Day 1

03-09-2021

Unix Training

Unix is an Operating system (OS) developed in Bell laboratories.

Unix is multi-tasking, multi-user, high secure and very high performance os

In Unix OS we can customize according to own requirement.

Unix is Open Source.

Unix is command base OS.

Unix flavors

Sun : Solaris

IBM : AIX

Apple : MAC

HP : HP-UX

Google : Android

Unix

And

Linux :it is GUI base

Shell : It is a command or instruction given to kernel through System call to execute some task.

Cd

Shell

1. Read line from standard input device (each of these line is called pipe line).
2. Splits the command into tokens.
3. Checks the token to see If it is a keyword.
4. Check the first word alias.
5. Then execute the command.

Unix Commands

1. echo
2. print
3. date --date ‘Date Format’: check other formats.
4. date +%d-%M-%Y : check other formats.

**File system :**

**pwd** : current directory path.

**ls** : it display all files and folder present in current directory.

mkdir folderName : This command is use to create the folder or directory

mkdir folderName folderName folderName : this command is use to create more than one folder.

rmdir folderName : This command is use to delete the folder.

cd folderName : this command is use to move inside a folder.

mv oldFolderName newFolderName : This command is use to change the folder.

folderName start with pre-fix . consider as hidden folder.

mkdir .folderName

ls : this command is use to display. Non hidden folder.

ls –a : This command I use to show normal as well as hidden folder.

Mv oldFolderName .folderName : This command is use to hide the folder.

cd .. : this command is use to move to parent directory of current directory.

cd ~ : this command is use to move to root directory.

Unix file system

Creating the file in Unix

1. cat command :
   1. creating file using cat command

cat > filename.txt : it create the new file with allow to enter the contents inside a file.

* 1. Reading the content from file using cat command

Cat < filename.txt or cat filename

* 1. Cat >> filename : this command is use to append the content in existing file.

1. Touch filename : This command is use to create the empty file

Touch filename filename filename

1. Using echo

echo “Welcome to Unix Training “ > filename

head and tail command :

head –n filename : this command is use to display the top n number line from a file

tail –n filename : this command is use to display the bottom n number of line from a file

wc : word count :

wc –l filename : number of lines

wc –w filename : number of words

wc –c filename : number of character

cp : copy the content from one file to another file.

cp sourcefilename destinationfilename.

Delete the file

rm filename

rm –I filename : it ask the confirmation to delete the file.

ls folderName : This command is use to display directory details.

rmdir folderName This command is use to delete the folder if folder is empty.

rm –r folderName This command is use to delete the folder doesn’t matter folder empty or contains set of files.

Filter command : flat file : in a file data entered by using delimiter ie space , \_, tab etc.

Employee.txt file with few records

cut -c 1 Employee.txt : 1 index position

cut -c 3 Employee.txt : 3 index position

cut -c 3-6 Employee.txt : 3 to 6 range index position

cut -d ' ' -f2 Employee.txt : -d delimiter ‘ ’ space f2 2nd columns

cut -d ',' -f1 Manager.txt : -d delimiter ‘,’

paste Employee.txt : display the content

paste –s Employee.txt : serial format

paste Employee.txt Manager.txt both file contents.

Tr ( translate command)

This command is use to translate data from one format to another format.

tr SET1 SET2 < Employee.txt

search SET1 content and replace by SET2 contents in file.

tr abcde ABCDE < Employee.txt

a-z A-Z < Employee.txt

tr ' ' , < Employee.txt

tr [:lower:] [:upper:] < Employee.txt : please work more pre-defined keywords to translate the content from file.

sort Employee.txt : ascending order as 1st field

sort –r Employee.txt : descending order

sort -k2 Manager.txt k2 send field

sort -k2 -t ',' Manager.txt -t delimiter ‘,’ by default space consider.

EmployeeInfo.txt

Id,Name,Salary,Desg, City, DeptId

4,Raj,19000,Developer,Bangalore,10

2,Raju,18000,Developer,Delhi,20

5,Ramesh,15000,Developer,Bangalore,20

8,Ajay,12000,Developer,Delhi,10

9,Vikash,19000,Developer,Bangalore,20

6,Ram,12000,Developer,Delhi,20

7,Vijay,18000,Developer,Bangalore,10

1. Sort by Id
2. Sort by city
3. Sort by City and sub sort by DeptId

Note : column name not sort(ignore).

