Basic Database Operations - I

Session Objectives:

- Import datasets into MySQL Workbench
- Use SELECT statements to explore data
- Understand and apply the DESCRIBE function
- Recognize SQL data types
- Use the WHERE clause for filtering
- Use logical & comparison operators in the WHERE clause
- Work with wildcards for pattern matching
- Remove duplicates with DISTINCT
- Sort data using ORDER BY
- Limit results using LIMIT & OFFSET
- Add comments for better documentation
- Use column aliases for readability

SQL is not a CASE SENSITIVE Language

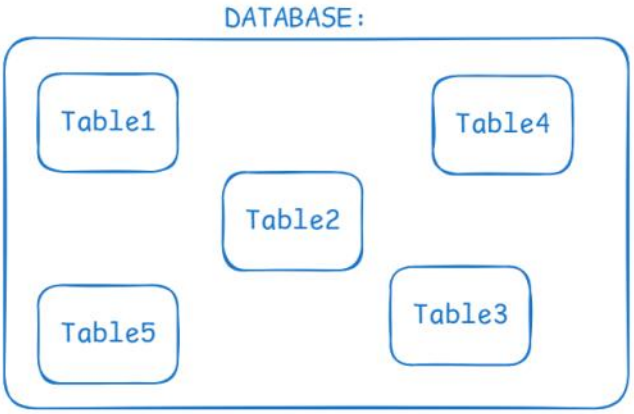
Comments:

Python Comment

```
# CREATE DATABASE <db_name>;
/*
        This is a multiline comment
    SELECT * FROM Customers;
*/
```

```
-- SELECT * FROM Customers;
```

Single Line Comment

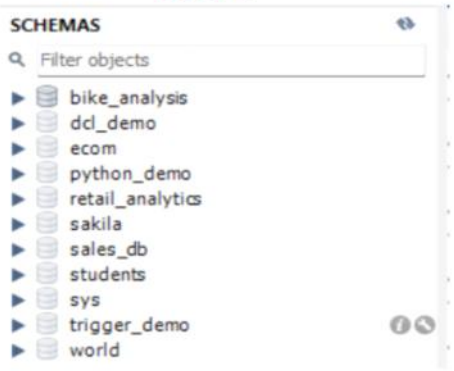


User-Defined

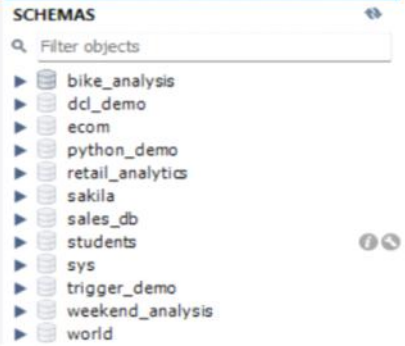
```
CREATE DATABASE weekend_analysis;
```

