

# Implementing SQL Queries Using MySQL - II

## Session 8



# Objectives

- ◆ *Explain the use of keys in a table*
- ◆ *Explain the use of indexes in a table*
- ◆ *Explain modification of tables*
- ◆ *Explain the use of the ORDER BY command*
- ◆ *Explain the use of the GROUP BY command*

- ◆ Keys are columns that uniquely identify information present in a table
- ◆ Some of the different types of keys include primary key, composite key, and foreign key

## ◆ Primary key

- ◆ A primary key is used to uniquely identify each row in a table
  - ◆ A primary key specifies that there cannot be a column in a table that contains two similar values
  - ◆ The primary key does not allow blank or `NULL` values for the column
- ◆ You must remember the following rules while defining a primary key:
- ◆ Contains a value and cannot contain a null value
  - ◆ Is unique for each record

- ◆ The syntax for defining a primary key while creating a table is:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name col_name  
column definition PRIMARY KEY (fieldname);
```

where,

CREATE – adds an object to the database

TABLE – adds a table to the database

tbl\_name – specifies the name for the table to add to the database

PRIMARY KEY – creates a primary key on the specified column

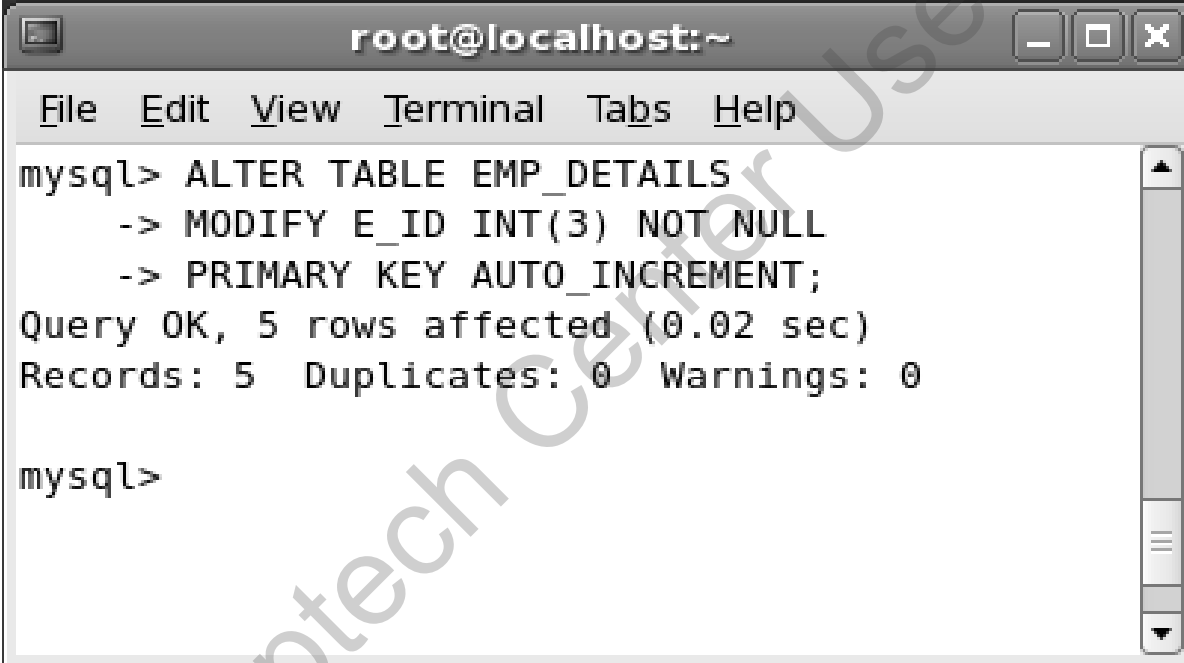
fieldname – specifies the name of the column to define the primary key

- ◆ MySQL provides different ways to define a primary key for a table:
  - ◆ You can define the primary key on a table while creating it
  - ◆ You can also modify an existing table structure and define a primary key using the `ALTER TABLE` command
  - ◆ You can modify the existing primary key on a table and specify conditions for the primary key

- ◆ For example, You have already created a primary key for the EMP\_DETAILS table
- ◆ You will now modify this key as NOT NULL and AUTO\_INCREMENT so that this primary key does not accept or contain a NULL value
- ◆ To modify the primary key as NOT NULL and AUTO\_INCREMENT, enter the following command at the command prompt:

```
ALTER TABLE EMP_DETAILS MODIFY E_ID INT(3) NOT  
NULL PRIMARY KEY AUTO_INCREMENT;
```

Figure displays the output of the command

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help) and standard window controls. The terminal shows a MySQL session where the 'ALTER TABLE EMP\_DETAILS' command is used to modify the 'E\_ID' column to 'INT(3) NOT NULL' and set it as a 'PRIMARY KEY AUTO\_INCREMENT'. The output indicates that 5 rows were affected in 0.02 seconds, with 0 duplicates and 0 warnings. The prompt returns to 'mysql>'.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> ALTER TABLE EMP_DETAILS  
      -> MODIFY E_ID INT(3) NOT NULL  
      -> PRIMARY KEY AUTO_INCREMENT;  
Query OK, 5 rows affected (0.02 sec)  
Records: 5  Duplicates: 0  Warnings: 0  
  
mysql>
```



## ◆ Composite Key

- ◆ When you use more than one column to define a primary key, then this type of a primary key is known as a composite key
- ◆ The rules for defining a composite key are the same as the primary key

## ◆ Foreign Key

- ◆ A foreign key is used to establish relationship between two tables
- ◆ A foreign key is a column in a table that has a corresponding primary key column in a different table
- ◆ The main goal of foreign key is to maintain referential integrity of the data
- ◆ You can define a column as a foreign key only if both the tables have `InnoDB` as their storage engine
- ◆ By default, MySQL assigns the `MyISAM` storage engine when you create a table
- ◆ you can change the storage engine of the table using the `ALTER TABLE` command

- ◆ The advantages of using InnoDB as the storage engine are:
  - ◆ Cancels INSERT and UPDATE commands that attempt to add records to the child table without having a corresponding record in the parent table
  - ◆ Allows the foreign key to reference a group of columns in the parent table
- ◆ To modify the storage engine of the EMP\_DETAILS to InnoDB, enter the following command at the command prompt:

```
ALTER TABLE EMP_DETAILS ENGINE=INNODB;
```

- ◆ To modify the EMP\_SALARY storage engine to InnoDB, enter the following command at the command prompt:

```
ALTER TABLE EMP_SALARY ENGINE=INNODB;
```

- ◆ The syntax to define a foreign key is:

```
ALTER TABLE tbl_name ADD [Constraint symbol] FOREIGN KEY  
[index_name] (index_col_name,...) REFERENCES table_name  
(index_col_name,...) [ON DELETE {CASCADE|SET NULL|NO  
ACTION|RESTRICT}] [ON UPDATE {CASCADE|SET NULL|NO  
ACTION|RESTRICT}];
```

where,

ALTER TABLE – modifies the table structure

tbl\_name – specifies the name of the table to edit

ADD – appends an object to the table structure

FOREIGN KEY – defines the foreign key to the column

index\_name – specifies the name of the index

index\_col\_name – specifies the name of the column defined as index

REFERENCES – defines the relationship with a primary key of another table

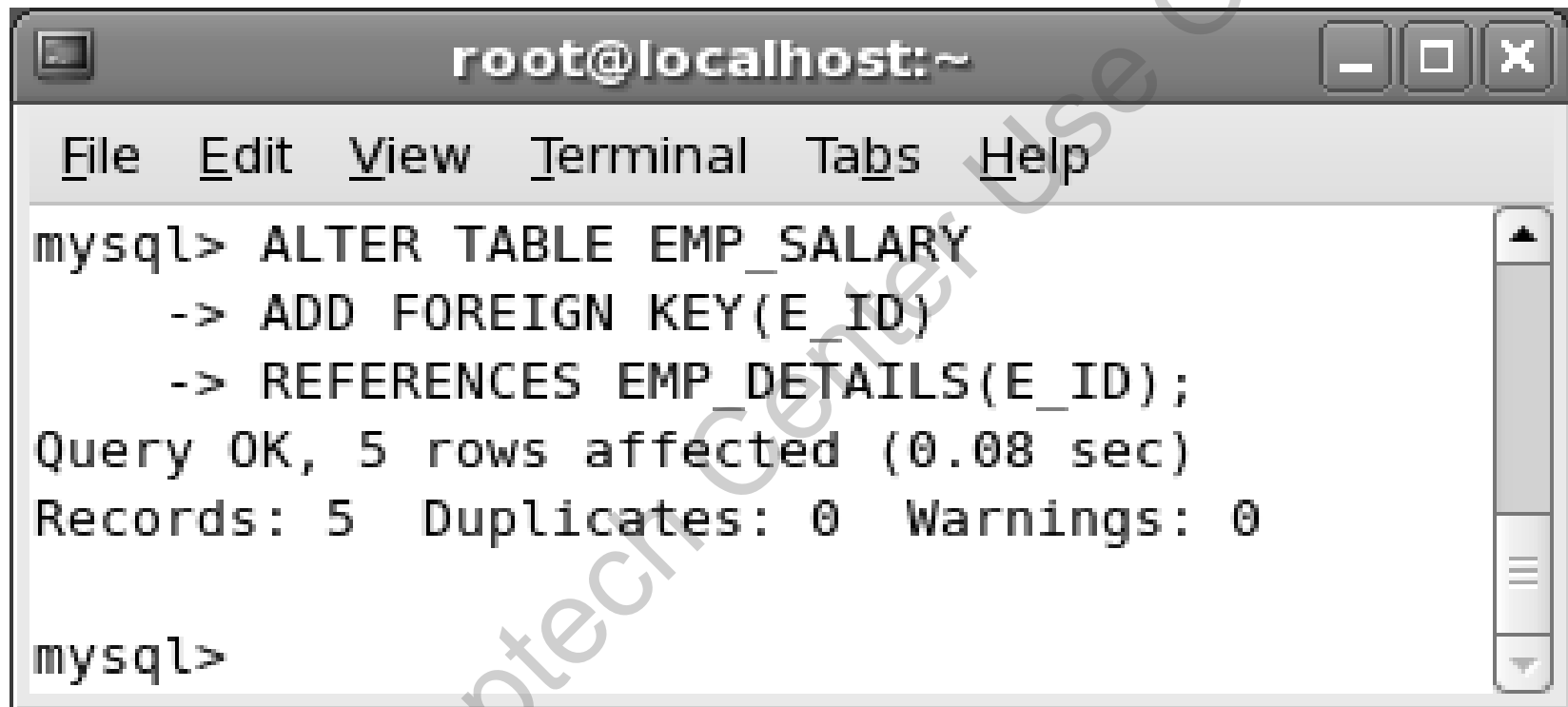
The options for defining a foreign key are listed in the table

