

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*

- ◆ 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:

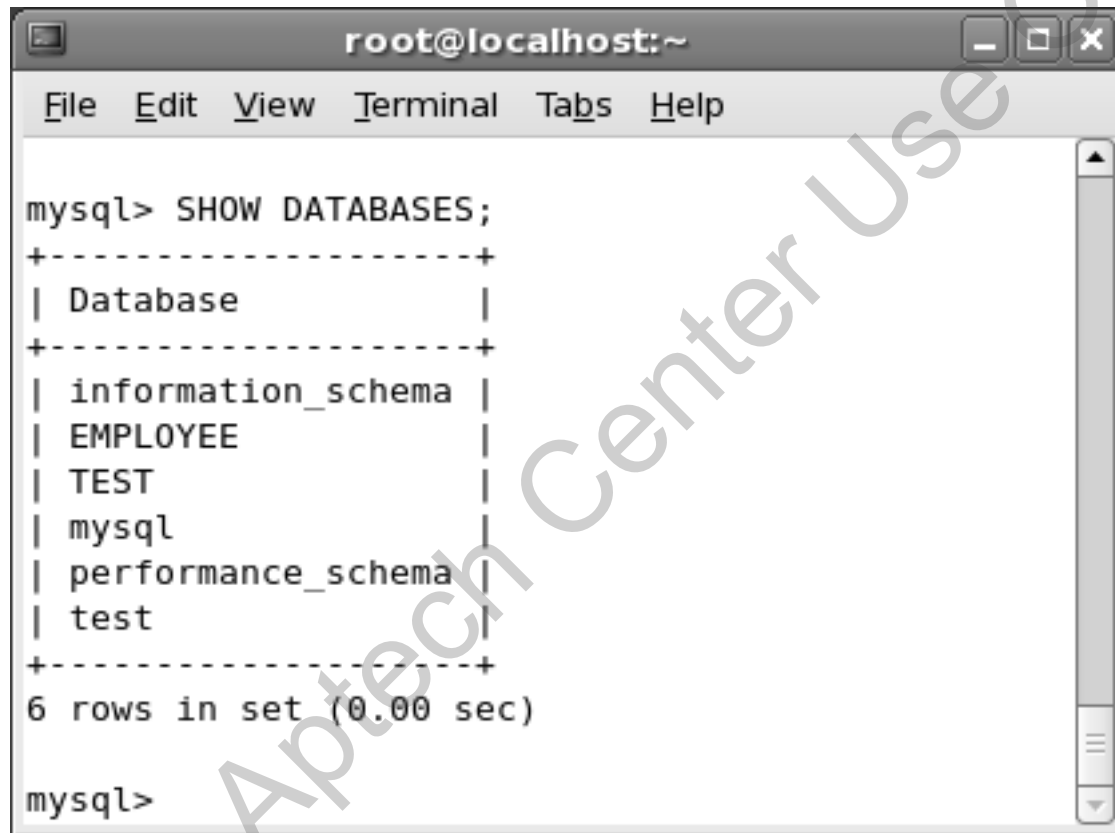
```
SHOW DATABASES;
```

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 screenshot shows a terminal window titled `root@localhost:~` with a menu bar containing `File`, `Edit`, `View`, `Terminal`, `Tabs`, and `Help`. The terminal content shows the execution of the `mysql> SHOW DATABASES;` command. The output is a table with one column, `Database`, listing the following databases: `information_schema`, `EMPLOYEE`, `TEST`, `mysql`, `performance_schema`, and `test`. Below the table, it states `6 rows in set (0.00 sec)`. The prompt `mysql>` is visible at the bottom of the terminal.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| EMPLOYEE          |
| TEST              |
| mysql              |
| performance_schema |
| test               |
+-----+
6 rows in set (0.00 sec)

mysql>
```

- ◆ 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

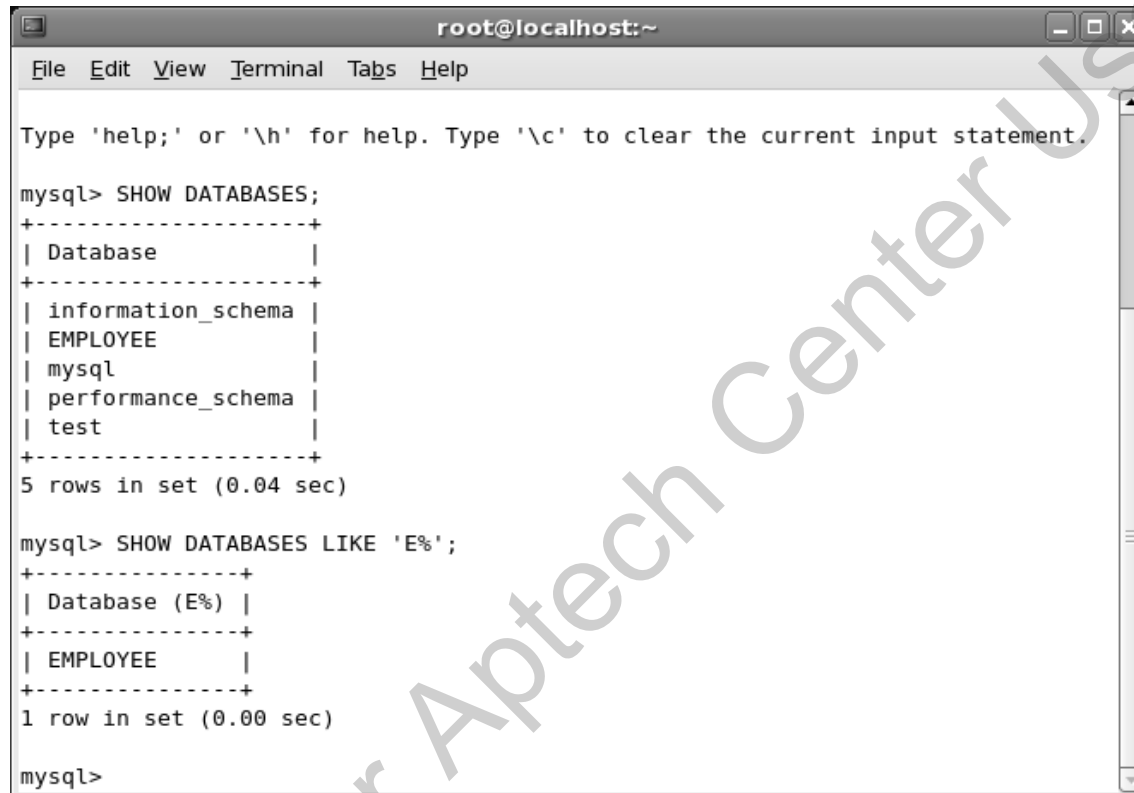
`LIKE <condition>` – contains conditions that must be satisfied before displaying the list of databases

- ◆ 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



The screenshot shows a terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal displays the output of two MySQL commands. The first command, 'SHOW DATABASES;', lists five databases: information_schema, EMPLOYEE, mysql, performance_schema, and test. The second command, 'SHOW DATABASES LIKE 'E%';', lists one database: EMPLOYEE. A large, diagonal watermark 'For Apteck Center Use Only' is overlaid on the terminal output.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| EMPLOYEE |  
| mysql |  
| performance_schema |  
| test |  
+-----+  
5 rows in set (0.04 sec)  
  
mysql> SHOW DATABASES LIKE 'E%';  
+-----+  
| Database (E%) |  
+-----+  
| EMPLOYEE |  
+-----+  
1 row in set (0.00 sec)  
  
mysql>
```

- ◆ 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

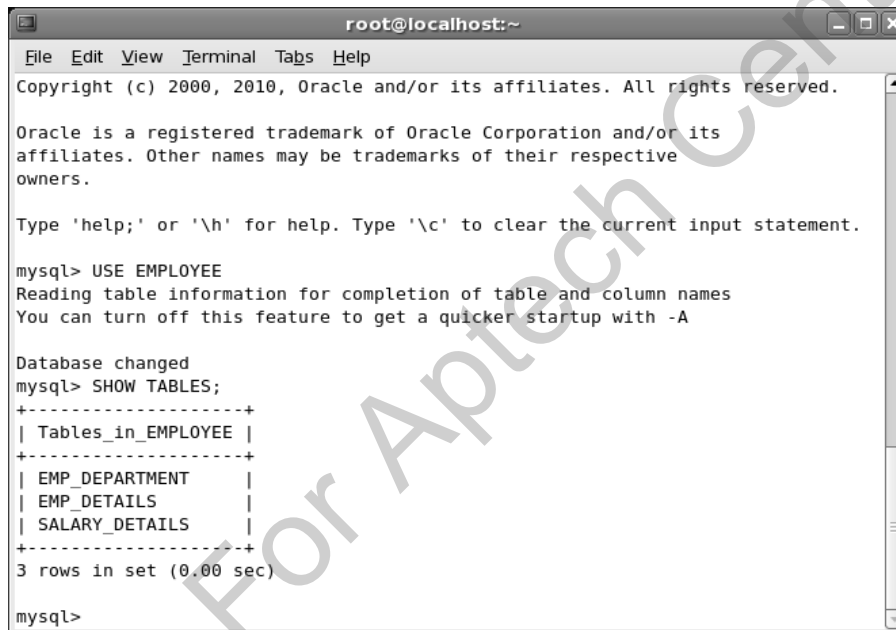
- ◆ The FROM clause enables you to specify the database name for which tables are to be listed
- ◆ You can use this keyword when you have not activated a database from the server with the USE command

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

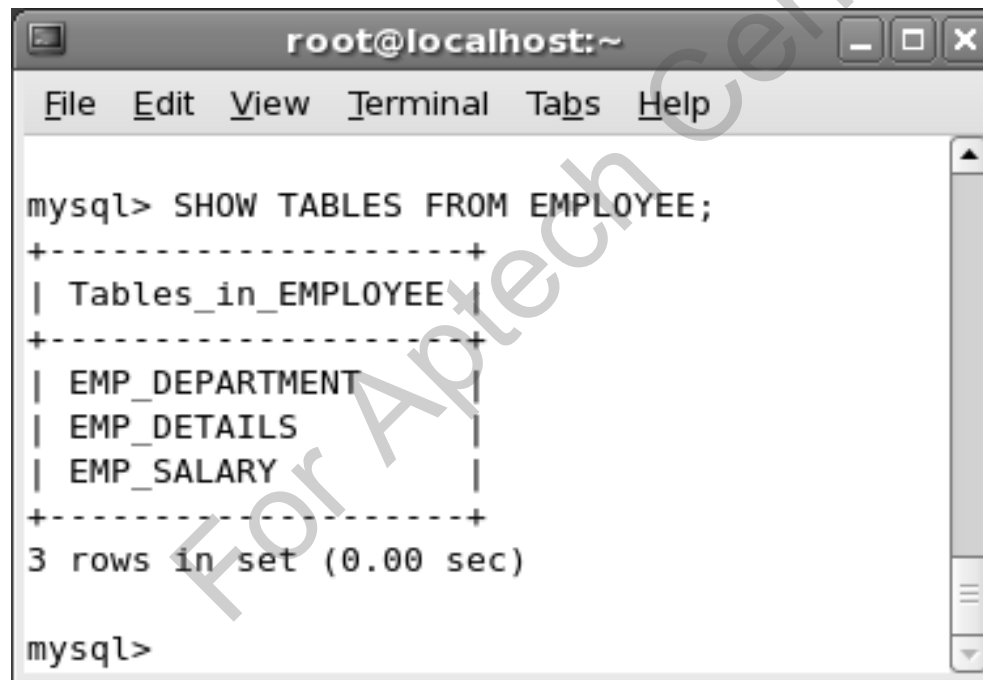


```
root@localhost:~  
File Edit View Terminal Tabs Help  
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> USE EMPLOYEE  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> SHOW TABLES;  
+-----+  
| Tables_in_EMPLOYEE |  
+-----+  
| EMP_DEPARTMENT      |  
| EMP_DETAILS          |  
| SALARY_DETAILS       |  
+-----+  
3 rows in set (0.00 sec)  
  
mysql>
```

- ◆ 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



The screenshot shows a terminal window titled 'root@localhost:~'. The menu bar includes 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The command prompt shows the command 'mysql> SHOW TABLES FROM EMPLOYEE;' being executed. The output is a table with one column, 'Tables_in_EMPLOYEE', containing three rows: 'EMP_DEPARTMENT', 'EMP_DETAILS', and 'EMP_SALARY'. Below the table, it says '3 rows in set (0.00 sec)'. The prompt 'mysql>' is visible at the bottom.

