<https://rextester.com/l/mysql_online_compiler>

<https://www.jdoodle.com/online-mysql-terminal>

Good editor that shows Execution Plan:

<http://sqlfiddle.com/#!9/a6c585/1>

show databases;

create database d1;

use rextester;

show tables;

create table try1(name varchar(10),no int);

desc try1;

insert into try1 values('ABC',101);

insert into try1 values('XYZ',102);

insert into try1 values('JKL',103);

insert into try1 values('MNO',104);

insert into try1 values('PQR',105);

insert into try1 values('ABCD',101);

insert into try1 values('HIJKL',103);

insert into try1 values('WXYZ',102);

select \* from try1;

desc try1;

alter table try1 ADD PRIMARY KEY(no);

alter table try1 modify name varchar(20);

alter table try1 drop no;

alter table try1 add (name varchar(10));

delete from try1;

alter table try1 change column no clientno int(5);

truncate table try1;

update try1 set name='newclient' where clientno<103;

SELECT statement

Comparison operators

=, <>/!=,<, >, <=, >=, like

Wild card characters

%, \_

Logical operators

AND, OR, NOT

Joins, groupby with aggregate functions, group by with HAVING clause, orderby

* Find customer details containing 2nd alphabet ‘X’ in their name.

select \* from try1 where name like '\_X\_\_';

* Find customer details with name starting from ‘X’.

select \* from try1 where name like 'X\_\_';

* Find customer details whose name contains 3rd last character ‘X’.

select \* from try1 where name like '%X\_\_';

* Find customer details whose name ends with ‘X’.

select \* from try1 where name like '%X';

* Find customer details whose name starts with ‘x’.

select \* from try1 where name like 'x%';

* Find customer details with ‘M’ in their name.

select \* from try1 where name like '%M%';

* Find customer details whose name ends with ‘M’.

select \* from try1 where name like '%M';

* Find customer details whose name contains 1 alphabet after ‘JKL’.

select \* from try1 where name like '%JKL\_';

* Find customer details whose name contains ‘JKL’ in name.

select \* from try1 where name like '%JKL%';

select \* from try1 where (name like '%JKL%');

* Find customer details whose does not contain ‘JKL’.

select \* from try1 where NOT(name like '%JKL%');

* Find customer details whose clientno is >102.

select \* from try1 where not(clientno<102);

* Find customer details whose clientno is >103.

select \* from try1 where clientno>103;

* Find customer details whose clientno is >102.

select \* from try1 where clientno>103 or not(clientno<102);

* Find customer details whose clientno is >101.

select \* from try1 where clientno>103 and not(clientno<101);

sorted orders : Ascending, Descending

select \* from try1;

select \* from try1 order by clientno;

select \* from try1 order by clientno desc;

select \* from try1;

select \* from try1 order by clientno;

select \* from try1 order by clientno, name;

select \* from try1 order by clientno desc;

select \* from try1 order by clientno asc, name desc;

create table clientdata(clientname varchar(15), clientproject varchar(15), clientbill float(7,2), no\_of\_employees int(4), no\_of\_hours int);

alter table clientdata add(clientid int(4));

ALTER TABLE clientdata ADD PRIMARY KEY(clientid);

create table projectdata(projectid int, projectname varchar(15) references clientdata(clientproject), duration int);

create table teams(teamid int, technology varchar(15), projectname varchar(15) references clientdata(clientproject), members int, hours int);

alter table projectdata add primary key (projectname);

alter table clientdata add constraint cons\_fk foreign key (clientproject) references projectdata(projectname);

alter table teams add constraint cons\_fk1 foreign key (projectname) references projectdata(projectname);

desc projectdata;

insert into projectdata values(1001, 'AutocallSOS',260);

insert into projectdata values(1002, 'FastCheck',390);

insert into projectdata values(1003, 'EvenServe',430);

insert into projectdata values(1004, 'SmartAppDev',600);

insert into projectdata values(1005, 'Payease',800);

insert into projectdata values(1006, 'prj5',200);

insert into projectdata values(1007, 'FastCheck1',900);

insert into projectdata values(1008, 'EaseGST',700);

select \* from projectdata;

desc teams;

insert into teams values(2001,'.Net', 'SmartAppDev',6,12);

insert into teams values(2002,'Java', 'FastCheck',7,90);

insert into teams values(2003,'PHP', 'Payease',10,12);

insert into teams values(2004,'.Net', 'prj5',6,12);

insert into teams values(2005,'Android', 'EvenServe',7,210);

select \* from teams;

desc clientdata;

insert into clientdata values("Client1","EaseGST", 50000,15,70,101);

insert into clientdata values("Client-xyz","EvenServe", 6000,15,70,102);