Option	Description
ON DELETE	Deletes the child row if the related parent row is deleted
ON UPDATE	Modifies the child row if the related parent row is modified
CASCADE	Delete or update the row in the parent table and automatically delete or update corresponding rows in the child table
SET NULL	Deletes or modifies the row in the parent table and sets the corresponding row in the child table to NULL provided NOT NULL is not specified in the foreign key column
NO ACTION	Prevents a delete or update to the primary key if a foreign key value exists
RESTRICT	Prevents the deletion or updation of row in the parent table if a dependent child row exists

- ◆ To define a foreign key, enter the following command at the command prompt:

```
ALTER TABLE EMP_SALARY  
ADD FOREIGN KEY(E_ID)  
REFERENCES EMP_DETAILS(E_ID);
```

Figure displays the output of the command

A terminal window titled 'root@localhost:~' with standard window controls (minimize, maximize, close). The menu bar includes 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows a MySQL session where the 'ALTER TABLE EMP\_SALARY' command is used to add a foreign key constraint. The output indicates that 5 rows were affected in 0.08 seconds, with no duplicates or warnings.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> ALTER TABLE EMP_SALARY  
-> ADD FOREIGN KEY(E_ID)  
-> REFERENCES EMP_DETAILS(E_ID);  
Query OK, 5 rows affected (0.08 sec)  
Records: 5  Duplicates: 0  Warnings: 0  
  
mysql>
```

- ◆ You can use indexes to improve the search characteristics of the database
- ◆ Indexing enables the search operation on a database to be faster
- ◆ MySQL creates separate files to store and track indexes when you define an index for a table
- ◆ MySQL allows upto 64 indexes for each table
- ◆ Each index allows upto 16 columns to be included



- ◆ The disadvantages of indexing are:
  - ◆ The process of altering data in a database becomes slow
  - ◆ The process such as update, insert, and delete require more time to execute
  - ◆ MySQL requires this additional time to record the changes in the index file as well

- ◆ You can index the field that:
  - ◆ Is used in the WHERE clause of query
  - ◆ Is used in ORDER BY clause of the query
  - ◆ Is having a unique value
- ◆ If you do not specify an index key, MySQL automatically indexes the primary key

- ◆ The syntax to create an index for a table is:

```
CREATE INDEX index_name ON  
tablename (column1, column2, ..., columnN);
```

where,

CREATE INDEX – appends an index to the

tableindex\_name – specifies the name for the index

tablename – specifies the name of the table to create the  
index

column1 – specifies the names of the columns to be indexed  
in the table

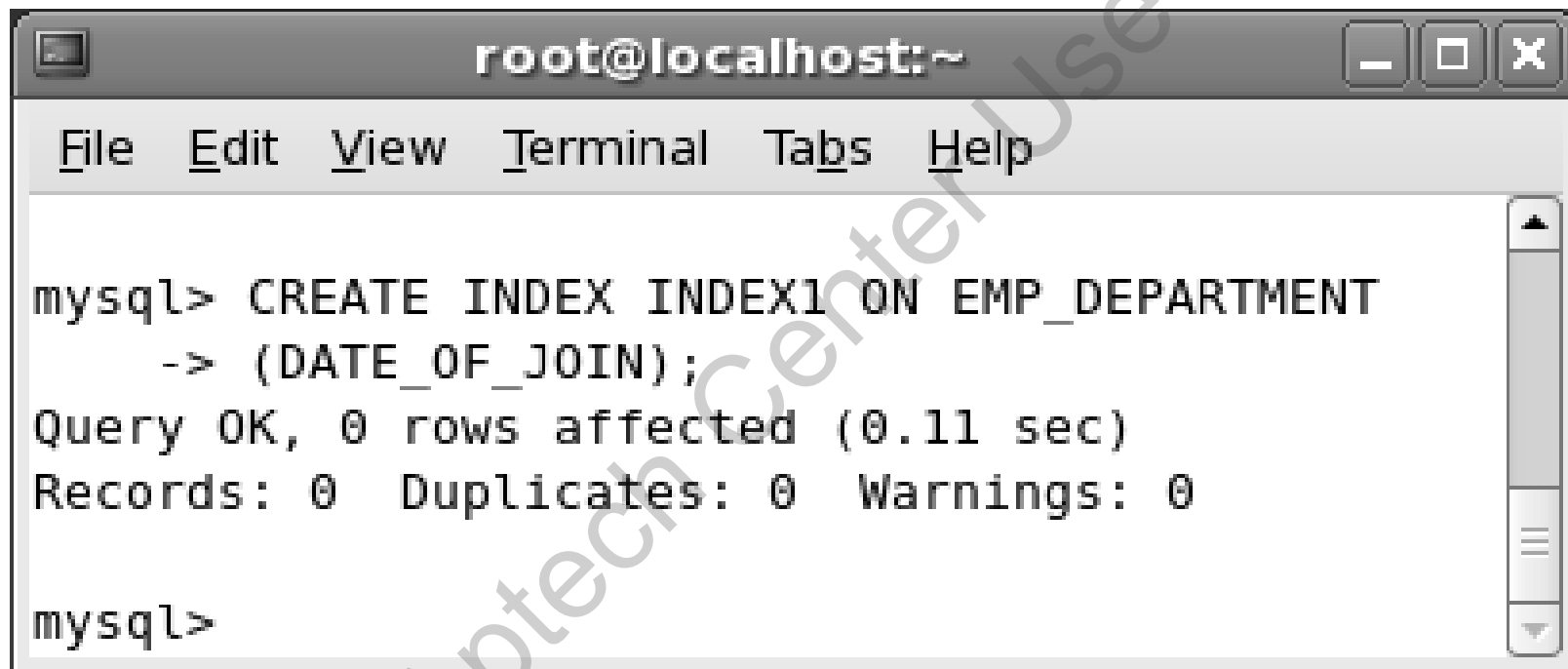
- ◆ To create an index named INDEX1 on the columns ID and NAME while creating a SAMPLE table, enter the following command at the command prompt:

```
CREATE TABLE SAMPLE (ID INT(2) NOT NULL, NAME  
CHAR(10), INDEX INDEX1 (ID, NAME), UNIQUE  
INDEX (ID) );
```

- ◆ You will now create an index on the DATE\_OF\_JOIN field in the EMP\_DEPARTMENT table to search data
- ◆ To create an index, INDEX1 on EMP\_DEPARTMENT, enter the following command at the command prompt:

```
CREATE INDEX INDEX1 ON  
EMP_DEPARTMENT (DATE_OF_JOIN);
```

Figure displays the output of the command

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> CREATE INDEX INDEX1 ON EMP\_DEPARTMENT -> (DATE\_OF\_JOIN);'. The output is: 'Query OK, 0 rows affected (0.11 sec)' and 'Records: 0 Duplicates: 0 Warnings: 0'. 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> CREATE INDEX INDEX1 ON EMP_DEPARTMENT  
-> (DATE_OF_JOIN);  
Query OK, 0 rows affected (0.11 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql>
```

- ◆ MySQL provides the `INSERT` command to add new records into a table
- ◆ MySQL appends the newly inserted record after the last record in the table if the table already contains records
- ◆ The syntax to insert data into a table is:

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE] [INTO] tbl_name  
[(col_name,...)] VALUES ({expression |  
DEFAULT},...), (...), ... [ON DUPLICATE KEY UPDATE  
col_name=expression,...];
```

where,

`INSERT` – adds new row to an existing table

`tbl_name` – specifies the name of the table where MySQL will append the row

`col_name` – specifies the name of the column where MySQL will add the data

`VALUES` – defines the data that will be added to the table

The options for the `INSERT` command are listed in table

Option	Description
<code>LOW_PRIORITY</code>	Postpones addition of records till all operations (reading) from the client have completed
<code>DELAYED</code>	Delays the insertion against all incoming <code>SELECT</code> statements
<code>IGNORE</code>	Overrides the error conditions for columns and adds the record to the table
<code>INTO</code>	Specifies the name of the table into which the columns are to be inserted
<code>col_name</code>	Specifies the name of the columns to insert values
<code>expression</code>	Includes a mathematical expression for a column
<code>DEFAULT</code>	Stores the default value into the specified columns
<code>ON DUPLICATE KEY UPDATE col_name=expression</code>	Updates the existing record if the addition of a new record creates a duplicate value in the <code>PRIMARY KEY</code> column

- ◆ MySQL allows you to insert values for specific column names
- ◆ All other columns, if not specified, will contain a `NULL` value
- ◆ You have created columns to contain default data, then you can use the `DEFAULT` clause of the `INSERT` command to specify a particular column name and the value for that column
- ◆ An error occurs if the column does not have a default value



- ◆ To insert rows in EMP\_DEPARTMENT, enter the following command at the command prompt:

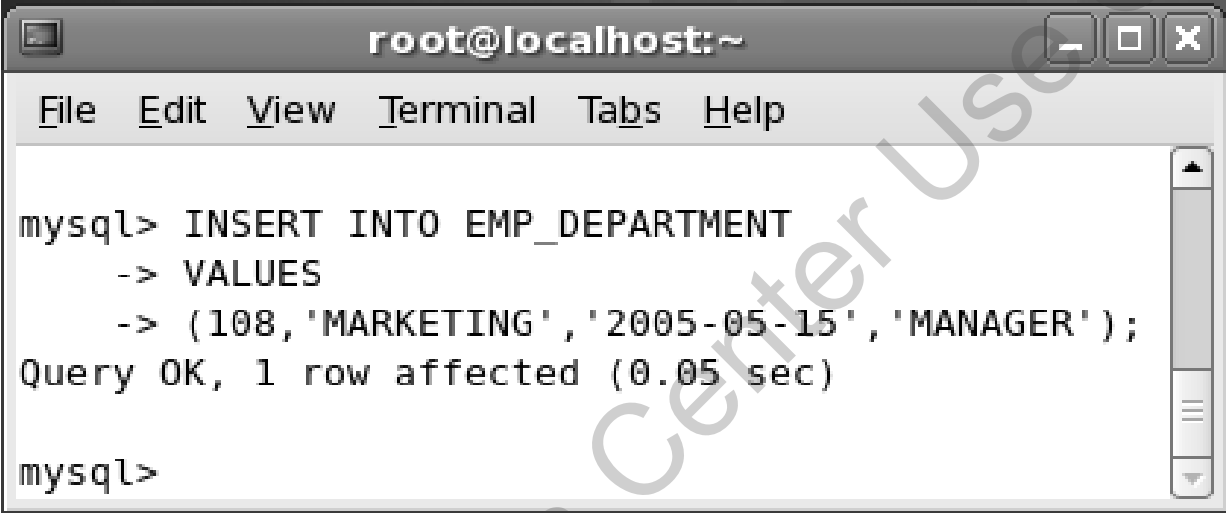
```
INSERT INTO EMP_DEPARTMENT (E_ID, D_NAME,  
DATE_OF_JOIN, DESIGNATION) VALUES (108, 'MARKETING',  
'2000-05-15', 'MANAGER');
```

- ◆ You can also enter the following command at the command prompt:

```
INSERT INTO EMP_DEPARTMENT VALUES (108,  
'MARKETING', '2000-05-15 ', 'MANAGER');
```

- ◆ This command will work only if values for all the columns are specified following the VALUES keyword

Figure displays the output of the command

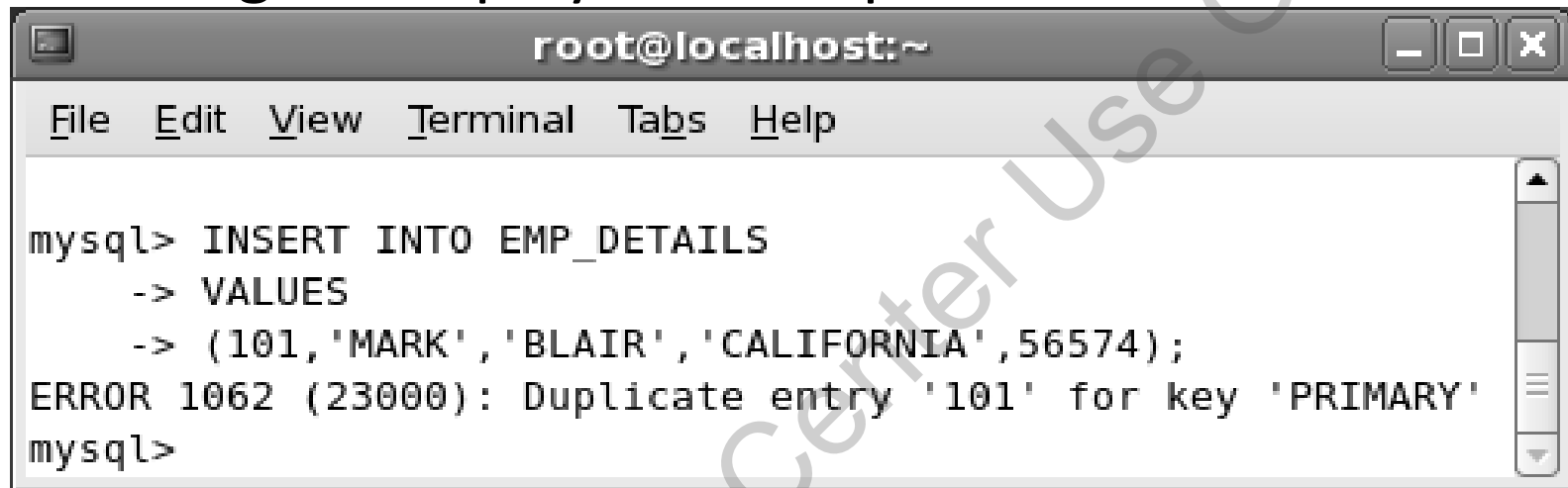
A screenshot of a terminal window titled 'root@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows a MySQL command sequence: 'mysql> INSERT INTO EMP\_DEPARTMENT' followed by a multi-line input '-> VALUES' and '-> (108, 'MARKETING', '2005-05-15', 'MANAGER');'. The output is 'Query OK, 1 row affected (0.05 sec)'. The prompt 'mysql>' is shown again at the bottom. A large, light gray watermark 'For Aptech Center Use Only' is diagonally across the image.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
  
mysql> INSERT INTO EMP_DEPARTMENT  
-> VALUES  
-> (108, 'MARKETING', '2005-05-15', 'MANAGER');  
Query OK, 1 row affected (0.05 sec)  
  
mysql>
```

- ◆ You had earlier defined the primary key of EMP\_DEPARTMENT table as NOT NULL and AUTO\_INCREMENT
- ◆ Now you can try to insert a row having the E\_ID that already exists, by entering the following command at the command prompt:

```
INSERT INTO EMP_DETAILS VALUES  
(101, 'MARK', 'BLAIR', 'CALIFORNIA', 56574);
```

Figure displays the output of the command



A terminal window titled 'root@localhost:~' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows a MySQL command sequence: 'mysql> INSERT INTO EMP\_DETAILS', followed by a prompt '-> VALUES' and another prompt '-> (101, 'MARK', 'BLAIR', 'CALIFORNIA', 56574);'. Below this, an error message is displayed: 'ERROR 1062 (23000): Duplicate entry '101' for key 'PRIMARY''. The prompt 'mysql>' appears again 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  
-> (101, 'MARK', 'BLAIR', 'CALIFORNIA', 56574);  
ERROR 1062 (23000): Duplicate entry '101' for key 'PRIMARY'  
mysql>
```

- ◆ A different form of the `INSERT` command enables you to set the column to a specific value
- ◆ The syntax is as follows:

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
[INTO] tbl_name SET col_name = {expression | DEFAULT}, ...
[ON DUPLICATE KEY UPDATE col_name=expression, ...];
```

where,

`INSERT` – adds data to the table

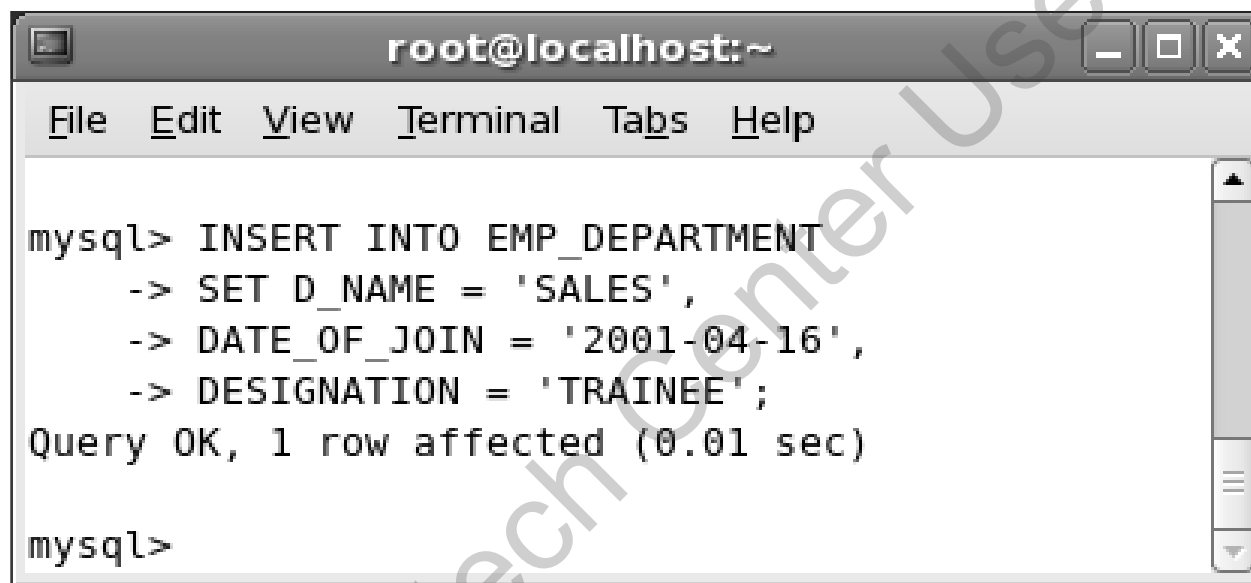
`tbl_name` – specifies the name of the table

`SET` – specifies the column names explicitly to which the values will be added

- ◆ To insert rows in EMP\_DEPARTMENT, enter the following command at the command prompt:

```
INSERT INTO EMP_DEPARTMENT SET D_NAME = 'SALES',  
DATE_OF_JOIN = '2001-04-16', DESIGNATION= 'TRAINEE';
```

