


INDEX

- INTRODUCTION OF TEACHER
- ACADEMIC CALENDER
- TIME-TABLE CLASSWISE
- TIME-TABLE PERSONAL
- SYLLABUS OF SUBJECT
- TEACHING PLAN
- PRACTICAL PLAN
- ATTENDANCE SHEET
- I. THEORY
- CENTRAL IDEA OF SUBJECT
- CHAPTERWISE REPLETION
- I. CENTRAL IDEA OF CHAPTER
- II. BIT BY BIT NOTES OF CHAPTER




Principal
New Satara College of B.C.A
Pandharpur.

Introduction of Teacher



Principal
New Satara College of B.C.A.
Pandharpur.

Introduction To Teacher

Name: Sanjivani Govind Atakare

Address: A/P-Degaon,
Tal-Mohol,
Dist-Solapur.

Contact No: 8459847880

Email Id: sanjivaniatakare@gmail.com

Qualification: BE(CSE)

Subject Name: Relational Database MySQL



Principal
New Satara College of B.C.A
Pandharpur.

TIME-TABLE CLASSWISE

TIME-TABLE CLASSWISE



Principal
New Satara College of B.C.A
Pandharpur.



New Satara College of BCA, Pandharpur
BCA-II Sem-IV
Time Table
Academic Year 2021-2022

PERIOD	1	2	3	01:00 To 01:25	4	5	6
TIME DAY	10:00 To 11:00	11:00 To 12:00	12:00 To 01:00		01:30 To 02:30	02:30 To 03:30	03:30 To 04:30
MONDAY	AJS (LAB)	Python (LAB)	AJS	L U N C H B R E A K	DS-II (LAB)	Python	C++
TUESDAY	AJS (LAB)	C++II (LAB)	AJS		Adv.Net.	ECL	Library
WEDNESDAY	MySQL (LAB)	C++II (LAB)	AJS		Adv.Net.	ECL	C++-II
THURSDAY	MySQL (LAB)	DS-II	C++		Adv.Net.	ECL	C++-II
FRIDAY	Python (LAB)	MySQL	Adv.Net.		DS-II	Python	C++-II
SATURDAY	Adv.Net.	MySQL	AJS		DS-II (LAB)	MySQL	C++-II

Sr.No	Subject	Staff Name
THEORY		
1.	OOPS with C++-II (C++)	Prof.Sonule A.P.
2.	Data Structures using 'C'- II (DS-II)	Prof.Pujari M.P.
3.	MySQL (MySQL)	Prof.Atakare S.G.
4.	Ethics and Cyber law (ECL)	Prof.Kulkarni B.P.
5.	Angular JS (AJS)	Prof.Godase U.R.
6.	Adv. Comp. Networks (Adv.Net)	Prof.Godase U.R.
7.	Python Programming (Python)	Prof.Kshirsagar V.S.
PRACTICAL		
1.	OOPS with C++-II (C++-II)	Prof.Sonule A.P.
2.	Data Structures using 'C'- II (DS-II)	Prof.Pujari M.P.
3.	MySQL (MySQL)	Prof. Atakare S.G.
4.	Angular JS (AJS)	Prof.Godase U.R.
5.	Python Programming (Python)	Prof.Kshirsagar V.S.




Principal
 New Satara College of B.C.A
 Pandharpur.

TIME-TABLE INDIVISUAL

TIME-TABLE INDIVISUAL



Principal
New Satara College of B.C.A.
Pandharpur.



New Satara College of BCA, Pandharpur
BCA-II Sem-IV
Time Table
Academic Year 2021-2022

PERIOD	1	2	3	01:00 To 01:30	4	5	6
TIME DAY	10:00 To 11:00	11:00 To 12:00	12:00 To 01:00		01:30 To 02:30	02:30 To 03:30	03:30 To 04:30
MONDAY				L U N C H B R E A K			
TUESDAY							
WEDNESDAY	BCA-II MySQL (LAB)						
THURSDAY	BCA-II MySQL (LAB)						
FRIDAY		BCA-II MySQL					
SATURDAY		BCA-II MySQL					BCA-II MySQL

Sr.No	Subject	Staff Name
THEORY		
1.	Relational Database MySQL (MySQL)	Prof. Atakare S.G.
PRACTICAL		
1.	Relational Database MySQL	Prof. Atakare S.G.



Principal
New Satara College of B.C.A
Pandharpur.

BCA (Science)-II Semester- IV

Course Code: DSC2D (Section-I)
Total Contact Hours: Hrs.
Teaching Scheme: Theory 3 Lect./Week

Course Title: Relational Database MySQL
Total Marks: 50(40 Lectures)
Total Credits: 02

Unit No	Content	No. of Lectures
Unit-1	Introduction to MySQL <ul style="list-style-type: none"> ▪ Installing and starting MySQL instance, History and Architecture of MySQL ▪ Components of MySQL -DML,DDL,DCL,DQL ▪ Data types in MySQL-Numeric, String, Complex, Date and Time, ▪ Creating databases and show databases 	4
Unit-2	MySQL Operators, Function and clauses <ul style="list-style-type: none"> ▪ MySQL operators- Arithmetic, Comparison, Logical, Bit, like ▪ MySQL Functions- Aggregate, Math, String, Date and Time, control flow functions and expressions, Type conversion, Formatting, Encryption ▪ MySQL clause-where, distinct, order by, group by, having, rollup. 	8
Unit-3	Performing Operation on Table Data <ul style="list-style-type: none"> ▪ Populating tables with data, Retrieving data from tables, Sorting data in a table, Deleting data from table, Updating data in tables, searching data ▪ Adding and Dropping columns, Modifying and Rename existing columns ▪ Renaming table using alter table, Changing a table type ▪ Finding out the tables created by user, Displaying a table structure ▪ Creating a table from a table, Inserting data into a table from another table 	8
Unit-4	MySQL constraints, Join and View <ul style="list-style-type: none"> ▪ Applying data constraints- column level and table level ▪ Types of Data constraints- <ul style="list-style-type: none"> • I/O constraints- Not null, Unique, Primary key, Foreign key, composite • Business rule constraints- Check, ▪ Adding, Modify and drop constraints using alter table command ▪ MySQL join:- Advantages & disadvantages of Join, Types of Joins ▪ MySQL View:- why view, Create, Update, Alter and Drop view 	8
Unit-5	SubQueries, Union and Indexing <ul style="list-style-type: none"> ▪ sub queries-use, example ▪ Set Operations- Union, Union all, Minus and Intersect ▪ Indexing:- Advantages and disadvantages of Indexing, creating index (simple, composite, unique), multiple indexing, drop index 	6
Unit-6	Stored Procedures, Transaction and cursor <ul style="list-style-type: none"> ▪ Stored Procedure:- Structure, use of stored procedure, Supported SQL statements in Procedures, creating dynamic procedure, Adding record to the table using procedure, procedure with IN,OUT,INOUT parameter, dropping procedure. ▪ Transaction :MySQL transactions, open and closing transaction, commit, rollback, savepoint in transaction, table lock ▪ Cursor:-use of cursor, types of cursor ,opening a cursor, fetching a record from the cursor, cursor fetch statement, closing cursor ▪ MySQL import & export- Import CSV File into MySQL Table, Export MySQL Table to CSV 	6

Reference Books: 1) MySQL(TM): The Complete Reference-Vikram Vaswani
2) Learning MySQL, by Seyed Tahaghoghi, Hugh Williams.
3) MYSQL 5 for professional, Ivan Bayross and Sharanam Shah


Principal
New Satara College of B.C.A
Pandharpur.



TEACHING PLAN

TEACHING PLAN



Principal
New Satara College of B.C.A
Pandharpur.



NEW SATARA COLLEGE OF B.C.A, PANDHARPUR
Teaching Plan

Name of Subject : A Relationl Database MySQL

ACADEMIC YEAR 2021-2022

Subject Code : Semester IV

NAME OF FACULTY: Sanjivani Atakare

Sr. No	Chapter Name	No. of Lectures	Sub. Topic be covered	L.R. / Media to be used	Date of commencement	Date of completion	Remark
1	Introduction to MySQL	1	Installing and starting MySQL instance	Black Board	11/4/22	11/4/22	
			History and Architecture of MySQL	Black Board	12/4/22	12/4/22	
		1	Components of MySQL -DML,DDL,DCL,DQL	Black Board	13/4/22	13/4/22	
		1	Data types in MySQL-Numeric, String, Complex Date and Time	Black Board	16/4/22	16/4/22	
				Black Board	18/4/22	18/4/22	
		1	Creating databases and show databases	Black Board	19/4/22	19/4/22	
2	MySQL Operators, Function and clauses	1	MySQL operators- Arithmetic, Comparison,	Black Board	20/4/22	20/4/22	
			Logical, Bit, like	Black Board	22/4/22	22/4/22	
		1	MySQL Functions- Aggregate, Math, String,	Black Board	23/4/22	23/4/22	
			Date and Time, control flow	Black Board	25/4/22	25/4/22	
		1	functions and expressions, Type conversion,	Black Board	26/4/22	26/4/22	
			Formatting, Encryption	Black Board	27/4/22	27/4/22	
			MySQL clause-where, distinct, order by,	Black Board	28/4/22	28/4/22	
		1	group by, having, rollup	Black Board	29/4/22	29/4/22	
3	Performing Operation on Table Data	1	Populating tables with data,Retrieving data from	Black Board	30/4/22	30/4/22	
			Sorting data in a table,Deleting data from table,	Black Board	2/5/22	2/5/22	
			Updating data in tables,searching data	Black Board	4/5/22	4/5/22	
		1	Adding and Dropping columns,Modifying and Rename existing columns,	Black Board	5/5/22	5/5/22	
			Renaming table using alter table,	Black Board	6/5/22	6/5/22	
		1	Changing a table type Finding out the tables created by user,	Black Board	7/5/22	7/5/22	
			Displaying a table structureCreating a table from	Black Board	11/5/22	11/5/22	
			Inserting data into a table from another table	Black Board	12/5/22	12/5/22	
			Applying data constraints- column level and table	Black Board	13/5/22	13/5/22	
	Types of Data constraints-I/O constraints- Not null, Unique,	Black Board	14/5/22	14/5/22			

Principal
New Satara College of B.C.A
Pandharpur.

