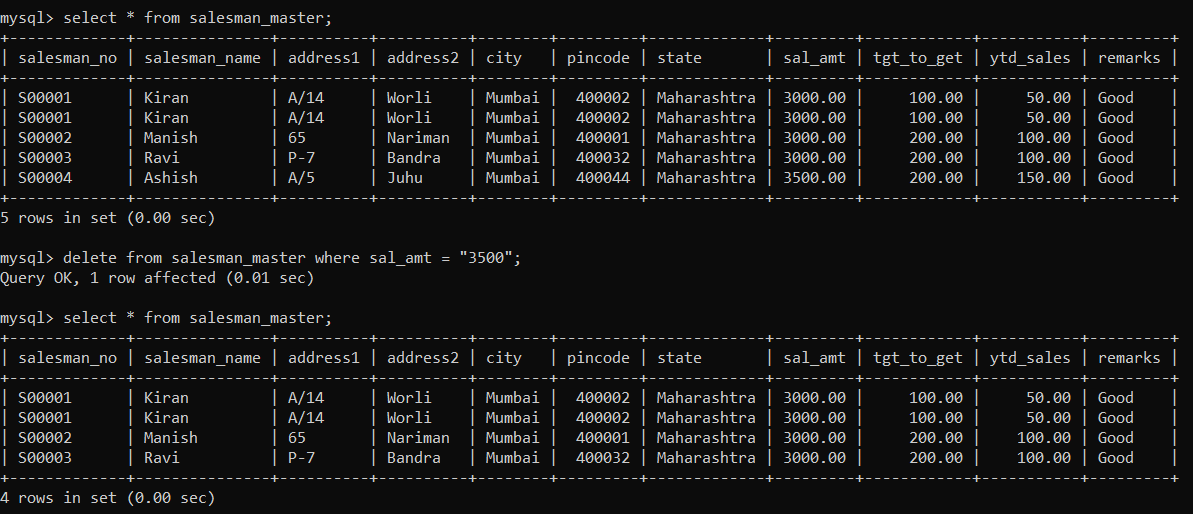
**3] Change the cost price of 1.22 Floppies to Rs.950.00.**



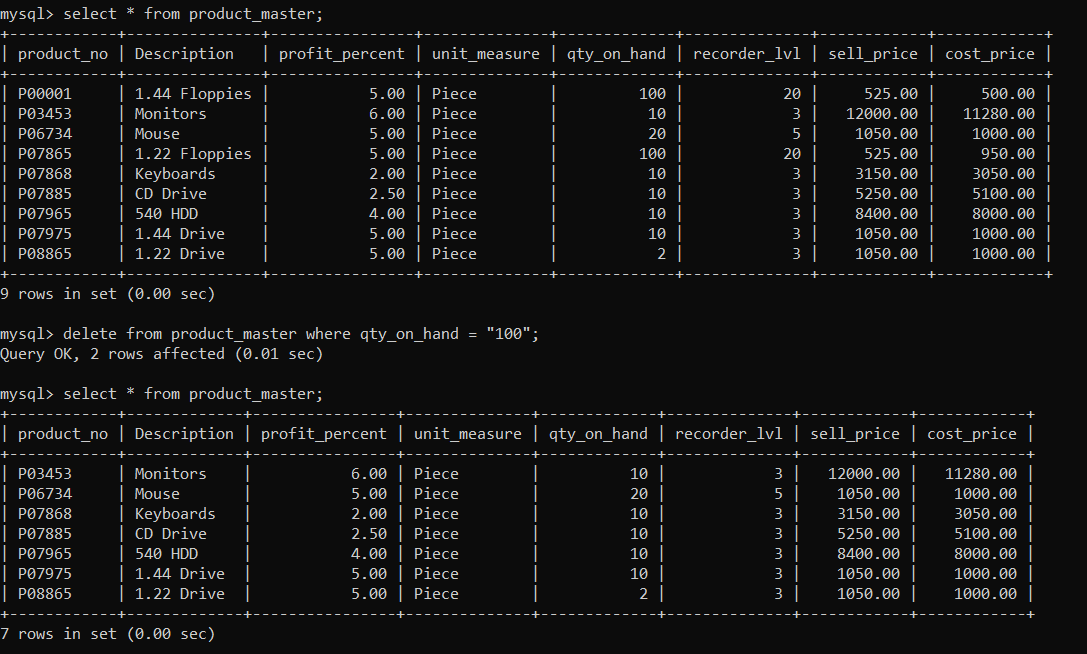
**4] Change the city of the salesman to Mumbai.**