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 a MySQL prompt 'mysql>' followed by an 'INSERT INTO EMP\_DEPARTMENT' command. The command is split into three lines: 'SET D\_NAME = 'SALES'', 'DATE\_OF\_JOIN = '2001-04-16'', and 'DESIGNATION = 'TRAINEE';'. The output of the command is 'Query OK, 1 row affected (0.01 sec)'. The prompt 'mysql>' is shown again at the bottom. A large, diagonal watermark 'For Aptech Center Use Only' is overlaid on the image.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
  
mysql> INSERT INTO EMP_DEPARTMENT  
-> SET D_NAME = 'SALES',  
-> DATE_OF_JOIN = '2001-04-16',  
-> DESIGNATION = 'TRAINEE';  
Query OK, 1 row affected (0.01 sec)  
  
mysql>
```



- ◆ The `INSERT` command enables you to add data to a table returned from a `SELECT` query

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE] [INTO]  
tbl_name [(col_name,...)] SELECT ...;
```

where,

`INSERT` – adds data into the table

`tbl_name` – specifies the name of the table to add data

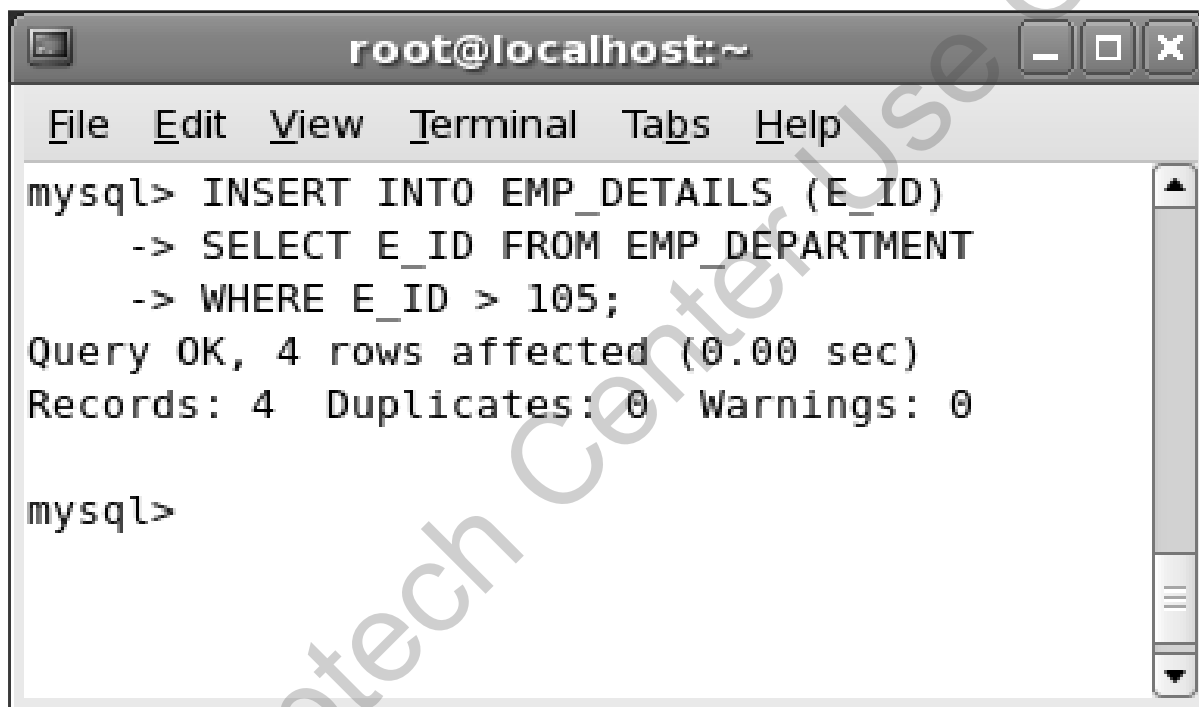
`col_name` – specifies the name of the column

`SELECT` – defines the data to be added in the table specified in the `tbl_name` clause

- ◆ To insert E\_ID in the EMP\_DETAILS table where the E\_ID is retrieved from the EMP\_DEPARTMENT table using the SELECT command, enter the following command at the command prompt:

```
INSERT INTO EMP_DETAILS (E_ID)
SELECT E_ID FROM EMP_DEPARTMENT
WHERE E_ID > 105;
```

Figure displays the output of the command



```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> INSERT INTO EMP_DETAILS (E_ID)  
-> SELECT E_ID FROM EMP_DEPARTMENT  
-> WHERE E_ID > 105;  
Query OK, 4 rows affected (0.00 sec)  
Records: 4  Duplicates: 0  Warnings: 0  
  
mysql>
```

- ◆ MySQL provides the UPDATE command to change or modify data within a table
- ◆ The syntax to update a table is:

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name SET  
column=value,... [WHERE clause][LIMIT n];
```

where,

UPDATE – modifies the column data

tbl\_name – specifies the name of the table to modify

SET – defines the value to change

column – specifies the name of the column

value – defines the new value to be entered for the column

The options for the `UPDATE` command are listed in table

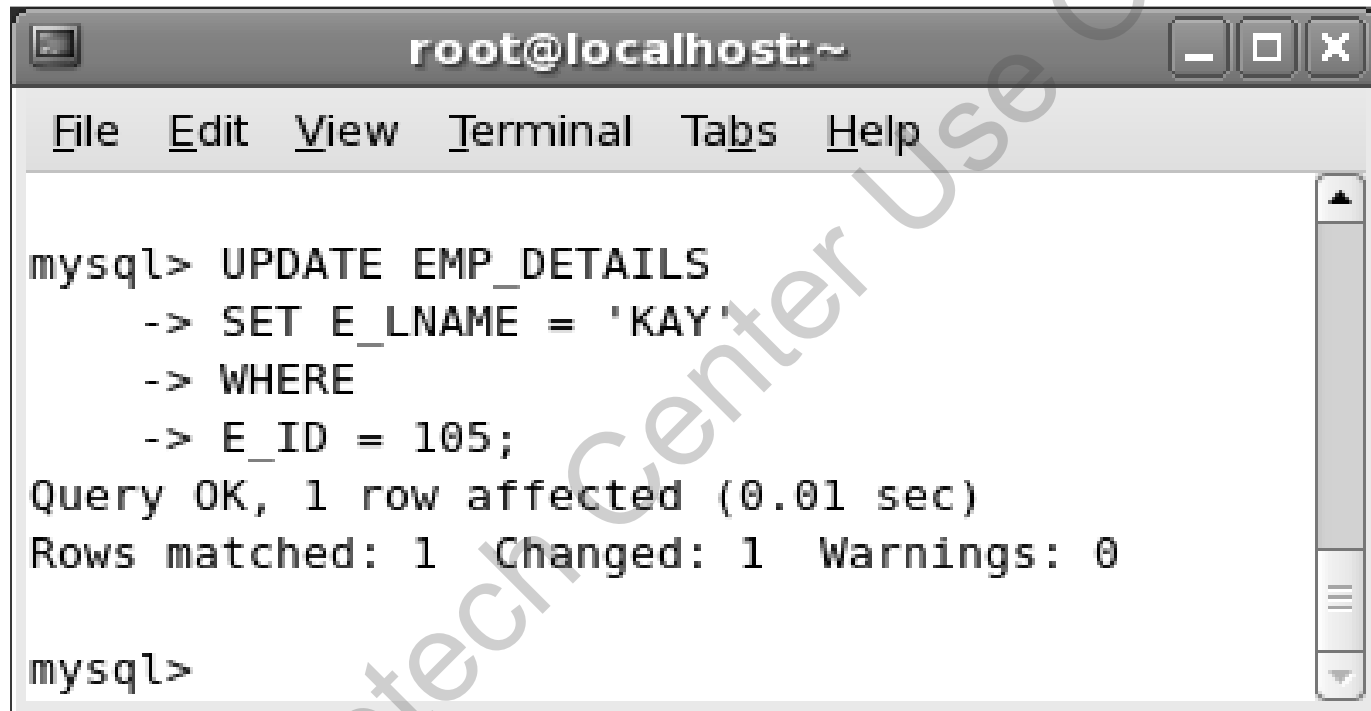
Option	Description
<code>LOW_PRIORITY</code>	Assigns a low priority to the update against all other commands
<code>IGNORE</code>	Updates a row even if there is an error
<code>WHERE</code>	Specifies the rows to be updated based on the specific condition
<code>LIMIT</code>	Specifies a limit on the number of rows that can be updated

- ◆ To update a row in EMP\_DETAILS table, enter the following command at the command prompt

```
UPDATE EMP_DETAILS SET E_LNAME = 'KAY' WHERE E_ID =  
105;
```

For Apteck Center Use Only

Figure displays the output of the command

A terminal window titled 'root@localhost:~' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal shows a MySQL session where the 'UPDATE EMP\_DETAILS' command is executed. The output indicates that 1 row was affected in 0.01 seconds, with 1 row matched and 1 row changed, and no warnings. The prompt 'mysql>' is shown at the end of the session.

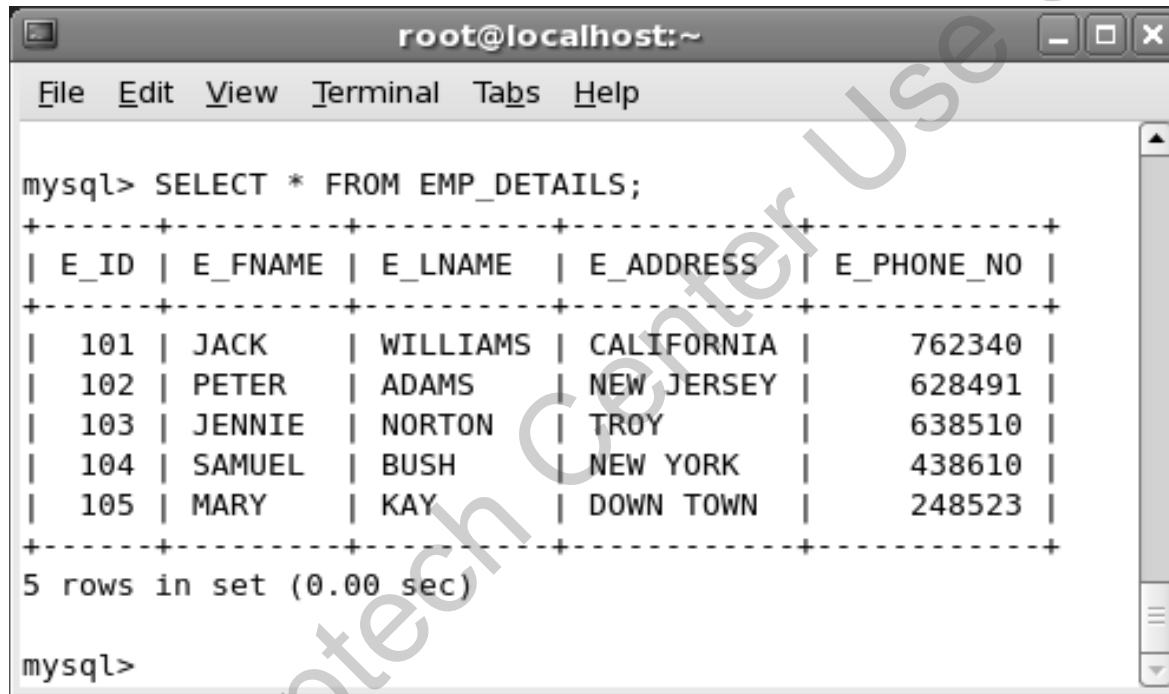
```
root@localhost:~  
File Edit View Terminal Tabs Help  
  