4	MySQL constraints, Join and View	1	Primary key, Foreign key, composite Key	Black Board	17/5/22	17/5/22	
		1	Business rule constraints- Check, Adding,	Black Board	20/5/22	20/5/22	
		1	Modify and drop constraints using alter table command	Black Board	21/5/22	21/5/22	
		1	MySQL join:- Advantages & disadvantages of Join,	Black Board	22/5/22	22/5/22	
		1	Types of Joins	Black Board	23/5/22	23/5/22	
		1	MySQL View:- why view, Create, Update, Alter	Black Board	29/5/22	29/5/22	
5	SubQueries, Union and Indexing	1	sub queries-use, example	Black Board	30/5/22	30/5/22	
		1	Set Operations- Union, Union all,	Black Board	1/6/22	1/6/22	
		1	Minus and Intersect	Black Board	2/6/22	2/6/22	
		1	Indexing:- Advantages and disadvantages of Indexing,	Black Board	3/6/22	3/6/22	
		1	creating index(simple, composite, unique),	Black Board	4/6/22	4/6/22	
		1	multiple indexing, drop index	Black Board	8/6/22	8/6/22	
6	Stored Procedures, Transaction and cursor	1	Stored Procedure:-Structure, use of stored	Black Board	9/6/22	9/6/22	
		1	creating dynamic procedure, Adding record to the table using procedure, procedure with IN,	Black Board	10/6/22	10/6/22	
		1	Transaction:MySQL transactions, open and closing transaction, commit, rollback, savepoint in transaction, table lock	Black Board	11/6/22	11/6/22	
		1	Cursor:-use of cursor, types of cursor, opening a cursor, fetching a record from the cursor	Black Board	13/6/22	13/6/22	
		1	cursor fetch statement, closing cursor	Black Board	14/6/22	14/6/22	
		1	MySQL import & export-Import CSV File into MySQL Table, Export MySQL Table to CSV	Black Board	15/6/22	15/6/22	




 Principal
 New Satara College of B.C.A.
 Pandharpur.

Orders

ONUM	AMOUNT	ODATE	CNUM	SNUM
3001	18	10/3/2019	2008	1007
3003	767	15/3/2019	2001	1001
3002	1900	10/3/2019	2007	1004
3005	5160	20/4/2019	2003	1002
3006	1098	20/4/2019	2008	1007
3007	1713	10/5/2019	2002	1003
3008	75	10/5/2019	2004	1002
3010	4723	15/6/2019	2006	1001
3011	1309	18/3/2019	2004	1002

Solve the following queries using above databases and where clause range searching and pattern matching.

1. Produce the order no, amount and date of all orders.
2. Give all the information about all the customers with salesman number 1001.
3. Display the following information in the order of city, sname, snumand commission.
4. List of rating followed by the name of each customer in Surat.
5. List of snum of all salesmen with orders in order table without any duplicates.

Solve the following queries using above databases and group by clause.

1. Find out the largest orders of salesman 1002 and 1007.
2. Count all orders of October 3, 1997.
3. Calculate the total amount ordered.
4. Calculate the average amount ordered.
5. Count the no. of salesmen currently having orders.

Solve the following queries using above databases and formatted output and order by clause.

1. List all salesmen with their % of commission.
2. Display the no. of orders for each day in the descending order of the no. of.
3. Display order number, salesman no and the amount of commission for that order.
4. Find the highest rating in each city in the form: For the city (city), the highest rating is (rating)
5. List all in descending order of rating.
6. Calculate the total of orders for each day and place the result in descending order.

Solve the following queries using above databases and join.

1. Show the name of all customers with their salesman's name.
2. List all customers and salesmen who shared a same city.
3. List all orders with the names of their customer and salesman.
4. List all orders by the customers not located in the same city as their salesman.
5. List all customers serviced by salespeople with commission above 12%.

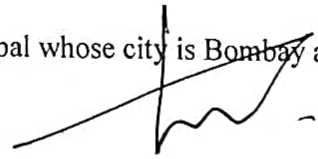
Solve the following queries using above databases and join and subquery.

1. Find all orders attributed to salesmen in 'London'.
2. List the commission of all salesmen serving customers in 'London'.
3. Find all customers whose cnum is 1000 above than the snum of 'Sejal'.
4. Count the no. of customers with the rating above than the average of 'Surat'.
5. List all orders of the customer 'Chirag'.

Solve the following queries using above databases and delete and update.

1. Remove all orders from customer Chirag from the orders table.
2. set the ratings of all the customers of Piyush to 400.
3. Increase the rating of all customers in Rome by 100.
4. Salesman Sejal has left the company. Assign her customers to Miti.
5. Salesman Miti has resigned. Reassign her number to a new salesman Gopal whose city is Bombay and commission is 10%.




Principal
New Satara College of B.C.A
Pandharpur.

ATTENDANCE

ATTENDANCE




Principal
New Satara College of B.C.A.
Pandharpur.

New Satara College of BCA, Korti-Pandharpur

Attendance Sheet

Academic Year -2021-2022		BCA-II Sem-IV																															Subject :- Relational Database M.Y.S.G.L		
Month- April																																	Name of Staff :- Atakore S.G		
Roll No	Student Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Total Present Days	Present In Percentage	
201	Babar Mansingrao Jitendra											P	P	P			P	P	A	P	P	P	P	P		A	P	P	P	P	P		13	87.5	
202	Bagal Sakshi Rajendra											P	P	P			P	P	P	P	P	P	P	P		P	P	P	P	P	P		15	93.75	
203	Bagal Omkar Baliram											A	A	A			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		0	0	
204	Chaware Parag Sunil											P	P	P			P	P	P	P	P	P	P	A		P	P	P	P	P	P		15	93.75	
205	Davare Kiran Pitambar											P	P	P			P	P	P	P	P	P	P	A		P	P	P	P	P	P		15	93.75	
206	Ghadage Sakshi Santosh											P	P	P			P	P	P	P	P	P	P	P		P	P	P	P	P	P		15	93.75	
207	Pakhare Shbham Apparao											P	P	P			P	P	P	P	P	P	P	P		P	P	P	P	P	P		14	87.5	
208	Randive Tushar Sanjay											P	P	P			P	P	P	P	P	P	P	P		P	P	P	P	P	P		11	68.75	
209	Ranjane Abhijit Mansing											P	P	P			P	P	P	P	P	P	P	P		P	P	P	P	P	P		15	93.75	
210	Sangitrao Sonali Bharat											A	A	P			A	A	P	P	P	P	P	P		P	P	P	P	P	P		14	87.5	
211	Shaikh Afroz Salim											A	A	P			P	P	P	P	P	P	P	P		A	P	P	P	P	P		7	43.75	
212	Shivpuje Kiran Pandurang											P	P	P			P	P	P	P	P	P	P	P		P	P	P	P	P	P		11	68.75	
213	Sorawale Ranjit Govind											P	P	P			P	P	P	P	P	P	P	P		P	P	P	P	P	P		13	81.25	
214	Ghatage Vaishnavi Vishwa											P	P	P			P	P	P	P	P	P	P	P		P	P	P	P	P	P		15	93.75	
215	Jagdale Pramod Ashok											A	A	A			A	A	A	A	A	A	A	A		A	A	A	A	A	A		0	0	
216	Sharma Gopal Santosh																																		
Total Absent Days												4	4	3			3	7	5	5	6	5	4		4	3	5	4	4	3					



