Dr. MPS GROUP OF INSTITUTIONS
COLLEGE OF BUSINESS STUDIES
UGC RECOGNIZED AND AICTE APPROVED COLLEGE

Practical File
FOR
DBMS

BCA IIIrd Semester

Second Year (Session: 2023-24)

Submitted by: **Syed mohammad hasnain**
Roll no.: **2202675010135**

Submitted By:
Syed mohammad hasnain

Submitted To
Mrs. Anupama Malhotra

Signature ___________

Delhi Highway, Near Naveen Sabji Mandi, Sikandra,
Agra, Uttar Pradesh, 282007

TABLE OF CONTENT

Introduction To Oracle

What is Oracle?

Oracle database is a relational database management system (RDBMS) from Oracle Corporation. This article will explain a complete overview of the Oracle database, features, history, and editions. Before discussing the oracle, we will first need to know about the database.

What is a database?

A database refers to the **organized collection of structured data** stored electronically in a device. It allows us to access, manage, and find relevant information frequently. The flat file structure was extensively used to store data before the database system was invented. The relational database approach becomes popular in comparison to the flat file model because it eliminates redundant data. **For example**, suppose we have an employee and contact information stored in the same file. In such a case, the employees with multiple contacts will show up in many rows.

The RDBMS system manages the relational data. Oracle Database is the most famous relational database system (RDBMS) because it shares the largest part of a market among other relational databases. Some other popular relational databases are MySQL, DB2, SQL Server, PostgreSQL, etc.

What is the Oracle database?

Oracle database is a relational database management system. It is also called **OracleDB**, or simply **Oracle**. It is produced and marketed by **Oracle Corporation**. It was created in **1977** by **Lawrence Ellison** and other engineers. It is one of the most popular relational database engines in the IT market for storing, organizing, and retrieving data.

Oracle database was the first DB that designed for **enterprise grid computing** and data warehousing. Enterprise grid computing provides the most flexible and cost-effective way to manage information and applications. It uses SQL queries as a language for interacting with the database.

Editions of Oracle database

Oracle database is compatible with a wide range of platforms such as Windows, UNIX, Linux, and macOS. It supports several operating systems like IBM AIX, HP-UX, Linux, Microsoft Windows Server, Solaris, SunOS, macOS, etc. In the late **1990s**, Oracle began supporting open platforms like GNU/Linux.

The following is a list of Oracle database editions in order of priority:

- **Enterprise Edition:** It is the most robust and secure edition. It offers all features, including superior performance and security.
- **Standard Edition:** It provides the base functionality for users that do not require Enterprise Edition's robust package.
- **Express Edition (XE):** It is the lightweight, free and limited Windows, and Linux edition.
- **Oracle Lite:** It is designed for mobile devices.
- **Personal Edition:** It's comparable to the Enterprise Edition but without the Oracle Real Application Clusters feature.

Introduction To Oracle

The Oracle Corporation

Oracle Corporation is the largest software company to develop and markets computer software applications for business. The company is best known for its Oracle database products and, more recently, cloud products and services. Its relational database was the first to support SQL, which has since become the industry standard.

Oracle database is one of the most trusted and widely used relational database engines. The biggest rival of Oracle database is Microsoft's SQL Server.

History of Oracle

Oracle Corporation was founded by Lawrence Ellison (Larry Ellison), Bob Miner, Ed Oates, and Bruce Scott in August 1977. They have a lot of experience in building database programs for several companies and builds their first project (a special database program) for the **CIA (Central Intelligence Agency).** Oracle was named after "Project Oracle," a project for one of their clients named Central Intelligence Agency, and the company that created Oracle was called **Systems Development Labs (SDL)**. Systems Development Labs was renamed Relational Software Inc. (RSI) in 1978 to expand their market for the new database. They had again changed the name of the company from RSI to Oracle Systems Corporation in 1982.

The first commercially available RDBMS named **Oracle V2 (Version 2)** was built using PDP-11 assembler language (SQL-based RDBMS). Although they already developed a commercial RDBMS in 1977, it wasn't available for purchase until 1979, when Oracle version 2 was released.

In 1983, Oracle database portable version named "Oracle version 3" was released. This version was written in the C programming language. It was the first relational database that can run in mainframes, minicomputers, PCs, or any hardware with a C compiler. It also supports SQL queries and transactions execution.

The other subsequent versions are:

•**In 1984**, Oracle 4 was released that supports Transactions [Commit/Rollback], export/import utilities, and the report writer.

•**In 1985**, Oracle 5 was released, which provides support for Client-Server Architecture. This new feature has the capability to connect the client's software to a database server through a network.

•**In 1989**, Oracle 6 added support for PL/SQL language. It also comes with new features such as OLTP high-speed systems, hot backup capability, and row-level locking.

•**In 1992**, Oracle 7 was released. This version comes in the market as a result of four years of hard work and two years of customer testing. It added some exciting features and capabilities in the area of security, administration, development, and performance.

•**In 1997**, Oracle 8 was released. This version comes with the support of ORDBMS that was designed to work with Oracle's network computer (NC). It also added support for Java, HTML, and OLTP.

•**In 1998**, Oracle 8i was released. Here 'I' stands for Internet. It was the first database version that added support for Web technologies such as Java and HTTP.

Introduction To Oracle

•**In 2001**, Oracle 9i was released with 400 new features such as XML, RAC (Real Application Clusters), etc. These features reduce database size and provide high availability & enhanced performance.

•**In 2003**, Oracle 10g was released with grid computing technology means grid. It was the first version that supports 64-bit LINUX OS.

•**In 2006**, Oracle 11g was released. This version comes with new features such as Oracle Database Replay, Transaction Management using Log Miner, Virtual Column Partitioning, Case sensitive passwords, Online Patching, Parallel Backups on the same file using RMAN, and many others.

•**In July 2014**, Oracle 12C was released with Cloud support.

