3. Interactive Sql and Performance Tuning

⇒ SQL.

Structure Query Language
Sql stands for Structure query language used to communicate with a database. It is the standard language for Relational database management system.

Sql Statemend are used to
Sql Statement are used to perform different operation on database insertion, updation, Deletion of data.

Sql is used in various advance Relational database management system. Some common Rdbms that used Sql are mysql, oracle microsoft access, microsoft Sql etc.

⇒ Charecteristics of SQL.

1) Sql allow the use to create, update, delete Data from a database.

2) It is small as possible very simple and easy to learn.

3) SQL is very powerful language / all the keywords of Sql can be express in any combination of all the upper & lower case ~~forechet~~ character.

Ex:

Upper case = Update / UPDA
Lowercase = Update
Mix = Update

4] Sql work with database program like DB_2 ~~DBMS to~~ oracle, Msexcess, mysql etc

① High speed
② Portable Run on any platform
③ No coding require
④ use for relational DB
⑤ Easy to learn & understand.
⑥ Complete language for a database.
⑦ multiple data use.
⑧ Client - Server language.
⑨ Using internet.

⇒ Datatypes in Sql.

Sql store the data in table format. table is the combination of rows & tuples and columns (fields) while creating table use have ~~to~~ assign datatype to the columns this datatypes are use to design decide that which type of data the column can store

1] Char (length)
This datatype accept character or string type of data including unicode its known as

Fixedlength datatype:

2] VARCHAR (length) (10)

3] Boolean
This datatype can accept value true or False No need to declare size while declaring this datatype
ex : true, false

4] SmallInt
This datatype is used to accept numerical value it store any value between range $2^n - 15$
ex: -3, 2, 7, 6, 8, 0

5] Integer (int)
This datatype is used to accept numerical value it is store any integer value between the rang $2^n - 31$.

6] (float) Decimal
This datatype is used to accept floating point value.
ex:
decimal (3.14)

7] float (p)
It This datatype accept appropriate number Value.

ex: 0.2

8] Date
This datatype value no need to assign size while declaring a data type
Datavalue should be specify in the from
#time yyyy : MM : DD → 2024 : 02 : 02

9] Time
This datatype accept time value no parameter are require when declaring HH:MM:ss a ty time Datatype.

⇒ Component of SQL / Data language:

1] DDL
DDL → Data defination language

This language allow the user to define the datatype and their relationship to other types of data

It is used to create datatables, dictionaries and files within database

The DDL also used to specify the structure of each table, set of value with each attribute security and authorization information for all the tables & physical storage structure of all the table on the disk.

Statement that come under ddl

① Create

To create the Database

CREATE TABLE table_name

column 1 datatype [size] :
column 2 datatype [size] ;

Ex:

CREATE TABLE Employee

Emp_Id int [10],
Emp_Name char [20]

Emp_n —11— ;

② Alter

To alter the structure of data/ used to add, delete, or modify columns in an existing table

ALTER TABLE table_name ;

- Add column_name datatype [size] ;

- modify Column Column_name new_datatype [siz

ex:

modify column ~~Col~~ Salary decimal (50);

Changing a datatype & size.

- Drop
The table Delete Permenantly from the database

Syntax:
drop table table_name
ex:
drop table emp;

- Rename
To Rename the data

Alter table can also be used to Rename a table
Syntax:
Alter table table_name
RENAME column column_name to new_column name

DDM
→ DML (Data manupulation language)
It is used for accessing and manuplating data in a database.

DML Provide a set of functionality to support the Basic ~~th~~ data operation on the data Stored in the database it allow user to insert, update & delete from the database

① insert

1] insert

To insert record to the table after creating of table the Insert Command is used to insert one or more Records in the tables.

```
(
insert into tablename Values (a,b,c);
```

ex:

insert into ~~tat~~ Student values (a,b,c);

2] update

Some time change to the database become necessecary to the main make change in the database update command is used

updation can be done in Single or multiple columns ~~best~~ based on the given Condition

Syntax:

Update table_name set new_data Column Condition

Ex:

Update Student SET marks = 80 where name = "Vaishnavi";

3] Delete
Per requirement the Record from existing table can be give using delete command.

Delete from Student.

3] DCL [Data Controll language]
The DCL is used to control the user access to the database related elements like tables used Views, Functions, procedure and Pacakages. It is provide different levels of access to the object in the database

GRANT
Whenever we coont to work with database if we dont have authority to database when we cant perform any operation in the database like insert, update, delete etc.

GRANT SELECT, INSERT, UPDATE, DELETON table name to User name;

Statement is used Remove to user by System.

Grant <privileges> on <table name> to <username>

Revoke <privileges> on <table name> from <username>

Revoke Select on emp from user1;
Grand Select on emp to user1;

4] Data language (TCL)
Traction control language
TCL statement allows used to control and manage transection to maintain the integrity of data with in sql statement

1] Commit;
to use save data permenantly of Database.
Syntax:
Commit

2] Rollback
one step back
Syntax:
Roll back

3] Savepoint

Q. Consider college schema Employee (Employee_no, Employee_name, Department, Employee_Salary, location); Solve the following query.

