|  |  |
| --- | --- |
| **DBMS\_MYSQL Commands** | |
| Create database and use it | create database if not exists chb\_practice  use chb\_practice |
| Create table | create table if not exists bank\_details(  age int,  job varchar(30),  marital varchar(30),  education varchar(30)); |
| Select query | select \* from bank\_details |
| Insert query | insert into bank\_details values(58,"management","married") |
| Total record | Select count(\*) from bank\_details |
| Specific column | select age,job,loan from bank\_details |
| Limited record | select \* from bank\_details limit 10 |
| and | select \* from bank\_details where age=60 and job='retired' |
| Distinct record | select distinct job from bank\_details |
| Order by | select \* from bank\_details order by age  select \* from bank\_details order by age desc |
| Aggregate function | select sum(balance) from bank\_details  select avg(balance) from bank\_details  select max(balance) from bank\_details |
| subquery | select \* from bank\_details where balance in (select min(balance) from bank\_details) |
| Dump data into database | LOAD DATA INFILE  'E:/AttributeDataSet.csv'  into table dress  FIELDS TERMINATED by ','  ENCLOSED by '"'  lines terminated by '\n'  IGNORE 1 ROWS; |
| Constraint1 | create table if not exists test1(  test\_id int auto\_increment,  test\_name varchar(30),  test\_mailid varchar(30),  test\_address varchar(30),  primary key(test\_id)) |
| Constraint1 | create table if not exists test4(  test\_id int not null auto\_increment,  test\_name varchar(30) not null default 'unknown',  test\_mailid varchar(30) unique not null,  test\_address varchar(30) check (test\_address='kalol') not null,  test\_salary int check (test\_salary>5000) not null,  primary key(test\_id)) |
| Constraint1 | create table if not exists test12(  test\_id int NOT NULL default 0,  test\_name varchar(30) ,  test\_mailid varchar(30) unique,  test\_adress varchar(30) check (test\_adress= 'bengalore'),  test\_salary int check(test\_salary > 10000)) |
| Insert Constraint | insert into test12(test\_name,test\_mailid,test\_adress,test\_salary) values('chb','rt@gmail.com','bengalore',15000) |
| view | create view first as select age,job,education,balance from bank\_details;  select \* from first where job='admin.'; |
| Safe mode error | SET SESSION sql\_mode = ''  SET SQL\_SAFE\_UPDATES = 0; |
| Date change | select str\_to\_date(order\_date,'%m/%d/%y') from sales1 |
| Alter query | alter table sales1 add column order\_date\_new date after order\_date |
| Update.alter record | update sales1 set order\_date\_new=str\_to\_date(order\_date,'%m/%d/%Y') |
|  | alter table sales1 add column ship\_date\_new date after ship\_date |
|  | select \* from sales1 where ship\_date\_new = '2011-01-05' |
|  | select \* from sales1 where ship\_date\_new between '2011-01-05' and '2011-08-30' |
| Time and date | select now()  select curdate()  select curtime() |
|  | select \* from sales1 where ship\_date\_new < date\_sub(now(),interval 1 week) |
|  | select date\_sub(now(),interval 30 month) |
|  | alter table sales1 add column flag date after order\_id  update sales1 set flag=now() |
|  | alter table sales1 add column year\_new int  alter table sales1 add column month\_new int  alter table sales1 add column day\_new int; |
|  | update sales1 set year\_new=year(order\_date\_new);  update sales1 set month\_new=month(order\_date\_new);  update sales1 set day\_new=day(order\_date\_new); |
|  | alter table sales1 modify column Year\_new int  alter table sales1 modify column Month\_new int  alter table sales1 modify column Day\_new int; |
|  | select month(order\_date\_new) from sales1 |
|  | select Year\_new, avg(sales) from sales1 group by Year\_new  select Year\_new, sum(sales) from sales1 group by Year\_new  select Year\_new, min(sales) from sales1 group by Year\_new  select Year\_new, max(sales) from sales1 group by Year\_new  select Year\_new, sum(quantity) from sales1 group by Year\_new |
|  | select order\_id, discount, if(discount > 0, 'YES', 'NO') as discount\_flag from sales1 |
|  | alter table sales1 add column discount\_flag varchar (20) after discount  update sales1 set discount\_flag=if(discount>0, 'YES','NO') |
|  |  |
|  | use sales2  select \* from sales1 |