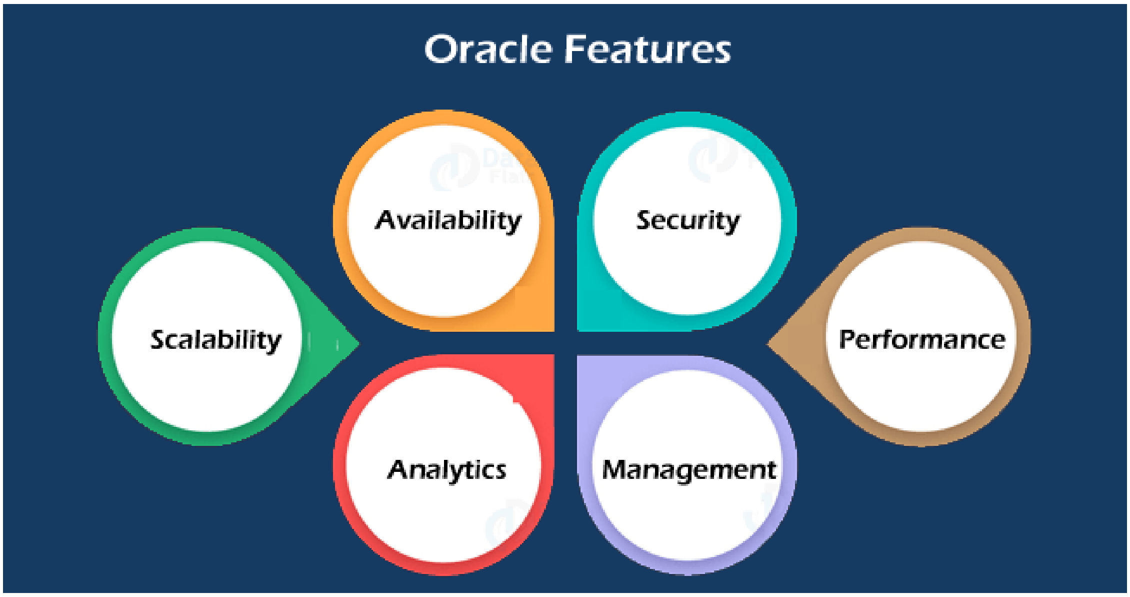
•**In Feb 2018**, Oracle 18 C was released. This version was the world's first autonomous database.

Importance of Oracle

It is one of the oldest database management companies in the world. It has always focused on enterprise needs and kept up with the newest technological developments. As a result, its products are constantly updated with new features. For example, the most recent Oracle database, 19C, is also available on Oracle Cloud. Oracle allows users to select from various database editions to meet their specific demands with a cost-effective solution.

Oracle Database Features

Oracle database manages data with the help of an open, complete, and integrated approach. The following are features that complete the demand for powerful database management:



Introduction To Oracle

Benefits of Oracle Database

The following are the main advantages of an Oracle database:

Performance: Oracle has procedures and principles that help us to get high levels of database performance. We can increase query execution time and operations with the use of performance optimization techniques in its database. This technique helps to retrieve and alter data faster.

Portability: The Oracle database can be ported on all different platforms than any of its competitors. We can use this database on around 20 networking protocols as well as over 100 hardware platforms. This database makes it simple to write an Oracle application by making changes to the OS and hardware in a secure manner.

Backup and Recovery: It is always better to take a proper backup of your entire oracle online backup and recovery. The Oracle database makes it easy to accomplish recovery quickly by using the. RMAN (Recovery Manager) functionality. It can recover or restore database files during downtime or outages. It can be used for online backups, archived backups, and continuous archiving. We can also use SQL* PLUS for recovery, which is known as user-managed recovery.

PL/SQL: One of the greatest benefits of using the Oracle database is to support PL/SQL extension for procedural programming.

Multiple Database: Oracle database allows several database instances management on a single server. It provides an instance caging approach for managing CPU allocations on a server hosting database instances. The database resource management and instance caging can work together to manage services across multiple instances.

Flashback Technology: This advantage comes with the recent Oracle version. It allows us to recover those data that are incorrectly deleted or lost by human errors like accidental deletion of valuable data, deleting the wrong data, or dropping the table.

Disadvantages of Oracle Database

The following are the disadvantages of the Oracle database:

Complexity: Oracle is not recommended to use when the users are not technically savvy and have limited technical skills required to deal with the Oracle Database. It is also not advised to use if the company is looking for a database with limited functionality and easy to use.

Cost: The price of Oracle products is very high in comparison to other databases. Therefore users are more likely to choose other less expensive options such as MS SQL Server, MySQL, etc.

Difficult to manage: Oracle databases are often much more complex and difficult in terms of the management of certain activities.

Data Types in SQL

- Each column in a database table is required to have a name and a data type.
- An SQL developer must decide what type of data that will be stored inside each column when creating a table. The data type is a guideline for SQL to understand what type of data is expected inside of each column, and it also identifies how SQL will interact with the stored data.

- **Note:** Data types might have different names in different database. And even if the name is the same, the size and other details may be different! **Always check the documentation!**

SQL Data Types

An SQL developer must be aware of what type of data will be stored inside each column while creating a table. The data type guideline for SQL is to understand what type of data is expected inside each column and it also identifies how SQL will interact with the stored data.

For every database, data types are primarily classified into three categories.

- Numeric Datatypes
- Date and Time Database
- String Database

MySQL Data Types (Version 8.0)

In MySQL there are three main data types: string, numeric, and date and time.

String Data Types

Data type	Description
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters). The *size* parameter specifies the column length in characters - can be from 0 to 255. Default is 1
VARCHAR(size)	A VARIABLE length string (can contain letters, numbers, and special characters). The *size* parameter specifies the maximum string length in characters - can be from 0 to 65535
BINARY(size)	Equal to CHAR(), but stores binary byte strings. The *size* parameter specifies the column length in bytes. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. The *size* parameter specifies the maximum column length in bytes.
TINYBLOB	For BLOBs (Binary Large Objects). Max length: 255 bytes
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT(size)	Holds a string with a maximum length of 65,535 bytes
BLOB(size)	For BLOBs (Binary Large Objects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large Objects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LONGBLOB	For BLOBs (Binary Large Objects). Holds up to 4,294,967,295 bytes of data

ENUM(val1, val2, val3, ...)	A string object that can have only one value, chosen from a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. The values are sorted in the order you enter them
SET(val1, val2, val3, ...)	A string object that can have 0 or more values, chosen from a list of possible values. You can list up to 64 values in a SET list

Numeric Data Types

Data type	**Description**
BIT(*size*)	A bit-value type. The number of bits per value is specified in *size*. The *size* parameter can hold a value from 1 to 64. The default value for *size* is 1.
TINYINT(*size*)	A very small integer. Signed range is from -128 to 127. Unsigned range is from 0 to 255. The *size* parameter specifies the maximum display width (which is 255)
BOOL	Zero is considered as false, nonzero values are considered as true.
BOOLEAN	Equal to BOOL
SMALLINT(*size*)	A small integer. Signed range is from -32768 to 32767. Unsigned range is from 0 to 65535. The *size* parameter specifies the maximum display width (which is 255)
MEDIUMINT(*size*)	A medium integer. Signed range is from -8388608 to 8388607. Unsigned range is from 0 to 16777215. The *size* parameter specifies the maximum display width (which is 255)
INT(*size*)	A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The *size* parameter specifies the maximum display width (which is 255)
INTEGER(*size*)	Equal to INT(size)

BIGINT(*size*)	A large integer. Signed range is from - 9223372036854775808 to 9223372036854775807. Unsigned range is from 0 to 18446744073709551615. The *size* parameter specifies the maximum display width (which is 255)
FLOAT(*size*, *d*)	A floating point number. The total number of digits is specified in *size*. The number of digits after the decimal point is specified in the *d* parameter. This syntax is deprecated in MySQL 8.0.17, and it will be removed in future MySQL versions
FLOAT(*p*)	A floating point number. MySQL uses the *p* value to determine whether to use FLOAT or DOUBLE for the resulting data type. If *p* is from 0 to 24, the data type becomes FLOAT(). If *p* is from 25 to 53, the data type becomes DOUBLE()
DOUBLE(*size*, *d*)	A normal-size floating point number. The total number of digits is specified in *size*. The number of digits after the decimal point is specified in the *d* parameter
DOUBLE PRECISION(*size*, *d*)	
DECIMAL(*size*, *d*)	An exact fixed-point number. The total number of digits is specified in *size*. The number of digits after the decimal point is specified in the *d* parameter. The maximum number for *size* is 65. The maximum number for *d* is 30. The default value for *size* is 10. The default value for *d* is 0.
DEC(*size*, *d*)	Equal to DECIMAL(size,d)

Note: All the numeric data types may have an extra option: UNSIGNED or ZEROFILL. If you add the UNSIGNED option, MySQL disallows negative values for the column. If you add the ZEROFILL option, MySQL automatically also adds the UNSIGNED attribute to the column.