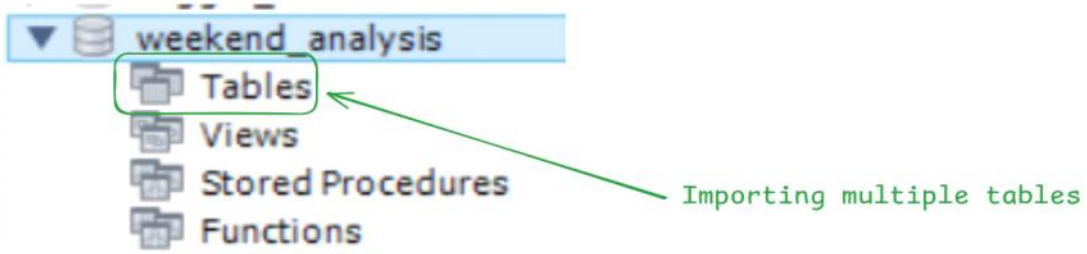
Predefined Keywords

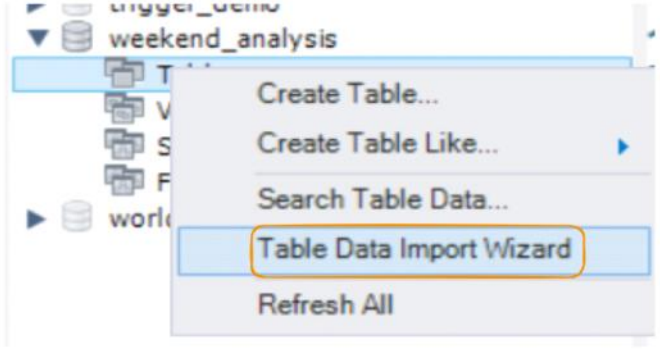
Before

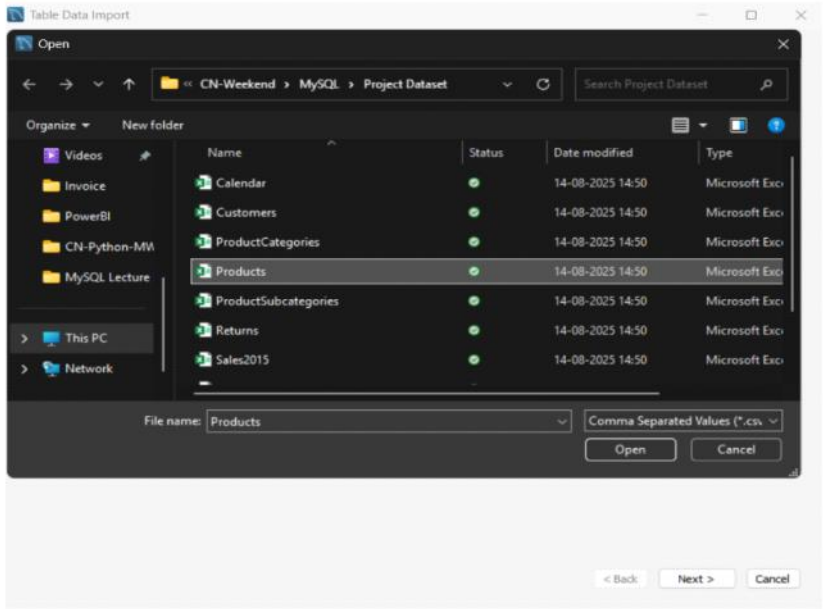


After









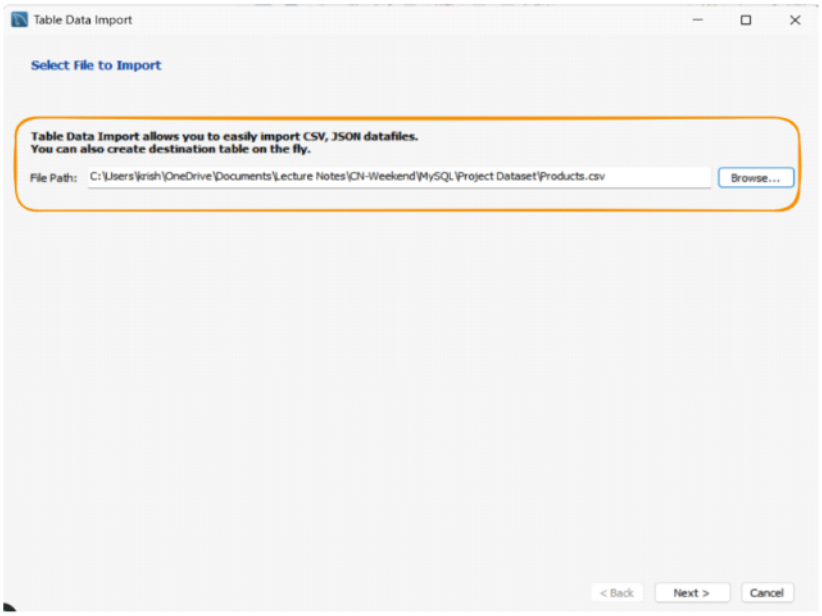


Table Data Import
Select File to Import
Table Data Import allows you to easily import CSV, JSON datafiles.
You can also create destination table on the fly.
File Path:
C:\Users\krish\OneDrive\Documents\Lecture Notes\CN-Weekend\MySQL\Project Dataset\Products.csv
Browse...
< Back
Next >
Cancel

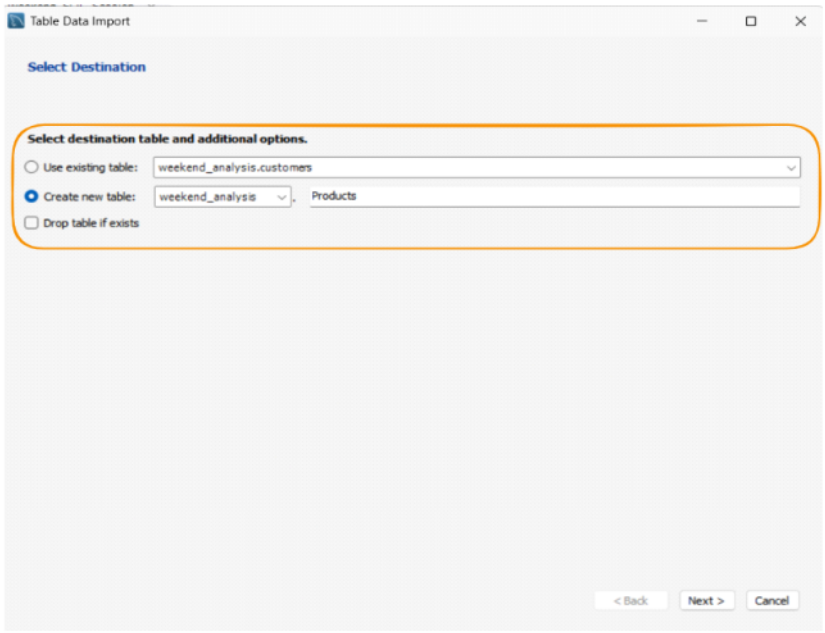


Table Data Import
Select Destination
Select destination table and additional options.
Use existing table:
weekend_analysis.customers
Create new table:
weekend_analysis
Products
Drop table if exists
< Back
Next >
Cancel

