Implementing SQL Queries Using MySQL - I

Session 6

Objectives

- *Describe the commands to view and alter a database*
- *Explain the commands to retrieve data from a table*
- *Describe the commands to modify the table definitions*
- *Describe the commands to delete the table definitions*

Viewing Information about Databases on the File System 1-39

- You will use the SHOW command to display the list of databases and index keys along with the privileges
- The SHOW command provides information about databases, tables, columns, and server status
- To view a list of databases present on the server, use the following syntax:

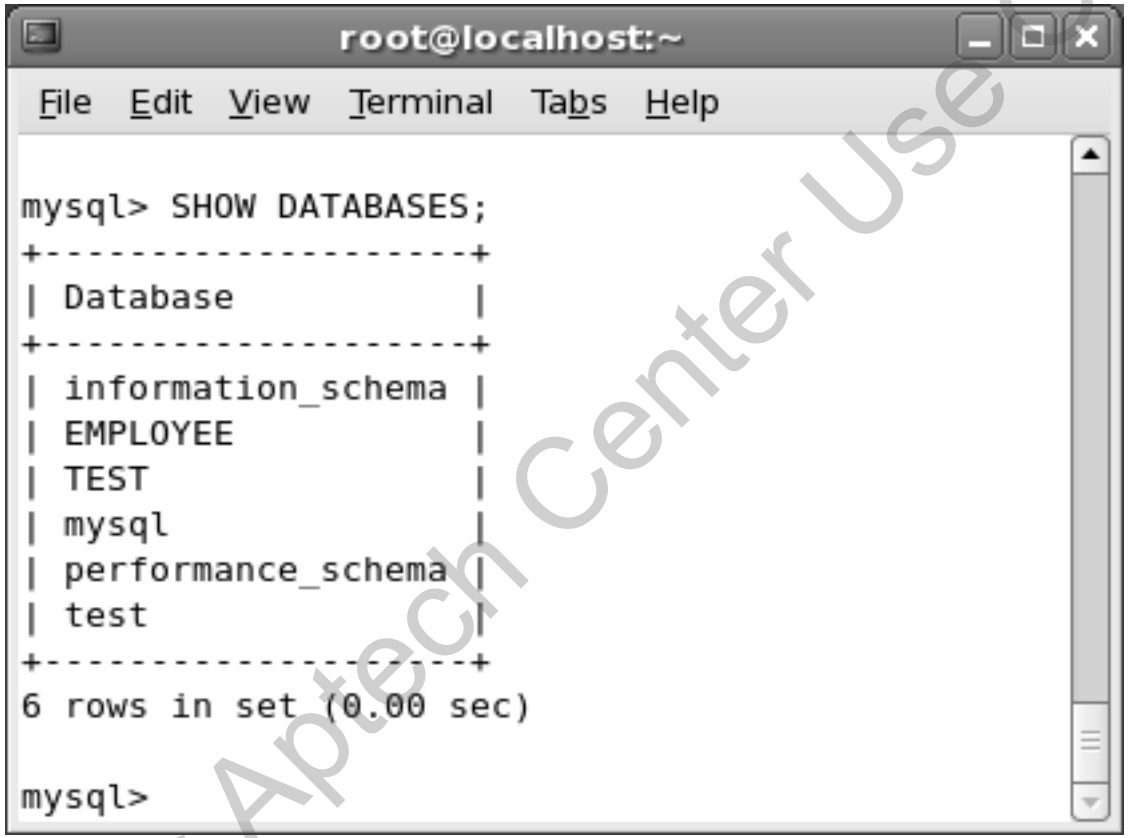
```
SHOW DATABASES;
```

where,

SHOW – displays the object specified in the clause

DATABASES – displays the databases present in the instance of MySQL

Figure displays the output of the `SHOW DATABASES` command



- The output lists all the databases present on the server
- In addition, it also displays the time taken to execute the command in seconds

- To view a list of databases that begin or contain specified characters in the database name, use the following syntax:

```
SHOW DATABASES [LIKE <condition>];
```

where,

`SHOW DATABASES` – displays the list of databases present in the instance of MySQL

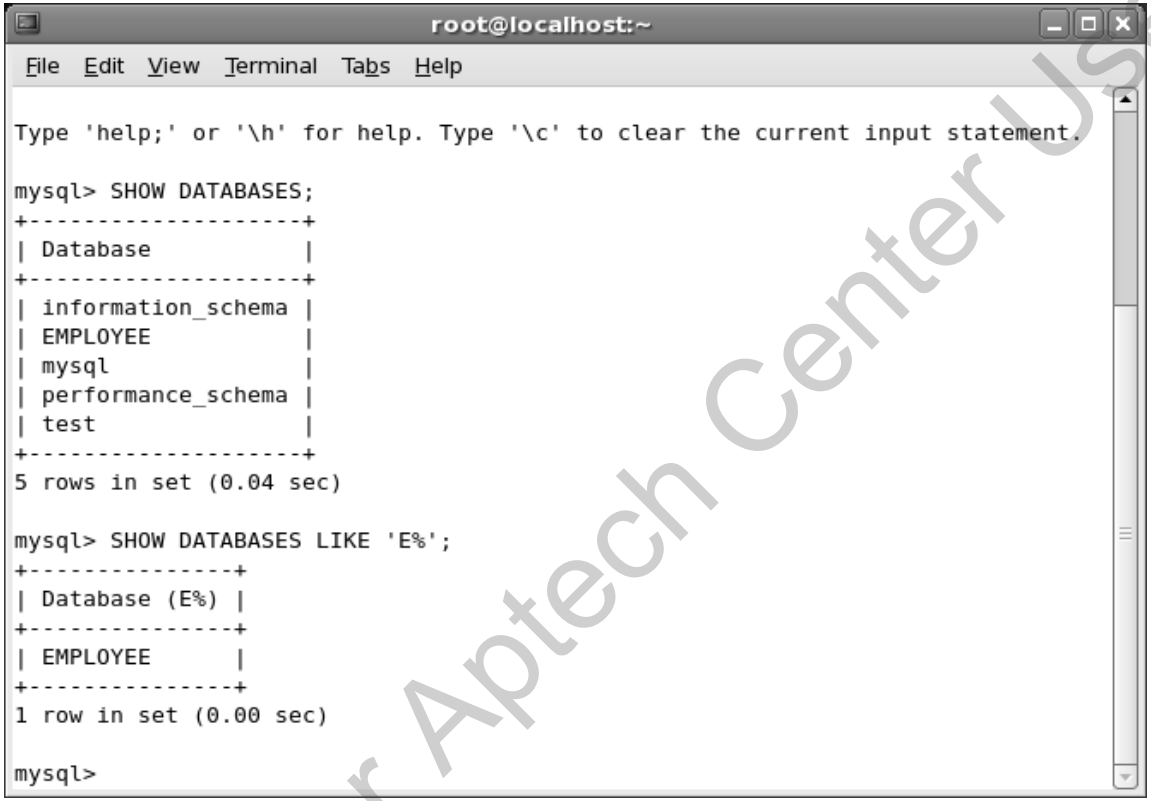
`LIKE <condition>` – contains conditions that must be satisfied before displaying the list of databases

- The `LIKE` clause can use conditions with wildcard characters such as `'%'` and `'_'`
- This clause is optional
- The `'%'` represents any string of zero or more characters
- The `'_'` represents any single character

- For example, to view all databases that begin with the letter ‘E’, enter the following command at the command prompt:

```
SHOW DATABASES LIKE 'E%';
```

The output lists all the databases starting with the letter ‘E’, present on the server



Viewing Information about Databases on the File System 8-39

- To view a list of tables in a database, use the following syntax:

```
SHOW TABLES [FROM database_name] [LIKE <condition>];
```

where,

`SHOW TABLES` – displays the tables from the selected database

`FROM database_name` – displays the tables from the database specified in the clause

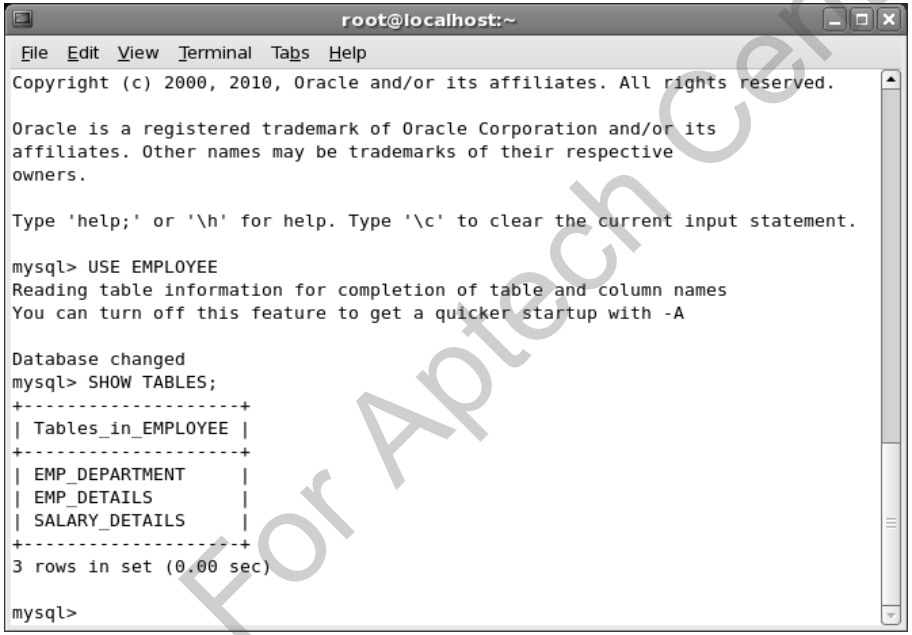
- The `FROM` clause enables you to specify the database name for which tables are to be listed
- You can use this keyword when you have not activated a database from the server with the USE command

`LIKE <condition>` – displays the tables from the database based on the condition specified in the clause

- For example, to view the tables of the EMPLOYEE database that has been set as current database through the USE command, enter the following command at the command prompt:

```
SHOW TABLES;
```

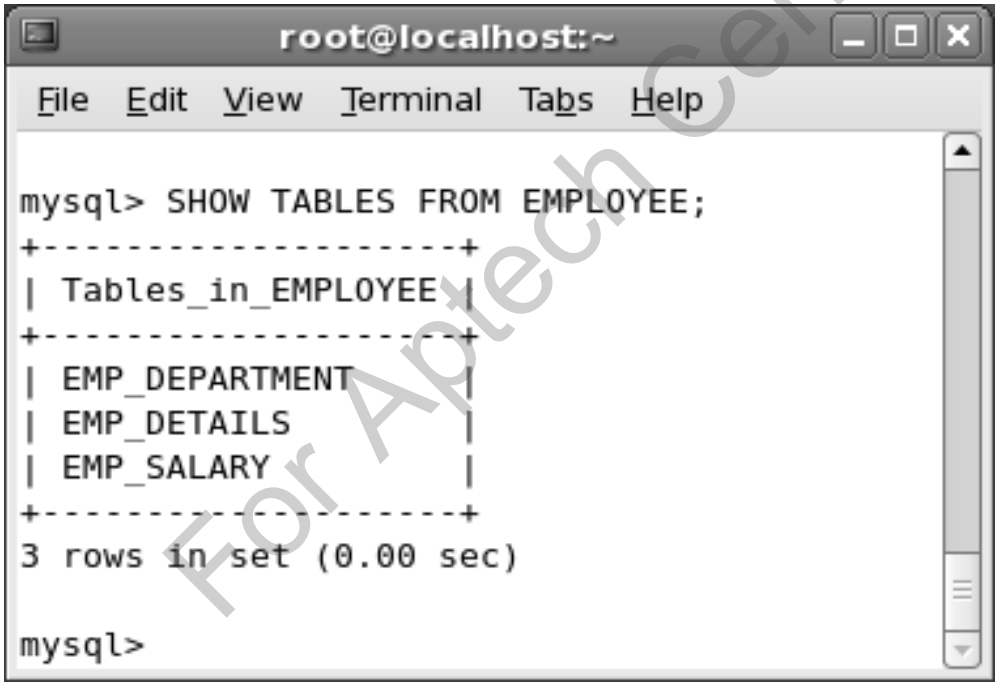
Figure displays the output of the command



- To view a list of tables in a database when the database has not been selected earlier through the `USE` command, enter the following command at the command prompt:

```
SHOW TABLES FROM EMPLOYEE;
```