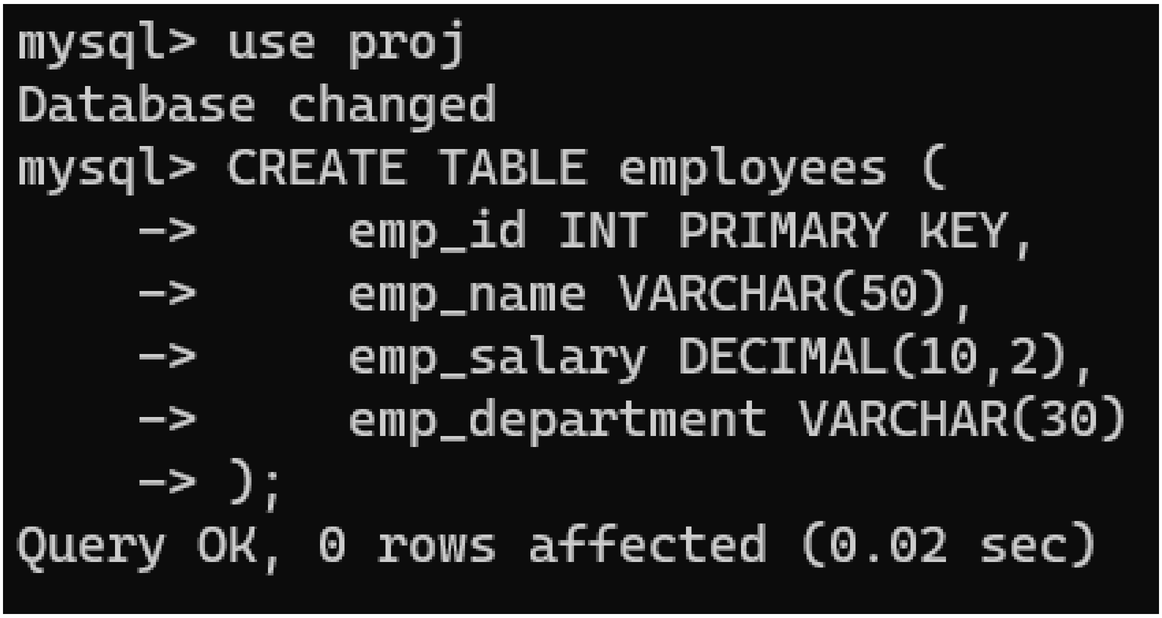
Date and Time Data Types

Data type	**Description**

DATE	A date. Format: YYYY-MM-DD. The supported range is from '1000-01-01' to '9999-12-31'
DATETIME(*fsp*)	A date and time combination. Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. Adding DEFAULT and ON UPDATE in the column definition to get automatic initialization and updating to the current date and time
TIMESTAMP(*fsp*)	A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD hh:mm:ss. The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC. Automatic initialization and updating to the current date and time can be specified using DEFAULT CURRENT_TIMESTAMP and ON UPDATE CURRENT_TIMESTAMP in the column definition
TIME(*fsp*)	A time. Format: hh:mm:ss. The supported range is from '-838:59:59' to '838:59:59'
YEAR	A year in four-digit format. Values allowed in four-digit format: 1901 to 2155, and 0000. MySQL 8.0 does not support year in two-digit format.

WAC for Creating a Table

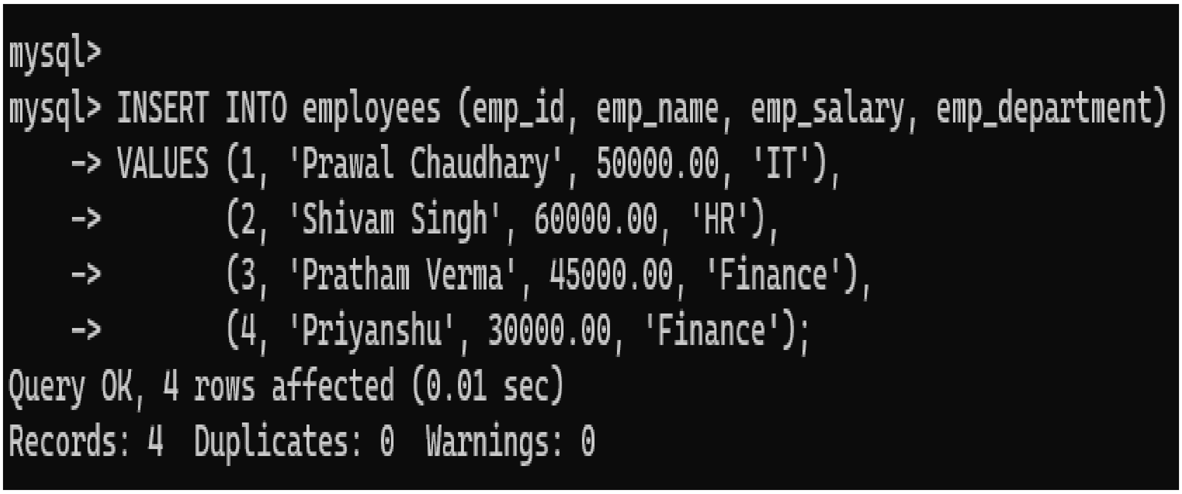
To create table in specific schema, we must use two-part name. The schema in which you are creating the table must exists in the database. For example, you want to create a table named **tblPatient** in **Patients** schema, the Create table definition will be as following:



```
mysql> use proj
Database changed
mysql> CREATE TABLE employees (
    ->     emp_id INT PRIMARY KEY,
    ->     emp_name VARCHAR(50),
    ->     emp_salary DECIMAL(10,2),
    ->     emp_department VARCHAR(30)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

WPC to insert the value into a table

To Insert data in a table we use **INSERT INTO** command followed by the table name



```
mysql>
mysql> INSERT INTO employees (emp_id, emp_name, emp_salary, emp_department)
    -> VALUES (1, 'Prawal Chaudhary', 50000.00, 'IT'),
    ->        (2, 'Shivam Singh', 60000.00, 'HR'),
    ->        (3, 'Pratham Verma', 45000.00, 'Finance'),
    ->        (4, 'Priyanshu', 30000.00, 'Finance');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