→

List all managers in mumbai location

Select * From Employee where location = "mumbai";

Intigrity Constructs

This are rules define in a database to maintain data Accuracy consitency and reliability they ensure that data store in the database following Predefined Rules or conditions.

Domain
Referential integrity.

1) Domain
In this type to the attributes a domain of permitted values is associated for attribute it define the default value the Range values or specific values the Domain integrity are easy to test when data is enter

the domain integrity check that whether the attribute having proper and write eight value in the database or not

Domain integrity means it is the collection of valid set of values for an attributes

① Not Null
It means column does not hold a null value where for a specific column no value is provided while inserting record into a table.
by default it take null value null is applied to avoid insertion of any null values in the specific column.
consider the table student having name field which not null.

Roll_no	Name
1	mahi
2	Preeti
3	

create table student (Roll int(5), constraint cn93 notnull, name varchar(20));

② check
This is used to set user defined for the column as per the requirement for business with we are developing the application we may have to set some rules while inserting or updating data on specific field.

create table Student (Roll int(5),
name varchar(20); age int(5), constraint
cns3 checkage between 15 & 20);

2] Referential integrity
It is also known as foreign key
In this type one field is common in
between two tables.
Foregin key Represent Relationship between
table where there is parant child
Relationship
between two table having common value
the master table can be refrence as
parent while the transaction table
is consider as child the common
field while work as primary key in
parent table while foreign key in
child table.

Ex:
consider training institute database
having two tables Course details and
Student.
There is a condition that the student
may Register for Course which are
a Javailable in a Institute currently. and
not for the course which are not
offer at the movement to Specific this
rule like inserting value into database
foreign key is used as course details
and Student.

⇒ Course Details

course_name	Course_code	fees
CS Engr	1154	10,000
CO Engr	1102	11,0000
IT Engr	1152	150000

⇒ Student table

student_id	student_name	course code
101	Vashnavi	1154
102	Sanskruti	1102
103	Prachi	1152

course_code references courses_details (course code)

In both table the field course code is common in course detail is refer as primary key & if Student table it is a refer as foreign Key.

So after assign foreign key to Student table the record entry for new student will not accept 0, course code which is not available in the master table course details.

PAGE NO.:

SQL operators.

An operator is a character or reverse word used in SQL Statement to perform different operations like Arithmatic, Logical, Relational SET operator.

operator are used to specific conditions in a SQL Statements & also used to integrate multiple condition in SQL Statement.

1] Arithmatic operator.

① Add (+)

This operator used to add to operands

~~SELECT Salary + where.~~

```
SELECT Salary Salary +5000 as new
Salary From emp;
```

② Sub (-)

This operator used to sub two operands

```
Select Salary Salary - 2000 as newsalary
From emp;
```

It is substract right hand operand & left hand operand.

③ multiply (*)
It is multiply both operands

Select Salary salary * 2.5 as increase from Salary emp;

④ Division (/)
It is divide left hand operand by right hand operands.

Select Salary Sal/2 as salary from emp;

2] logical operator
logical operator in Sql are used to combine multiple conditions

1] AND
The and operator is used to combine two or more condition all conditions connected by AND must be true for record to be selected.

= Select hier_date AND years from emp;

= Select * from emp where department = sales AND Salary > 50,000.

2] OR

The OR operator is used to combine two or more conditions, if any of condition connected by OR is true the Record will be selected.

Select * from emp where department = 'sales' OR department = 'marketing';

3] NOT

Is used to condition return true or False

Select * from emp NOT department = 'sales'

4] IN / between

Select * from emp
where IN ('sales', 'marketing')
AND salary between 40,000 AND 60,000;

3] Relational, Compression operator.

① equal to (=)

Check whether both the operands have same value if condition become true

Update Emp set salary = 10,000 where person name "—".

(III) grater than (>)
check whether left operand is greater than right if yes condition become true

Select * from emp where Salary > 5000 ;

(IV) Less than (<)
check whether left operand is less than right if yes condition become true.

select * from emp where Salary < 10,000;

(V) Less than equal to
Check whether left operand is less than or equal to the right operand or not if yes condition become true

Select * from emp where Salary <= 20,000

(V) greater than equal to
Check whether left operand is less than or equal to the right operand or not if yes condition become true

Select * from emp where Sal >= 20,000

(VI) NOT equal to (!= / <>)
Check whether both the operands have same value if not condition become true

Select * from emp where Sal != 2000;

SET operator

SET operators are supported by Sql to be perform on table data. For this operation Special operators known as Set operators are used to use the to join the result of multiple Select statement.

this operater help to get meaningful result from data under different specific conditions

query which contain Set operator are called Compound queries

The differentiate operator are as follow

1] union
2] union_all
3] intersection
4] minus

Consider following two table employee & department

emp_no	emp_name	Job	Dept_no	Salary
101	Rahul	manager	10	17000
102	Vinay	Clerk	20	10000
103	Kunal	manager	30	18,000
104	Rajesh	Salesman	20	13000
105	Kushal	Clerk	10	11000

Department table

Dep_No	Dep_Name	Location
10	Sales	Nashik
20	Production	Pune
30	Accounts	Mumbai
40	Research	Bangalore

1] Union
This operator returns all the rows selected by either query.

