

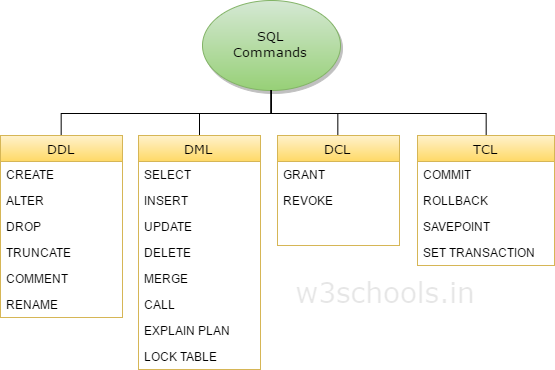
**Department of Computer Science and Engineering**

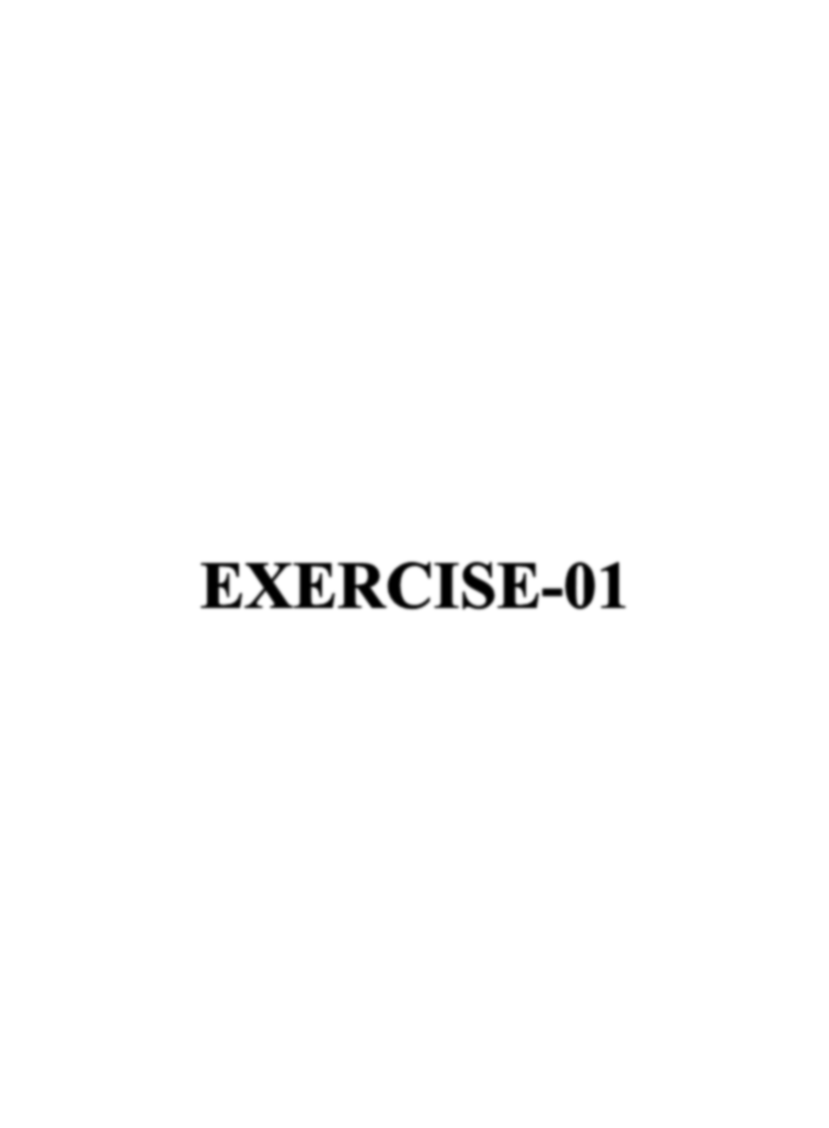
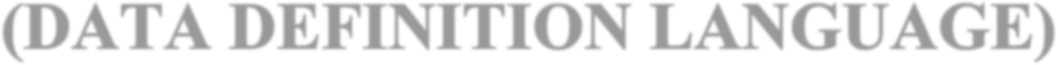
**19CSE101 COMPUTER SYSTEM ESSENTIALS LAB MANUAL**

###### MYSQL

It is RDBMS (Relational Data Base Management System). It is open-source database software.

Commands under MYSQL





#### What is DDL?

DDL means ‘DATA DEFINITION LANGUAGE’. It is a syntax used for creating and modifying database objects such as table indexes, and users.

* Commands under DDL

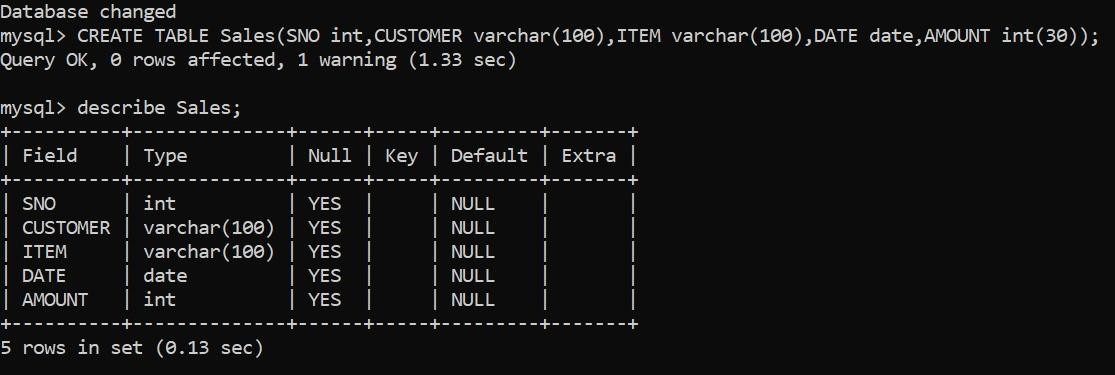
##### CREATE

It is a command used to ***create*** a new table in the database.

**Syntax:** CREATE TABLE table\_name ( column\_name1 datatype1(size), column\_name2 datatabe2,

…..);

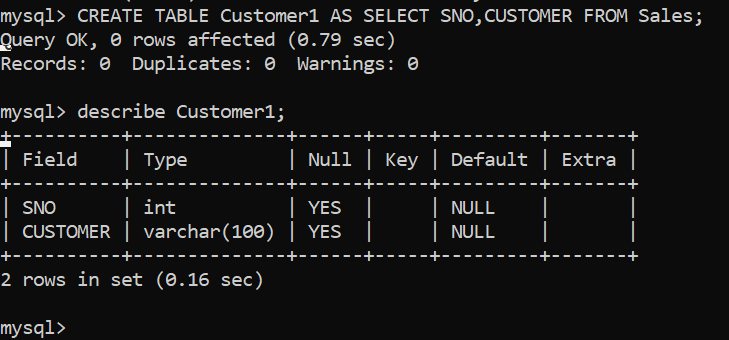
Example:



* + - **Create Table using another table. Syntax:** CREATE TABLE new\_table\_name AS

SELECT column\_name1,column\_name2,…

FROM existing\_table\_name WHERE….;

Example:

Here we created a table from the table ‘Sales’ in the above example.

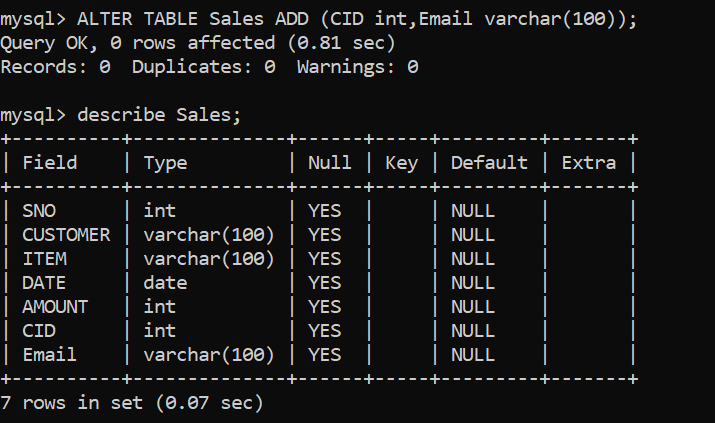
##### ALTER

It is a command used to ***add, delete or modify*** the columns in an existing table.

* **ADD COLUMN**

It is used add a column to the existing table.

**Syntax:** ALTER TABLE table\_name ADD column\_name datatype;

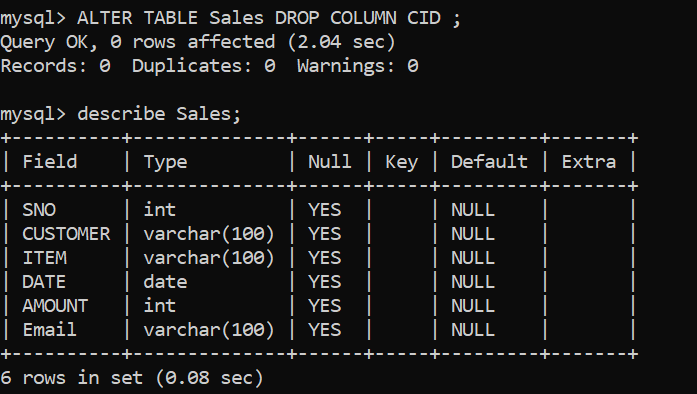
Example:

Here we added new columns ‘CID (customer ID), email’ into the table Sales.

* **DROP COLUMN**

It is used to delete a column form existing table.

**Syntax:** ALTER TABLE table\_name DROP COLUMN column\_name;

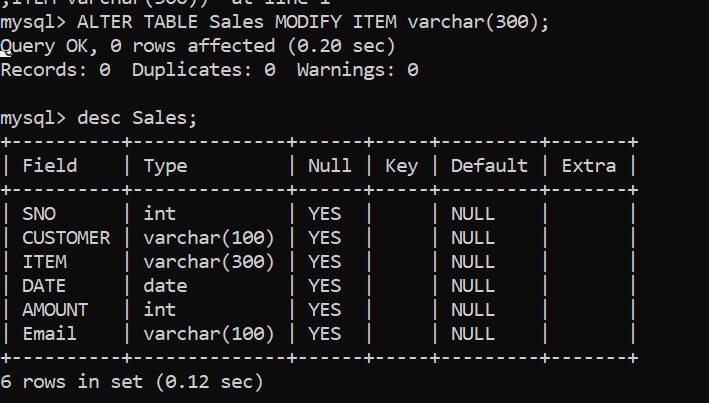
Example:

Here we deleted ‘CID’ column from table Sales.

* **MODIFY COLUMN**

It is used to alter or modify a column from existing table.

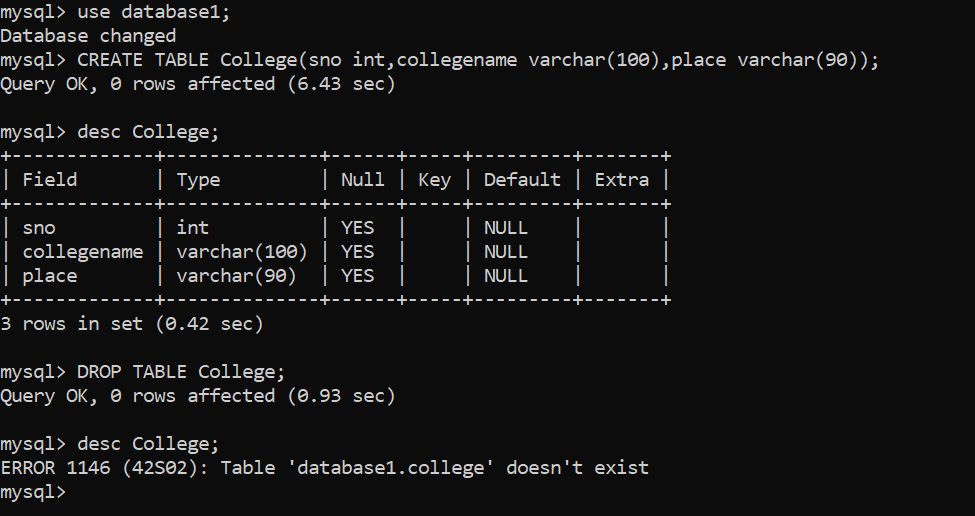
**Syntax:** ALTER TABLE table\_name MODIFY column\_name datatype;

Example:

##### DROP

It is a command used to ***remove*** table definition and content from database.

**Syntax:** DROP TABLE table\_name; Example:

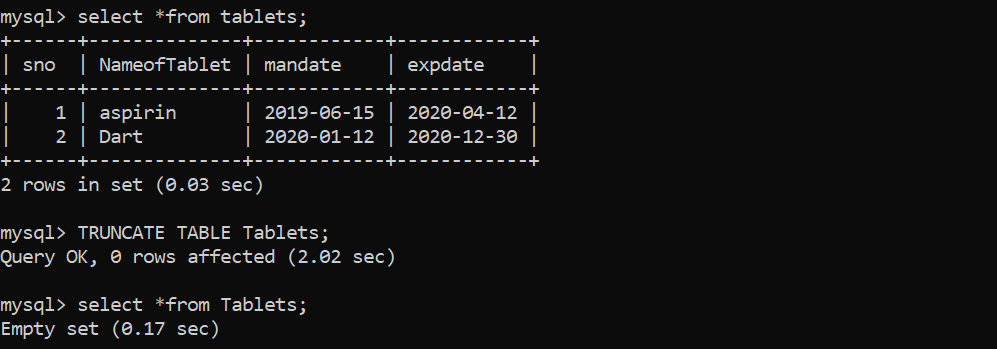
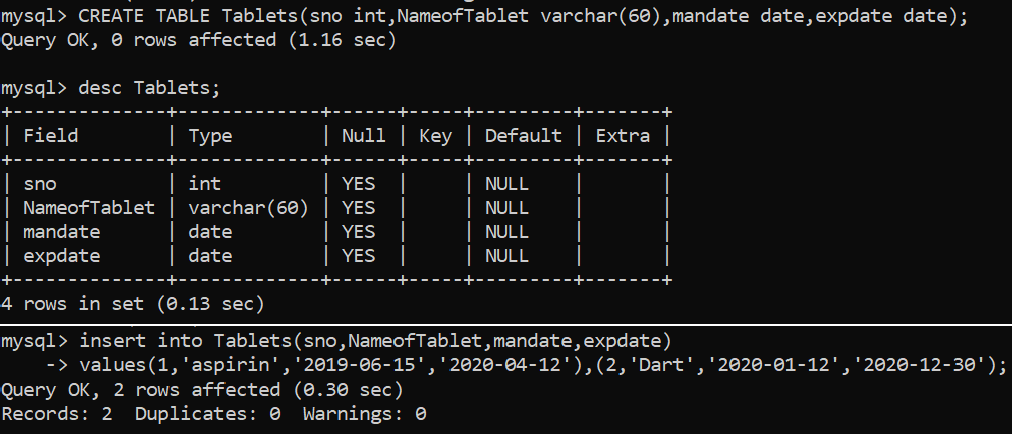


##### TRUNCATE

It is a command used to ***delete*** all the ***rows*** from the table.

**Syntax**: TRUNCATE TABLE table\_name;

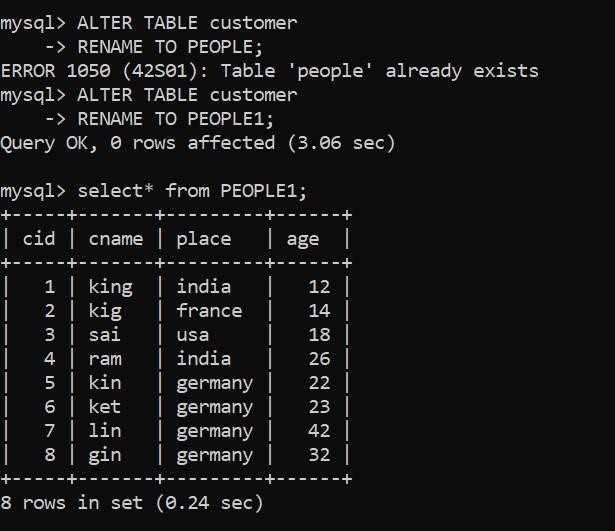
Example:

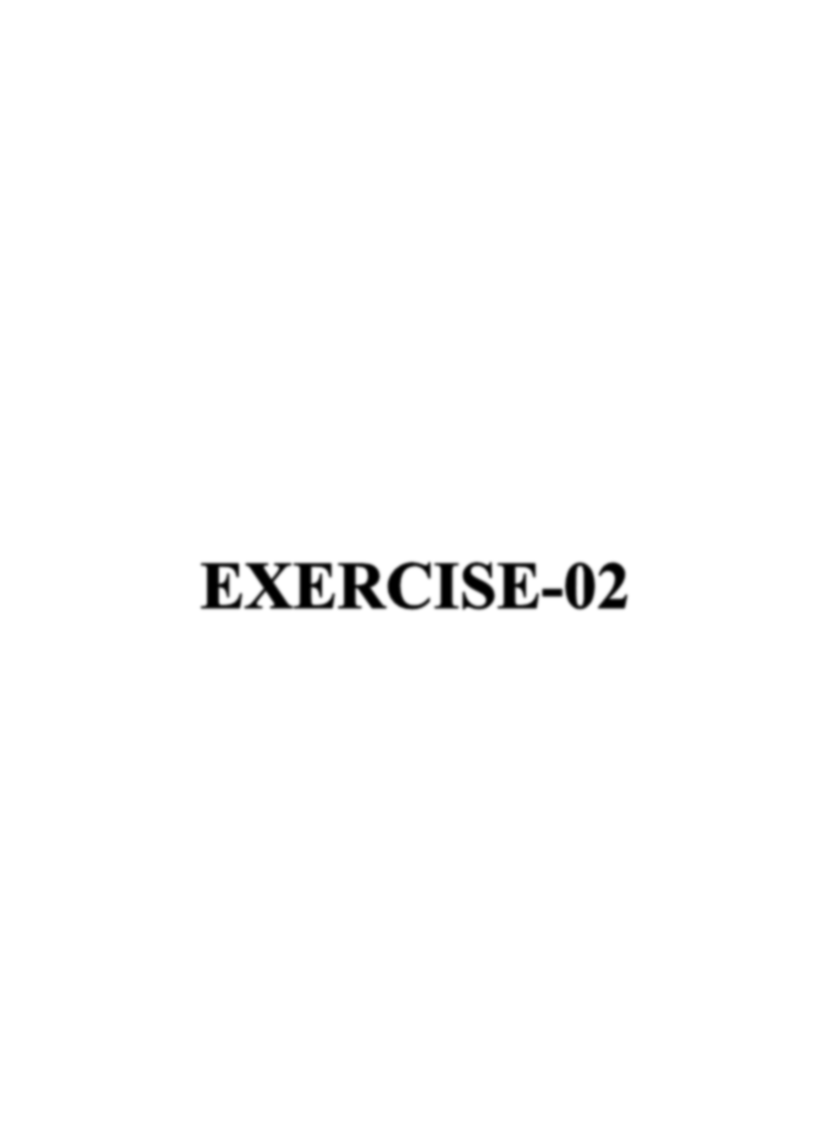
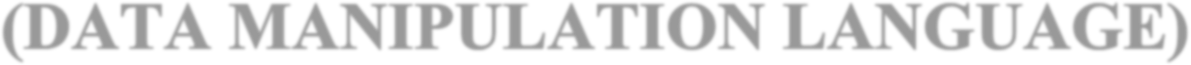


##### RENAME

It is command used to rename the existing table.

**Syntax:** ALTER TABLE table\_name RENAME TO new\_table\_name;

Example:



What is DML?

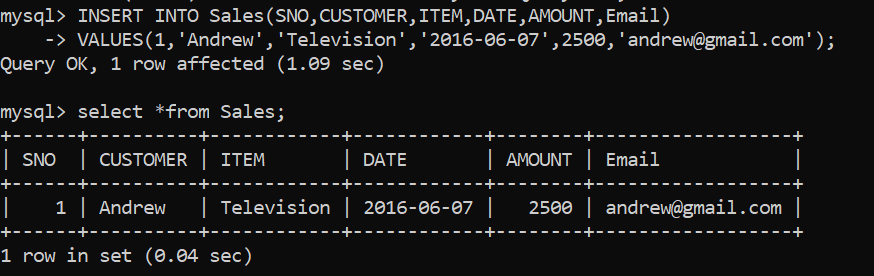
DML means ‘DATA MANIPULATION LANGUAGE’. DML statements are used access or manipulate data in existing tables.