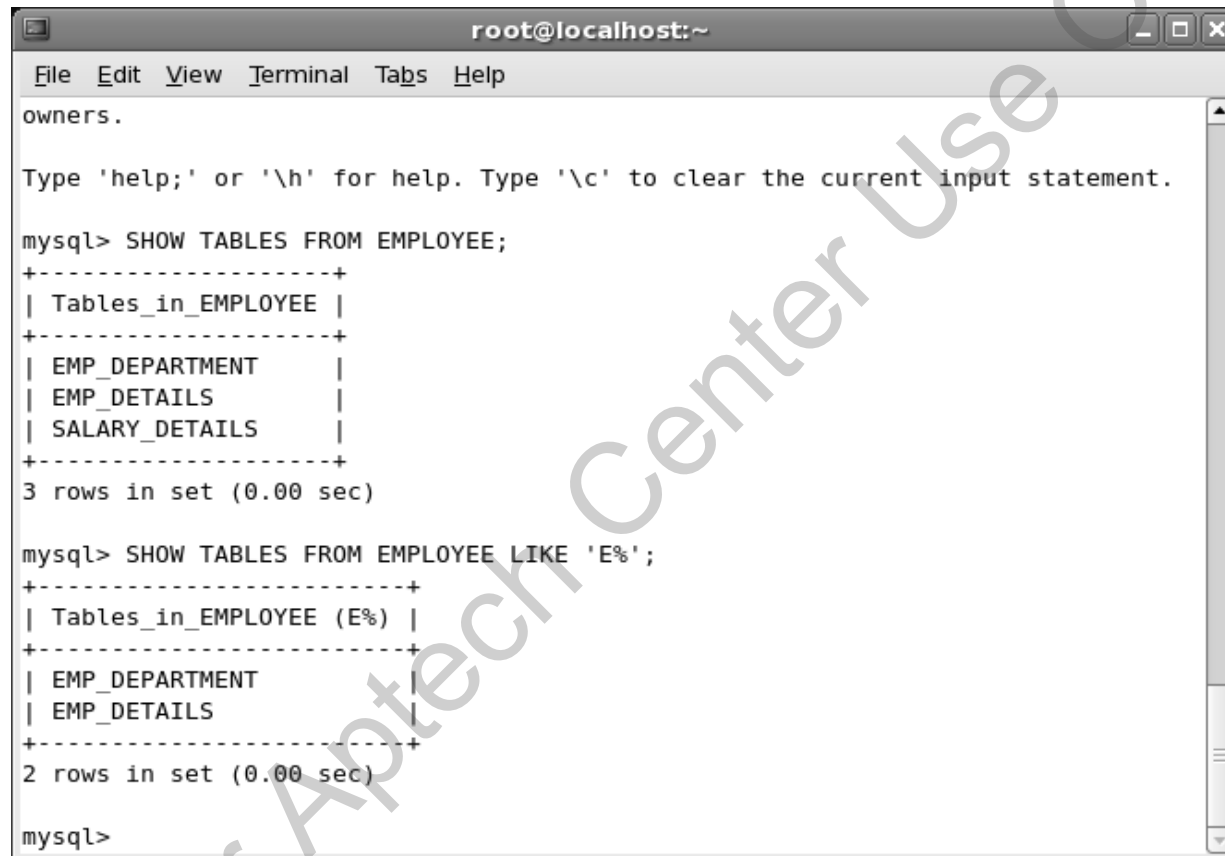
```
mysql> SHOW TABLES FROM EMPLOYEE;
+-----+
| Tables_in_EMPLOYEE |
+-----+
| EMP_DEPARTMENT     |
| EMP_DETAILS         |
| EMP_SALARY          |
+-----+
3 rows in set (0.00 sec)

mysql>
```

- ◆ 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



The screenshot shows a terminal window titled 'root@localhost:~'. The window contains the following text:

```
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> SHOW TABLES FROM EMPLOYEE;  
+-----+  
| Tables_in_EMPLOYEE |  
+-----+  
| EMP_DEPARTMENT      |  
| EMP_DETAILS          |  
| SALARY_DETAILS       |  
+-----+  
3 rows in set (0.00 sec)  
  
mysql> SHOW TABLES FROM EMPLOYEE LIKE 'E%';  
+-----+  
| Tables_in_EMPLOYEE (E%) |  
+-----+  
| EMP_DEPARTMENT          |  
| EMP_DETAILS              |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql>
```

A large, diagonal watermark reading 'For Aptech Center Use Only' is overlaid on the terminal output.

Viewing Information about Databases on the File System 13-39

- ◆ 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

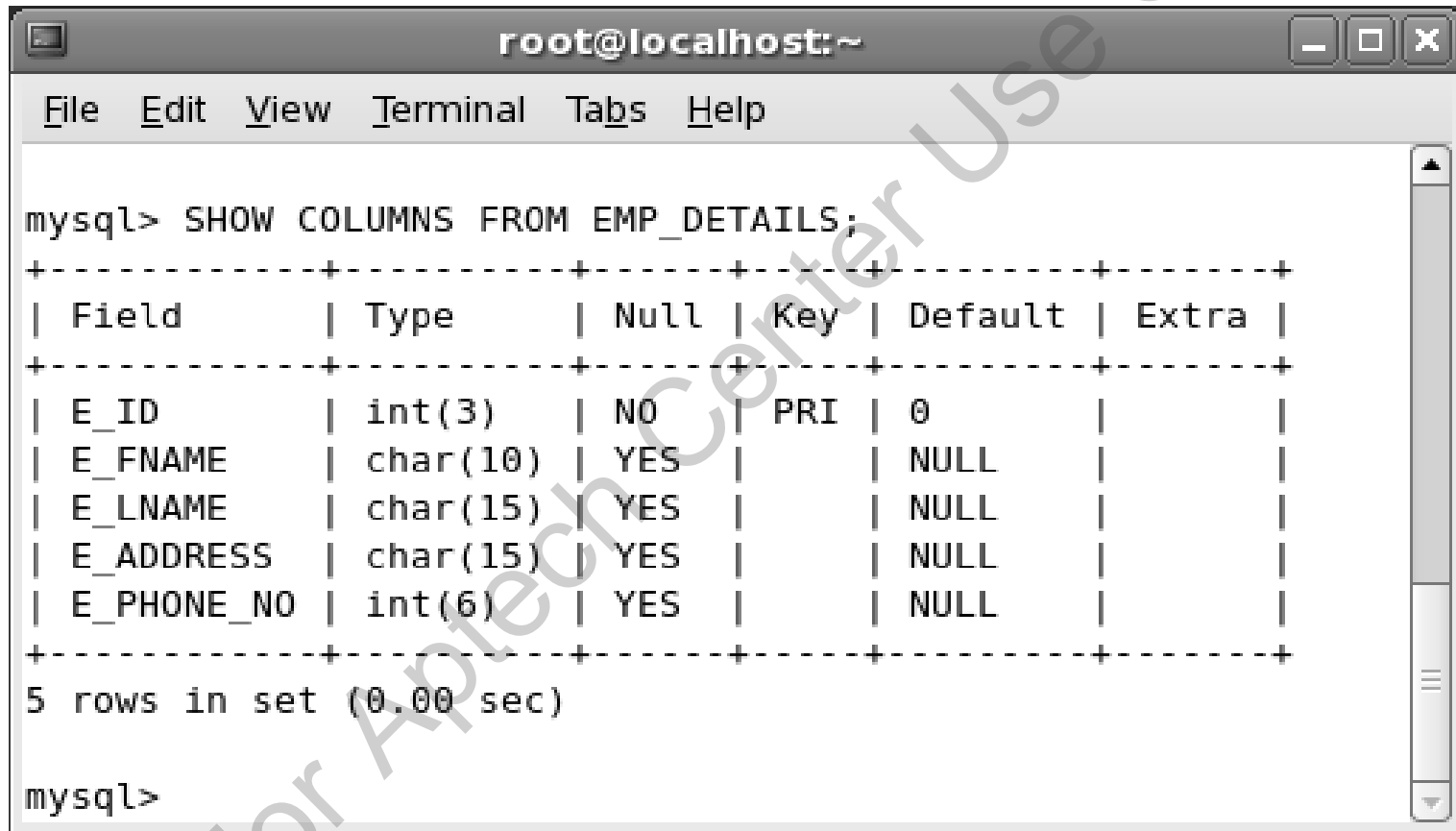
`database_name` – specifies the name of the database

`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



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal displays the command 'mysql> SHOW COLUMNS FROM EMP_DETAILS;' and its output. The output is a table with 6 columns: Field, Type, Null, Key, Default, and Extra. There are 5 rows of data. Below the table, it says '5 rows in set (0.00 sec)'. The prompt 'mysql>' is visible at the bottom.

```
mysql> SHOW COLUMNS FROM EMP_DETAILS;
```

Field	Type	Null	Key	Default	Extra
E_ID	int(3)	NO	PRI	0	
E_FNAME	char(10)	YES		NULL	
E_LNAME	char(15)	YES		NULL	
E_ADDRESS	char(15)	YES		NULL	
E_PHONE_NO	int(6)	YES		NULL	

5 rows in set (0.00 sec)

```
mysql>
```

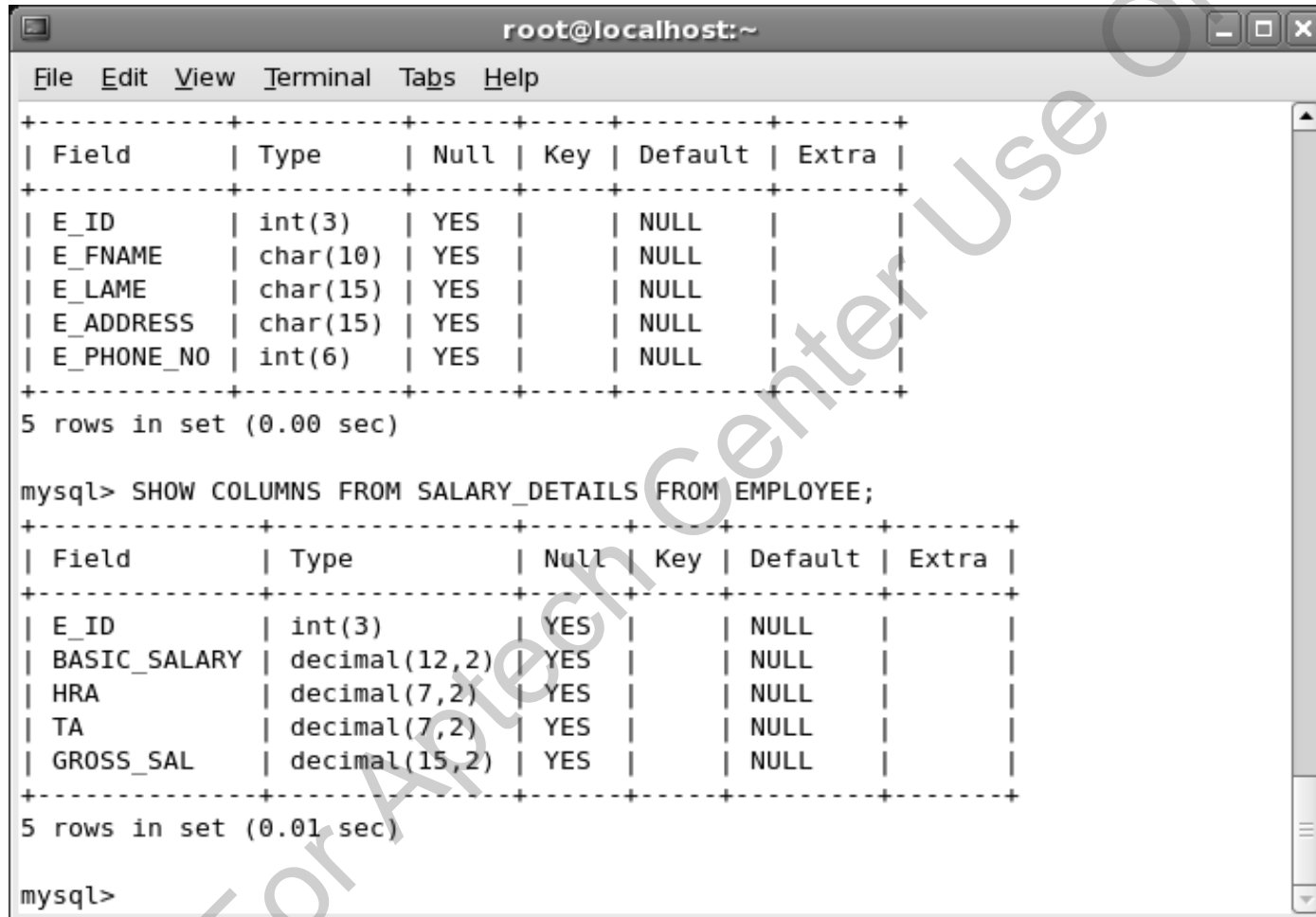
Table lists the description of each of the columns displayed in the figure