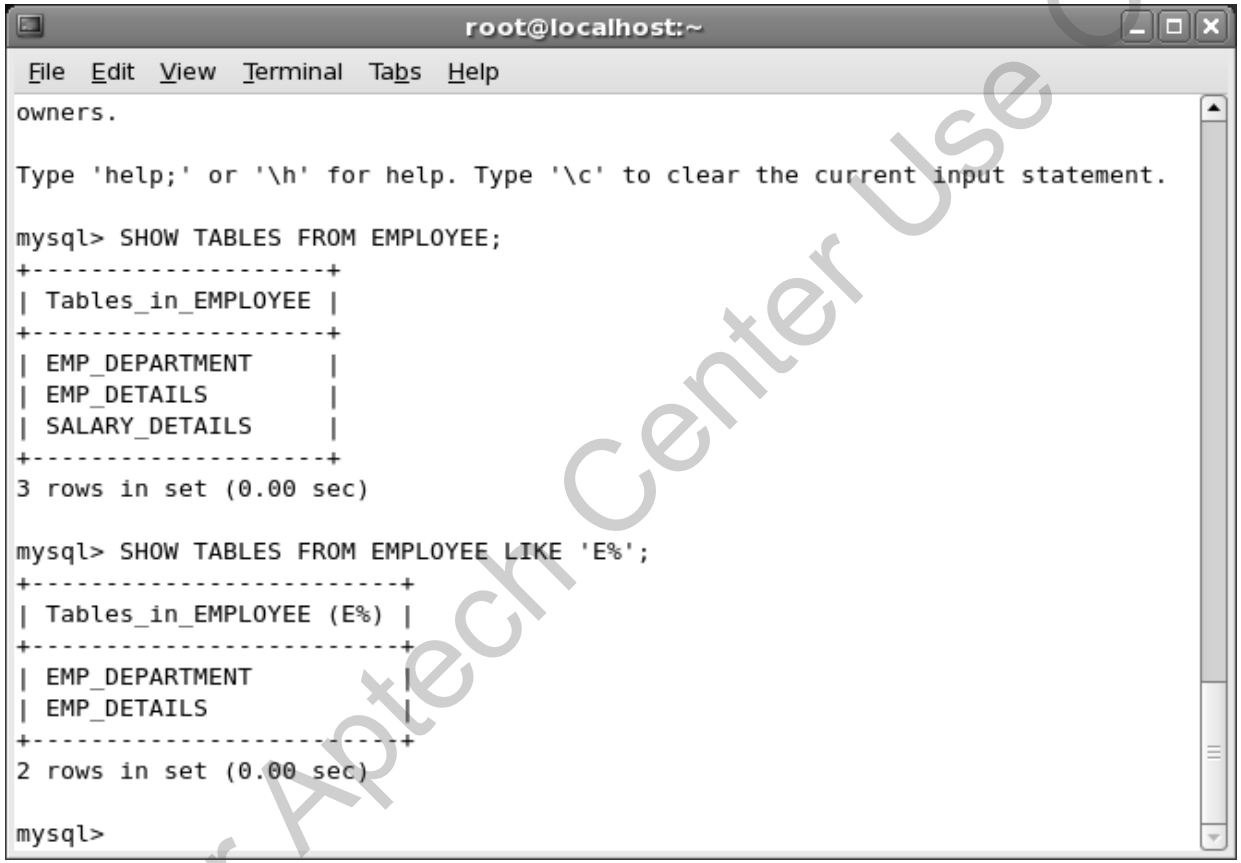
Figure displays the output of the command



- You will use the FROM keyword here because the EMPLOYEE database has not been activated with the USE command before displaying the tables;
- To view tables that begin with the letter 'E' from the EMPLOYEE database, enter the following command at the command prompt:

```
SHOW TABLES FROM EMPLOYEE LIKE 'E%';
```

Figure displays the output of the command



- Figure displays all the tables present in the EMPLOYEE database starting with letter 'E' and any number of characters after the letter E
- The LIKE keyword is used to display only those tables, which matches with the specified letter
- To view the column structure of a table from the database, use the following syntax:

```
SHOW COLUMNS FROM table_name [FROM database_name] [LIKE
clauses];
```

where,

table_name – specifies the name of the table

COLUMNS – displays the columns from the specified table

table_name – specifies the name of the table that contains the columns

database_name – specifies the name of the database

LIKE – specifies conditions, if any

- For example, to view the column structure of the EMP_DETAILS table, enter the following command at the command prompt:

```
SHOW COLUMNS FROM EMP_DETAILS;
```

Figure displays the output of the command

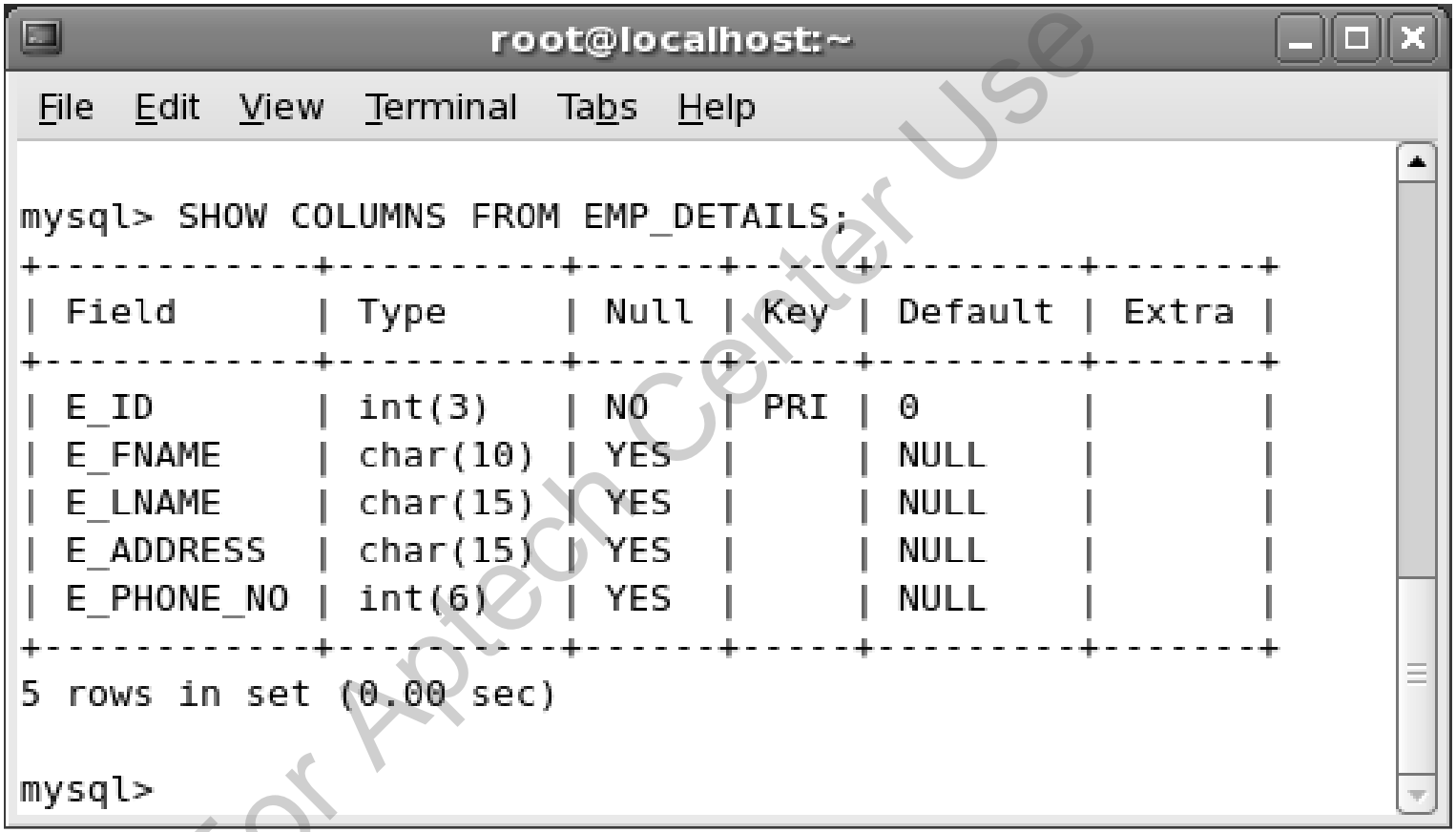


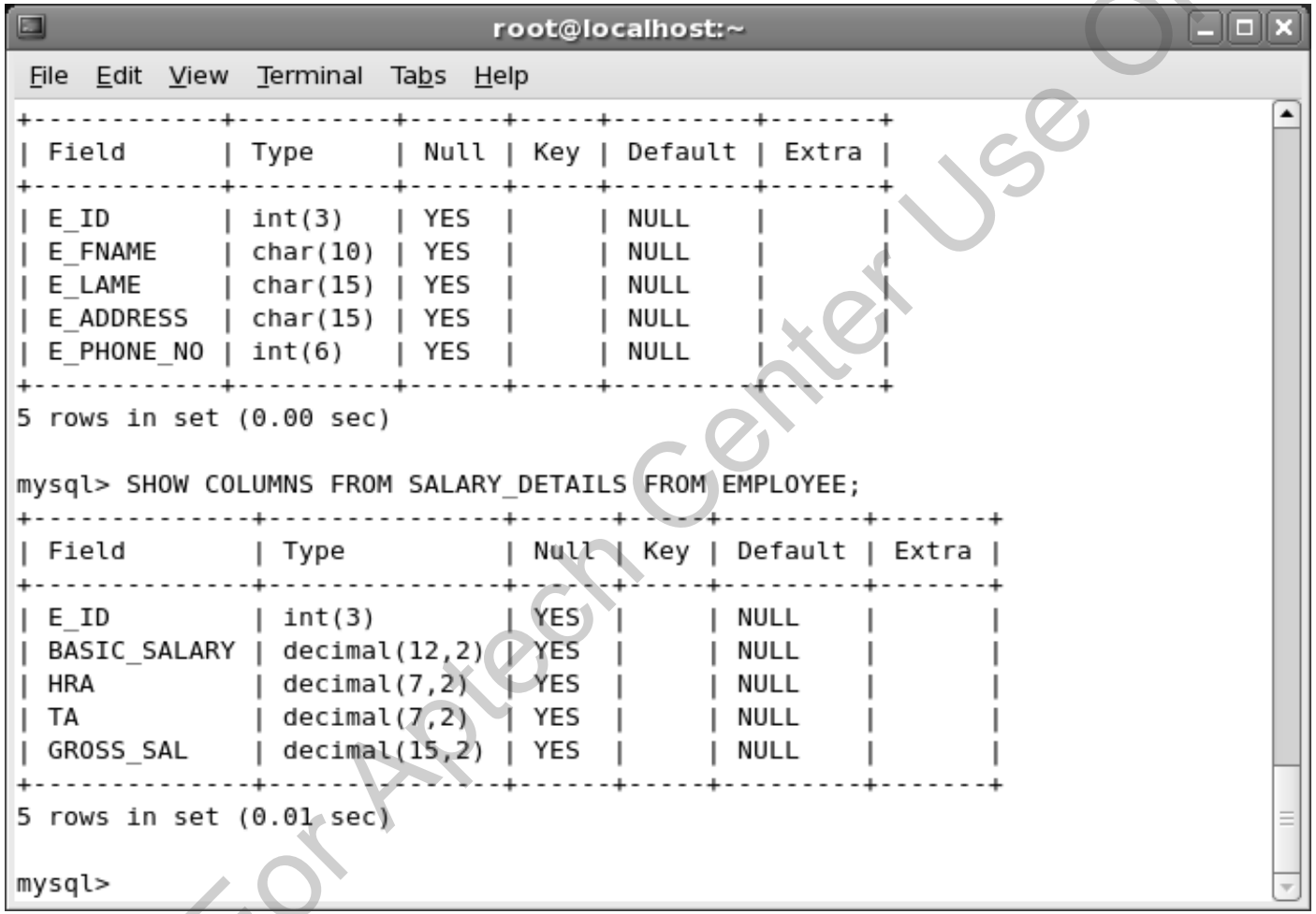
Table lists the description of each of the columns displayed in the figure

Column	Description
FIELD	Indicates column name
TYPE	Indicates the data type for a column
NULL	Specifies that the column can contain empty values
KEY	Specifies if the column is indexed
DEFAULT	Specifies the default value of the column
EXTRA	Specifies additional characteristics for the columns

- To display the same output with the EMPLOYEE database specified explicitly, enter the following command at the command prompt:

```
SHOW COLUMNS FROM SALARY_DETAILS FROM EMPLOYEE;
```

Figure displays the output of the command



- The first `FROM` keyword specifies the table to use for retrieving data
- The second `FROM` keyword is used because you have not selected the database earlier with the `USE` command

Viewing Information about Databases on the File System 20-39

- A database index is a data structure that speeds up retrieval operations on a table
- Indexes can be created using one or more columns of a database table
- They enable faster searches and efficient organization of data

- To view the index in a table from a database, use the following syntax:

```
SHOW INDEX FROM table_name [FROM database_name];
```

where,

`SHOW INDEX` - displays the index information of the table specified in the table_name clause of the command

`FROM table_name` – specifies the name of the table to retrieve the index

`FROM database_name` - specifies the name of the database where the table exists

The `SHOW INDEX` command displays index information as shown in table:

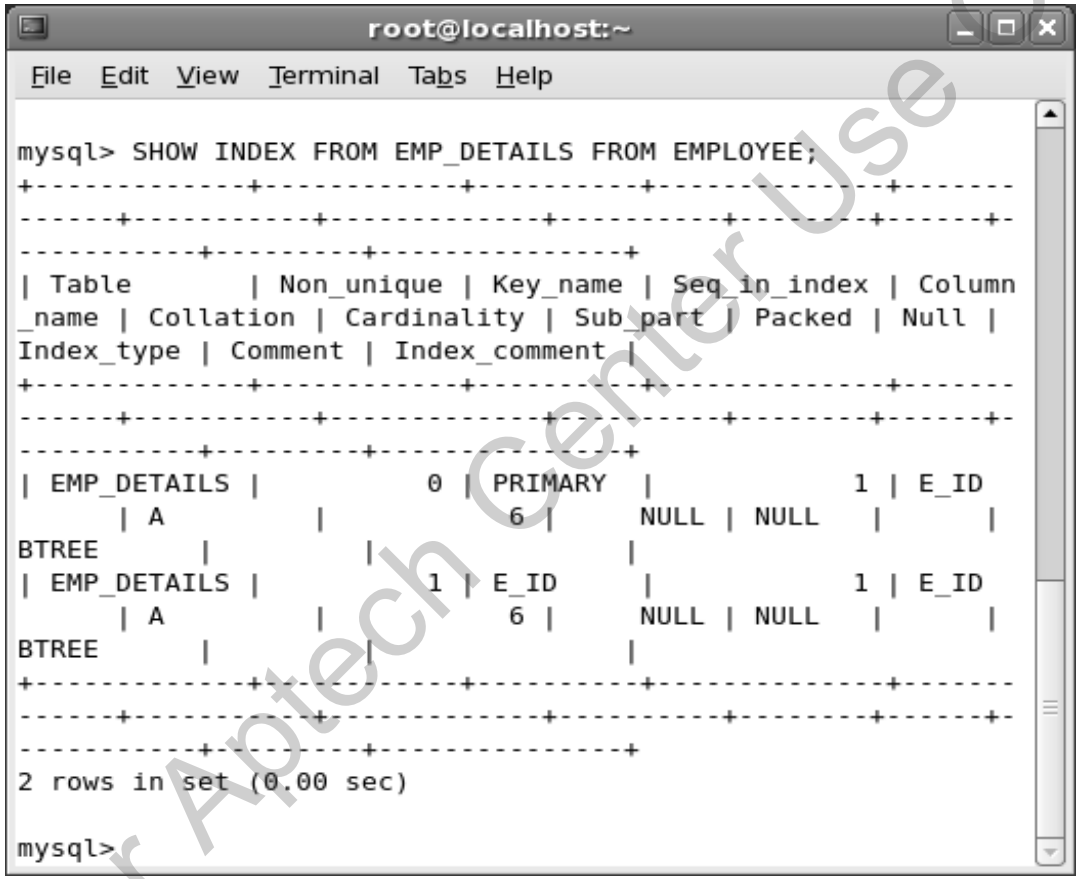
Column	Description
Table	Displays name of the table
Non_unique	Displays 0 if the index cannot contain duplicate value
Key_name	Displays index name
Seq_in_index	Displays sequence number of columns in index, starting with 1
Column_name	Displays column name
Collation	Displays the sorting order of columns in the index
Cardinality	Displays number of unique values in the index
Sub_part	Displays number of indexed characters when the column is partly indexed