GREP Commands GREP Global Regular Express Print

Grep command actually search then the content in file base upon the pattern.

The grep command mainly divided into three parts.

1st grep command followed by pattern to search in file and then filename or contents.

grep 'e' grepdemo.txt : display the line where e character present

grep –i 'e' grepdemo.txt : ignore case sensitive

grep -c -i 'E' grepdemo.txt : it display the number of line where E character present.

grep -n -i 'e' grepdemo.txt : it display the line which contains e character as well as it display the line numbers.

Please work more GREP Commands

SED : The SED command Unix stands for stream editing commands.

Using SED command we can modify the contents of the file.

SED command in Unix basically use search and replace the text or contents from a file.

Pipe commands

Pip e command is use to combine more than command

So first command execute it provide the output and that is input for another command.

echo “Welcome to Unix Training “ |

SED command

1. Read a line from input stream
2. Execute the command on a line
3. Display the result on output stream.

In SED command we use s and g

S : substitution

G : means many occurrence.

sed s/Raj/'Raj Deep'/g seddemo.txt

sed s/EE/i/ seddemo.txt

sed s/ee/i/g seddemo.txt

sed s/ee/i/ seddemo.txt

sed '2 s/R/r/g' seddemo.txt

sed -i s/Raj/'Raj Deep'/gi seddemo.txt : this command search, replace and update in same file.

sed s/'Hor r you'/'How are You'/g a.txt

SED commands

Day 2

06-09-2021

**Awk : awk is one of the most powerful tool in Unix use for processing the rows and column in the file base upon the delimiter.**

**Syntax**

**Awk ‘BEGIN {start\_action} {action} END {stop\_action}’ filename**

**Start action : initialization**

**Action : looping**

**End action : at the last**

**awk -F ',' '{print $0}' Employee.txt : display all rows**

**awk -F ',' '{print $1}' Employee.txt : display specific fields from a file**

**awk -F ',' '{print $2}' Employee.txt : display specific fields from a file**

**NF : Number of Fields**

**NR : Number of records**

**GREP : grep is useful if you want to quickly search data from line or file that match our pattern. It also return some other simple information like matching number, match count, and file names**

**SED : it is use when you want to make any changes in a file base upon the pattern match. If allow you to easily match part of line, makes modification and print out result or update in file.**

**AWK : it is entire programming language. Build around reading any type of file (CSV (comma separated values) style file, processing records and optionally print out the result.**

**File access permission**

**File**

**Types of users**

1. **File owner**
2. **Group owner**
3. **Others**

**U -> it represent the owner**

**G -> it represent group of owner**

**O-> it represents any other user but not belong to user group.**

**Permission for regular file**

1. **Open file : Read mode**
2. **Write file or modify : write mode**
3. **Execute : Read and Execute**

**ls –l filename : this command is use to display the properties of a file.**

1. **First field display - : means it is a file**
2. **The owner has over the file. rw**
3. **The group has over the file. r**
4. **Other people has over the file r**
5. **Fifth : number of links for that folder or file. 1**
6. **User that own the file or directory**
7. **Group details.**
8. **Size of the file in byte**
9. **Time when last modification happen**
10. **File name**

**chmod : change mode**

**chmod <permission> <filename>**

**code**

**a : all**

**u : user**

**g : group**

**o : other**

**+ : add**

* **: remove**

**R : read , w : write , x : execute**

**0 🡪 no permission**

**1 🡪 execute permission**

**2 -> write permission**

**4 🡪 read permission**

**4 + 2 +1 : 7 (read, write and execute)**

**4+0+1 : 5 (read, no write and execute)**

**4+0+0 : 4 (read, no write and no execute)**

**chmod 754 fileName**

**User 🡪 read, write and execute**

**Group 🡪 read, no write and execute**

**Other 🡪 read, not write and execute**

**Chmod 777 filename**

Shell scripting

A shell is a program that takes command typed by user and send the instruction to kernel through system call and Kernel run the command.

A shell is a program that acts as the interface between you and the Unix or Linux operating system.

Types of Shell

1. The Bourne Shell
2. Sh : Posix
3. Ssh : Korne shell
4. Bash :
5. C shell
   1. csh
   2. Tops tcsh

sudo apt install shellName

shell programming

we can do shell programming using two ways

1. You can type a sequence of command and allow the shell to execute them interactively.
2. You can store those commands in a file and you can run the file.