Principal
New Satara College of B.C.A
Pandharpur

New Satara College of BCA, Korti-Pandharpur

Attendance Sheet

Academic Year -2020-2021

Subject :- Relational Database MySQL

Month- May

BCA-II Sem-

Name of Staff :- Atakare S. G

Roll No	Student Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Total Present Days	Present In Percentage		
201	Babar Mansingrao Jitendra	A		P	P	P	P					A	A	P	P			A	A	A	A	A		A	A	A	P	A	A		A	A	7	31.9		
202	Bagal Sakshi Rajendra	P		P	P	P	P					P	A	P	P			P	P	P	P	P		P	P	P	P	A	P		P	P	20	90.90		
203	Bagal Omkar Baliram	A		A	A	A	A					A	A	A	A			A	A	A	A	A		A	A	A	A	A	A		A	A	0	0		
204	Chaware Parag Sunil	A		P	P	P	P											A	A	A	A	A		A	P	P	P	P	P		P	P	15	62.18		
205	Davare Kiran Pitambar	P		P	P	P	P					P	P	P	P			A	A	P	P	P		P	P	P	P	P	P		P	P	20	90.90		
206	Ghadage Sakshi Santosh	A		P	P	P	P					P	P	P	P			P	P	P	P	P		P	P	P	P	P	P		P	P	21	95.45		
207	Pakhare Shbham Apparao	P		P	P	P	P					P	A	P	A			A	A	A	P	P		P	P	P	P	A	A		A	A	13	65		
208	Randive Tushar Sanjay	P		P	P	P	P					P	P	P	P			A	A	P	P	P		P	P	P	P	P	P		A	A	16	72.7		
209	Ranjane Abhijit Mansing	A		A	A	A	A					P	P	P	P			P	P	P	P	P		P	A	A	A	P	P		P	P	14	63.63		
210	Sangitrao Sonali Bharat	A		P	P	P	P					P	P	A	P			P	P	A	P	P		P	P	P	P	P	P		P	P	19	86.36		
211	Shaikh Afroz Salim	A		A	A	P	P					P	P	P	P			P	P	P	P	P		P	A	A	P	P	P		P	P	16	72.7		
212	Shivpuje Kiran Pandurang	P		P	P	A	A					P	P	A	P			P	P	P	A	P		P	P	P	P	P	P		P	P	18	81.81		
213	Sonawale Ranjit Govind	P		P	P	P	P					P	P	P	P			A	A	P	P	P		P	P	P	P	P	P		P	A	20	90.90		
214	Ghatage Vaishnavi Vishwa	P		P	P	P	P					P	P	P	A			P	P	P	P	P		P	P	P	P	P	P		P	P	21	95.45		
215	Jagdale Pramod Ashok																																			
216	Jadhav Anil Santosh																																			
	Total Absent Days	7		3	3	3	3					2	3	3	4			7	6	5	5	3		3	4	4	2	4	4		3	4				



(Signature)
Principal
New Satara College of B.C.A
Pandharpur.

New Satara College of BCA, Korti-Pandharpur		Attendance Sheet																																	
Academic Year -2021-2022		Subject :- Relational Database MySQL																																	
Month- June		BCA-II Sem-IV																																	
		Name of Staff :- Atokare S.G.																																	
Roll No	Student Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Total Present Days	Present In Percentage	
201	Babar Mansingrao Jitendra	A	A	P	P		A	P	P	P	A	P		P	A	P																		8	61.53
202	Bagal Sakshi Rajendra	P	P	P	P		P	A	A	P	P	P		A	P	P																		10	76.92
203	Bagal Omkar Baliram	A	A	A	A		A	A	A	A	A	A		A	A	A																		0	0
204	Chaware Parag Sunil	P	P	P	P		P	A	D	A	P	P		P	A	P																		10	76.92
205	Davare Kiran Pitambar	P	A	P	A		P	A	P	P	P	A		P	P	P																		9	69.23
206	Ghadage Sakshi Santosh	P	P	P	P		A	P	P	P	P	P		P	P	P																		12	92.30
207	Pakhare Shbham Apparao	P	P	P	A		A	A	P	P	P	A		P	A	P																		8	61.53
208	Randive Tushar Sanjay	P	P	P	A		A	P	D	P	A	P		A	P	P																		9	69.23
209	Ranjane Abhijit Mansing	P	A	P	A		P	P	P	P	P	A		P	P	P																		10	76.92
210	Sangitrao Sonali Bharat	A	P	P	P		A	P	P	P	P	P		P	A	P																		10	76.92
211	Shaikh Afroz Salim	A	P	P	P		P	P	P	P	P	P		P	P	P																		12	92.30
212	Shivpuje Kiran Pandurang	A	P	P	P		P	P	A	P	P	A		A	A	P																		8	61.53
213	Sonawale Ranjit Govind	P	A	P	P		P	P	P	P	P	P		P	P	P																		12	92.30
214	Ghatage Vaishnavi Vishwa	P	P	P	A		P	P	P	P	P	P		P	P	P																		11	84.61
215	Gadage Pramod Ashok																																		
Total Absent Days		5	6	1	6		6	5	3	1	3	5		4	6	1																			





Principal
 New Satara College of B.C.A
 Pandharpur.

CENTRAL IDEA OF SUBJECT

CENTRAL IDEA OF SUBJECT





Principal
New Satara College of B.C.A
Pandharpur.

CENTRAL IDEA OF SUBJECT

MySQL Relational Database is an assemblage of relational data that is structured or organized in the form of tables, columns, and rows, where tables represent the objects, columns represent the fields and rows represent the records. It is the broadly used relational database management system, as it is available for free of cost and available as an open-source for anyone to use. In this application, SQL (Structured Query Language) programming language is used to create, update, delete & manage the tables and its contents, as MySQL is supported with the basic SQL queries.





Principal
New Satara College of B.C.A
Pandharpur.

HOME ASSIGNMENT QUESTIONS

HOME ASSIGNMENT QUESTIONS





Principal
New Satara College of B.C.A
Pandharpur.

Assignment No-1

- 1] Explain the Architecture of MySQL.
- 2] Explain the Components of MySQL.
- 3] List the Data types of MySQL and explain with example.
- 4] Explain the Creating database and Show the database.
- 5] Explain the MySQL Operators with Example.




Principal
New Satara College of B.C.A
Pandharpur.

Assignment No-2

1] Explain the MySQL function with Example.


2] Explain the MySQL Clause with Example.

3] Performing Operation in table data. Populating table with data ,Retriving data from table , Sorting data from table ,Deleting data in table ,Updating data in table ,Searching data in table.

4] Explain Renaming table using Alter table ,Findout the tables created by user and change a table type.

5] Explain the Creating a table from a table. Inserting data into a table from another table.





Principal
New Satara College of B.C.A.
Pandharpur.

Assignment No-3

- 1] Explain the Join and its types. Advantages & Disadvantages.
- 2] Explain the View. Create, Update, Alter, Drop view.
- 3] Explain the set operations.
- 4] Explain the indexing. Advantages & Disadvantages of indexing.
- 5] Explain the types of data constraints.




Principal
New Satara College of B.C.A.
Pandharpur.

BIT BY BIT NOTES OF CHAPTER

BIT BY BIT NOTES OF CHAPTER




Principal
New Satara College of B.C.A.
Pandharpur.

Unit I - Introduction to MySQL

* History of MySQL :-

- MySQL is a relational database management system based on structured Query Language, which is designed for beginners and professionals. which is the popular language for accessing and managing the records in the database.

- MySQL is open source and free software under the GNU license. It is supported by Oracle Company.

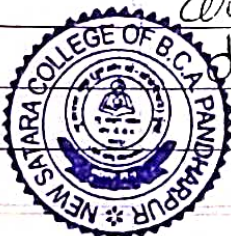
- Each database contains distinct API for performing database operations such as creating, managing, accessing and searching the data it store.

Today many databases available like MySQL, Sybase, Oracle, MangoDB, PostgreSQL, SQL server, etc. In this section, we are going to focus on MySQL mainly.

- It is open source software (database) which is supported by Oracle Company. It is fast, scalable, and easy to use database management system. Comparison with Microsoft SQL Server and Oracle Database.

* The project of MySQL was started in 1979 when MySQL's inventor Michael Widenius developed an in-house database tool called UNIREG for managing databases.

- After that, UNIREG for managing database and After some has been rewritten in several different language and extended to handle big database.



MySQL is named after the daughter of co-founder Michael Widenius whose name is 'My'.

* Feature of MySQL ~~Features~~ :-

- MySQL is a relational database management system (RDBMS) based on the SQL (Structured Query Language) queries. It is most popular language.

* The following are the most important features of MySQL

① Relational database Management System:-

- MySQL is a relational database management system. This database language is based on the SQL queries to access and manage the records of the table.

② Easy to use:-

MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements.

- MySQL consist of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.

③ Client/Server Architecture :-

MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs) which communicate with the server; that is they can query data save changes, etc.

④ Free download :-

- MySQL is free to use so that we can download it from MySQL official website without any cost.

⑤ It is scalable :-

- MySQL support multithreading that makes it easily scalable.
- It can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4GB. However, we can increase this number to a theoretical limit of 8 TB of data.

⑥ High flexibility :-

- MySQL supports a language number of embedded application which makes MySQL very flexible.

⑦ More efficiency is high because it has a very low memory leakage problem.



Principal
New Satara College of B.C.A.
Pandharpur.

⑧ High performance:-

- MySQL is faster, more reliable, and cheaper because of its unique storage engine architecture.

- It provides very high performance

- It has fast loading utilities because of the different cache memory.

⑨ Platform Independent.

* Architecture of MySQL:-

- Architecture of MySQL describes the relational database management system which is free open source software GNU ~~license~~ License.

- It is also supported by Oracle company.

- It fast, scalable, easy to use database management system.

x Architecture of MySQL describes the relational among the different components of MySQL system.

- MySQL follow Client-server Architecture. It is designed so that end user that

is clients can access the resources from computer that server using various networking services.

Architecture of MySQL contain following major layers:

- client
- server
- Storage Layer.