Viewing Information about Databases on the File System 23-39

- For example, to view the index in the `EMP_DETAILS` table of the `EMPLOYEE` database, enter the following command at the command prompt:

```
SHOW INDEX FROM EMP_DETAILS FROM EMPLOYEE;
```

Figure displays the output of the command



- Alternatively, you can use the following syntax

```
SHOW INDEX FROM database_name.table_name;
```

where,

`SHOW INDEX` – displays the table index

`database_name` – specifies the name of the database that contains the table

`table_name` – specifies the name of the table for which you want to display the index

- For example, to display index keys of `EMP_DETAILS` from `EMPLOYEE` database, enter the following command at the command prompt:

```
SHOW INDEX FROM EMPLOYEE.EMP_DETAILS;
```

- This command is more compact and concise
- To view the server status, use the following syntax:

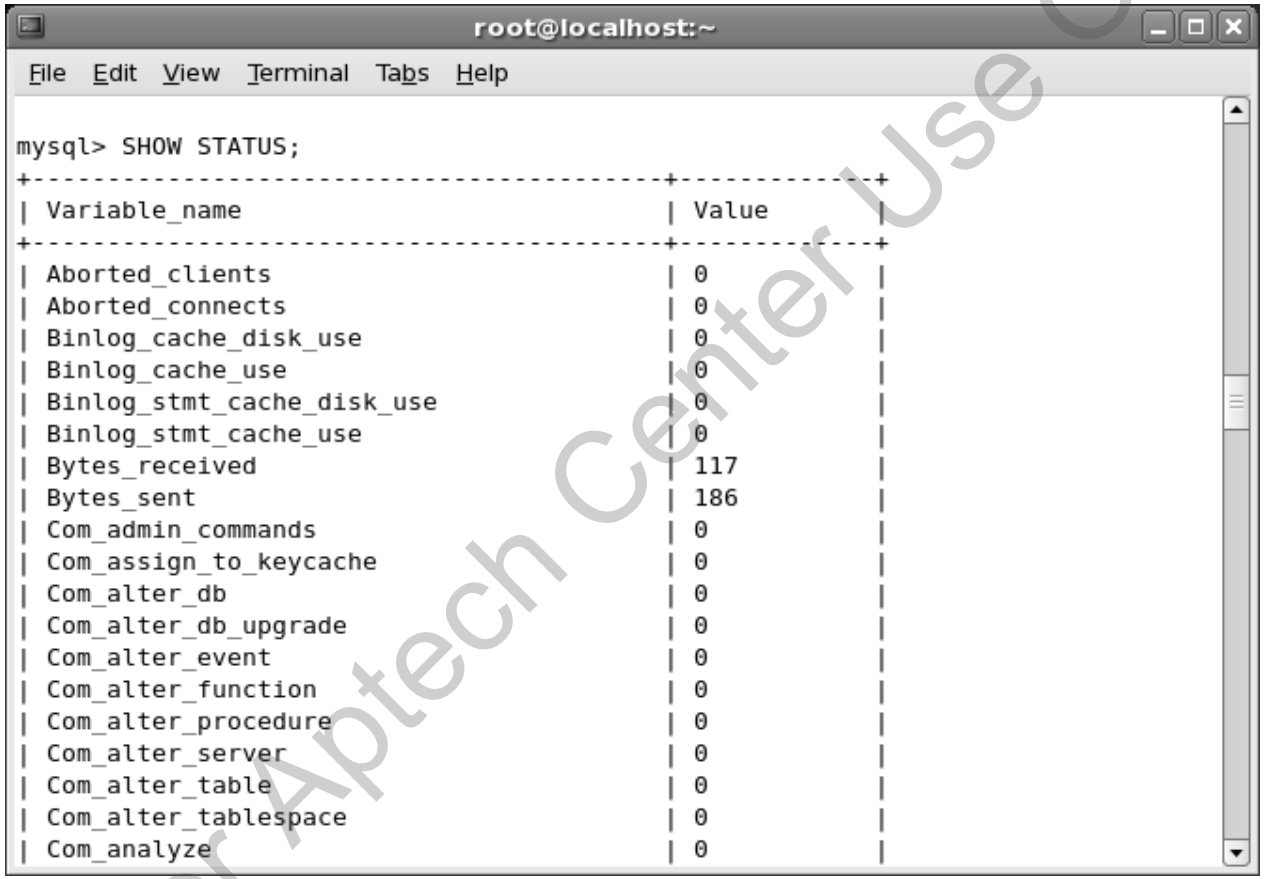
```
SHOW STATUS;
```

where,

`SHOW` – displays the object specified in the clause

`STATUS` – displays information about the server

Figure displays the output of the command

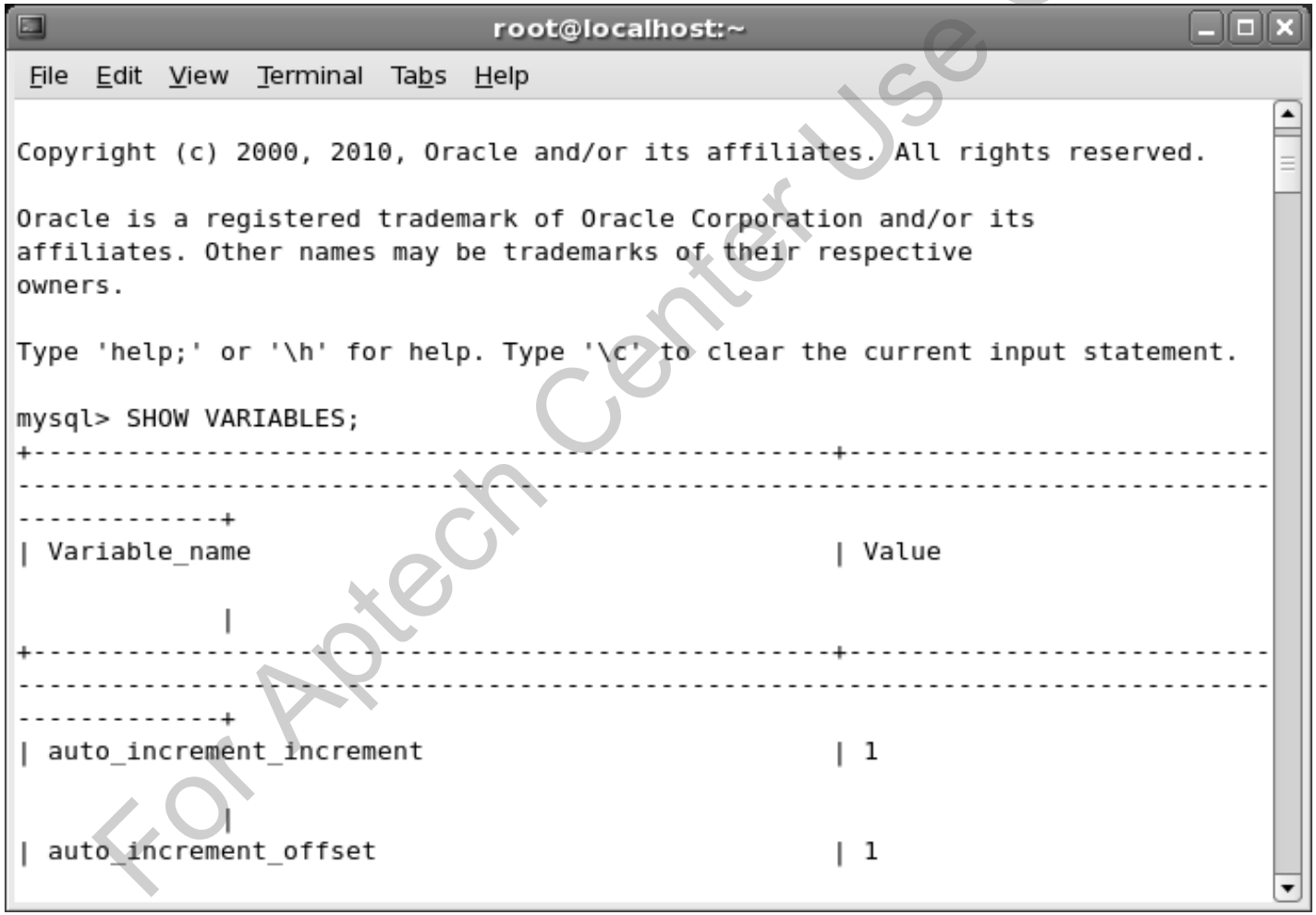


Viewing Information about Databases on the File System 28-39

- To view the values of the system variables, use the following syntax:

```
SHOW VARIABLES;
```

Figure displays the output of the command



Viewing Information about Databases on the File System 30-39

- You can change the default values of the variables using the command-line options when MySQL starts
- The `SET` statement can also be used to edit the default values at runtime
- You can also view the system variable information using the `mysqladmin` command

Viewing Information about Databases on the File System 31-39

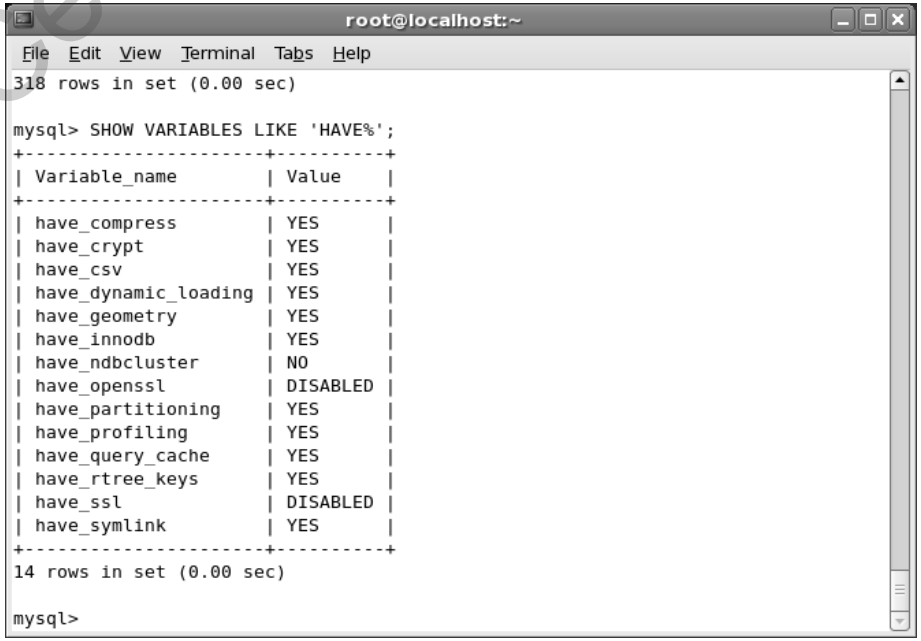
- To view only the variables that match a specified condition use the following syntax:

```
SHOW VARIABLES [LIKE <condition>];
```

- To display the variables that start with 'HAVE', enter the following command at the command prompt:

```
SHOW VARIABLES LIKE 'HAVE%';
```

The output of the command



Viewing Information about Databases on the File System 32-39

- To view the running threads or processes, use the following syntax:

```
SHOW [FULL] PROCESSLIST;
```

- You must have `SUPER` privileges to view all the threads
- The user accounts that do not have `SUPER` privileges can view only their threads

Viewing Information about Databases on the File System 33-39

- The `SHOW TABLE STATUS` command is similar to the `SHOW TABLE` command except that it displays more information about each table. The syntax for `SHOW TABLE STATUS` command is as follows:

```
SHOW TABLE STATUS [FROM database_name] [LIKE <condition>];
```

where,

`SHOW TABLE STATUS` – displays information about tables in the database

`FROM database_name` – enables you to specify the database to select tables

`LIKE <condition>` – specifies one or more conditions to be satisfied before displaying the table information