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 screenshot shows a terminal window titled 'root@localhost:~'. The window contains the output of a MySQL command. The output is divided into two sections, each displaying a table of column information. The first section shows the output of a command that has been partially obscured by a watermark. The second section shows the output of the command 'mysql> SHOW COLUMNS FROM SALARY_DETAILS FROM EMPLOYEE;'. Both sections display a table with columns: Field, Type, Null, Key, Default, and Extra. The first table has 5 rows, and the second table has 5 rows.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| E_ID | int(3) | YES | | NULL | |  
| E_FNAME | char(10) | YES | | NULL | |  
| E_LAME | char(15) | YES | | NULL | |  
| E_ADDRESS | char(15) | YES | | NULL | |  
| E_PHONE_NO | int(6) | YES | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)  
  
mysql> SHOW COLUMNS FROM SALARY_DETAILS FROM EMPLOYEE;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| E_ID | int(3) | YES | | NULL | |  
| BASIC_SALARY | decimal(12,2) | YES | | NULL | |  
| HRA | decimal(7,2) | YES | | NULL | |  
| TA | decimal(7,2) | YES | | NULL | |  
| GROSS_SAL | decimal(15,2) | YES | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.01 sec)  
  
mysql>
```

- ◆ 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

- ◆ 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

- ◆ 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

```

root@localhost:~
File Edit View Terminal Tabs Help

mysql> SHOW INDEX FROM EMP_DETAILS FROM EMPLOYEE;
+-----+-----+-----+-----+-----+-----+-----+
| Table      | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-----+-----+-----+-----+-----+-----+-----+
| EMP_DETAILS | 0          | PRIMARY | 1            | E_ID        | A         | 6           | NULL     | NULL   |      | BTREE      |         |               |
| EMP_DETAILS | 1          | E_ID    | 1            | E_ID        | A         | 6           | NULL     | NULL   |      | BTREE      |         |               |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
  
```

- ◆ 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:

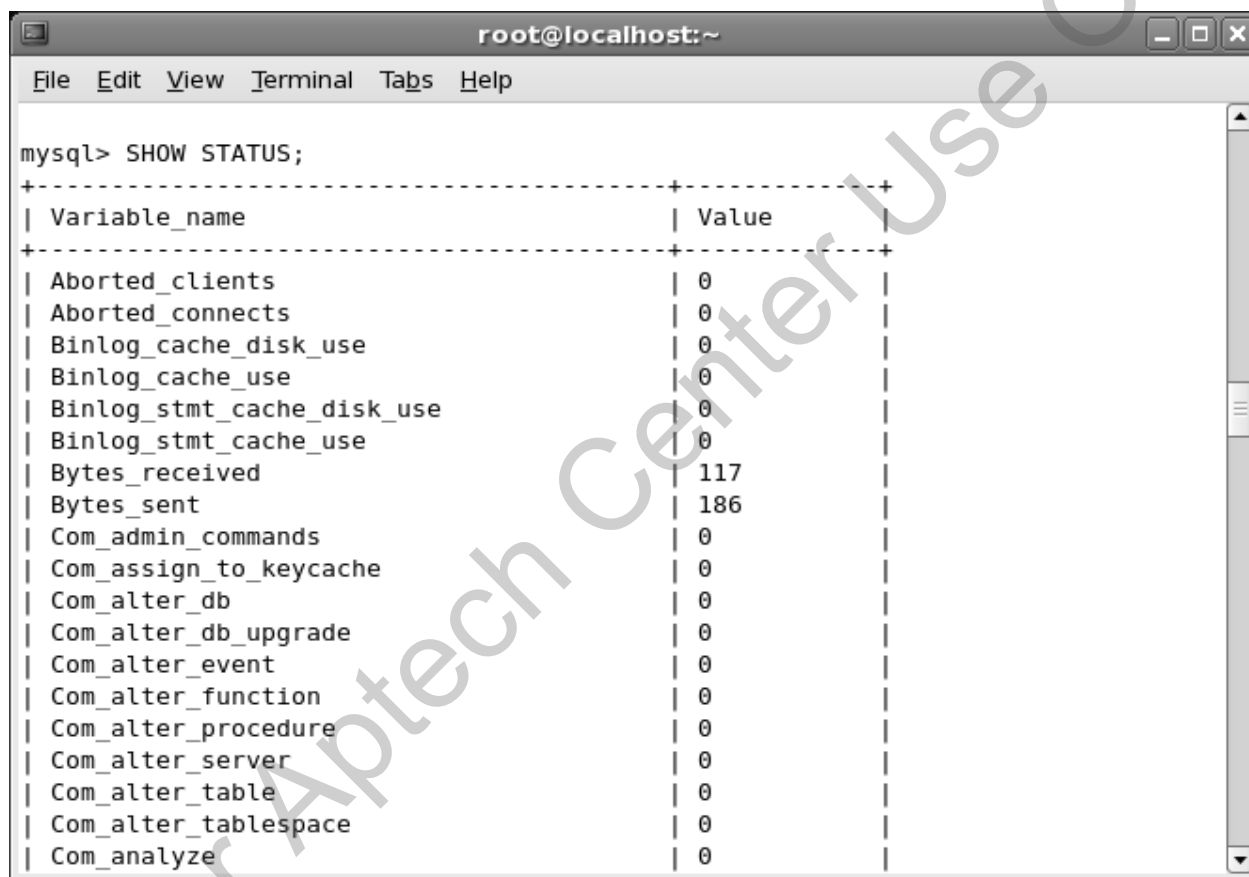
```
SHOW STATUS;
```

where,

SHOW – displays the object specified in the clause

STATUS – displays information about the server

Figure displays the output of the command



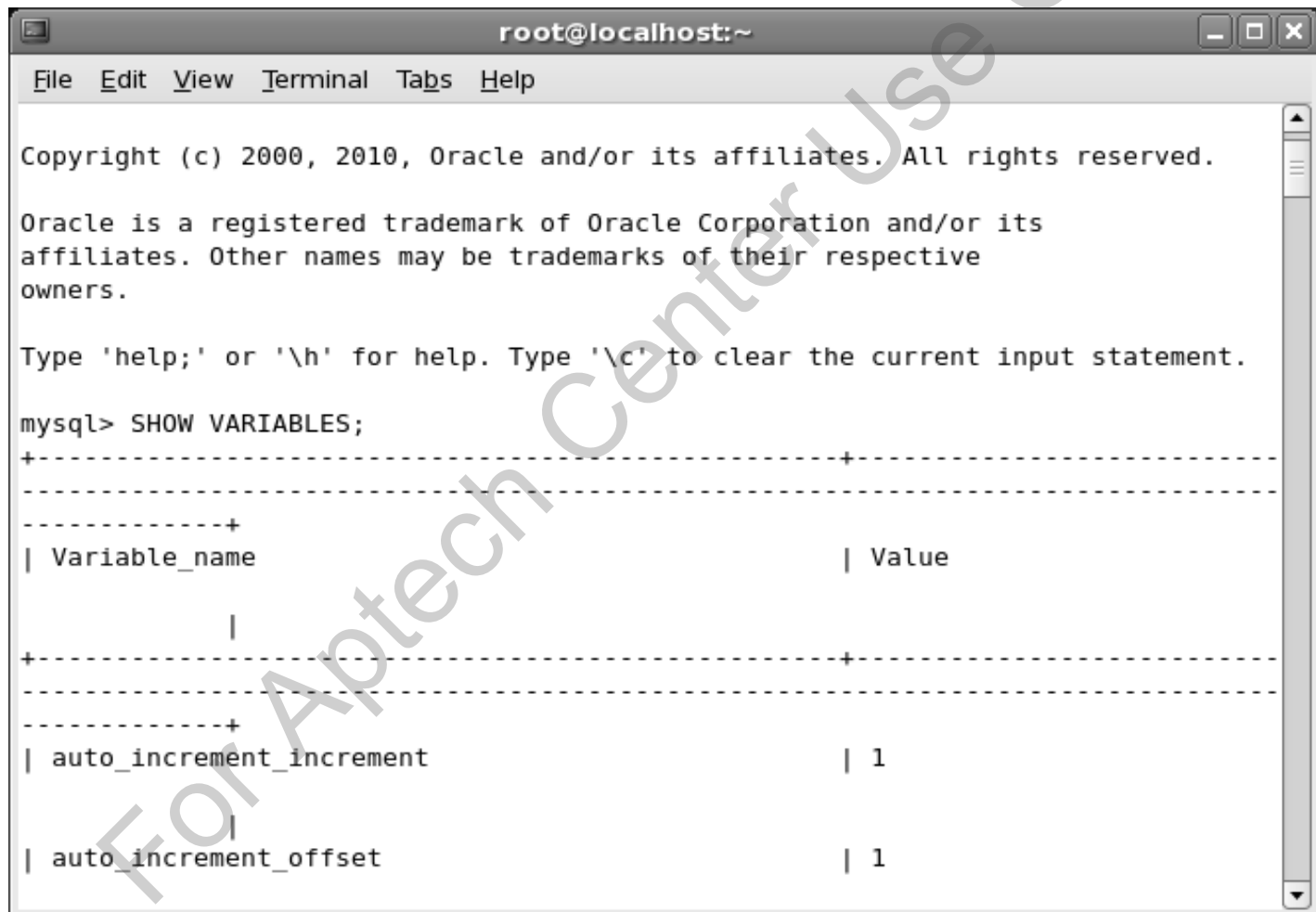
A terminal window titled 'root@localhost:~' displays the output of the MySQL command 'SHOW STATUS;'. The output is a table with two columns: 'Variable_name' and 'Value'. The table lists various MySQL status variables and their current values. A large, semi-transparent watermark 'For Apteck Center Use Only' is overlaid diagonally across the terminal window.

Variable_name	Value
Aborted_clients	0
Aborted_connects	0
Binlog_cache_disk_use	0
Binlog_cache_use	0
Binlog_stmt_cache_disk_use	0
Binlog_stmt_cache_use	0
Bytes_received	117
Bytes_sent	186
Com_admin_commands	0
Com_assign_to_keycache	0
Com_alter_db	0
Com_alter_db_upgrade	0
Com_alter_event	0
Com_alter_function	0
Com_alter_procedure	0
Com_alter_server	0
Com_alter_table	0
Com_alter_tablespace	0
Com_analyze	0

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

```
SHOW VARIABLES;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The window contains the following text:

```
Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> SHOW VARIABLES;  
+-----+  
-----+  
| Variable_name | Value  
+-----+  
+-----+  
| auto_increment_increment | 1  
+-----+  
| auto_increment_offset | 1
```

The output is a table with two columns: 'Variable_name' and 'Value'. The table shows two rows of data: 'auto_increment_increment' with a value of 1, and 'auto_increment_offset' with a value of 1.

- ◆ 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

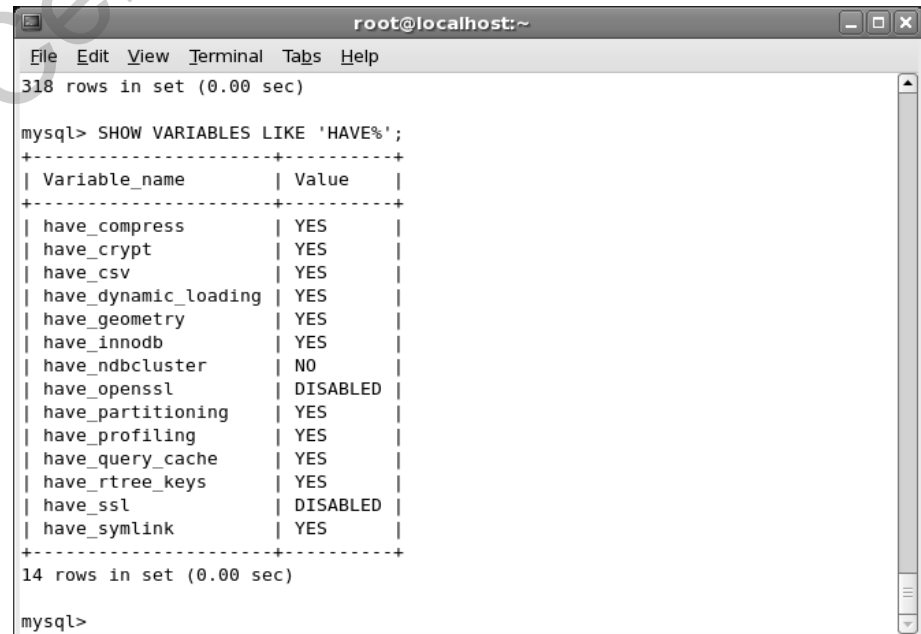
- ◆ To view only the variables that match a specified condition use the following syntax:

```
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



```
root@localhost:~  
File Edit View Terminal Tabs Help  
318 rows in set (0.00 sec)  
  
mysql> SHOW VARIABLES LIKE 'HAVE%';  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| have_compress | YES   |  
| have_crypt    | YES   |  
| have_csv      | YES   |  
| have_dynamic_loading | YES   |  
| have_geometry | YES   |  
| have_innodb   | YES   |  
| have_ndbcluster | NO    |  
| have_openssl  | DISABLED |  
| have_partitioning | YES   |  
| have_profiling | YES   |  
| have_query_cache | YES   |  
| have_rtree_keys | YES   |  
| have_ssl      | DISABLED |  
| have_symlink  | YES   |  
+-----+-----+  
14 rows in set (0.00 sec)  
  
mysql>
```

- ◆ 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

- ◆ 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

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