* **Commands** Under DML

##### INSERT

It is a command used insert new records into an existing table.

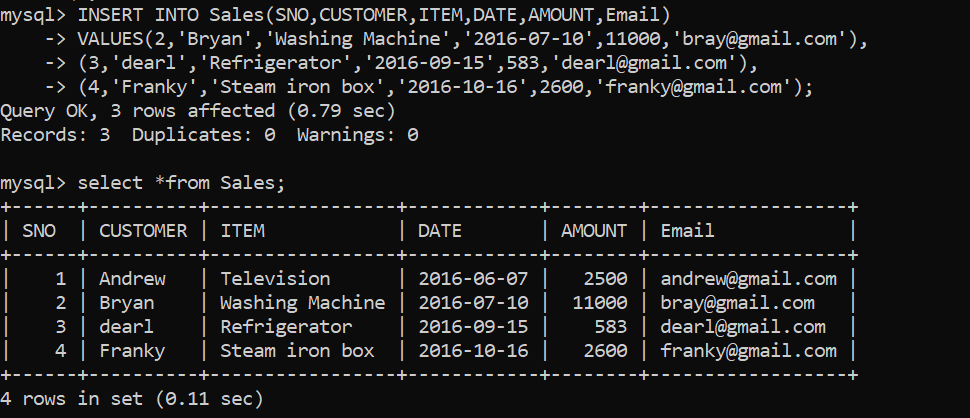
**Syntax:** INSERT INTO table\_name ( column\_name1, column\_name2…….) VALUES (value1, value2……);

Example:

* + To insert more than one column at a time

**Syntax:** INSERT INTO table\_name ( column\_name1, column\_name2…….)

VALUES (value1, value2……), (value1, value2……);

Example:

##### UPDATE

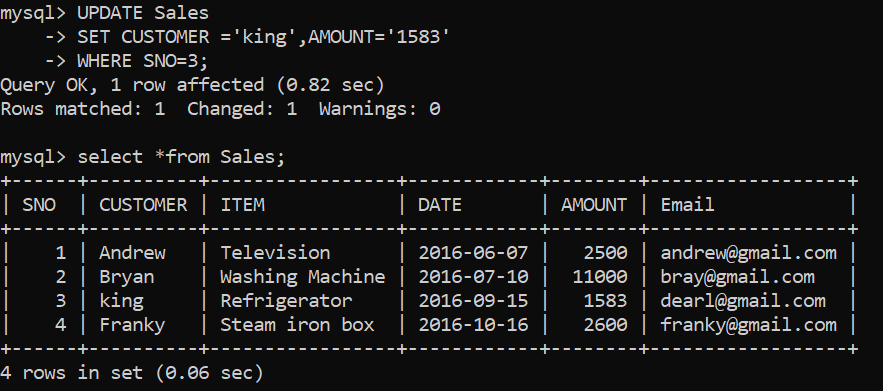
It is a command used to modify existing records in table.

**Syntax:** UPDATE table\_name

SET column\_name1=value1, column\_name2=value2, WHERE condition;

* + If you omit the WHERE clause, all records in the table will be updated.

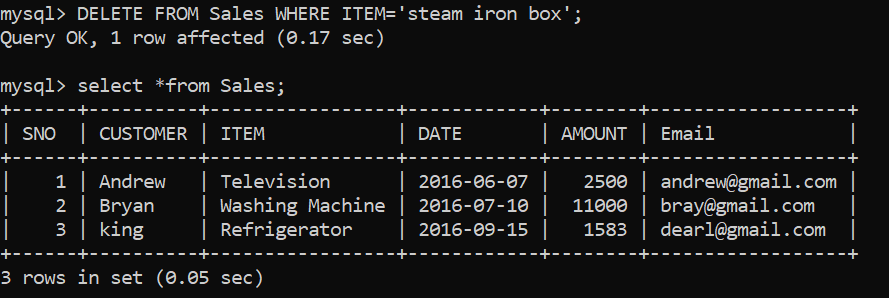
Example:



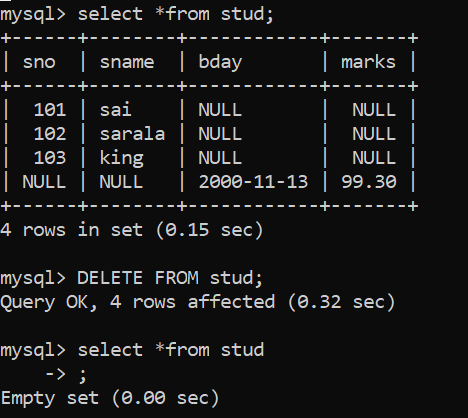
##### DELETE

It is a command used to delete the existing records in the table.

**Syntax:** DELETE FROM table\_name WHERE condition;

Example:

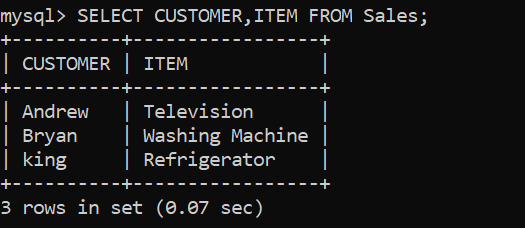
* + To delete all records at a time. **Syntax:** DELETE FROM table\_name; Example:



##### SELECT

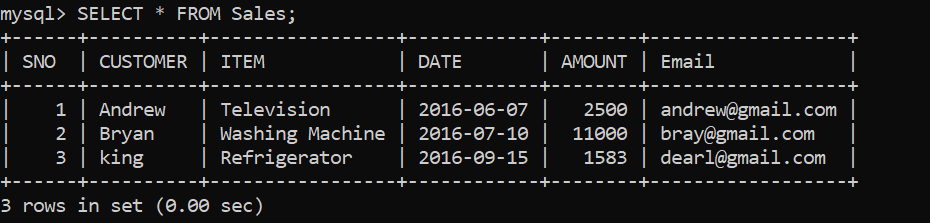
It is command used to select the data from database.

**Syntax:** SELECT column\_name1, column\_name FROM table\_name;

Example:

* + If you want to select all the fields available in the table.

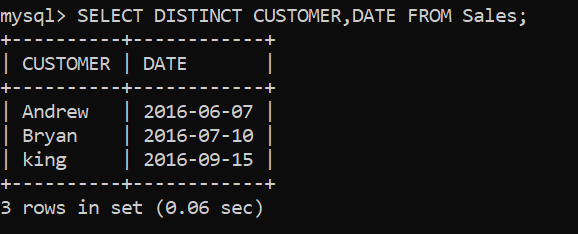
**Syntax**: SELECT \* FROM table\_name; Example:



* + To select distinct from a table.

**Syntax**: SELECT DISTINCT column\_name1, column\_name2 FROM table\_name;

Example:





###### OPERATORAS IN WHERE CLAUSE [DATE: 17-10-2020]

The following Operators can be used in the WHERE clause:

1. **“=” Equal**
2. **“>” Greater Than**
3. **“<” Less Than**
4. **“>=” Greater than or Equal**
5. **“<=” Less Than or Equal**
6. **“<>” Not equal**

The WHERE clause can be combined with AND, OR, and NOT operators.

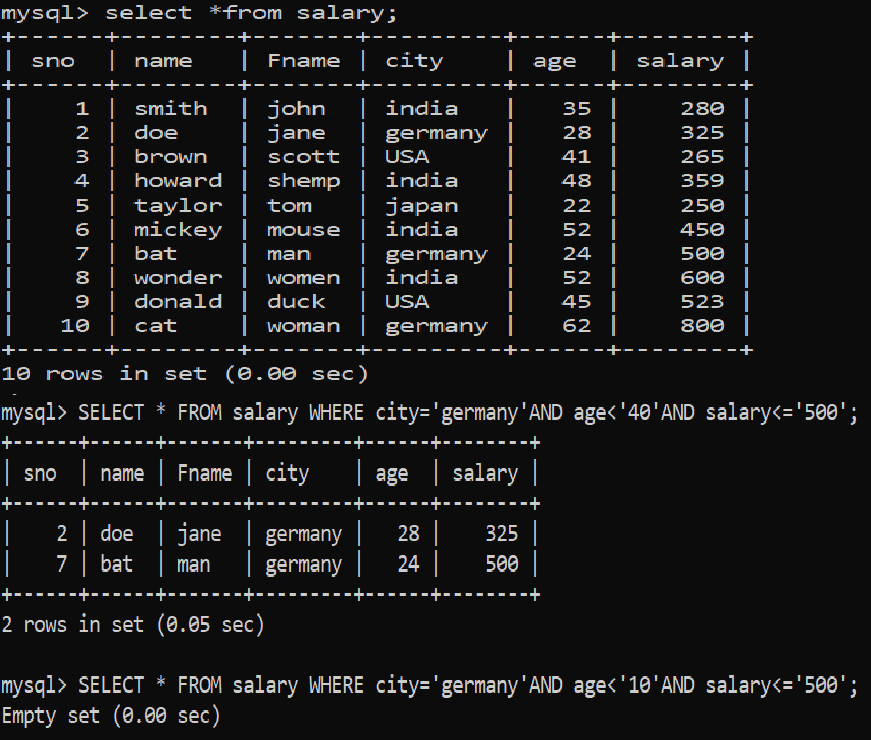
##### AND OPERATOR

The AND operator is used to filter records based on more than one condition.

* + AND operator displays a record if all the conditions which are separated by AND are TRUE.

**Syntax:** SELECT column1, column2… FROM table\_name WHERE condition1 AND condition2 AND …;

Example:



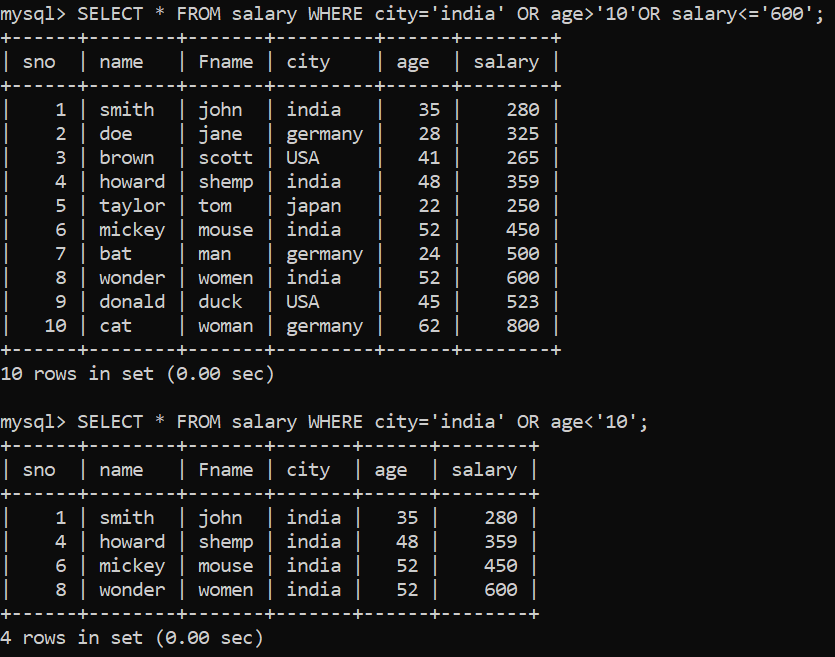
##### OR OPERATOR

The OR operator is used to filter records based on more than one condition in the database tables.

* + OR operator displays a record if any of the conditions separated by OR is TRUE.

**Syntax:** SELECT column1, column2 … FROM table\_name WHERE condition1 OR condition2 OR….;

Example:

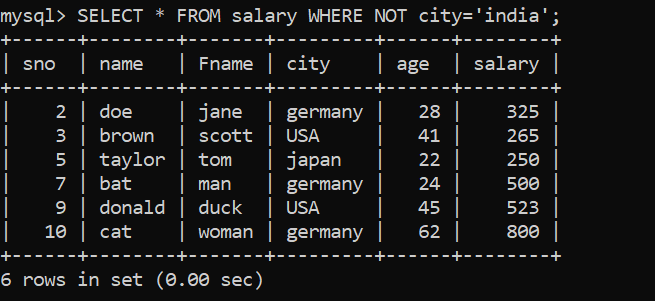


##### NOT OPERATOR

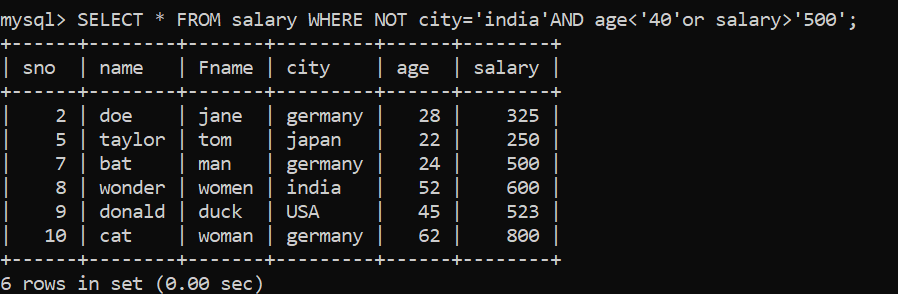
The NOT operator is used to filter the record which are not belong to the given condition from existing tables.

**Syntax:** SELECT column1, column2… FROM table\_name WHERE NOT condition;

Example:



* + Example by combining AND, OR and NOT operator:



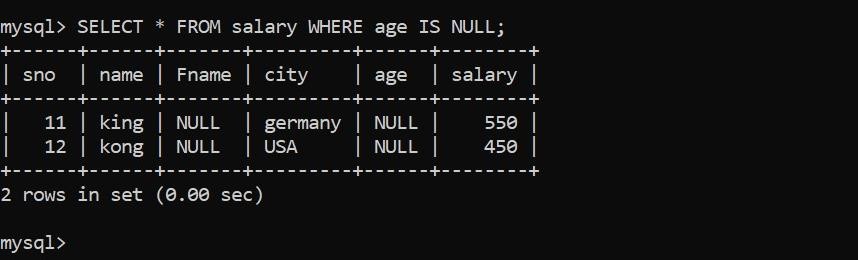
###### NULL & NOT NULL COMMANDS:

A field with a NULL value is a field with no value. It is not possible to test for NULL values with comparison operators, such as =, <, or<>.

We have to use the IS NULL and IS NOT NULL operators instead.

**IS NULL Syntax:** SELECT column\_name FROM table\_name WHERE column\_name IS NULL;

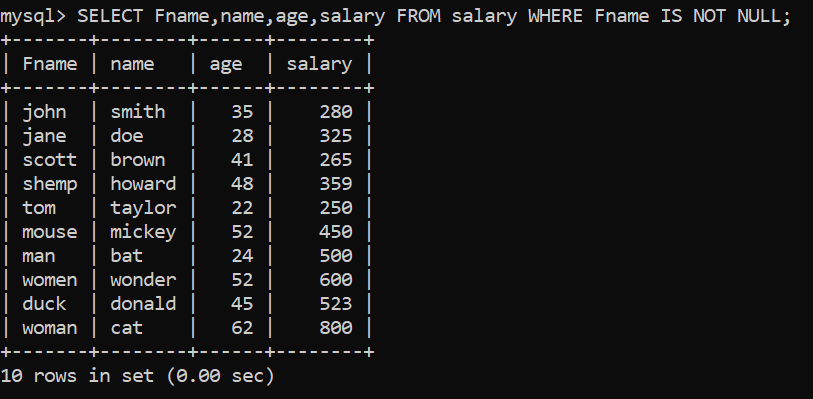
Example:



**IS NOT NULL Syntax:** SELECT column\_name FROM table \_name

WHERE column\_name IS NOT NULL;

Example:



###### LIKE OPERATOR

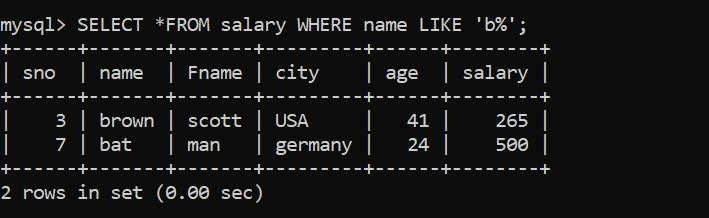
The LIKE operator is used in a WHERE clause search for a specified pattern in a column.

* + There are two main symbols used with like operator:
    - **“%”-** The percentage sign represents zero, one, or multiple characters.
    - **“\_”** - The underscore represents a single character.
  + We can also combined any no of conditions using AND or OR operators.

**Syntax:** SELECT column1, column2,…. FROM table\_name WHERE column LIKE pattern;

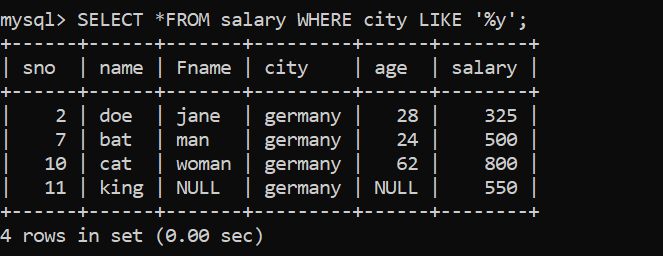
1. WHERE column LIKE ‘x%’. Finds any values that start with “x”.

Example:



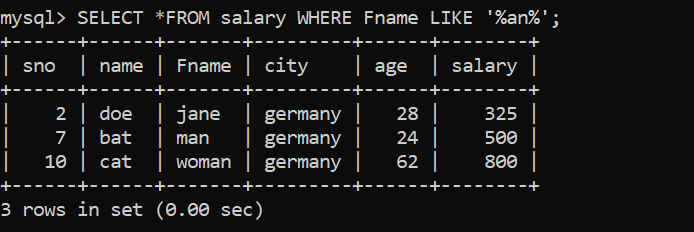
1. WHERE column LIKE ‘%x’. Finds any values that end with “x”.

Example:



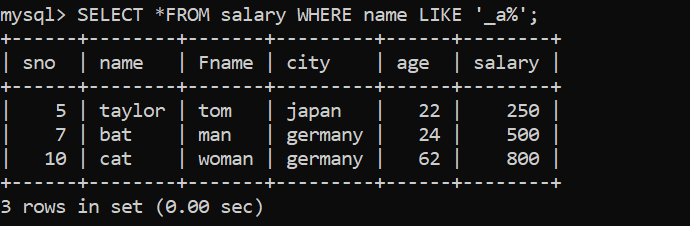
1. WHERE column LIKE ‘%as%’. Finds any values that have “as” in any position.

Example:



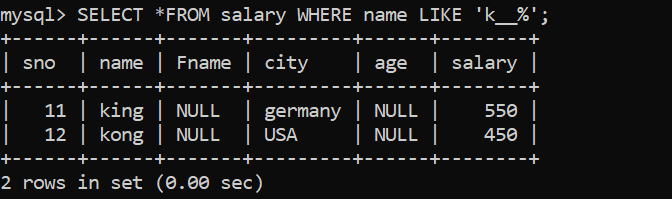
1. WHERE column LIKE ‘\_r%’. Finds any values that have “r” in the second position.

Example:



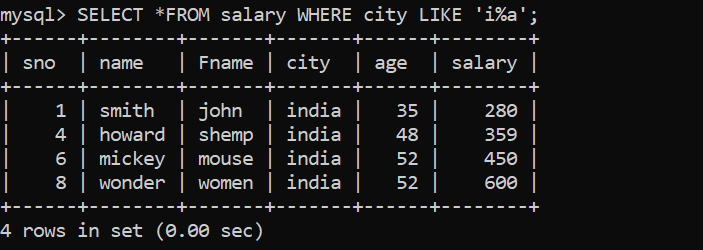
1. WHERE column LIKE ‘a %’. Finds any values that start with “a” and are at least 3 characters in length.

