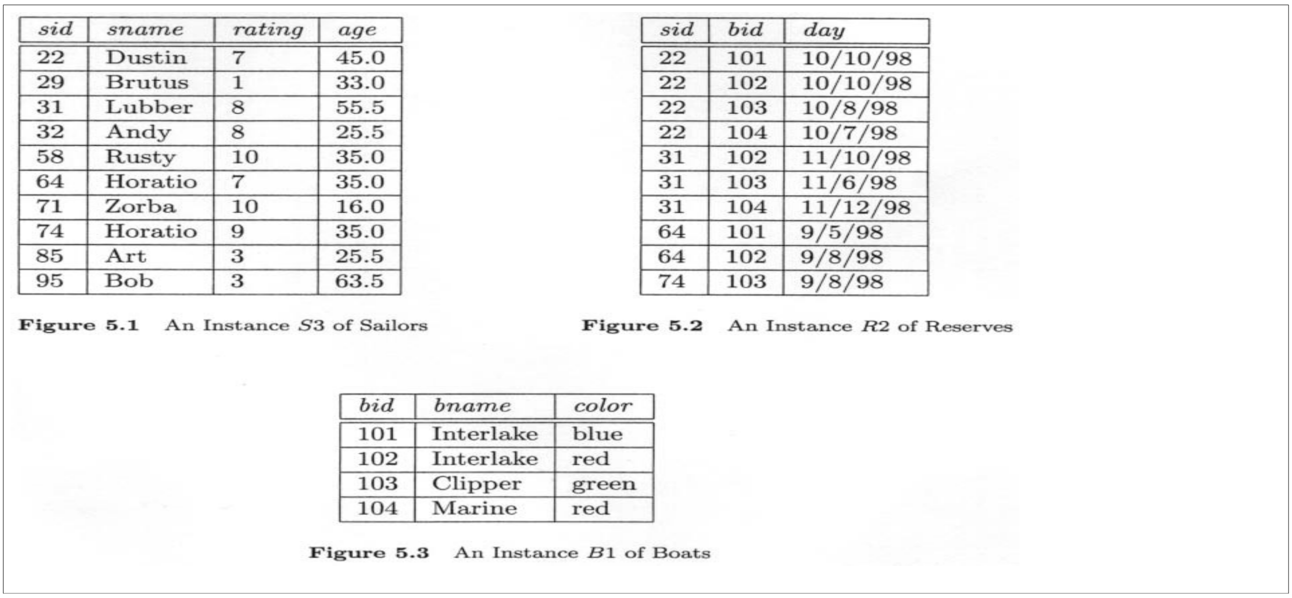
Questions&Solutions of Course Slide



sid	sname	rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Figure 5.1 An Instance *S*3 of Sailors

sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Figure 5.2 An Instance *R*2 of Reserves

bid	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Figure 5.3 An Instance *B*1 of Boats

1. Find sid's of sailors who’ve reserved at least one boat. (list distinct sid's in ascending order)

```
SELECT  distinct S.sid
FROM  Sailors S, Reserves R
WHERE  S.sid=R.sid order by s.sid;
```

2. Find sid’s of sailors who’ve reserved a red or a green boat. (list distinct sid's in ascending order)

```
SELECT  distinct S.sid
FROM  Sailors S, Boats B, Reserves R
WHERE  S.sid=R.sid AND R.bid=B.bid
  AND (B.color='red' OR B.color='green')
order by s.sid;
```

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

3. Find sid’s of sailors who’ve reserved a red and a green boat. (list distinct sid's in ascending order)

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

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

```
SELECT  distinct S.sid
FROM  Sailors S, Boats B, Reserves R
WHERE  S.sid=R.sid AND R.bid=B.bid AND B.color='red'
       AND S.sid IN  (SELECT  S2.sid
                      FROM  Sailors S2, Boats B2, Reserves R2
                      WHERE  S2.sid=R2.sid AND R2.bid=B2.bid
                             AND  B2.color='green')
order by S.sid;
```

4. Find sname’s of sailors who’ve reserved a red and a green boat. (list distinct name's in ascending order)

```
SELECT  distinct S3.sname
FROM  Sailors S3
WHERE  S3.sid IN
      (SELECT  R.sid
      FROM  Boats B, Reserves R
      WHERE  R.bid=B.bid
        AND B.color='red'
      INTERSECT
      SELECT R2.sid
      FROM  Boats B2, Reserves R2
      WHERE  R2.bid=B2.bid
        AND B2.color='green')
order by S3.sname;
```