```

root@localhost:~
File Edit View Terminal Tabs Help

mysql> SHOW TABLE STATUS FROM EMPLOYEE;
+-----+-----+-----+-----+-----+-----+-----+
| Name          | Engine | Version | Row_format | Rows | Avg_row_length | Data_
length | Max_data_length | Index_length | Data_free | Auto_increment | Create_ti
me      | Update_time | Check_time | Collation      | Checksum | Create_op
tions | Comment |
+-----+-----+-----+-----+-----+-----+-----+
| EMP_DEPARTMENT | InnoDB | 10      | Compact    | 0    | 0              | 0    |
16384 | 0              | 0      | 10485760 | NULL | 2011-03-0
3 16:51:43 | NULL   | NULL    | latin1_swedish_ci | NULL |
| EMP_DETAILS    | InnoDB | 10      | Compact    | 0    | 0              | 0    |
16384 | 0              | 0      | 10485760 | NULL | 2011-03-0
3 16:48:12 | NULL   | NULL    | latin1_swedish_ci | NULL |
| SALARY_DETAILS | InnoDB | 10      | Compact    | 0    | 0              | 0    |
16384 | 0              | 0      | 10485760 | NULL | 2011-03-0

```

- ◆ 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

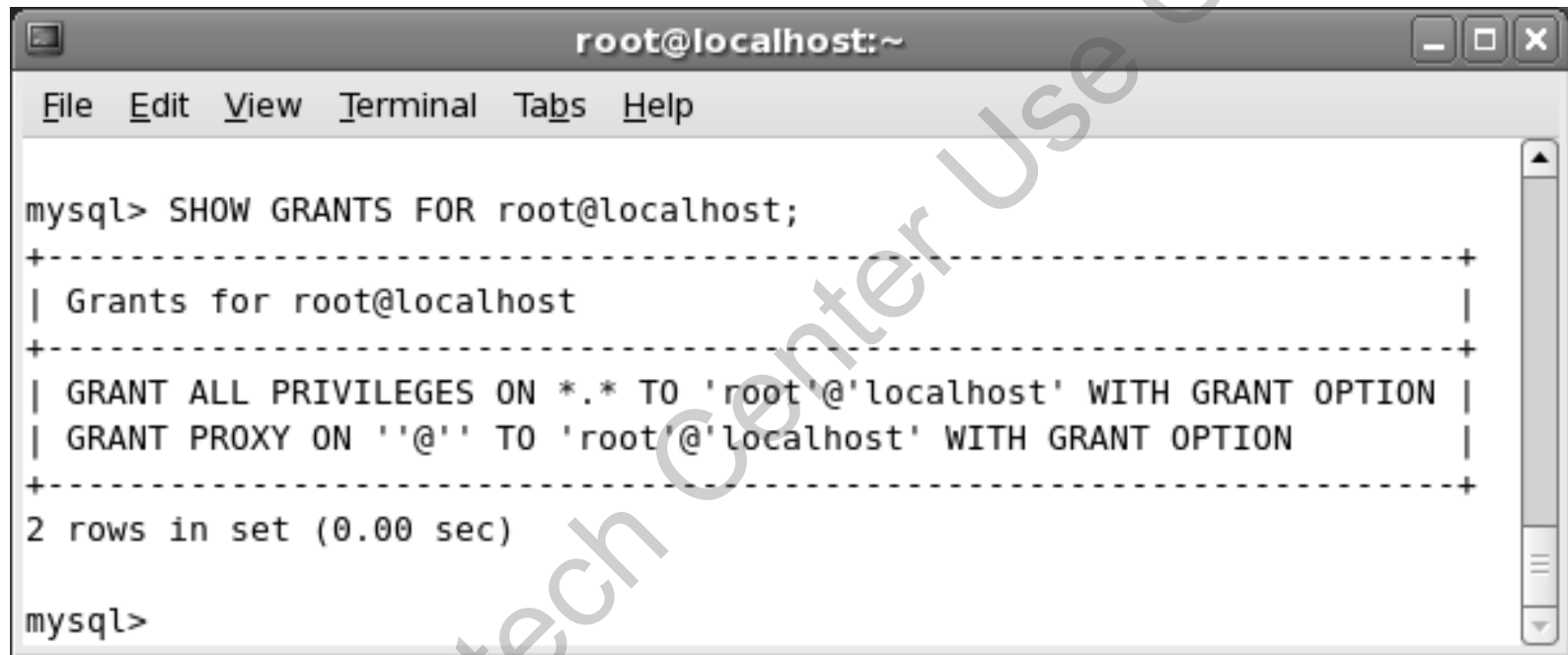
FOR – specifies the object to display the privileges

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

- ◆ 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



```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> SHOW GRANTS FOR root@localhost;  
+-----+  
| Grants for root@localhost |  
+-----+  
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION |  
| GRANT PROXY ON ''@' TO 'root'@'localhost' WITH GRANT OPTION |  
+-----+  
2 rows in set (0.00 sec)  
mysql>
```

- ◆ 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

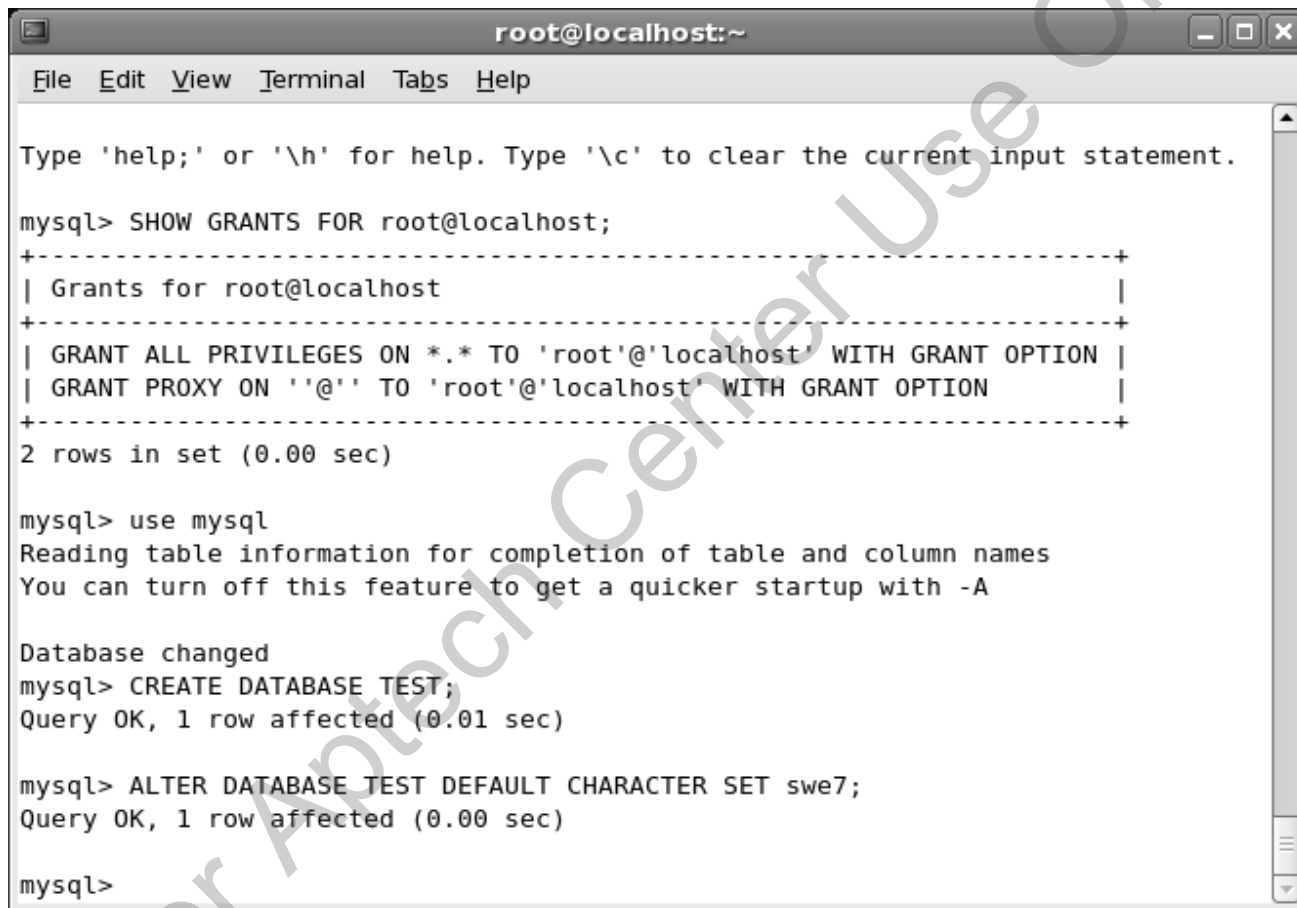
DEFAULT CHARACTER SET – specifies the default character set for the database

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 screenshot shows a terminal window titled 'root@localhost:~'. The terminal displays the output of the 'SHOW GRANTS' command for the 'root@localhost' user. The output is as follows:

```
mysql> SHOW GRANTS FOR root@localhost;
+-----+
| Grants for root@localhost                                     |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
| GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION       |
+-----+
2 rows in set (0.00 sec)
```

Following this, the user enters 'use mysql', and the terminal shows the message: 'Reading table information for completion of table and column names. You can turn off this feature to get a quicker startup with -A'. Then, the user enters 'CREATE DATABASE TEST;', and the terminal shows: 'Database changed' and 'Query OK, 1 row affected (0.01 sec)'. Finally, the user enters 'ALTER DATABASE TEST DEFAULT CHARACTER SET swe7;', and the terminal shows: 'Query OK, 1 row affected (0.00 sec)'. The prompt 'mysql>' is visible at the bottom of the terminal.

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

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

For Aptech Center Use Only

- ◆ 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,

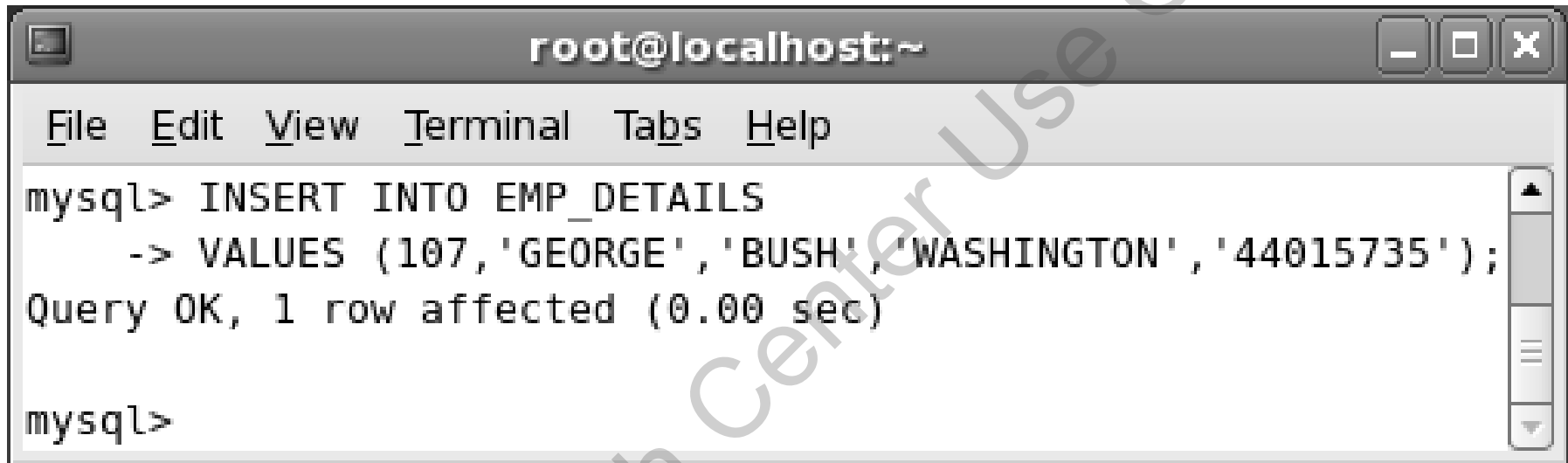
INSERT INTO – adds a new record to the table

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

A screenshot of a terminal window titled 'root@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows a MySQL command: 'mysql> INSERT INTO EMP_DETAILS -> VALUES (107, 'GEORGE', 'BUSH', 'WASHINGTON', '44015735');'. Below the command, it says 'Query OK, 1 row affected (0.00 sec)'. The prompt 'mysql>' is visible at the bottom. A large, diagonal watermark 'For Aptech Center Use Only' is overlaid across the terminal window.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> INSERT INTO EMP_DETAILS  
-> VALUES (107, 'GEORGE', 'BUSH', 'WASHINGTON', '44015735');  
Query OK, 1 row affected (0.00 sec)  
  
mysql>
```