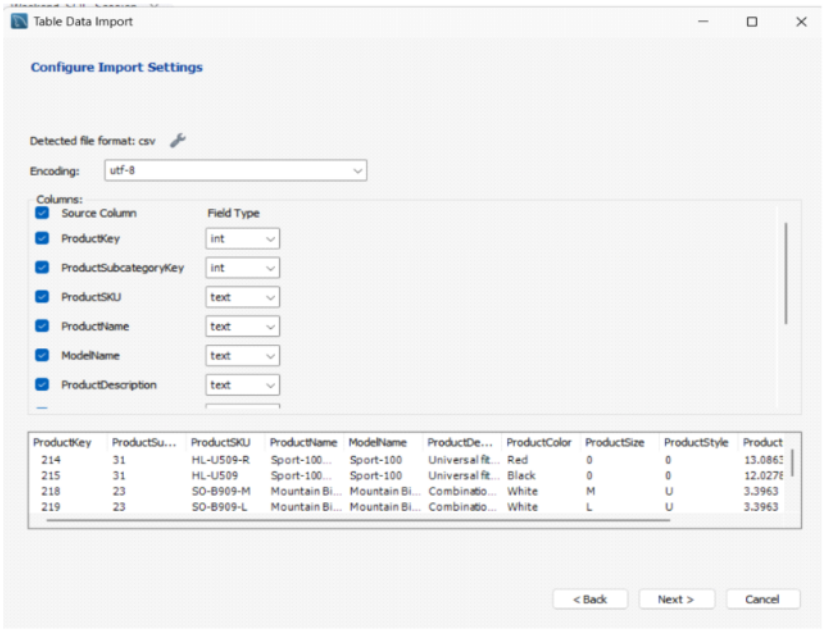


Table Data Import
Configure Import Settings
Detected file format: csv
Encoding:
utf-8
Columns:
Source Column
Field Type
ProductKey
int
ProductSubcategoryKey
int
ProductSKU
text
ProductName
text
ModelName
text
ProductDescription
text
ProductKey ProductSu... ProductSKU ProductName ModelName ProductDe... ProductColor ProductSize ProductStyle Product
214 31 HL-U509-R Sport-100... Sport-100 Universal fit... Red 0 0 13.086
215 31 HL-U509 Sport-100... Sport-100 Universal fit... Black 0 0 12.027
218 23 SO-B909-M Mountain Bi... Mountain Bi... Combinatio... White M U 3.3963
219 23 SO-B909-L Mountain Bi... Mountain Bi... Combinatio... White L U 3.3963
< Back
Next >
Cancel

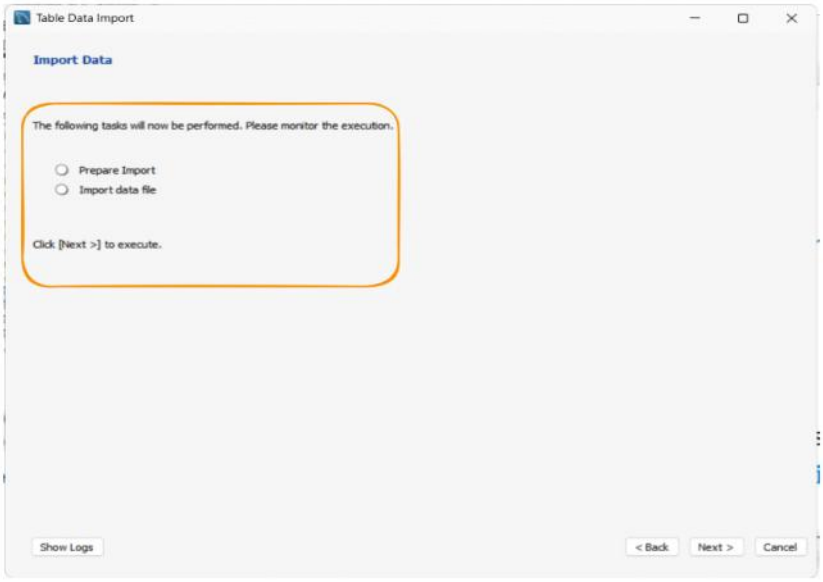


Table Data Import
Import Data
The following tasks will now be performed. Please monitor the execution.
Prepare Import
Import data file
Click [Next >] to execute.
Show Logs
< Back
Next >
Cancel

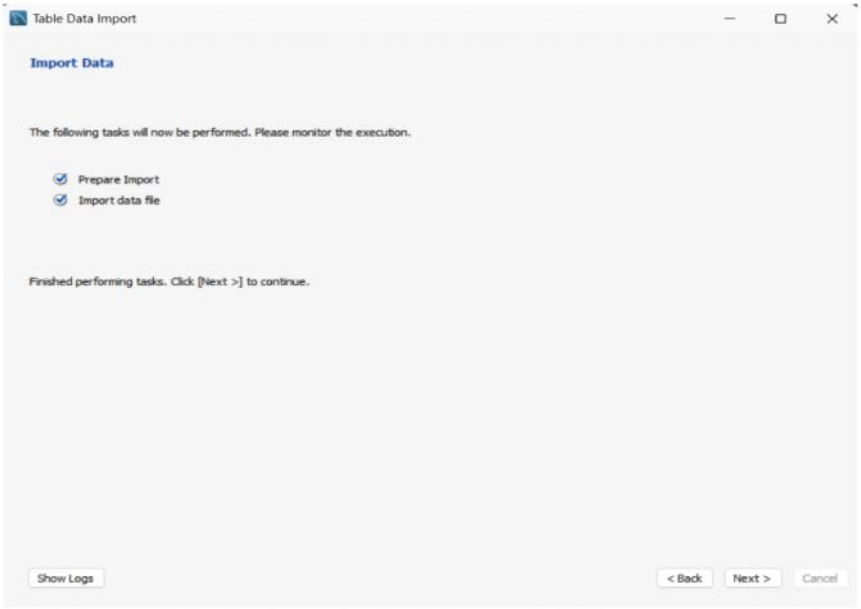


Table Data Import
Import Data
The following tasks will now be performed. Please monitor the execution.
Prepare Import
Import data file
Finished performing tasks. Click [Next >] to continue.
Show Logs
< Back
Next >
Cancel