Viewing Information about Databases on the File System 34-39

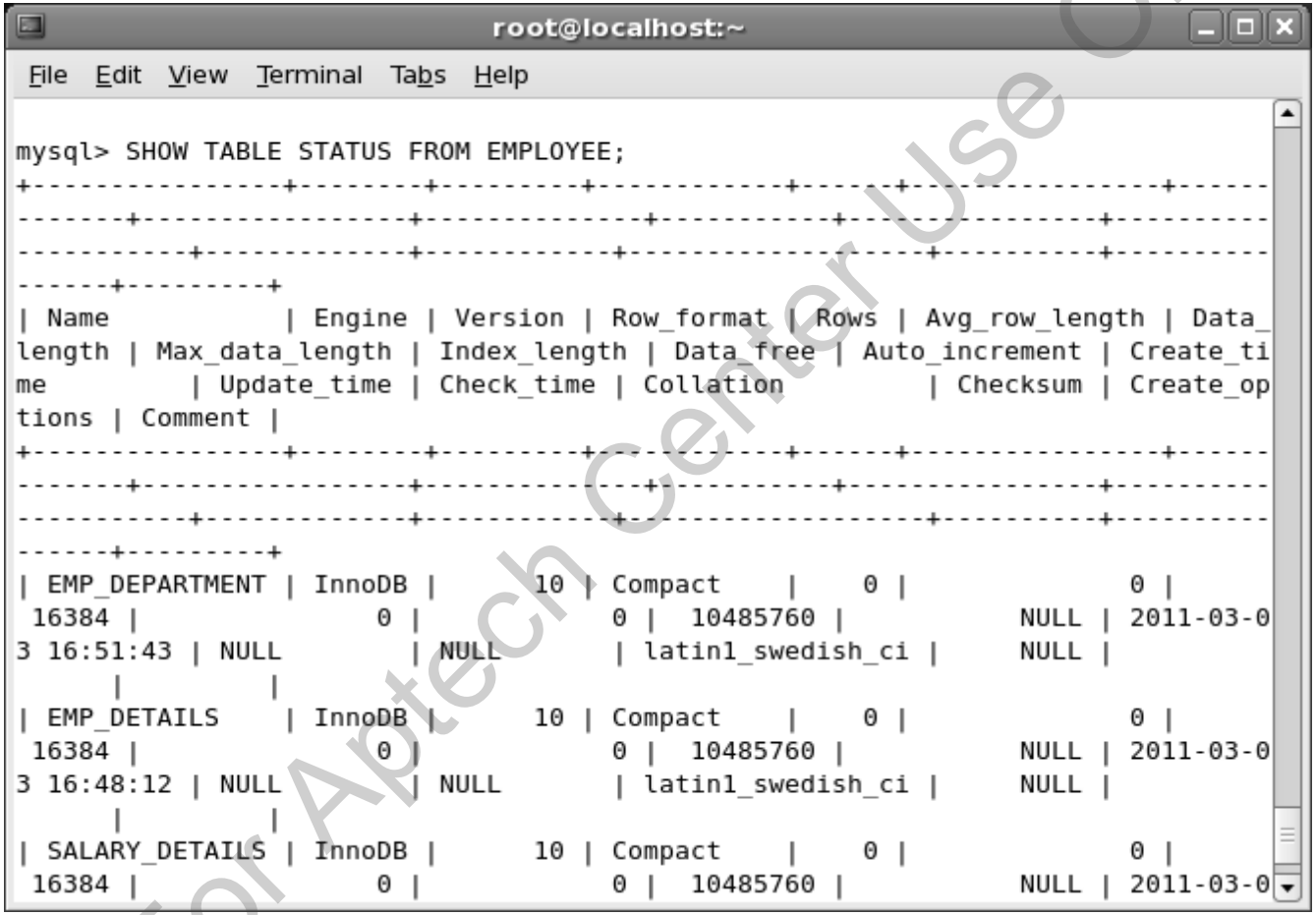
Table lists the information returned by the `SHOW TABLE STATUS` command

Column	Description
Name	Displays the name of the table
Type	Displays type of table
Row_format	Displays storage format of the row (such as fixed, dynamic, or compressed)
Rows	Displays number of rows
Avg_row_length	Displays average row length
Data_length	Displays length of the data file
Max_data_length	Displays max length of the data file
Index_length	Displays length of the index file
Data_free	Displays number of unused allocated bytes
Auto_increment	Displays next auto increment value
Create_time	Displays when the table was created
Update_time	Displays the date of last updation of data file
Check_time	Displays when the table was last checked
Comment	Displays the comments for a column. The comment can be 255 characters long

- For example, to view the status of all the tables of the EMPLOYEE database, enter the following command at the command prompt:

```
SHOW TABLE STATUS FROM EMPLOYEE;
```

Figure displays the output of the command



Viewing Information about Databases on the File System 37-39

- To view a list of grants that are assigned for a user, use the following syntax:

```
SHOW GRANTS FOR user;
```

where,

`SHOW` – displays information specified in the clause

`GRANTS` – displays privileges or account rights

`FOR` – specifies the object to display the privileges

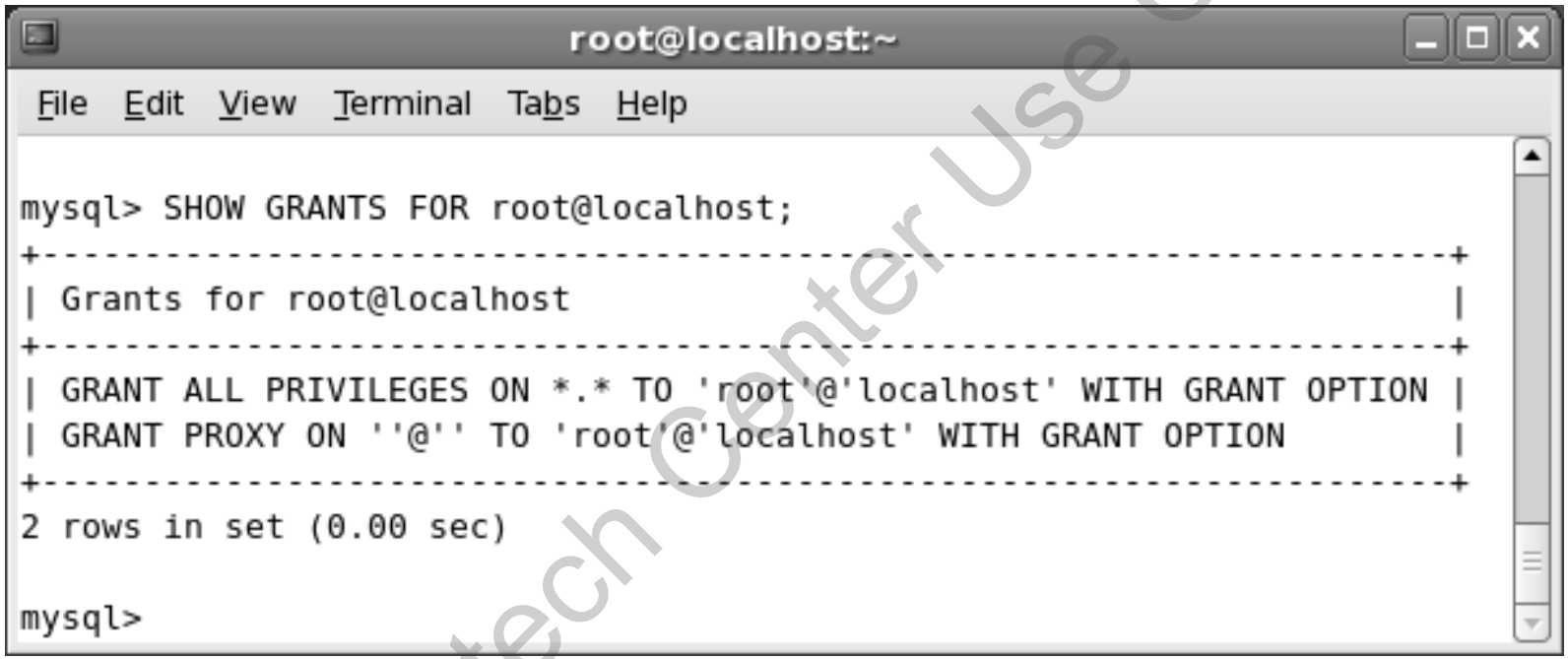
`user` – specifies the type of object for which privileges are to be displayed

Viewing Information about Databases on the File System 38-39

- For example, to display the rights granted to the root user, enter the following command at the command prompt:

```
SHOW GRANTS FOR root@localhost;
```

Figure displays the output of the command



Altering Database on the File System

- Alteration of a database involves making changes to the database
- MySQL provides the `ALTER DATABASE` command to modify the global characteristics or attributes of a database stored in the `db.opt` file of the database directory
- You will use the `CHARACTER SET` clause to modify the default database character set
- You will use the `COLLATE` clause to modify the default database collation
- A collation in MySQL database is a set of rules used in data comparisons

- The syntax for modifying the character set of a database is:

```
ALTER DATABASE database_name DEFAULT CHARACTER SET
charset_name;
```

where,

`ALTER DATABASE` – edits the database

`database_name` – specifies the name of the database on which you need to make changes

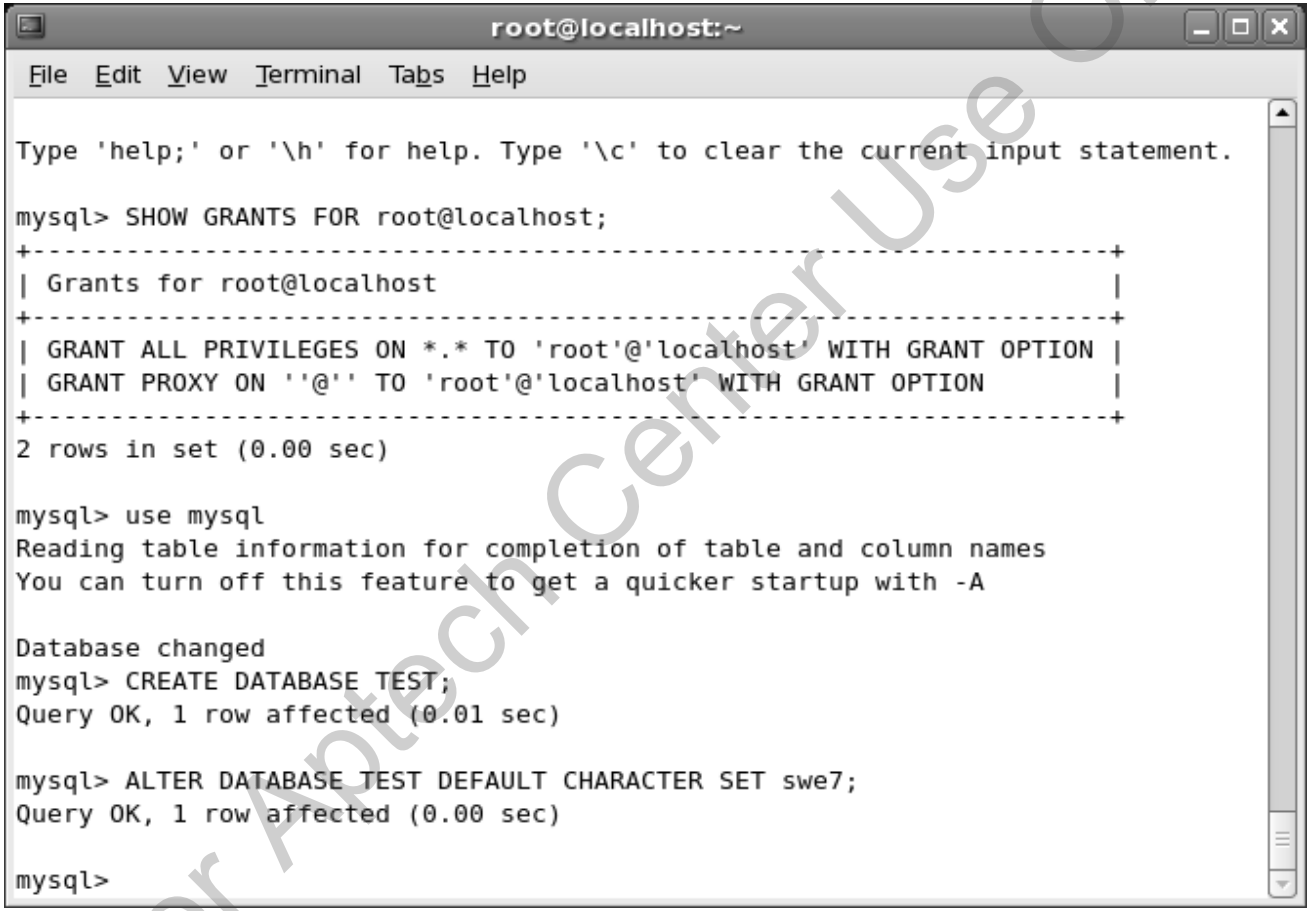
`DEFAULT CHARACTER SET` – specifies the default character set for the database

`charset_name` – specifies the type of character set for the database

- For example, to modify the character set of the database, create a temporary database named TEST and then alter the database by entering the following command at the command prompt:

```
ALTER DATABASE TEST DEFAULT CHARACTER SET swe7;
```

Figure displays the output of the command



- The syntax to modify the collation of a database is:

```
ALTER DATABASE database_name DEFAULT COLLATE collation_name;
```

- A table contains structured data in the form of rows and columns
- You cannot view the contents of a table if it does not contain data
- MySQL provides the `INSERT` command to add data to the table
- The syntax for `INSERT` command is:

```
INSERT INTO table_name {VALUES | VALUE} (value1, value2…);
```

where,

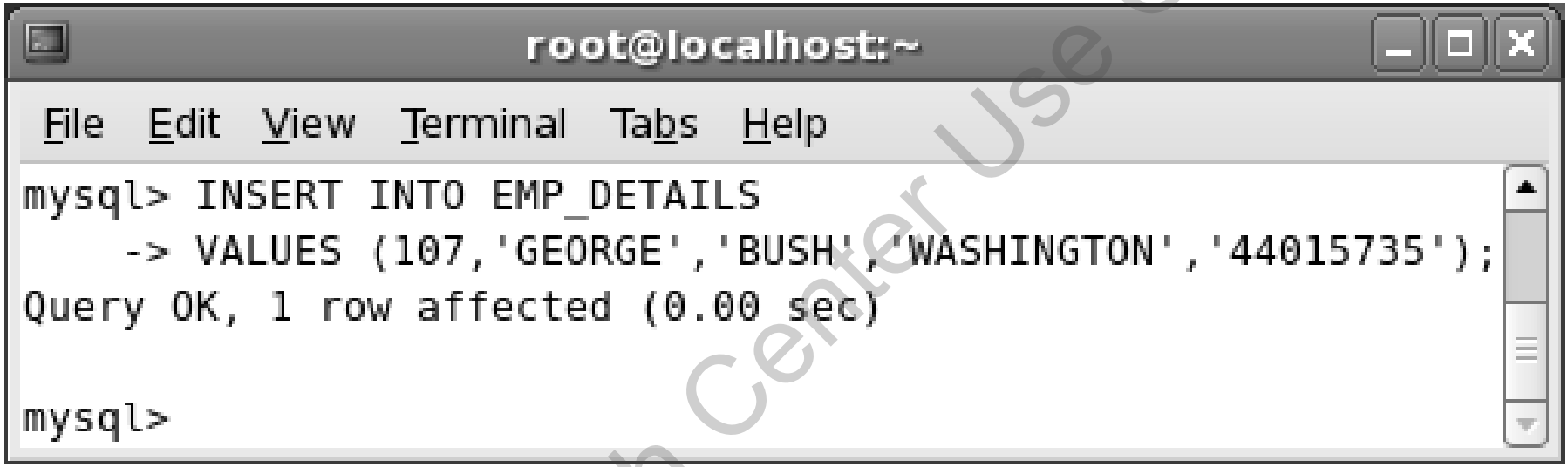
`INSERT INTO` – adds a new record to the table