- ◆ 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;
```

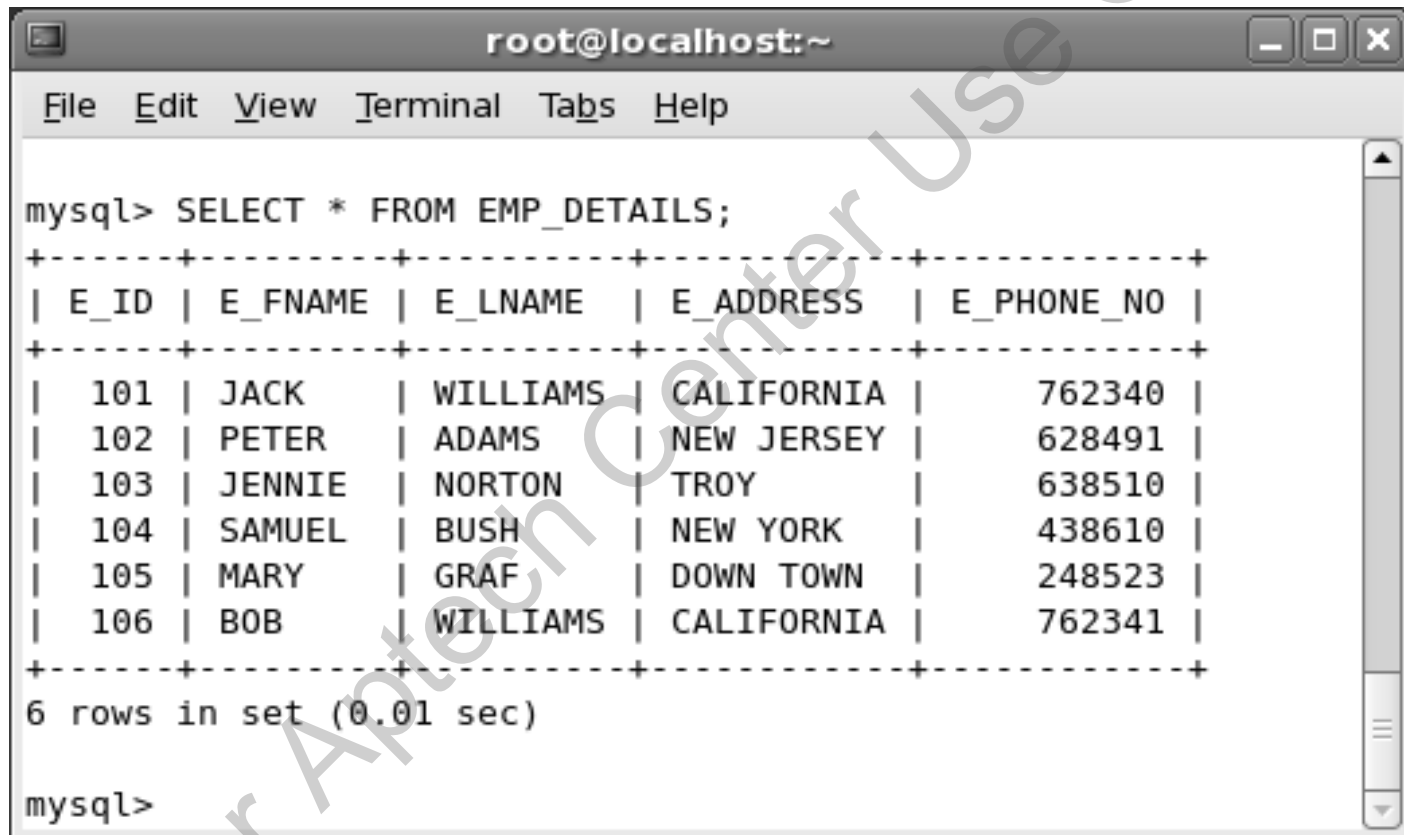
where,

`*` 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



The screenshot shows a terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal displays the command 'mysql> SELECT * FROM EMP_DETAILS;' and its output. The output is a table with 6 rows and 5 columns: E_ID, E_FNAME, E_LNAME, E_ADDRESS, and E_PHONE_NO. The data is as follows:

E_ID	E_FNAME	E_LNAME	E_ADDRESS	E_PHONE_NO
101	JACK	WILLIAMS	CALIFORNIA	762340
102	PETER	ADAMS	NEW JERSEY	628491
103	JENNIE	NORTON	TROY	638510
104	SAMUEL	BUSH	NEW YORK	438610
105	MARY	GRAF	DOWN TOWN	248523
106	BOB	WILLIAMS	CALIFORNIA	762341

Below the table, the terminal shows '6 rows in set (0.01 sec)' and the prompt 'mysql>'.

- ◆ 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

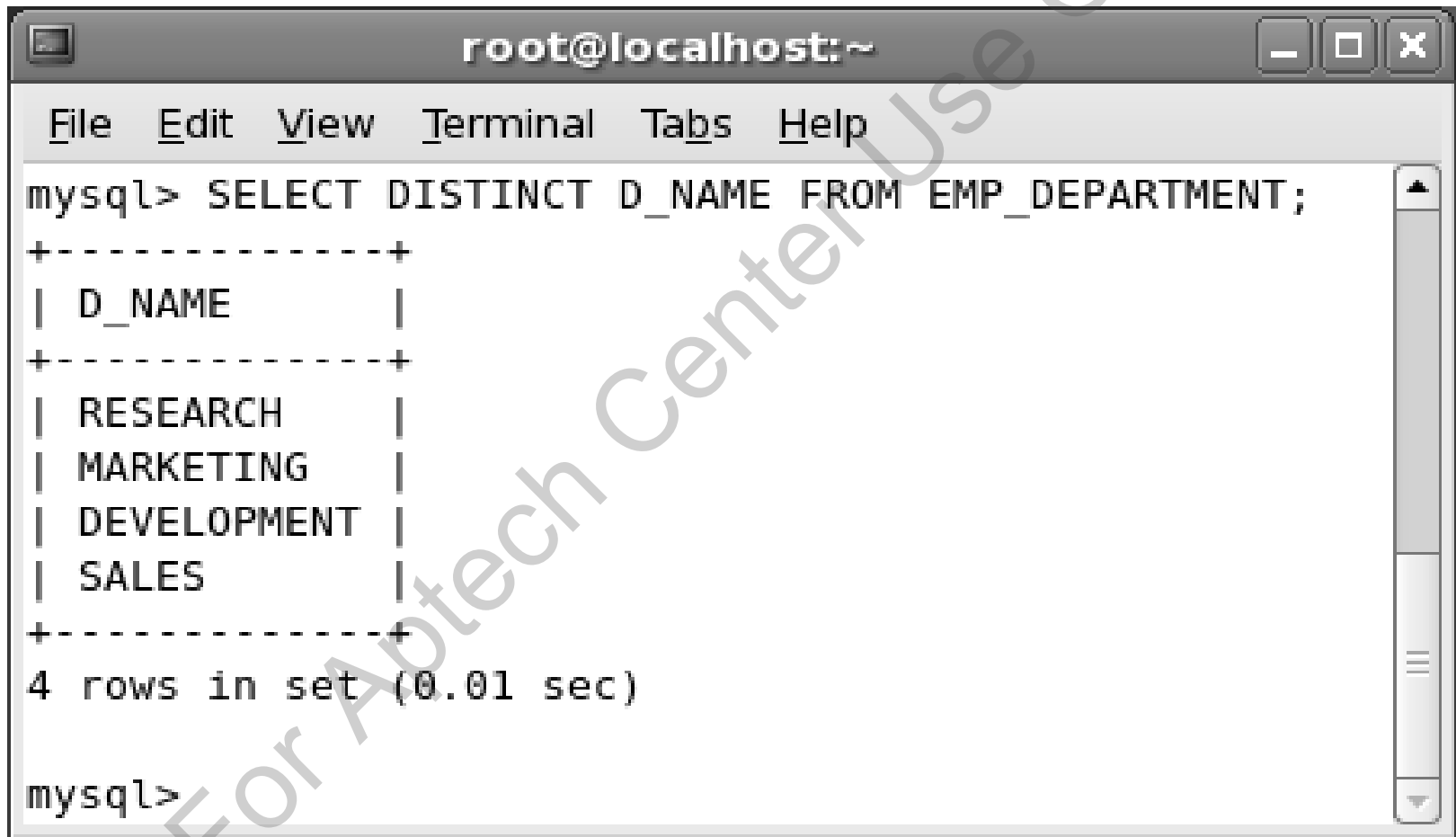
column_name – specifies the name of the column

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 screenshot shows a terminal window titled 'root@localhost:~'. The terminal displays the following text:

```
mysql> SELECT DISTINCT D_NAME FROM EMP_DEPARTMENT;  
+-----+  
| D_NAME |  
+-----+  
| RESEARCH |  
| MARKETING |  
| DEVELOPMENT |  
| SALES |  
+-----+  
4 rows in set (0.01 sec)
```

The output is formatted as a table with a header row and four data rows. The header row is 'D_NAME'. The data rows are 'RESEARCH', 'MARKETING', 'DEVELOPMENT', and 'SALES'. The table is enclosed in a box with '+' and '-' characters. Below the table, it says '4 rows in set (0.01 sec)'. The terminal prompt 'mysql>' is visible at the bottom.

- ◆ 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

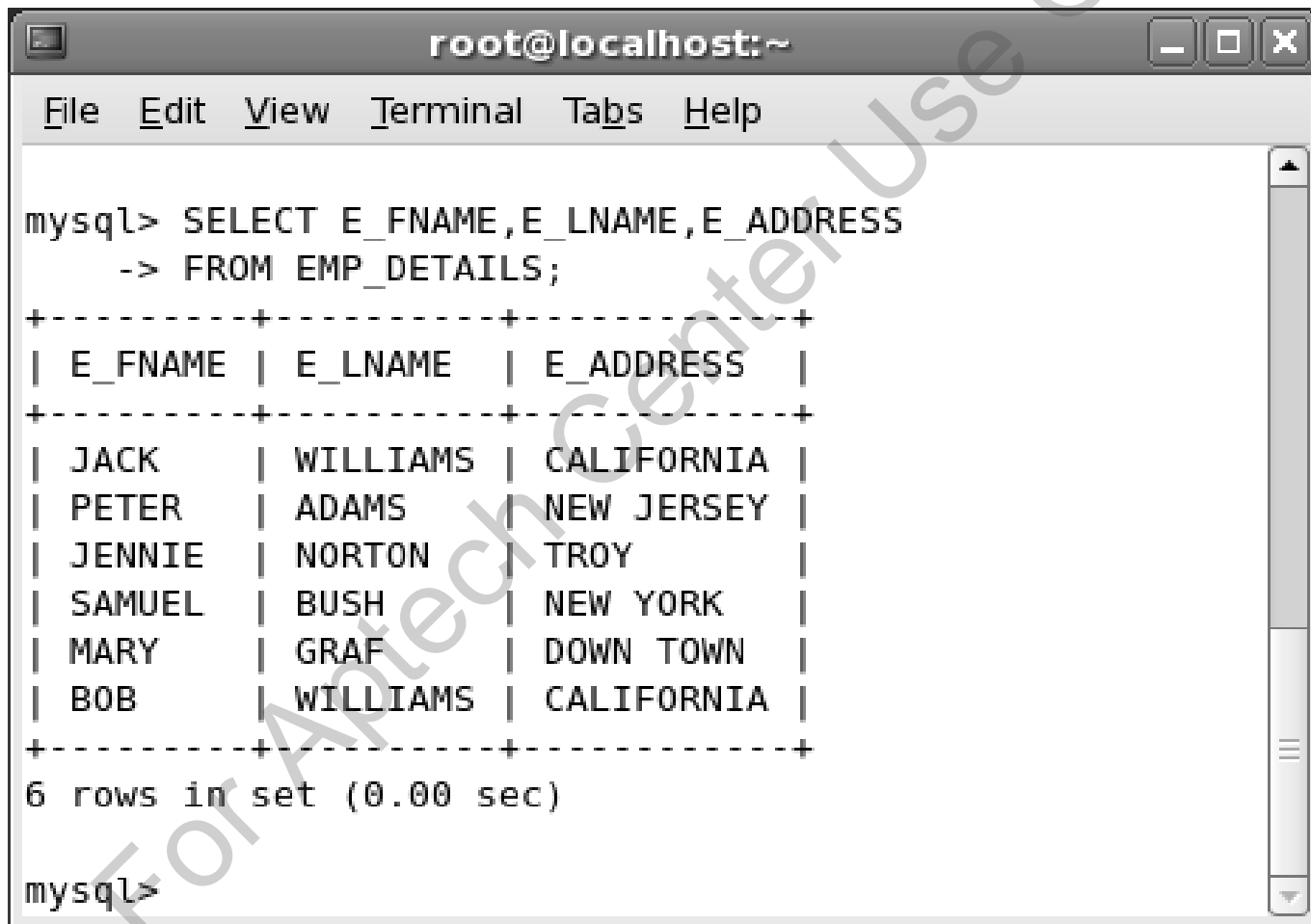
table_name – specifies the name of the table that contains the columns

Retrieving Data Using SELECT, FROM, DISTINCT, and WHERE Clauses 10-13

- ◆ 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



The screenshot shows a terminal window titled 'root@localhost:~' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal displays the following MySQL command and its output:

```
mysql> SELECT E_FNAME,E_LNAME,E_ADDRESS  
-> FROM EMP_DETAILS;
```

E_FNAME	E_LNAME	E_ADDRESS
JACK	WILLIAMS	CALIFORNIA
PETER	ADAMS	NEW JERSEY
JENNIE	NORTON	TROY
SAMUEL	BUSH	NEW YORK
MARY	GRAF	DOWN TOWN
BOB	WILLIAMS	CALIFORNIA

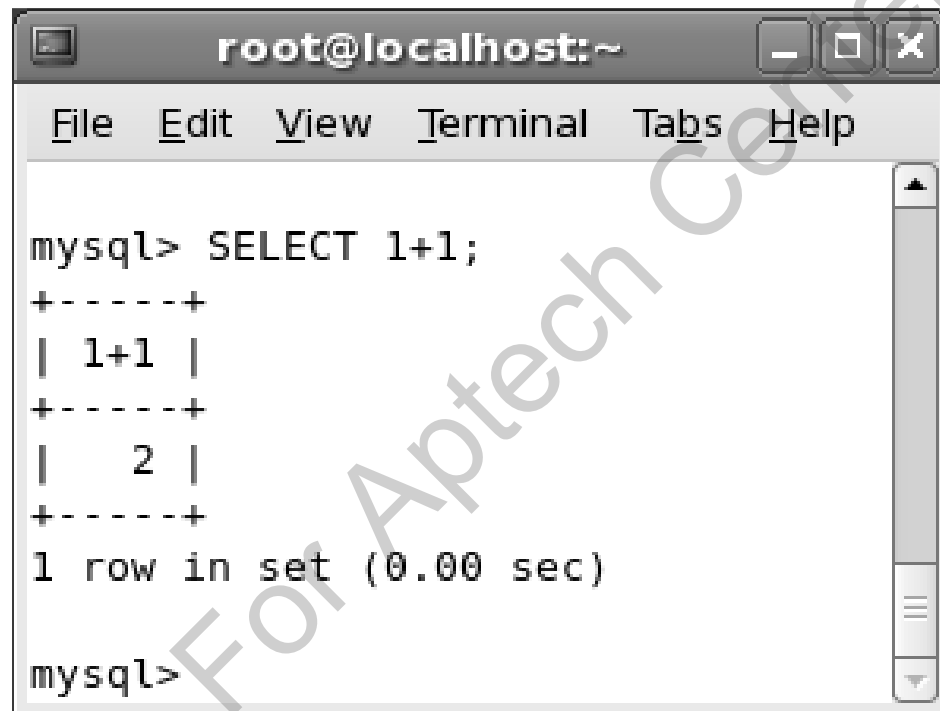
6 rows in set (0.00 sec)

```
mysql>
```

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

```
SELECT 1+1;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

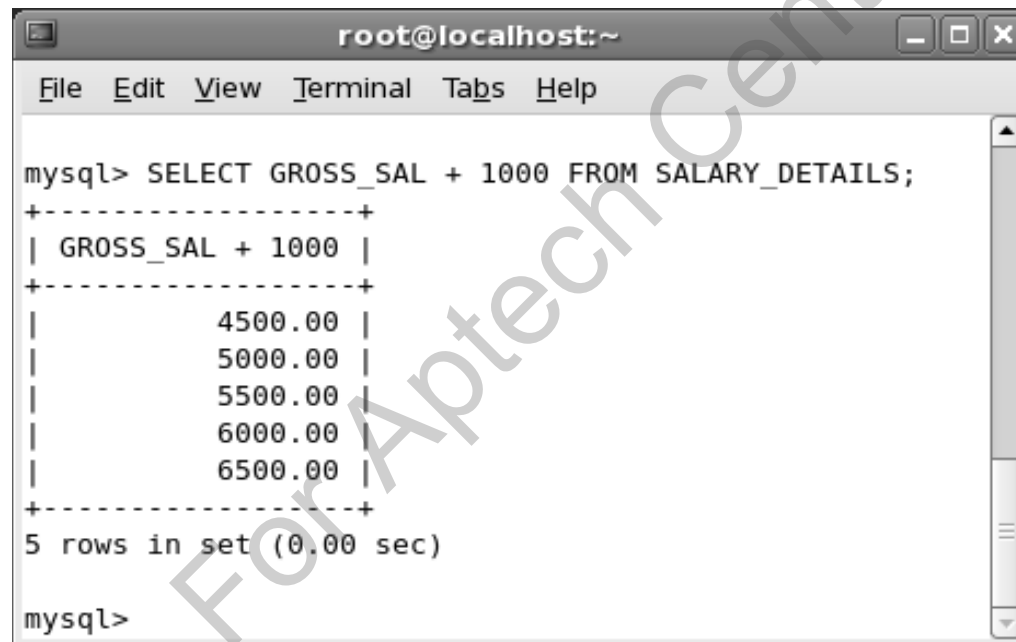
```
mysql> SELECT 1+1;
+-----+
| 1+1 |
+-----+
|    2 |
+-----+
1 row in set (0.00 sec)

mysql>
```

- ◆ 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



The screenshot shows a terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal displays the following text:

```
mysql> SELECT GROSS_SAL + 1000 FROM SALARY_DETAILS;  
+-----+  
| GROSS_SAL + 1000 |  
+-----+  
|          4500.00 |  
|          5000.00 |  
|          5500.00 |  
|          6000.00 |  
|          6500.00 |  
+-----+  
5 rows in set (0.00 sec)  
  
mysql>
```

The output is a table with one column, 'GROSS_SAL + 1000', containing five rows of values: 4500.00, 5000.00, 5500.00, 6000.00, and 6500.00. The table is enclosed in a box with dashed lines. Below the table, it says '5 rows in set (0.00 sec)'. The prompt 'mysql>' is shown at the bottom.

- ◆ 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

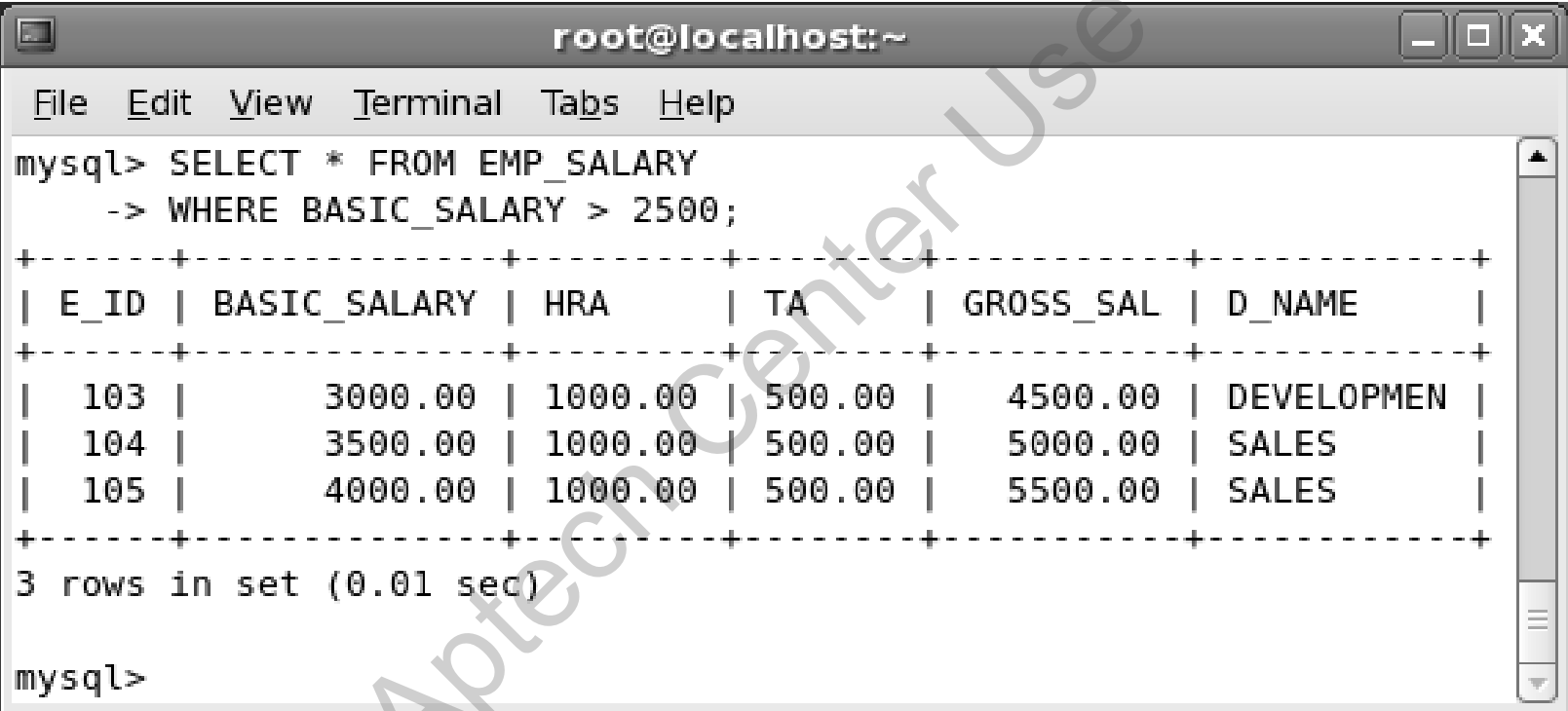
WHERE – specifies the clause to filter data

<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



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

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

E_ID	BASIC_SALARY	HRA	TA	GROSS_SAL	D_NAME
103	3000.00	1000.00	500.00	4500.00	DEVELOPMEN
104	3500.00	1000.00	500.00	5000.00	SALES
105	4000.00	1000.00	500.00	5500.00	SALES

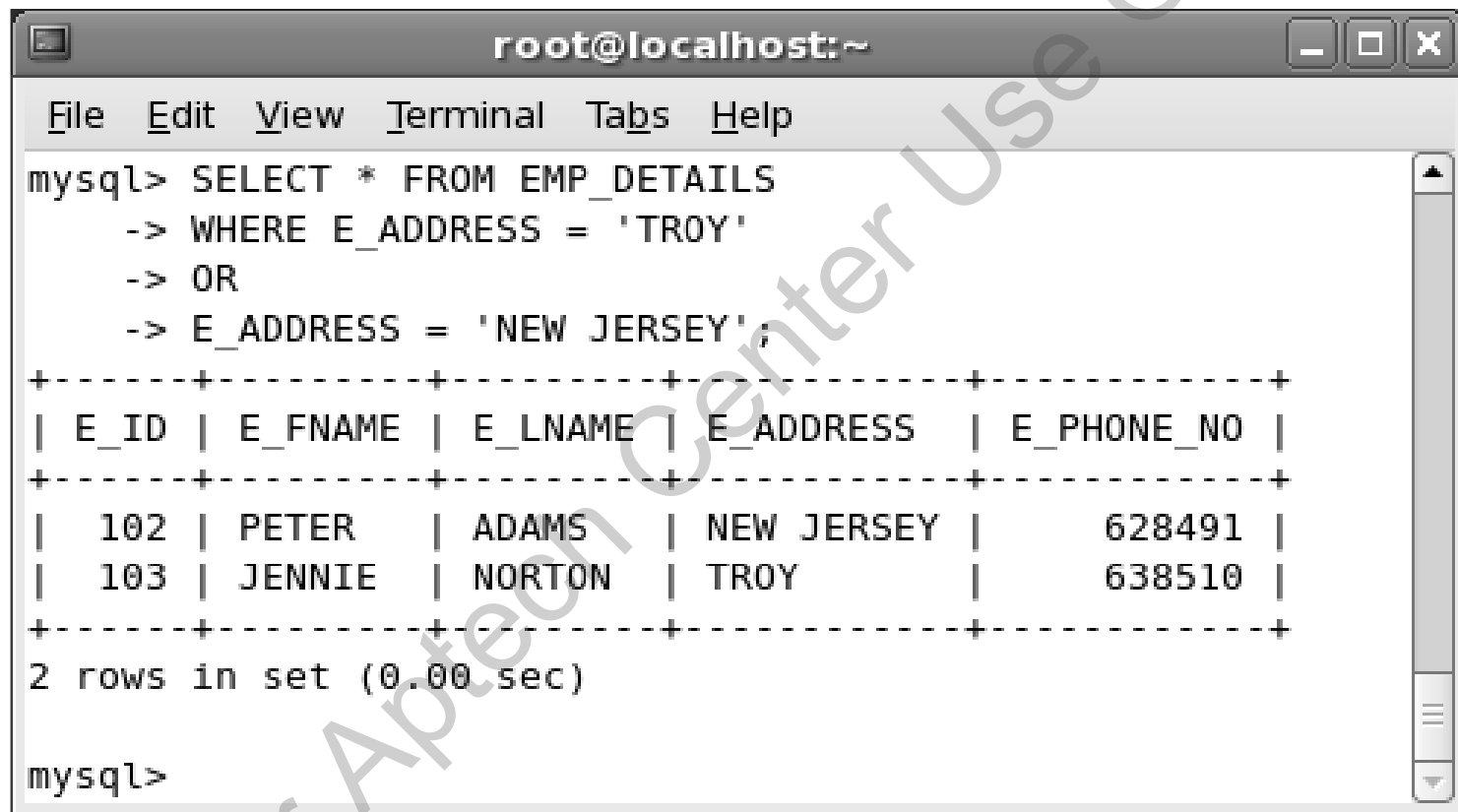
3 rows in set (0.01 sec)

```
mysql>
```