Example:



1. WHERE column LIKE ‘a%o’. Finds any values that start with “a” and ends with “o”**.**

Example**:**

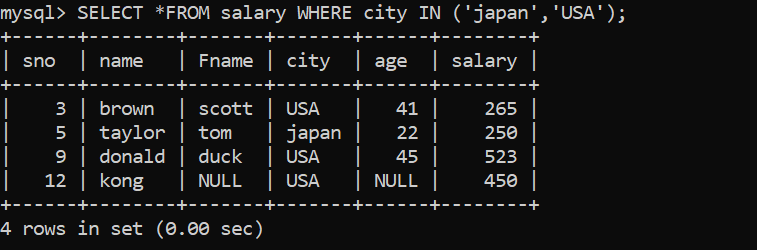


###### IN OPERATOR:

IN operator allows you to specify multiple values in a WHERE clause.

**Syntax**: SELECT column\_name(s) FROM table\_name WHERE column \_name IN (value1,value2,….);

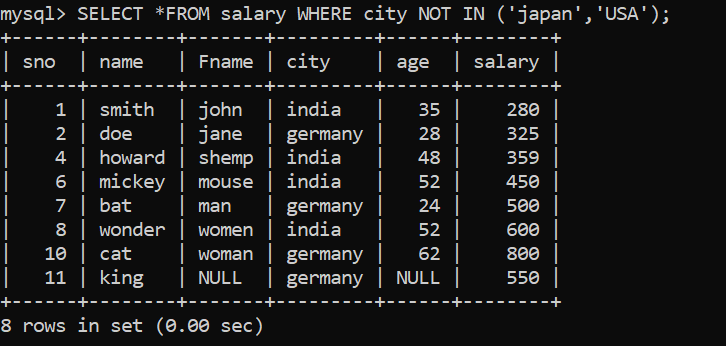
Example:



* + We can use ‘NOT IN’ operator to select the values which are not belongs to that element.

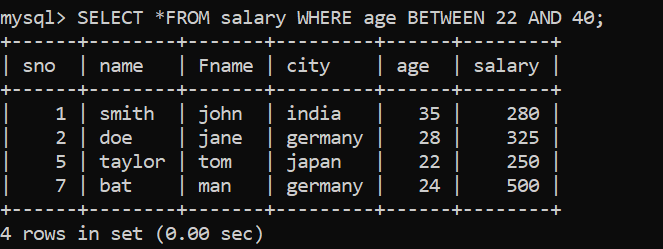
**Syntax:** SELECT column\_name(s) FROM table\_name WHERE column \_name NOT IN (value1,value2,….);

Example:



**Syntax**: SELECT column\_name(s) FROM table\_name

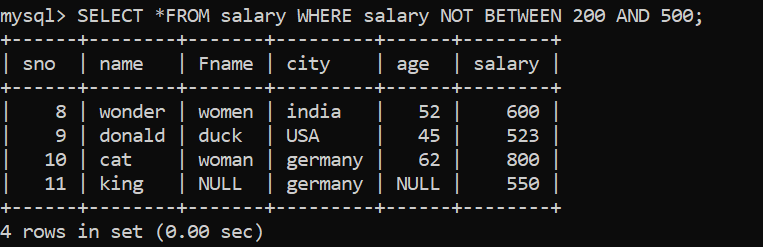
WHERE column \_name BETWEEN value1 AND value; Example:



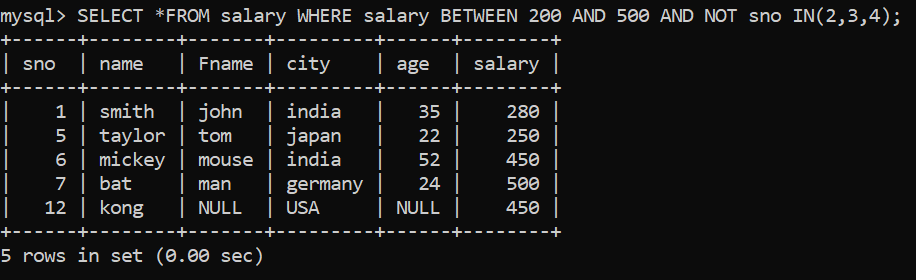
* + We can use ‘NOT BETWEEN’ operator to select the values which are not belongs to that range.

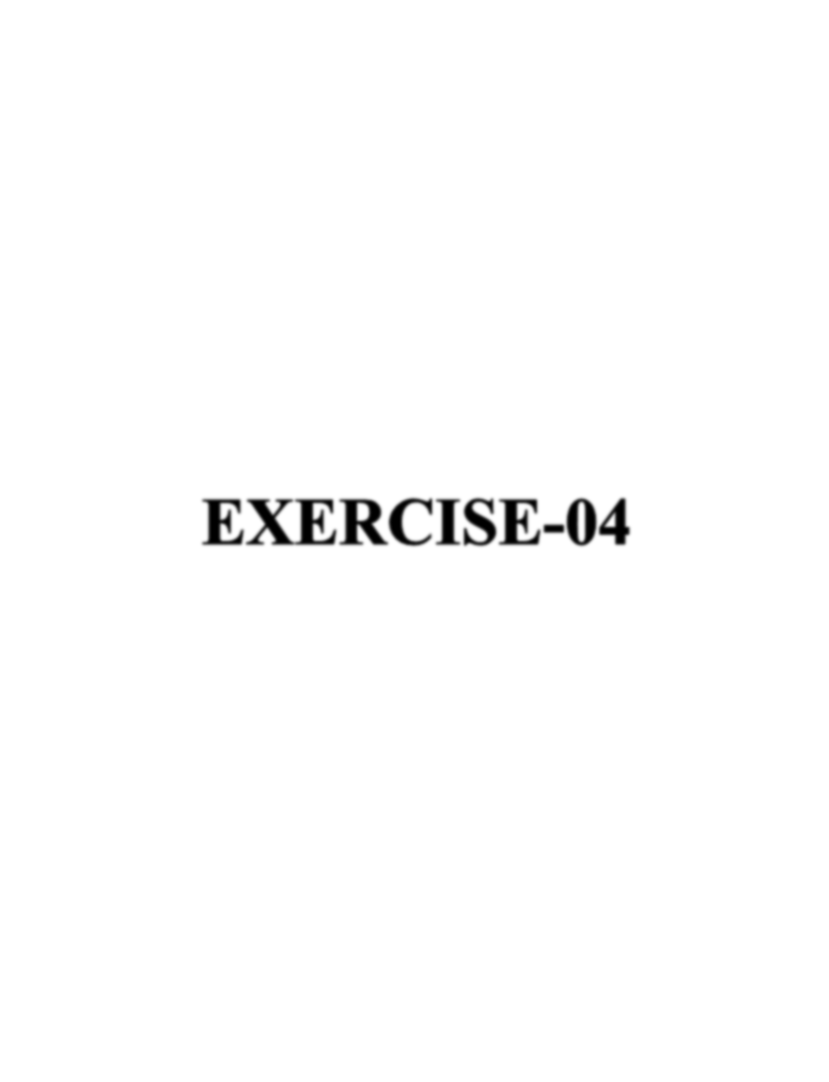
**Syntax**: SELECT column\_name(s) FROM table\_name

WHERE column \_name NOT BETWEEN value1 AND value; Example:



* + BETWEEN with IN example:





* **CREATE CONSTRAINTS**

Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.

**Syntax:** CREATE TABLE table\_name ( Column1 datatype constraint, Column2 datatype constraint, Column3 datatype constraint,……

);

* + Constraints can be column level (apply only for specific column) and table level (apply for whole table).
  + Commonly used constraints:

1. NOT NULL
2. UNIQUE
3. PRIMARY KEY
4. FOREIGN KEY
5. CHECK
6. DEFAULT
7. INDEX

###### NOT NULL CONSTRAINT

It ensures that a column cannot have a NULL value.

**Syntax:** CREATE TABLE table\_name ( Column1 datatype NOT NULL, Column2 datatype NOT NULL,…);

Example:

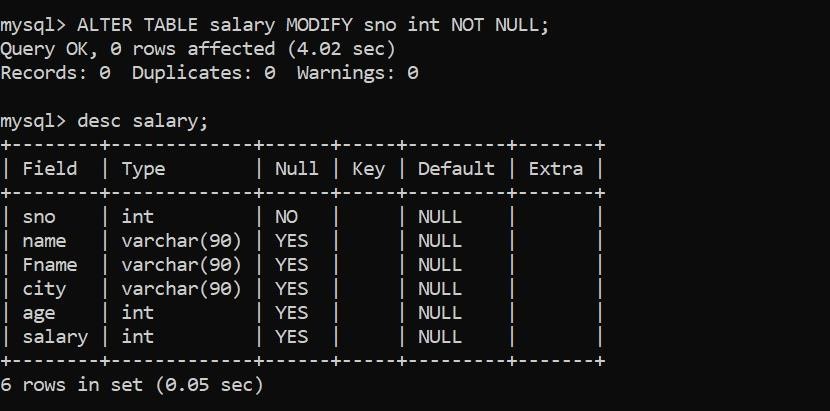


* + NOT NULL on ALTER TABLE

**Syntax:** ALTER TABLE table\_name

MODIFY column1 datatype NOT NULL;

Example:



###### UNIQUE CONSTRAINT

It ensures that all the values in a column are different.

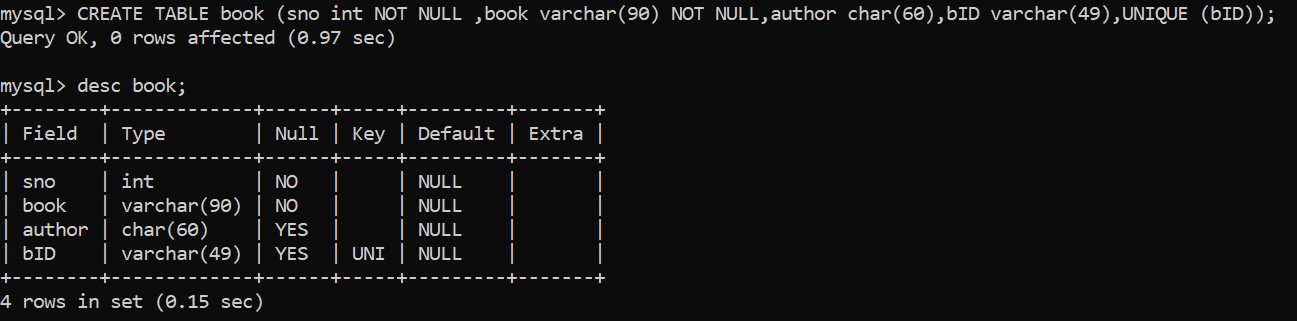
We can have many UNIQUE constraints per table.

**Syntax:** CREATE TABLE table\_name ( Column1 datatype , Column2 datatype ,

UNIQUE (column1)

);

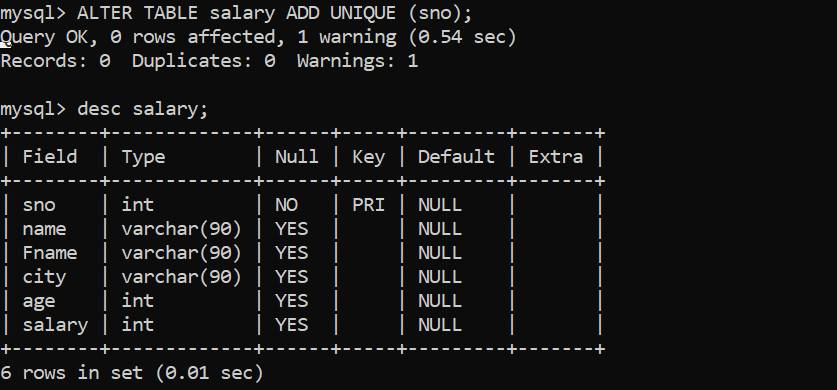
Example:



* UNIQUE on ALTER TABLE

**Syntax:** ALTER TABLE table\_name ADD UNIQUE (column1);

Example:



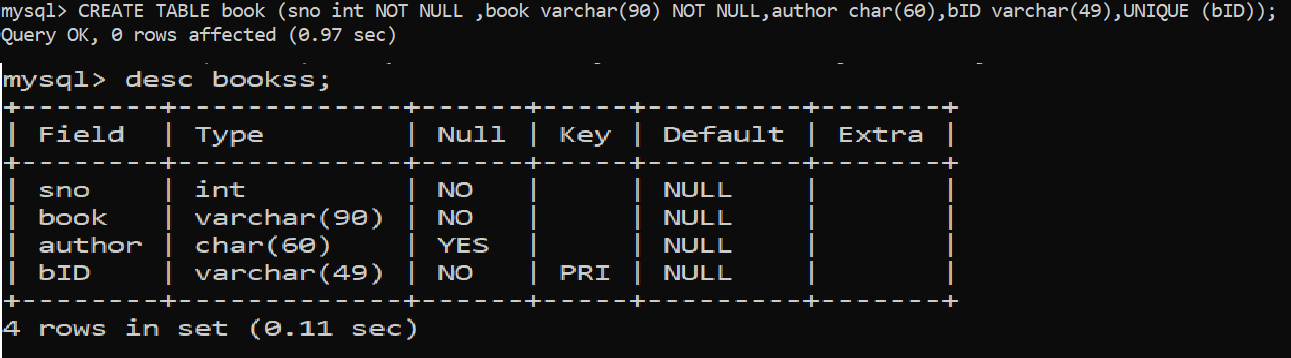
###### PRIMARY KEY CONSTRAINT

The PRIMARY KEY constraint uniquely identifies each record in table. Primary keys must contain UNIQUE values, and cannot contain NULL values.

**Syntax:** CREATE TABLE table\_name ( Column1 datatype, Column2 datatype , PRIMARY KEY (column1),lip

);

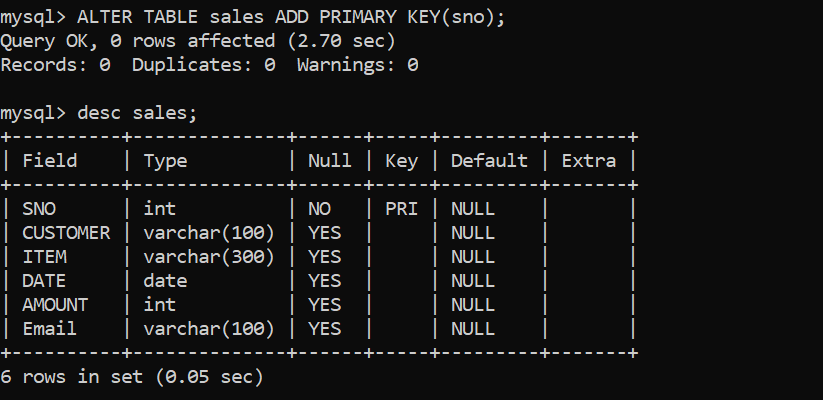
Example:



* + PRIMARY on ALTER TABLE

**Syntax:** ALTER TABLE table\_name ADD PRIMARY KEY (column1);

Example:



* + The PRIMARY KEY must contain UNIQUE values, and cannot contain NULL values.
  + There can be only one PRIMARY KEY constraint per table.

###### FOREIGN KEY CONSTRAINT

The FOREINGN KEY is a key used to link two tables together.

* A FOREINGN KEY is a field in one table that refers to PRIMARY KEY in other table.
* The table containing foreign key is called child table and other is parent table.

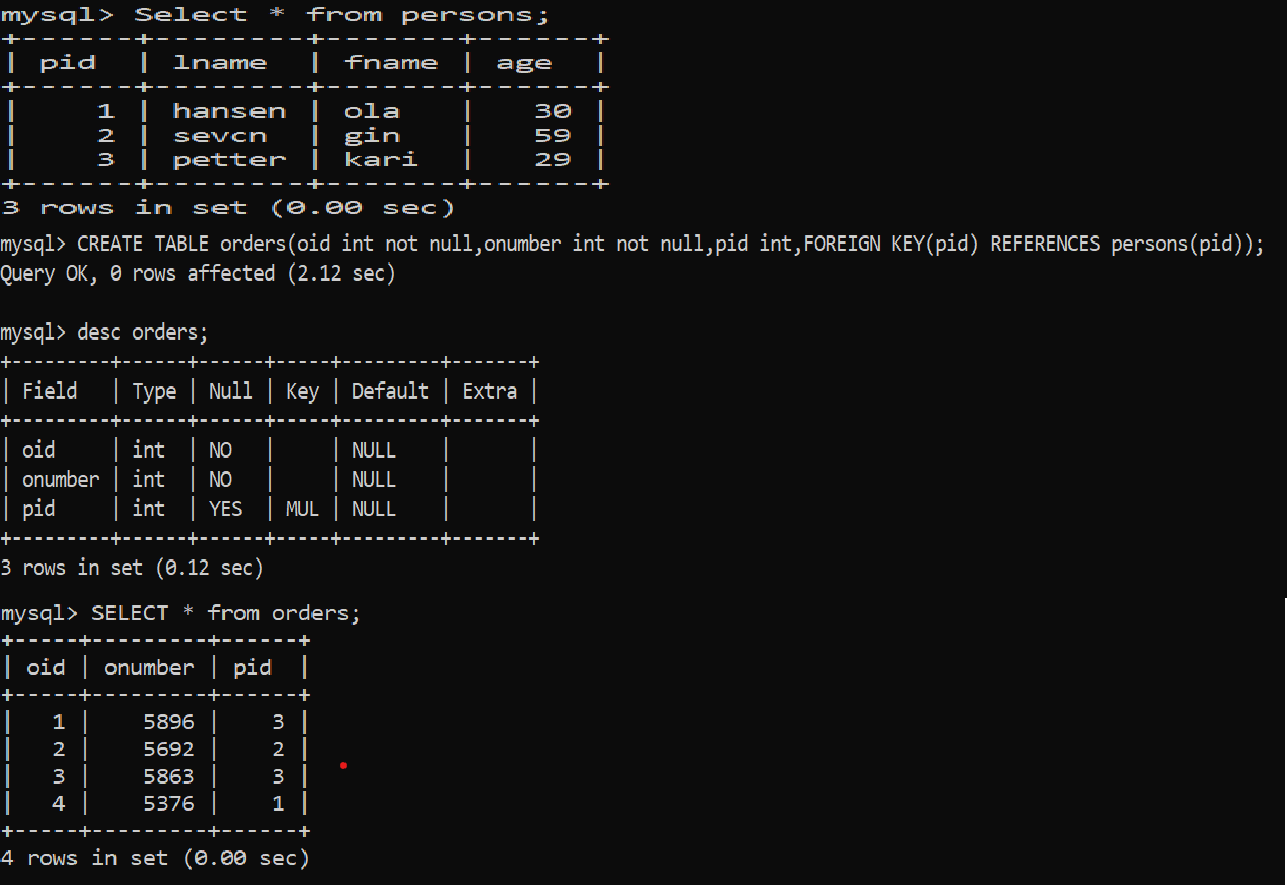
**Syntax:** CREATE TABLE table\_name (

Column1 datatype, Column2 datatype , PRIMARY KEY (column2),

FOREINGN KEY (column\_name) REFERENCE table\_name (column\_name)

);

Example:



###### CHECK CONSTRAINT

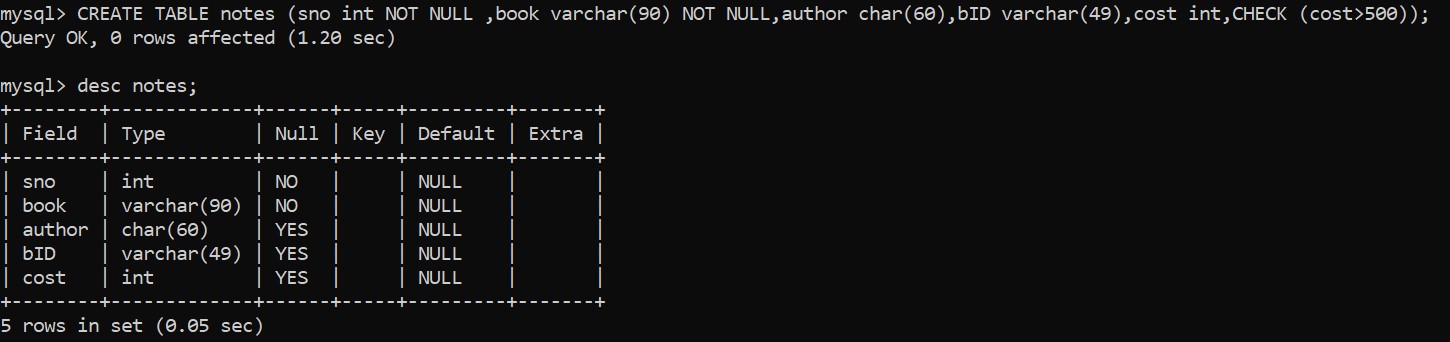
The CHECK constraint is used to limit the value range that can be placed in a column.