`table_name` – specifies the name of the table to add the record

`{VALUES | VALUE}` – any one of these two may be used to specify the values and either may be used for a single values list or multiple lists

`value1` – specifies the data that will be added to the column

Figure displays the insertion of records in the table



- You can use the `SELECT` command to retrieve data from one or more tables. The `SELECT` command has two clauses:
  - `FROM` - specifies the table name whose records are to be retrieved
  - `WHERE` - specifies the condition, based on which the records are retrieved. This clause is optional
- The syntax for retrieving all the records of a table is:

```
SELECT [*] FROM table_name;
```

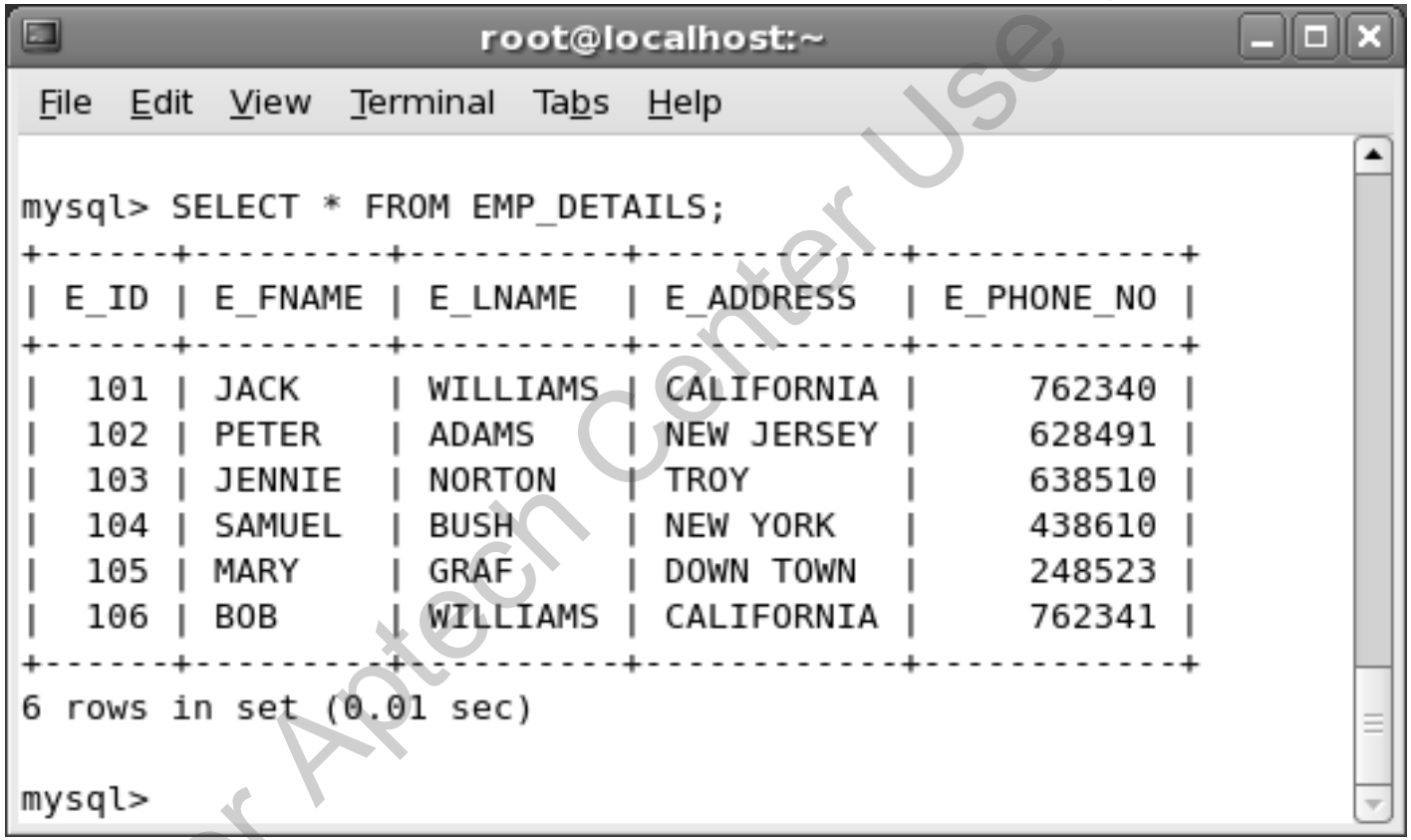
where,

'*' displays all the columns of the specified table

- For example, to view all records of the `EMP_DETAILS` table, enter the following command at the command prompt:

```
SELECT * FROM EMP_DETAILS;
```

Figure displays the output of the command



- To display unique information from columns using `DISTINCT` keyword with `SELECT` command, use the syntax:

```
SELECT DISTINCT column_name FROM table_name;
```

where,

`SELECT` – displays data from the table

`DISTINCT` – is the keyword that displays unique information from columns

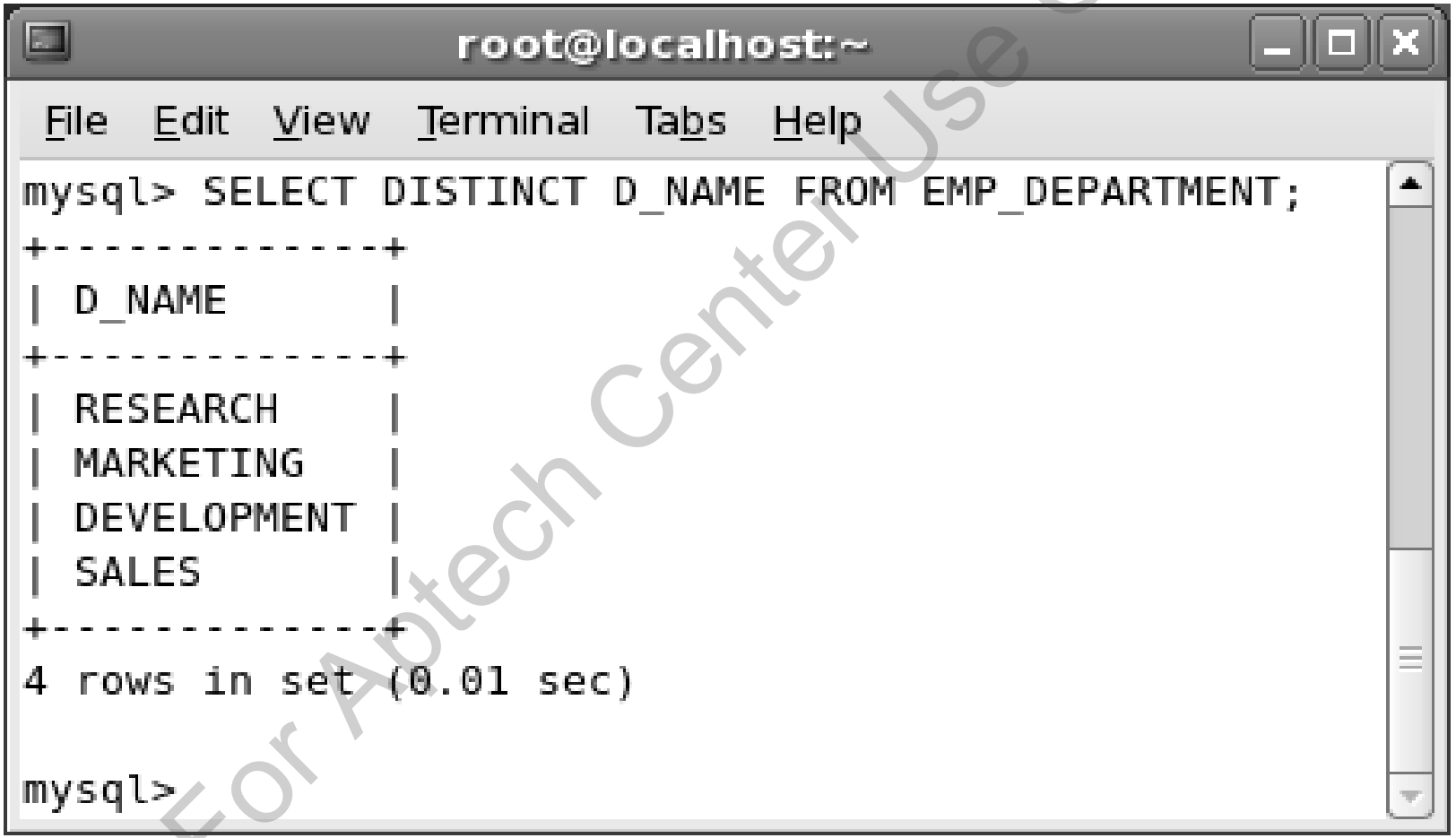
`column_name` – specifies the name of the column

`table_name` – specifies the name of the table where the column exists

- This command will display the unique records from the columns specified in the `column_name` keyword
- For example, to display only the unique department names from the `EMP_DEPARTMENT` table, enter the following command at the command prompt:

```
SELECT DISTINCT D_NAME FROM EMP_DEPARTMENT;
```

Figure displays the output of the command



- The `SELECT` command also enables you to view specific columns of a table
- The syntax for viewing only specific columns of a table is:

```
SELECT column_name1, column_name2 FROM
table_name;
```

where,

`SELECT` – displays the information

`column_name1` – specifies the name of the column to be retrieved

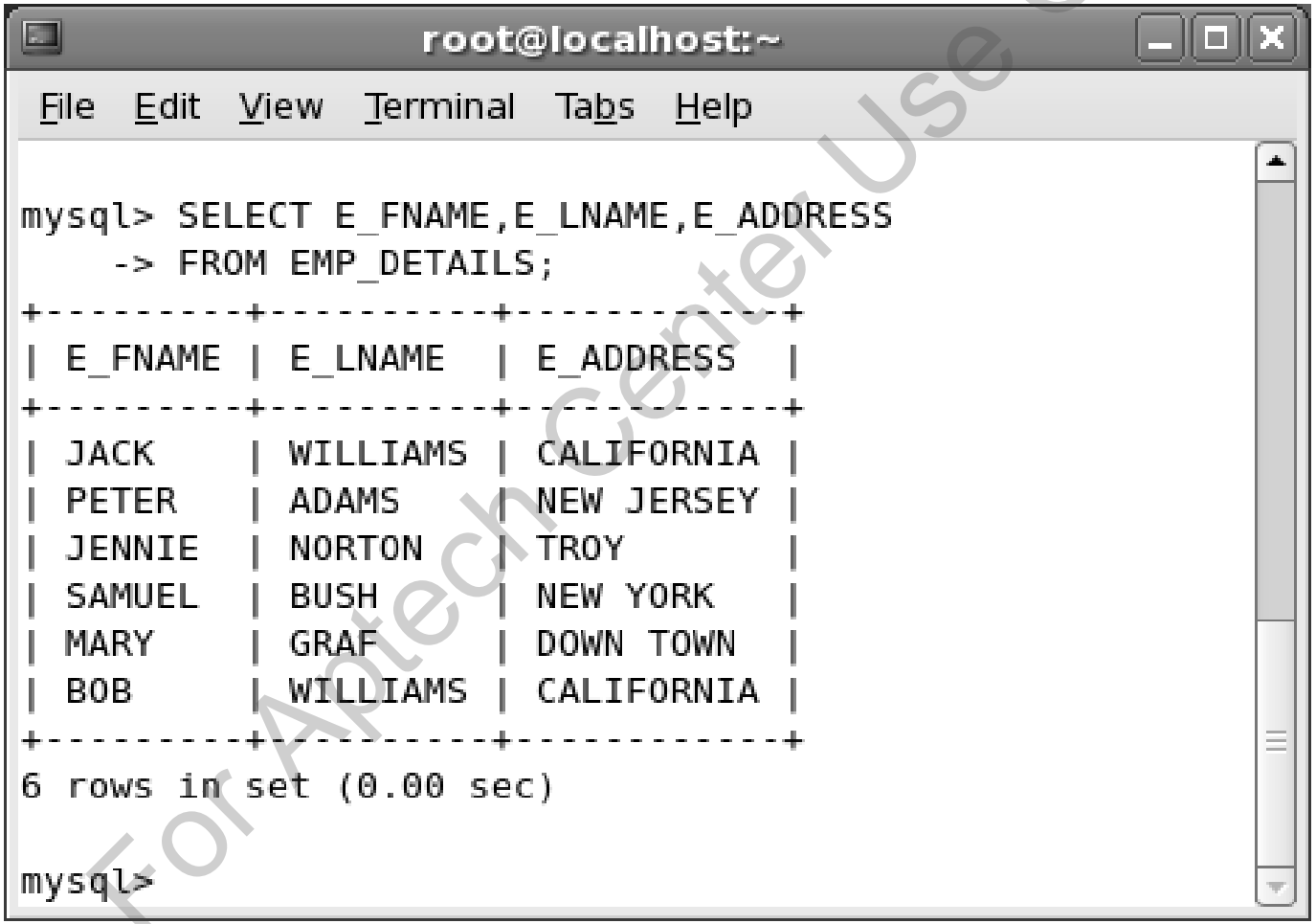
`column_name2` – specifies the name of the column to be retrieved