| delimiter $$  create function add\_to\_new(a int)  returns int  deterministic  begin  declare b int;  set b=a+10;  return b;    end $$  select add\_to\_new(25)  select quantity, add\_to\_new(quantity) from sales1 | |
| DELIMITER $$  create function final\_profits\_real(profit decimal(20,6) , discount decimal(20,6) , sales decimal(20,6) )  returns int  Deterministic  Begin  Declare final\_profit int ;  set final\_profit = profit - sales \* discount ;  return final\_profit;  end $$    select profit, discount ,sales , final\_profits\_real(profit, discount,sales) from sales1 ; | |
| DELIMITER &&  create function mark\_sales3(sales int )  returns varchar(30)  DETERMINISTIC  begin  declare flag\_sales varchar(30);  if sales<=100 then  set flag\_sales="super affordable product";  elseif sales > 100 and sales < 300 then  set flag\_sales="affordable";  elseif sales > 300 and sales < 600 then  set flag\_sales="moderate price";  else  set flag\_sales="expensive";    end if;  return flag\_sales;  end &&    select mark\_sales3(2565)  select sales,mark\_sales3(sales) from sales1 | |
| use sales2  create table if not exists loop\_table12(val1 int,val2 int)  Delimiter $$  create procedure insert\_data431()  Begin  DECLARE a INT Default 1 ;  DECLARE b INT Default 1 ;  generate\_data : loop  insert into loop\_table12 values (a,b);  SET a=a+1;  set b=a\*a;  if a = 15 then  leave generate\_data;  end if ;  end loop generate\_data;  End $$  call insert\_data431()  select \* from loop\_table12 | |
| /\* create a UDF to find out a log base 10 of any given number \*/  select log10(8)  delimiter $$  create function lognum132(a double(10,8))  returns double(10,8)  deterministic  begin  declare c double(10,8);  set c=log10(a);  return c;    end $$  select lognum132(8) as log10vaue | |
| select curdate()  /\* create a user defined function to find out a date differences in number of days \*/  delimiter $$  create function date\_diff(a date,b date )  returns int  deterministic  begin  declare c int;  set c=datediff(a,b);  return c;    end $$  select date\_diff(CURDATE(),'2016-02-01') as numdays | |
| /\* create a UDF which will be able to check a total number of records avaible in your table \*/  use sales2  select profit from sales1 order by profit desc limit 4,1  delimiter $$  create function chkrec423()  returns int  deterministic  begin  return (select profit from sales1 order by profit desc limit 4,1);    end $$  select chkrec423() | |
| /\* create a UDF which will be able to check a total number of records avaible in your table \*/  use sales2  # select count(\*) as first from sales1;  delimiter $$  create function chkrec21()  returns int  deterministic  begin  SELECT COUNT(\*) FROM sales1;    end $$  select chkrec21() | |
| **Procedure**  delimiter &&  create procedure chb()  begin  select \* from chb\_practice.bank\_details;  end &&  call chb() | |
| delimiter &&  create procedure max\_blc()  begin  select \* from chb\_practice.bank\_details where balance in (select max(balance) from bank\_details);  end &&  call max\_blc() | |
| DELIMITER &&  create procedure sel\_edu\_job11(in v1 varchar(30) , in v2 varchar(30) )  BEGIN  select count(\*) from bank\_details where education = v1 and job = v2;  END &&  call sel\_edu\_job1('secondary' , 'technician') | |
| **Primary Key and Foreign key** | |
| create table if not exists carera(  course\_id int not null,  course\_name varchar(30),  course\_status varchar(35),  num\_of\_enroll int,  primary key(course\_id)) | |
| create table if not exists stu\_carera(  stu\_id int ,  course\_id1 int,  course\_name varchar(30),  student\_name varchar(30),  student\_email varchar(30),  foreign key(course\_id1) references carera(course\_id)) | |
| alter table carera drop primary key  alter table class drop primary key  alter table carera add constraint test\_prim primary key(course\_id,course\_name)    drop table carera    drop table class | |
| create table if not exists test(  id int not null ,  name varchar(60),  email\_id varchar(60),  mobile\_no varchar(9),  address varchar(50))  alter table test add primary key(id)  alter table test drop primary key  alter table test add constraint test\_prim primary key(id,name) | |
| create table parent(  id int not null ,  primary key(id))  create table child (  id int ,  parent\_id int ,  foreign key (parent\_id) references parent(id)) | |
| drop table child | |