- ◆ 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



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

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

E_ID	E_FNAME	E_LNAME	E_ADDRESS	E_PHONE_NO
102	PETER	ADAMS	NEW JERSEY	628491
103	JENNIE	NORTON	TROY	638510

2 rows in set (0.00 sec)

```
mysql>
```


- ◆ 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

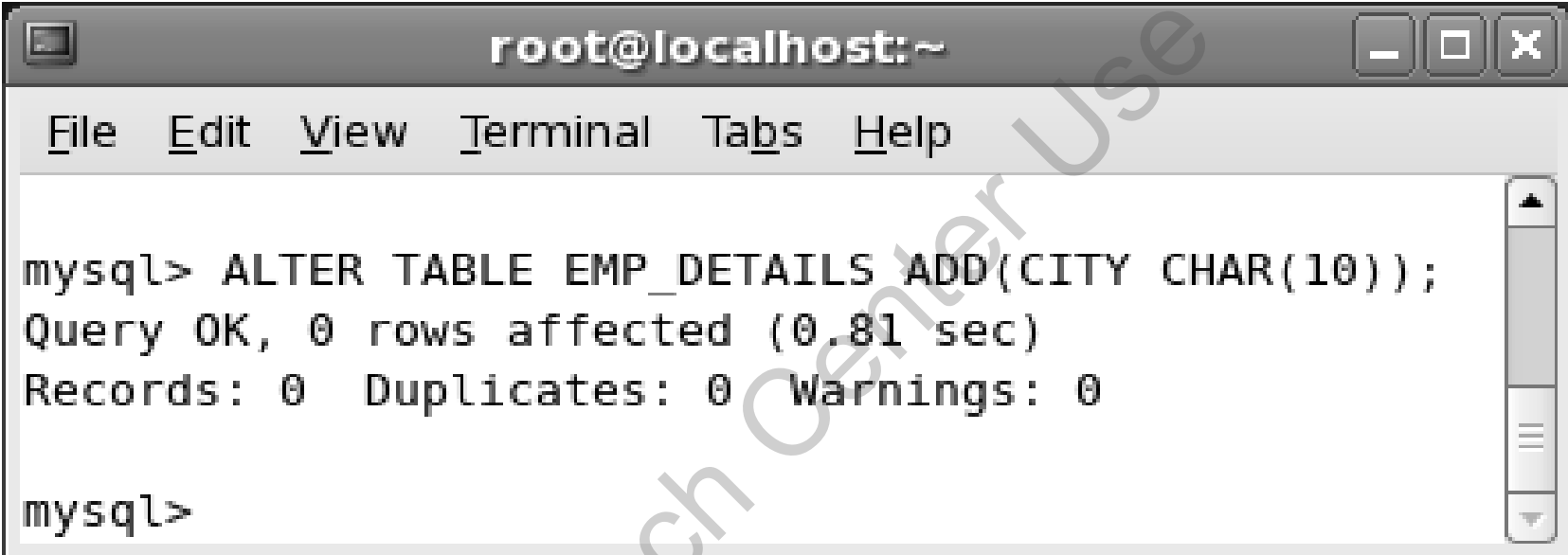
FIRST | AFTER column_name – specifies the location of the new column in the table

- ◆ 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)) ;
```

For Apteck Center Use Only

Figure displays the output of the command



The image shows a terminal window titled "root@localhost:~". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows a MySQL prompt "mysql>" followed by the command "ALTER TABLE EMP_DETAILS ADD(CITY CHAR(10));". The output of the command is "Query OK, 0 rows affected (0.81 sec)" and "Records: 0 Duplicates: 0 Warnings: 0". The prompt "mysql>" is shown again at the bottom. A large, diagonal watermark "For Aptech Center Use Only" is overlaid on the terminal window.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> ALTER TABLE EMP_DETAILS ADD(CITY CHAR(10));  
Query OK, 0 rows affected (0.81 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
mysql>
```

- ◆ 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

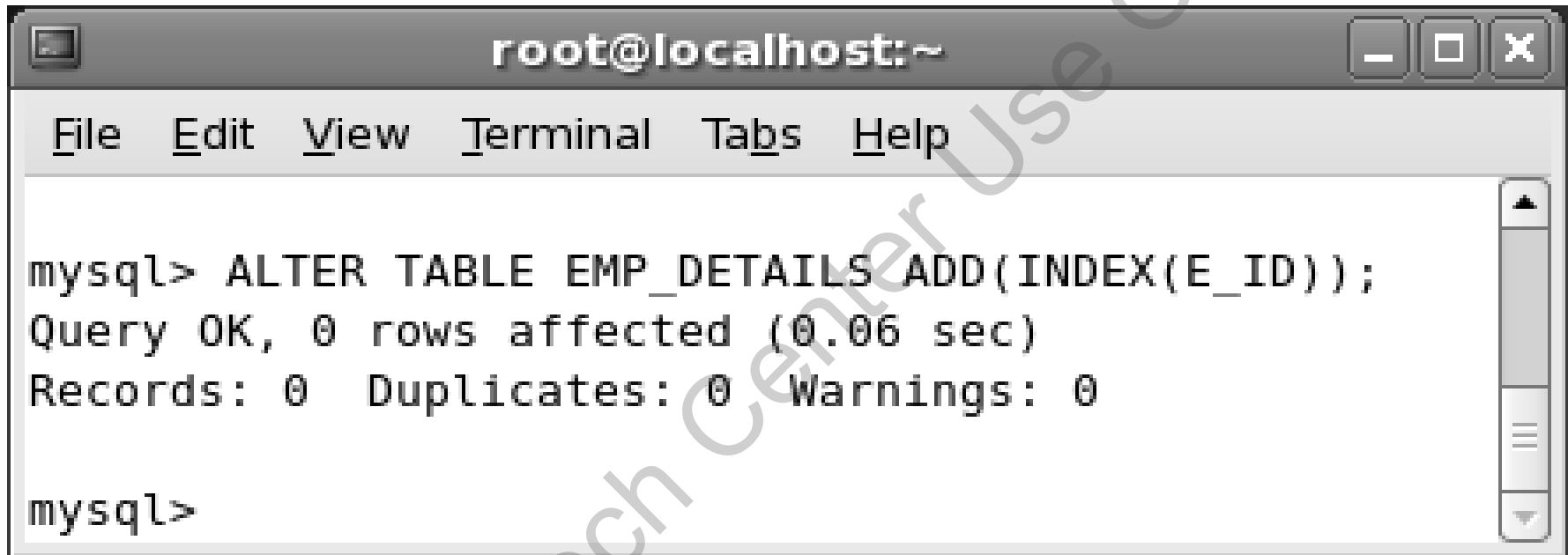
index_name – specifies a name for the index

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 image shows a terminal window titled "root@localhost:~". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows the following sequence of commands and output:

```
mysql> ALTER TABLE EMP_DETAILS ADD(INDEX(E_ID));  
Query OK, 0 rows affected (0.06 sec)  
Records: 0  Duplicates: 0  Warnings: 0  
  
mysql>
```

A large, diagonal watermark reading "For Apteck Center Use Only" is overlaid across the terminal window.

- ◆ 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

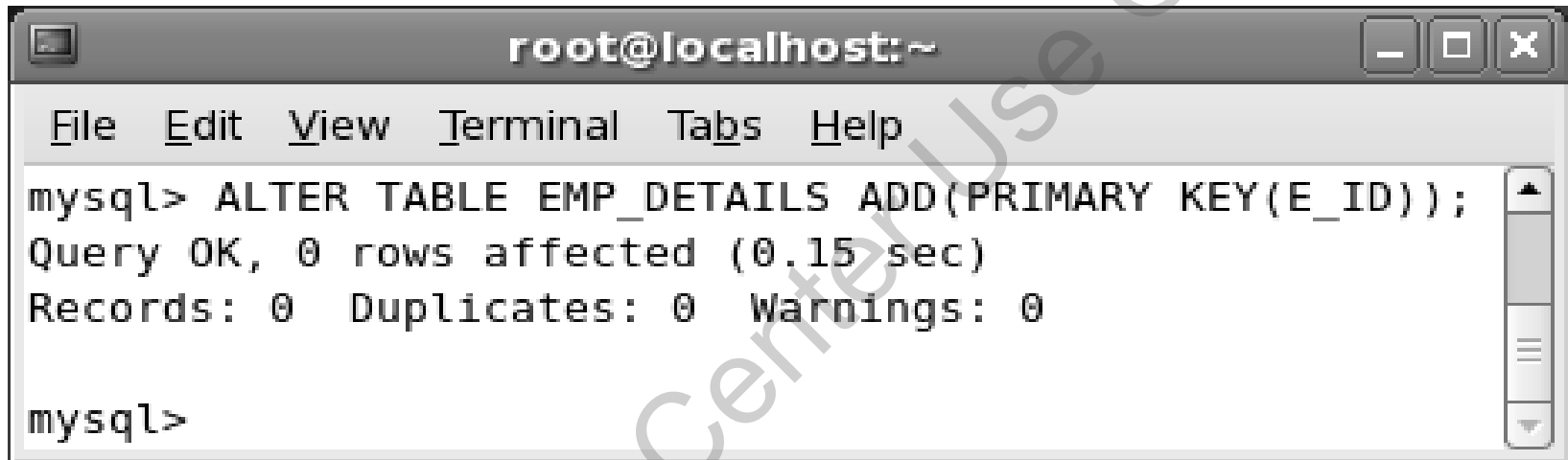
ADD PRIMARY KEY – appends a primary key to the table

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



The screenshot shows a terminal window titled "root@localhost:~". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows the following sequence of commands and output:

```
mysql> ALTER TABLE EMP_DETAILS ADD(PRIMARY KEY(E_ID));  
Query OK, 0 rows affected (0.15 sec)  
Records: 0  Duplicates: 0  Warnings: 0  
  
mysql>
```

A large, diagonal watermark reading "For Apteck Certified Use Only" is overlaid across the terminal window.

- ◆ 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

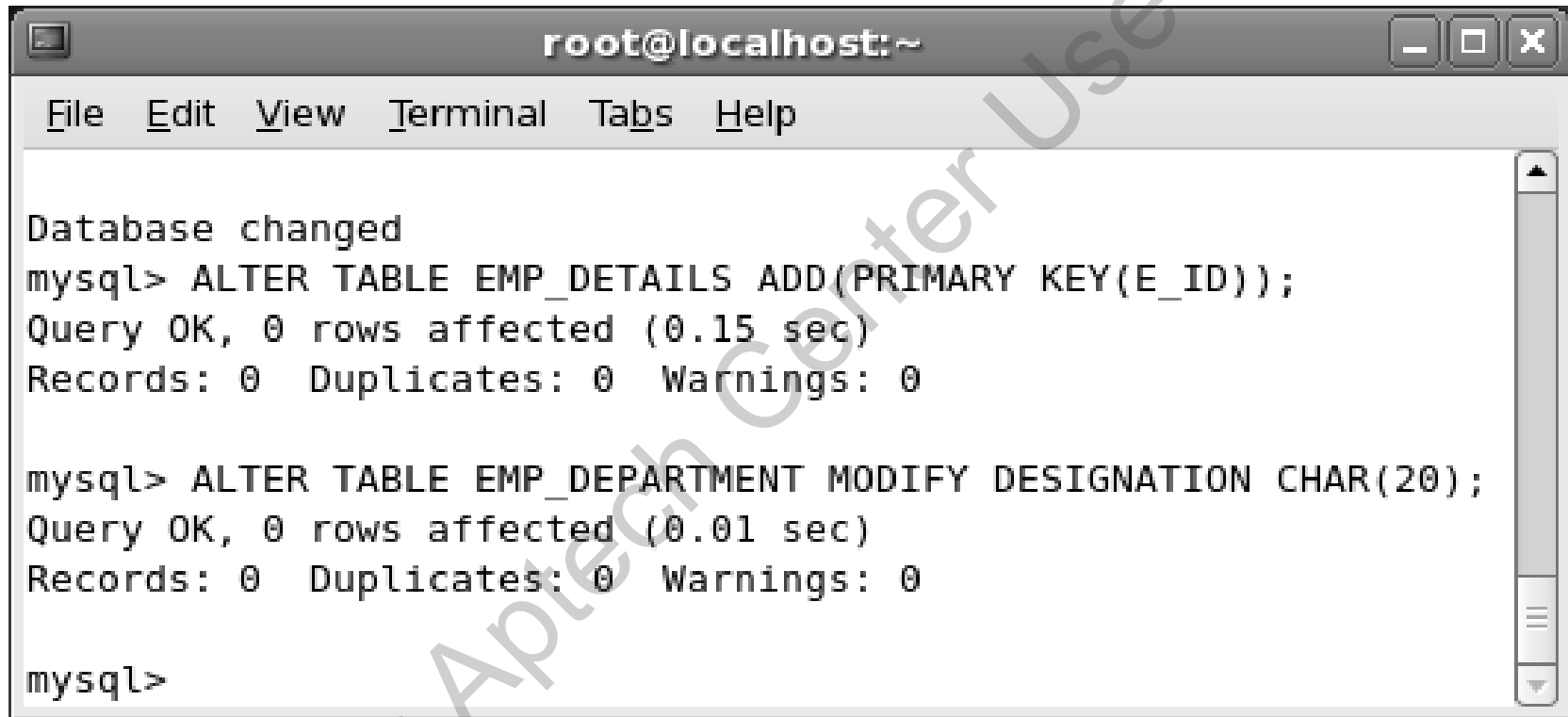
MODIFY [COLUMN] – specifies to change the column structure