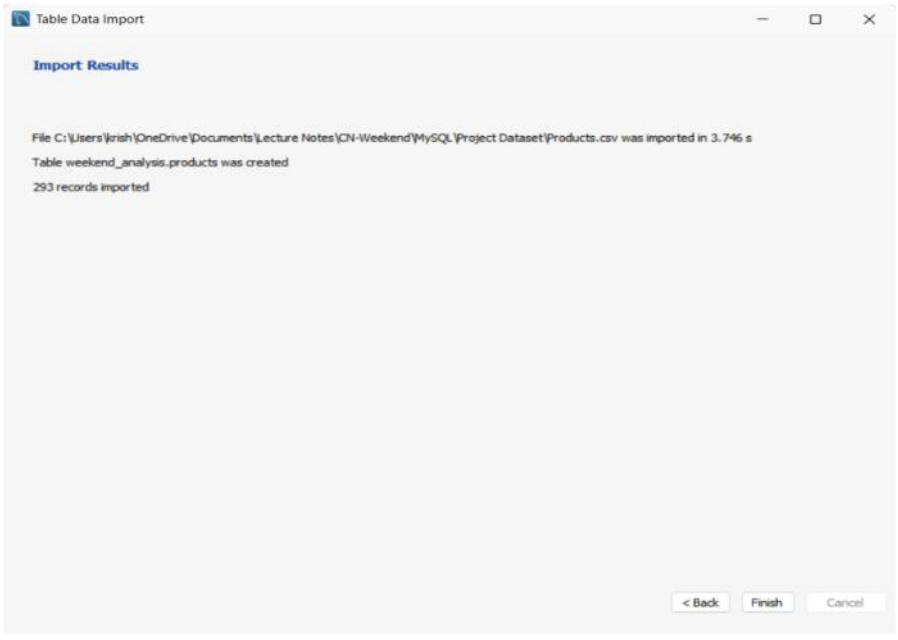
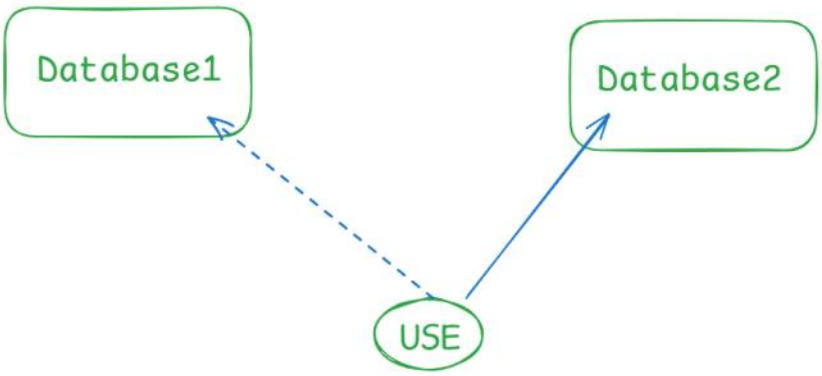


Table Data Import
Import Results
File C:\Users\krish\OneDrive\Documents\Lecture Notes\CN-Weekend\MySQL\Project Dataset\Products.csv was imported in 3.746 s
Table weekend_analysis.products was created
293 records imported
< Back
Finish
Cancel