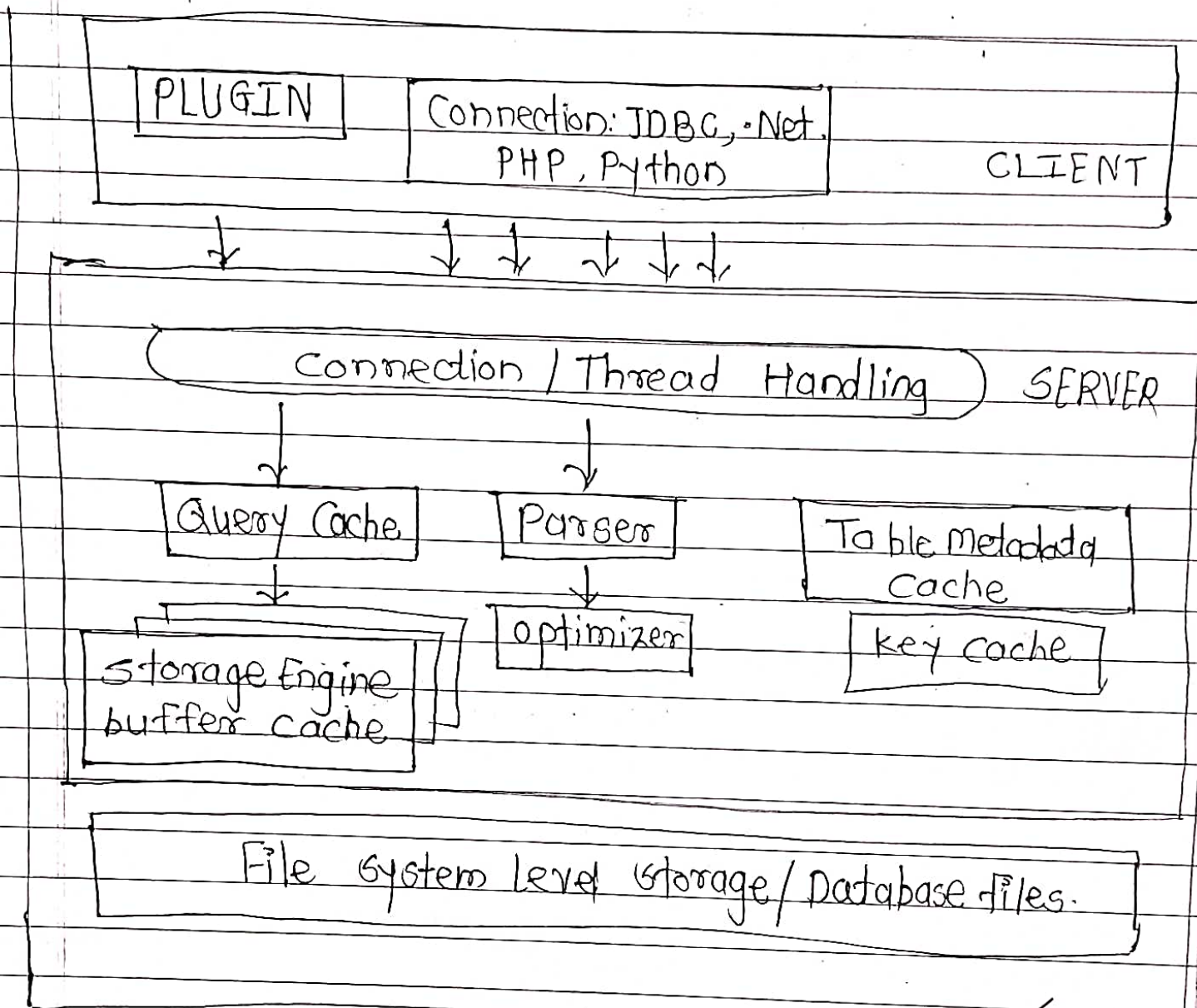


Fig:- Architecture of MySQL



Principal
New Satara College of B.C.A
Pandharpur.

* Client Layer :-

- This is the topmost layer in the above diagram.
- The client give request instructions to the server with the help of client layer.
- The client make request through command prompt or through GUI screen by using valid MySQL commands and expression.
- If the expressions and commands are valid then the output is obtained on the screen.
- So important services of client layer are

- Connection Handling
- Authentication.
- Security

⊛ Connection Handling :

- When a client send request to the server and server gives the response to the client will accept the request and the client connected. When client is connected to the server at that time, a client get it's own thread for it's connection. With the help of this thread all the queries from client side is executed.



*) Authentication :-

Authentication is speed is performed on the server side when client is connected to the MySQL server. Authentication is done with the help of username and password.

*) Security :-

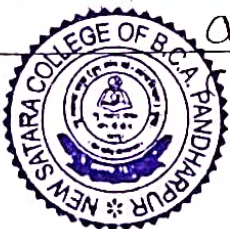
After authentication when the client gets connected successfully to MySQL server, the server will check that a particular client has the privileges to issue in certain queries against MySQL server.

* Server Layer :-

- The second layer of MySQL architecture is responsible for all logical functionalities of relational database management system is also known a "Brain of MySQL Architecture". When the client give request instructions to the server and the server gives the output as soon as the instruction is matched. The various sub components of MySQL server are:

* Thread Handling :-

- When a client send request to the server and server will accept the request and the client is connected to the server at that time, a client get it's own thread for it's connection.



Principal
New Satara College of B.C.A.
Pandharpur.

This thread handling of server layer. Also the queries of client side which is executed by the thread is also handled by Thread Handling module.

* Parser:-

- A Parser is a type of software component that built a data structure (Parse tree) of given input.
- Before parsing lexical analysis is done i.e. input is broken into number of tokens.
- After the data is available in the smaller elements parser perform syntax Analysis, semantics Analysis after that parser tree is generated as output.

* Optimizer:-

- As soon as the parsing is done, various types of optimization techniques are applied at ~~optimization techniques are applied~~ Block. These techniques may include rewriting the query, order of scanning of tables and choosing the right indexes to use etc.

* Query Cache:-

- Query cache store the complet result set for inputted query statement. Even before parsing, MySQL server consult query cache. when client write a query if the query written

by the client is identical in the cache then the server simply skip the parsing, optimization and even execution, it just simply display the output from the cache.

* Buffer and cache :-

- cache and will buffer store the previous query or problem asked by user.
- When user write a query then it firstly goes to query cache then query cache will check that the same query or problem is available in the cache.
- If the same query is available then it will provide output without interfering Parser, Optimizer.

* Table Metadata Cache :-

- The metadata cache is a reserved area of memory used for tracking information on databases, indexes, or objects. The greater the number of open databases, indexes, or objects, the larger the metadata cache size.

* Key cache :-

- A key cache is an index entry that uniquely identifies an object in a cache. By default, edge servers cache content based on the entire resource path and query string.



Principal
New Satara College of B.C.A
Pandharpur.

* Storage Layer:

- This Storage Engine layer of MySQL Architecture make it's unique and most preferable for developer's. Due to this layer MySQL layer is counted as the mostly used RDBMS and is widely used. In MySQL server, for different situations and requirement's different types of storage engines are used which are InnoDB, MyISAM, NDB, Memory etc.
- These storage engines are used as pluggable storage engines where tables created by user are plugged with them.

* Components of MySQL

DDL (Data Definition language)

DML (Data Manipulation language)

DCL (Data Control language)

DQL (Data Query language)

1) Data Definition language)

- DDL stands for data definition language. It is used to define database structure.
- It is used to create schema, tables, indexes, constraints etc. in the database.
- Data Definition language is used to store the information of metadata like the number of tables and schema, their name, indexes columns in each table, constraints etc.

Unit-II MySQL Operators, Function and Classes

*] MySQL Operators - Arithmetic, Comparison, Logical Bit, like.

D) MySQL Arithmetic Operators

Operator	What is done
+	Addition
-	Subtraction
*	Multiplication
/	Division; return quotient
%	Division; return modulus.

- Like most programming language, MySQL support most common arithmetic operators, allowing you to perform calculations on the fly. Table 6-4 list the arithmetic operators available in MySQL.

- Let's begin with the simple ones; addition and subtraction.