mysql> UPDATE EMP_DETAILS  
-> SET E_LNAME = 'KAY'  
-> WHERE  
-> E_ID = 105;  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1  Changed: 1  Warnings: 0  
  
mysql>
```

- ◆ When you include the `ORDER BY` clause in an `UPDATE` command, MySQL will update the rows in the order specified in the `ORDER BY` clause
- ◆ You can include the `ORDER BY` clause in an `UPDATE` command to modify records in a table sorted on specific conditions
- ◆ You can now use the `SELECT` command to view the updated data in the table
- ◆ To view the updated results, 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 5 rows and 6 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	KAY	DOWN TOWN	248523

Below the table, the terminal shows '5 rows in set (0.00 sec)' and the prompt 'mysql>'.

- ◆ To update a row in EMP\_DETAILS table, enter the following command at the command prompt

```
UPDATE EMP_DETAILS SET E_LNAME = 'KAY' WHERE E_ID =  
105;
```

For Aptech Center Use Only

- ◆ MySQL provides the `REPLACE` command to substitute data in the table
- ◆ The `REPLACE` command is similar to the `INSERT` command, except that it replaces the old value of a column with a new value
- ◆ The `REPLACE` command will delete the old row only if the new row satisfies the conditions of the column definitions
- ◆ If the value in the `REPLACE` command does not match the existing record then MySQL executes it as an `INSERT` statement

- ◆ The syntax for replacing data in a table is:

```
REPLACE [DELAYED|LOW_PRIORITY] INTO tbl_name  
[(column,...)] VALUES (value,...);
```

where,

REPLACE – substitutes the column data

tbl\_name – specifies the name of the table to modify

value – defines the new value for the column

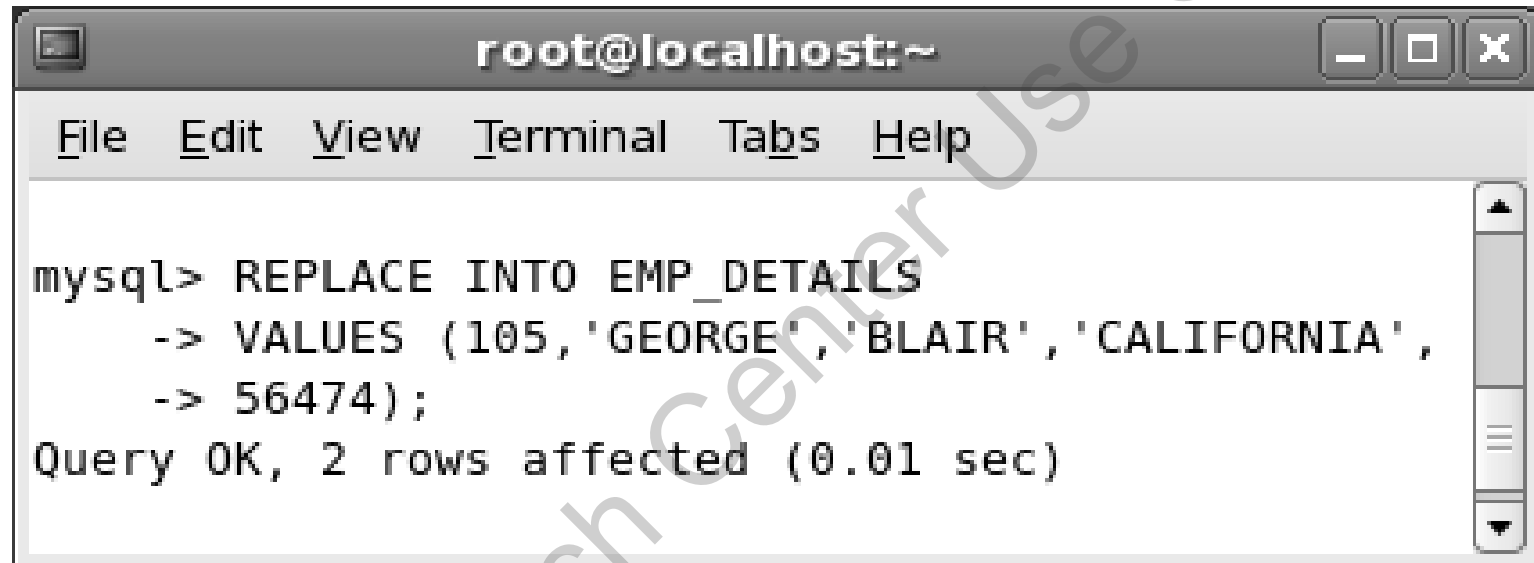
The options of the REPLACE command are listed in table

Option	Description
DELAYED	Postpones the execution of the command until no client is accessing the table
LOW_PRIORITY	Specifies the rows to be inserted into a buffer
column	Specifies the column names for which the values are to be replace
value	Defines the new values for the columns

- ◆ To replace a row in EMP\_DETAILS, enter the following command at the command prompt:

```
REPLACE INTO EMP_DETAILS  
VALUES (105, 'GEORGE', 'BLAIR', 'CALIFORNIA', 56474  
) ;
```

Figure displays the output of the command



A terminal window titled 'root@localhost:~' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal displays the following text:

```
mysql> REPLACE INTO EMP_DETAILS  
      -> VALUES (105, 'GEORGE', 'BLAIR', 'CALIFORNIA',  
      -> 56474);  
Query OK, 2 rows affected (0.01 sec)
```

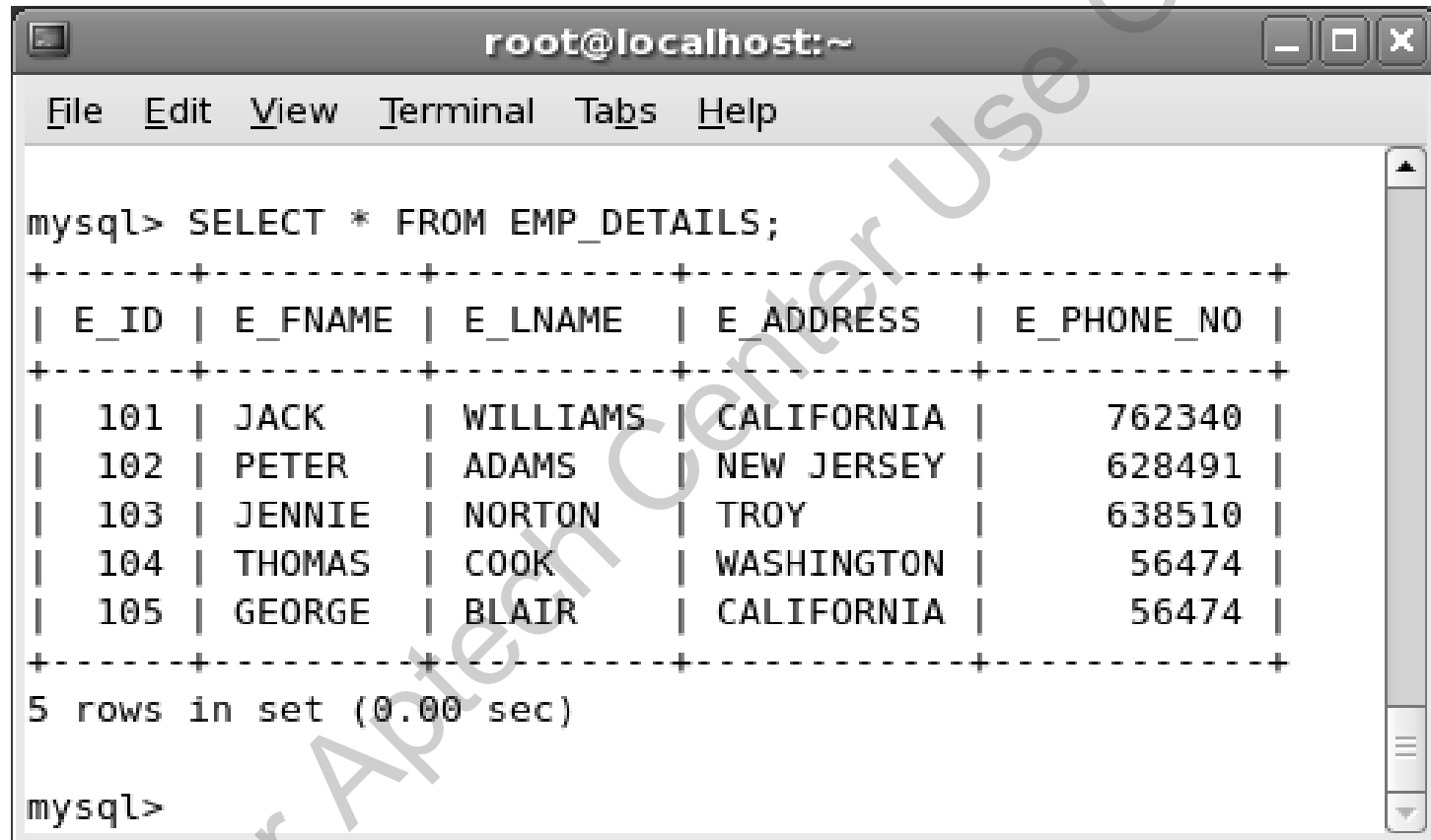
The terminal window has standard window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

- ◆ MySQL displays the number of rows affected by the REPLACE query
- ◆ If MySQL returns the number of rows affected as one for a single row modification, it means that a row has been added and existing rows have not been deleted
- ◆ When the number of rows affected is more than one, it means that an existing record has been deleted and a new record has been added to the table
- ◆ To view the data in the EMP\_DETAILS table, enter the following command at the command prompt:

```
SELECT * FROM EMP_DETAILS;
```



Figure displays the output of the command



A terminal window titled 'root@localhost:~' displays the output of a MySQL query. The query is 'mysql> SELECT \* FROM EMP\_DETAILS;'. The output is a table with 5 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	THOMAS	COOK	WASHINGTON	56474
105	GEORGE	BLAIR	CALIFORNIA	56474

Below the table, the text '5 rows in set (0.00 sec)' is displayed. The prompt 'mysql>' is shown at the bottom of the terminal window.

- ◆ You can also use the `SELECT` query within a `REPLACE` command. The syntax to include a `SELECT` query within a `REPLACE` command is:

```
REPLACE [DELAYED|LOW_PRIORITY] INTO tbl_name  
[(column,...)] SELECT select_clause;
```

where,

`REPLACE` – substitutes the column data

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

`SELECT` – executes the select query specified in the clause

- ◆ The REPLACE command also enables you to specify the column names and the values together
- ◆ MySQL assigns a NULL value to the columns if you fail to specify a value in the REPLACE command

```
REPLACE [DELAYED|LOW_PRIORITY] INTO tbl_name SET  
column=value, column=value,...;
```

where,

REPLACE – substitutes the column data

tbl\_name – specifies the name of the table to modify

SET – defines the new value to be entered in the column

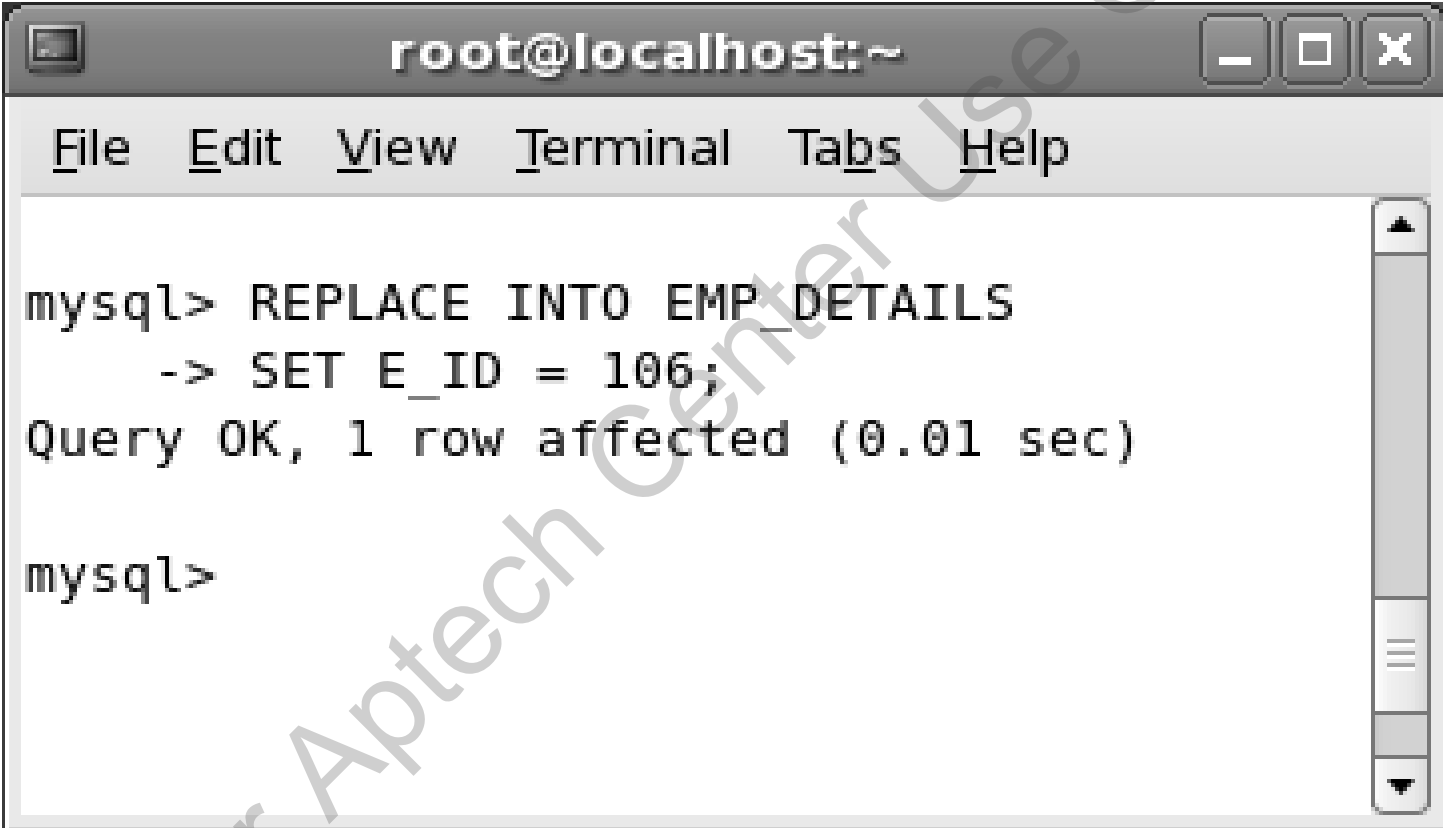
column – specifies the name of the column

value – defines the new value for the specified column

- ◆ For example, to add a new record in the EMP\_DETAILS table, where you only specify the E\_ID, enter the following command at the command prompt:

```
REPLACE INTO EMP_DETAILS  
SET E_ID = 106;
```

Figure displays the output of the command



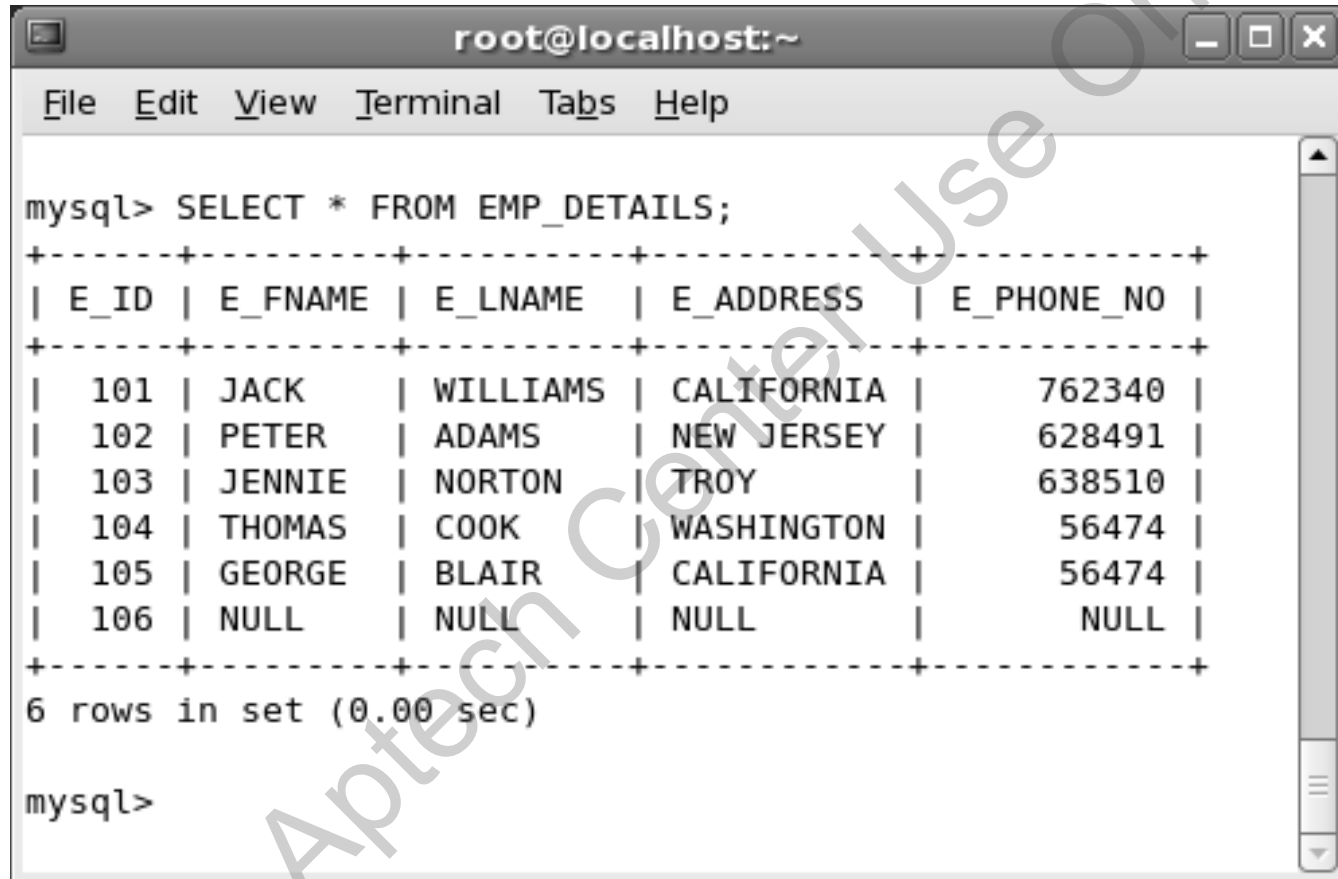
A terminal window titled 'root@localhost:~' with standard window controls (minimize, maximize, close). The menu bar includes 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows a MySQL prompt 'mysql>' followed by the command 'REPLACE INTO EMP\_DETAILS' and an indented line '-> SET E\_ID = 106;'. The output is 'Query OK, 1 row affected (0.01 sec)'. The prompt 'mysql>' appears again on the next line. A large, light-gray watermark 'For Aptech Center Use Only' is oriented diagonally across the terminal window.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
  
mysql> REPLACE INTO EMP_DETAILS  
        -> SET E_ID = 106;  
Query OK, 1 row affected (0.01 sec)  
  
mysql>
```

- ◆ In this command, you have specified a value only for the `E_ID` field
- ◆ MySQL will assign `NULL` values to the other columns in the table for this record
- ◆ To view the `NULL` values for this record, enter the following command at the command prompt:

```
SELECT * FROM EMP_DETAILS;
```

Figure displays the output of the command



A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the execution of the MySQL command 'SELECT \* FROM EMP\_DETAILS;'. 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	THOMAS	COOK	WASHINGTON	56474
105	GEORGE	BLAIR	CALIFORNIA	56474
106	NULL	NULL	NULL	NULL

Below the table, the terminal displays '6 rows in set (0.00 sec)' and the prompt 'mysql>'.

