

W03. MySQL Practice

Create Database

- Error '**No database selected**' appears unless you create database at the first place.

```
CREATE database w03;  
USE w03;
```

Remove Database

```
drop database w03;
```

Create three tables: Sailors, Boats, Reserved.

```
create table Sailors (  
    sid integer primary key,  
    sname varchar(32) not null,  
    rating integer,  
    age integer  
);  
  
create table Boats (  
    bid integer primary key,  
    bname varchar(32) not null,  
    color varchar(16)  
);  
  
create table Reserved (  
    sid integer,  
    bid integer,  
    day date,  
    primary key (sid, bid, day),  
    foreign key (sid) references Sailors(sid),  
    foreign key (bid) references Boats(bid)  
);
```

Insert values into Table

```
insert into Sailors (sid, sname, rating, age) values (22, 'Dustin', 7, 45);  
insert into Sailors (sid, sname, rating, age) values (29, 'Brutus', 1, 33);  
insert into Sailors (sid, sname, rating, age) values (31, 'Lubber', 8, 55);  
insert into Sailors (sid, sname, rating, age) values (32, 'Andy', 8, 25);  
insert into Sailors (sid, sname, rating, age) values (58, 'Rusty', 10, 35);  
insert into Sailors (sid, sname, rating, age) values (64, 'Horatio', 1, 35);  
insert into Sailors (sid, sname, rating, age) values (71, 'Zorba', 10, 16);  
  
insert into Boats (bid, bname, color) values (101, 'Interlake', 'blue');  
insert into Boats (bid, bname, color) values (102, 'Interlake', 'red');  
insert into Boats (bid, bname, color) values (103, 'Clipper', 'green');  
insert into Boats (bid, bname, color) values (104, 'Marine', 'red');  
  
insert into Reserved (sid, bid, day) values (22, 101, DATE '2017-08-10');  
insert into Reserved (sid, bid, day) values (22, 102, DATE '2017-08-10');
```

```

insert into Reserved (sid, bid, day) values (22, 103, DATE '2017-08-12');
insert into Reserved (sid, bid, day) values (22, 103, DATE '2017-08-15');
insert into Reserved (sid, bid, day) values (22, 104, DATE '2017-08-12');
insert into Reserved (sid, bid, day) values (31, 102, DATE '2017-08-25');
insert into Reserved (sid, bid, day) values (31, 103, DATE '2017-08-30');
insert into Reserved (sid, bid, day) values (64, 101, DATE '2017-08-31');
insert into Reserved (sid, bid, day) values (71, 101, DATE '2017-08-31');

```

Single Table Queries

```

SELECT *
FROM Sailors
WHERE age > 30;

```

[RESULT]

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
58	Rusty	10	35
64	Horatio	1	35

5 rows in set (0.00 sec)

```

SELECT DISTINCT bname
FROM Boats;

```

[RESULT]

bname
Interlake
Clipper
Marine

3 rows in set (0.00 sec)

```

SELECT sname, age
FROM Sailors;

```

[RESULT]

sname	age
Dustin	45
Brutus	33
Lubber	55
Andy	25
Rusty	35
Horatio	35
Zorba	16

7 rows in set (0.00 sec)

```
SELECT DISTINCT bname
FROM Boats;
```

[RESULT]

bname
Interlake
Clipper
Marine

3 rows in set (0.00 sec)

```
SELECT DISTINCT bname, color
FROM Boats;
```

[RESULT]

bname	color
Interlake	blue
Interlake	red
Clipper	green
Marine	red

4 rows in set (0.00 sec)

Ambiguous Attribute Names

```
SELECT *
FROM Sailors
WHERE age > 30;
```

[RESULT]

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
58	Rusty	10	35
64	Horatio	1	35

5 rows in set (0.00 sec)

```
SELECT *
FROM Sailors
WHERE Sailors.age > 30;
```

[RESULT]

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
58	Rusty	10	35
64	Horatio	1	35

5 rows in set (0.00 sec)

```
SELECT *
FROM Sailors S
WHERE S.age > 30;
```

[RESULT]

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
58	Rusty	10	35
64	Horatio	1	35

5 rows in set (0.00 sec)

Substring Patten Matching

```
SELECT *
FROM Sailors S
WHERE S.sname LIKE 'A%';
```

[RESULT]

sid	sname	rating	age
32	Andy	8	25

1 row in set (0.00 sec)