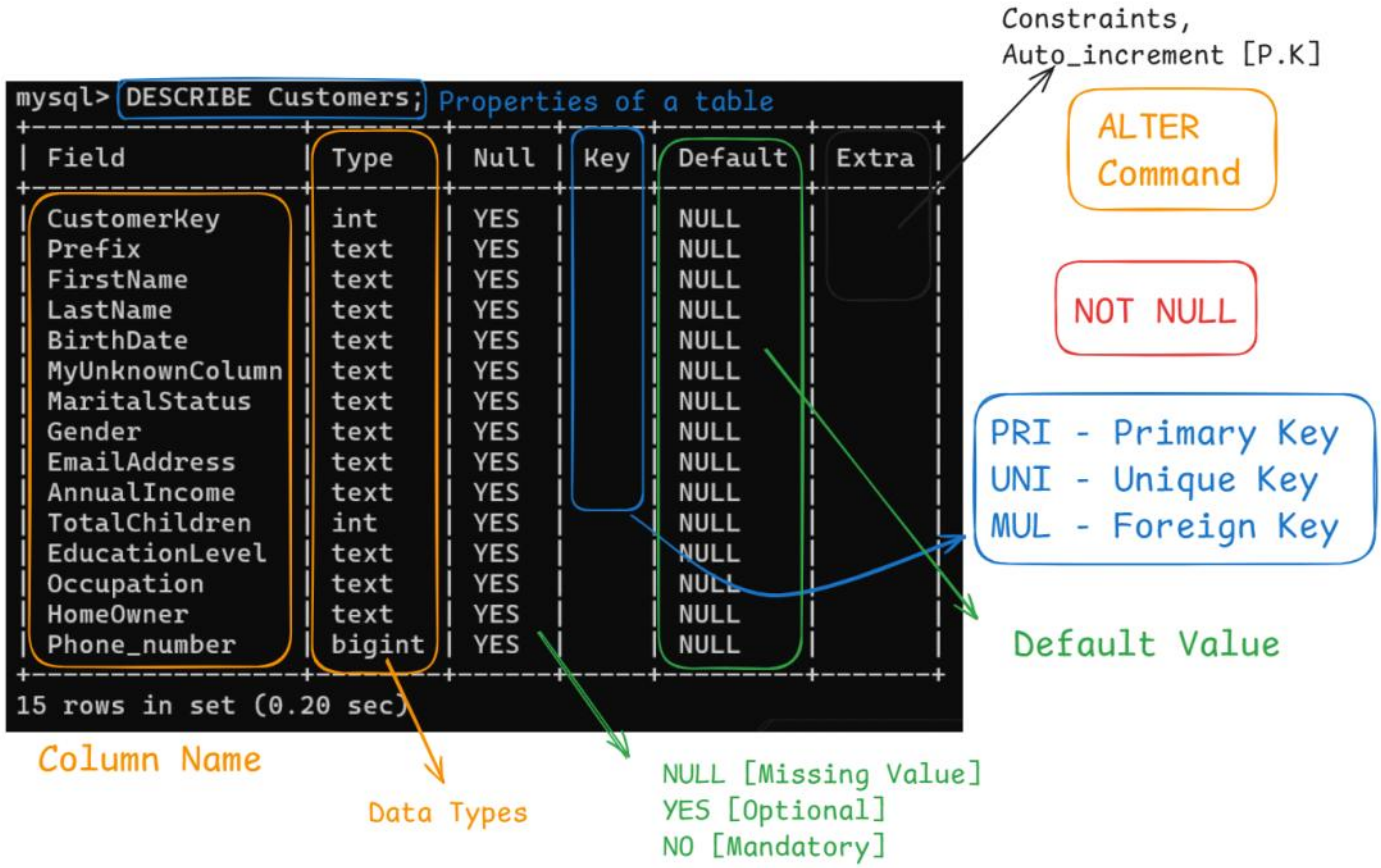

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| bike_analysis      |
| dcl_demo           |
| ecom               |
| information_schema |
| mysql              |
| performance_schema |
| python_demo        |
| retail_analytics   |
| sakila             |
| sales_db           |
| students           |
| sys                |
| trigger_demo       |
| weekend_analysis   |
| world              |
+--------------------+
15 rows in set (0.20 sec)
```

```
mysql> SELECT DATABASE();
+------------+
| DATABASE() |
+------------+
| NULL       |
+------------+
1 row in set (0.06 sec)

mysql> USE weekend_analysis;
Database changed
mysql> SELECT DATABASE();
+------------------+
| DATABASE()       |
+------------------+
| weekend_analysis |
+------------------+
```

```
mysql> SHOW TABLES;
+----------------------------+
| Tables_in_weekend_analysis |
+----------------------------+
| calendar                   |
| customers                  |
| productcategories          |
| products                   |
| productsubcategories       |
| returns                    |
| sales2015                  |
| sales2016                  |
+----------------------------+
8 rows in set (0.19 sec)
```

USE

```
mysql> USE bike_analysis;
Database changed
mysql> SELECT DATABASE();
+---------------+
| DATABASE()    |
+---------------+
| bike_analysis |
+---------------+
1 row in set (0.00 sec)

mysql> SHOW TABLES;
+--------------------------+
| Tables_in_bike_analysis  |
+--------------------------+
| bank_transactions        |
| calendar                 |
| children                 |
| customer                 |
| customer_unique          |
| customers                |
| employee_detail          |
| events                   |
| orders                   |
| product-categories       |
| product-subcategories    |
| products                 |
| returns                  |
| sales-2015               |
| sales-2016               |
| sales-2017               |
| salestrend               |
| student_info             |
| students                 |
| territories              |
+--------------------------+
20 rows in set (0.01 sec)
```



Field	Type	Null	Key	Default	Extra
CustomerKey	int	YES		NULL	
Prefix	text	YES		NULL	
FirstName	text	YES		NULL	
LastName	text	YES		NULL	
BirthDate	text	YES		NULL	
MyUnknownColumn	text	YES		NULL	
MaritalStatus	text	YES		NULL	
Gender	text	YES		NULL	
EmailAddress	text	YES		NULL	
AnnualIncome	text	YES		NULL	
TotalChildren	int	YES		NULL	
EducationLevel	text	YES		NULL	
Occupation	text	YES		NULL	
HomeOwner	text	YES		NULL	
Phone_number	bigint	YES		NULL	

Google Form

Name - _____(*)

Age - __(*)

Gender : _ [F/M]

Email : _____ [*]

Phone : _____(*)

City : _____

```
USE weekend_analysis;
SELECT * FROM Customers;
SELECT
        FirstName,
    EmailAddress,
    AnnualIncome,
    EducationLevel,
    Occupation
FROM Customers;

SELECT * FROM Products;
```

```
SYNTAX :
        SELECT col1, col2, col3 ....
        FROM TableName
        WHERE Condition;
```

```
DESC Products;
DESCRIBE  Products;
```

Challenge 1 : Find the Product Details having ProductPrice > 1000.

```
mysql> DESCRIBE Products;
+-----------------------+--------+------+-----+---------+-------+
| Field                 | Type   | Null | Key | Default | Extra |
+-----------------------+--------+------+-----+---------+-------+
| ProductKey            | int    | YES  |     | NULL    |       |
| ProductSubcategoryKey | int    | YES  |     | NULL    |       |
| ProductSKU            | text   | YES  |     | NULL    |       |
| ProductName           | text   | YES  |     | NULL    |       |
| ModelName             | text   | YES  |     | NULL    |       |
| ProductDescription    | text   | YES  |     | NULL    |       |
| ProductColor          | text   | YES  |     | NULL    |       |
| ProductSize           | text   | YES  |     | NULL    |       |
| ProductStyle          | text   | YES  |     | NULL    |       |
| ProductCost           | double | YES  |     | NULL    |       |
| ProductPrice          | double | YES  |     | NULL    |       |
+-----------------------+--------+------+-----+---------+-------+
11 rows in set (0.18 sec)
```

FLOAT VS DOUBLE VS DECIMAL()

```
float -> 1999.99
Double -> 1999.00
Decimal -> 1999.99
```

```
Float(p) -> Approximate Float

Double(p,s) -> Larger Float
Decimal(p,s) -> Exact Float
```

Float is storing data more efficiently but not being much precise.

Float - 8 bits
Double - 16 bits [More Precise Value]