- The first (+) operator is used to obtain the sum of two or more values, as in the following example.

```
mysql> select 10+5, 100+65
```

10+5	100+65
15	165

1 row is set

The (-) operator used to subtract one value from another

5-1	999-1000	0.24-0.2
4	-1	

1 row is set



O. Principal
New Satara College of B.C.A
Pandharpur.

- You can multiply numbers with the * op which is used to two or more values.

```
mysql> SELECT 7 * 10, 7 * 0, -11.1 * 2.5
```

7 * 10	7 * 0	-11.1 * 2.5
70	0	-27.8

1 row in set

- The / operator is used to divide one value by the other to obtain the quotient.

```
mysql> select 10/2, 12.1/0.01, 22/7
```

10/2	12.1/0.01	22/7
5.00	-1210.0000	3.14

1 row in set (0.00 sec)

```
mysql> select 2
```

- You can also use the % operator to obtain the modulo of a division operation.

```
mysql> SELECT 25%7, -33%7;
```

25%7	-33%7
4	-5

1 row in set

- You can multiply numbers with the * operator, which is used to two or more values.

```
mysql > SELECT 7 * 10, 7 * 0, -11.1 * 2.5
```

7 * 10	7 * 0	-11.1 * 2.5
70	0	-27.8

1 row in set

- The / operator is used to divide one value by the other to obtain the quotient.

```
mysql > select 10/2, 12.1/-0.01, 22/7
```

10/2	12.1/-0.01	22/7
5.00	-1210.0000	3.14

1 row in set (0.00 sec)

```
mysql > select 2
```

- You can also use the % operator to obtain the modulo of a division operation:

```
mysql > SELECT 25%7, -33%7;
```

25%7	-33%7
4	-5

1 row in set



Principal
New Satara College of B.C.A.
Pandharpur.

g) Logical Operators :-

- MySQL also comes with the four logical operators which make it possible to test the logical validity of one or more expressions.
- The result of an operation involving these operators is always 1 (true), 0 (false), or NULL.

operator	What IS Done.
! (NOT)	Logical NOT
&& (AND)	Logical AND
(OR)	Logical OR
XOR	Logical XOR (exclusive OR)

The operator is NOT operator, which reverse the logical sense the test following it, turning true into false and false into true.

```
mysql> SELECT NOT 1, NOT 0
```

NOT 1	NOT 0
0	1

- The AND operator makes it possible to the validity of two or more values (or expressions evaluating to values):

```
mysql> SELECT (2=2) AND (900<100), ('a'='a')
AND ('c'<'d');
```

(2=2) AND (900<100)	('a'='a') AND ('c'<'d')
0	1

- A twist on this is the OR operator, which returns true if any of the values or expressions involved are true and not NULL.

```
mysql > SELECT (2=2) OR (900<100), ('a'='a') OR ('c'<'d')
```

$(2=2) \text{ OR } (900<100)$	$('a'='a') \text{ OR } ('c'<'d')$
1	1

- A twist on this is the OR operator, which returns true if any of the values or expressions involved are true and not NULL.

```
mysql > SELECT (2=2) OR (900<100), ('a'='a') OR ('c'<'d')
```

$(2=2) \text{ OR } (900<100)$	$('a'='a') \text{ OR } ('c'<'d')$
1	1

- It also includes an additional XOR operator, which returns true if either one (but not both) of its arguments is true.

```
mysql > SELECT (1=1) XOR (2=4), (1<2) XOR (9<10)
```

$(1=1) \text{ XOR } (2=4)$	$(1<2) \text{ XOR } (9<10)$
1	0

Principal

New Satara College of B.C.A
Pandharpur.

4] Bit Operator:-

MySQL also includes six operators designed specifically for bit manipulation.

MySQL bit Operators:

Operator	What It Does.
&	Bitwise AND
	Bitwise OR
^	Bitwise XOR
~	Bit inversion
>>	Bitwise right shift
<<	Bitwise left shift

The | operator is used to perform a bitwise OR, while the & operator is used to perform a bitwise AND.

mysql > SELECT 16 | 32, 9 | 4;

16 32	9 4
48	13

mysql > SELECT 30 & 10, 8 & 16;

30 & 10	8 & 16
10	0

mysql > SELECT 1 << 7, 64 >> 1;

1 << 7	64 >> 1
128	32

Consider the following example, which demonstrates how the IS NULL operator can be used to test whether or not a value is null.

```
mysql> SELECT 88 IS NULL, 0 IS NULL, NULL IS NULL
```

88 IS NULL	0 IS NULL	NULL IS NULL
0	0	1

Adding the NOT logical operator reverse that returning those record that do not match the specified search string. The following example

```
mysql> SELECT 'Harry' NOT LIKE '%ry', NOT LIKE 'inbeensect'
```

'Harry' NOT LIKE '%ry'	'been' NOT LIKE 'insect'
0	1

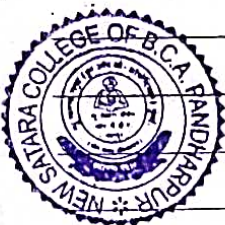
```
mysql> SELECT 'mysql' LIKE '%sql', BINARY 'mysql' LIKE '%sql'
```

'mysql' LIKE '%sql'	BINARY 'mysql' LIKE '%sql'
1	0

Addition of the NOT logical operator to the REGEXP operator reverse the behavior.

```
mysql> SELECT 'hammer' NOT REGEXP 'ham$'?
```

'hammer' NOT REGEXP 'ham\$'?
1



Principal
New Satara College of B.C.A.
Pandharpur.

g) Logical Operators :-

- MySQL also comes with the four logical operators which make it possible to test the logical value of one or more expressions.
- The result of an operation involving these operators is always 1 (true), 0 (false), or NULL.

operator	What IS Done.
! (NOT)	Logical NOT
&& (AND)	Logical AND
(OR)	Logical OR
XOR	Logical XOR (exclusive OR)

The operator is NOT operator, which reverse the logical sense the test following it, turning true into false and false into true.

```
mysql> SELECT NOT 1, NOT 0
```

NOT 1	NOT 0
0	1

- The AND operator makes it possible to the valid of two or more values (or expressions evaluation to values).

```
mysql> SELECT (2=2) AND (900<100), ('a'='a') AND ('c'<'d');
```

(2 = 2) AND (900 < 100)	('a' = 'a') AND ('c' < 'd')
0	1

- The \wedge Operator is used to perform a bitwise XOR

```
mysql > SELECT 1^0, 0^1, 17^9
```

1^0	0^1	17^9
1	1	24

- mysql \sim operator is used to perform a bit inverse and return 64-bit integer result:

```
mysql > SELECT ~18446744073, ~1;
```

~18446744073	~1
1	18446744073

* MySQL functions :- Aggregate, Math, String, Date and Time, Control flow functions and expressions, Type conversion, Formatting Encryption.

1) Aggregate function :-

1) AVG (col) - Return the average of the values in the named column.

```
mysql > SELECT AVG(return) FROM mutualFunds;
```

Count (col) - Returns a count of the number of non NULL records in the named column.

```
mysql > SELECT count(*) FROM members
```



3) MIN (col) :- Returns the minimum value in the named column.

```
mysql> SELECT MIN(quantity) FROM inventory;
```

4) MAX (col) :- Return the maximum value in the named column.

```
mysql> SELECT MAX(return) FROM investments;
```

5) SUM :- Returns the total of the values in the named column.

```
mysql> SELECT SUM(Units) FROM mutualfunds;
```

6) STD (col) :- Return the standard deviation of the values in the named column.

```
mysql> SELECT STD(price) FROM stocks;
```

7) VARIANCE (col) :- Return the statistical variance of the values in the named columns.

```
mysql> SELECT VARINCE (invoiceamt) FROM invoice  
where customerID = 125;
```

8) GROUP_CONCAT (col) :- Returns a concatenated list of column values belonging to a group.

Unit-3 Performing operation on Table Data

■ populating tables with data:-

- To add the data in tables using the INSERT using the INSERT statement

1) Insert :- Insert statement is a DML (Data modification language) statement which is used to insert data in the MySQL table. Using Insert query, we can add one or more rows in the table.

Syntax :- Insert into <table_name> (column_1, column_2, ----- column_n)
Values (value_1, value_2, ... value_n)

example :- Insert into employee (employee_first_name, employee_last_name) values ('Nisarg', 'Bagal')

- Once the query executed successfully, let us run the SELECT query to verify the data. execute the following query.

```
select * from employee;
```

* Retrieving data from tables:-

- The first SQL command you will learn, and the one you will use most frequently, is SELECT. In this lesson, you begin learning how to fetch data records from a table.



- A SELECT statement begins with the SELECT keyword and is used to retrieve information from MySQL database tables.
- You must specify the table name to fetch data from - using the FROM keyword and one or more columns that you want to retrieve from table.

Syntax :- SELECT * FROM table-name;

Example :- SELECT * FROM employee;

* Retrieving Database information :-

- The SHOW command is used to retrieve information about database components.

* Retrieving a List of Databases :-

- Use the SHOW DATABASES command to retrieve a list of databases that you have access to execute the SHOW command just like a SELECT statement from the MySQL program.

```
mysql> SHOW DATABASES;
```

* Retrieving a List of Tables :-

```
mysql> SHOW TABLES
```

* Explain the sorting data in a table:-

- When SQL query returns that output, the values are not sorted. To sort the result of a query, we use the ORDER BY clause.

Syntax :- SELECT <Column 1>, <Column 2>... FROM
<TABLE-NAME> ORDER BY <COLUMN-1>

e.g :- SELECT * FROM customer order by
first_name asc.

IF ASC in the ORDER BY clause, the result will be stored in ascending order, and if you specify DESC, then the result will be sorted in descending order.

Syntax :- SELECT <Column 1>, <Column 2>... FROM
<TABLE-NAME> ORDER BY <COLUMN-1>

e.g :- SELECT * FROM customer order by
first_name desc.

Syntax :- SELECT <Column List> FROM TABLE-NAME WHERE
query: select * from customer where first_name='LINDA' condition

* Deleting data from table:-

- To delete data from a table, you use the MySQL DELETE statement. The following

Syntax :- DELETE FROM table name
WHERE condition;

example :- DELETE FROM employees
Principal
New Satara College of B.C.A
Pandharpur.



- To delete all rows from the employee table, you use the DELETE statement without the WHERE clause as follows:

```
DELETE FROM employees;
```

* Updating data in table:-

- Update MySQL command is used to modify rows in a table. The update command can be used to update a single field or multiple fields at the same time. It can also be used to update a MySQL table with values from another table.