Substring Patten Matching

```
SELECT *
FROM Sailors S
WHERE S.sname LIKE '%us%';
```

[RESULT]

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
58	Rusty	10	35

3 rows in set (0.00 sec)

```
SELECT *
FROM Sailors S
WHERE S.sname LIKE '_r%';
```

[RESULT]

sid	sname	rating	age
29	Brutus	1	33

1 row in set (0.00 sec)

Substring Patten Matching

```
SELECT *
FROM Sailors S
WHERE S.sname LIKE 'a_%_';
```

[RESULT]

sid	sname	rating	age
32	Andy	8	25

1 row in set (0.00 sec)

```
SELECT *
FROM Sailors S
WHERE S.sname LIKE 'a%y';
```

[RESULT]

sid	sname	rating	age
32	Andy	8	25

1 row in set (0.00 sec)

Substring Patten Matching

```
SELECT *
FROM Sailors S
WHERE S.sname NOT LIKE '%y';
```

[RESULT]

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
64	Horatio	1	35
71	Zorba	10	16

5 rows in set (0.00 sec)

```
SELECT *
FROM Sailors S
WHERE S.sname NOT LIKE '%o%';
```

[RESULT]

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35

5 rows in set (0.00 sec)

Arithmetic Operations

```
SELECT sname, rating + 1
FROM Sailors
WHERE age > 30;
```

[RESULT]

sname	rating + 1
Dustin	8
Brutus	2
Lubber	9
Rusty	11
Horatio	2

5 rows in set (0.00 sec)

Evaluating Multiple Table Queries

```
SELECT S.sname
FROM Sailors S, Reserved R
WHERE S.sid = R.sid AND R.bid = 101;
```

[RESULT]

sname
Dustin
Horatio
Zorba

3 rows in set (0.00 sec)

INNER JOIN in SQL

```
SELECT S.sname
FROM Sailors S
INNER JOIN Reserved R ON S.sid = R.sid
WHERE R.bid = 101;
```

[RESULT].

sname
Dustin
Horatio
Zorba

3 rows in set (0.00 sec)

NATURAL JOIN

```
SELECT S.sname
FROM Sailors S
NATURAL JOIN Reserved R
WHERE R.bid = 101;
```

[RESULT]

sname
Dustin
Horatio
Zorba

3 rows in set (0.00 sec)

SELF JOIN in SQL

```
create table Sailors2 (
  sid integer primary key,
  sname varchar(32) not null,
  mgrid integer,
  rating integer,
  age integer
);

insert into Sailors2 (sid, sname, mgrid, rating, age) values (22, 'Dustin', 29, 7, 45);
insert into Sailors2 (sid, sname, mgrid, rating, age) values (29, 'Brutus', 31, 1, 33);
insert into Sailors2 (sid, sname, mgrid, rating, age) values (31, 'Lubber', 31, 8, 55);
insert into Sailors2 (sid, sname, mgrid, rating, age) values (32, 'Andy', 22, 8, 25);
```

```
SELECT M.sname
FROM Sailors2 S, Sailors2 M
WHERE S.mgrid = M.sid;
```

[RESULT]

sname
Brutus
Lubber
Lubber
Dustin

4 rows in set (0.00 sec)

```
SELECT M.sname
FROM Sailors2 S
INNER JOIN Sailors2 M ON S.mgrid = M.sid;
```

[RESULT]

sname
Brutus
Lubber
Lubber
Dustin

4 rows in set (0.00 sec)

```
DROP TABLE Sailors2;
```

LEFT JOIN in SQL

```
SELECT R.sid, S.sname
FROM Reserved R
LEFT JOIN Sailors S ON R.sid = S.sid;
```

[RESULT]

sid	sname
22	Dustin
64	Horatio
71	Zorba
22	Dustin
31	Lubber
22	Dustin
22	Dustin
31	Lubber
22	Dustin

9 rows in set (0.00 sec)

```
SELECT S.sid
FROM Sailors S, Boats B, Reserved R
WHERE S.sid=R.sid AND R.bid=B.bid
      AND (B.color='red' OR B.color='green');
```

[RESULT]

```
+----+
| sid |
+----+
| 22  |
| 31  |
| 22  |
| 22  |
| 31  |
| 22  |
+----+
6 rows in set (0.00 sec)
```