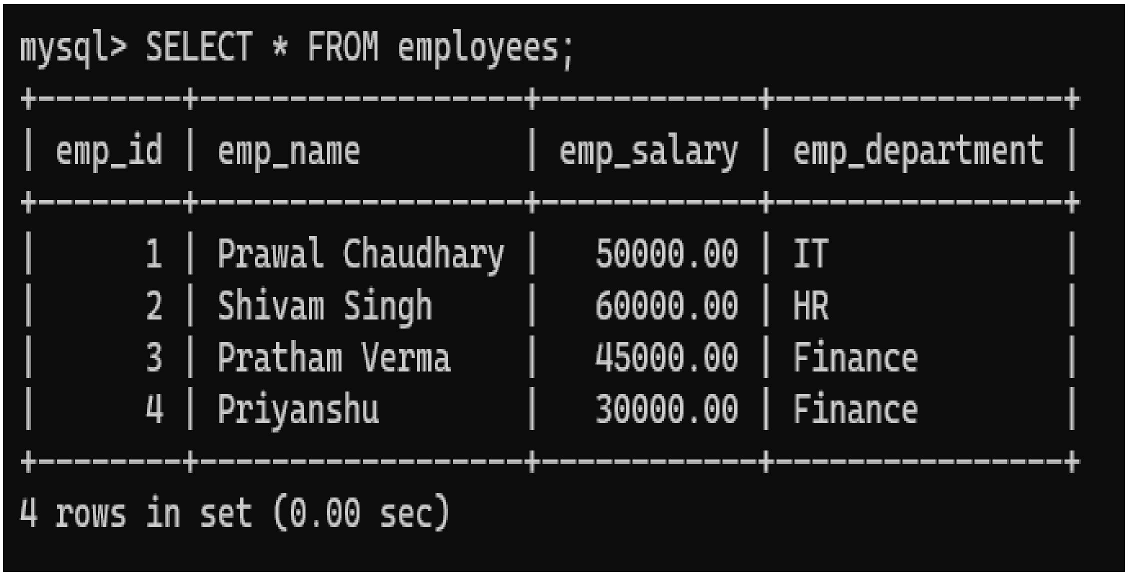
WPC to view the data into a table

The MySQL SELECT Statement

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

```
SELECT column1, column2,
...
FROM table_name;

Or

SELECT * FROM table_name;
```



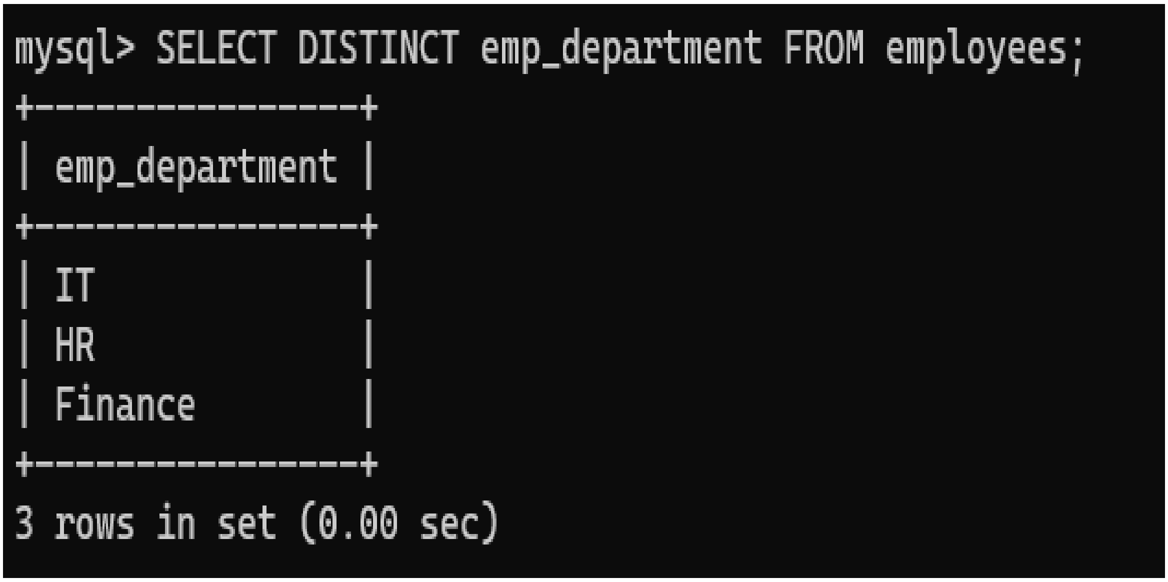
```
mysql> SELECT * FROM employees;
+--------+------------------+------------+----------------+
| emp_id | emp_name         | emp_salary | emp_department |
+--------+------------------+------------+----------------+
|      1 | Prawal Chaudhary |   50000.00 | IT             |
|      2 | Shivam Singh     |   60000.00 | HR             |
|      3 | Pratham Verma    |   45000.00 | Finance        |
|      4 | Priyanshu        |   30000.00 | Finance        |
+--------+------------------+------------+----------------+
4 rows in set (0.00 sec)
```

WPC to Eliminate duplicate rows

The SQL SELECT DISTINCT Statement

The `SELECT DISTINCT` statement is used to return only distinct (different) values.

```
SELECT DISTINCT Country F
ROM Customers;
```



```
mysql> SELECT DISTINCT emp_department FROM employees;
+----------------+
| emp_department |
+----------------+
| IT             |
| HR             |
| Finance        |
+----------------+
3 rows in set (0.00 sec)
```

WPC to sort data into a table

ORDER BY

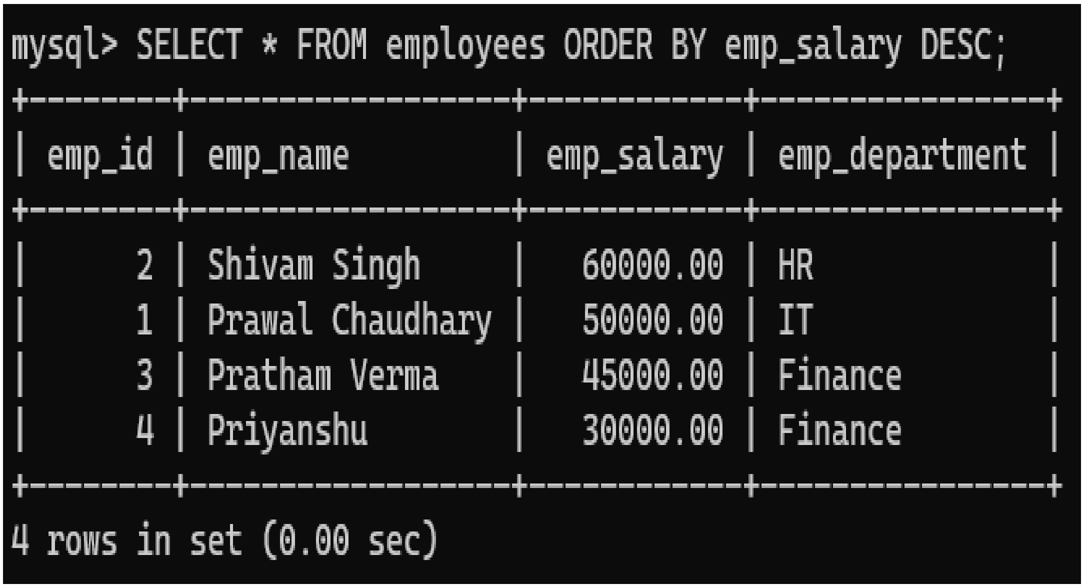
The ORDER BY command is used to sort the result set in ascending or descending order.