```
Decimal(5,2) -> 999.99
```

Total 5 digits including 2 decimals

```
-- Challenge 1 : Find the Product Details having ProductPrice > 1000.
SELECT
        ProductName,
    ProductColor,
    ProductStyle,
    ProductPrice
FROM Products
WHERE ProductPrice > 1000; -- row filter
```

ProductName	ProductColor	ProductStyle	ProductPrice
HL Road Frame - Red, 62	Red	U	1263.4598
HL Road Frame - Red, 44	Red	U	1263.4598
HL Road Frame - Red, 48	Red	U	1263.4598
HL Road Frame - Red, 52	Red	U	1263.4598
HL Road Frame - Red, 56	Red	U	1263.4598
HL Mountain Frame - Silver, 42	Silver	U	1204.3248
HL Mountain Frame - Silver, 44	Silver	U	1364.5
HL Mountain Frame - Silver, 48	Silver	U	1364.5
HL Mountain Frame - Silver, 46	Silver	U	1204.3248
HL Mountain Frame - Black, 42	Black	U	1191.1739
HL Mountain Frame - Black, 44	Black	U	1349.6
HL Mountain Frame - Black, 48	Black	U	1349.6
HL Mountain Frame - Black, 46	Black	U	1191.1739

```
mysql> SELECT
    ->  ProductName,
    ->     ProductColor,
    ->     ProductStyle,
    ->     ProductPrice
    -> FROM Products
    -> WHERE ProductPrice > 1000 -- row filter
    -> LIMIT 10;
+--------------------------------+--------------+--------------+--------------+
| ProductName                    | ProductColor | ProductStyle | ProductPrice |
+--------------------------------+--------------+--------------+--------------+
| HL Road Frame - Red, 62        | Red          | U            |    1263.4598 |
| HL Road Frame - Red, 44        | Red          | U            |    1263.4598 |
| HL Road Frame - Red, 48        | Red          | U            |    1263.4598 |
| HL Road Frame - Red, 52        | Red          | U            |    1263.4598 |
| HL Road Frame - Red, 56        | Red          | U            |    1263.4598 |
| HL Mountain Frame - Silver, 42 | Silver       | U            |    1204.3248 |
| HL Mountain Frame - Silver, 44 | Silver       | U            |       1364.5 |
| HL Mountain Frame - Silver, 48 | Silver       | U            |       1364.5 |
| HL Mountain Frame - Silver, 46 | Silver       | U            |    1204.3248 |
| HL Mountain Frame - Black, 42  | Black        | U            |    1191.1739 |
+--------------------------------+--------------+--------------+--------------+
10 rows in set (0.00 sec)
```

Challenge 2 : Find the unique Occupation available in Customers?

Hint : DISTINCT

```
60 •  SELECT
61        DISTINCT Occupation
62    FROM Customers;
63
```

Result Grid | Filter Rows: | Exp

Occupation
Professional
Management
Skilled Manual
Clerical
Manual

```
mysql> SELECT
    ->  DISTINCT Occupation
    -> FROM Customers;
+----------------+
| Occupation     |
+----------------+
| Professional   |
| Management     |
| Skilled Manual |
| Clerical       |
| Manual         |
+----------------+
5 rows in set (0.17 sec)
```

Where Clause

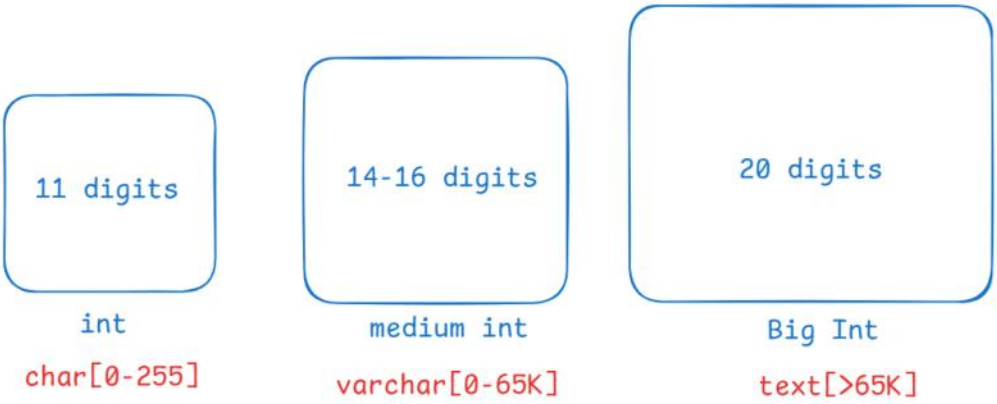
```
SELECT
    FirstName,
    EmailAddress,
    AnnualIncome,
    EducationLevel,
    Occupation
FROM Customers
WHERE Occupation = 'Skilled Manual';
```

FirstName	EmailAddress	AnnualIncome	EducationLevel	Occupation
CHLOE	chloe23@learnsector.com		Partial College	Skilled Manual
WYATT	wyatt32@learnsector.com	$30,000	Partial College	Skilled Manual
SHANNON	shannon1@learnsector.com	$20,000	High School	Skilled Manual
LUKE	luke18@learnsector.com	$40,000	High School	Skilled Manual
JORDAN	jordan73@learnsector.com	$40,000	High School	Skilled Manual
DESTINY	destiny7@learnsector.com	$40,000	Partial College	Skilled Manual
ETHAN	ethan20@learnsector.com	$40,000	Partial College	Skilled Manual
SETH	seth46@learnsector.com	$40,000	Partial College	Skilled Manual
RUSSELL	russell7@learnsector.com	$60,000	Partial College	Skilled Manual
THERESA	theresa13@learnsector.com	$20,000	High School	Skilled Manual
DENISE	denise10@learnsector.com	$20,000	High School	Skilled Manual
JAIME	jaime41@learnsector.com	$20,000	High School	Skilled Manual
EBONY	ebony19@learnsector.com	$20,000	High School	Skilled Manual