Set Operators in SQL

```
SELECT S.sid
FROM Sailors S, Boats B, Reserved R
WHERE S.sid=R.sid AND R.bid=B.bid
      AND B.color='red'

UNION

SELECT S.sid
FROM Sailors S, Boats B, Reserved R
WHERE S.sid=R.sid AND R.bid=B.bid
      AND B.color='green';
```

[RESULT]

```
+----+
| sid |
+----+
| 22  |
| 31  |
+----+
2 rows in set (0.00 sec)
```

Set Operators in SQL

```
SELECT DISTINCT S.sid
FROM Sailors S, Boats B, Reserved R
WHERE S.sid=R.sid AND R.bid=B.bid
      AND B.color='blue' AND S.sid IN(
SELECT S.sid
FROM Sailors S, Boats B, Reserved R
WHERE S.sid=R.sid AND R.bid=B.bid
      AND B.color='green');
```

[CAUTION]

MySQL does not support INTERSECT. The following code does not work in MySQL.

```
SELECT S.sid
FROM Sailors S, Boats B, Reserved R
WHERE S.sid=R.sid AND R.bid=B.bid AND B.color='blue'
INTERSECT
SELECT S.sid
FROM Sailors S, Boats B, Reserved R
WHERE S.sid=R.sid AND R.bid=B.bid AND B.color='green';
```

[RESULT]


```
+----+
| sid |
+----+
| 22  |
+----+
1 row in set (0.03 sec)
```

Nested Queries (or Subqueries)

```
SELECT S.sname
FROM   Sailors S, Reserved R
WHERE  S.sid=R.sid AND bid=103;
```

[RESULT]

```
+----+
| sname |
+----+
| Dustin |
| Dustin |
| Lubber |
+----+
3 rows in set (0.00 sec)
```

```
SELECT S.sname
FROM   Sailors S
WHERE  S.sid IN (SELECT R.sid
                  FROM   Reserved R
                  WHERE  R.bid=103);
```

[RESULT]

```
+----+
| sname |
+----+
| Dustin |
| Lubber |
+----+
2 rows in set (0.02 sec)
```

Nested Queries (or Subqueries)

```
SELECT S.sname
FROM   Sailors S
WHERE  S.sid IN (SELECT R.sid
                  FROM   Reserved R
                  WHERE  R.bid=103);
```

[RESULT]

```
+----+
| sname |
+----+
| Dustin |
| Lubber |
+----+
2 rows in set (0.00 sec)
```

Multiple Nested Queries

```

SELECT S.sname
FROM Sailors S
WHERE S.sid IN (SELECT R.sid
                FROM Reserved R
                WHERE R.bid IN ( SELECT B.bid
                                FROM Boats B
                                WHERE B.color='red'));

```

[RESULT]

```

+-----+
| sname |
+-----+
| Dustin |
| Lubber |
+-----+
2 rows in set (0.00 sec)

```

Nested Queries with Correlation

```

SELECT S.sname
FROM Sailors S
WHERE EXISTS (SELECT *
              FROM Reserved R
              WHERE R.bid=103 AND S.sid=R.sid);

```

[RESULT]

```

+-----+
| sname |
+-----+
| Dustin |
| Lubber |
+-----+
2 rows in set (0.00 sec)

```

Nested Queries with Correlation

```

SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS (SELECT *
                  FROM Reserved R
                  WHERE S.sid=R.sid);

```

[RESULT]

```

+-----+
| sname |
+-----+
| Brutus |
| Andy   |
| Rusty  |
+-----+
3 rows in set (0.00 sec)

```

Nested Queries with Correlation

```

SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS
    (SELECT B.bid
     FROM Boats B
     WHERE NOT EXISTS(
         SELECT R.bid
         FROM Reserved R
         WHERE R.bid=B.bid AND R.sid=S.sid));

```

[RESULT]

sname
Dustin

1 row in set (0.00 sec)

More Set-Comparison Operators

```
SELECT *
FROM Sailors S
WHERE S.age > ANY (SELECT S2.age
                   FROM Sailors S2
                   WHERE S2.sname='Andy');
```

[RESULT]

sid	sname	rating	age
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
58	Rusty	10	35
84	Horatio	1	35

5 rows in set (0.00 sec)

Aggregate Operators

```
SELECT COUNT(*)
FROM Sailors S;
```

[RESULT]

COUNT(*)
7

1 row in set (0.00 sec)

[CAUTION]