Syntax :- UPDATE 'table-name' SET 'column-name' = 'new value' [WHERE condition];

- We will start with making updates for membership numbers before we make any updates to our data, let's retrieve the record for membership numbers. The script shown below helps us to do that.

example :- SELECT * FROM 'members'
WHERE 'membership_number' = 1;

* searching data from table :-

- This article will introduce a couple of ways for MySQL search data in tables. First, how to search for data in tables will be shown using the classic Where clause and Like operator. Then, MySQL full-text search feature will be introduced and, in the end, how to perform data search will be shown using a third-party extension for VS Code called ApexSQL Database Power Tools for VS Code.

example :-

```
SELECT actor.actor_id
       actor.first_name
       actor.last_name,
       actor.last_update
FROM sakila.actor
WHERE actor.last_name = 'AKROYD';
```

* Adding and Dropping Columns :-

- ALTER table statement to add a column, modify a column, drop a column, rename a column or rename a table.

- ALTER Table statement is used to add, modify or drop/delete columns in table. ~~The MySQL~~ ALTER TABLE statement is also used to rename a table.



* Add a Column in table :-

Syntax :-

```
ALTER TABLE table-name  
ADD TABLE table-name  
ADD new-column-name column_definition  
[ FIRST | AFTER column-name];
```

Example :-

```
ALTER TABLE Contacts  
ADD last_name VARCHAR(40) NOT NULL  
AFTER contact_id;
```

* Add Multiple Columns in table :-

Syntax :-

```
ALTER TABLE table-name  
ADD new-column-name column_definition  
[ FIRST | AFTER column-name],  
ADD new-column column_definition  
[ FIRST | AFTER column-name],
```

Example :-

```
ALTER TABLE contacts.  
ADD last_name VARCHAR(40), NOT NULL  
AFTER contact_id,  
ADD first_name VARCHAR(35) NULL  
AFTER last_name;
```



(+)

* Drop column in table:-

Syntax :- ALTER TABLE table_name
DROP COLUMN column_name;

table_name :- The name of the table of modify
column_name :- The name of the column to
delete from the table.

Example :-

ALTER TABLE Contacts
DROP COLUMN contact-type;

* Modifying and Rename existing Column.

* Modify column in table :-

Syntax :- ALTER TABLE Table_name
MODIFY column_name Column_definition
[FIRST | AFTER column_name];

table_name - The name of the table to modify.
column_name - The name of the column to
modify in the table.

column_definition - The modified datatype &
definition of the column (NULL & NOT NULL etc)

FIRST | AFTER column_name -
optional. It tells MySQL where in the
table to position the column, if you



Principal
New Sakara College of B.C.A.
Pandharpur.

Example:-

```
ALTER TABLE contacts
MODIFY last_name varchar(50) NULL;
```

* Rename Column in table :-

Syntax :-

```
ALTER TABLE table-name
CHANGE COLUMN old_name new_name
column_definition
[FIRST | AFTER column-name];
```

Example:-

```
ALTER TABLE contacts
CHANGE COLUMN contact_type, c_type
varchar(20) NOT NULL;
```

* Renaming table using alter table :-

Syntax :-

```
ALTER TABLE table-name
RENAME TO new-table-name;
```

table-name - The table to rename.

new-table-name - The new table name to use.

Example:-

```
ALTER TABLE contacts
RENAME TO people;
```



* Changing a table type in MySQL.

• MySQL Change column type:-

- Changing the column definition is a very useful feature for the database whenever you want.
 - MySQL allows a command to alter the column definition such as name and type according to our needs. We can do this with the help of an ALTER TABLE statement in MySQL.
- In this article, we are going to see step by step how to change by column definition by using ALTER TABLE Command.

Syntax :-

① ALTER TABLE table-name
MODIFY column-name data-type;

Syntax :-

~~Example~~ ② ALTER TABLE table-name
MODIFY COLUMN column-name1 data-type
MODIFY COLUMN column-name2 data-type

Example :- ① ALTER TABLE employees MODIFY column
emp_id varchar(10);

② ALTER TABLE employees ~~MODIFY~~
COLUMN emp_id int(5),
MODIFY COLUMN income ~~varchar(20)~~



* Finding out table created by user:-

* The show or list table is very important when we have many databases that contain various tables. Sometimes the table names are the same in many databases; in that case, in that case, this query is very useful. We can get the number of a data database using the following statement.

```
mysql> SHOW TABLES;
```

Specific database or table:-

syntax:-

```
mysql> SHOW TABLES FROM database_name;  
SHOW TABLES FROM FROM IN database_name;
```

Example:-

```
mysql> SHOW TABLES FROM mystudentdb;
```

OR

```
mysql> SHOW TABLES IN mystudentdb;
```

* How to Displaying a table structure:-

- In mysql describe the table structure in data base Table.

- DESCRIBE means to show the information in detail, since we have tables in MySQL. So we will use the DESCRIBE command to

* MySQL Constraints :-

- The constraint in MySQL is used to specify the rule that allows or restricts what values / data will be stored in the table.
- They provide a suitable method to ensure data accuracy and integrity inside the table.
- It also helps to limit the type of data that will be inserted inside the table. If any interruption occurs between the constraint and data action, the action is failed.

* Types of MySQL Constraints.

- Constraints in MySQL is classified into two types

1. Column Level Constraints :-

These constraints are applied only to the single column that limits the type of particular column data.

2. Table Level Constraints :-

These constraints are applied to the entire table that limits the type of data for the whole table.



* How to create constraints in MySQL:

- We can define the constraints during a table created by using the CREATE TABLE statement.
- MySQL also uses the ALTER TABLE statement to specify the constraints in the case of the existing table schema.

- Syntax :-

```
CREATE TABLE new_table_name (  
    col_name1 datatype constraint,  
    col_name2 datatype constraint,  
    col_name3 datatype constraint,  
    .....  
);
```

* Types of Data Constraints:-

• I/O Constraints :-

1) NOT NULL :-

- This constraint specifies that the column cannot have NULL or empty values. The below statement creates a table with NOT NULL constraint.
- By default, a column can hold NULL values.
- The NOT NULL constraint enforces a column

to NOT accept NULL Values.

This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

Example :-

```
CREATE TABLE Persons(
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255) NOT NULL,
    Age int
);
```

2] UNIQUE Constraint :-

- The UNIQUE constraint ensure that all values in a column are different.

- Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

- A PRIMARY key constraint automatically has a UNIQUE constraint.

- However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.



example :-

```
CREATE TABLE Persons(
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  UNIQUE (ID)
);
```

③ PRIMARY KEY Constraint :-

- The PRIMARY KEY constraint uniquely identifies each record in a table.
- Primary keys must contain UNIQUE values, and cannot contain NULL values.
- A table can have only ONE primary key, and in the table, this primary key can consist of single or multiple columns (fields)

- PRIMARY KEY ON CREATE TABLE :-

The following SQL creates a PRIMARY KEY on the "ID" column when the "Persons" table is created.

```
CREATE TABLE Persons(
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  PRIMARY KEY (ID)
);
```

- To allow naming of a PRIMARY KEY constraint, and for defining a PRIMARY KEY constraint on multiple columns, use the following SQL Syntax :-

```
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  CONSTRAINT PK_Person PRIMARY KEY
  (ID, LastName)
);
```

NOTE :- In the example above there is only ONE PRIMARY KEY (PK_Person). However, the VALUE of the primary key is made up of two COLUMNS (ID + LastName).

4] MySQL FOREIGN KEY Constraint.

- The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.
- A FOREIGN KEY is a field (or collection of fields) on one table, that refers to the PRIMARY KEY in another table.

- The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent tables.



Look at the following two tables:

Persons Table :-

PersonID	LastName	FirstName	Age
1	Hansen	Ola	30
2	Svendson	Tove	23
3	Pettersen	Kari	20

Orders Table :-

OrderID	OrderNumber	PersonID
1	77895	3
2	44678	3
3	22456	2
4	24562	1

- Notice that the "PersonID" Column in the "Orders" table points to the "PersonID" Column in the "Persons" table.

- The "PersonID" Column in the "Persons" table is the PRIMARY KEY in the "Persons" table.

- The "PersonID" Column in the "Orders" table is a FOREIGN KEY in the "Orders" table.

- The FOREIGN KEY constraint prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the parent table.

```

CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID)
    REFERENCES Persons (PersonID)
);

```

To allow naming of a FOREIGN KEY constraint and for defining a FOREIGN KEY constraint on multiple columns, use the following SQL Syntax:-

```

CREATE TABLE Order (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    CONSTRAINT FK_PersonOrder
    FOREIGN KEY (PersonID)
    REFERENCES Person (PersonID)
);

```

5] MySQL Composite key:-

- A composite key in MySQL is a combination of two or more columns in a table that allows us to identify each row of the table uniquely.
- It is a type of candidate key which is formed by more than one column.



Principal

New Satara College of B.C.A.
Pandharpur.

MySQL guaranteed the uniqueness of the Column only when they are Combined. If they have taken individually, the uniqueness cannot maintain.

- Any key such as primary key, super key, or candidate key can be called composite key when they have combined with more than one attribute.