5. Find names of sailors who’ve reserved boat #103. (list distinct name's in ascending order)

```
SELECT distinct S.sname
FROM  Sailors S
WHERE  S.sid IN  (SELECT  R.sid
                 FROM  Reserves R
                 WHERE  R.bid=103)
order by sname;
```

```
SELECT  distinct S.sname
FROM  Sailors S
WHERE   EXISTS  (SELECT  *
                FROM  Reserves R
                WHERE  R.bid=103 AND S.sid=R.sid)
order by sname;
```

6. Find sailors whose rating is greater than some sailor called 'Horatio'. (list sid,name,rating and age in ascending order).

NOTE: sqlite does not support ANY/ALL

This does **not**	SELECT *

work on
sqlite3
	FROM Sailors S
	WHERE S.rating > ANY (SELECT S2.rating
	FROM Sailors S2
	WHERE S2.sname='Horatio');

Alternative	SELECT *
	FROM Sailors S
	WHERE S.rating > (SELECT min(S2.rating)
	FROM Sailors S2
	WHERE S2.sname='Horatio')
	order by S.sid, S.sname, S.rating, S.age;

7. Find sailors whose rating is greater than every sailor called Horatio. (list sid,name,rating and age in ascending order).
NOTE: sqlite does not support ANY/ALL

This does **not** work on sqlite3	SELECT *
	FROM Sailors S
	WHERE S.rating > ALL (SELECT S2.rating
	FROM Sailors S2
	WHERE S2.sname='Horatio');

Alternative	SELECT *
	FROM Sailors S
	WHERE S.rating > (SELECT max(S2.rating)
	FROM Sailors S2
	WHERE S2.sname='Horatio')
	order by S.sid, S.sname, S.rating, S.age;

8. Find sailors with highest rating. (list sid,name,rating and age in ascending order)
NOTE: sqlite does not support ANY/ALL

This does **not** work on sqlite3	SELECT *
	FROM Sailors S
	WHERE S.rating >= ALL (SELECT S2.rating
	FROM Sailors S2)

Alternative	SELECT *
	FROM Sailors S
	WHERE S.rating = (SELECT max(S2.rating)
	FROM Sailors S2)
	order by S.sid, S.sname, S.rating, S.age;

9. Find names of sailors who’ve reserved all boats. (list name's in ascending order)

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

```
ORDER BY S.sname
```

```
SELECT  S.sname
FROM  Sailors S
WHERE  NOT EXISTS
             (SELECT  B.bid
                FROM  Boats B
               EXCEPT
                SELECT  R.bid
                 FROM  Reserves R
                 WHERE  R.sid=S.sid)
ORDER BY  S.sname;
```

10. Find name and age of the oldest sailor(s). (list name and age in ascending order)

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

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

11. Find the age of the youngest sailor with age>=18, for each rating with at least 2 such sailors. (list **rating** and **age** in ascending order)

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

12. For each red boat, find the number of reservations for this boat. (list **bid** and **count** in ascending order)

```
SELECT  B.bid,  COUNT (*) AS scount
FROM  Sailors S, Boats B, Reserves R
WHERE  S.sid=R.sid AND R.bid=B.bid AND B.color='red'
GROUP BY  B.bid
order by b.bid, scount;
```

13. Find the age of the youngest sailor with age >=18, for each rating with at least 2 sailors (of any age) (list **rating** and **age** in ascending order)

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

```
HAVING  1  <  (SELECT  COUNT (*)
                       FROM  Sailors S2
                       WHERE  S.rating=S2.rating )
ORDER BY S.rating,  MIN (S.age);
```

14. Find those ratings for which the average age is the minimum over all ratings. (list **rating** and **avg(age)** in ascending order)

This does **not** work on sqlite3	SELECT Temp.rating, Temp.avgage
	FROM (SELECT S.rating, AVG (S.age) AS avgage
	            FROM Sailors S
	            GROUP BY S.rating) AS Temp
	WHERE Temp.avgage = (SELECT MIN (Temp.avgage)
	                                  FROM Temp);

Alternative	select rating, avg(age) from Sailors S group by S.rating having avg(age)=(select min(avgage) from(select avg(S.age) avgage from Sailors S group by S.rating)) order by rating, avg(age);