VI Editor

Open the file with vi editor

vi filename.txt

1. Command mode
2. Insert mode
3. Exit mode

Open the open the file to move from command mode to insert mode we have write i

Then write the contents

To come out from insert mode to command mode we have to click esc key.

To save the write :w

To quite the file :q

To save and quite : wq!

**Shell scripting**

You create the file and write one or more command and save the file with extension .sh

Variable : variable is a name which hold some value and the value can change during the execution of a program.

2 types

1. pre-defined variable : This provide the OS details. All pre-defend variable are in upper case. When we are using those variable we have to start with $ symbol.
2. User-defined variable

Taking the value through keyboards

1. Command line arguments.

sh ./filename firstargument secondarguemnt thirdargument

$1 : first parameter

$2 : second parameter

$3 : third parameter

$0 : script file name

$\* : number of arguments

$# : all arguments .

1. Using read property

-gt

-gte

-lt

-lte

-eq

-ne

Day 3

07-09-2021

Oracle SQL and Pl SQL

C, C++, Java, Python etc.

Input read a,b

Process sum = a+b

Output display sum

File handling programs

API : Application Programming interface.

Store the data permanently

1. File base system

File base system limitation

1. We can store same data again and again ie data redundancy(duplicate records).
2. Data is not a secure. (read or write mode).
3. CRUD Operation is very complex (Create, Read, Update and Delete).
4. Consistency ( format of the file, type of file etc).

Data : raw facts.

Information : processed data or meaningful data.

Database : it is used to store the data in table format using column and row.

DBMS : Database Management System. It is a software which help to store the data in a table format.

Excel DBMS.

Employee

Id Name Salary

100 Raj 12000

101 Raju 14000

102 Mahesh 16000

103 Rajesh 18000

Database Model

1. Hierarchical model: two tables or files are connected to each other from top to bottom like parent and child relationship.

1. Network model
2. Relational model : logical relationship between two tables.

Trainer\_Students\_Details

TId TName Tech Sid SName age

100 Raj Java 1 Seeta 21

100 Raj Java 2 Reeta 22

100 Raj Java 3 Meeta 23

100 Raj Java 4 Veeta 24

Trainer

PK

TId TName Tech

100 Raj Java

101 Raju Python

Students

PK FK

SId SName Age TSId

1 Reeta 21 100

2 Meeta 22 100

3 Veeta 23 101

4 Leeta 24 101

Database Model

RDBMS : Relational Database management system.

Dr. EF Codd’s Rules : 12 rules 0 to 11 rules.

In DBMS : Table column row

In RDBMS : Relation attribute or fields tuple

RDBMS :

Database Name Company

MySQL Sun Micro system (Oracle) : 8.x

Oracle Oracle

Db2 IBM

Sql Server 2020 Micro soft

Application Layer to interact with the Database Server

Console (Command prompt) or GUI

SQL : Structure Query Language :

SQL divided into 5 sub category.

1. DRL or DQL (Data Retrieval/Query Language)
2. DDL (Data Definition Language)
3. DML (Data Manipulation Language)
4. DCL (Data Control Language)
5. TCL (Transactional Control Language).

80 to 85% query is common in all RDBMS database.

MySQL : by default username in MySQL : root

show databases : to display all database available in MySQL Database.

use databaseName : to switch or move inside a database.

create database databaseName: This command is use to create the database

show tables; This command is use to display all tables present in that database.

Oracle : by default oracle username and password

scott and tiger.

show databases

in Oracle database the login name itself is database consider.

scott is database in my machine for oracle sever.

Oracle database provide 1 pre-defined table ie tab. This table help to display all pre-defined or user-defined table name available in your account.

This query is use to find all table available in our account.

select \* from tab;

select \* from tableName;

employees

departments

job\_history

jobs

locations

select \* from tableName;

RDBMS is case insensitive but values are case sensitive.

Day 4

08-09-2021

Select \* from tab; This command is use to display all pre-defined or user-defined table present in you’re a account.

Select \* from tableName;

Select \* from employees;

Select \* from departments

Select \* from locations;

Select \* from employees;

DRL or DQL : Data query or retrieval language

Select clause

select columnname, columnName,columnName from tableName;

desc employees; This command is use to provide the table structure.