`table_name` – specifies the name of the table that contains the columns

- This command will display only those columns that are specified in the `column_name` clause of the command
- For example, to view the `E_FNAME`, `E_LNAME`, and `E_ADDRESS` columns of the `EMP_DETAILS` table, enter the following command at the command prompt:

```
SELECT E_FNAME, E_LNAME, E_ADDRESS FROM
EMP_DETAILS;
```

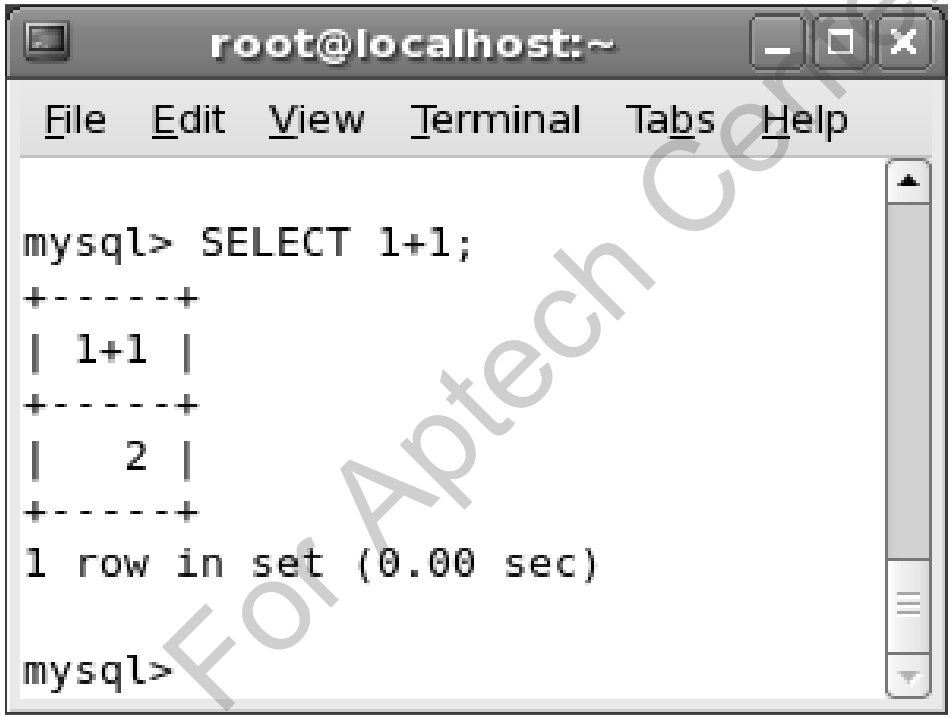
Figure displays the output of the command



- You can also use the `SELECT` command to make calculations

```
SELECT 1+1;
```

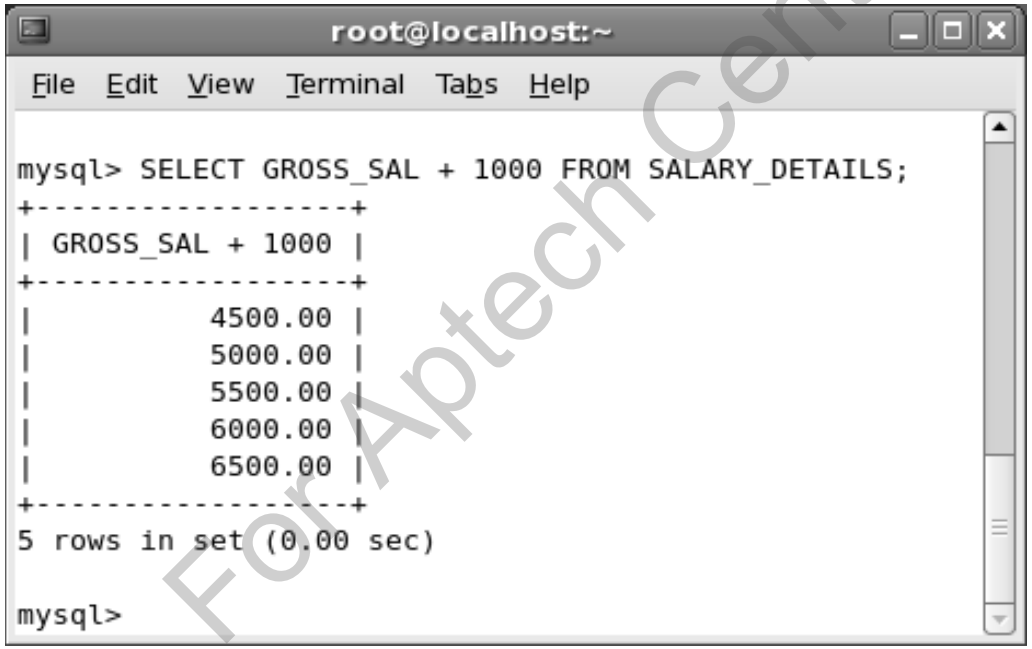
Figure displays the output of the command



- To perform arithmetic operations using the SELECT command, consider the following example:

```
SELECT GROSS_SAL + 1000 FROM SALARY_DETAILS;
```

Figure displays the output of the command



SELECT command Using WHERE Clause

- The WHERE clause contains one or more conditions that must be satisfied before the query retrieves a row
- The WHERE clause uses logical and conditional operators in the query
- The syntax to specify a condition using the WHERE clause is:

```
SELECT * FROM table_name WHERE <condition to satisfy>;
```

where,

SELECT – retrieves the data

* – specifies to retrieve all the columns of the table

table_name – specifies the name of the table

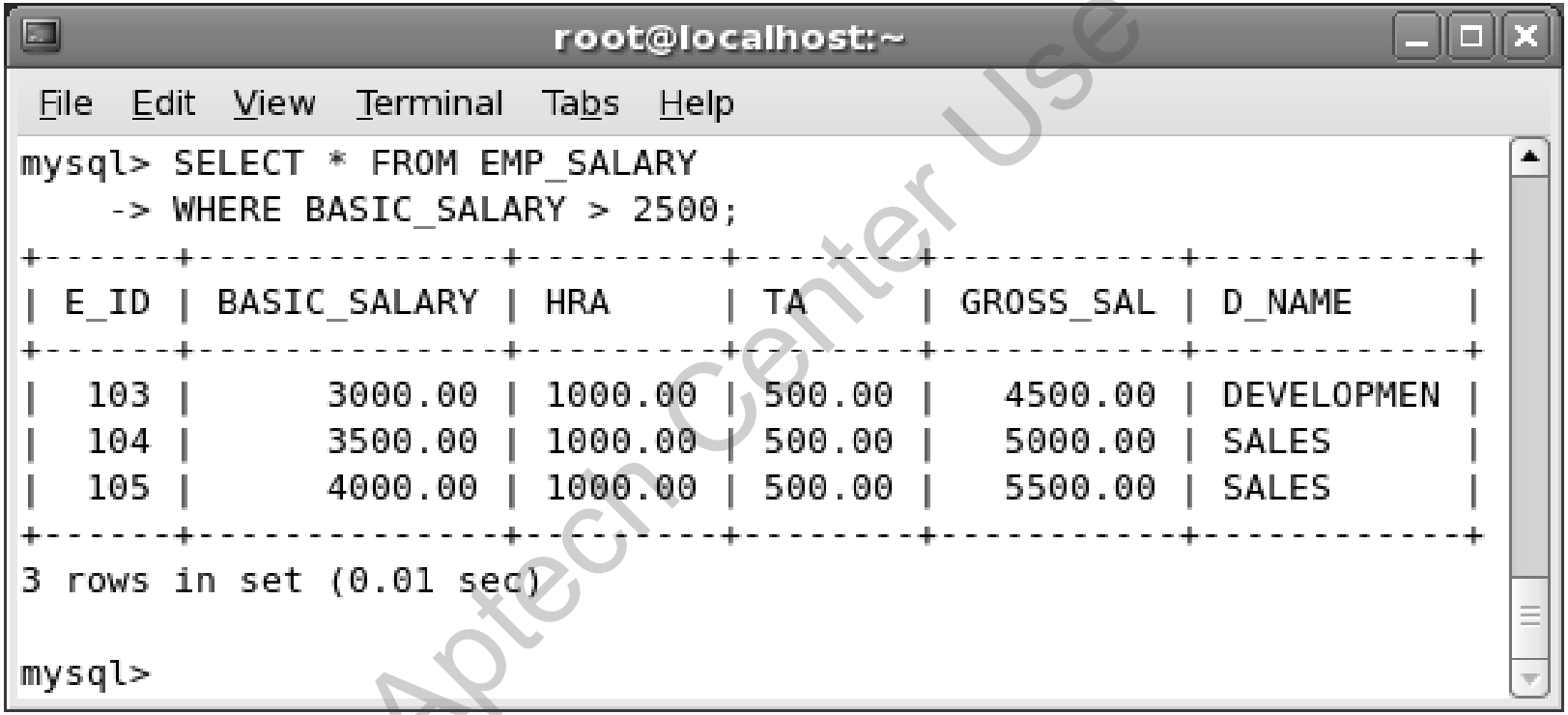
WHERE – specifies the clause to filter data

<condition to satisfy> – contains conditions to satisfy before retrieving and displaying data

- For example, to view the records of EMP_SALARY table, where the basic salary is greater than 2500, enter the following command at the command prompt:

```
SELECT * FROM EMP_SALARY WHERE BASIC_SALARY > 2500;
```

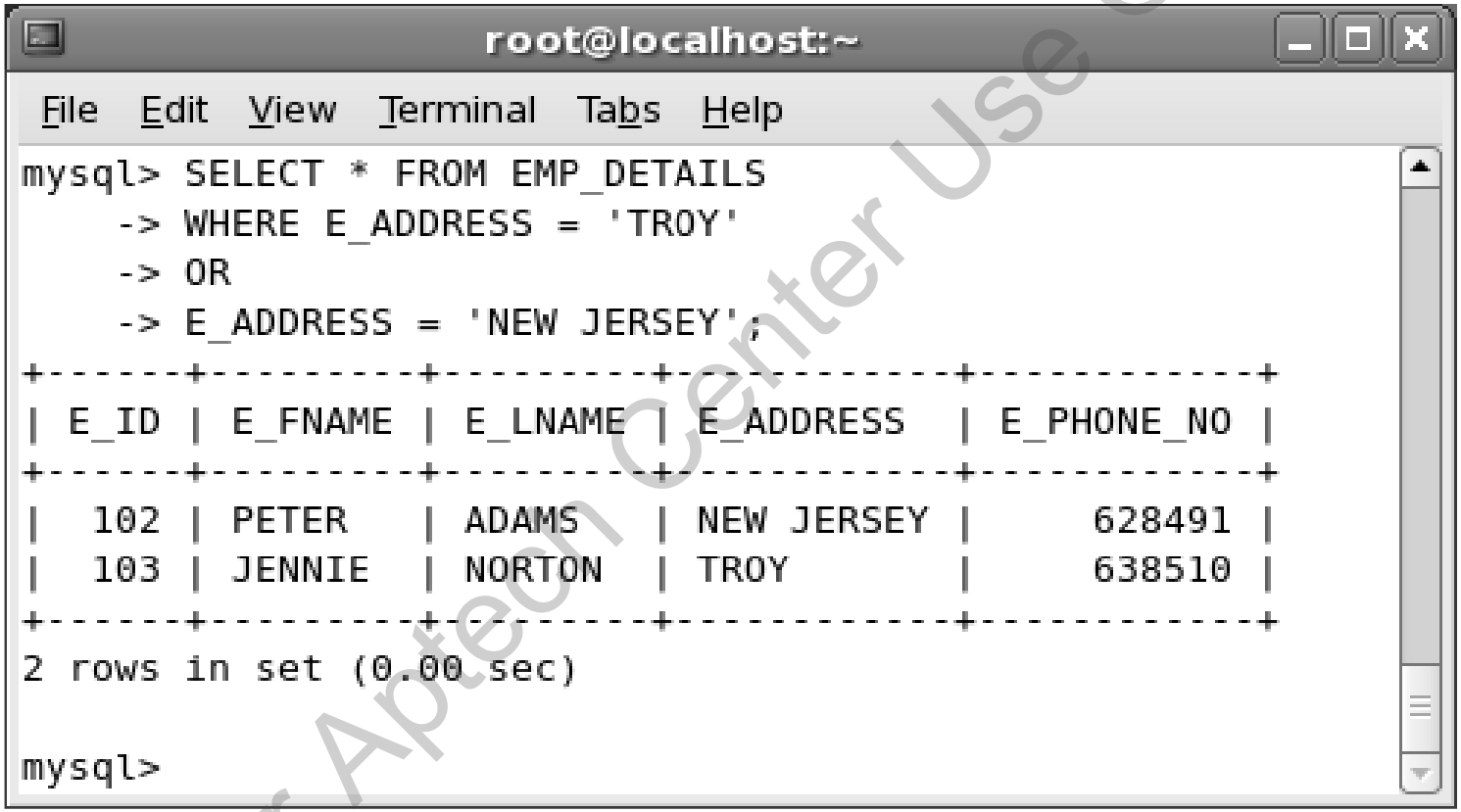
Figure displays the output of the command



- To view all the fields of the EMP_DETAILS table, where the address of the employee is TROY or the address of employee is NEW JERSEY, enter the following command at the command prompt:

```
SELECT * FROM EMP_DETAILS WHERE E_ADDRESS =
'TROY' OR E_ADDRESS = 'NEW JERSEY';
```

Figure displays the output of the command



Modifying Table Definitions Using ALTER Command

- MySQL provides the `ALTER TABLE` command to modify the structure of a table
- You can add or delete columns, rename columns or the table, create or destroy indexes, and modify the column type
- The `ALTER TABLE` command creates a temporary copy of the original table, on which the alteration is performed
- MySQL uses this feature as a security measure to prevent data loss in case the table modification fails

