Consider the following queries:

1. List the days appearing in the reservation table for which only red boats are reserved.
   (i) tuple calculus using (at least one) universal quantification

   \[ \{ d : day | \exists x \in \text{reservation}(d(day) = x(day) \land \forall r \in \text{reservation} (r(day) = x(day) \rightarrow \exists b \in \text{boat}( r(bname) = b(bname) \land b(color) = \text{red} )) \} \]

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

   \[ \{ d : day | \exists x \in \text{reservation} (d(day) = x(day) \land \neg \exists r \in \text{reservation} (r(day) = x(day) \land \neg \exists b \in \text{boat}( r(bname) = b(bname) \land b(color) = \text{red} )) \} \]

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

   ```sql
   select x.day from reservation x
   where not exists
   (select * from reservation r
   where r.day = x.day and not exists
   (select * from boat b
   where r.bname = b.bname and b.color = 'red'))
   ```

2. List the days when no red boat is reserved.
   (i) write the query in relational calculus using (at least one) universal quantification \( \forall \)

   \[ \{ d : day | \text{weekday}(d) \land \forall r \in \text{reservation} \forall b \in \text{boat} ((r(day) = d(day) \land r(bname) = b(bname)) \rightarrow b(color) \neq \text{red}) \} \]

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

   \[ \{ d : day | \text{weekday}(d) \land \neg \exists r \in \text{reservation} \exists b \in \text{boat} (r(day) = d(day) \land r(bname) = b(bname) \land b(color) = \text{red}) \} \]
(iii) write the SQL query corresponding directly to the relational calculus query in (ii), that uses NOT EXISTS tests on nested queries

```sql
select d.day from weekday d
where not exists
  (select * from reservation r, boat b
   where r.day = d.day and r.name = b.bname AND b.color= 'red')
```