| create table child (  id int ,  parent\_id int ,  foreign key (parent\_id) references parent(id) on delete cascade ) | |
| create table child2 (  id int ,  parent\_id int ,  foreign key (parent\_id) references parent1(id) on update cascade ) | |
| select \* from child2  insert into child2 values(1,1),(1,2),(3,2),(2,2)  update parent1 set id=5 where id=2 | |
| **Window Function**  create table if not exists chb\_stu(  student\_id int,  student\_batch varchar(30),  student\_name varchar(30),  student\_stream varchar(30),  srudent\_marks int,  student\_email varchar(30)) | |
| select student\_batch,sum(srudent\_marks) from chb\_stu group by student\_batch  select student\_batch,min(srudent\_marks) from chb\_stu group by student\_batch  select student\_batch,max(srudent\_marks) from chb\_stu group by student\_batch  select student\_batch,avg(srudent\_marks) as avg\_marks from chb\_stu group by student\_batch  select count(student\_batch) from chb\_stu  select student\_batch,count(\*) from chb\_stu group by student\_batch | |
| select student\_id,student\_batch,student\_stream,srudent\_marks,row\_number() over(order by srudent\_marks desc) as 'row\_number' from chb\_stu | |
| select \* from(select student\_id,student\_batch,student\_stream,srudent\_marks,row\_number() over(partition by student\_batch order by srudent\_marks desc) as 'row\_num' from chb\_stu)as test where row\_num=3 | |
| select student\_id,student\_batch,student\_stream,srudent\_marks,row\_number() over(partition by student\_batch order by srudent\_marks desc)  as 'row\_num' from chb\_stu  select student\_id,student\_batch,student\_stream,srudent\_marks,row\_number() over(order by srudent\_marks desc)  as 'row\_num',rank() over(order by srudent\_marks desc ) as chb1\_rank from chb\_stu  select \* from(select student\_id,student\_batch,student\_stream,srudent\_marks,row\_number() over(partition by student\_batch order by srudent\_marks desc) as 'row\_num',rank() over (partition by student\_batch order by srudent\_marks desc ) as chb\_rank from chb\_stu) as new1 where chb\_rank=3  select \* from(select student\_id,student\_batch,student\_stream,srudent\_marks,row\_number() over(partition by student\_batch order by srudent\_marks desc) as 'row\_num',rank() over (partition by student\_batch order by srudent\_marks desc ) as 'chb\_rank', dense\_rank() over( partition by student\_batch order by srudent\_marks desc) as 'dense\_rank' from chb\_stu ) as test where `dense\_rank` = 2 | |
| **Joins** | |
| create table if not exists course (  course\_id int ,  course\_name varchar(50),  course\_desc varchar(60),  course\_tag varchar(50))  create table if not exists student(  student\_id int ,  student\_name varchar(30),  student\_mobile int ,  student\_course\_enroll varchar(30),  student\_course\_id int )  insert into course values(101 , 'fsda' , 'full stack data analytics' , 'Analytics'),  (102 , 'fsds' , 'full stack data analytics' , 'Analytics'),  (103 , 'fsds' , 'full stack data science' , 'DS'),  (104 , 'big data' , 'full stack big data' , 'BD'),  (105 , 'mern' , 'web dev' , 'mern'),  (106 , 'blockchain' , 'full stack blockchain' , 'BC'),  (101 , 'java' , 'full stack java' , 'java'),  (102 , 'testing' , 'full testing ' , 'testing '),  (105 , 'cybersecurity' , 'full stack cybersecurity' , 'cybersecurity'),  (109 , 'c' , 'c language' , 'c'),  (108 , 'c++' , 'C++ language' , 'language')  select \* from course  insert into student values(301 , "sudhanshu", 3543453,'yes', 101),  (302 , "sudhanshu", 3543453,'yes', 102),  (301 , "sudhanshu", 3543453,'yes', 105),  (302 , "sudhanshu", 3543453,'yes', 106),  (303 , "sudhanshu", 3543453,'yes', 101),  (304 , "sudhanshu", 3543453,'yes', 103),  (305 , "sudhanshu", 3543453,'yes', 105),  (306 , "sudhanshu", 3543453,'yes', 107),  (306 , "sudhanshu", 3543453,'yes', 103)  select \* from student  select c.course\_id,c.course\_name,c.course\_desc,s.student\_id,s.student\_name,s.student\_course\_id from  course c inner join student s on c.course\_id=s.student\_course\_id  select c.course\_id,c.course\_name,c.course\_desc,s.student\_id,s.student\_name,s.student\_course\_id from  course c left join student s on c.course\_id=s.student\_course\_id  select c.course\_id,c.course\_name,c.course\_desc,s.student\_id,s.student\_name,s.student\_course\_id from  course c left