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

<table>
<tr><td></td><td>alter table sales1 modify column Day_new int;</td></tr>
<tr><td></td><td>select month(order_date_new) from sales1</td></tr>
<tr><td></td><td>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</td></tr>
<tr><td></td><td>select order_id, discount, if(discount > 0, 'YES', 'NO') as discount_flag from sales1</td></tr>
<tr><td></td><td>alter table sales1 add column discount_flag varchar (20) after discount

update sales1 set discount_flag=if(discount>0, 'YES','NO')</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td>use sales2

select * from sales1</td></tr>
<tr><td colspan="2">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</td></tr>
<tr><td colspan="2">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;</td></tr>
</table>

```
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;
```

<table>
<tr><td colspan="2">END &&

call sel_edu_job1('secondary' , 'technician')</td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>