The ORDER BY command sorts the result set in ascending order by default. To sort the records in descending order, use the DESC keyword.

The following SQL statement selects all the columns from the "Customers" table, sorted by the "CustomerName" column:

Example

```
SELECT * FROM
Customers ORDER BY
CustomerName;
```



```
mysql> SELECT * FROM employees ORDER BY emp_salary DESC;
+--------+------------------+------------+----------------+
| emp_id | emp_name         | emp_salary | emp_department |
+--------+------------------+------------+----------------+
|      2 | Shivam Singh     |   60000.00 | HR             |
|      1 | Prawal Chaudhary |   50000.00 | IT             |
|      3 | Pratham Verma    |   45000.00 | Finance        |
|      4 | Priyanshu        |   30000.00 | Finance        |
+--------+------------------+------------+----------------+
4 rows in set (0.00 sec)
```

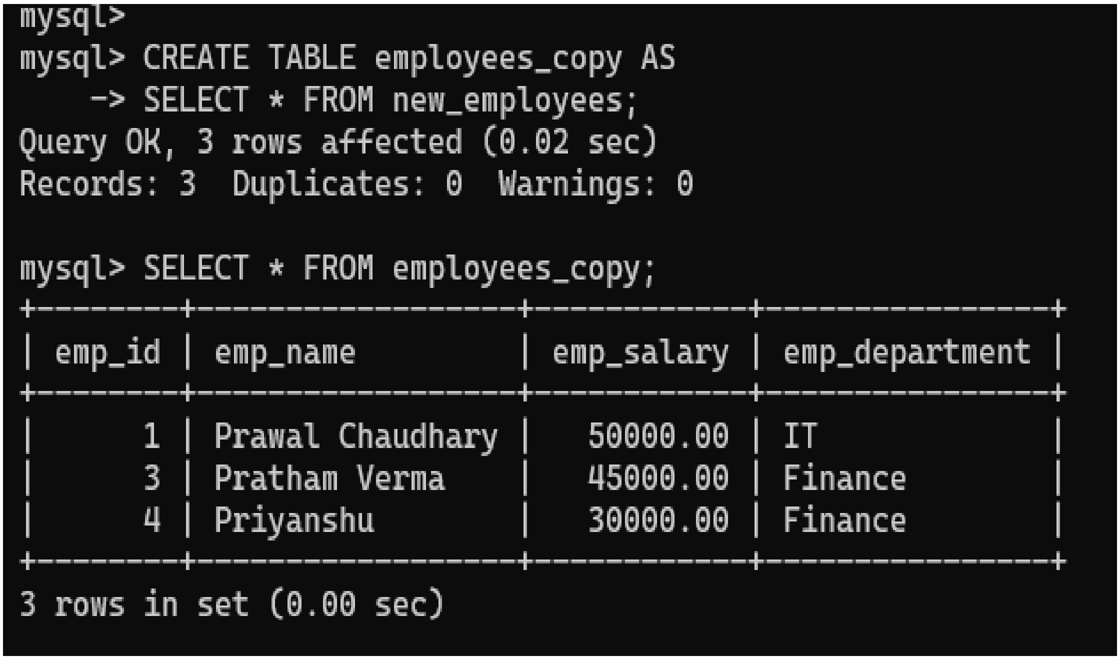
WPC for creating a table from a table

Create Table - By Copying all columns from another table

Syntax

The syntax for the CREATE TABLE AS statement when copying all of the columns in SQL is:

```
CREATE TABLE new_table
 AS (SELECT * FROM old_table);
```



```
mysql>
mysql> CREATE TABLE employees_copy AS
    -> SELECT * FROM new_employees;
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM employees_copy;
+--------+------------------+------------+----------------+
| emp_id | emp_name         | emp_salary | emp_department |
+--------+------------------+------------+----------------+
|      1 | Prawal Chaudhary |   50000.00 | IT             |
|      3 | Pratham Verma    |   45000.00 | Finance        |
|      4 | Priyanshu        |   30000.00 | Finance        |
+--------+------------------+------------+----------------+
3 rows in set (0.00 sec)
```

WPC to insert the data into a table from another table

The SQL INSERT INTO SELECT Statement

The `INSERT INTO SELECT` statement copies data from one table and inserts it into another table.

The `CREATE TABLE AS` statement creates the table and copy all the data from the data

Note: The existing records in the target table are unaffected.

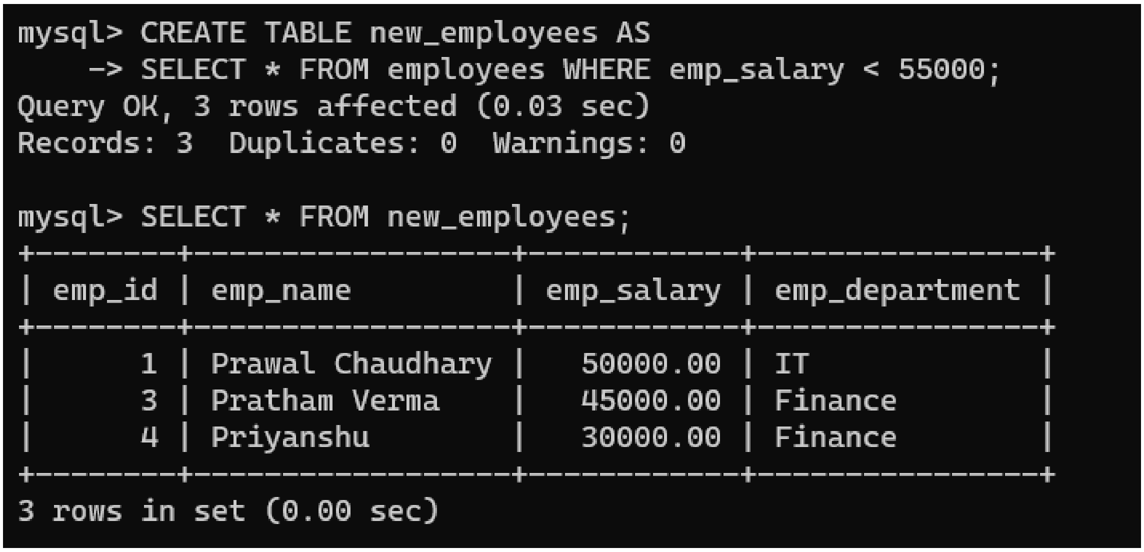
INSERT INTO SELECT Syntax

Copy all columns from one table to another table:

```
INSERT INTO
table2  SELECT *
FROM table1
WHERE condition;
```

