Periyakaruppan N
12-July-2022

Basic SQL Commands:

DDL , DML
Basic Operations using various DML.
Limiting Results
Sorting Results
Skipping Results
Case statements
Aggregation
Constraints
Index and its uses.
Alias
GROUP BY and Having Clause

DDL: Data Definition Language:

CREATE, ALTER, DROP, TRUNCATE

CREATE Table:

MYSQL> create database bank;

MYSQL> use bank;

MYSQL> create table ACCOUNTS (acc_number int NOT NULL, cust_name varchar(25), age int, acc_balance int, acc_type varchar(20), branch varchar(20), acc_startdate date);

0 row(s) affected

0.157 sec

ALTER Table:

MYSQL> alter table ACCOUNTS ADD email varchar(50);

0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

MYSQL> alter table ACCOUNTS rename to ACCOUNT_DETAILS;

0 row(s) affected

TRUNCATE:

Before Truncate:

acc_number	cust_name	age	acc_balance	acc_type	branch	acc_startdate	email
1001	HARI	22	100000	SAVINGS	TRICHY	2022-08-04	hari@gmail.com
1002	SURESH	25	1000000	CURRENT	CHENNAI	2022-10-14	suresh@gmail.com
1003	BANUPRIYA	30	600000	SAVINGS	SALEM	2022-03-10	banu@gmail.com

MYSQL> truncate ACCOUNTS;

After Truncate:

acc_number	cust_name	age	acc_balance	acc_type	branch	acc_startdate	email

DROP:

MYSQL> drop table ACCOUNTS;

The table rows as well as the schema of the table is totally dropped and the table is deleted.

MYSQL> select * from ACCOUNTS;

Error Code: 1146. Table 'bank.accounts' doesn't exist

DML: Data Manipulation Language

INSERT, UPDATE, DELETE

INSERT into Table:

MYSQL> insert into ACCOUNTS values (1001, 'HARI', 22, 100000, 'SAVINGS', 'TRICHY', '2022-08-04','hari@gmail.com');

acc_number	cust_name	age	acc_balance	acc_type	branch	acc_startdate	email
1000	PERI	20	150000	SAVINGS	TRICHY	2022-02-14	peri@gmail.com
1001	HARI	22	100000	SAVINGS	TRICHY	2022-08-04	hari@gmail.com

UPDATE Row:

MYSQL> update ACCOUNTS set acc_balance = acc_balance + 100000 where acc_number = 1000;

acc_number	cust_name	age	acc_balance	acc_type	branch	acc_startdate	email
1000	PERI	20	250000	SAVINGS	TRICHY	2022-02-14	peri@gmail.com

DELETE Row:

MYSQL> DELETE from ACCOUNTS where cust_name = 'Peri';

Row Deleted

1 row(s) affected

acc_number	cust_name	age	acc_balance	acc_type	branch	acc_startdate	email
1001	HARI	22	100000	SAVINGS	TRICHY	2022-08-04	hari@gmail.com

Basic Operations using various DML:

INSERT Commands:

MYSQL> insert into ACCOUNTS values (1000, 'PERI', 20, 150000, 'SAVINGS', 'TRICHY', '2022-02-14','peri@gmail.com')

MYSQL> insert into ACCOUNTS values (1001, 'HARI', 22, 100000, 'SAVINGS', 'TRICHY', '2022-08-04','hari@gmail.com')

MYSQL> insert into ACCOUNTS values (1002, 'SURESH', 25, 1000000, 'CURRENT', 'CHENNAI', '2022-10-14','suresh@gmail.com')

MYSQL> insert into ACCOUNTS values (1003, 'BANUPRIYA', 30, 600000, 'SAVINGS', 'SALEM', '2022-03-10','banu@gmail.com');

SELECT Statement:

MYSQL> select * from ACCOUNTS;

acc_number	cust_name	age	acc_balance	acc_type	branch	acc_startdate	email
1000	PERI	20	150000	SAVINGS	TRICHY	2022-02-14	peri@gmail.com
1001	HARI	22	100000	SAVINGS	TRICHY	2022-08-04	hari@gmail.com
1002	SURESH	25	1000000	CURRENT	CHENNAI	2022-10-14	suresh@gmail.com
1002	BANUPRIYA	30	600000	SAVINGS	SALEM	2022-03-10	banu@gmail.com

MYSQL> select cust_name, acc_balance, age from accounts where age >= 25;

cust_name	acc_balance	age
SURESH	1000000	25
BANUPRIYA	600000	30

LIMITING RESULTS:

Limits and fetches only top 3 rows. All returns the same output.