Column alias : temporary name for the column

Select columnName as columnAlias from tableName;

select employee\_id as emp\_id,first\_name as fname,salary as emp\_salary from employees;

or

select employee\_id emp\_id,first\_name fname,salary emp\_salary from employees;

maths operation on column if column contains numerical value.

select employee\_id,first\_name,salary,salary+500 as gross\_salary from employees;

|| This operator is use to combine two columns as one column.

select first\_name||last\_name from employees;

Display the output as

Employee\_id, Full\_name, Basic\_Salary,HRA, DA, PF, Gross\_Salary

Salary is basic salary

Hra is 10% on salary

Da is 5% on salary

Pf is 7% on salary

Gross salary is hra + da –pf +salary

Where clause : where clause is use to filter the data.

Relational operators

select \* from tableName where columname RO value;

>

>=

<

<=

=

!= or <>

select \* from employees where salary > 15000;

select \* from employees where employee\_id < 150;

select \* from employees where salary = 17000;

select \* from employees where hire\_date='28-SEP-97';

select salary from employees where first\_name = 'Steven';

between operator (range selection)

select \* from tableName where columnName between minValue and maxValue

select \* from employees where salary between 5000 and 10000

select \* from employees where employee\_id between 110 and 120

oracle by default date format consider as dd-mon-yy

We can use between operator for date column also.

In operator (in operator is use to check more than one specific value).

Select \* from employees where employee\_id in(110,116,112,118);

Select \* from employees where job\_id in(‘IT\_PROG’,’ST\_MAN’,’PU\_CLERK’);

Like operator

Select \* from employees where first\_name=’Steven’

Select \* from employees where first\_name like ‘Steven’;

% zero or 1 or many

select \* from employees where first\_name like '%n' : end n character

select \* from employees where first\_name like 'A%' : start with A character

select \* from employees where first\_name like '%e%' : contains e character

select \* from employees where first\_name like '\_e%' : 1st character can be anything and 2nd must be 2 character.

not null : column doesn’t contains any value.

select first\_name,salary,commission\_pct from employees where commission\_pct is null;

and, or, not

and means both the condition must be true

or means any one condition must be true

not if condition true it make it false and vice-versa.

Select \* from employees where employee\_id=100 and first\_name like ‘Steven’;

Select \* from employees where job\_id like ‘IT\_PROG’ and salary > 10000;

Select \* from employees where job\_id like ‘IT\_PROG’ or salary > 10000;

Select \* from employees where first\_name not like ‘Steven’;

Select \* from employees where job\_id not like ‘IT\_PROG’;

Select \* from emplyees where salary not between 5000 and 10000

Select \* from employees where hire\_date not in(’01-mar-90’)

Select \* from employees where commission\_pct is not null

Order by clause : this clause is use to do the sorting may be ascending or descending order.

select employee\_id,first\_name,salary from employees order by salary asc;

select employee\_id,first\_name,salary from employees order by salary desc;

select employee\_id,first\_name,salary from employees order by salary;

we can do multi sorting

select employee\_id,first\_name,salary from employees order by salary asc;

select employee\_id,first\_name,salary from employees order by salary desc;

select employee\_id,first\_name,job\_id,salary from employees order by job\_id, salary desc;

Join :

Create table table1(srno int, name varchar(10), salary float)

Create table table2(accno int, name varchar(10), amount float)

Join is use to retrieve the records from one or more than one table with out without conditions.

1. Cartesian product :

select srno,salary, accno, amount from table1, table2;

output is m \*n

select srno,salary, table1.name,accno, amount from table1, table2;

select srno,salary, table2.name,accno, amount from table1, table2;

select srno,salary, table1.name,table2.name,accno, amount from table1, table2;

table alias

select t1.srno,t1.name,t1.salary,t2.accno,t2.name,t2.amount from table1 t1, table2 t2;

Equi – join : display common records present in both the tables.

Using where clause

select t1.srno,t1.name,t1.salary,t2.accno,t2.name,t2.amount from table1 t1, table2 t2 where t1.name = t2.name;

equi join also known as inner join.

On clause

select t1.srno,t1.name,t1.salary,t2.accno,t2.name,t2.amount from table1 t1 inner join table2 t2 on t1.name = t2.name;

select t1.srno,t1.name,t1.salary,t2.accno,t2.name,t2.amount from table1 t1 join table2 t2 on t1.name = t2.name;

outer join

left outer join : common + left side or first table remaining records

where clause :

select t1.srno,t1.name,t1.salary,t2.accno,t2.name,t2.amount from table1 t1, table2 t2 where t1.name = t2.name(+);

on clause

select t1.srno,t1.name,t1.salary,t2.accno,t2.name,t2.amount from table1 t1 left outer join table2 t2 on t1.name = t2.name;

right outer join : common + right side or second table remaining records.