**Syntax:** CREATE TABLE table\_name ( Column1 datatype, Column2 datatype ,

CHECK (column>number)

);

Example:



###### DEFAULT CONSTRAINT

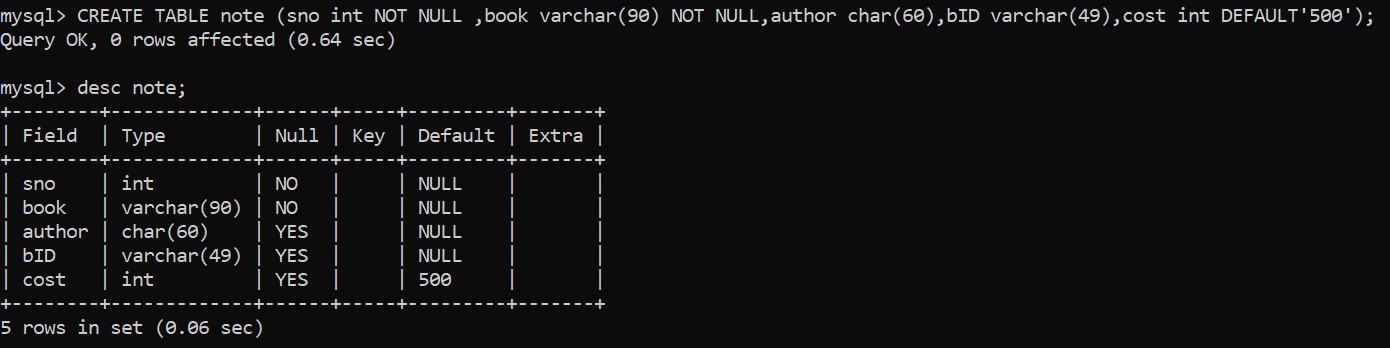
It is used provide a default value for a column. It is added to all new records if no other value is specified.

**Syntax:** CREATE TABLE table\_name ( Column1 datatype, Column2 datatype ,

Column3 datatype DEFAULT ‘any thing’

);

Example:



###### INDEX CONSTRAINT

It used to create and retrieve data from the database very

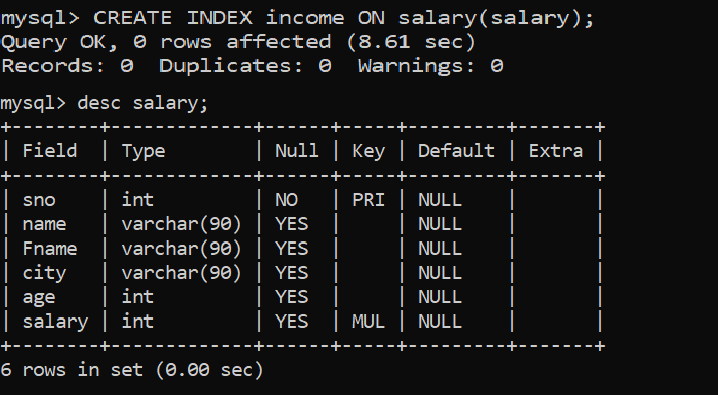
quickly.

1. Duplicate are allowed

**Syntax:** CREATE INDEX index\_name

ON table\_name (column1, column2,…);o

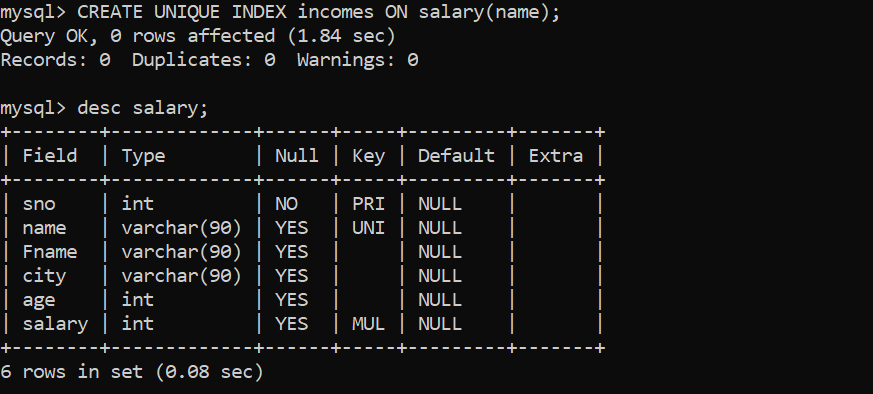
Example:

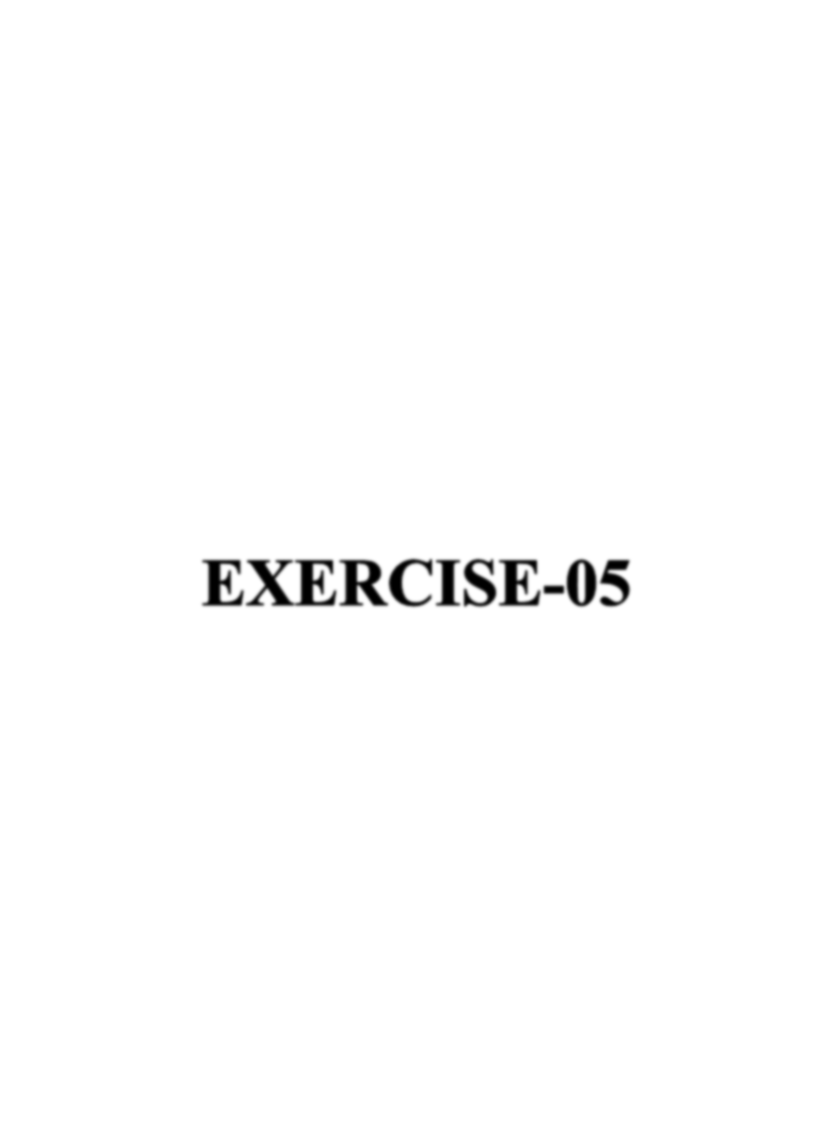
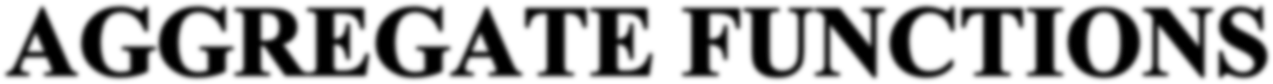


1. Duplicate are not allowed

**Syntax:** CREATE UNIQUE INDEX index\_name ON table\_name (column1, column2,…);

Example:





##### AGGREGATE FUNCTIONS:

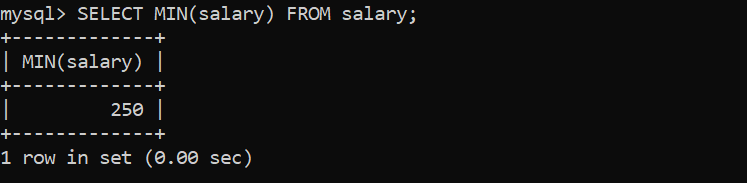
###### MAXIMUM, MINIMUM, COUNT, AVERAGE, SUM FUNCTIONS

1. **MIN()**

The MIN() function returns the smallest value of the selected column.

**Syntax:** SELECT MIN (column\_name)

FROM table\_name WHERE condition; Example:

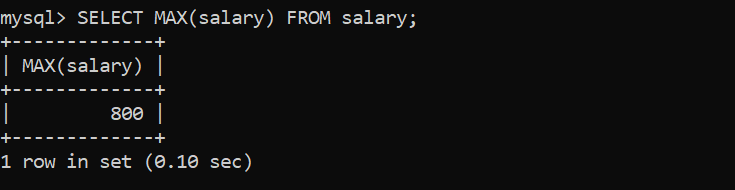
 n ;

###### MAX()

The MAX() function returns the largest value of the selected column.

**Syntax:** SELECT MAX (column\_name)

FROM table\_name WHERE condition; Example:

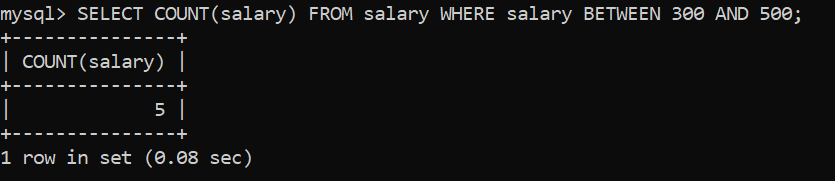


###### COUNT()

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

**Syntax:** SELECT COUNT (column\_name) FROM table\_name WHERE condition;

Example:

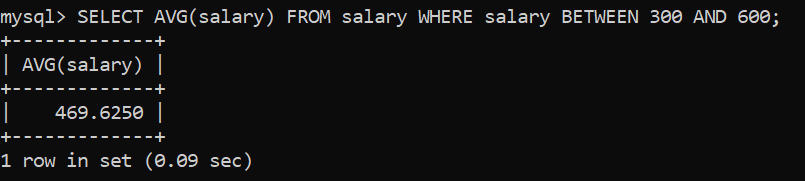


###### AVG()

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

**Syntax:** SELECT AVG (column\_name)

FROM table\_name WHERE condition; Example:

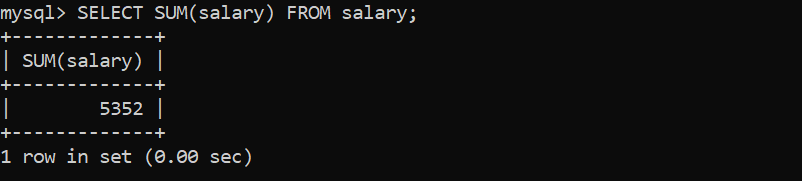


###### SUM()

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

**Syntax:** SELECT AVG (column\_name)

FROM table\_name WHERE condition; Example:



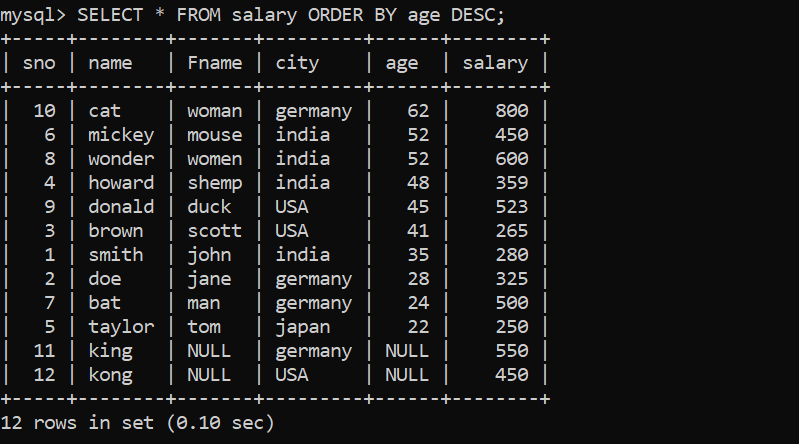
###### ORDER BY KEYWORD:

The ORDER BY keyword is used to sort the result set in ascending and descending order.

* + The ORDER BY keyword sorts the records in ascending order by default.
  + To sort the records in descending order, use the DESC keyword.

**Syntax:** SELECT column1,column2,…. FROM table\_name ORDER BY column1, column2, ….ASC/DESC;

Example:



###### GROUP BY STATEMENT:

The GROUP BY statement group rows that have the same values into summary rows

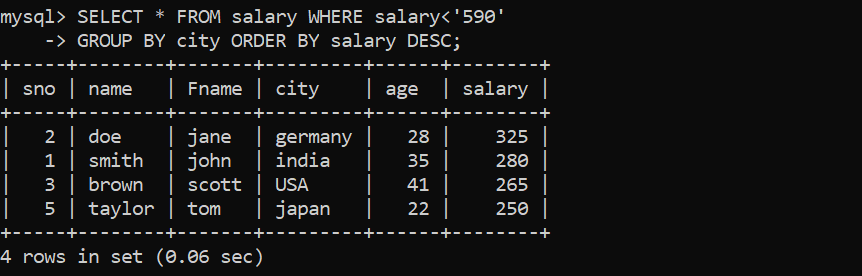
* + The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to the result-set by one or more columns.

**Syntax:** SELECT column\_name(s)

FROM table\_name WHERE condition GROUP BY column\_name(s)

ORDER BY column\_name(s);

Example:



###### HAVING CLAUSE

Having Clause was added to for group by command because the WHERE keyword cannot be used with aggregate functions.

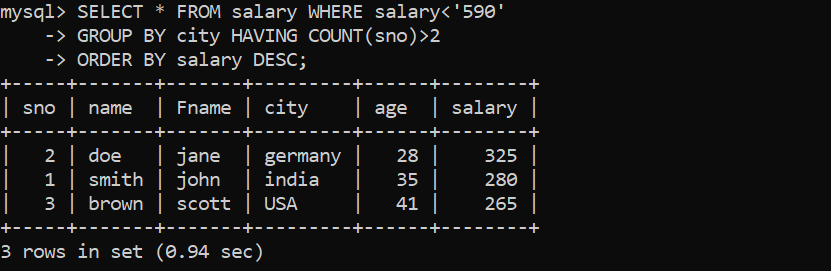
**Syntax:** SELECT column\_name(s)

FROM table\_name WHERE condition GROUP BY column\_name(s)

HAVING condition

ORDER BY column\_name(s);

Example:



# OPERATING SYSTEM

EXERCISE-01

### SECTION-01

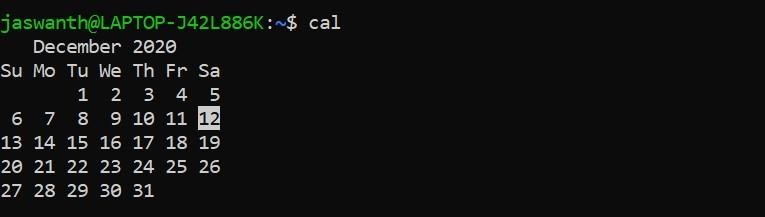
* BASIC LINUX COMMANDS

###### CAL COMMAND

It is used to display the calendar of current month.

**Syntax:** cal

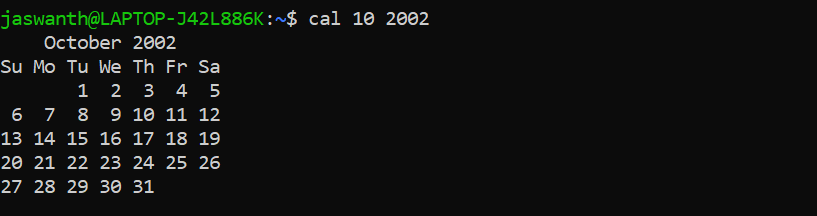
**Example:**



* + We can also use cal command for specified month and year.

**Syntax:** cal <month> <year>

**Example:**

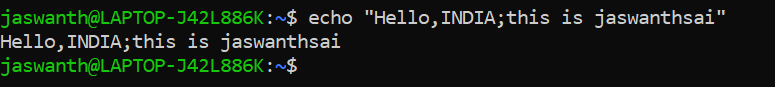


###### ECHO COMMAND

It is used to print whatever we written.

**Syntax:** echo “type whatever we want print on screen”

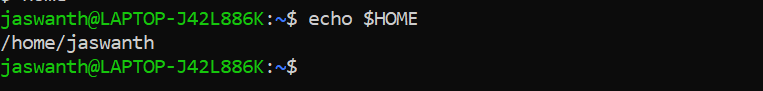
**Example:**



* + We can also used to display the values of variable.

**Syntax:** echo $variable

**Example:**



###### DATE COMMAND

The date command used display current date of the day.

**Syntax:** date

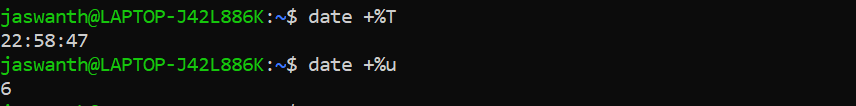
**Example:**



* + We can add certain elements to syntax to find specify time, no of the day in week

**Syntax:** date +%T (for time) or date +%u (for weekday no)

**Example:**



###### TTY COMMAND

It displays current terminal.

**Syntax:** tty

**Example:**



###### WHOAMI COMMAND

This command reveals the user who is currently logged in.

**Syntax:** whoami

**Example:**

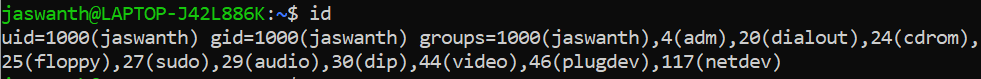


###### ID COMMAND

This command prints the userid and groupid of the current user.

**Syntax:** id

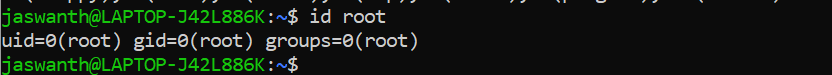
**Example:**



* + we can use id root command to display the id of root user.

**Syntax:** id root

**Example:**



###### CLEAR COMMAND

This command used to clear the page of command line.