- A Composite key is useful when the table needs to identify each record with more than one attribute uniquely.

- A Column used in the composite key can have different data types. Thus, it is not required to be the same data type for the Columns to make a composite key in MySQL.

- A Composite key can be added in two ways:

1. USING CREATE Statement.

2. USING ALTER Statement.

1] Composite key using CREATE Statement.

Here, we are going to understand how Composite key works in MySQL. Let us first create a table "Product", using the following statement.

```
CREATE TABLE Student(
  Stud-id int NOT NULL,
  Stud-Code Varchar(15),
  Stud-name Varchar(35),
  Subject Varchar(25),
  marks int
);
```

Now, execute the ALTER TABLE statement to add a composite primary key as follows.

```
ALTER TABLE Student add primary key (Stud-id, Subject)
```

2] Business Rule Constraints - check,

* Check constraint :-

- The CHECK constraint is used to limit the value range that can be placed in a column.
- If you define a CHECK constraint on a column it will allow only certain values for this column.
- If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

* CHECK on CREATE TABLE



The following SQL creates a CHECK constraint on the "Age" column when the "Persons" table is created. The CHECK constraint ensures that the age of a person must be 18 or older.

Principal,
New Satara College of B.C.A.
Bardhaman.

```

CREATE TABLE Product (
    Prod_ID int NOT NULL,
    Name Varchar (45),
    Manufacture Varchar (45),
    PRIMARY KEY (Name, Manufacturer)
);

```

- In the above statement, we have created a composite primary with the column names Name and Manufacturer.

Next, we need to insert the values into this table as given below:

```

INSERT INTO Product (Product_ID, Name, Manufacture)
VALUES (101, 'Soap', 'Hamam'),
(102, 'Shampoo', 'Teresme'),
(103, 'Oil', 'Dabur Almond');

```

2] Composite key using ALTER TABLE statement

- ALTER statement always used to do the modification into the existing table. Sometimes it is required to add the composite key to uniquely identify each record of the table with more than one attribute. In that case, we use an ALTER TABLE statement.

- Let us first create a table "student" using the below statement

• sub Queries, Union and Indexing.

* Sub queries - use: example.

~~subquery~~

- A subquery in MySQL is a query, which is nested into another SQL query and embedded with SELECT, INSERT, UPDATE or DELETE statement along with the various operators. We can also nest the subquery with another subquery.
- A subquery is known as the inner query, and the query that contains subquery is known as the outer query.
- The inner query executed first gives the result to the outer query, and then the main/outer query will be performed. MySQL allows us to use subquery anywhere, but it must be closed within parenthesis.
- All subquery forms and operations supported by the SQL standard will be supported in MySQL also.

* The following are the rules to use subqueries:

- Subqueries should always use in parentheses
- If the main query does not have multiple columns for subquery, then a subquery can have only one column in the SELECT command.
- We can use various comparison operators with the subquery such as $>$, $<$, $=$, IN, ANY, SOME and ALL. A multiple row subquery is very useful when the subquery returns more than one row.



one row

- We cannot use the ORDER BY clause in a subquery although it can be used inside the main query.
- If we use a subquery in a set function, it cannot be immediately enclosed in a set function.

* Advantages of subqueries:-

- The subqueries make the queries in a structured form that allows us to isolate each part of a statement.
- The subqueries provide alternative ways to query the data from the table; otherwise, we need to use complex joins and unions.
- There are subqueries are more readable than complex join or union statements.

* The subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause.

- The subquery can be used with return data that will be used in the main query as a condition to further restrict the data to be retrieved.

SELECT statement with subqueries:-

- Subqueries are most frequently used with the SELECT statement.

Syntax :-
 SELECT column_name [Column-name]
 FROM table1 [table2]
 WHERE column_name OPERATOR
 (SELECT column_name [, column_name]
 FROM table1 [table2]
 (WHERE))

Customer table :-

ID	Name	AGE	ADDRESS	SALARY
1	Ramesh	35	Ahmedabad	2000.00
2	khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Now, let us check the following subquery with a SELECT statement.

```
SQL > SELECT * FROM CUSTOMERS
WHERE ID IN (SELECT ID FROM CUSTOMERS
WHERE SALARY > 4500);
```

* subqueries with the INSERT statement.

- Subqueries also can be used with INSERT statements. The INSERT statement uses the data returned from the subquery to insert into another table.
- The selected data in the subquery can be modified with any of the character and number functions.



Syntax :-

```
INSERT INTO table_name [(Column 1
Column 2)]
SELECT [* | Column 1 | Column 2]
FROM table 1 [, table 2]
[WHERE VALUE OPERATOR]
```

Example :-

```
INSERT INTO Customers_BKP
SELECT * FROM CUSTOMERS
WHERE ID IN (SELECT ID
FROM CUSTOMERS)
```

Subqueries with the UPDATE statement.

- The subquery can be used in conjunction with the UPDATE statement. Either single or multiple columns in a table can be updated when using a subquery with the UPDATE statement.

Syntax :-

```
UPDATE table_name
SET Column_name = new_value
[WHERE OPERATOR [VALUE]
(SELECT COLUMN_NAME
FROM TABLE-NAME)
[WHERE]]
```

Example :-

```
UPDATE CUSTOMERS
SET SALARY = SALARY * 0.25
WHERE AGE IN (SELECT AGE FROM
CUSTOMERS_BKP WHERE AGE >= 27);
```

This would impact two rows and finally CUSTOMER table would have the following records.

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	35	Ahmedabad	125.00
2	khilan	25	Delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	2125.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

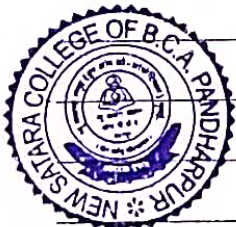
* Subqueries with the DELETE statement

- The subquery can be used in conjunction with the DELETE statement like with any other statements mentioned above.

Syntax :-

```
DELETE FROM TABLE_NAME
[WHERE ORDER OPERATOR [VALUE]
(SELECT COLUMN_NAME
FROM TABLE_NAME)
[WHERE]]
```

example :- DELETE FROM CUSTOMERS
WHERE AGE IN (SELECT AGE FROM
CUSTOMERS-BKP WHERE AGE >= 27);



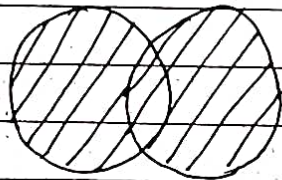
SET Operations in MySQL :-

- MySQL supports few set operations which can be performed on the table data. These are used to get meaningful results from data stored in the table, under special conditions.
- In this tutorial, we will cover 4 different types SET operations, along with example:

1. UNION
2. UNION ALL
3. INTERSECT
4. MINUS

1] UNION Operation :-

- UNION is used to combine the results of two or more SELECT statements. However it will eliminate duplicate rows from its resultset. In case of union, number of columns and datatype same in both the tables, on which UNION operation is being applied.



1) first table

2) second table

ID	Name
1	abhi
2	adam

ID	Name
2	adam
3	chester

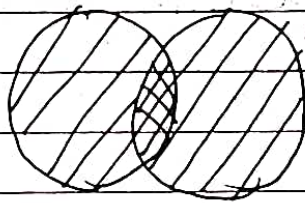
Query :- `SELECT * FROM First`
`UNION`
`SELECT * FROM Second;`

The result table will look like,

ID	NAME
1	abhi
2	adam
3	Chester

2] UNION ALL :-

- This operation is similar to Union.
 But it also shows the duplicate rows.



Example of Union All

① The first Table ② The second table.

ID	NAME	ID	NAME
1	abhi	2	adam
2	adam	3	chester

Query :- `SELECT * FROM First`
`UNION ALL`
`SELECT * FROM Second`



Principal
 New Satara College of B.C.A.
 Pandharpur.

The resultset table will look like

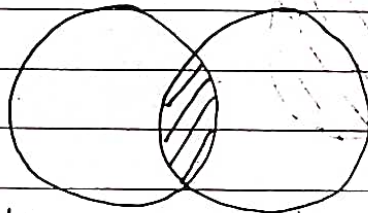
ID	NAME
1	abhi
2	adam
3	Chester

3) INTERSECT :-

Intersect operation is used to combine two SELECT statements, but it only returns the records which are common from both SELECT statements. In case of Intersect the number of columns and datatype must be same.

MySQL does not support INTERSECT operator

Example



① The first table

ID	NAME
1	abhi
2	adam

② The second table,

ID	NAME
2	adam
3	Chester

Query

SELECT * FROM First

INTERSECT

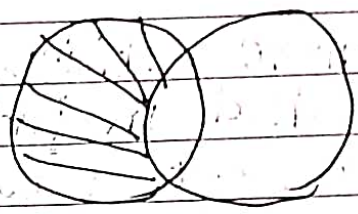
SELECT * FROM Second;

Result :-

ID	NAME
2	adam

4] MINUS :-

The Minus operation combines results of two SELECT statements and return only those in the final result, which belongs to the first set of the result.



* The first table

ID	NAME
1	abhi
2	adam

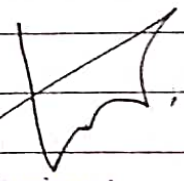
* The second table

ID	NAME
2	adam
3	chester

Query :-
SELECT * FROM First
MINUS
SELECT * FROM Second;

ID	NAME
1	abhi




Principal
New Satara College of B.C.A
Pandharpur.