Where clause

select t1.srno,t1.name,t1.salary,t2.accno,t2.name,t2.amount from table1 t1, table2 t2 where t1.name(+) = t2.name;

on clause

select t1.srno,t1.name,t1.salary,t2.accno,t2.name,t2.amount from table1 t1 right outer join table2 t2 on t1.name = t2.name;

full join : common + left and right side both table remaining records.

select t1.srno,t1.name,t1.salary,t2.accno,t2.name,t2.amount from table1 t1, table2 t2 where t1.name(+) = t2.name;

Employees 10 employees

6 employee working in projects.

Projects 5 projects

3 project started.

Equi join and inner join : display all employee details in project with project details.

Left outer join : display all employee details with the people working in project as well as not working in projects.

Right outer join : display project details with employee details if any employee working in that project or not.

Equi join

select emp.first\_name,emp.salary,dept.department\_name from employees emp,departments dept where emp.department\_id=dept.department\_id;

left outer join

select emp.first\_name,emp.salary,dept.department\_name from employees emp,departments dept where emp.department\_id=dept.department\_id(+);

right outer join

select emp.first\_name,emp.salary,dept.department\_name from employees emp,departments dept where emp.department\_id(+)=dept.department\_id;

self join : joining the same table itself is known as self join

while join concept it is not mandatory column name must be same but it contains same type of values with data types.

select emp1.first\_name,emp1.job\_id,emp2.first\_name from employees emp1, employees emp2 where emp1.manager\_id= emp2.employee\_id;

Reports

Self join

Lex is a Ad\_VP Report to Steven, He/She is Pred.

Self join with equi join : makes sure records not repeat.

Lex is a Administration Vice President Report to Steven, He/She is President.

Alexzander is a Programmer Report to Steven, Report to He/She is Administration Vice President.

Oracle functions

Function is to write set of instruction to perform specific task.

Function mainly divided into two types.

1. Pre-defined function
2. User-defined function : Pl SQL

Oracle provide lot of pre-defined function

2 types

1. Single row function : This function apply the function functionality to each records individually and return every row of tables.
2. Multi row function : This function apply the function functionality to all table records or base upon the group and it return only one single output.

Every function take one or more than one parameter and return the value.

String function

select first\_name,upper(first\_name) from employees;

select \* from dual; This function is use to check the function functionality.

select upper('raj'), lower('RAJ'), initcap('raj') from dual;

select substr('rajdeep',2),substr('rajdeep',2,5) from dual;

select length('raj deep') from dual;

number function

select round(345.678,0) from dual;

select round(345.678,1) from dual;

select round(345.678,2) from dual;

select round(345.678,3) from dual;

select round(344.678,-1) from dual;

select round(344.678,-2) from dual;

select round(644.678,-3) from dual;

select trunc(345.678,0) from dual;

select trunc(345.678,1) from dual;

select trunc(345.678,2) from dual;

select trunc(345.678,3) from dual;

select trunc(344.678,-1) from dual;

select trunc(344.678,-2) from dual;

select trunc(644.678,-3) from dual;

select ceil(10.1), ceil(10.9), floor(10.1), floor(10.9) from dual;

date function

select sysdate from dual;

select ceil(10.1), ceil(10.9), floor(10.1), floor(10.9) from dual;

select sysdate from dual;

select add\_months(sysdate, 2) from dual;

select next\_day(sysdate,'Mon') from dual;

select months\_between(sysdate, '01-jan-21') from dual;

1. display all employee first\_name, salary, numberofyearexp from employees
2. display only those employee details whose year of experience is > 20 employee first\_name, salary, numberofyearexp from employees

DDL and DML

Day 5

09-09-2021

Multi row functions

Multi row function also known as aggregate functions.

Sum, max, min, avg and count

Multi row function combine more than one records base upon the groups. By default the whole table itself is consider as one group. We can make sub group base upon some property or attribute.

select sum(salary) from employees;

select max(salary) from employees;

select min(salary) from employees;

select avg(salary) from employees;

select count(manager\_id) from employees

select count(commission\_pct) from employees;

select count(employee\_id) from employees;

select count(\*) from employees;

select first\_name,salary,commission\_pct from employees;

select first\_name,salary,nvl(commission\_pct,0) from employees;

select count(nvl(commission\_pct,0)) from employees;

group by clause

select sum(salary) from employees group by job\_id;

select sum(salary) from employees group by department\_id;

select sum(salary) from employees group by manager\_id

select job\_id, sum(salary) from employees group by job\_id;

select job\_id, sum(salary) from employees group by job\_id;

select department\_id,sum(salary) from employees group by department\_id;

select department\_id,sum(salary),max(salary) from employees group by department\_id;

having clause : having is like a where clause but having clause must be use after group by clause.