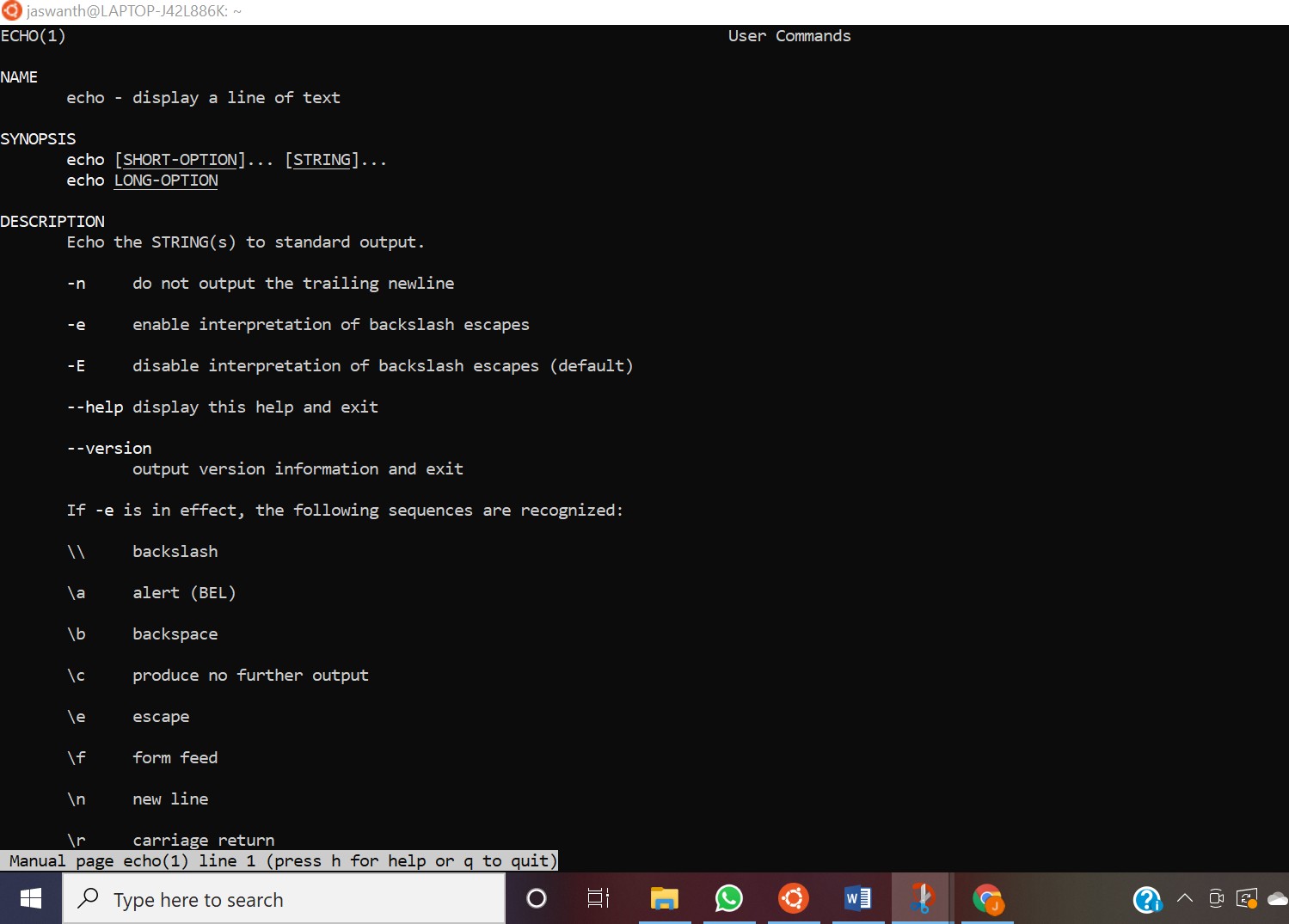
**Syntax:** clear

###### MAN COMMAND

Man command is used to display the use of another commands.

**Syntax:** man <any command>

**Example:**

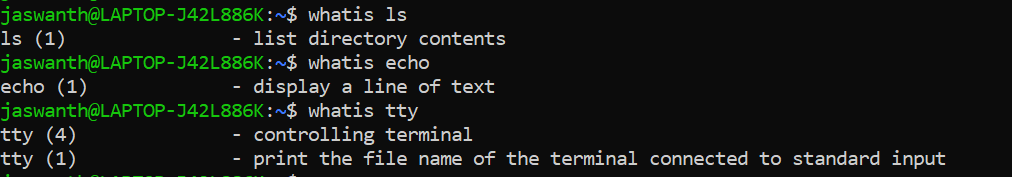


###### WHATIS

What is command used to describe the manual page use of command.

**Syntax:** whatis <any command>

**Example:**



## SECTION-02

###### WORKING WITH DIRECTORIES

1. **PWD COMMAND**

The pwd command is used to display the location of the current working directory.

**Syntax:** pwd

**Example:**



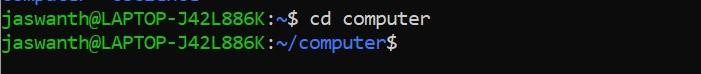
###### CD COMMANDS

* + cd

This cd command is used to change directory.

**Syntax:** cd

**Example:**



* + cd ~

This command used to get back to the home directory.

**Syntax:** cd ~

**Example:**



* + cd ..

This command used to get back to parent directory.

**Syntax:** cd ..

**Example:**



###### LS COMMANDS

* + ls

This command used to list the contents of the directory.

**Syntax:** ls

**Example:**

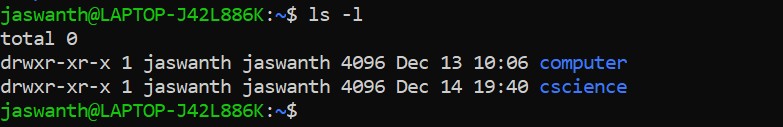


* + ls -l

This command used to summarize all the most important information about the file on one line and displays it.

**Syntax:** ls -l

**Example:**



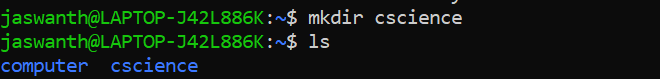
###### MKDIR COMMANDS

* + mkdir

This command used to create our own directory.

**Syntax:** mkdir <directory name>

**Example:**

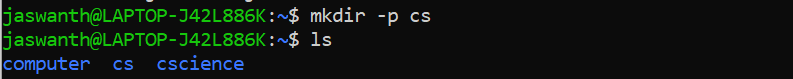


* + mkdir -p

By this command you can create sub-directories of a directory. It will create parent directory first, if it doesn't exist it fails.

**Syntax:** mkdir -p <directory name>

**Example:**

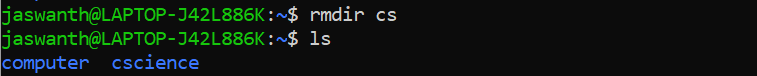


* + rmdir

This command used to remove a directory when it is an empty directory.

**Syntax:** rmdir <directory name>

**Example:**

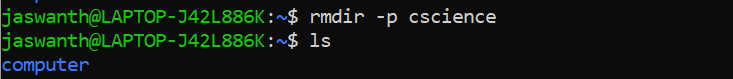


* + rmdir -p

This command used to recursively remove a directory .

**Syntax:** rmdir -p<directory name>

**Example:**



## SECTION-03

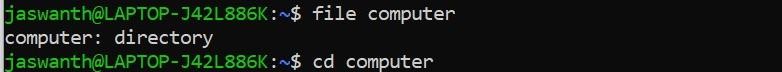
###### WORKING WITH FILES

1. **FILE COMMAND**

The file command displays the type of file which we entered.

**Syntax:** file <file name>

**Example:**

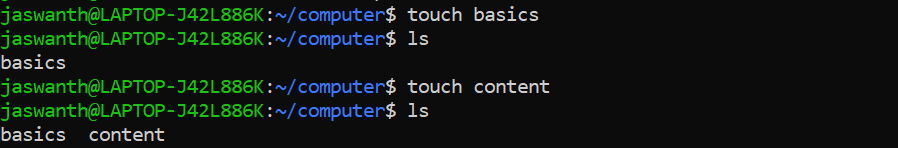


###### TOUCH COMMAND

This command creates an empty file in the directory.

**Syntax:** touch <file name>

**Example:**

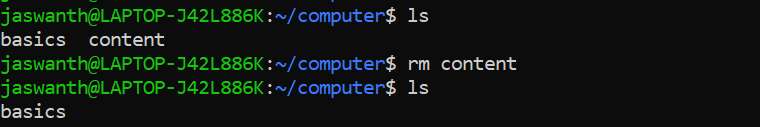


###### RM COMMAND

This command is used to remove the file forever.

**Syntax:** rm <file name>

**Example:**



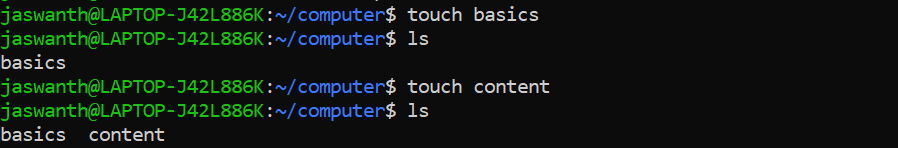
###### COPY COMMANDS

* + cp

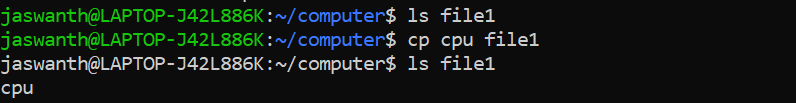
This command used copy an existing file to new file.

**Syntax:** cp < exiting file name> < new file name>

**Example:**



* we can also copy file from one directory to other. **Syntax:** cp < exiting file name> < other directory name> **Example:**

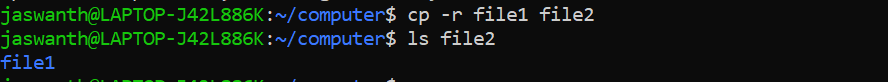


* + cp –r

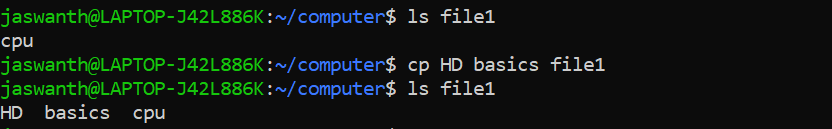
This command is used to copy all files from one directory to another Another directory recursively

**Syntax:** cp –r < existing directory name> < new directory name>

**Example:**



* we can also copy multiple files from one directory to other. **Syntax:** cp < existing file names> < new directory name> **Example:**



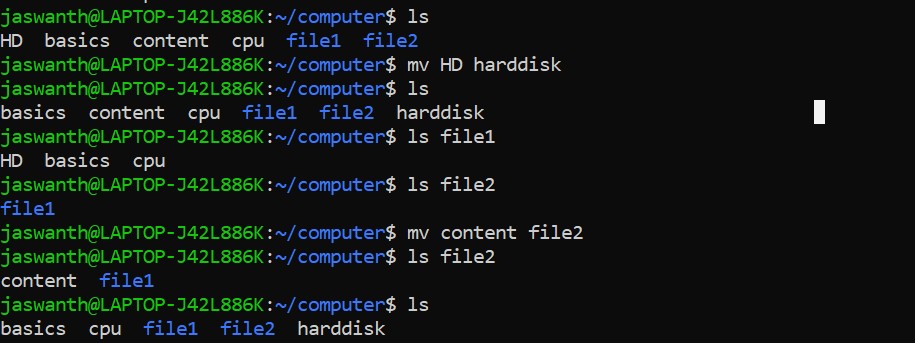
###### MOVE COMMANDS

* + **Rename file with move**

This is the command used to rename or move the file to another directory

**Syntax:** mv < exiting file name> < new file name>

**Example:**



* + **Rename directory with move**

This command is used to rename the directory.

**Syntax:** mv < existing directory name> < new directory name>

**Example:**



## SECTION-04

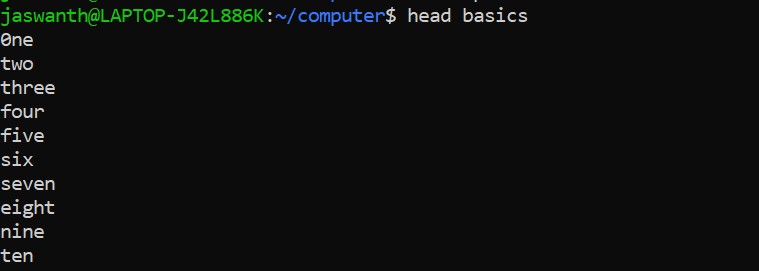
###### WORKING WITH FILE CONTENTS

1. **HEAD COMMAND**

This command used to display the first ten lines of a file.

**Syntax:** head <file name>

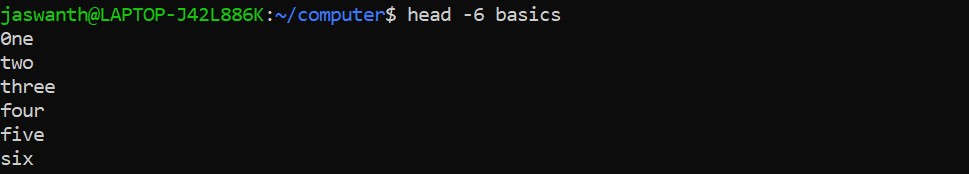
**Example:**



* + By this command we can display first n lines.

**Syntax:** head –n <file name>

**Example:**

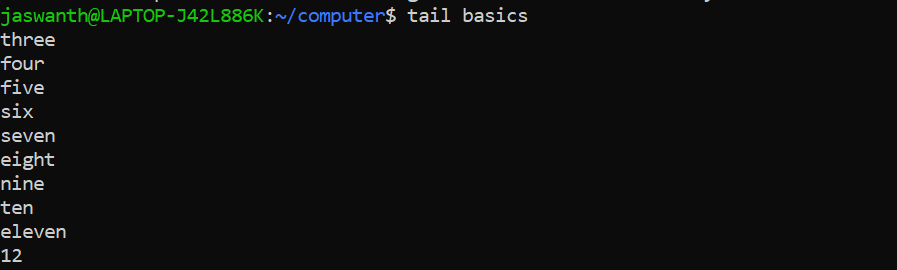


###### TAIL COMMAND

This command used to display the last ten lines of a file.

**Syntax:** tail <file name>

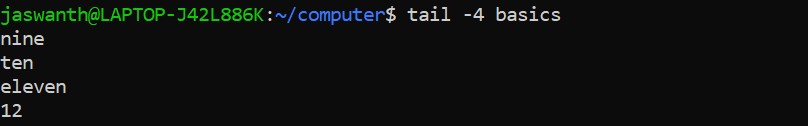
**Example:**



* + By this command we can display last n lines.

**Syntax:** tail –n <file name>

**Example:**



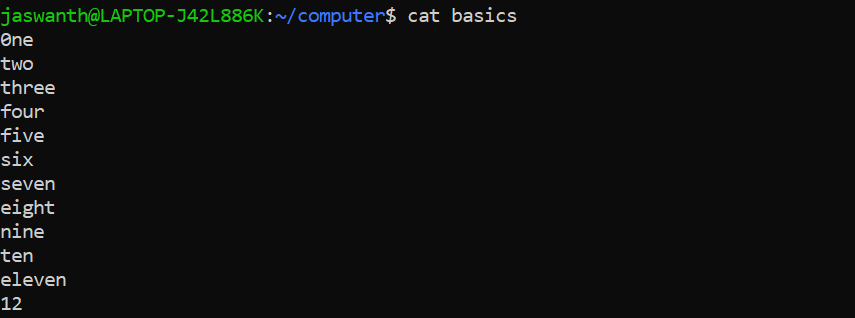
###### CAT COMMANDS

* cat

This command is used to display all the contents of a file

**Syntax:** cat <file name>

**Example:**

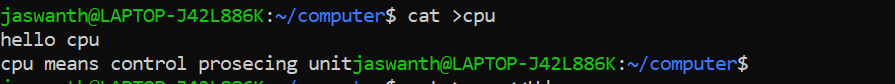


* create file using cat command

This command is used to insert content in the file.

**Syntax:** cat > <file name>

**Example:**

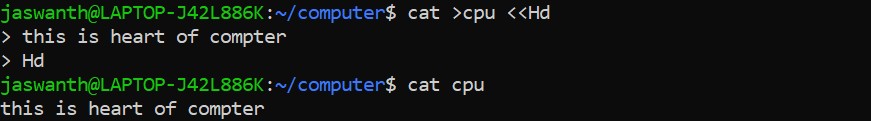


* custom end marker

It is an end marker for cat with << which ends the content after that text.

**Syntax:** cat > <file name> <<eof

**Example:**

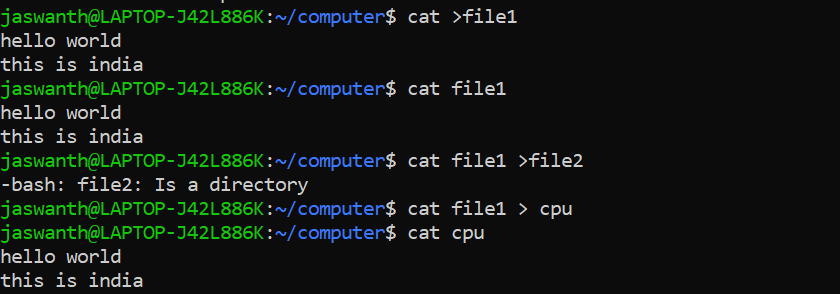


* copy files

We can use cat command to copy one file content to other.

**Syntax:** cat <file name> > <another file name>

**Example:**

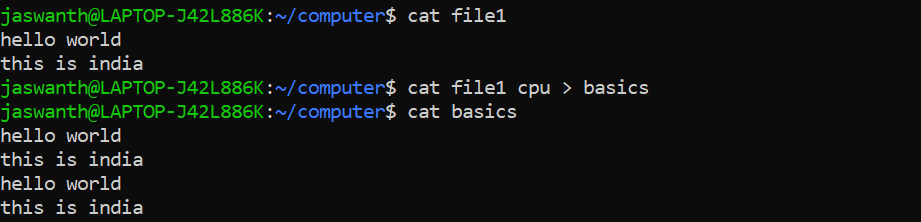


* + concatenate

The cat is short form of concatenate. By using cat command, we can combined the content in all files to one file.

**Syntax:** cat <file name> <another file name> > <final file name>

**Example:**

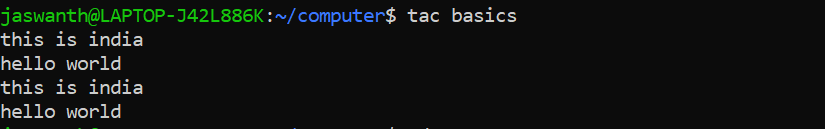


* + tac

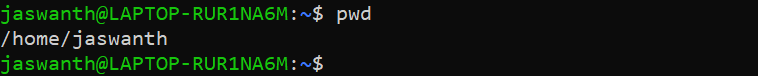
This command is used to display the content bottom to top.

**Syntax:** tac <file name>

**Example:**



## LAB EXERCISE-1

1. Display the path of your current directory.
2. Make a new directory named main.



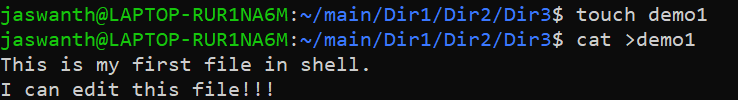
1. Now change to the directory main.
2. Make the directories in the following hierarchy using a single command.

Dir1--Dir2--Dir3

1. Print the path of the current directory.

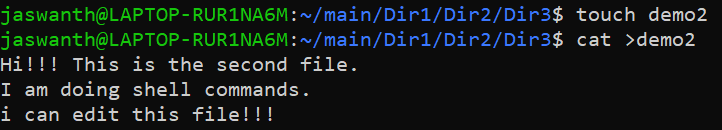


1. Go to Dir3 using a single command.
2. Create a new file **demo1**, type and save the following contents, This is my first file in shell.

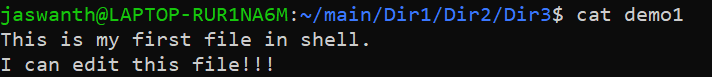
I can edit this file!!!

1. Create a new file **demo2**, type and save the following contents, Hi!!! This is the second file.

I am doing shell commands.

I can edit this file!!!

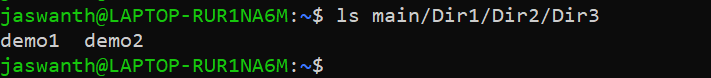
1. Display the contents of file **demo1** in terminal.

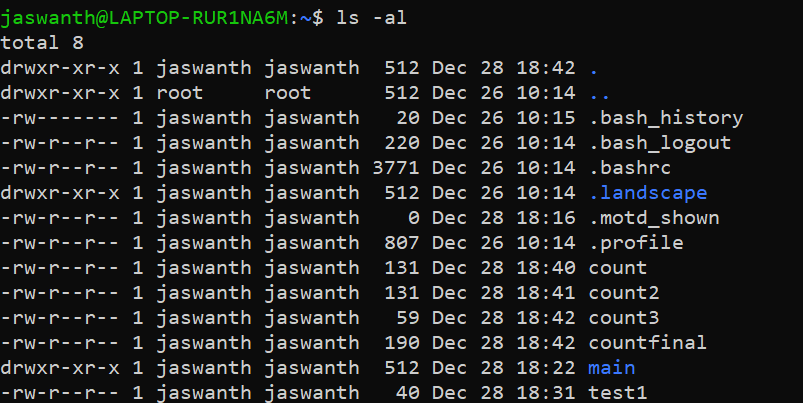


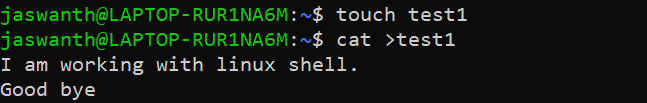
1. List the files and folders present in Dir3.
2. Go to Dir 2.