MYSQL> SELECT * FROM ACCOUNTS LIMIT 3;

MYSQL> SELECT * FROM ACCOUNTS FETCH FIRST 3 ROWS ONLY;

MYSQL> SELECT TOP 3 * FROM ACCOUNTS;

acc_number	cust_name	age	acc_balance	acc_type	branch	acc_startdate	email
1000	PERI	20	150000	SAVINGS	TRICHY	2022-02-14	peri@gmail.com
1001	HARI	22	100000	SAVINGS	TRICHY	2022-08-04	hari@gmail.com
1002	SURESH	25	1000000	CURRENT	CHENNAI	2022-10-14	suresh@gmail.com

SORTING Results:

ORDERBY:

We sort the rows with specific row using ORDERBY Clause.

MYSQL> SELECT * FROM ACCOUNTS ORDER BY cust_name;

acc_number	cust_name	age	acc_balance	acc_type	branch	acc_startdate	email
1002	BANUPRIYA	30	600000	SAVINGS	SALEM	2022-03-10	banu@gmail.com
1001	HARI	22	100000	SAVINGS	TRICHY	2022-08-04	hari@gmail.com
1000	PERI	20	150000	SAVINGS	TRICHY	2022-02-14	peri@gmail.com
1002	SURESH	25	1000000	CURRENT	CHENNAI	2022-10-14	suresh@gmail.com

MYSQL> SELECT * FROM ACCOUNTS ORDER BY acc_startdate DESC;

Order by in descending order.

acc_number	cust_name	age	acc_balance	acc_type	branch	acc_startdate	email
1002	SURESH	25	1000000	CURRENT	CHENNAI	2022-10-14	suresh@gmail.com
1001	HARI	22	100000	SAVINGS	TRICHY	2022-08-04	hari@gmail.com
1002	BANUPRIYA	30	600000	SAVINGS	SALEM	2022-03-10	banu@gmail.com
1000	PERI	20	150000	SAVINGS	TRICHY	2022-02-14	peri@gmail.com

SKIPPING RESULTS:

MYSQL> SELECT * FROM ACCOUNTS LIMIT 2, 1; -- offset, limit

This SQL query Skips first 2 rows and fetches one row after that. (i.e, 3rd Row)

acc_number	cust_name	age	acc_balance	acc_type	branch	acc_startdate	email
1002	SURESH	25	1000000	CURRENT	CHENNAI	2022-10-14	suresh@gmail.com

CASE Statements:

Case statement is similar to an If-Else Statement

MYSQL> SELECT acc_number, cust_name,

CASE

WHEN acc_balance >= 200000 THEN 'You can avail Platinum Card'

WHEN acc_balance >= 100000 and acc_balance < 200000 THEN 'You can avail Gold Card'

ELSE 'You can avail Silver Card'

END AS Card_Details

FROM ACCOUNTS;

acc_number	cust_name	Card_Details
1000	PERI	You can avail Gold Card
1001	HARI	You can avail Gold Card
1002	SURESH	You can avail Platinum Card
1003	BANUPRIYA	You can avail Silver Card

AGGREGATION:

COUNT:

MYSQL> select count(*) from ACCOUNTS;

count(*)
4

AVERAGE:

MYSQL> select avg(age) from ACCOUNTS;

avg(age)
24.2500

SUM:

MYSQL> select SUM(acc_balance) from ACCOUNTS;

SUM(acc_balance)
1310000

CONSTRAINTS:

MYSQL> ALTER TABLE ACCOUNTS

MODIFY acc_number int NOT NULL UNIQUE;

MYSQL> ALTER TABLE ACCOUNTS

MODIFY acc_number int PRIMARY KEY;

INDEXES and its USES:

It is a schema object used by Oracle server to speed up the retrieval of rows by using a pointer. It is independent of the table it indexes. It can reduce disk I/O by using a rapid path access method to locate data quickly.

MYSQL> create index index_bank on ACCOUNTS (cust_name,branch);

Index is created.

MYSQL> show index from ACCOUNTS;

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
accounts	1	index_bank	1	cust_name	A	4	NULL	NULL	YES	BTREE			YES	NULL
accounts	1	index_bank	2	branch	A	4	NULL	NULL	YES	BTREE			YES	NULL

MYSQL> alter table ACCOUNTS drop index index_bank;

Index is dropped.

ALIAS Statements:

MYSQL> select acc_number AS 'ACCOUNT ID', cust_name AS 'CUSTOMER' from ACCOUNTS;

Gives an alias name for the column to display.

ACCOUNT ID	CUSTOMER
1000	PERI
1001	HARI
1002	SURESH
1003	BANUPRIYA

GROUP BY and HAVING Clause:

MYSQL> select acc_type,COUNT(acc_type) from ACCOUNTS GROUP BY acc_type;

acc_type	COUNT(acc_type)
SAVINGS	3
CURRENT	1

(With having clause):

MYSQL> select acc_type,COUNT(acc_type) from ACCOUNTS GROUP BY acc_type HAVING count(acc_type) > 1 ;

acc_type	COUNT(acc_type)
SAVINGS	3