- ◆ MySQL provides the `DELETE` command to remove rows from a table
- ◆ You can choose to remove selected rows of a table or the entire table
- ◆ The syntax to delete rows from a table is:

```
DELETE [LOW_PRIORITY|QUICK] [IGNORE] FROM tbl_name [WHERE  
clause] [ORDER BY column,...] [LIMIT n];
```

where,

`DELETE` – removes data from the table

`tbl_name` – specifies the name of the table from which rows are to be deleted

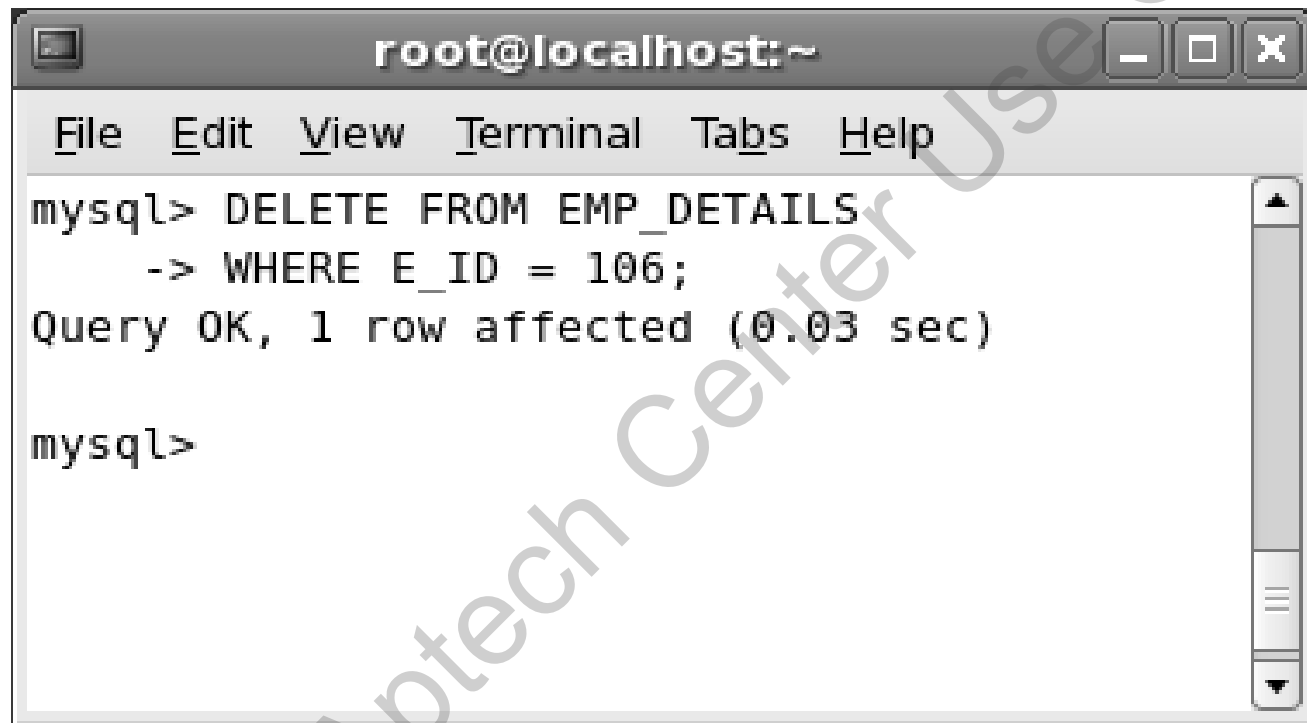


The options for the `DELETE` command are provided in table

Option	Description
<code>LOW_PRIORITY</code>	Postpones deletion until all other operations on the table have been completed by the client such as reading
<code>QUICK</code>	Suspends the merging of indexes during execution
<code>IGNORE</code>	Ignores error while deleting a row
<code>WHERE</code>	Deletes rows that satisfy the specified conditions
<code>ORDER BY</code>	Deletes data in the order specified
<code>LIMIT n</code>	Specifies the maximum number of rows to be deleted in a single attempt

- ◆ If you use the `DELETE` command without the `WHERE` clause, the entire table will be deleted
- ◆ If you specify the `WHERE` clause in the `DELETE` command, the system prompts a message displaying the number of rows that will be deleted
- ◆ If no `WHERE` clause is used, all the rows are deleted
- ◆ MySQL returns a `'0'` value, when the entire table is deleted because MySQL cannot identify the number of rows to be deleted

Figure displays the output of the command



The image shows 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 the following sequence of commands and output:

```
mysql> DELETE FROM EMP_DETAILS
      -> WHERE E_ID = 106;
Query OK, 1 row affected (0.03 sec)

mysql>
```

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

- ◆ You use the `SELECT` command to retrieve data from a database
- ◆ MySQL displays the `SELECT` command result in the order in which the data is inserted in the table
- ◆ You can also sort the output on more than one column in ascending or descending order
- ◆ You will use the `ORDER BY` clause with `SELECT` command to sort the data
- ◆ The use of `DISTINCT` keyword causes only one row of data to be displayed for every group of rows that is identical

- ◆ The syntax for ORDER BY is:

```
SELECT column_name FROM tbl_name ORDER BY column  
[ASC|DESC] [,column2[ASC|DESC],...];
```

where,

SELECT – retrieves data from the table

column\_name – specifies the name for the column to retrieve data from

tbl\_name – specifies the name of the table that contains the column and data

ORDER BY – sorts the column data in the specified order

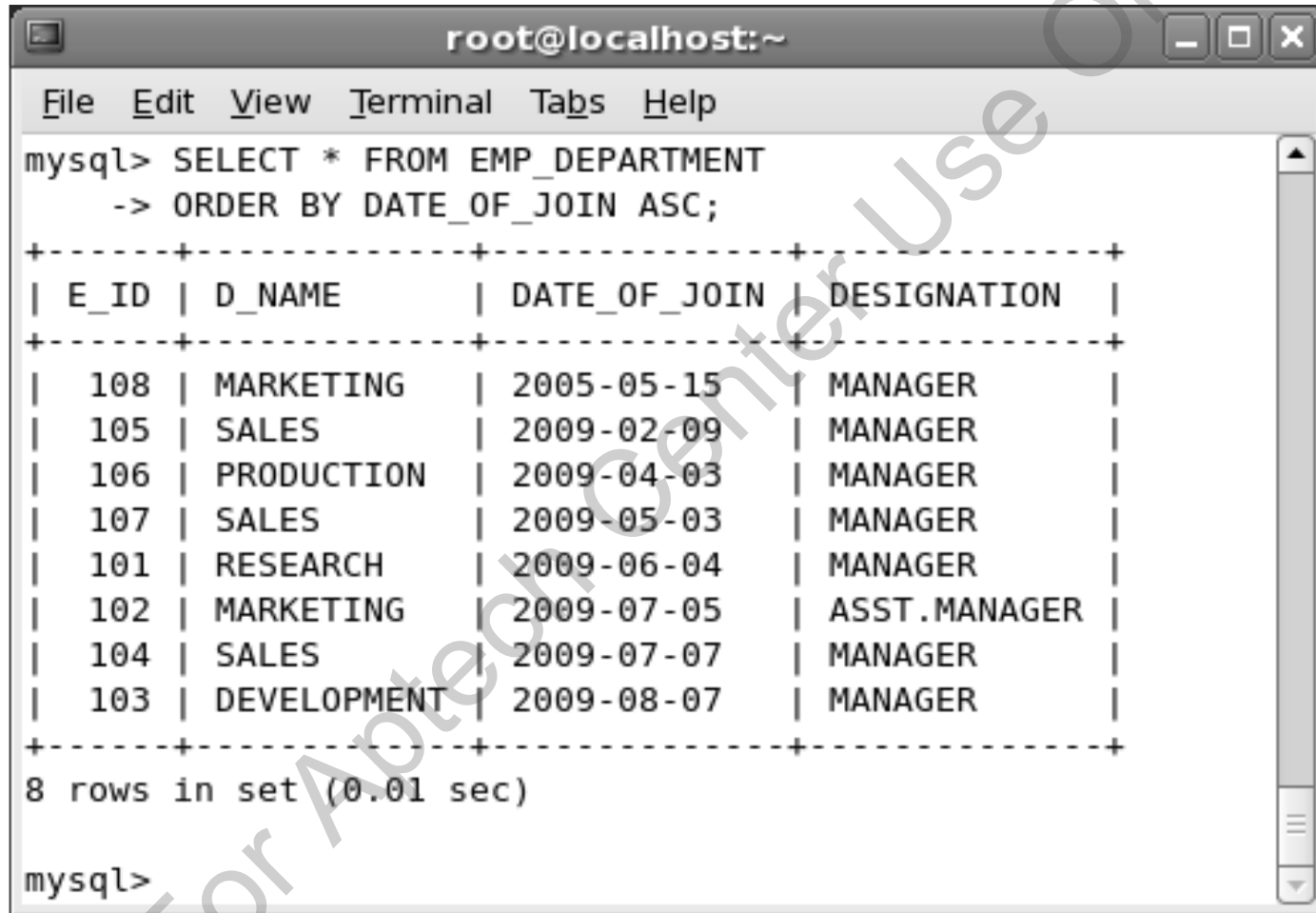
column – specifies the name of the column