1. Go to your home directory.
2. Stay where you are, and list the contents of Dir3.



1. List all the files (including hidden files) in your home directory.
2. Create a new file **test1**, type and save the contents into your file. I am working with linux shell.

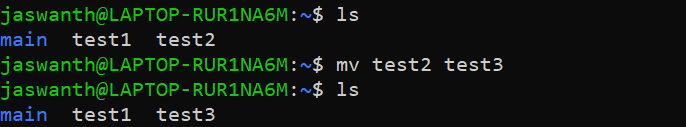
Good bye

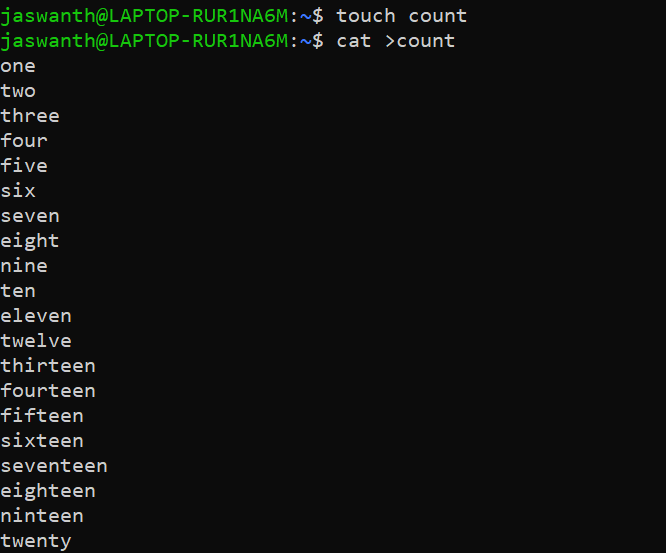
1. Copy the contents of **test1** to **test2** in the same directory.



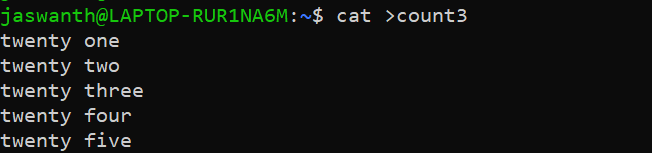
1. Rename **test2** as **test3**.
2. Determine the file type of **test3**.



1. Move the file **test3** to the directory Dir3.
2. Create a file **count**, with content one to twenty in words with one line having only one number using a single command.



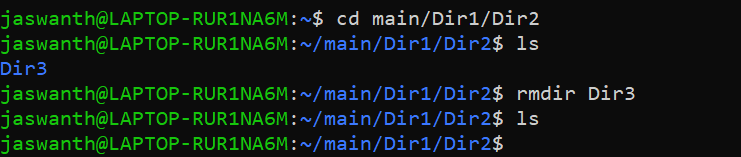
1. Copy the file **count** to **count2** using cat command.
2. Create another file **count3** with numbers twenty one to twenty five (in five lines).

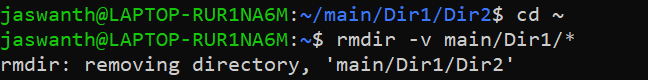


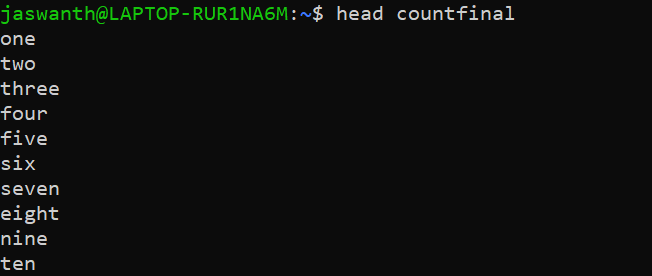
1. Concatenate the contents of files **count2** and **count3** and write it into the file **countfinal**.

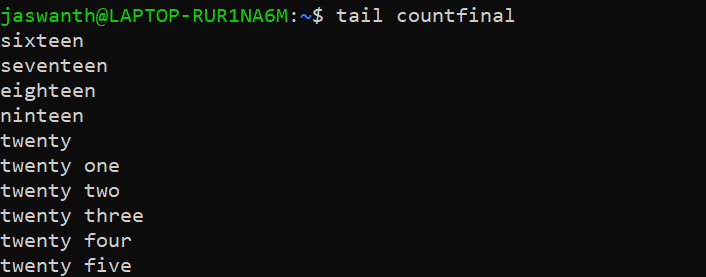


1. Remove the files **demo1** and **demo2** in directory Dir3.
2. Go to Dir2 and remove the subdirectory Dir3.

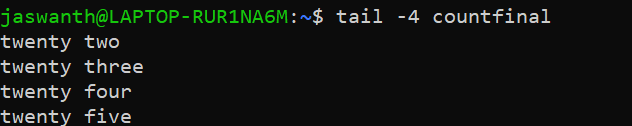


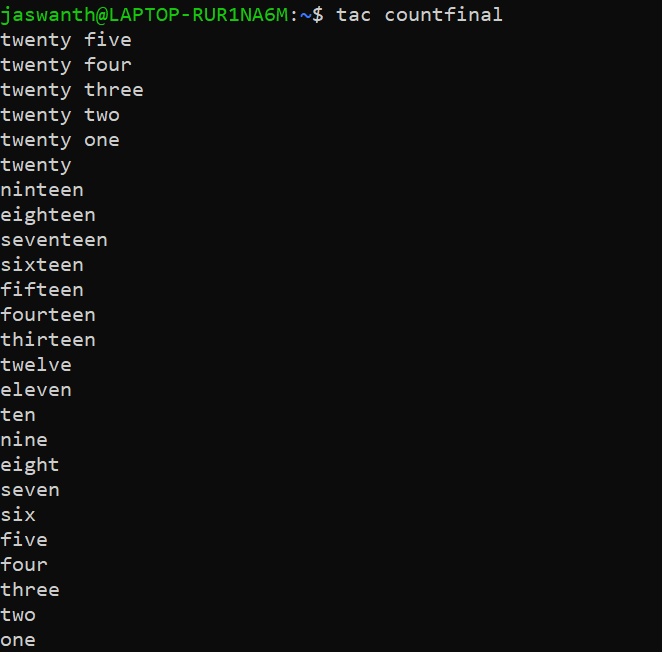
1. Come back to your home folder and remove Dir2.
2. Display first 10 lines of the file **countfinal** in terminal.



1. Display last 10 lines of the file **countfinal** in terminal.
2. Display first 5 lines of the file **countfinal** in terminal.



1. Display last 4 lines of the file **countfinal** in terminal.
2. Display the contents of the file **countfinal** in the inverted form.(last line first and first line last)



# OPERATING SYSTEM

EXERCISE-02

### SECTION-05

* FILTERS

1. >stdout

Stdout can be redirected with a greater than sign.>stdout

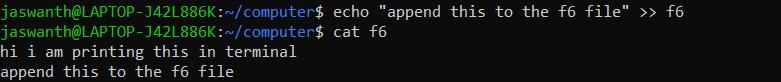
**Example:**



1. >> append

append is used to add the content to file at last.

**Example:**



###### PIPES

Pipe is a command of symbol ‘|’ which is used to combined more than one command

**Syntax:** command1 |command2 |command3 …

**Example:**

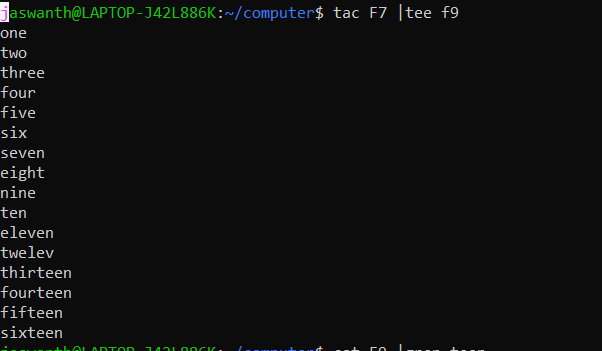


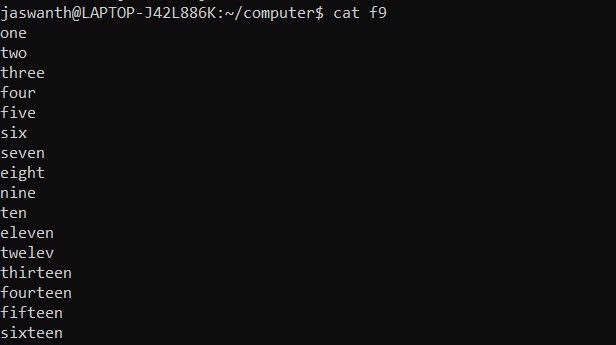
###### TEE

This tee command used filter puts stdin on stdout and also into a file.

**Syntax:** cat <file Name> | tee <new File> | cat or tac | …

**Example:**



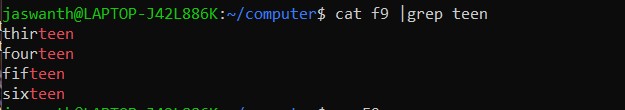


###### GREP COMMANDS

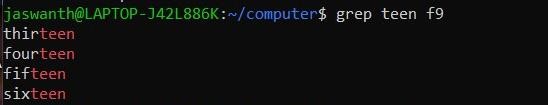
The greap command is used to filter the line of text containing or not containing a certain string.

**Syntax:** command | grep <search Word>

**Example:**



* + We can write grep with out cat command **Syntax:** grep <search Word> <File name> **Example:**



* + grep -i

this command is used to filter in case of insensitive way.

**Syntax:** grep -i <search Word> <File name>

**Example:**



* + grep -v

this command is used to show output which are not matching to that string.

**Syntax:** grep -v <search Word> <File name>

**Example:**



* + grep -vi

this command is used to filter the output which are not matching to that string in case of insensitive way.

**Syntax:** grep -vi <search Word> <File name>

**Example:**

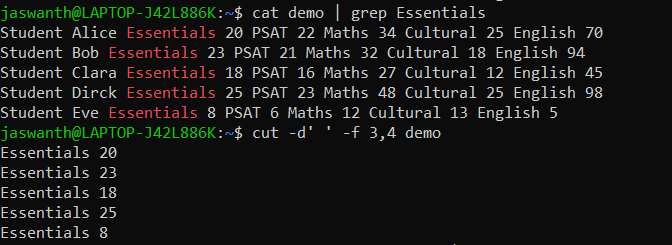


###### CUT

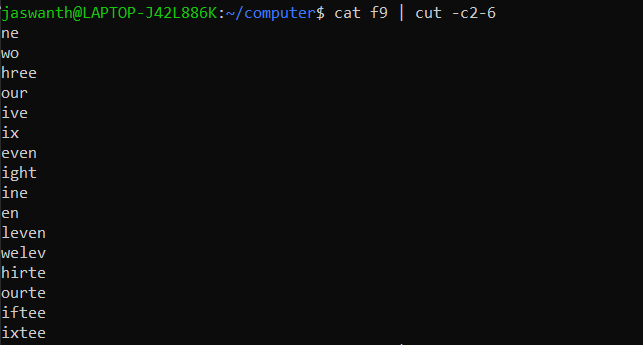
The cut filter can be used to select a particular column from the file depending on a delimiter or count of bytes.

**Syntax**: cut -d(delimiter) -f (column Number) <file Name>

**Example:**

**`**

* + we can also display puticular characters by using this command. **Syntax**: cat <file name> | cut -c <starting char>-<ending char> **Example:**

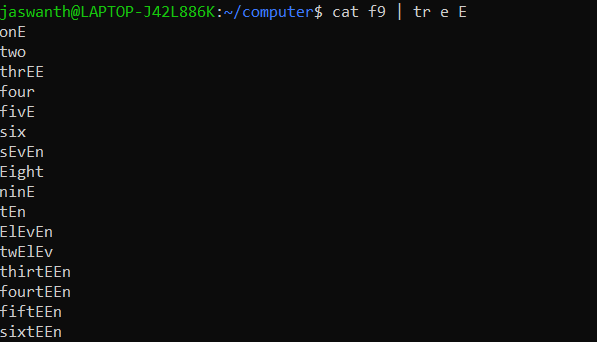


###### TR COMMANDS

By this command we can translate the data.

**Syntax**: command | tr <'old'> <'new'>

**Example:**



* + We can use this command to change the all letters to upper case.

**Syntax**: command | tr a-z A-Z

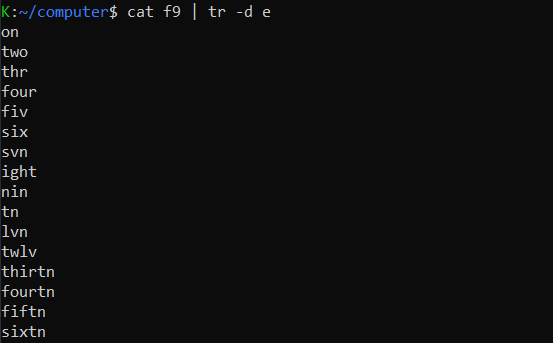
**Example:**



* + We can use this command to delete a particular character.

**Syntax**: command | tr -d <character>

**Example:**



* + We can use this command to combine line with space.

**Syntax**: command | tr ‘\n’ ‘ ‘

**Example:**



###### WC

This command is used to counting words, line, characters.

**Syntax**: wc -l/w/c <file name>

**Example:**

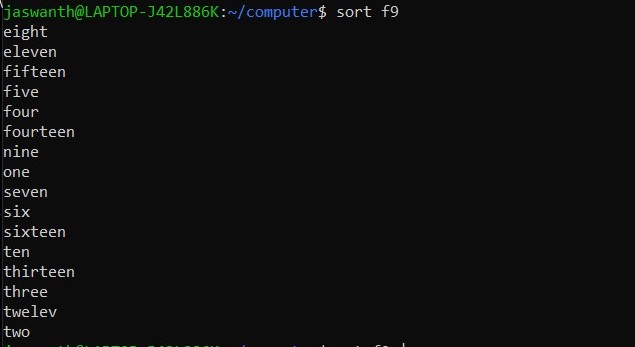


###### SORT

This command is used to sort alphabetical in a file.

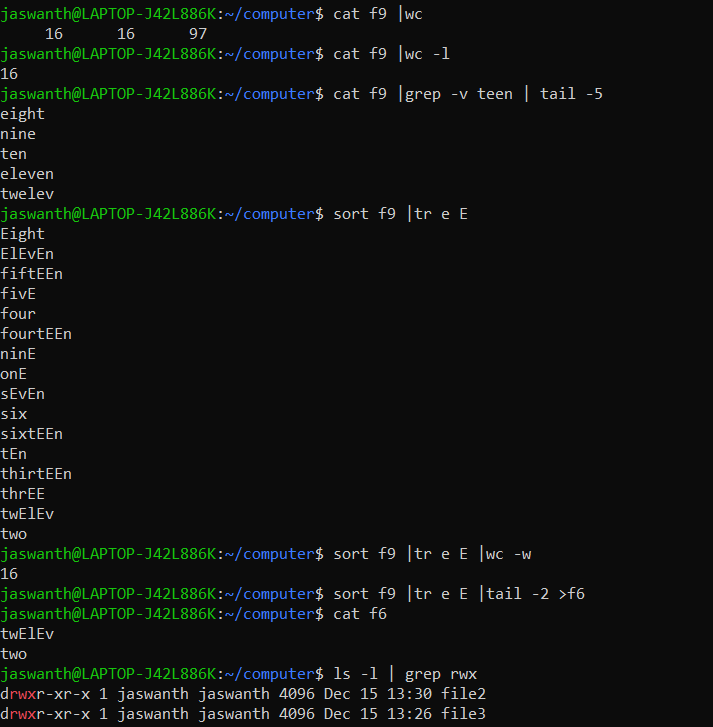
**Syntax**: sort <file name>

**Example:**



* Pipe Examples

### SECTION-06



#### LAB EXERCISE-02

1. Create a file demo with the following contents

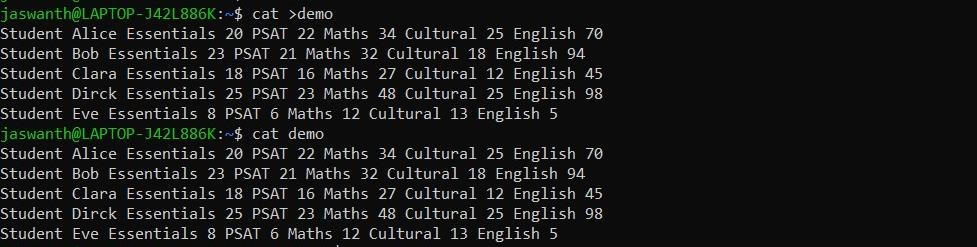
Student Alice Essentials 20 PSAT 22 Maths 34 Cultural 25 English 70

Student Bob Essentials 23 PSAT 21 Maths 32 Cultural 18 English 94

Student Clara Essentials 18 PSAT 16 Maths 27 Cultural 12 English 45

Student Dirck Essentials 25 PSAT 23 Maths 48 Cultural 25 English 98

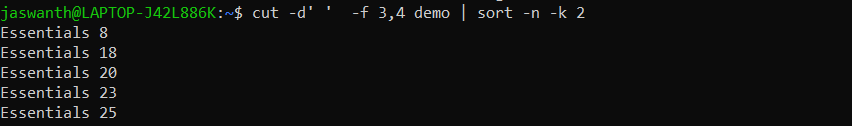
Student Eve Essentials 8 PSAT 6 Maths 12 Cultural 13 English 5



1. Find the marks obtained by Clara in all the subjects



1. Print the marks for essentials in the increasing order



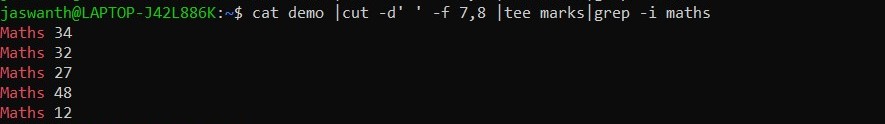
1. Find the maximum marks scored in PSAT



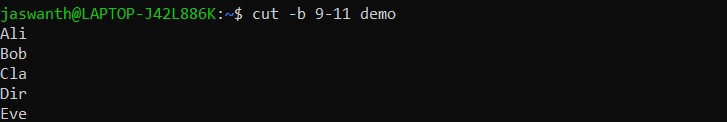
1. Find the minimum marks obtained in Cultural



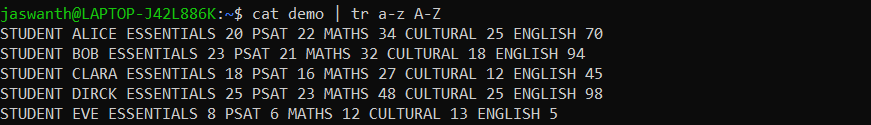
1. Save the marks obtained by all the students in maths into a file and display it in the terminal using a single command



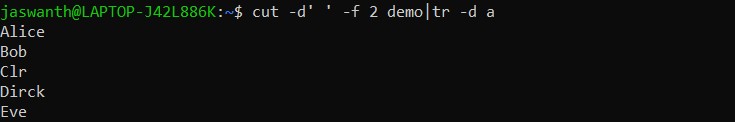
1. Print the first 3 letters of all student names.