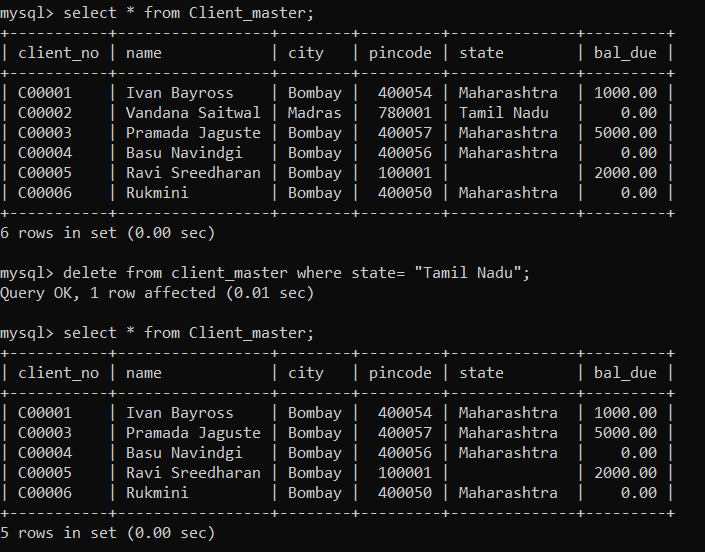
**5] Exercise on deleting records in a table:**

**1] Delete all salesmen from the salesman\_master whose salaries are equal to Rs.3500**

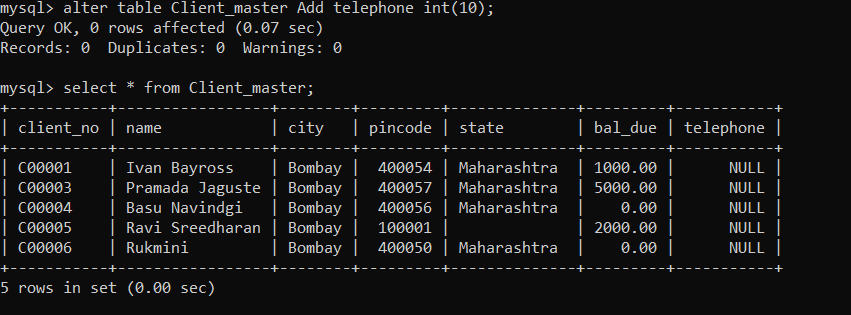
**2] Delete all products from product\_master where the quantity on hand is equal to 100**

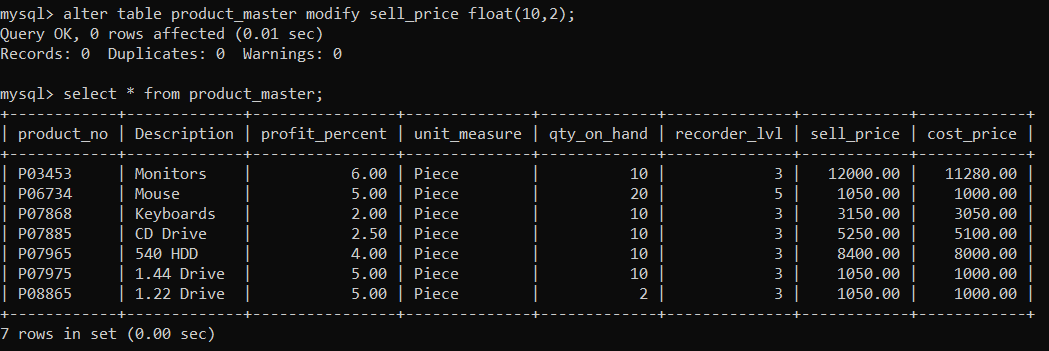


**3] Delete from client\_master where the column state holds the value Tamil Nadu 39;**

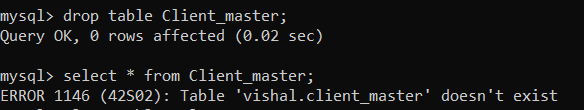


**6] Exercise on Altering the table structure:**

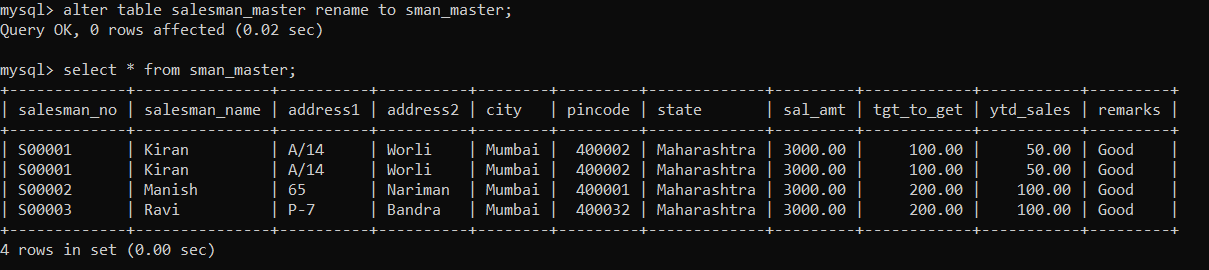




**7]Exercise on deleting the table structure along with data:**



**8]Exercise on renaming the table:**



**Conclusion**: Thus from this experiment we implement queries on DML as well as DDL and studied about DBMS.