ASC|DESC – specifies the sort direction for data. MySQL supports types of sorting

- ◆ To organize the output of EMP\_DEPARTMENT table on the DATE\_OF\_JOIN field in an ascending order, enter the following command at the command prompt:

```
SELECT * FROM EMP_DEPARTMENT ORDER BY DATE_OF_JOIN  
ASC;
```

Figure displays the output of the command



The screenshot shows a MySQL terminal window titled 'root@localhost:~'. The command entered is 'mysql> SELECT \* FROM EMP\_DEPARTMENT -> ORDER BY DATE\_OF\_JOIN ASC;'. The output is a table with 4 columns: E\_ID, D\_NAME, DATE\_OF\_JOIN, and DESIGNATION. The data is sorted by DATE\_OF\_JOIN in ascending order. Below the table, it says '8 rows in set (0.01 sec)'. The prompt 'mysql>' is visible at the bottom.

```
mysql> SELECT * FROM EMP_DEPARTMENT
-> ORDER BY DATE_OF_JOIN ASC;
+-----+-----+-----+-----+
| E_ID | D_NAME      | DATE_OF_JOIN | DESIGNATION |
+-----+-----+-----+-----+
| 108 | MARKETING   | 2005-05-15   | MANAGER     |
| 105 | SALES       | 2009-02-09   | MANAGER     |
| 106 | PRODUCTION  | 2009-04-03   | MANAGER     |
| 107 | SALES       | 2009-05-03   | MANAGER     |
| 101 | RESEARCH    | 2009-06-04   | MANAGER     |
| 102 | MARKETING   | 2009-07-05   | ASST.MANAGER |
| 104 | SALES       | 2009-07-07   | MANAGER     |
| 103 | DEVELOPMENT | 2009-08-07   | MANAGER     |
+-----+-----+-----+-----+
8 rows in set (0.01 sec)


mysql>
```

- ◆ You will now sort the data in the EMP\_DETAILS table on the E\_FNAME and E\_LNAME columns
- ◆ You will sort data from the E\_FNAME column in the descending order
- ◆ To sort the output of EMP\_DETAILS table on the E\_FNAME field in the descending order and then by E\_LNAME field, enter the following command at the command prompt:

```
SELECT * FROM EMP_DETAILS ORDER BY E_FNAME DESC,  
E_LNAME;
```



Figure displays the output of the command

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows a MySQL query: 'mysql> SELECT \* FROM EMP\_DETAILS -> ORDER BY E\_FNAME DESC,E\_LNAME;'. The output is a table with 5 rows, sorted by E\_FNAME and E\_LNAME in descending order. The table has columns: E\_ID, E\_FNAME, E\_LNAME, E\_ADDRESS, and E\_PHONE\_NO. The rows are: (104, THOMAS, COOK, WASHINGTON, 56474), (102, PETER, ADAMS, NEW JERSEY, 628491), (103, JENNIE, NORTON, TROY, 638510), (101, JACK, WILLIAMS, CALIFORNIA, 762340), and (105, GEORGE, BLAIR, CALIFORNIA, 56474). Below the table, it says '5 rows in set (0.00 sec)'. The prompt 'mysql>' is at the bottom.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
  
mysql> SELECT * FROM EMP_DETAILS  
-> ORDER BY E_FNAME DESC,E_LNAME;  
  
+-----+-----+-----+-----+-----+  
| E_ID | E_FNAME | E_LNAME | E_ADDRESS | E_PHONE_NO |  
+-----+-----+-----+-----+-----+  
| 104 | THOMAS | COOK | WASHINGTON | 56474 |  
| 102 | PETER | ADAMS | NEW JERSEY | 628491 |  
| 103 | JENNIE | NORTON | TROY | 638510 |  
| 101 | JACK | WILLIAMS | CALIFORNIA | 762340 |  
| 105 | GEORGE | BLAIR | CALIFORNIA | 56474 |  
+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)  
  
mysql>
```

- ◆ MySQL provides the GROUP BY command to group rows with similar values for a specific column into a single row
- ◆ This will enable you to execute tasks on these grouped rows
- ◆ The syntax for grouping data is:

```
SELECT column_name FROM tbl_name GROUP BY column  
[,column2...];
```

where,

SELECT – retrieves data from the table

column\_name – specifies the column name to retrieve data from

tbl\_name – specifies the name of the table that contains the column and data

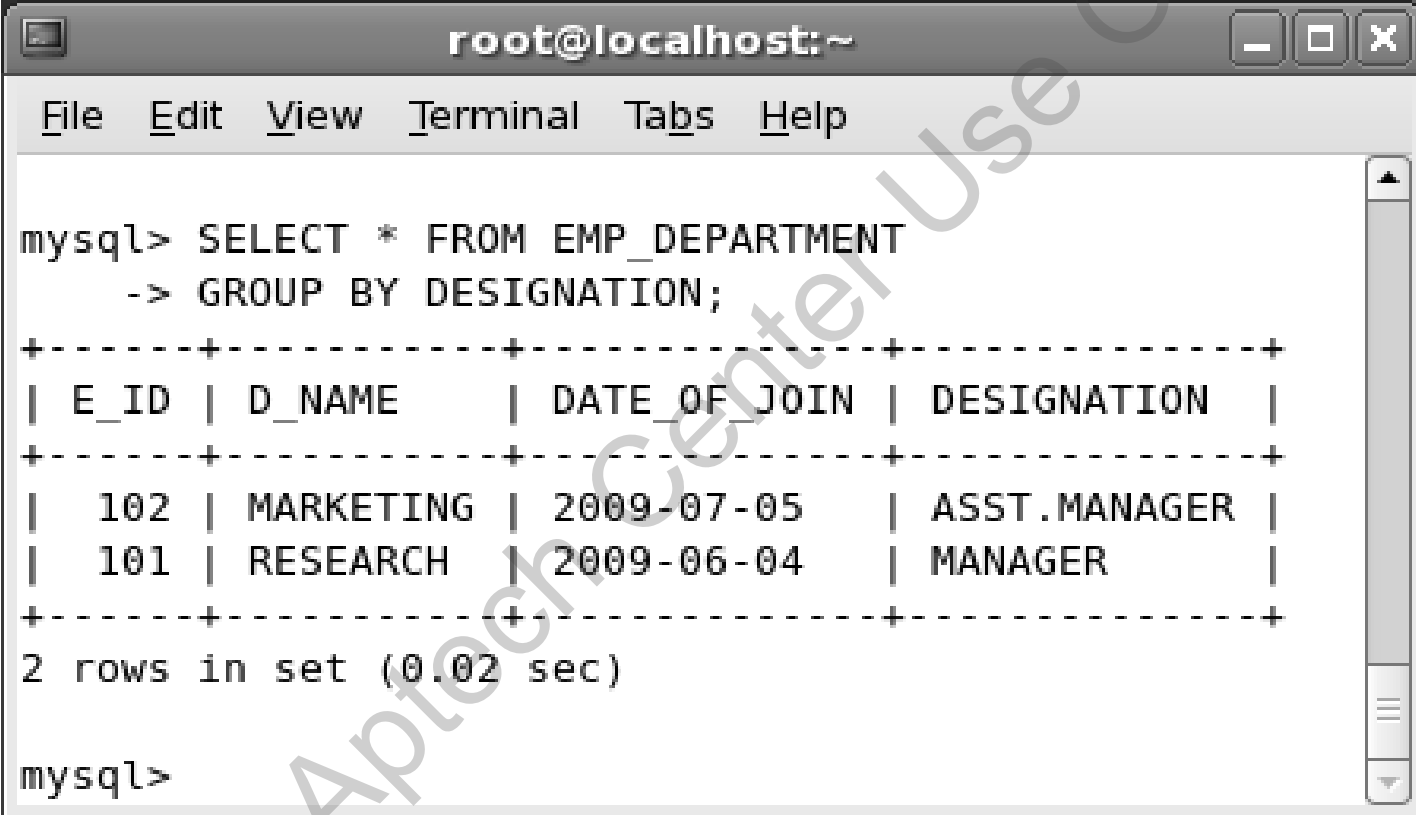
GROUP BY – categorizes the data

column – specifies the name of the column in the table

- ◆ You can group the employees based on their designations from the EMP\_DEPARTMENT table on the DESIGNATION column
- ◆ To group the output of EMP\_DEPARTMENT table on the DESIGNATION field, enter the following command at the command prompt:

```
SELECT * FROM EMP_DEPARTMENT GROUP BY  
DESIGNATION;
```

Figure displays the output of the command



A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows a MySQL command and its output. The command is 'SELECT \* FROM EMP\_DEPARTMENT GROUP BY DESIGNATION;'. The output is a table with 4 columns: E\_ID, D\_NAME, DATE\_OF\_JOIN, and DESIGNATION. It contains 2 rows of data. Below the table, it says '2 rows in set (0.02 sec)'. The prompt 'mysql>' is visible at the bottom.

```
mysql> SELECT * FROM EMP_DEPARTMENT
-> GROUP BY DESIGNATION;
+-----+-----+-----+-----+
| E_ID | D_NAME   | DATE_OF_JOIN | DESIGNATION |
+-----+-----+-----+-----+
| 102  | MARKETING | 2009-07-05   | ASST.MANAGER |
| 101  | RESEARCH  | 2009-06-04   | MANAGER      |
+-----+-----+-----+-----+
2 rows in set (0.02 sec)

mysql>
```

- ◆ Keys help to identify a record in the table
- ◆ A primary key uniquely identifies the records in a table
- ◆ A foreign key in a table must have a corresponding primary key in a different table
- ◆ A primary key and a foreign key can be defined while creating a table
- ◆ The ALTER TABLE command enables to define keys in a table
- ◆ The INSERT command allows you to add a row in a table
- ◆ The UPDATE command allows you to modify the values present in the columns of a table

- ◆ MySQL enables you to substitute values in a column using the REPLACE command
- ◆ MySQL provides the DELETE command to remove unwanted or obsolete rows in a table
- ◆ The ORDER BY clause can be used with the SELECT command to sort the data
- ◆ Grouping categorizes rows with similar values for a specific column into a single row to execute combined data processing operations