Recall the boat reservations database:

<table>
<thead>
<tr>
<th>sailor</th>
<th>sname</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>boat</td>
<td>bname</td>
<td>color</td>
</tr>
<tr>
<td>reservation</td>
<td>sname</td>
<td>bname</td>
</tr>
</tbody>
</table>

The rating attribute for boats indicate the minimum rating required of a sailor reserving the boat. We assume that `sname` is the primary key in `sailor` and `bname` is the primary key in `boat`. Also, `sname` is a foreign key of `reservation` referencing `sailor`, and `bname` is a foreign key of `reservation` referencing `boat`. Note that there may be sailors who have no reservations, and boats that are not currently reserved.

1. List the sailors who have at least one reservation and only reserved red boats.
   (i) tuple calculus using (at least one) universal quantification \( \forall \)

\[
\{ p : \text{sname} \mid \exists t \in \text{reservation} \ (p(\text{sname}) = t(\text{sname}) \land \forall r \in \text{reservation} \ \forall b \in \text{boat} \ ((r(\text{sname}) = t(\text{sname}) \land b(\text{bname}) = r(\text{bname})) \rightarrow b(\text{color}) = \text{red})) \} \]

(ii) rewrite the query in (i) in using only existential quantification \( \exists \)

\[
\{ p : \text{sname} \mid \exists t \in \text{reservation} \ (p(\text{sname}) = t(\text{sname}) \land \neg \exists r \in \text{reservation} \ \exists b \in \text{boat} \ ((r(\text{sname}) = t(\text{sname}) \land b(\text{bname}) = r(\text{bname})) \land b(\text{color}) \neq \text{red})) \}
\]

(iii) write the SQL query corresponding to the tuple calculus query in (ii), that uses NOT EXISTS tests on nested queries

```sql
select t.sname from reservation t
where not exists
  (select * from reservation r, boat b
   where r.sname = t.sname and b.bname = r.bname and b.color <> 'red')
```

The above answers use the fact that `bname` is a key in `boat` (so a boat has only one color). The following answers are also correct and do not make this assumption. In other words they work even if a boat can have multiple colors (and a red boat is defined as a boat for which at least one of the colors is red).

(i)

\[
\{ p : \text{sname} \mid \exists t \in \text{reservation} \ (p(\text{sname}) = t(\text{sname}) \land \forall r \in \text{reservation} \ (r(\text{sname}) = t(\text{sname}) \rightarrow \exists b \in \text{boat} \ (b(\text{bname}) = r(\text{bname}) \land b(\text{color}) = \text{red})) \} \}
(ii)

\[
\{ p : \text{name} \mid \exists t \in \text{reservation} (p(\text{name}) = t(\text{name}) \land \neg \exists r \in \text{reservation} (r(\text{name}) = t(\text{name})) \\
\quad \land \neg \exists b \in \text{boat} (b(\text{bname}) = r(\text{bname}) \land b(\text{color}) = \text{red})) \}
\]

```sql
select t.\text{name} from \text{reservation} t
where not exists
  (select * from \text{reservation} r
   where r.\text{name} = t.\text{name} and not exists
    (select * from \text{boat} b
     where b.\text{bname} = r.\text{bname} and b.\text{color} = \text{red})))
```

2. List the sailors who have reserved every red boat.

(i) tuple calculus using (at least one) universal quantification \(\forall\)

\[
\{ p : \text{name} \mid \exists s \in \text{sailor} [p(\text{name}) = s(\text{name}) \land \forall b \in \text{boat} ((b(\text{color}) = \text{red}) \\
\quad \rightarrow \exists r \in \text{reservation} (r(\text{name}) = s(\text{name}) \land r(\text{bname}) = b(\text{bname}))))] \}
\]

(ii) rewrite the query in (i) in using only existential quantification \(\exists\)

\[
\{ p : \text{name} \mid \exists s \in \text{sailor} [p(\text{name}) = s(\text{name}) \land \neg \exists b \in \text{boat} ((b(\text{color}) = \text{red}) \\
\quad \land \neg \exists r \in \text{reservation} (r(\text{name}) = s(\text{name}) \land r(\text{bname}) = b(\text{bname}))))] \}
\]

(iii) write the SQL query corresponding to the tuple calculus query in (ii), that uses NOT EXISTS tests on nested queries

```sql
select s.\text{name} from \text{sailor} s
where not exists
  (select * from \text{boat} b
   where b.\text{color} = \text{red} and
   not exists (select * from \text{reservation} r
    where r.\text{name} = s.\text{name} and r.\text{bname} = b.\text{bname} ))
```