1. Print the contents of file demo in terminal with all alphabets in capital letters.



1. Print all student names after deleting the letter ‘a’



1. Count the number of lines, words and characters in demo file after removing the letter ‘S’



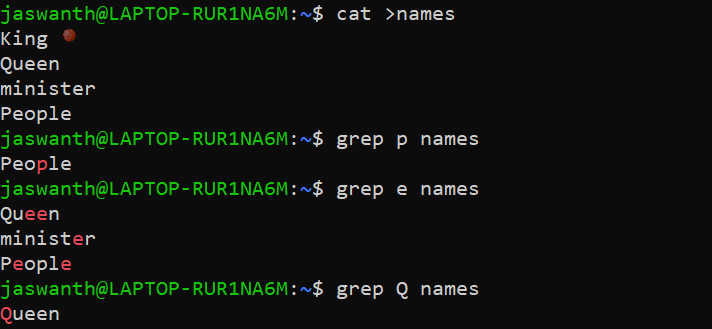
# OPERATING SYSTEM EXERCISE-03

SECTION-08

* Regular Expressions
  1. **GREP**

When we use grep for single character the line which containing that character only returned.

**Syntax:** grep <character> <file name>

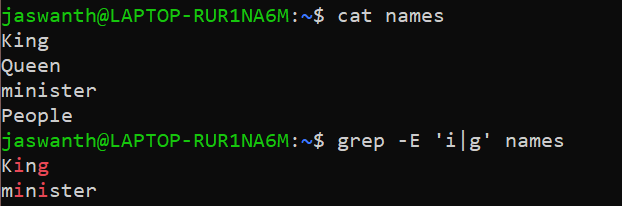
**Example:**

* 1. **One or The Other**

Here pipe (|) symbol is used as OR to signify one or the other. All the three versions are shown

**Syntax:** grep <option> <pattern | pattern> <fileName>

**Example:**

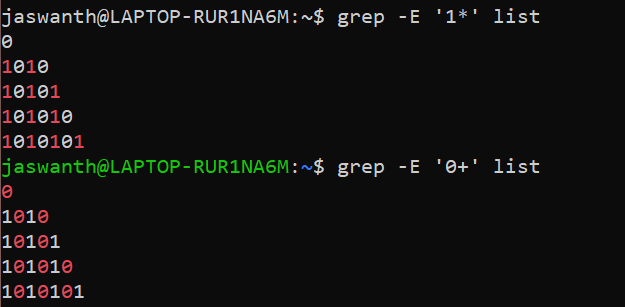
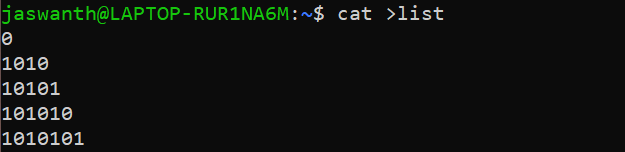


* 1. **One or More**

The \* signifies zero or more times occurence of a pattern and + signifies one or more times occurence.

**Syntax:** grep <option> ‘<pattern>(+or\*)’ <fileName>

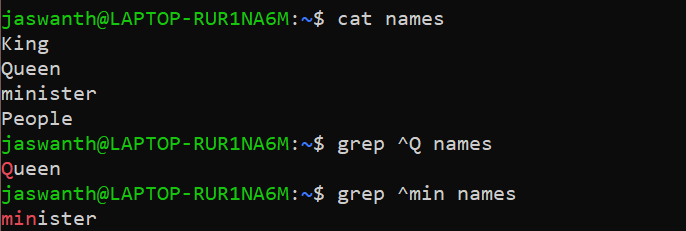
**Example:**



* 1. **Match the starting of string**
     + To match the start or beginning of a file we use caret sign (^).

**Syntax:** grep ^<pattern> <fileName>

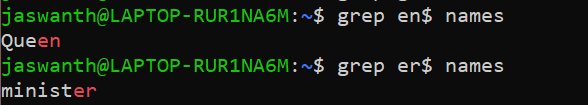
**Example:**



* 1. **Match the ending of string**
     + To match the end of a string we use $ sign.

**Syntax:** grep <pattern>$ <fileName>

**Example:**

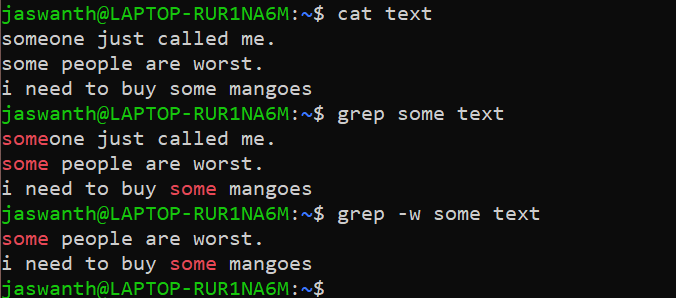


* 1. **Separating words**

Regular expressions use a -w option to reference a word separator.

**Syntax:** grep <pattern> <fileName>

**Example:**



Lab Exercise

1. Write shell scripts for the following actions in the given order:
   1. Create a file **Studdetails** with the following contents

Student Alice Essentials 20 PSAT 22 Maths 34 Cultural 25 Physics 80 English 70

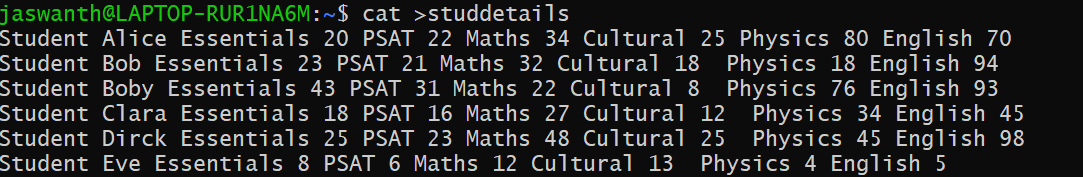
Student Bob Essentials 23 PSAT 21 Maths 32 Cultural 18 Physics 18 English 94

Student Boby Essentials 43 PSAT 31 Maths 22 Cultural 8 Physics 76 English 93

Student Clara Essentials 18 PSAT 16 Maths 27 Cultural 12 Physics 34 English 45

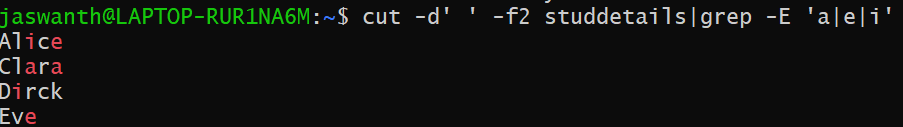
Student Dirck Essentials 25 PSAT 23 Maths 48 Cultural 25 Physics 45 English 98+

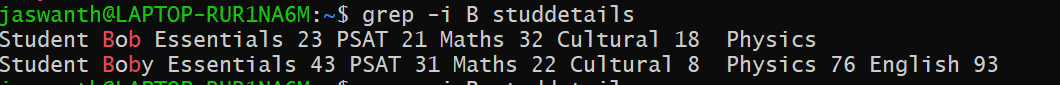
Student Eve Essentials 8 PSAT 6 Maths 12 Cultural 13 Physics 4 English 5



* 1. Filter the details of the student Bob



* 1. Find the number of students with their names containing the letter a, e or i
  2. Find the marks of students whose names starts with ‘b’ (case insensitive)



* 1. Find the names of students whose names starts with ‘b’ and ends with ‘y’ (case insensitive)



1. Write a shell script to perform the following actions in the given order:
   1. Create a file **numericdata** with the following contents Kollam 56754 6754 7654

Vallikkavu 54328 7548 45675

Trivandrum 16423 6654 6754

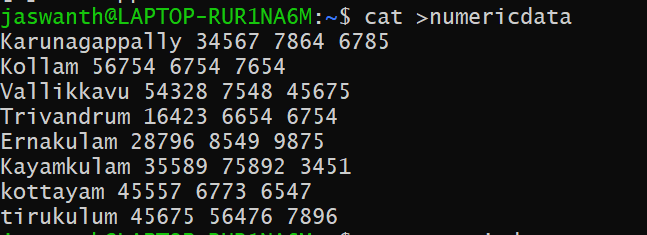
Ernakulam 28796 8549 9875

Kayamkulam 35589 75892 3451

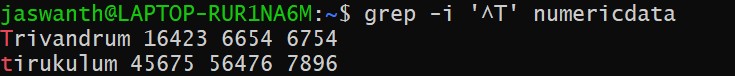
kottayam 45557 6773 6547

tirukulum 45675 56476 7896

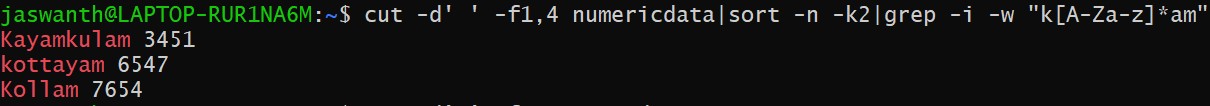
(Hint : First field is referred as Place second as code1 third as code2 and fourth as code3)

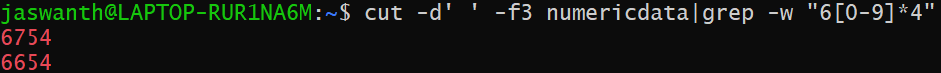


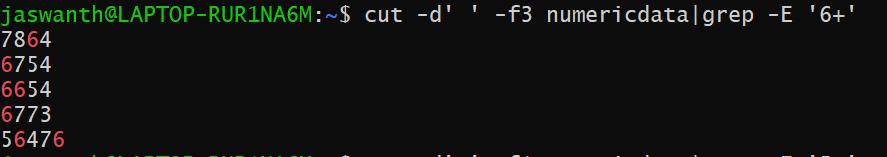
* 1. Display the details of Places that starts with ‘T’(case sensitive)

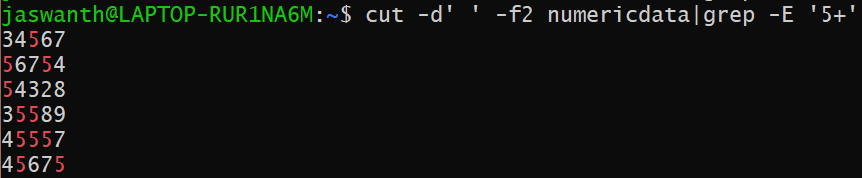


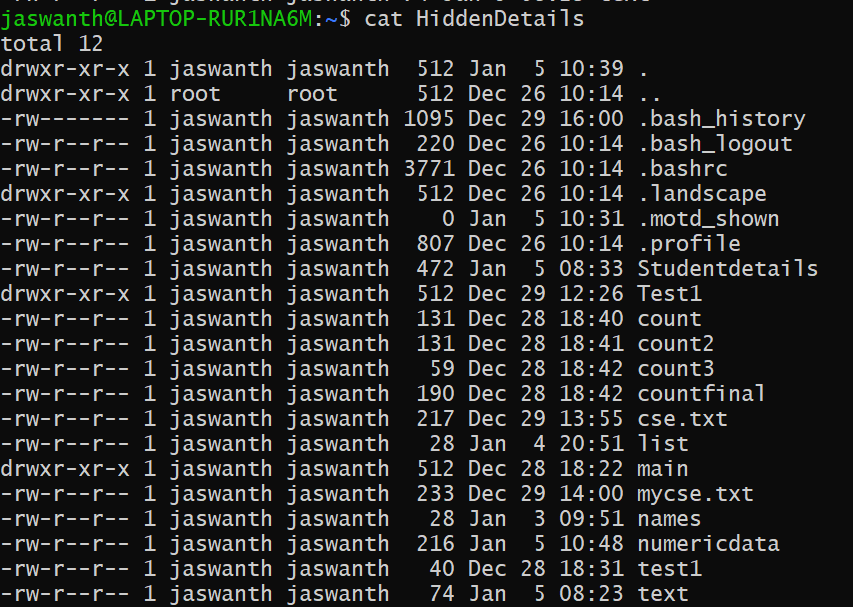
* 1. Display code3 in sorted order(ascending) of the places that start with ‘K’(case insensitive) and ends with am



* 1. Filter code2 that starts with 6 and ends with 4
  2. Filter code2 having one or more occurrence of the digit 6



* 1. Filter all code1 having one or more occurrence of the digit 5.

1. Write a shell script to perform the following actions in the given order:
   1. Write the contents of ls -la to a file named **HiddenDetails**
   2. Write the contents of file **HiddenDetails** to the file **HiddenFiles** after replacing multiple occurrences of space with a single space.



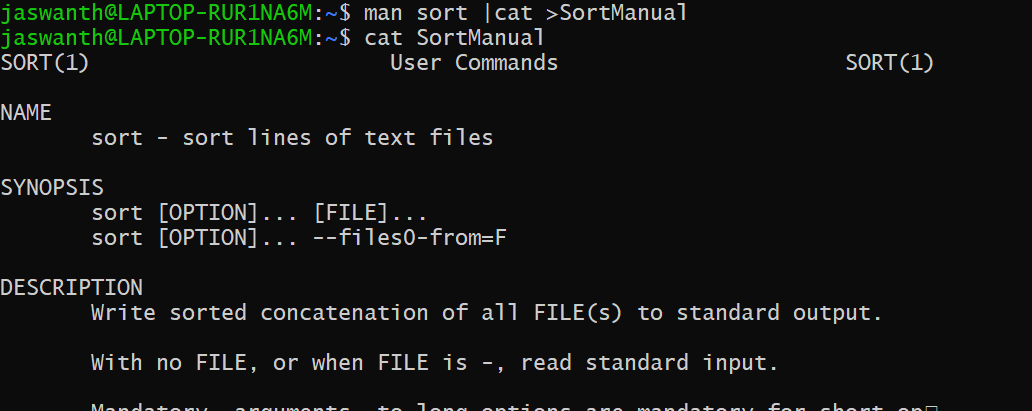
* 1. Using the file **HiddenFiles,** find the total number of files and folders that starts with ‘p’(case insensitive)

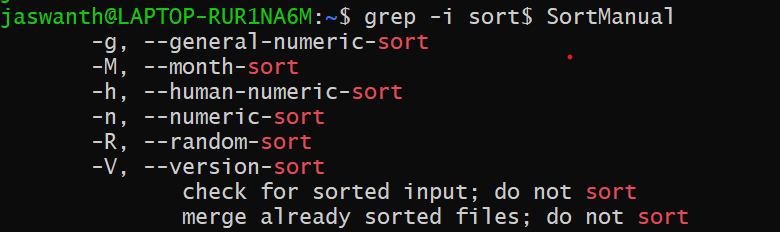


* 1. Using the file **HiddenFiles,** find the names of files and folders that starts with ‘.’ and ends with ‘nda’
  2. Using the file **HiddenFiles,** list all the directories

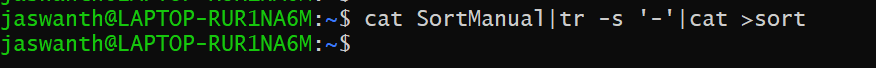
(The first field in ls -la denotes the permissions associated with the file. If the first letter in permission is ‘d’ then it is a directory)



1. Write shell program to perform the following actions:
   1. Write the contents of the command man sort into a file **SortManual**.
   2. Filter all lines form the file **SortManual** that starts with ‘compare’.
   3. Filter all lines form the file **SortManual** that ends with ‘sort’.

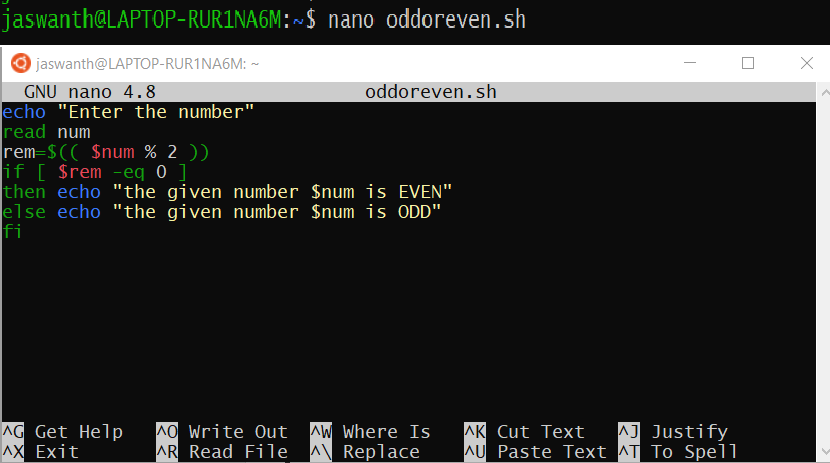


* 1. Display the number of lines in **SortManual** that has the word ‘numeric’.
  2. Write the contents of file **SortManual** to the file **sort** after replacing multiple occurrences of ‘-’ with a single ‘-’

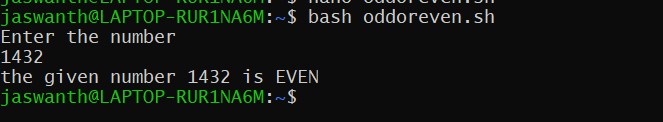


SHELL PROGRAMS:

1. ODD OR EVEN

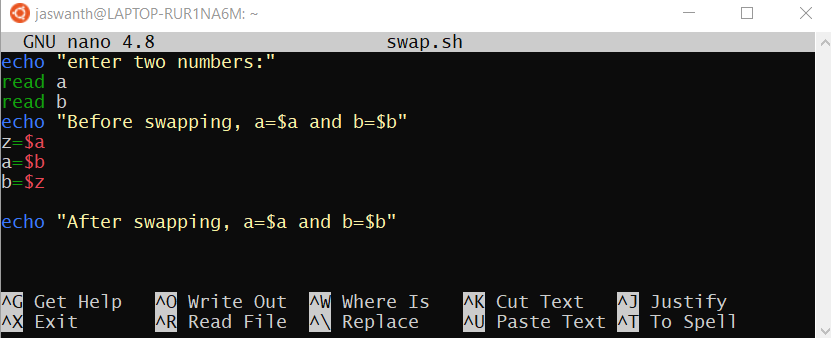
Code:

Result:

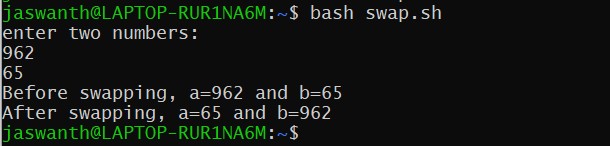


1. SWAP

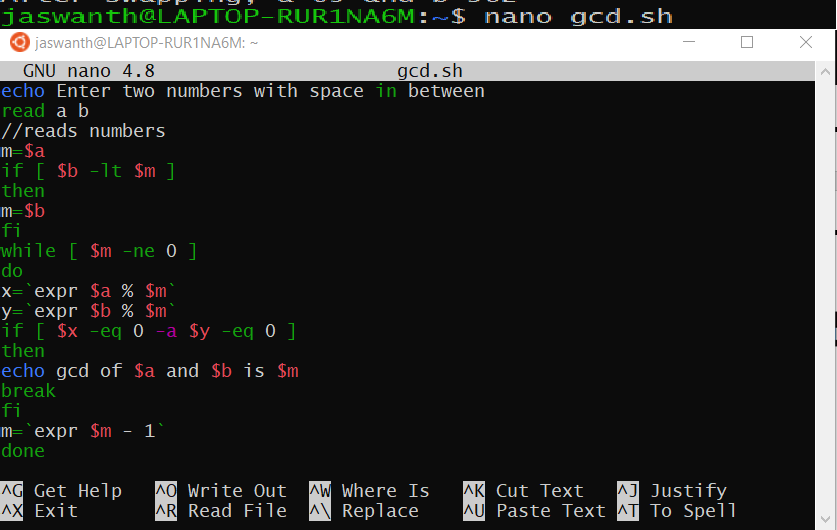
Code:



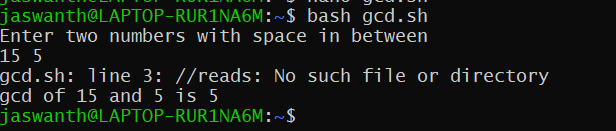
Result:



1. GCD

Code:

Result:

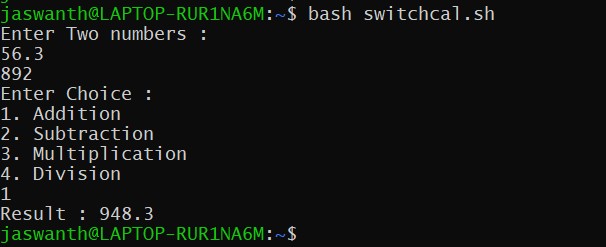


1. SIMPLE CALUCLATOR

Code:



Result:



LINUX NETWORK COMMANDS

Linux ifconfig stands for interface configurator. It is one of the most basic commands used in network inspection.