Error occurs if you write as below.

```
SELECT COUNT (*)
FROM Sailors S;
```

```
SELECT AVG (S.age)
FROM Sailors S
WHERE S.age > 10;
```

[RESULT]

AVG (S.age)
34.8571

1 row in set (0.00 sec)

```
SELECT COUNT(DISTINCT S.age)
FROM Sailors S
WHERE S.sname = 'Bob';
```

[RESULT]

```
+-----+
| COUNT(DISTINCT S.age) |
+-----+
| 0 |
+-----+
1 row in set (0.00 sec)
```

```
SELECT S.sname
FROM Sailors S
WHERE S.rating = (SELECT MAX(S2.rating)
                  FROM Sailors S2);
```

[RESULT]

```
+-----+
| sname |
+-----+
| Rusty |
| Zorba |
+-----+
2 rows in set (0.00 sec)
```

Aggregate Operators

This query is illegal

```
SELECT S.sname, MAX(S.age)
FROM Sailors S;
```

[RESULT]

```
+-----+
| sname | MAX(S.age) |
+-----+
| Dustin | 55 |
+-----+
1 row in set (0.00 sec)
```

This query is correct!

```
SELECT S.sname, S.age
FROM Sailors S
WHERE S.age =
      (SELECT MAX(S2.age)
       FROM Sailors S2);
```

[RESULT]

```
+-----+
| sname | age |
+-----+
| Lubber | 55 |
+-----+
1 row in set (0.00 sec)
```

Evaluating GROUP BY in SQL

```
SELECT S.rating, MIN(S.age)
FROM Sailors S
WHERE S.age > 18
GROUP BY S.rating;
```

[RESULT]

rating	MIN(S.age)
7	45
1	33
8	25
10	35

4 rows in set (0.00 sec)

GROUP BY in SQL

```
SELECT S.rating, AVG(S.age)
FROM Sailors S
GROUP BY S.rating;
```

[RESULT]

rating	AVG(S.age)
7	45.0000
1	34.0000
8	40.0000
10	25.5000

4 rows in set (0.00 sec)

```
SELECT S.rating, MIN(S.age)
FROM Sailors S
WHERE S.age >= 18
GROUP BY S.rating;
```

[RESULT]

rating	MIN(S.age)
7	45
1	33
8	25
10	35

4 rows in set (0.00 sec)

Example: GROUP BY in SQL

```
SELECT B.bid, COUNT(*) AS scount
FROM Boats B, Reserved R
WHERE B.bid=R.bid AND B.color='red'
GROUP BY B.bid;
```

[RESULT]

bid	scount
102	2
104	1

2 rows in set (0.00 sec)

GROUP BY in SQL

```
SELECT S.sname
FROM Sailors S
WHERE S.age > ALL (SELECT S2.age
                  FROM Sailors S2
                  WHERE S2.rating = 10);
```

[RESULT]

sname
Dustin
Lubber

2 rows in set (0.00 sec)

```
SELECT S.sname
FROM Sailors S
WHERE S.age > (SELECT MAX(S2.age)
              FROM Sailors S2
              WHERE S2.rating = 10);
```

[RESULT]

sname
Dustin
Lubber

2 rows in set (0.00 sec)

Evaluating HAVING Clause

```
SELECT S.rating, MIN(S.age)
FROM Sailors S
WHERE S.age >= 18
GROUP BY S.rating
HAVING COUNT(*) > 1;
```

[RESULT]

rating	MIN(S.age)
1	33
8	25

2 rows in set (0.00 sec)

Example: HAVING in SQL

```
SELECT S.rating, AVG(S.age)
FROM Sailors S
GROUP BY S.rating
HAVING COUNT(*) > 1;
```

[RESULT]

rating	AVG (S age)
1	34.0000
8	40.0000
10	25.5000

3 rows in set (0.00 sec)

Example: HAVING in SQL

```
SELECT S.sname
FROM Sailors S
WHERE s.sid IN(SELECT R.sid
                FROM Reserved R
                WHERE R.bid=103 AND S.sid=R.sid
                GROUP BY R.bid HAVING COUNT(*)=1);
```

[RESULT]

```
+-----+
| sname |
+-----+
| Lubber |
+-----+
1 row in set (0.00 sec)
```

NULL Value

```
SELECT *
FROM   Sailors
WHERE  age IS NULL;
```

[RESULT]

```
Empty set (0.00 sec)
```