join student s on c.course\_id=s.student\_course\_id where s.student\_id is null  select c.course\_id,c.course\_name,c.course\_desc,s.student\_id,s.student\_name,s.student\_course\_id from  course c right join student s on c.course\_id=s.student\_course\_id  select c.course\_id,c.course\_name,c.course\_desc,s.student\_id,s.student\_name,s.student\_course\_id from  course c right join student s on c.course\_id=s.student\_course\_id where course\_id is null  select c.course\_id,c.course\_name,c.course\_desc,s.student\_id,s.student\_name,s.student\_course\_id from  course c cross join student s on c.course\_id=s.student\_course\_id  select c.course\_id,c.course\_name,c.course\_desc,s.student\_id,s.student\_name,s.student\_course\_id from  course c cross join student s  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| **Index**  create table if not exists course1 (  course\_id int ,  course\_name varchar(50),  course\_desc varchar(60),  course\_tag varchar(50),  index(course\_id)  )  show index from course1  insert into course1 values(101 , 'fsda' , 'full stack data analytics' , 'Analytics'),  (102 , 'fsds' , 'full stack data analytics' , 'Analytics'),  (103 , 'fsds' , 'full stack data science' , 'DS'),  (104 , 'big data' , 'full stack big data' , 'BD'),  (105 , 'mern' , 'web dev' , 'mern'),  (106 , 'blockchain' , 'full stack blockchain' , 'BC'),  (101 , 'java' , 'full stack java' , 'java'),  (102 , 'testing' , 'full testing ' , 'testing '),  (105 , 'cybersecurity' , 'full stack cybersecurity' , 'cybersecurity'),  (109 , 'c' , 'c language' , 'c'),  (108 , 'c++' , 'C++ language' , 'language')  select \* from course1  create table if not exists course2 (  course\_id int ,  course\_name varchar(50),  course\_desc varchar(60),  course\_tag varchar(50),  index(course\_id,course\_name)  )  show index from course2  create table if not exists course4 (  course\_id int ,  course\_name varchar(50),  course\_desc varchar(60),  course\_tag varchar(50),  index(course\_desc,course\_name,course\_tag)  )  insert into course4 values(101 , 'fsda' , 'full stack data analytics' , 'Analytics'),  (102 , 'fsds' , 'full stack data analytics' , 'Analytics'),  (103 , 'fsds' , 'full stack data science' , 'DS'),  (104 , 'big data' , 'full stack big data' , 'BD'),  (105 , 'mern' , 'web dev' , 'mern'),  (106 , 'blockchain' , 'full stack blockchain' , 'BC'),  (101 , 'java' , 'full stack java' , 'java'),  (102 , 'testing' , 'full testing ' , 'testing '),  (105 , 'cybersecurity' , 'full stack cybersecurity' , 'cybersecurity'),  (109 , 'c' , 'c language' , 'c'),  (108 , 'c++' , 'C++ language' , 'language')  show index from course4  select \* from course4  EXPLAIN ANALYZE select \* from course4 where course\_id = 106 or course\_name = 'fsds'  explain select \* from course4 where course\_id = 106  analyze table course4  describe course4  create table if not exists course6 (  course\_id int ,  course\_name varchar(50),  course\_desc varchar(60),  course\_tag varchar(50),  unique index(course\_desc,course\_name)  )  show index from course6 | |
| **UNION**  select course\_id , course\_name from course  union all  select student\_id , student\_name from student  (select course\_desc , course\_name from course  union all  select student\_id , student\_name from student ) | |
| **CTE**  with sample as(select \* from course where course\_id in (101,102,103))  select \* from sample where course\_tag='java'  with into1 as (select c.course\_id,c.course\_name,c.course\_desc,s.student\_id,s.student\_name,s.student\_course\_id from  course c cross join student s on c.course\_id=s.student\_course\_id) select course\_id,course\_name,student\_id,student\_name  from into1 where course\_id=102 | |