Select column_name from table 1 union
Select Column_name From table 2.

2] Union all
This operator return all c rows selected by either query including duplicates.

Select Column_name From table 1 union all
Select column_name From table 2.

3] intersection
This operator return only this row which are common to both the queries.

Select Column name from table 1 Intersect
Select Column_name From table 2.

Dept no
10
20
30

4] minus
this operator display the row which are present in the 1st query but absent in the 2nd query with no duplicate & data is arranged in ascending order by default.

Select Column_name from table1 minus
Select Column_name from table2

ex:
Select column_name from Dep minus emp Select column_name from table

40

- Inbuilt Function
Various function

1] String function
① Lower string
Return the string in lower case.
Select lower("VAISHU" from table_name;

② upper string
written the string in uppercase.
select upper("vaishu") from table_name;

③ INITCAP (string)
to convert 1st letter of each string in capita latter

Select initcap ('xyz') from emp;

output : Xyz

④ LPAD (string) (str1, n, str2)
written string 1 left padded to length n with the character string specify in string2

Select lpad ('xyz', 10, '*') from emp;
*******xyz

⑤ RPAD (str1, n, str2)
written string 1 right padded to length n with the character specified in string 2.

Select rpad ('xyz', 10, '*') from emp;

xyz*******

⑥ LTRIM (string, char)
Remove character from the left beginning of the string

Select ltrim ('xyz', 'x') from emp;

o/p yz

⑦ RTRIM (string, char)

Remove character from the ending of the string

Select Rtrim ('xyz', 'z') from emp;

o/p xy

⑧ Translate (string, from, to)

Replace sequence of character in string with another ~~state~~ set of character this is the character to character replacement.

Select translate ('Jack', 'J', 'b') from emp;

⑨ length (string)

contain the length of string

Select length ('xyz') from emp;

2] Arithmatical function

This function rather than the aggrigate function there are some numerical function.

① Power (m, n)
To find n th power to the number

Selelect power(2,2) from emp;

② Round
It will round between It will round up number m upto n digits.

Select Round (100256, 2) from emp;
100.256 , 100.25

③ Sqrt
To find out square root of given number

Select sqrt(25) from emp;
5

④ greatest (ex1, ex2, ex3 .. n)
written the greatest in list of expression

Select greatest (4, 5, 20) from emp;

⑤ ~~aggregate function~~

PAGE NO.:

3] Agreegate function

Agreegate function perform a calculation on set of value and coritten a single value usely this function ignor null value

① min (minimum)
This function corittens small value from specific column of the table.

Select min (column_name) from table_name;

ex:
Select min (Salary) from emp;

② max (maximum)
This function coritten greatest value from specify column of the table.

Select max (column_name) from table_name;

③ Sum
This function coritten sum of all the value of specify column of the table.

Select Sum (column_name) from table_name;

ex:
Select + Sum (Salary) from emp;

④ Average [Avg]

This function written average of all the valu of specific column of the table.

Select Avg (colum_name) from table_name;

ex:

Select Avg (salary) from emp;

⑤ Count

This function written total number of valu of specific column of the table

Select count (column_name) from table_name

ex:

select count (emp_no) from emp;

* Sql Clause

In DBMS inbuilt is a built in function that helps to fetech the required records from database table a receive a condition expression therefore column name or some terms involve the column calcculate the result based on the given statement in the expression.

emp table

emp_no	empname	Job	Salary	departmnt_no
7839	Vaishnavi	president	50000	10
7698	Nakshtra	manager	30,000	30
7782	Mahi	manager	25,000	10
7566	Pooja	manager	28000	20
7788	Sanchita	Salesman	10000	20
7902	Sanskruti	Clerk	12000	30
7369	Samruddhi	Clerk	11000	10
7499	Rutuja	Salesman	13000	20

1] GROUP BY

Sql Group by statement is used to arrange identical data into groups the group by statement is used with the Sql Select statement

The Group by statement is used with aggregate function.

SELECT column_name, aggregate function from table_name Group by column name;

Ex:
SELECT dept_no, Sum (Salary) From emp Group by dept_no;

dept no	Sum (Salary)
10	86,000
20	51,000
30	42,000

2] HAVING :
It is used to specify the search condition for a group or aggregate function. Having is used in Group by

If you are not using Group by then you can use having function like, where

SELECT column_name, aggregate function from table-name Group by column name Having column = condition ;

Ex:
SELECT Dept_no, Sum (Salary) from emp GROUP BY Dept_no HAVING Dept_no = 10;

dept no	Sum (Salary)
10	86000

3] ORDER BY

To arrange the display Row in ascending or in decending order or given field (column), order by

Sorting a ascending and desending record.