insert into clientdata values("Client3","AutocallSOS",40000.00,3,100,103);

insert into clientdata values("Client4","EaseGST",7000.00,9,120,104);

insert into clientdata values("Client5","prj5",90000.00,6,144,105);

insert into clientdata values("Client8","prj5",70000.00,6,80,106);

insert into clientdata values("TasteIT","EvenServe",10000.00,7,210,107);

insert into clientdata values("NHAI","FastCheck",98000.00,2,90,108);

insert into clientdata values("NHAI","FastCheck1",80000.00,2,90,109);

insert into clientdata values("Client8","Payease",3000.00,5,12,110);

insert into clientdata values("SmartAppDev","FastCheck1",9000.00,5,12,111);

insert into clientdata values("LearnFast","FastCheck",17000.00,5,12,112);

select \* from clientdata;

Group by :

select clientproject, clientbill from clientdata group by clientproject;

select clientproject, sum(clientbill) from clientdata group by clientproject;

select count(no\_of\_hours) from clientdata;

select count(no\_of\_hours) from clientdata group by no\_of\_employees;

select no\_of\_employees, count(no\_of\_hours) from clientdata group by no\_of\_employees;

JOINS :

Set Theory :

Set A = {1,2,A}

Set B = {A, B, 1}

Cartesian Product : 3\*3=9

A×B = {(1,A),(1,B),(1,1),

(2,A),(2,B),(2,1),

(A,A),(A,B),(A,1)}

Intersection : A∩B = {1,A} = inner join/natural join/equi join (= operator in query)

Union : AUB = {1,2,A,B,A,1} = Full Outer join / non equality join (< or > or <> operators in query)

Set Difference : Set A = {1,2,A}

Set B = {A, B, 1}

A-B = {2}

B-A = {B}

LEFT OUTER JOIN : A∩BU(A-B)

RIGHT OUTER JOIN : A∩BU(B-A)

Teams

work

order

Projectdata

Clientdata

Use of Single table:

select projectname,technology from teams where projectname='EvenServe';

Use of two tables without join : (Cartesian Product)

select clientdata.clientname, teams.technology, teams.projectname from clientdata, teams where teams.projectname='EvenServe';

Use of two tables with join :

select clientdata.clientname, teams.technology, teams.projectname from clientdata, teams where teams.projectname='EvenServe' and clientdata.clientproject=teams.projectname;

select clientdata.clientname, teams.technology, teams.projectname, projectdata.duration from clientdata, teams, projectdata where teams.projectname='EvenServe' and clientdata.clientproject=teams.projectname and projectdata.projectname=teams.projectname ;

OUTER JOIN

/\* MySQL does not support full outer join. It can be emulated using UNION of LEFT and RIGHT OUTER JOINS \*/

select clientdata.clientname, teams.projectname from clientdata full[outer] join teams on clientdata.clientproject=teams.projectname;

select clientdata.clientname, teams.projectname from clientdata left outer join teams on clientdata.clientproject=teams.projectname;

select clientdata.clientname, teams.projectname from clientdata right outer join teams on clientdata.clientproject=teams.projectname;

select clientdata.clientname, teams.projectname from clientdata left outer join teams on clientdata.clientproject=teams.projectname

union

select clientdata.clientname, teams.projectname from clientdata right outer join teams on clientdata.clientproject=teams.projectname;

create table from other table’s data :

create table projectdata13 select \* from projectdata where projectid >1004;

select \* from projectdata13;

String, date, number functions :

String Functions :

[ASCII](https://www.w3schools.com/sql/func_mysql_ascii.asp)

[CHAR\_LENGTH](https://www.w3schools.com/sql/func_mysql_char_length.asp)

[CONCAT](https://www.w3schools.com/sql/func_mysql_concat.asp)

[FIND\_IN\_SET](https://www.w3schools.com/sql/func_mysql_find_in_set.asp)

[FORMAT](https://www.w3schools.com/sql/func_mysql_format.asp)

[INSERT](https://www.w3schools.com/sql/func_mysql_insert.asp)

[INSTR](https://www.w3schools.com/sql/func_mysql_instr.asp)

[LEFT](https://www.w3schools.com/sql/func_mysql_left.asp)

[LENGTH](https://www.w3schools.com/sql/func_mysql_length.asp)

[LOCATE](https://www.w3schools.com/sql/func_mysql_locate.asp)

[LOWER](https://www.w3schools.com/sql/func_mysql_lower.asp)

[LPAD](https://www.w3schools.com/sql/func_mysql_lpad.asp)

[LTRIM](https://www.w3schools.com/sql/func_mysql_ltrim.asp)

[MID](https://www.w3schools.com/sql/func_mysql_mid.asp)

[POSITION](https://www.w3schools.com/sql/func_mysql_position.asp)

