```
CREATE DATABASE commands;


use commands;


CREATE TABLE bank_transaction(id int,b_name varchar(30),amount_withdraw decimal(7,2),transaction_time timestamp,is_active boolean);


INSERT INTO bank_transaction values(1,'icici',30000.00,now(),true);


INSERT INTO bank_transaction values(2,'hdfc',60000.00,now(),true),(3,'axis',40000.00,now(),false),(4,'vijaya',80000.00,now(),true);

INSERT INTO bank_transaction values(5,'icici',90000.00,now(),false);



SELECT * FROM bank_transaction;


SELECT * FROM bank_transaction WHERE id=3;

SELECT * FROM bank_transaction WHERE b_name='sbi';

SELECT b_name FROM bank_transaction WHERE b_name='icici';

/*

to shaffle the column

AND: command- returns true if only both the conditions are true

if first conditions is false it doesnt check for second condition


cond1      cond2       result(AND)    result(OR)
```

|  |  |  |  |
|---|---|---|---|
| TRUE | FALSE | FALSE | TRUE |
| FALSE | TRUE | FALSE | TRUE |
| TRUE | TRUE | TRUE | TRUE |
| FALSE | FALSE | FALSE | FALSE |

```
UPDATE-

UPDATE table_name SET column_name='data' where condition;

*/

ALTER TABLE bank_transaction MODIFY b_name varchar(30) After is_active;


SELECT * FROM bank_transaction WHERE b_name='sbi' AND is_active=1;

SELECT * FROM bank_transaction WHERE b_name='icici' AND is_active=0;

SELECT * FROM bank_transaction WHERE b_name='iciCi' OR is_active=1;


SELECT * FROM bank_transaction WHERE (b_name='kotak' or is_active=1) and
b_name='icici';

UPDATE bank_transaction SET b_name='karur vyshya bank' where b_name='icici';


ALTER TABLE festivals RENAME COLUMN guests to place;

ALTER TABLE festivals RENAME COLUMN region to native_state;


DESC festivals;
```

```
ALTER TABLE festivals MODIFY COLUMN fest_name varchar(50);
ALTER TABLE festivals MODIFY COLUMN fest_name bigint;
ALTER TABLE festivals MODIFY COLUMN id bigint;
ALTER TABLE festivals MODIFY COLUMN id varchar(10);

/*
DESC: describe
DESC table_name;

RENAME the table name
RENAME table OLD_table_name TO NEW_TABLE_NAME;
*/

RENAME TABLE festivals TO festival_info;

SELECT * FROM festival_info;

/*
DROP-DDL statement(to drop entire table)

DROP TABLE table_name

TRUNCATE: to delete the data but table structure remains same

syntax to truncate the table
```

```sql
TRUNCATE TABLE table_name;

*/

ALTER TABLE festival_info ADD COLUMN (origin varchar(30),state varchar(20));



CREATE TABLE bank_transaction(id int,b_name varchar(30),amount_withdraw
decimal(7,2),transaction_time timestamp,is_active boolean);


INSERT INTO bank_transaction values(1,'icici',30000.00,now(),true);


INSERT INTO bank_transaction
values(2,'hdfc',60000.00,now(),true),(3,'axis',40000.00,now(),false),(4,'vijaya',80000.00,now(),true);


SELECT * FROM bank_transaction;


/*where clause - to filter the data


SLECT * FROM table_name WHERE column_name=?;


*/



SELECT * FROM festival_info Where id=3;

SELECT * FROM festival_info Where id=3;
```

AND,OR,NOT,IN,BETWEEN,LIKE,UPPER,LOWER,CONCAT,INSTR,SUBSTR,DISTINCT,

LIKE : used to perform pattern matching-can match any string of any length - used for varchar of any columns

%: will match any string of any length.

```
SELECT column_name FROM table_name;
```

```
SELECT * FROM movies WHERE cinema_name LIKE 's%';/*name starting with s*/
SELECT * FROM movies WHERE cinema_name LIKE 'm%';
```

```
SELECT * FROM movies WHERE cinema_name LIKE '%s';
```

```
SELECT * FROM movies WHERE nayaka LIKE '%p';
```

```
SELECT * FROM movies WHERE cinema_name LIKE '%A%';
SELECT * FROM movies WHERE nayaki LIKE '%h%';
```

UPPER: fetch the data and convert it into upper case

```
SELECT UPPER('shwetha');
SELECT UPPER(cinema_name) from movies;
SELECT LOWER('ShWETHAa');

SELECT LOWER(cinema_name) from movies;
SELECT LOWER(cinema_name,nayaka) from movies;/*cannot perform for multiple columns*/
```

CONCAT: join the 2 string characters or columns

```
SELECT CONCAT ('xworkz','odc');
SELECT CONCAT(cinema_name,nirdeshaka) from movies;
```

as keyword is used for alias name(temporary name)

```
SELECT CONCAT(cinema_name,nirdeshaka) as film_info from movies;
```

Instr: used to find position of the character in the existing string(position starts from 1)

takes 2 parameter- INSTR (STRING VALUE,POSITION OF A CHARACTER)

```
SELECT INSTR('XWORKSODC','r') as position;
SELECT INSTR('XWORKSODC','O') as position;

SELECT INSTR(cinema_name,'m'),cinema_name as position FROM movies;
```

Substr: to find substring from a string

'Bangalore'

```
SELECT SUBSTR('STRINGVALUE',STARTINGPOSITION,NOOFCHARACTERS);

SELECT SUBSTR('Bangalore',3,5);

SELECT SUBSTR('Xworkzodc',7,3);

SELECT SUBSTR('Xworkzodc',7,15);
```

```
SELECT SUBSTR(cinema_name,3,7) cinema_name from movies;

SELECT SUBSTR(cinema_name,5,7) cinema_name from movies;
```

DISTINCT: used to avoid duplicate values from table(will return unique values)

```
SELECT * from FESTIVALS;
```

```
select distinct(no_of_days_celebrated) from festivals;

select distinct(day) from festivals;
```

aggregate functions: perform calculations on perticular column, can be used only with SELECT statement,only with numeric value

1. count- will return number of rows from the table or column

2.sum- return sum all rows from perticular column

3.max

4. min

5.avg

count

```
SELECT COUNT(*) FROM currency;


SELECT COUNT(COUNTRY) as no_of_records FROM currency;

/*

SELECT * FROM task.currency_code;

*/
```

sum: add total  in column, only for numeric values

```
SELECT * FROM currency;

SELECT SUM(numeric_code) from currency;
```

max: returns maximum value from perticular column

```
SELECT MAX(currency_year) as maximum_year from currency;
```

min: returns minimum value from perticular column

```
SELECT min(currency_year) from currency;


SELECT * FROM currency WHERE currency_year = 1487;
```

avg: return the sum of all the values/no of records from the column

```
SELECT AVG(minor_unit) from currency;
SELECT AVG(indian_rupess_convert) from currency;
```

ENUM: datatype

```
CREATE TABLE crickets(id int,type enum('odi','test','t20'),overs int,no_of_runs int,location varchar(30));

SELECT * FROM crickets;

INSERT INTO crickets values(1,'odi',30,100,'bangalore');

INSERT INTO crickets values(2,3,30,100,'bangalore');

INSERT INTO crickets values(3,2,20,30,'chenani');
```

length: return no_of_characters in a perticular data (in column)

```
SELECT length(currency) as length from currency;
```