SELECT ~~Column_name~~ * from emp order by emp_name;

Select * from emp order by emp_name desc;

⇒ Joint

A Joint is a means for combine columns from one or more tables by using values common to each. In the database having large amount of data it is not possible to store in the entire data in a single table the related data may be stored in different table

The employee table contaiint the basic information of employee like emp_name, emp_number, emp_post, emp_salary, Dept_no, etc.

The Department table contaiint the information of some employee about their department name & location.

if we want to display hole information Basic information with department name & location then we have to combine this two table in query in using Joint.

for using Joint we require a common column between both tables in this example the employee & department table Contains the a column dept_no.

Emp table

emp_name	emp_no	emp_post	emp_salary	Dept_no
Vaishnavi	1	Manager	100000	1001
Keyur	2	HR	200000	1002
Raj	3	Accountant	200000	1003

Department table

Dept_no	Dept_name	location
1001	It	Pune
1002	CS	Kolkata
1003	DS	Nashik

→ Cartesion product / Cross Joint.

A cross Joint perform relation product or cartesion product of two tables specify in query in dbms cross Joint ! where condition is missing in query or some invalid operation in where lead to understand result or cross Joint

At least one column should be common it means second table column

Emp table

emp_id	emp_name	Dept_id
101	Nayan	540
102	Tejas	550
103	Ram	560

Dept table

dept_id	Dept_name
540	Chemical
550	HR
560	Research

First table Column Add 2nd table Column
3+2 = 5 Columns in output

First table tuple into second table tuple
3 x 2 = 6 rows

Cartesian product is $R_1 \times R_2$
(Relation 1 to Relation 2)

output

emp_id	emp_name	Dept_id	dept_id	dname
101	Nayan	540	540	Chemical
101	Nayan	540	550	HR
102	Tejas	550	540	Chemical
102	Tejas	550	550	HR
103	Ram	560	540	Chemical
103	Ram	560	550	HR

SELECT Column List From table 1 cross Joint table 2.

Ex:
~~SELECT emp_id, name from~~
SELECT emp_id, emp_name, Dept_id, dept_id, Dept_name From emp cross Joint department;

This query automatically select common column and ~~Common~~ Joint table base on that Column Hence no need to specify name

of name of column common explicity but joint table must have common column with same name otherwise error.

⇒ inner joint

The inner join is used to display record that have matching value in both table

Syntax:
SELECT Column_name list from table 1 inner join table 2 on table 1. Column_name = table 2 . Column_name ;

E_name	Job	Salary	Dept_no
Vaishnavi	HR	50000	1
Gayatri	HR	10000	2
Raj	manager	40000	3

Dept_no	Dept_name
1	Chemical
2	account
3	Software

SELECT E_name, E_salary, E_depno, dept_no, Dept_nam
From employee inner join dept ON Employee.
dept_no = dept_no

output :

E_name	Sal	dept_id	dept_name
xyz	15000	10	Chemical
PQR	10000	20	account
STJ	20000	30	Software
Cat	15000	10	Chemical

The inner join T1 Select all rows from both table employee and and department as long as there is match between the column number if there are record in the department table that do not have match in employee then such Records will not get display ~~in department table Department operation of number.~~

In Employee table employee operation or no 40 is present but it is not display as no matching record is available in table department

⟹ Outer Join

outer join is based on both match and unmatched data outer join sub divided further into

A]. Left outer
b] Right outer
c] Full outer

Consider following two tables student data 1 and student data 2.

Student 1

Roll no	name
1	XYZ
2	PQR
3	STU
4	abc
5	etc

Student 2

Roll_no	Location
1	pune
2	mumbai
3	nanded
4	pune
7	Aurangabad
8	Hydrabad

A] Left outer

The SQL left join returns all rows from the left table even if there are no match in the right table null value are show at the place of right table value

SELECT column_name is from table 1 left outer join table 2 on table 1. column_name = table 2. column name ;

SELECT * From Student 1 left outer join student 2 on student 1. Roll_no = Student 2 (Roll_no ;

output

Roll_no	name	Roll_no	city
1	xyz	1	Pune
2	PQR	2	mumbai
3	STU	3	nanded
4	abc	4	Pune
5	efa	Null	Null

B] Right outer join

Return all row the Right table even if there are no matches in the left table null value are show the Place of left table value.

output

Roll_no	name	Roll_no	address/city
1	xyz	1	Pune
2	PQR	2	mumbai
3	stu	3	nanded
4	abc	4	pune
Null	Null	7	Aurangabad
Null	Null	8	Hydrabad

Select * from student 1 Right outer Join student data 2 ON Student 1. Roll_no = student 2 . Roll_no ;

3) Full outer Join :
The Full outer Join return a result with the matching data of both the table and remaining rows of both left table then the right table Null value are show the place of table value.

output :

Roll_no	name	Roll_no	city
1	xyz	2	Pune
2	PeeR	3	mumbai
3	sty		nanded
4	abc	4	pune
5	efg	Null	Null
Null	Null	8	Aurangabad
Null	Null		hydrabad

SELECT * from student 1 Full outer Join Student 2 ON Student 1. Roll_no = Student 2 . Roll_no ;

Nested Sub queries

A Sub query is a query with in query Sub query is a query appear with in where or HAVING of another query

Syntax:

SELECT . . .
From . . .
WHERE . . . (=, <, >) Subquery
HAVING . . (IN, NOTIN) Subquery

Outer query is called as main query and inner query which is written in (where, OR, Having)
Main query is called Sub query
Sub query in the where the result of the Sub query (inner query) is used to select some row From the main query

Sub query in the HAVING the result of the Sub query (inner query is used to select some groups from main query.