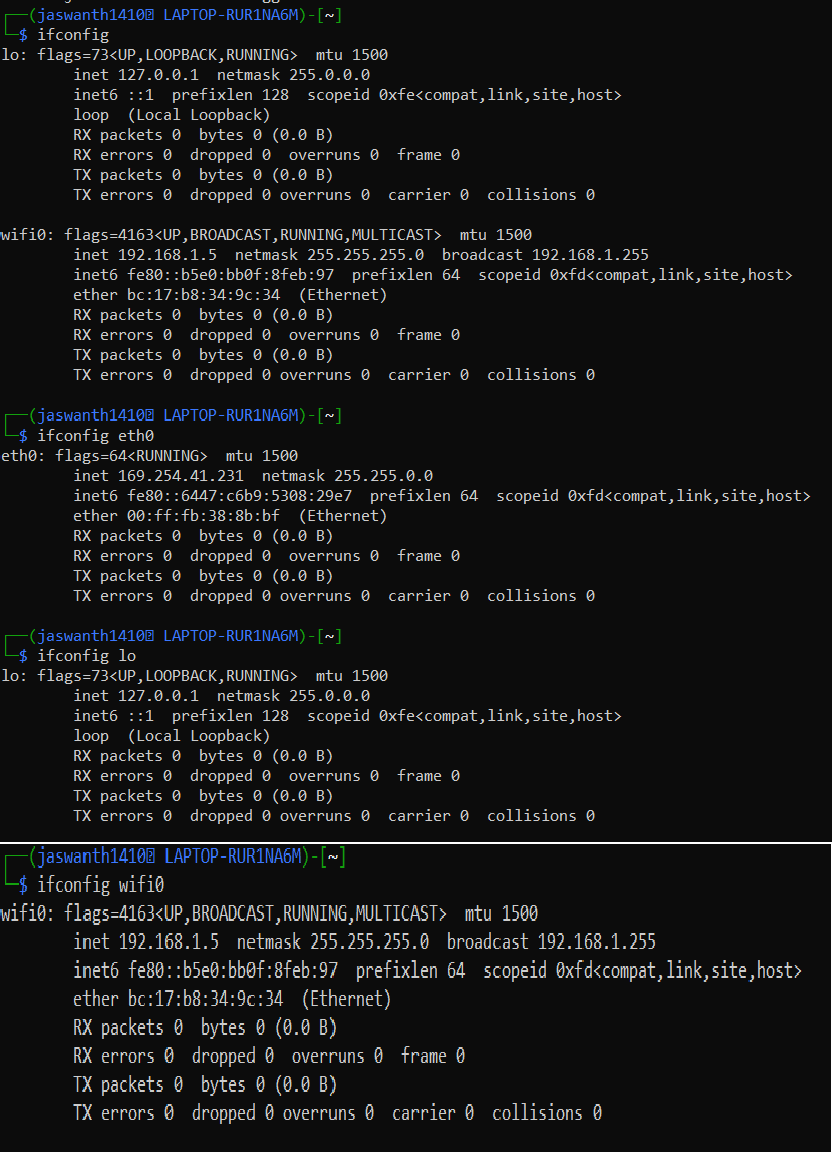
Basic information displayed upon using ifconfig are:

* + IP address
  + MAC address
  + MTU(Maximum Transmission Unit)

**Syntax**: ifconfig

Commands for specific interface ifconfig eth0

ifconfig lo ifconfig wlan0 **Example:**



1. Ip

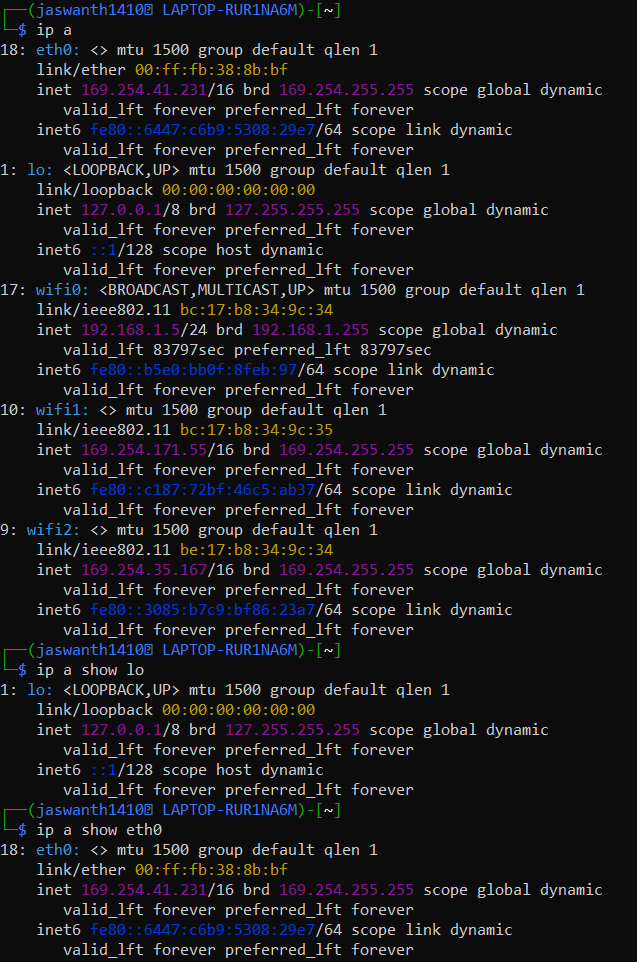
This is the latest and updated version of ifconfig command.

**Syntax:** ip a

ip addr commands for specific ip ip a show eth0

ip a show lo

ip a show wlan0

**Example:**

1. Ping(Packet INternet Groper)

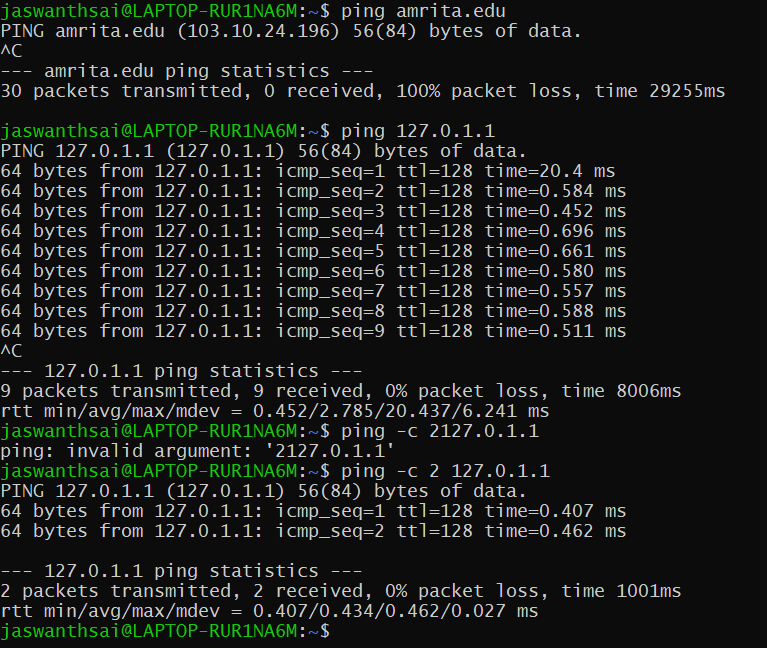
Linux ping is one of the most used network troubleshooting commands. It basically checks for the network connectivity between two nodes.

It keeps executing until it is interrupted.Use **Ctrl+C** Key to interrupt the execution.

**Syntax:** ping <destination>

Destination can be ip address or website .and ping can be terminated by

count ‘-c’

**Example:**

1. Netstat

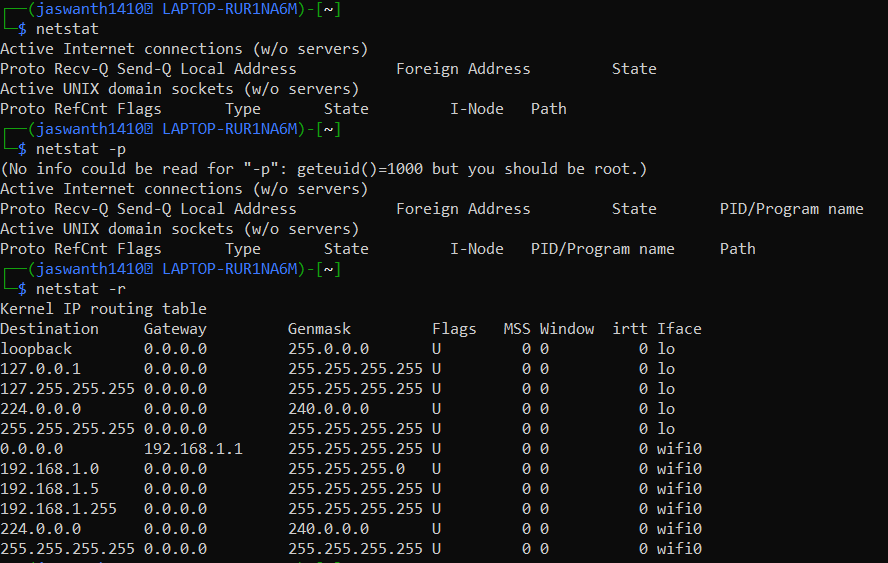
Linux netstat command refers to the network statistics. It provides statistical figures about different interfaces which include open sockets, routing tables, and connection information.

**Syntax:** netstat

netstat –p : It displays programs associated with open socket. netstat –s : It displays detiled statistics for all ports.

netstat –r : It displays routing table information

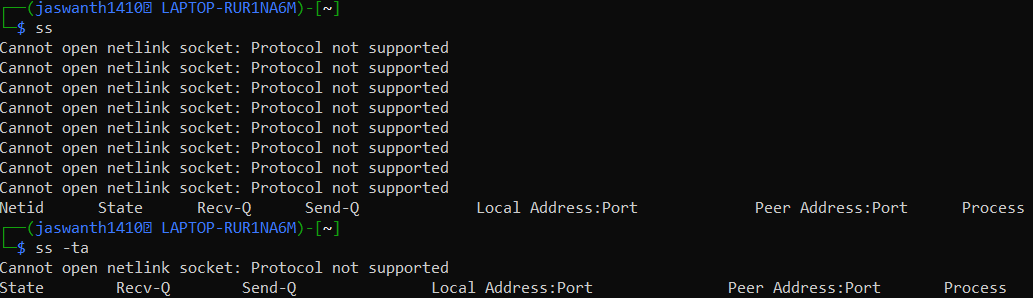
**Example:**



1. SS

Linux ss command is the replacement for netstat command. It is regarded as a much faster and more informative command than netstat. **Syntax:** ss

**Example:**

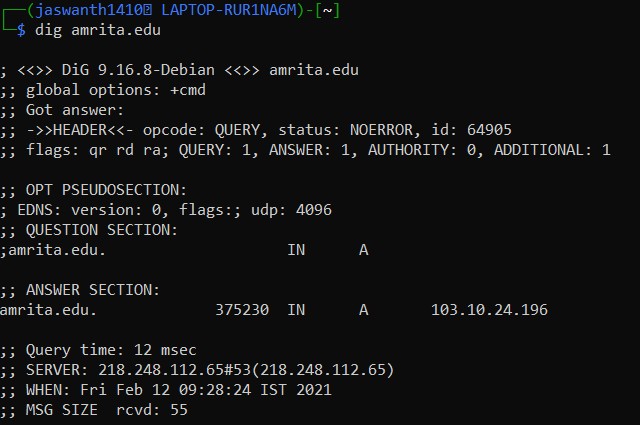


1. Dig

Linux dig command stands for Domain Information Groper. This command is used in DNS lookup to query the DNS name server. It is also used to troubleshoot DNS related issues.

**Syntax:** dig <domain name>

**Example:**

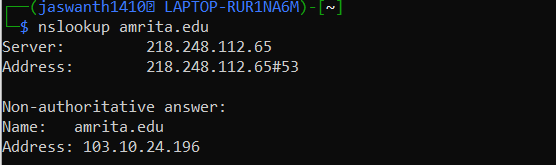


1. Nslookup

Linux nslookup is also a command used for DNS related queries. It is the older version of dig.

**Syntax:** nslookup <domin name>

**Example:**



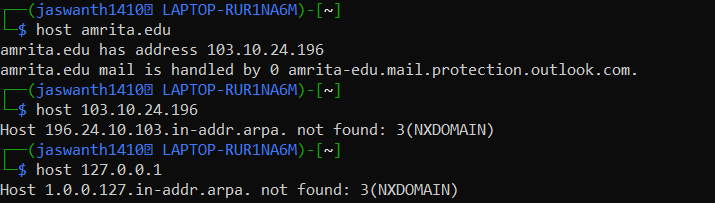
1. Host

*Linux* host command displays the domain name for a given IP address and IP address for a given hostname. It is also used to fetch DNS lookup for DNS related query.

**Syntax:** host <domin name>

host <ip address>

**Example:**

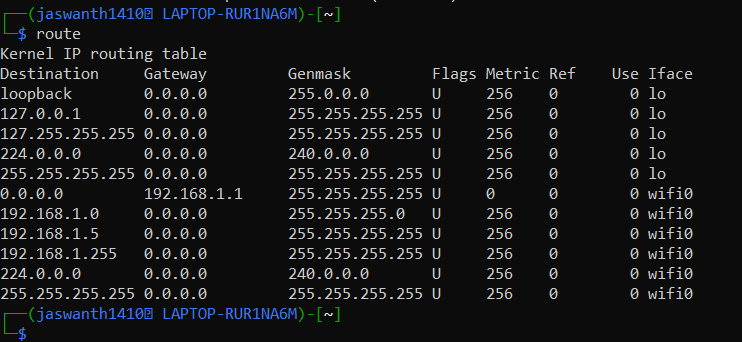


1. Route

Linux route command displays and manipulates the routing table existing for your system.

**Syntax:** route

**Example:**

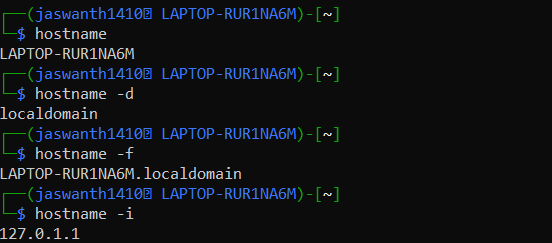


1. Hostname

Linux hostname is the simple command used to view and set the hostname of a system.

**Syntax:** hostname or hostname –“i||d||f”

**Example:**

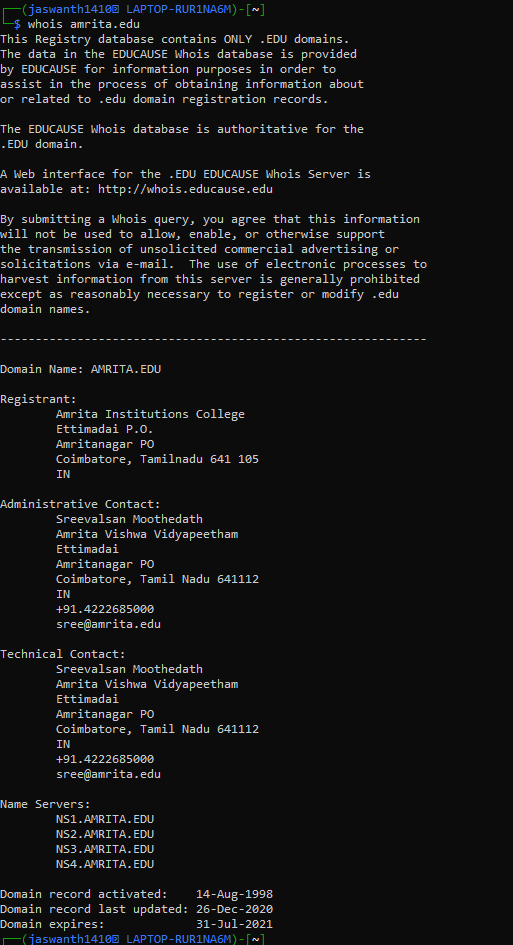


1. Whois

Linux whois command is used to fetch all the information related to a website. You can get all the information about a website including the registration and the owner information.

**Syntax:** whois <domain name>

**Example:**



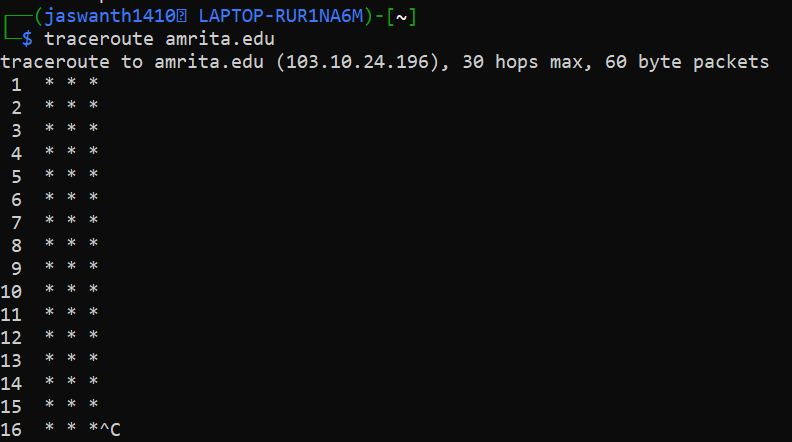
1. Traceroute

Linux traceroute is one of the most useful commands in networking. It is used to troubleshoot the network. It detects the delay and determines the pathway to your target. It basically helps in the following ways:

1. It provides the names and identifies every device on the path.
2. It follows the route to the destination
3. It determines where the network latency comes from and reports it.

**Syntax:** traceroute <destination>

**Example:**

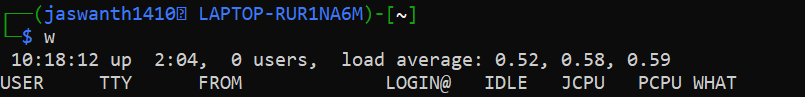


1. W

w prints a summary of the current activity on the system, including what each user is doing, and their processes.Also list the logged in users and system load average for the past 1, 5, and 15 minutes.

**Syntax:** w

**Example:**



1. Tracepath

Linux tracepath is similar to traceroute command. It is used to detect network delays. However, it doesn't require root privileges.

**Syntax:** tracepath <destination>

**Example:**



1. Iftop

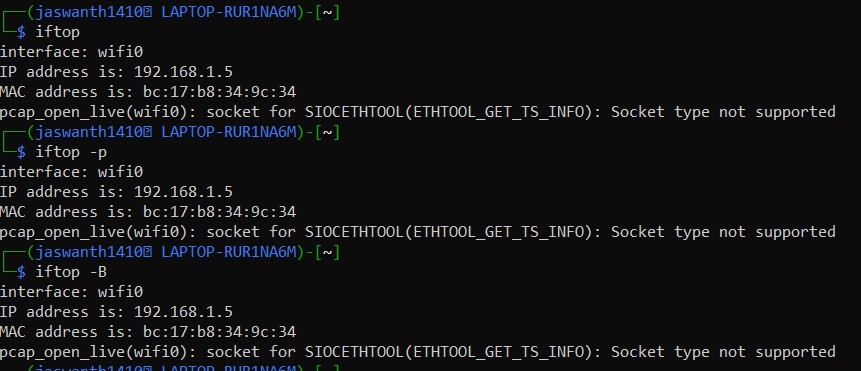
iftop utility or iftop command is used to monitor the traffic and display real-time results.

**Syntax:** iftop,

iftop -p :displays ports

iftop -B :display the output in bytes

**Example:**

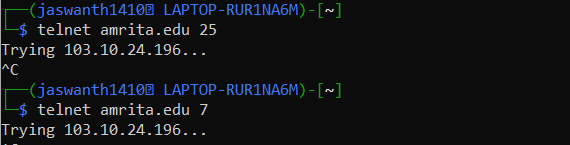


1. Telnet

It is used to communicate with another hostname via the telnet protocol, if telnet connection establishes on any port means connectivity between two hosts is working fine.

**Syntax:** telnet hostname <port>

**Example:**



1. Finger

It is used for viewing user information(i.e) displays a user’s login name, real name, terminal name and write status.

**Syntax:** finger

**Example:**