| CASE() Function in MySQL  CASE  WHEN condition1 THEN result1  WHEN condition2 THEN result2  WHEN conditionN THEN resultN  ELSE result  END;  1.CREATE TABLE float01001  (  user\_id int NOT NULL AUTO\_INCREMENT,  float\_val float,  PRIMARY KEY(user\_id)  );  INSERT float01001(float\_val)  VALUES (1.9);  INSERT float01001(float\_val) VALUES (1.1);  INSERT float01001(float\_val) VALUES (3.9);  INSERT float01001(float\_val) VALUES (5.0);  INSERT float01001(float\_val) VALUES (10.9);  SELECT float\_val,  CASE  WHEN float\_val > 5 THEN "The value is greater than 5"  WHEN float\_val = 5 THEN "The value is 5"  ELSE "The value is under 5"  END as float\_txt  FROM float01001;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  CREATE TABLE package099  (  user\_id int NOT NULL AUTO\_INCREMENT,  item VARCHAR(10),  mrp int,  PRIMARY KEY(user\_id)  );  INSERT package099(item, mrp) VALUES ('book1', 350);  INSERT package099(item, mrp) VALUES ('book2', 500);  SELECT mrp,  CASE  WHEN mrp > 400 THEN "Buy this item"  ELSE "Don't buy this item"  END as txt  FROM package099;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  CREATE TABLE package72  (  user\_id int NOT NULL AUTO\_INCREMENT,  item VARCHAR(10),  mrp int,  sp int,  PRIMARY KEY(user\_id)  );  INSERT package72(item, mrp, sp) VALUES ('book1', 250, 255);  INSERT package72(item, mrp, sp) VALUES ('book2', 350, 370);  INSERT package72(item, mrp, sp) VALUES ('book3', 400, 350);  SELECT mrp,sp,  CASE  WHEN sp > mrp THEN "Profit"  ELSE "Loss"  END as PL  FROM package72;  create database if not exists practice  use practice  create table if not exists student(student\_id int(10),  subject varchar(30),  marks int(10))  select \* from student where marks<>80  insert into student(student\_id,subject,marks) values (1001,'English',88)  insert into student(student\_id,subject,marks) values (1001,'Science',90)  insert into student(student\_id,subject,marks) values (1001,'Maths',85)  insert into student(student\_id,subject,marks) values (1002,'English',70)  insert into student(student\_id,subject,marks) values (1002,'Science',80)  insert into student(student\_id,subject,marks) values (1002,'Maths',83)  select marks, subject,  case  when marks > 80 then "good"  else "not good"  end  from student;  select student\_id,  sum(case when subject="English" then marks else 0 end) as English,  sum(case when subject="Science" then marks else 0 end) as Science,  sum(case when subject="Maths" then marks else 0 end) as Maths  from student group by student\_id;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  use practice  create table if not exists pair(a int,b int)  select \* from pair  insert into pair(a,b) values(1,2)  insert into pair(a,b) values(3,2)  insert into pair(a,b) values(2,4)  insert into pair(a,b) values(2,1)  insert into pair(a,b) values(5,6)  insert into pair(a,b) values(4,2)  select \* from pair t1 left join pair t2 on t1.b=t2.a and t1.a =t2.b  select \* from pair t1 left join pair t2 on t1.b=t2.a and t1.a =t2.b where t2.a is null or t1.a < t2.a  select t1.a,t1.b from pair t1 left join pair t2 on t1.b=t2.a and t1.a =t2.b where t2.a is null or t1.a < t2.a  use practice  create table if not exists merchant(mername varchar(30),  amount int(10),  payment\_mode varchar(30))  select \* from merchant  insert into merchant(mername,amount,payment\_mode) values ('chirag',520,'CASH')  insert into merchant(mername,amount,payment\_mode) values ('chirag',150,'ONLINE')  insert into merchant(mername,amount,payment\_mode) values ('dipak',450,'ONLINE')  insert into merchant(mername,amount,payment\_mode) values ('mayur',720,'CASH')  insert into merchant(mername,amount,payment\_mode) values ('dipak',580,'ONLINE')  insert into merchant(mername,amount,payment\_mode) values ('mayur',820,'CASH')  select student\_id,  sum(case when subject="English" then marks else 0 end) as English,  sum(case when subject="Science" then marks else 0 end) as Science,  sum(case when subject="Maths" then marks else 0 end) as Maths  from student group by student\_id;      select mername,  sum(case when payment\_mode="ONLINE" then amount else 0 end) as Online2,  sum(case when payment\_mode="CASH" then amount else 0 end) as Cash  from merchant group by mername;  use practice  create table if not exists col11(col1 int(5))  create table if not exists col22(col2 int(5))  select \* from col11  select \* from col22  insert into col11 values(1)  insert into col11 values(2);  insert into col11 values(1);  insert into col11 values(5);  insert into col11 values(NULL);  insert into col11 values(NULL);  insert into col22 values(null);  insert into col22 values(2);  insert into col22 values(5);  insert into col22 values(5);  select \* from col11 inner join col22 on col11.col1=col22.col2  select \* from col11 inner join col22 on col11.col1=col22.col2 or (col11.col1 is null and col22.col2 is null);  select \* from col11 inner join col22 on col11.col1 <=> col22.col2 | |
|  | |