Syntax:
SELECT SELECT_list from table WHERE Expression operator (SELECT SELECT list From table);

Expression operator can be of two types like A & B
B] Single row operator (>, <, >=, <=, <>)

② multiple row operator (IN, ANY, ALL)

① Single row operator
A Sub query is 1 which returns only one row to main query
A Sub query which used single row operator in above syntax is called as single row Sub query.

Find all employee whose salary less than "Vaishnavi"

Select e_name, salary from employee where salary <# (Select salary from employee where e_name = "Vaishnavi");

o/p

Ename	Salary
Vaishnavi	50,000
Keyur	20,000

As only one row value is written by inner query (Sub query), so this type of query is called as single row Sub query.

(11) Multiple Row Subquery

If subquery returns more that one row then that types of query called as multiple row sub query.

A subquery which multiple row operator in above syntax is called as multiple row subquery

The ANY and ALL uses any one of the simple comparison operators to compare a test value to all of the values return by a subquery checking to see whether the comparison hold for some or all of the values

Sr-no	operator	Meaning
1.	IN	Equal to any member in the list
2.	ANY	Compare Value to each Value return by the sub query.
3.	ALL	Compare Value to every Value return by the sub query.

Q. Find out all employee having Salary not equal to any of the other manager Salary. He should not be manager.

Emp_id	Last_name	Job_id	Salary
~~101~~ 124	Keshavshetty	~~Vaishnavi~~ manager	58000
~~102~~ 141	Khakare	~~Keyur~~ Clerk	10000
~~103~~ 142	Deshmukh	~~Gayatri~~ Account	35000
143	Hukke	~~Vaishnavi~~ Clerk	30000
144	Joshi	Salesman	30000
178	Salunke	manager	68000

```
SELECT Emp_id, Last_name, Job_id, Salary from
Employee where Salary NOT IN (SELECT
Salary from employee where Job_id=
"manager");
```

Output

Emp_id	Last_name	Job_id	Salary
141	Khakare	Clerk	10000
142	Deshmukh	Account	35000
143	Hukke	Clerk	30000
144	Joshi	Salesman	30000

Q. Find out all employee having Salary less than any of the manager He should not be manager.

SELECT Emp-id, Last-name, Job-id, Salary From Employee cohere Salary < ~~NOT~~ ANY (SELECT Salary From employee where Job_id = "manager");

Output:

Emp_id	Last_name	Job-id	Salary
124	Keshavshetty	Manager	58000
141	Khakare	Clerk	10000
142	Deshmukh	Accaunt	35000
143	Hukke	Clerk	30000
144	Joshi	Salesman	30000

* Introduction of View / creating a View

A view is defined as a database that allows as to create a virtual table in the database whose content are defined by a query or taken from one or more tables

View is defined to hide complexity of query from user.

base table on which view is defined is called as based table.

Ex:
Consider a student table content following columns Student_id, Student_name, Student_div, address, Sport, fees

Student table

Now for a sport teacher require only Sport related data of student so we can create view called as Student_Sport view for teacher as below which will only Sport data of student display to Sport teacher.

Logical table: Create from

Types of View

1) Simple view
Logical table create from simple base table. does not hold data. The views which are based on only one table called as simple view.
Allow to perform DML (Data manupalation language) operation with some Restriction.
Query define Simple View cannot have any Join or grouping conditions.

Syntax:

Create View view_name;

As Select column list From table name;

Ex:

Student

Id	name	sport
1	Vaishu	Chess
2	Nayan	Khokho
3	Shonakshi	Kabaddi

Create view "v1" as select id, name from student ;

Select * from V1 ;

Output:

id	name
1	Vaishu
2	Nayan
3	Shonakshi

2] Complex View

Logical table create from more than one table Does not hold data, the view which are based on more than one table called as complex view

Do not allow Dml operation to be perform. Query Defined Complex view can have join or grouping condition

Syntax: Ex:
Create View V1: as Select id, sub from S1, S2 where S1.id = S2.sub.

S1

id	name	Sports
1	Vaishy	chess
2	Nayan	CRroom
3	Shonakshi	khokho

S2

id	name	sub
1	Vaishy	Kavitha
2	Nayan	Japnen
3	Shonakshi	Karen

output:

id	name	sub

* Advantage of View

1] Security: View can user from accessing all data
In Case of View only data that is given in View is accessible to user show so all data of based table is not accessible to user which will given you security of information.