```
mysql> SELECT
    ->  FirstName,
    ->     EmailAddress,
    ->     AnnualIncome,
    ->     EducationLevel,
    ->     Occupation
    -> FROM Customers
    -> WHERE Occupation = 'Skilled Manual'
    -> LIMIT 10;
+-----------+---------------------------+--------------+-----------------+----------------+
| FirstName | EmailAddress              | AnnualIncome | EducationLevel  | Occupation     |
+-----------+---------------------------+--------------+-----------------+----------------+
| CHLOE     | chloe23@learnsector.com   |              | Partial College | Skilled Manual |
| WYATT     | wyatt32@learnsector.com   | $30,000      | Partial College | Skilled Manual |
| SHANNON   | shannon1@learnsector.com  | $20,000      | High School     | Skilled Manual |
| LUKE      | luke18@learnsector.com    | $40,000      | High School     | Skilled Manual |
| JORDAN    | jordan73@learnsector.com  | $40,000      | High School     | Skilled Manual |
| DESTINY   | destiny7@learnsector.com  | $40,000      | Partial College | Skilled Manual |
| ETHAN     | ethan20@learnsector.com   | $40,000      | Partial College | Skilled Manual |
| SETH      | seth46@learnsector.com    | $40,000      | Partial College | Skilled Manual |
| RUSSELL   | russell7@learnsector.com  | $60,000      | Partial College | Skilled Manual |
| THERESA   | theresa13@learnsector.com | $20,000      | High School     | Skilled Manual |
+-----------+---------------------------+--------------+-----------------+----------------+
10 rows in set (0.11 sec)
```

```
-- I'm a LIC Agent, I wanted to sell the Policy to whom?
-- ['Married'['M'],'Male' ['M'],'Professional',should be a HomeOwner]

SELECT
    FirstName,
    MaritalStatus,
    Gender,
    Occupation,
    HomeOwner
FROM Customers
WHERE MaritalStatus = 'M'
    AND Gender = 'M'
    AND Occupation = 'Professional'
    AND HomeOwner = 'Y';
```

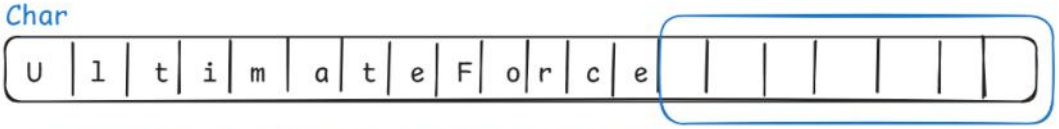
FirstName	MaritalStatus	Gender	Occupation	HomeOwner
JON	M	M	Professional	Y
RUBEN	M	M	Professional	Y
MARCO	M	M	Professional	Y
CURTIS	M	M	Professional	Y
RYAN	M	M	Professional	Y
DALTON	M	M	Professional	Y
ANDRÃ‰S	M	M	Professional	Y
ADAM	M	M	Professional	Y
MICHEAL	M	M	Professional	Y
LUIS	M	M	Professional	Y
BLAKE	M	M	Professional	Y
EDWARD	M	M	Professional	Y
CHARLES	M	M	Professional	Y
NATHAN	M	M	Professional	Y
DENNIS	M	M	Professional	Y
TODD	M	M	Professional	Y
ANGEL	M	M	Professional	Y



Char() VS Varchar() VS Text()

char >> varchar() -> If the input is always of fixed length

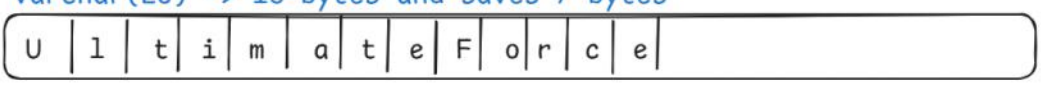
Char



Memory

char(20)
varchar(20)
char = 20 bytes
varchar = 13 bytes

varchar(20) -> 13 bytes and saves 7 bytes




```
username1 = 'UltimateForce'
username2 = 'Tryantatom'
username3 = 'victor'
username4 = 'Demonslayer'
username5 = 'Maverick'
```

Gender -> Char(1) -> 'F'/'M'

Female
Male
→ 'F' & 'M' and other characters are trunctated.

Marital Status -> char(1) -> 'M'/'U'/'S'/'D'

Married
Unmarried
Single
Divorce

State-Abbreviations : Number Place -> Char(2)

```
HR - Haryana
DL - Delhi
MH - Maharashtra
HP - Himachal Pradesh
UP - Uttar Pradesh
RJ - Rajasthan
UK - Uttarakhand
OD - Odisha
TN - Tamil Nadu
KL - Kerala
MP - Madhya Pradesh
WB - West Bengal
```

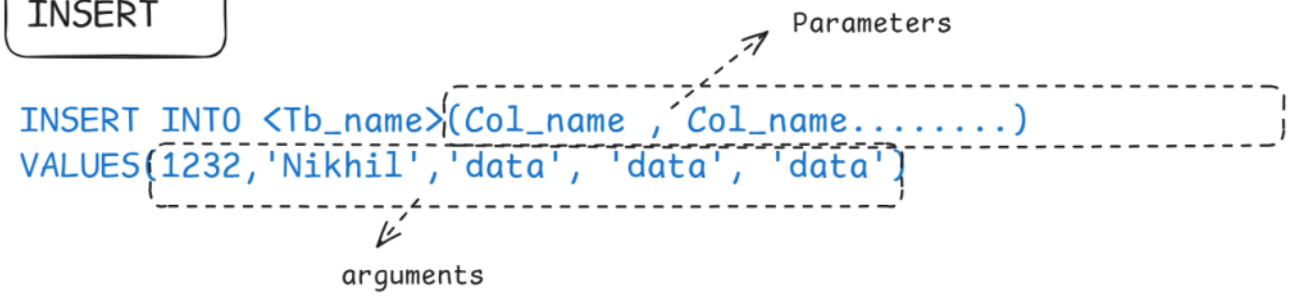
```
Syntax:
        CREATE TABLE <tb_name>(
            col_name data_type Constraints,
            col_name data_type Constraints,
            col_name data_type Constraints
    );
```

