



IMPLEMENTATION OF SIMPLE JOIN OPERATIONS

Ishita Sinha

Consider the following schema:

Sailors (sid, sname, rating, age)

Boats (bid, bname, color)

Reserves (sid, bid, day(date))

- In sailors table, sid is NOT NULL and is the PRIMARY KEY. This was achieved using following queries:

```
[mysql> Alter table boats modify bid varchar(10) not null;  
Query OK, 0 rows affected (0.02 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> alter table sailors add primary key (sid);  
[Query OK, 0 rows affected (0.02 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

- In boats table, bid is NOT NULL and is the PRIMARY KEY
- In reserves table, all attributes are NOT NULL and sid, bid together is PRIMARY KEY

```
[mysql> describe sailors;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| sid   | varchar(10)   | NO   | PRI | NULL    |       |  
| sname | varchar(15)   | YES  |     | NULL    |       |  
| rating | float         | YES  |     | NULL    |       |  
| age   | int           | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
[mysql> describe boats;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| bid   | varchar(10)   | NO   | PRI | NULL    |       |  
| bname | varchar(15)   | YES  |     | NULL    |       |  
| color | varchar(10)   | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.01 sec)  
  
[mysql> describe reserves;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| sid   | varchar(10)   | NO   | PRI | NULL    |       |  
| bid   | varchar(10)   | NO   | PRI | NULL    |       |  
| day   | varchar(10)   | NO   |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

Displaying the values entered in the three tables:

```
[mysql> select * from sailors;
+-----+-----+-----+-----+
| sid    | sname | rating | age  |
+-----+-----+-----+-----+
| 1989001 | Bob   | 8.5    | 40   |
| 2001003 | Sam   | 9       | 35   |
| 1995293 | Rob   | 7.5     | 36   |
| 1980297 | Tim   | 8.5     | 45   |
| 1986929 | Tom   | 9       | 40   |
| 1975399 | Lan   | 10      | 50   |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
[mysql> select * from boats;
+-----+-----+-----+
| bid | bname      | color |
+-----+-----+-----+
| 101 | Titanic    | White |
| 202 | Oasis      | Red   |
| 303 | Liberty    | Green |
| 404 | Santa Maria | Red   |
| 505 | Orion      | Purple |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
[mysql> select * from reserves;
+-----+-----+-----+
| sid    | bid | day       |
+-----+-----+-----+
| 1989001 | 202 | 2009-02-28 |
| 1995293 | 404 | 1996-12-31 |
| 1989001 | 404 | 2002-08-25 |
| 1975399 | 101 | 1980-05-11 |
| 1980297 | 303 | 1999-9-9   |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

1. Find all information of sailors who have reserved boat number 101.

```
[mysql> select * from sailors s, reserves r where s.sid=r.sid and r.bid="101";
+-----+-----+-----+-----+-----+-----+-----+
| sid   | sname | rating | age  | sid   | bid  | day       |
+-----+-----+-----+-----+-----+-----+-----+
| 1975399 | Lan   | 10     | 50   | 1975399 | 101 | 1980-05-11 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

2. Find the name of boat reserved by Bob.

```
[mysql> select bname from sailors s, boats b, reserves r
[   -> where s.sid = r.sid and b.bid=r.bid and s.sname="Bob";
+-----+
| bname   |
+-----+
| Oasis   |
| Santa Maria |
+-----+
2 rows in set (0.00 sec)
```

3. Find the names of sailors who have reserved a red boat, and list in the order of age.

```
[mysql> select s.sname,s.age,r.bid from sailors s, boats b, reserves r
[   -> where s.sid=r.sid and b.bid=r.bid and b.color="Red"
[   -> order by s.age;
+-----+-----+-----+
| sname | age  | bid  |
+-----+-----+-----+
| Rob   | 36   | 404  |
| Bob   | 40   | 404  |
| Bob   | 40   | 202  |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

4. Find the names of sailors who have reserved at least one boat.

```
[mysql> select s.sname from sailors s, reserves r where s.sid=r.sid group by s.sname;
+-----+
| sname |
+-----+
| Bob   |
| Rob   |
| Tim   |
| Lan   |
+-----+
4 rows in set (0.00 sec)
```

Alternatively, using DISTINCT:

```
[mysql> select distinct s.sname from sailors s, reserves r where s.sid=r.sid;
+-----+
| sname |
+-----+
| Lan   |
| Tim   |
| Bob   |
| Rob   |
+-----+
4 rows in set (0.00 sec)
```

5. Find the ids of sailors who have reserved a red boat or a green boat.

```
[mysql> select s.sid,b.color from sailors s, boats b, reserves r
[   -> where s.sid=r.sid and b.bid=r.bid and b.color in ("red","green");
+-----+-----+
| sid   | color |
+-----+-----+
| 1980297 | Green |
| 1989001 | Red   |
| 1989001 | Red   |
| 1995293 | Red   |
+-----+-----+
4 rows in set (0.00 sec)
```

6. Find the name and the age of the youngest sailor.

```
[mysql> select sname,age from sailors where age = (select min(age) from sailors);
+-----+-----+
| sname | age |
+-----+-----+
| Sam   | 35  |
+-----+-----+
1 row in set (0.00 sec)
```

7. Count the number of different sailor names.

```
[mysql> select count(sname) from sailors;
+-----+
| count(sname) |
+-----+
|           6 |
+-----+
1 row in set (0.00 sec)
```

8. Find the average age of sailors for each rating level.

```
[mysql> select rating,avg(age) from sailors group by rating;
+-----+-----+
| rating | avg(age) |
+-----+-----+
|      10 | 50.0000 |
|      8.5 | 42.5000 |
|       9 | 37.5000 |
|      7.5 | 36.0000 |
+-----+-----+
4 rows in set (0.00 sec)
```