Ex:
Sport teacher can see that related to Sport only, & View him from manupelating data

2] hide Complexity :
This view may be result of very complex query hence writing such complicate query again we can store such result to a view & access it whenever we want to access

So by writing query we can hide the complexity of original query.

3] Dynamic Nature :
Does not hold true in case based table if base table is drop or the column selected by view is alter

4] No direct access to data dictionary :
User cannot change data dictionary to damage database

5] Data integrity
if data is access through a view the dbms can automatically check a data to check for specific integrity

⇒ Dropping View

To drop a view we use drop view statement drop view statement require a name to the view.

To drop the view on should have must have previllage to from dbms to drop a view in database.

The drop view does not affect based table or any column of base table.

Syntax:

Drop View view_name (Restrict) (Cascade)

Restrict: Delete View only if there are is know other view depend on this view.

Cascade: Delete View along with all depend view on original view

ex:

drop view emp_view ;

Syntax:

Drop View <view_name> [cascade]

ex:

Drop view emp_view [cascade];

⇒ Modefying View

There are some situation in which some modefication is required in view define (the create) THE CREATE OR REPLACE statement in view Syntax is used to modify view this statement is used by SQL overvise the old view defined with new defination without any error like existing view with some name.

Syntax:

CREATE OR REPLACE View View_name AS Sub query;

Ex:

Consider A view define above it "IT" Faculty and change it to Select all IT Faculty having Salary above rs 20,000

CREATE OR REPLACE View IF Faculty As Select * from Faculty where Faculty Dep = 'IT' AND Salary > 20,000;

⇒ RENAME View

There are some situation in which some modification are required in the name of view

The rename statement in View Syntax is used to rename view

This statement is used by SQL to overwrite the old view name with new name.

- Range Searching operator (Between):

Between statement used to find out all tuple whose attribute value is ranging between given range of values.

This also include the boundary values.

Ex:
Find all teachers taken hr 10 to 20

SELECT * From Faculty where Hours between 10 AND 20;

Q2. Find all teachers not taken hrs between 10 to 20

→

```
Select * From Faculty where Hours Not between
10 AND 20.
```

- Pattern matching operation. (Like)

Pattern matching statement are used to find out particular string in attribute value.

LIKE is keyword that is used in the WHERE for pattern matching.

Determine whether a specific character string matching a specific pattern.

Sql provide 2 characters for use with LIKE percentage sign & underscore sign

1) [(%) moduls] Percentage (%)

Percentage sign means any string of zero or more characters

Percentage sign is used to define the start & ending or your string

2) Underscore [_]

underscore sign means single character underscore sign is used to determine later at particular position.

Syntax:
SELECT Column_name from table-name where
Column-name Like 'Patterm'

Ex:
Consider A faculty table

Faculty code	Faculty name
100	Yogesh
102	omprakash
703	Nitin
104	Mahesh
101	Amit

* Pattern 1: Determine a string whe which start with given a 'A' Letter find all faculty whose name start with A.

Select * From faculty where faculty name LIKE 'A%';

101 Amit o/p

Pattern 2:
Determine a string end with the given letter 'H'.

Syntax:

```
Select * from faculty where Faculty_name
LIKE '%.h';
```

output :

Faculty code	Faculty name
100	Yogesh
102	omprakash
104	Mahesh

Pattern 3:
Determine a string contain given a letter A fingd all Faculty where A is present.

Syntax:

```
Select * From faculty where faculty_name
LIKE '%A%';
```

output:

Facultycode	faculty name
102	omprakash
104	Mahesh
101	Amit

Pattern 4:
Determine a string in which given Pattern is combine of above forms.

Q. Find all faculty whose 2nd letter 'A' contain letter 'E' and name Ending with letter h

Select * from Faculty where faculty name LIKE '_a%e%h';

output:

Facultycode	Facultyname
104	Mahesh

Patten 5:
Determine a string in which given letter is present at particular position. Find all faculty whose name second letter is 'A'

Select * from Faculty where Facultyname LIKE '_a%';

output

104 Mahesh

Pattern 6:
Determine a string not contant given A letter

Syntax:
Select * from Faculty where faculty_name
Not LIKE 'Y.A%';

output:

Facultycode	Faculty_name
100	Yogesh
103	Nitien

⇒ Introduction of Sequence

is a user created database object that can be share by multiple user to generate unique integer

use for sequence is to create a primary key value which much be unique for each row

It is store & generate independently therefore the sum Sequence can be used for generating primary key of multiple tables.

A Sequence is share obrect it can be share between multiple users
You can either built code into application to generate primary key or used a

Sequence to generate unique members in database. So Sequence can replace application code.

Speed up the efficiency of accessing Sequence value when catch memory.

A Sequence automatically generate unique number by using a internal oracle counting

this can be time saving object because it can reduce the amount of application code needed to write a Sequence generating counting.