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

A screenshot of a MySQL terminal window. The window title is 'root@localhost:~'. The menu bar includes 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows two successful ALTER TABLE commands. The first command adds a primary key to the EMP_DETAILS table, and the second command modifies the DESIGNATION column in the EMP_DEPARTMENT table. Both commands report 'Query OK, 0 rows affected' and 'Records: 0 Duplicates: 0 Warnings: 0'.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
Database changed  
mysql> ALTER TABLE EMP_DETAILS ADD(PRIMARY KEY(E_ID));  
Query OK, 0 rows affected (0.15 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> ALTER TABLE EMP_DEPARTMENT MODIFY DESIGNATION CHAR(20);  
Query OK, 0 rows affected (0.01 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql>
```

- ◆ 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

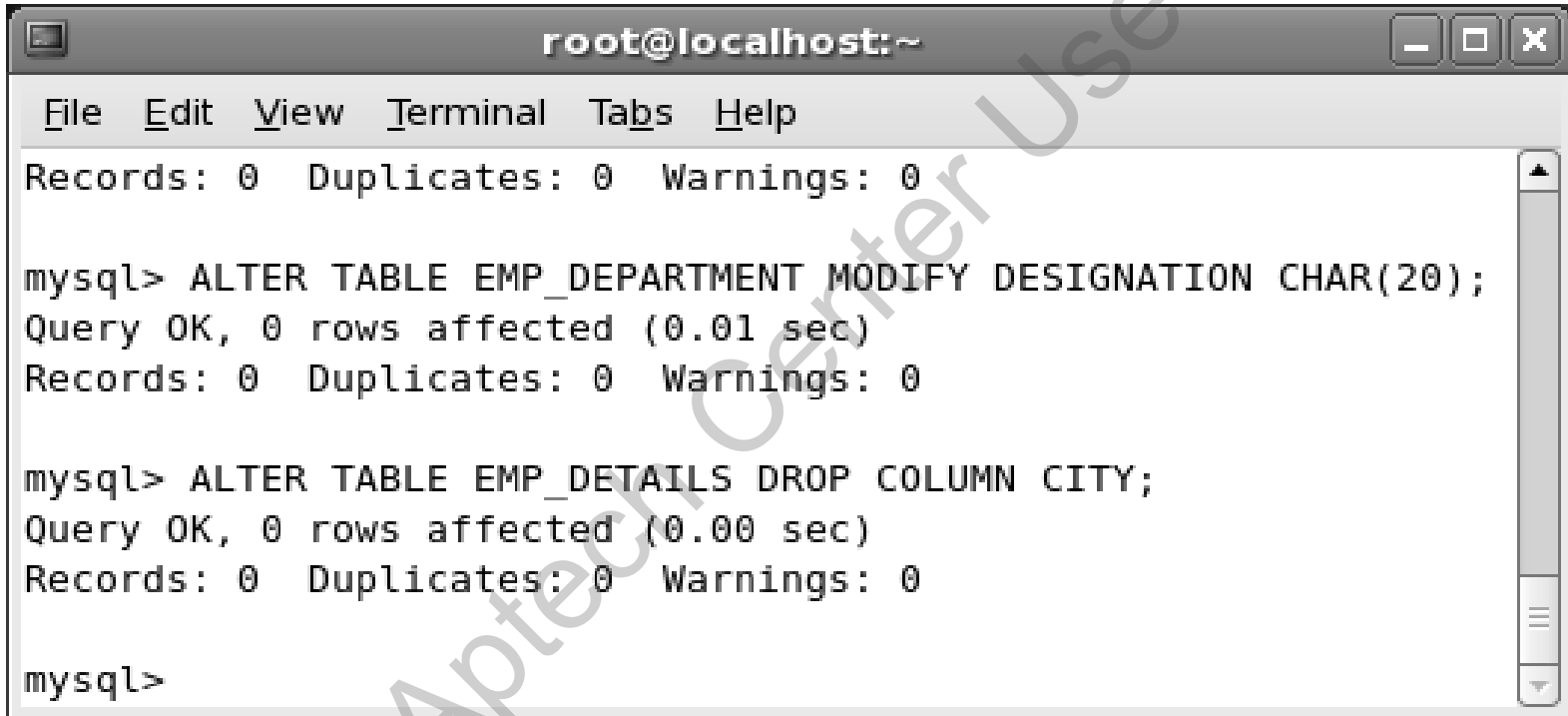
DROP [COLUMN] – removes the column from the table

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



```
root@localhost:~  
File Edit View Terminal Tabs Help  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> ALTER TABLE EMP_DEPARTMENT MODIFY DESIGNATION CHAR(20);  
Query OK, 0 rows affected (0.01 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> ALTER TABLE EMP_DETAILS DROP COLUMN CITY;  
Query OK, 0 rows affected (0.00 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql>
```

- ◆ 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

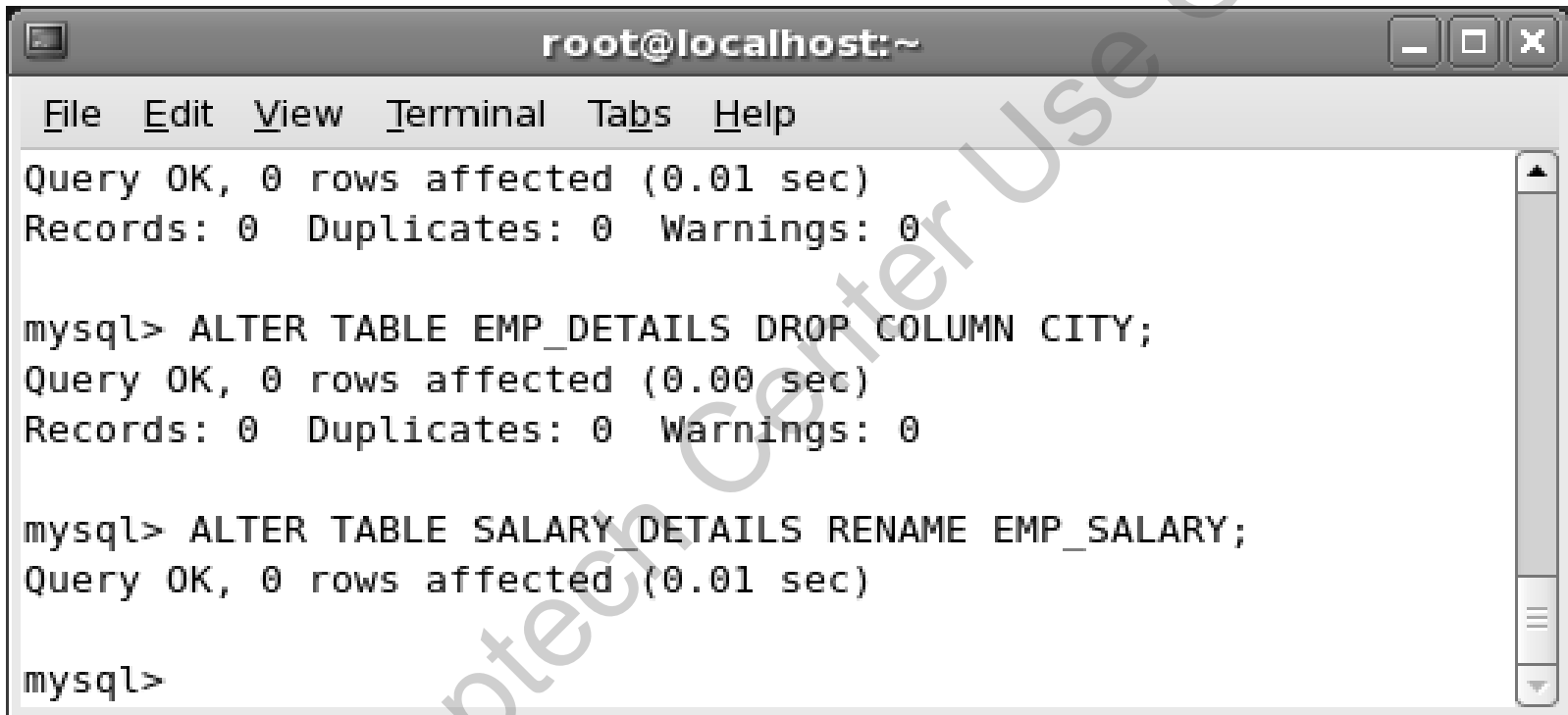
RENAME – changes the name of the table

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



The image shows a terminal window titled 'root@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal output shows the following sequence of commands and their results:

```
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE EMP_DETAILS DROP COLUMN CITY;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE SALARY_DETAILS RENAME EMP_SALARY;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

- ◆ 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

- ◆ 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

For Aptech Center Use Only

- ◆ 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

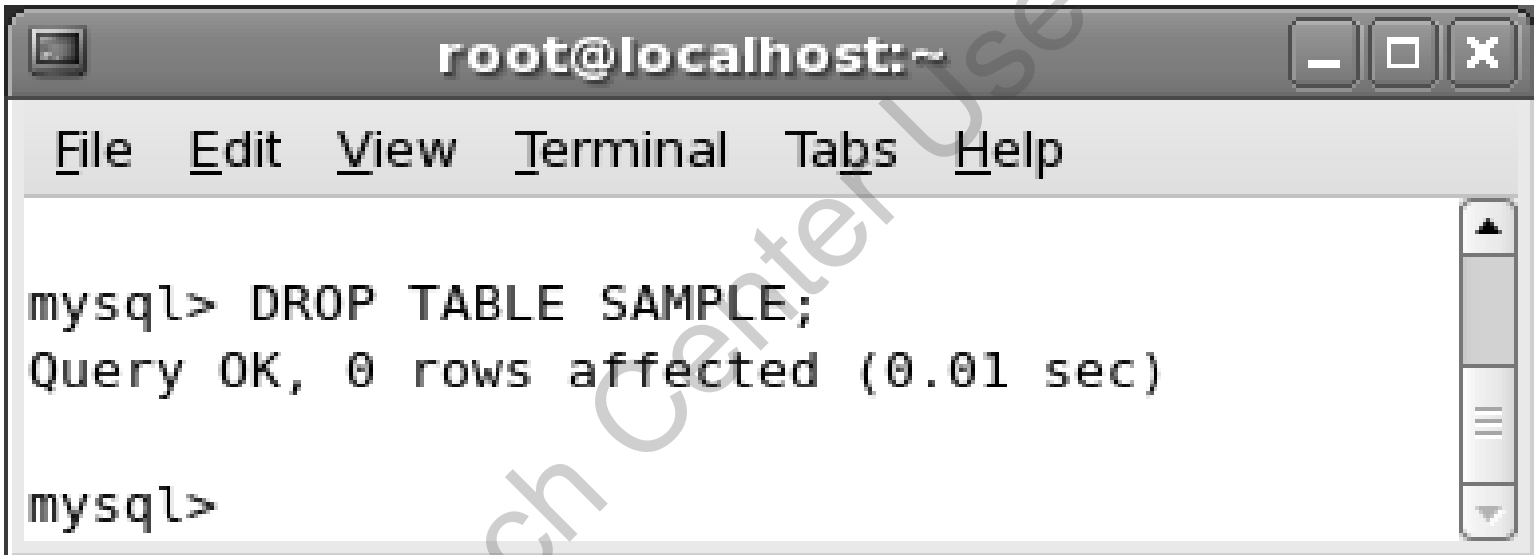
RESTRICT - specifies that if any dependencies exist, MySQL will not delete the table, if dependencies exist

CASCADE - specifies to remove dependencies before deleting the table

- ◆ 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



The image shows a terminal window titled "root@localhost:~". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows the MySQL command prompt "mysql>" followed by the command "DROP TABLE SAMPLE;". The output is "Query OK, 0 rows affected (0.01 sec)". The prompt "mysql>" is shown again on the next line. A large, diagonal watermark "For Aptech Center Use Only" is overlaid across the terminal window.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> DROP TABLE SAMPLE;  
Query OK, 0 rows affected (0.01 sec)  
mysql>
```

- ◆ 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 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

- ◆ 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 EXISTS` option in the `DROP` command to check the existence of the table before you delete it