Copy only some columns from one table into another table:

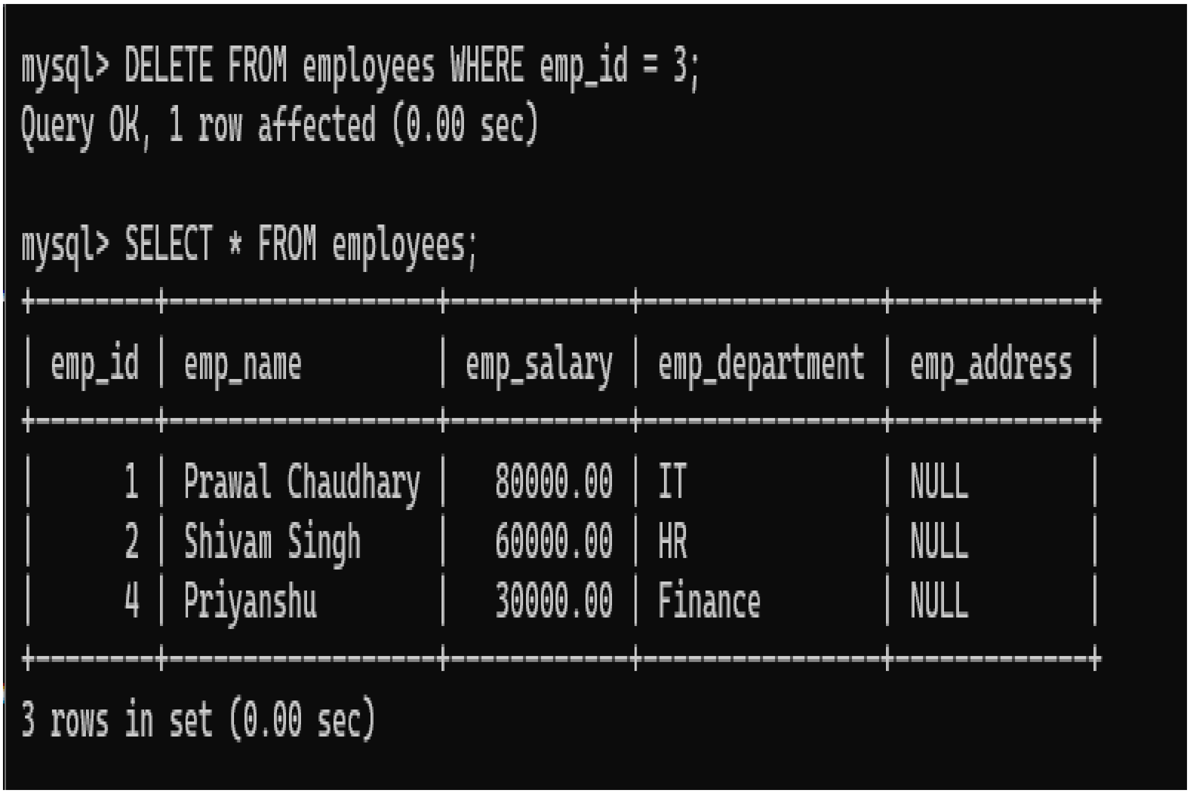
```
INSERT INTO table2 (column1, column2,
column3, ...) SELECT column1, column2,
column3, ...
FROM table1
WHERE condition;
```



```
mysql> CREATE TABLE new_employees AS
    -> SELECT * FROM employees WHERE emp_salary < 55000;
Query OK, 3 rows affected (0.03 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM new_employees;
+--------+------------------+------------+----------------+
| emp_id | emp_name         | emp_salary | emp_department |
+--------+------------------+------------+----------------+
|      1 | Prawal Chaudhary |   50000.00 | IT             |
|      3 | Pratham Verma    |   45000.00 | Finance        |
|      4 | Priyanshu        |   30000.00 | Finance        |
+--------+------------------+------------+----------------+
3 rows in set (0.00 sec)
```

WPC for delete operation

The **DELETE COMMAND** is used to delete data from table which is followed by WHERE condition.



```
mysql> DELETE FROM employees WHERE emp_id = 3;
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM employees;
+--------+-----------------+------------+----------------+-------------+
| emp_id | emp_name        | emp_salary | emp_department | emp_address |
+--------+-----------------+------------+----------------+-------------+
|      1 | Prawal Chaudhary |   80000.00 | IT             | NULL        |
|      2 | Shivam Singh    |   60000.00 | HR             | NULL        |
|      4 | Priyanshu       |   30000.00 | Finance        | NULL        |
+--------+-----------------+------------+----------------+-------------+
3 rows in set (0.00 sec)
```

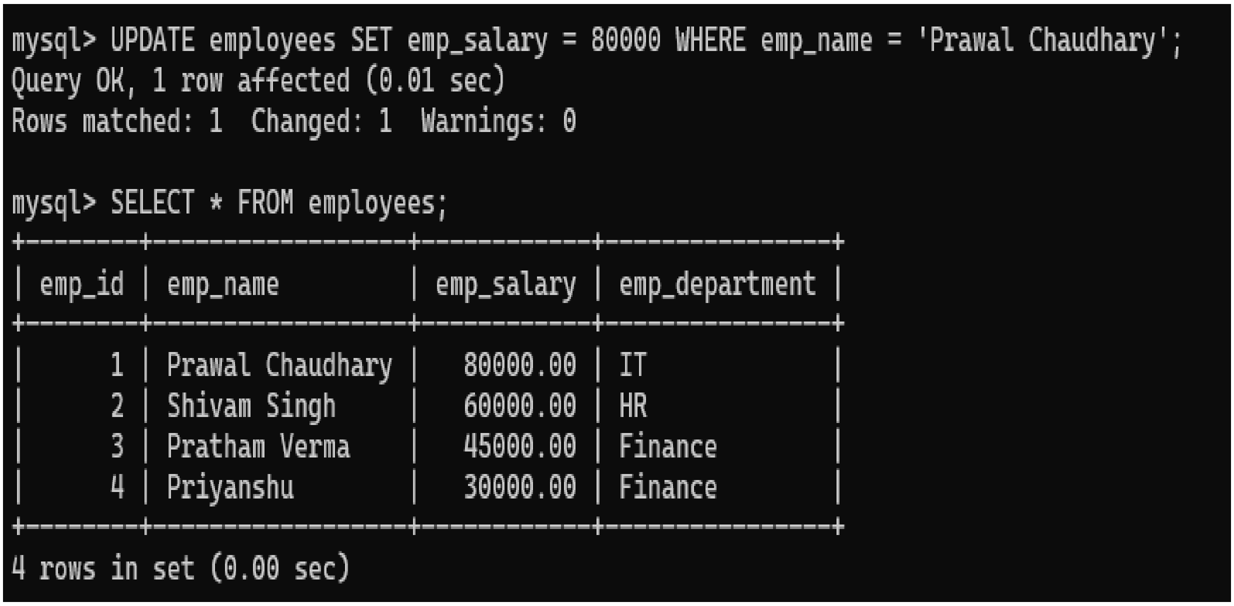
WPC for updating content of table

The SQL UPDATE Statement

The UPDATE statement is used to modify the existing records in a table.