- The syntax for altering a table is:

```
ALTER [IGNORE] TABLE table_name alter_spec [,
alter_spec...];
```

where,

`ALTER` – specifies to edit the object

`IGNORE` – checks for duplicate records on the index keys in the new table

`TABLE` – specifies the type of object to edit

`table_name` – specifies the name of the table

`alter_spec` – contains modification information

- The syntax to add a column in the existing table is:

```
ALTER TABLE table_name ADD [COLUMN] create_definition
[FIRST | AFTER column_name];
```

where,

`table_name` – specifies the name of the table to modify

`ADD [COLUMN]` – appends a new column to the table

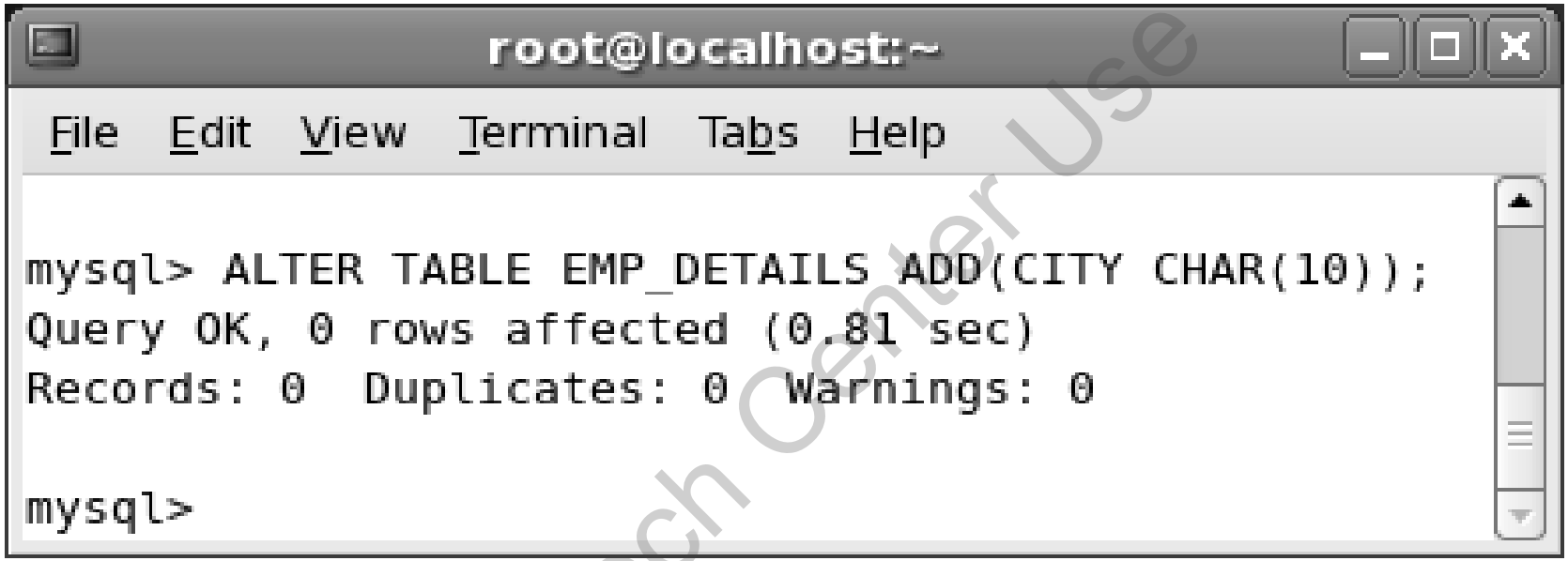
`create_definition` – defines the column type in a table creation

`FIRST | AFTER column_name` – specifies the location of the new column in the table

- For example, to add a column `CITY` to the `EMP_DETAILS` table for entering the name of the city of employees, enter the following command at the command prompt:

```
ALTER TABLE EMP_DETAILS ADD (CITY CHAR (10));
```

Figure displays the output of the command



- The syntax to add an index key to a column of a table is:

```
ALTER TABLE table_name ADD INDEX [index_name]
(index_column_name...);
```

where,

`ALTER TABLE` – specifies to modify the table

`table_name` – specifies the name of the table

`ADD INDEX` – appends an index to the table

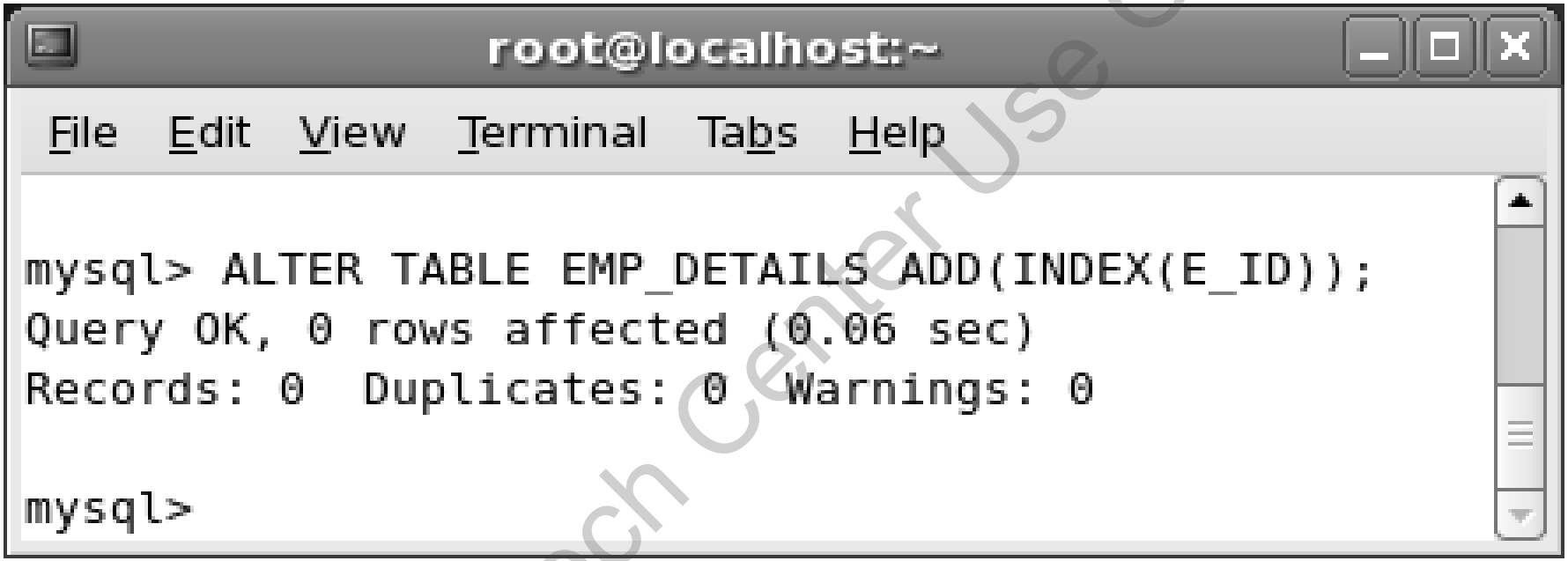
`index_name` – specifies a name for the index

`Index_column_name` – specifies the column in the table to index

- This command adds an index to the table and indexes the specified column
- For example, to add an index on column `E_ID` on `EMP_DETAILS` table, enter the following command at the command prompt:

```
ALTER TABLE EMP_DETAILS ADD (INDEX (E_ID));
```

Figure displays the output of the command



- The syntax to add a primary key to the column of a table is:

```
ALTER TABLE table_name ADD PRIMARY KEY
(index_column_name...);
```

where,

`ALTER TABLE` – modifies the table structure

`table_name` – specifies the name of the table

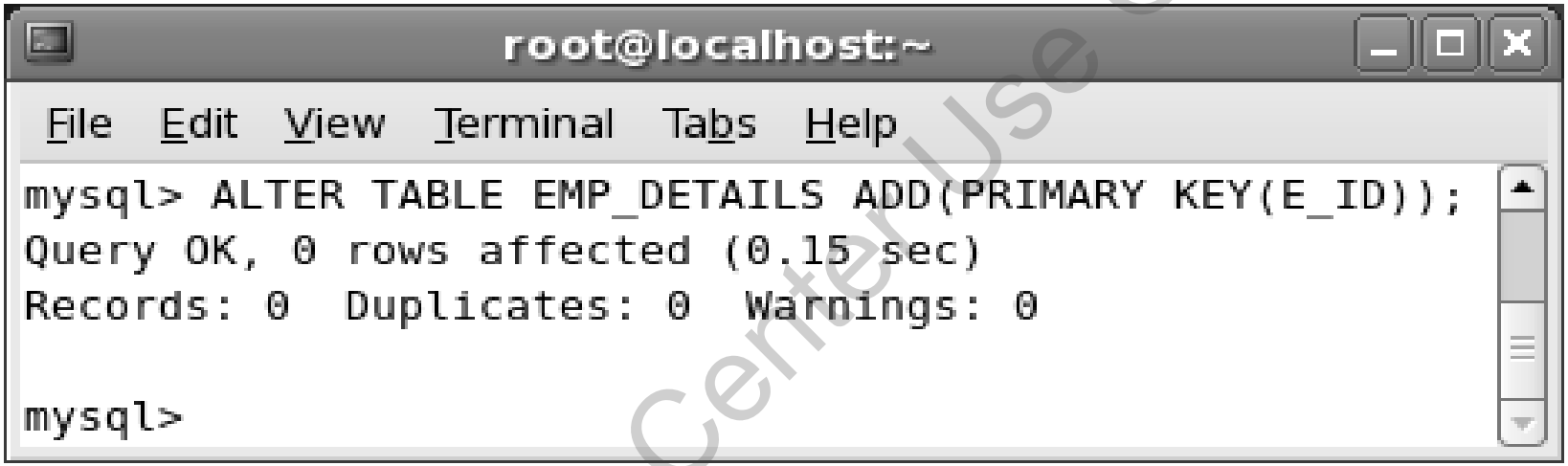
`ADD PRIMARY KEY` – appends a primary key to the table

`index_column_name` – specifies the name of the column to use as an index for the primary key

- This command adds a primary key to the specified indexed column in the table
- For example, to add a primary key on column `E_ID` on `EMP_DETAILS` table, enter the following command at the command prompt:

```
ALTER TABLE EMP_DETAILS ADD (PRIMARY KEY
(E_ID));
```

Figure displays the output of the command



- To modify a column definition, use the following syntax:

```
ALTER TABLE table_name MODIFY [COLUMN] create_definition;
```

where,

`ALTER TABLE` – specifies to edit the table

`table_name` – specifies the name of the table

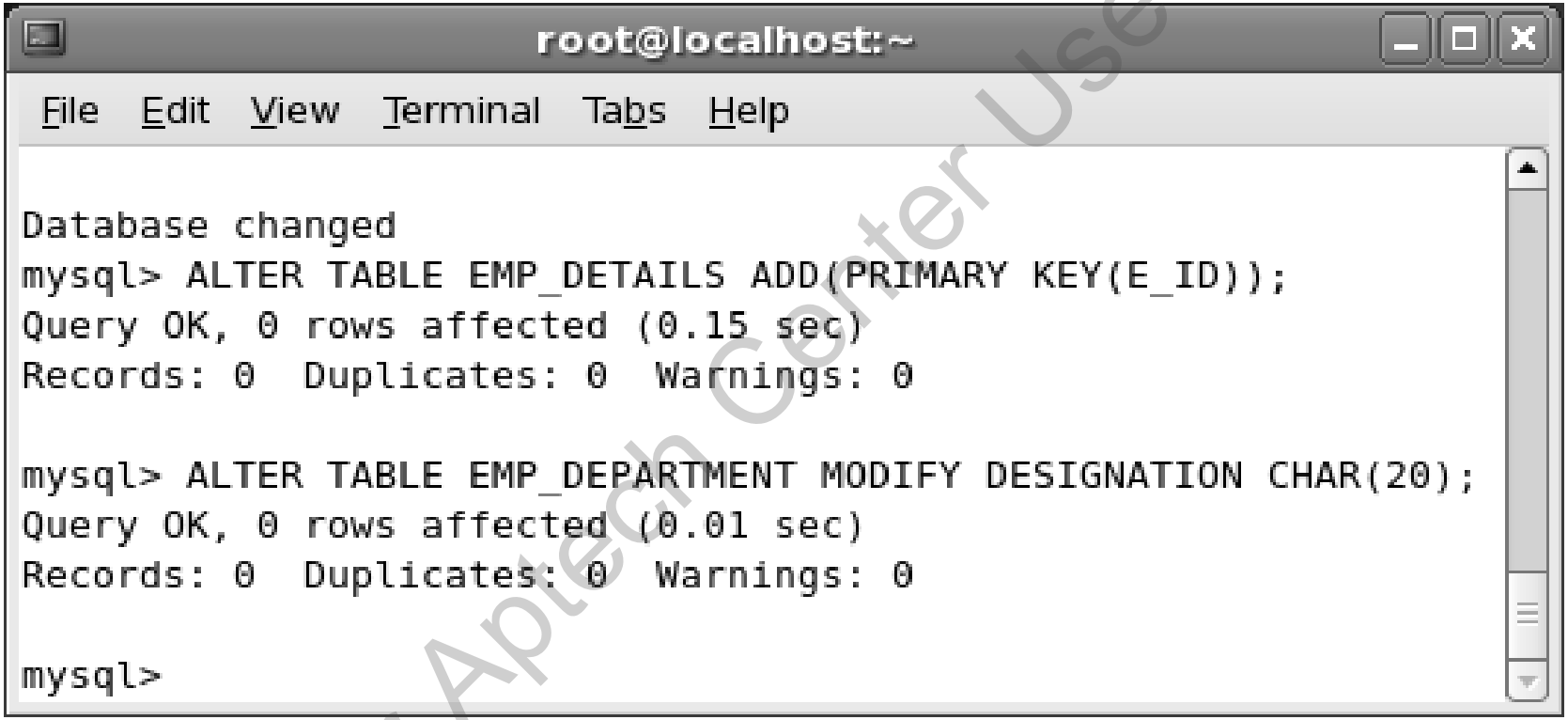
`MODIFY [COLUMN]` – specifies to change the column structure