```
-- PRIMARY KEY -> UNIQUE + NOT NULL
CREATE TABLE Students(
        stud_id int PRIMARY KEY AUTO_INCREMENT, -- Mandatory
        stud_name VARCHAR(50) NOT NULL, -- Mandatory
        email_address VARCHAR(50) NOT NULL,
    gender CHAR(1),
    phone_number int DEFAULT 0, -- 11 digit
    course VARCHAR(15)
);

DESCRIBE Students;
```

```
mysql> DESCRIBE Students;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| stud_id       | int         | NO   | PRI | NULL    | auto_increment |
| stud_name     | varchar(50) | NO   |     | NULL    |                |
| email_address | varchar(50) | NO   |     | NULL    |                |
| gender        | char(1)     | YES  |     | NULL    |                |
| phone_number  | int         | YES  |     | 0       |                |
| course        | varchar(15) | YES  |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
6 rows in set (0.23 sec)
```

```
mysql> SELECT * FROM Students;
Empty set (0.23 sec)
```

INSERT



Types of Insert :
1. Single Insertion.
2. Multiline Insertion.

```
110 •  INSERT INTO students(stud_name, email_address, gender, course)
111    VALUES('Deepak Chhikara', 'deepak@gmail.com', 'M', 'Data Analyst');
112
113 •  SELECT * FROM students;
114
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

stud_id	stud_name	email_address	gender	phone_number	course
1	Deepak Chhikara	deepak@gmail.com	M	0	Data Analyst
NULL	NULL	NULL	NULL	NULL	NULL

```
115 •  INSERT INTO students(stud_name, email_address, gender, course)
116    VALUES('Aditya Sharma', 'aditya.sharma@gmail.com', 'M', 'Data Science'),
117    ('Ankita Dutta','ankita@yahoo.com','F','Data Engineer'),
118    ('Nikhil', 'nikhil@outlook.in','M','AI Engineer');
119
120 •  SELECT * FROM Students;
121
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

stud_id	stud_name	email_address	gender	phone_number	course
1	Deepak Chhikara	deepak@gmail.com	M	0	Data Analyst
5	Aditya Sharma	aditya.sharma@gmail.com	M	0	Data Science
6	Ankita Dutta	ankita@yahoo.com	F	0	Data Engineer
7	Nikhil	nikhil@outlook.in	M	0	AI Engineer
NULL	NULL	NULL	NULL	NULL	NULL

```
mysql> SELECT * FROM Students;
+---------+----------------+-------------------------+--------+--------------+---------------+
| stud_id | stud_name      | email_address           | gender | phone_number | course        |
+---------+----------------+-------------------------+--------+--------------+---------------+
|       1 | Deepak Chhikara | deepak@gmail.com        | M      |            0 | Data Analyst  |
|       5 | Aditya Sharma  | aditya.sharma@gmail.com | M      |            0 | Data Science  |
|       6 | Ankita Dutta   | ankita@yahoo.com        | F      |            0 | Data Engineer |
|       7 | Nikhil         | nikhil@outlook.in       | M      |            0 | AI Engineer   |
+---------+----------------+-------------------------+--------+--------------+---------------+
4 rows in set (0.14 sec)
```

```
DELETE FROM Students; -- Remove the data From Students;

/* Error Code: 1175. You are using safe update mode and you tried to update a table
without a WHERE that uses a KEY column.  To disable safe mode, toggle the option in
Preferences -> SQL Editor and reconnect.
*/
SET SQL_SAFE_UPDATES = 0; -- Settings Changed
```

```
DELETE FROM Students; -- Remove all the data From Students;

SELECT * FROM Students;

INSERT INTO students(stud_id ,stud_name, email_address, gender, course)
VALUES(729249,'Deepak Chhikara', 'deepak@gmail.com', 'M', 'Data Analyst');

SELECT * FROM students;

INSERT INTO students(stud_name, email_address, gender, course)
VALUES('Aditya Sharma', 'aditya.sharma@gmail.com', 'M', 'Data Science'),
('Ankita Dutta','ankita@yahoo.com','F','Data Engineer'),
('Nikhil', 'nikhil@outlook.in','M','AI Engineer');
```

stud_id	stud_name	email_address	gender	phone_number	course
729249	Deepak Chhikara	deepak@gmail.com	M	0	Data Analyst
729250	Aditya Sharma	aditya.sharma@gmail.com	M	0	Data Science
729251	Ankita Dutta	ankita@yahoo.com	F	0	Data Engineer
729252	Nikhil	nikhil@outlook.in	M	0	AI Engineer
NULL	NULL	NULL	NULL	NULL	NULL

```
mysql> SELECT * FROM Students;
+---------+-----------------+-------------------------+--------+--------------+---------------+
| stud_id | stud_name       | email_address           | gender | phone_number | course        |
+---------+-----------------+-------------------------+--------+--------------+---------------+
|  729249 | Deepak Chhikara | deepak@gmail.com        | M      |            0 | Data Analyst  |
|  729250 | Aditya Sharma   | aditya.sharma@gmail.com | M      |            0 | Data Science  |
|  729251 | Ankita Dutta    | ankita@yahoo.com        | F      |            0 | Data Engineer |
|  729252 | Nikhil          | nikhil@outlook.in       | M      |            0 | AI Engineer   |
+---------+-----------------+-------------------------+--------+--------------+---------------+
4 rows in set (0.23 sec)
```