Where clause condition apply for individual records and having clause condition apply for group of records.

select department\_id,sum(salary)from employees group by department\_id having sum(salary) > 50000

select department\_id,sum(salary)from employees group by department\_id having sum(salary) > 50000

select department\_id,sum(salary)from employees where department\_id <> 100 group by department\_id having sum(salary) > 50000

select department\_id,sum(salary)from employees where department\_id <> 100 group by department\_id having sum(salary) > 50000 order by department\_id asc;

sub query

query within another query is known as sub query.

Syntax

Outer query (inner query)

Outer query (inner Query (Inner Inner query))

Find the person name whose salary is greater than average salary of all employee working in organization or working in some particular department.

In sub query inner query will execute and it generate the output or result and that output is input for outer query.

Inner query can return single row or multiple row(make sure query must be retrieve single column).

Single row sub query : aggregate operator or condition with primary key.

Relational operator >, <, >=, <=, =, <> or !=

Multi row sub query: in,

relational operator with any

relational operator with all

Inner query if return more than one records I you want to apply relational operator with any or all.

First we have to check inner query maximum and minimum condition values.

Max 🡪12000

Min 🡪 6900

>all : greater than maximum salary of inner query.

select first\_name,salary from employees where salary >all (select salary from employees where department\_id=100);

>any : greater than minimum salary of inner query

select first\_name,salary from employees where salary >any (select salary from employees where department\_id=100);

exists

select \* from employees where EXISTS (select \* from departments where department\_id=1000)

if inner query return records then outer query display the records.

DDL and DML :

Data definition language

Syntax to create the table

select avg(salary) from employees;

select avg(salary) from employees where department\_id=100

select first\_name,salary from employees where salary > 6461

select first\_name salary from employees where salary > (select avg(salary) from employees)

select first\_name from employees where job\_id in (select job\_id from employees where department\_id=90)

select first\_name from employees where department\_id in (select department\_id from departments where location\_id=1400)

desc departments

select \* from departments

select first\_name,salary from employees where salary >any (select salary from employees where department\_id=100);

select \* from employees where EXISTS (select \* from departments where department\_id=1000)

Create table

Syntax

Create table tableName(columnName datatype, columnName datatype)

create table employee(id int, name varchar(10),salary float,dob date)

create table employee(id number(2), name varchar2(10),salary number(10,2),dob date)

DML : Data Manipulation language

DDL :

drop table name; It remove table with all records.

Alter command

1. add new column to existing table.

alter table employee add desg varchar(2)

1. drop column from existing table

alter table employee drop column dob

1. modify the column data type size.

alter table employee modify desg varchar(10)

DML

Insert

Update :

Update tableName set columnName = value;

update employee set salary = 45000

Above query update all records but if you want to update with condition then we have to use where clause.

update employee set salary = 35000 where id =100;

update employee set salary = 37000 where name like 'Ravi';

both the condition must be satisfy.

update employee set salary = 40000 where id = 102 and name like =’Ramesh’;

if any condition satisfied

update employee set salary = 40000 where id = 102 or name like =’Ramesh’;

if we want to update more than value with one or more condition

update employee set salary = 42000, name = ‘Raj Deep’ where id = 100;

delete query :

delete from tableName; : it delete all records.

delete from employee

Delete with specific conditions.

delete from employee where id = 100

delete from employee where id = 101 and name like ‘Raj’

delete from employee where id = 101 or name like ‘Raj’

This is query is part of DDL

truncate table tableName; it remove all records but maintain the table structure.

delete, drop and truncate difference

truncate and drop is a part of DDL

drop : it remove table structure as well as all records.

Truncate : it remove all records but maintain the table structure.

With truncate we can’t use where clause.

Truncate we can’t use rollback (TCL command).

delete is part of DML

delete remove all the records if we fire query without conditions.

We can use where clause.

If we deleted record using delete query we can rollback with TCL concept (rollback and commit).