* Indexing :-

- Indexing is a powerful structure in MySQL which can be leveraged to get the fastest response times from common queries.
- MySQL queries achieve efficiency by generating a smaller table, called an index, from a specified column or set of columns. These columns, called a key, can be used to enforce uniqueness.
- A database index is a data structure that improves the speed of operations in a table. Indexes can be created using one or more columns, providing the basis for both rapid random lookups and efficient ordering of access records.
- While creating index, it should be taken into consideration which all columns will be used to make SQL queries and create one or more indexes on those columns.

* (Creating Index) :- Simple, Composite and Unique.

1] Simple Index :-

- An index is a data structure that allows us to add indexes in the existing table.
- It enables you to improve the faster retrieval of records on a database table.
 - It creates an entry for each value of the indexed columns. We use it to quickly find the record without

■ Stored Procedure:-

MySQL Stored Procedure. A procedure (often called a stored procedure) is a collection of pre-compiled SQL statements stored inside the database.

- It is a subroutine or a subprogram in the regular computing language. A procedure always contains a name, parameter lists, and SQL statements.
- If we consider the enterprise application, we always need to perform specific tasks such as database cleanup, processing payroll, and many more on the database regularly. Such tasks involve multiple SQL statements for executing each task. This process might be easy if we group these tasks into a single task. We can fulfill this requirement in MySQL by creating a stored procedure in our database.
- A procedure is called recursive stored procedure.

* Stored Procedure Features:-

- A stored procedure increases the performance of the applications. Once stored procedures are created they are compiled and stored in the database.
- Stored procedure reduces the specific between application and database server. Because the application has to send only the stored procedure's name and parameters instead of sending multiple SQL statements.



- Stored procedures are reusable and transparent to any applications.
- A procedure is always secure. The database administrator can grant permission to application that access stored procedures in the database without giving any permissions on the database table.

* How to create a procedure?

- The following syntax is used for creating a stored procedure in MySQL. It can be return one or more value through parameters or sometime may not return at all. By default a procedure is associated with our current database. But we can also create it into another database from the current database by specifying the name as database name . procedure name.

* Syntax :-

```

DELIMITER $$
CREATE PROCEDURE procedure_name [[IN|OUT|
INOUT] parameter_name datatype [, param
datatype] $$
BEGIN
    Declaration-section
    Executable-section
END $$
DELIMITER ;

```

MySQL procedure parameters has one of three modes:

IN Parameters:-

It is the default mode. It takes a parameter as input, such as an attribute. When we define it, the calling program has to pass an argument to the stored procedure. This parameter value is always protected.

OUT parameters:-

It is used to pass a parameter as output. Its value can be changed inside the stored procedure, and the changed (new) value is passed back to the calling program. It is noted that a procedure cannot access the OUT parameter's initial value when it starts.

INOUT Parameters:-

It is a combination of IN and OUT parameters. It means the calling program can pass the argument, and the procedure can modify the INOUT parameter and then passes the new value back to the calling program.

* How to call a stored procedure?

We can use the CALL statement to call a stored procedure. This statement returns the values to its caller through its parameters (IN, OUT or INOUT).

Syntax:-

CALL procedure_name (parameter(s))



Example:-

```
mysql > USE database-name;
```

- Suppose this database has a table named student-info that contains the following data:

procedure without parameter:-

```
DELIMITER $$
```

```
CREATE PROCEDURE get_merit(student())
```

```
BEGIN
```

```
SELECT * FROM student_info WHERE mark > 70;
```

```
SELECT COUNT(stud-code) AS total-student
```

```
FROM student_info;
```

```
END $$
```

```
DELIMITER;
```

```
mysql > CALL get_merit_student();
```

* Procedures with IN Parameter:-

In this procedure we have used the IN parameter as var1 of integer type to accept a number from users. Its body part fetches the records from the table using a SELECT statement and returns only those rows that will be supplied by the user.

- It also returns the total number of row of the specified table. See the procedure code:

A

DELIMITER; &&

```
CREATE PROCEDURE get_student (IN var1 INT)
```

```
BEGIN
```

```
    SELECT * FROM student info LIMIT var1;
```

```
    SELECT COUNT (stud=code) AS Total-student  
    FROM END &&
```

DELIMITER;

AFTER successful execution, we can call the procedure as follows:

```
mysql > CALL get_student (4);
```

* procedure with OUT Parameters:

- In this procedure, we have used to OUT parameter as the 'highestmark' of integer type. Its body part ~~for~~ fetches the maximum ~~code~~ marks for the table using a MAX() function. see the procedure code.

DELIMITER; &&

```
CREATE PROCEDURE display_max_mark (OUT high  
- mark INT)
```

```
BEGIN
```

```
    SELECT MAX (marks) INTO highestmark  
    FROM student info;
```

```
END &&
```

DELIMITER;



Principal
New Satara College of B.C.A
Pandharpur.

After successful execution, we can call the procedure as follow:

```
mysql> SET @M = '8';
mysql> CALL display_marks (@M);
mysql> SELECT @M;
```

- This procedure's parameter will get the highest marks from the student-info table. When we call the procedure, the OUT parameter tells the database systems that its value goes out for the procedure. Now, we will pass its value to a session variable @M in the CALL statement as follows:

```
mysql> CALL display_max_mark (@M);
mysql> SELECT @M;
```

* Procedures with INOUT Parameter :-

- In this procedure, we have used the INOUT parameter as 'var1' of integer type. Its body part first fetches the marks from the table with the specified id and then stores it into the same variable var1. The var1 first acts as the IN parameter and then OUT parameter. Therefore, we can call it the INOUT parameter mode. see the procedure code.

DELIMITER &&

```
CREATE PROCEDURE display_marks(INOUT var1 INT)
```

```
BEGIN
```

```
    SELECT marks INTO var1 FROM student_info
```

```
    WHERE stud_id = var1;
```

```
END &&
```

```
DELIMITER;
```

* After successful execution, we can call the procedure as follows:

```
mysql > SET @M = '3';
```

```
mysql > CALL display_marks(@M);
```

```
mysql > SELECT @M;
```

* How to show or list stored procedures in MySQL?

- When we have several procedures in the MySQL server, it is very important to list all procedures. It is because sometimes the procedure names are the same in many databases. In that case, this query is very useful. We can list all procedure stored on the current MySQL server as follows.

```
x) SHOW PROCEDURE STATUS [LIKE 'pattern']  
    WHERE search_condition;
```

This statement displays all stored procedure name, including their characteristics



If we want to display procedures in a particular database, we need to use the WHERE clause. In case we want to display procedures in a particular database, we ~~don't~~ need to use the WHERE clause. In case we want to list stored procedures with a specific word, we need to use the LIKE clause.

- We can list all stored procedure in the MySQL mystudentdb database using the below statement.

```
mysql) SHOW PROCEDURE STATUS WHERE db = 'mystudentdb';
```

- It will give the below output where we can see that the mystudentdb database contains four stored procedures

* How to delete / drop stored procedures in MySQL

- MySQL also allows a command to drop the procedure is dropped, it is removed from the database server also. The following statement is used to drop a stored procedure in MySQL:

```
DROP PROCEDURE [IF EXISTS] Procedure_name;
```

- Suppose we want to remove the procedure named display_marks from the mystudentdb database. we can do this by first selecting the database using the SHOW PROCEDURE STATUS command.

* How to alter the procedure in mysql?

→ MySQL does not allow any command to alter the procedure in mysql. However, it provides a command may alter more than one change in the procedure but does not modify the stored procedure's parameters or body. If we want to make such changes, we must drop and re-create the procedure using the DROP PROCEDURE and CREATE PROCEDURE statement.

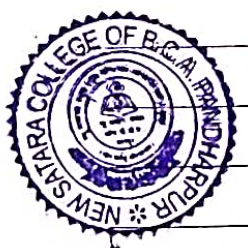
* ALTER PROCEDURE procedure_name [characteristics

Characteristics: {

- COMMENT 'string'
 - LANGUAGE SQL
 - { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }
 - SQL SECURITY { DEFINER | INVOKER }
- }

- Suppose we want to add a commit to the existing procedure. In such a case, we can use the ALTER statement as follows to accomplish this task.

```
mysql> ALTER PROCEDURE get_merit_student
COMMENT 'It display all records'
```



Principal
New Satara College of B.C.A
Pandharpur.

- After executing this statement, we can verify it by using the below statement:

```
mysql> SHOW CREATE PROCEDURE get_merit_student;
```

- It will display the below output where we can see that the comment is added successfully.

* Transaction :-

A transaction in MySQL is a sequential group of statements, queries, or operations such as select, insert, update or delete to perform as a one single work unit that can be committed or rolled back.

- If the transaction makes multiple modifications into the database, two things happen:

- Either all modification is successful when the transaction is committed.
- Or, all modifications are undone when the transaction is roll back.

- In other words, a transaction cannot be successful without completing each operation available in ~~set~~ the set. It means if any statement fails, the transaction operation cannot produce results.

- A transaction in MySQL start with the first executable SQL statement and ends when it finds a commit or rolled back either explicitly or implicitly. It explicitly uses COMMIT or ROLLBACK statement and implicitly.