UPDATE Syntax

```
UPDATE table_name
SET column1 = value1, column2 = value2
, ...
WHERE condition;
```



```
mysql> UPDATE employees SET emp_salary = 80000 WHERE emp_name = 'Prawal Chaudhary';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM employees;
+--------+------------------+------------+----------------+
| emp_id | emp_name         | emp_salary | emp_department |
+--------+------------------+------------+----------------+
|      1 | Prawal Chaudhary |   80000.00 | IT             |
|      2 | Shivam Singh     |   60000.00 | HR             |
|      3 | Pratham Verma    |   45000.00 | Finance        |
|      4 | Priyanshu        |   30000.00 | Finance        |
+--------+------------------+------------+----------------+
4 rows in set (0.00 sec)
```

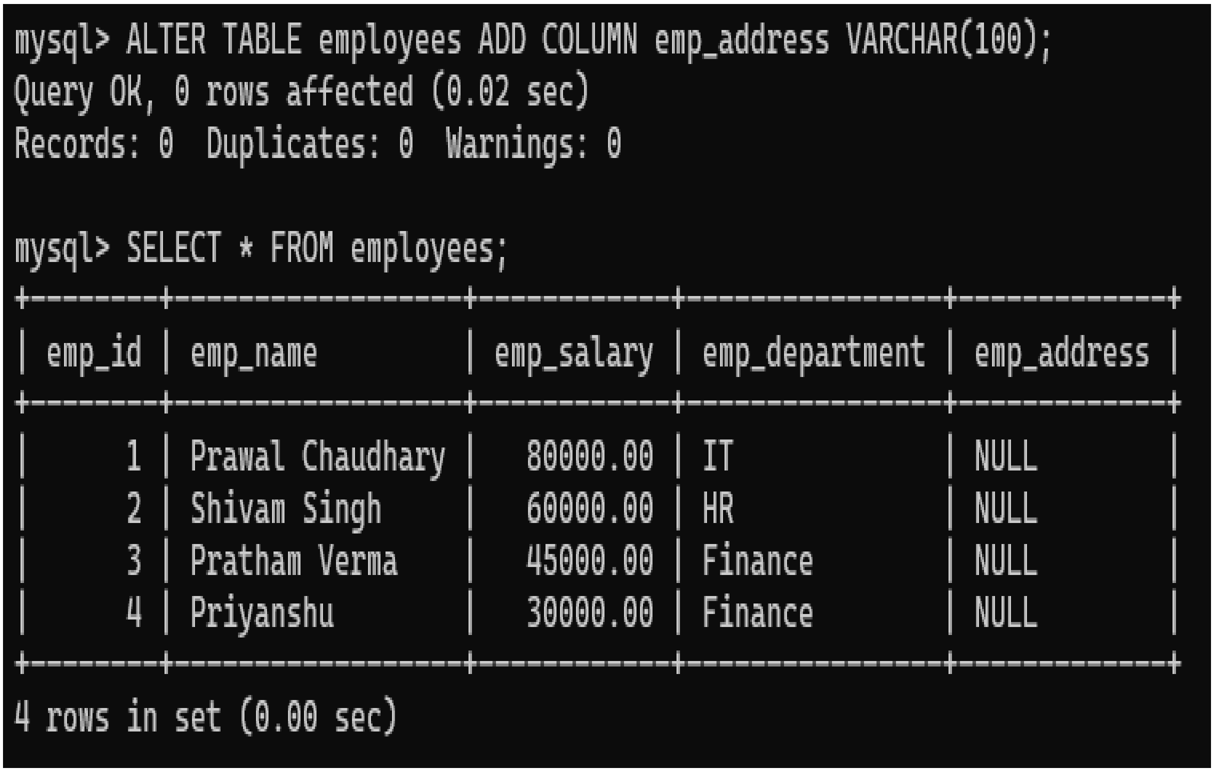
WPC for modifying the structure of a table

SQL ALTER TABLE Statement

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

```
ALTER TABLE employees
ADD COLUMN emp_address
varchar(255);
```



```
mysql> ALTER TABLE employees ADD COLUMN emp_address VARCHAR(100);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM employees;
+--------+-----------------+------------+----------------+-------------+
| emp_id | emp_name        | emp_salary | emp_department | emp_address |
+--------+-----------------+------------+----------------+-------------+
|      1 | Prawal Chaudhary |   80000.00 | IT             | NULL        |
|      2 | Shivam Singh    |   60000.00 | HR             | NULL        |
|      3 | Pratham Verma   |   45000.00 | Finance        | NULL        |
|      4 | Priyanshu       |   30000.00 | Finance        | NULL        |
+--------+-----------------+------------+----------------+-------------+
4 rows in set (0.00 sec)
```

WPC for destroying a table

The SQL DROP TABLE Statement

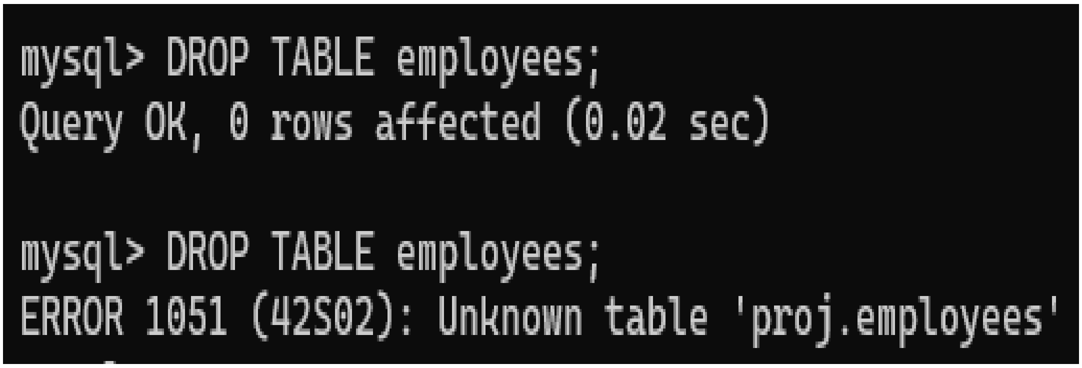
The DROP TABLE statement is used to drop an existing table in a database.