Sequence no are stored and generate independantly of tables therefore the same Sequence can be used for multiple tables.

Syntax:

```
Create Sequence <Sequence_name>
[increment by n]
[start with n]
[maxvalue n / No maxvalue n]
[Cycle / No cycle]
[Cache n / No Cache];
```

Syntax	Discription.
i) Create Sequence <Sequence_name>	It is the name of Sequence generator. Ex: Stud_id
ii) increment by n	Value of n Specify the step value between Sequence no.
iii) Start with n	Value of n Specify the first Sequence number to be generate from Sequence.
iv) Max value n	Value of n Specify the maximum value that Sequence can generate
v) minimum Value	Value of n Specify the minimum Sequence Value. minimum Value n.
vi) Cycle / No cycle	Specify wether the Sequence continue to generate values after it is a minimum or maximum Values. Defaut option is the No cycle.

viii	Cache, n / No cache	Value of n specify how many values the oracle server pre allocate and keep in memory. Default value of n is 20 there fore cache 20 values.

Start Value must be Value of minimum value & maximum Value Range.

Ex:
Create a Sequence name student Sequence to be used for the primary key of the student table the Sequence must start with 100 & Every key Value should be multiple of 5 it must not access gggg an a key value must not be repeted

```
Create Sequence < student_id_seq >
[increment by 5]
[start with 100]
[max Value gggg]
 No cycle
 No cache;
```

The above example create a sequence name student_id sequence to be used for the student_id column of the student table.

The sequence start at 100 does not all cache & does not cycle.

Generally we do not use the cycle option if the sequence is used to generate primary key values as it may defination of primary key.

Q. Create a sequence name stu_Rollno sequence to be used for the allow Rollno to student table max :18 increment 1

```
Create Sequence Stu_Rollno_Seq
[increment by 1]
[Start with 1]
[Max Value 80]
No cycle
Cache 10;
```

* Referencing Sequence

The Sequence is created is generate Sequential numbers for inserting in your tables

1] nextValue

Given Value of next no in a Sequence this is active whenever we call Sequence name.nextValue For a Sequence it will generate next no in a Sequence and no is generated it can not be taken back

You must qualify (next Value) Nextval with the Sequence name

It is not to use next value in Select querey we use this Time of inserting a row in a Specify table

Ex:

insert a new student name ("Vaishnavi", 101);

insert into student Values (stud_id_seq NEXTVAL , "Vaishnavi", 101);

2) Currval

Given number which is currently generated by a Sequence

nextvalue must be utilize to generate a Sequence number in the current user session before curve value is refer

Ex:
Find the currentValue for the student id sequence

Select stud_id_seq.currval from students.

3) Alter
Altering the Sequence

If we want to change any of parameter define in create statement then we will go for alter statement to change defination for define Sequence.

with help of alter statement we can change the increment value, maximumvalue, minimum Value, cycle option or catch option.

Syntax:

Alter Sequence Sequence_name
increment by n.
maximum value n / no maximum value
minimum value n / no minimum value

Cycle | No cycle
Catch n | No catch;

After Sequence Stud_id_seq
increment by 6
Start 200
MaxValue 99999
Nocache 20
cycle;

4) Deleting or Droping a Sequence:

To remove a Sequence from the data dictionary & from database we used the drop Sequence Statement.

You must be owner of the Sequence or have the drop Any Sequence to remove it

drop Sequence Sequence_name;