[REPEAT](https://www.w3schools.com/sql/func_mysql_repeat.asp)

[REPLACE](https://www.w3schools.com/sql/func_mysql_replace.asp)

[REVERSE](https://www.w3schools.com/sql/func_mysql_reverse.asp)

[RIGHT](https://www.w3schools.com/sql/func_mysql_right.asp)

[RPAD](https://www.w3schools.com/sql/func_mysql_rpad.asp)

[RTRIM](https://www.w3schools.com/sql/func_mysql_rtrim.asp)

[STRCMP](https://www.w3schools.com/sql/func_mysql_strcmp.asp)

[SUBSTR](https://www.w3schools.com/sql/func_mysql_substr.asp)

[TRIM](https://www.w3schools.com/sql/func_mysql_trim.asp)

[UPPER](https://www.w3schools.com/sql/func_mysql_upper.asp)

Date Functions :

[CURDATE](https://www.w3schools.com/sql/func_mysql_curdate.asp)

[CURTIME](https://www.w3schools.com/sql/func_mysql_curtime.asp)

[ADDDATE](https://www.w3schools.com/sql/func_mysql_adddate.asp)

[ADDTIME](https://www.w3schools.com/sql/func_mysql_addtime.asp)

[DATEDIFF](https://www.w3schools.com/sql/func_mysql_datediff.asp)

[DATE\_FORMAT](https://www.w3schools.com/sql/func_mysql_date_format.asp)

[DATE\_SUB](https://www.w3schools.com/sql/func_mysql_date_sub.asp)

[DAYNAME](https://www.w3schools.com/sql/func_mysql_dayname.asp)

[DAYOFMONTH](https://www.w3schools.com/sql/func_mysql_dayofmonth.asp)

[DAYOFWEEK](https://www.w3schools.com/sql/func_mysql_dayofweek.asp)

[DAYOFYEAR](https://www.w3schools.com/sql/func_mysql_dayofyear.asp)

[EXTRACT](https://www.w3schools.com/sql/func_mysql_extract.asp)

[HOUR](https://www.w3schools.com/sql/func_mysql_hour.asp)

[MINUTE](https://www.w3schools.com/sql/func_mysql_minute.asp)

[YEAR](https://www.w3schools.com/sql/func_mysql_year.asp)

[QUARTER](https://www.w3schools.com/sql/func_mysql_quarter.asp)

[MONTH](https://www.w3schools.com/sql/func_mysql_month.asp)

[WEEK](https://www.w3schools.com/sql/func_mysql_week.asp)

[DAY](https://www.w3schools.com/sql/func_mysql_day.asp)

[MONTHNAME](https://www.w3schools.com/sql/func_mysql_monthname.asp)

[TIME\_FORMAT](https://www.w3schools.com/sql/func_mysql_time_format.asp)

Numeric Function :

[ABS](https://www.w3schools.com/sql/func_mysql_abs.asp)

[AVG](https://www.w3schools.com/sql/func_mysql_avg.asp)

[CEIL](https://www.w3schools.com/sql/func_mysql_ceil.asp)

[COUNT](https://www.w3schools.com/sql/func_mysql_count.asp)

[DIV](https://www.w3schools.com/sql/func_mysql_div.asp)

[EXP](https://www.w3schools.com/sql/func_mysql_exp.asp)

[FLOOR](https://www.w3schools.com/sql/func_mysql_floor.asp)

[GREATEST](https://www.w3schools.com/sql/func_mysql_greatest.asp)

[LEAST](https://www.w3schools.com/sql/func_mysql_least.asp)

[LOG](https://www.w3schools.com/sql/func_mysql_log.asp)

[MAX](https://www.w3schools.com/sql/func_mysql_max.asp)

[MIN](https://www.w3schools.com/sql/func_mysql_min.asp)

[MOD](https://www.w3schools.com/sql/func_mysql_mod.asp)

[PI](https://www.w3schools.com/sql/func_mysql_pi.asp)

[POWER](https://www.w3schools.com/sql/func_mysql_power.asp)

[RAND](https://www.w3schools.com/sql/func_mysql_rand.asp)

[ROUND](https://www.w3schools.com/sql/func_mysql_round.asp)

[SQRT](https://www.w3schools.com/sql/func_mysql_sqrt.asp)

[SUM](https://www.w3schools.com/sql/func_mysql_sum.asp)

[TRUNCATE](https://www.w3schools.com/sql/func_mysql_truncate.asp)

select projectname,ASCII(projectname) from projectdata;

select projectname,char\_length(projectname) from projectdata;

select projectname,concat("Name of the project for ", projectid," is ",projectname) from projectdata;

select duration,format(duration,2) from projectdata;

Subquery :