SQL DROP TABLE Example

The following SQL statement drops the existing table "Shippers":

Example

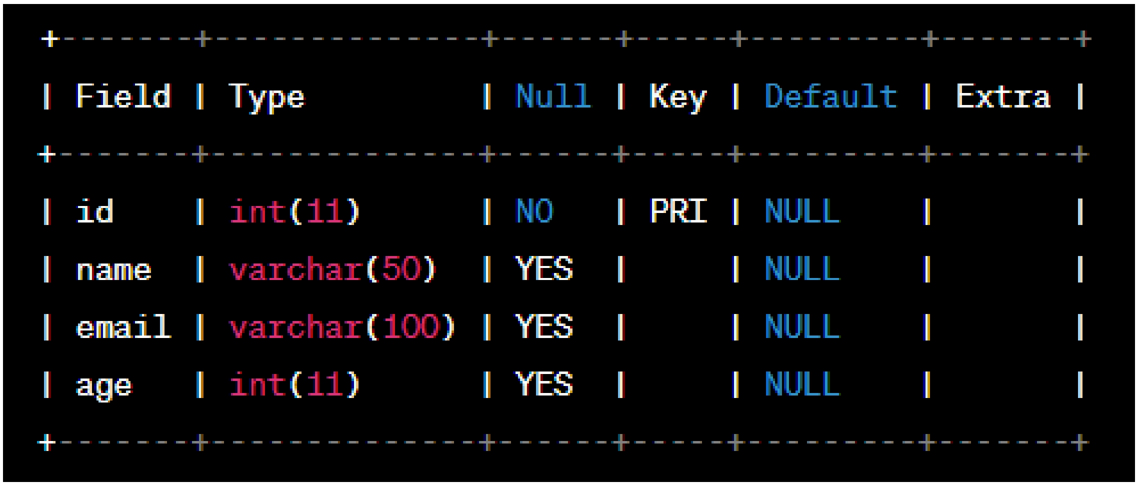
```
DROP TABLE Shippers;
```



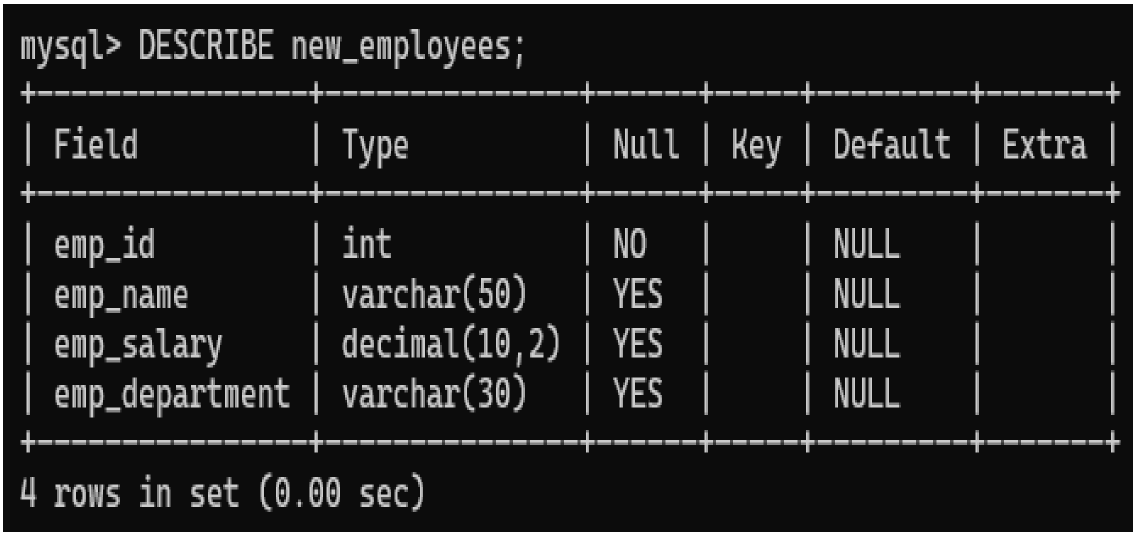
```
mysql> DROP TABLE employees;
Query OK, 0 rows affected (0.02 sec)

mysql> DROP TABLE employees;
ERROR 1051 (42S02): Unknown table 'proj.employees'
```

WPC for displaying a table structure

As the name suggests, DESCRIBE is used to describe something. Since in a database, we have tables, that’s why do we use **DESCRIBE** or **DESC**(both are the same) commands to describe the **structure** of a table.



```
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| id    | int(11)      | NO   | PRI | NULL    |       |
| name  | varchar(50)  | YES  |     | NULL    |       |
| email | varchar(100) | YES  |     | NULL    |       |
| age   | int(11)      | YES  |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
```



```
mysql> DESCRIBE new_employees;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| emp_id         | int           | NO   |     | NULL    |       |
| emp_name       | varchar(50)   | YES  |     | NULL    |       |
| emp_salary     | decimal(10,2) | YES  |     | NULL    |       |
| emp_department | varchar(30)   | YES  |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

WPC for Count(), AVG(), SUM() functions

The SQL COUNT(), AVG() and SUM() Functions

The COUNT() function returns the number of rows that matches a specified criterion.

COUNT() Syntax

```
SELECT
COUNT(column_name) FROM
table_name
WHERE condition;
```

The AVG() function returns the average value of a numeric column.

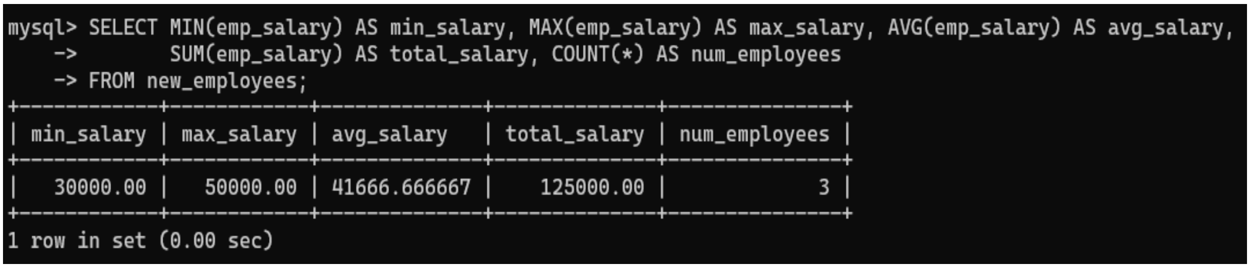
AVG() Syntax

```
SELECT
AVG(column_name)
FROM table_name
WHERE condition;
```

The SUM() function returns the total sum of a numeric column.

SUM() Syntax

```
SELECT
SUM(column_name)
FROM table_name
WHERE condition;
```



```
mysql> SELECT MIN(emp_salary) AS min_salary, MAX(emp_salary) AS max_salary, AVG(emp_salary) AS avg_salary,
    ->        SUM(emp_salary) AS total_salary, COUNT(*) AS num_employees
    -> FROM new_employees;
+------------+------------+--------------+--------------+---------------+
| min_salary | max_salary | avg_salary   | total_salary | num_employees |
+------------+------------+--------------+--------------+---------------+
|   30000.00 |   50000.00 | 41666.666667 |    125000.00 |             3 |
+------------+------------+--------------+--------------+---------------+
1 row in set (0.00 sec)
```