Index

An Index can be created in a table to access data in database more quickly

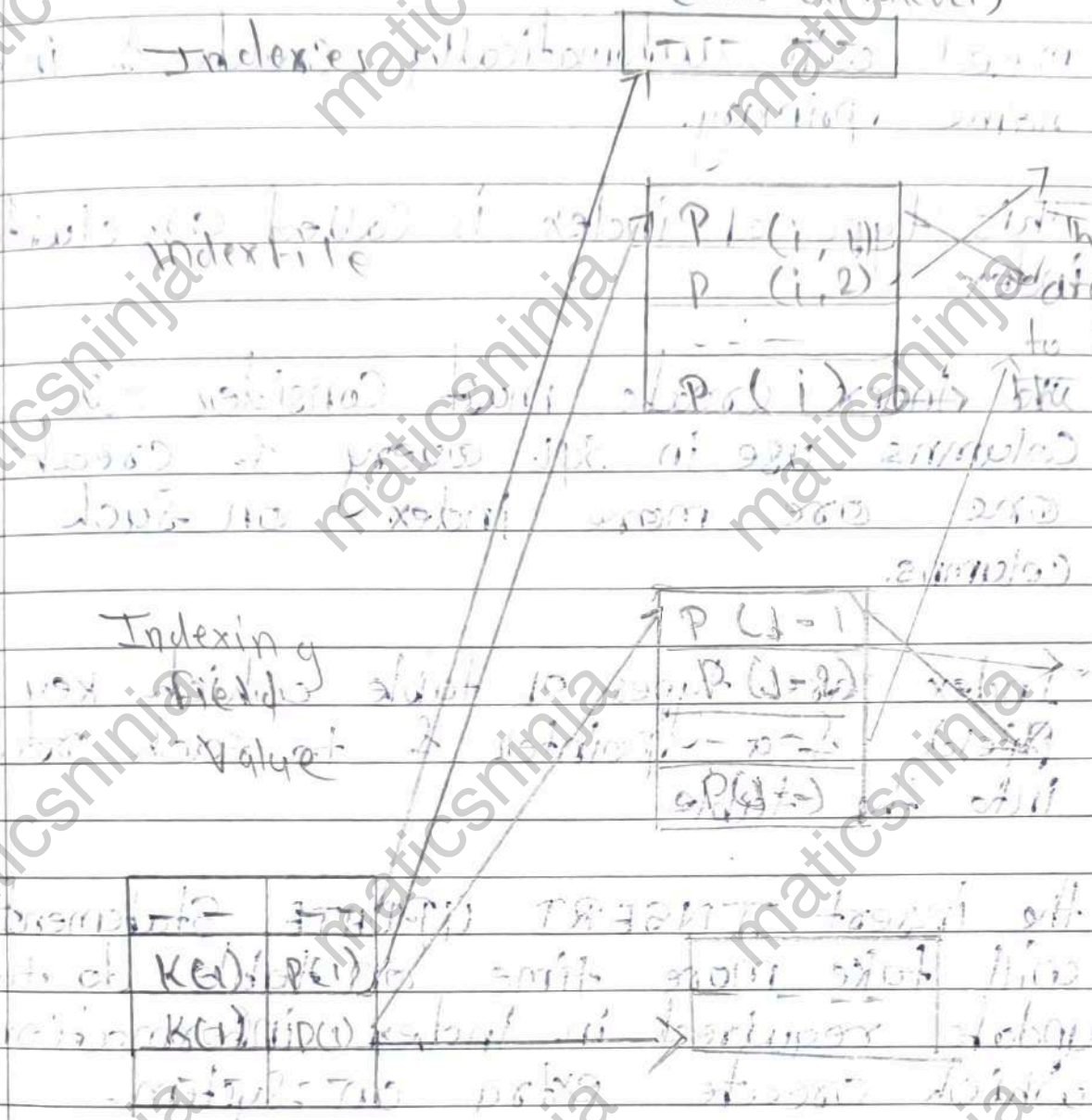
Mysql use Index to quickly Search rows with Specific Column Values as specify in query without and index mysql ~~can~~ engine need to ~~can~~ Scan the entire table to locate the row as a table Side increase the Search Speed will be reduced.

DataIndex is datastructure which being into the Speed of operations.

In a table index can be created using one ~~or~~ more column.

Need of Indexeing

Sometime While Fetching data one or more column in table are more repeting in a where of query then Index on Such column can be Improve Performace and Speed of data Retrive.



An Index will make use of some datastructure like B+T.

which will improve the speed of data retry on a table it may include some additional like operation some & storage for maintaining it.

when we create any column of a table with primary key or unique ~~or~~ key

mysql will automatically create & index name primary.

This type of index is called as clustered index.

The index create must consider all columns use in sql query & create one are more index on such columns.

Index are types of table which key field & a pointer & to each record into the table

the insert INSERT UPDATE Statement will take more time on table do to update required in index information which create extra on system.

The Select Statement will become fast on such table.

The primary index is a together with the data in the some table the clustere index will maintain the order of row in the table.

- Types of index

1] Simple index
2] Compasit index

1] Simple index
There are two types. Simple index is a unique index is created automatically when you define a primary key or unique or in a create table statement user can create non unique index on columns to speed up access to the Rows.

Syntax:
CREATE INDEX INDEN_NAME ON TABLE (column)

Ex:
CREATE INDEN ON last_NAME Colume in the employee table

CREATE INDEX Last_NAME_Index on employee (Last_name);

2] Composit index
A Composit index is an index created on mulltiples columns of a table.
mysql will permit to create a Composit index which include upto 16 columns.
A Composit index is also known as a multiple Column index.

The order of index column is very important and it should be arranged in descending order. as per their importance.

Syntax:

```
CREATE INDEX INDEX_NAME ON TABLE
(Column1, Column3, Column2, Columnn);
```

Ex:

CREATE INDEN on Department_name and age_column in the employee table.

```
CREATE INDEN Department_age_Index ON
employee (department, age);
```

- Unique Index

The unique value of one or more columns you of to use the primary key. Each table can have only one primary key.

Henced if you want to have a more than one column or a set of column with unique values you cannot use the primary key.

Mysql provide another kind of index called unique index that allow you to inforce

The uniqueness of values in one or more columns
Unlike the primary key index you can have more than one unique index per table

To create the unique index at the time of table creation we can use given syntax.

Syntax:

```
create unique index name on table name (index col_1, col_2)
```

View table index

To view the mysql internals perform this query at the beginning of the Select Statement

Ex.

```
Select * from employee where Job_title = "Sales";
```

Remove Index

Drop index

```
Drop index last name: index;
```