`create_definition` – specifies the new rules for the column

- For example, to modify DESIGNATION column from CHAR(50) to CHAR(20) data type of EMP_DEPARTMENT table, enter the following command at the command prompt:

```
ALTER TABLE EMP_DEPARTMENT MODIFY DESIGNATION
CHAR(20);
```

Figure displays the output of the command



- To drop a column from a table, use the following syntax:

```
ALTER table_name DROP [COLUMN] column_name;
```

where,

`ALTER TABLE` – modifies the table structure

`table_name` – specifies the name of the table to modify

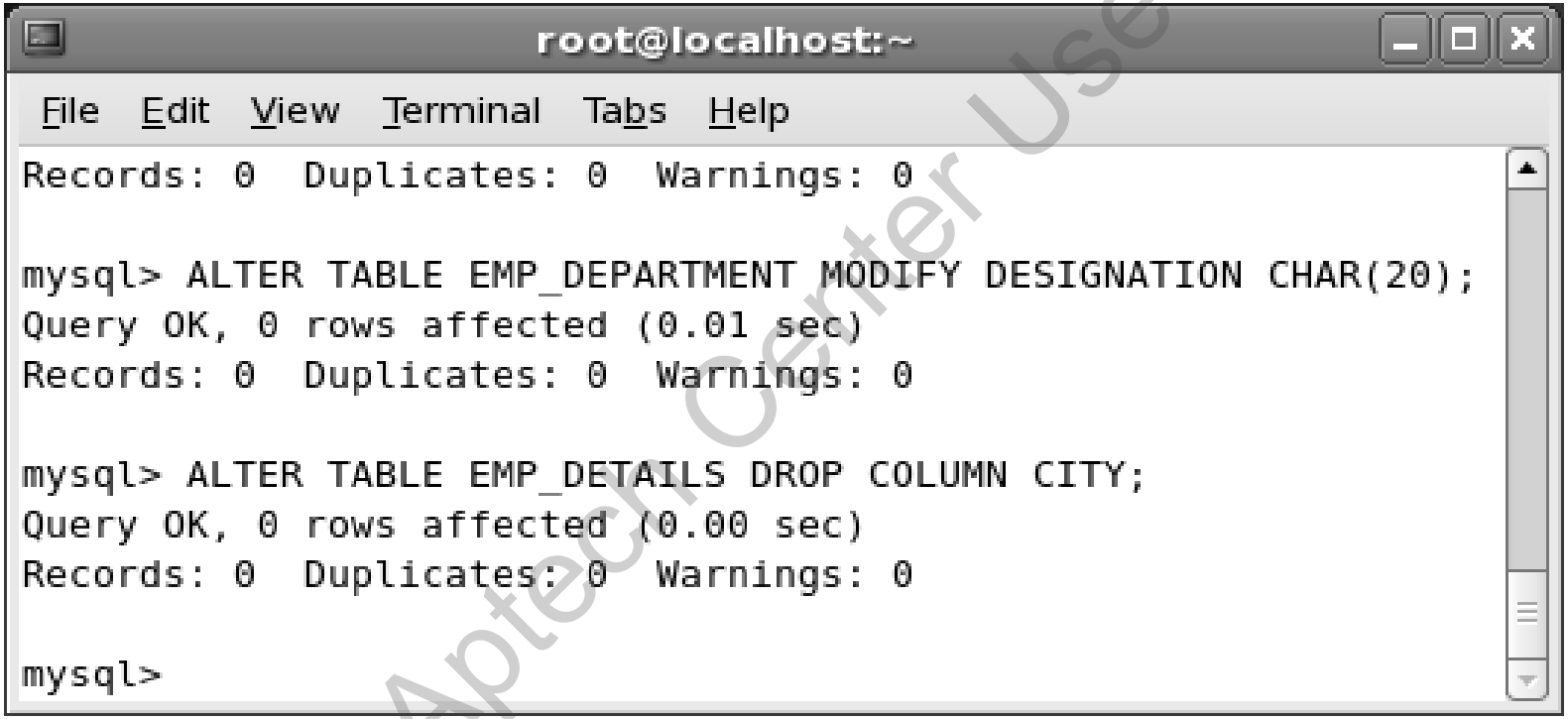
`DROP [COLUMN]` – removes the column from the table

`column_name` – specifies the name of the column to remove from the table

- This command removes a column from the table
- For example, to remove the column `CITY` from `EMP_DETAILS` table, enter the following command at the command prompt:

```
ALTER TABLE EMP_DETAILS DROP COLUMN CITY;
```

Figure displays the output of the command



- To drop a primary key of a column from the table, use the following syntax:

```
ALTER TABLE table_name DROP PRIMARY KEY;
```

where,

`ALTER TABLE` – edits the table structure

`table_name` – specifies the name of the table to modify

`DROP PRIMARY KEY` – removes the primary key from the table

- For example, to remove primary key constraint from `EMP_DETAILS` table, enter the following command at the command prompt:

```
ALTER TABLE EMP_DETAILS DROP PRIMARY KEY;
```

- To change the name of a table, use the following syntax:

```
ALTER TABLE table_name RENAME new_table_name;
```

where,

`ALTER TABLE` – edits the table structure

`table_name` – specifies the name of the table to modify

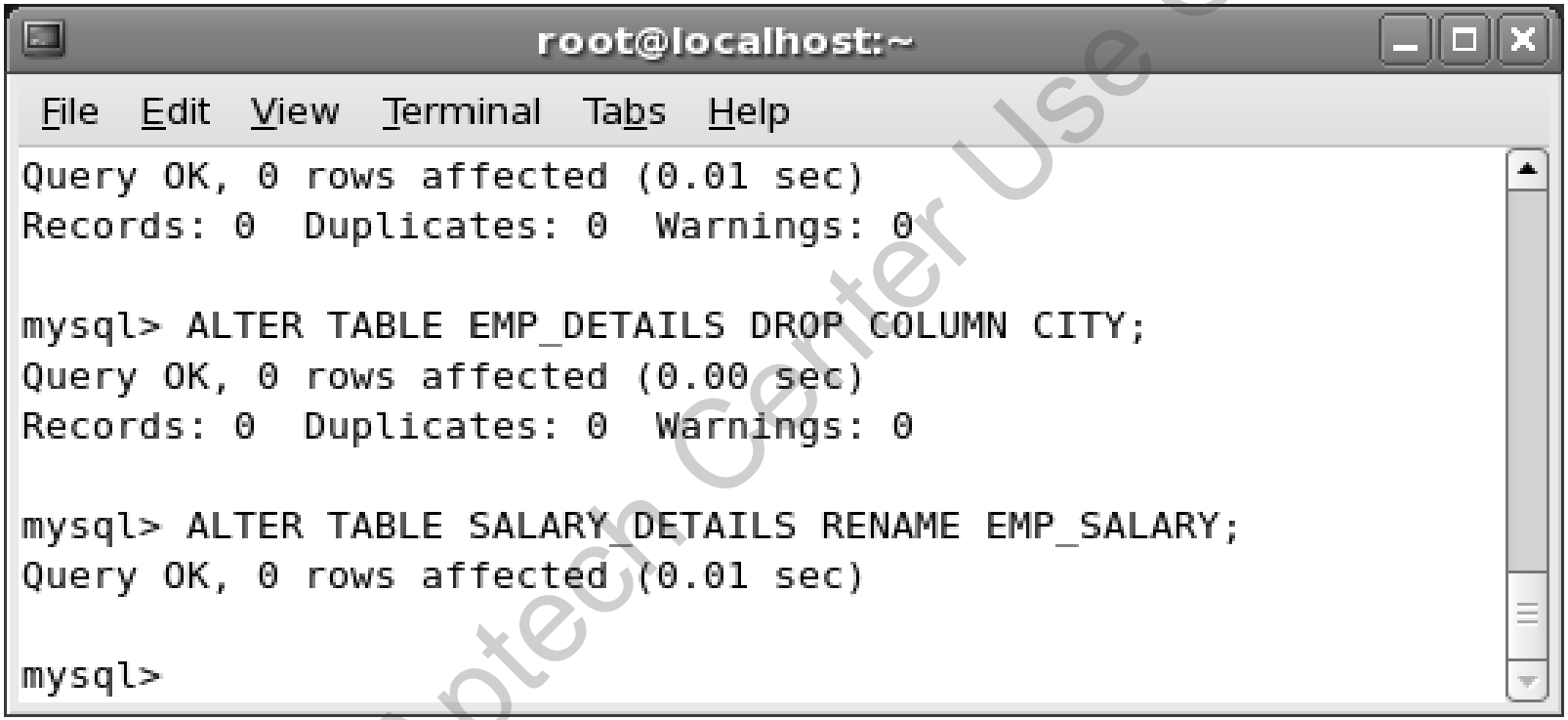
`RENAME` – changes the name of the table

`new_table_name` – specifies the new name for the table

- For example, to rename the table from SALARY_DETAILS to EMP_SALARY, enter the following command at the command prompt:

```
ALTER TABLE SALARY_DETAILS RENAME
EMP_SALARY;
```

Figure displays the output of the command



- Following are some of the points to be remembered while working with ALTER TABLE command:
  - MySQL deletes the index values of columns when you drop columns from a table. MySQL also removes the index when you drop or delete all indexed columns
  - If `DESCRIBE table_name` displays that the column specification, modified using the `ALTER TABLE` command, has not changed, then it is possible that the modification is ignored because the size of the column to be changed may be greater than or less than the required size length
  - A single ALTER TABLE command can contain several ADD, ALTER, DROP, and CHANGE clauses

Deleting Table Definitions Using DROP Command

- MySQL provides the `DROP TABLE` command to remove or delete tables from a database
- This command removes table definition, data, indexes, triggers, constraints, and permissions for that table

Deleting Table Definitions Using DROP Command

- The syntax for the `DROP` command is:

```
DROP TABLE [IF EXIST] table_name[table_name1,…][RESTRICT | CASCADE];
```

where,

`table_name` – specifies the name of the table to delete

`IF EXIST` - prevents error occurrence while executing the command if the table does not exist

`RESTRICT` - specifies that if any dependencies exist, MySQL will not delete the table, if dependencies exist

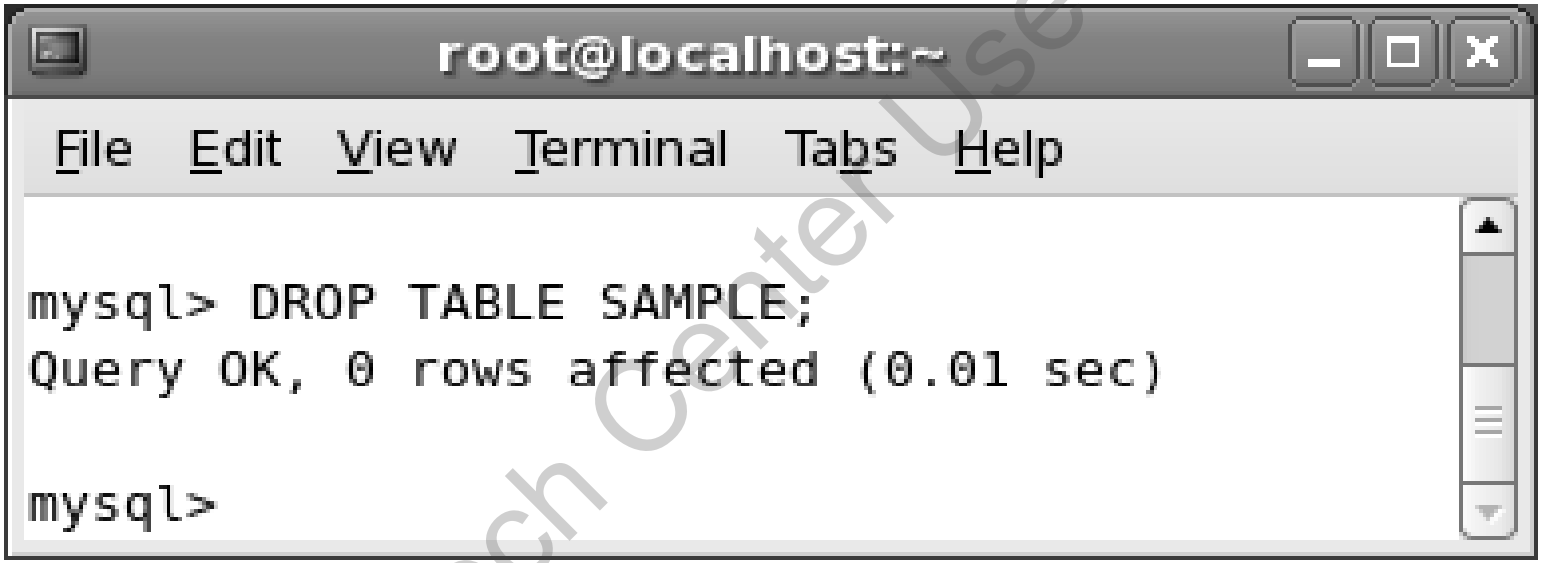
`CASCADE` - specifies to remove dependencies before deleting the table

Deleting Table Definitions Using DROP Command

- For example, to remove a table named SAMPLE from the EMPLOYEE database, enter the following command at the command prompt:

```
DROP TABLE SAMPLE;
```

Figure displays the output of the command



Summary

- MySQL provides the SHOW command to view the list of databases and tables on the server, displays the server status, and user account privileges
- The `ALTER` command enables modification of the characteristics or attributes of a table or a database
- The `CHARACTER SET` clause defines the default character database
- The default collation set of a of a database can be defined using the `COLLATE` clause
- The `SELECT` command is used to retrieve data from a table
- The FROM clause in the `SELECT` command specifies the table name whose records are to be retrieved

Summary

- The WHERE clause in the SELECT command specifies conditions for retrieving data from the table. MySQL displays only those records that satisfy the conditions in the WHERE clause
- Conditional and logical operators can be used with SELECT command to retrieve data after satisfying data retrieval conditions
- The DROP command removes or deletes a table from a database. This command removes all the data and the table definition from the database
- MySQL provides the IF EXIST option in the DROP command to check the existence of the table before you delete it