

# CS245: Databases

## SQL

Vijaya saradhi

Department of Computer Science and Engineering  
Indian Institute of Technology Guwahati

# Reading Data From Tables

- Set/Bag operations
  - Union of two tables
  - Intersection between two tables
  - Difference of two tables
  - Cross product of two tables
  - Joining two tables
- Join
  - Natural join
  - Inner join (theta join)
  - Left outer join
  - Right outer join
  - Full outer join
  - Re-naming
  - Creating tables using queries
  - Having clause
  - Using all clauses of SELECT
  - Nested queries
  - Querying tables on go
  - Correlated nested queries
  - Examples

# Example Database

Sailors			
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

Reserves		
sid	bid	day
22	101	10-Oct-2019
22	102	10-Oct-2019
22	103	08-Oct-2019
22	104	07-Oct-2019
31	102	10-Nov-2019
31	103	06-Nov-2019
31	104	12-Nov-2019
64	101	05-Sep-2019
64	102	08-Sep-2019
74	103	08-Sep-2019

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

# Re-naming

## Example - 01

table		table2	
A	B	B	D
1	2	2	3
1	2	4	5
		4	5

New Table			
A1	B1	B2	D1
1	2	2	3
1	2	2	3

```
SELECT A AS A1, table1.B AS B1,  
table2.B AS B2, D AS D1  
FROM table1  
JOIN table2  
ON table1.B = table2.B  
WHERE table1.A < table2.D;
```

# Create Table

Create a table out of select query

table		table2	
A	B	B	D
1	2	2	3
1	2	4	5
		4	5

Result			
A1	B1	B2	D1
1	2	2	3
1	2	2	3

```
CREATE TABLE Result (
    SELECT A AS A1, table1.B AS B1,
           table2.B AS B2, D AS D1
    FROM   table1
    JOIN   table2
    ON     table1.B = table2.B
    WHERE  table1.A < table2.D);
```

# Select continued - 01

Group by rating such that each group has at least two sailors

```
SELECT      *
FROM        Sailors
GROUP BY    rating
HAVING COUNT(rating) > 1;
```

## Output

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

$\gamma_{rating}(Sailors)$			
sid	sname	rating	age
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0
85	Art	3	25.5

## Select continued - 02

Group by rating such that each group has at least two sailors where sailor age  $\geq 30$

```
SELECT      *
FROM        Sailors
WHERE       age >= 30
GROUP BY    rating
HAVING COUNT( rating ) > 1;
```

Output

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

$\gamma_{rating}(Sailors)$			
sid	sname	rating	age
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

# All six clauses of SELECT

Group by rating such that each group has at least two sailors where sailor age  $\geq 20$  sort by sailor names

```
SELECT      *
FROM        Sailors
WHERE       age >= 20
GROUP BY    rating
HAVING COUNT( rating ) > 1
ORDER BY    sname;
```

Output

Sailors			
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

$\gamma_{rating}(Sailors)$			
sid	sname	rating	age
85	Art	3	25.5
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

# (Correlated) Nested Queries

Nested Query

Query containing a sub-query

Correlated Nested Query

A sub-query which refers to columns of the tables of its containing query

# Queries on Example Database

Q1: Find the names of the **Sailors** who have reserved **Boat 103**

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

test condition

Is S.sid in {22, 31, 74} then  
retrieve sname

Result: {Dustin, Lubber, Horatio}

Sailors			
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

ε

Reserves		
sid	bid	day
22	101	10-Oct-2019
22	102	10-Oct-2019
22	103	08-Oct-2019
22	104	07-Oct-2019
31	102	10-Nov-2019
31	103	06-Nov-2019
31	104	12-Nov-2019
64	101	05-Sep-2019
64	102	08-Sep-2019
74	103	08-Sep-2019

# Nested Queries - Example 02

Find the names of sailors who have NOT reserved boat 103

```
SELECT S1.sname
FROM Sailors AS S1
WHERE S1.sid NOT IN
    (SELECT R1.sid
     FROM Reserves AS R1
     WHERE R1.bid = 103)
```

# Nested Queries - Example 03

Find the names of Sailors who have reserved a red boat

```
SELECT S1.sname
FROM Sailors AS S1
WHERE S1.sid IN
    (SELECT R1.sid
     FROM Reserves AS R1
     WHERE R1.bid IN
        (SELECT B1.bid
         FROM Boats AS B1
         WHERE B1.color = 'red')
    )
```

# Correlated Nested Queries

## Correlated nested queries

- The inner sub-query has been completely independent of the outer query
- In general, the inner sub-query could depend on the row currently being examined in the outer query
- Such queries are known as **Correlated** nested queries

Find the names of sailors who have reserved boat number 103

```
SELECT S1.sname
FROM Sailors AS S1
WHERE EXISTS
  (SELECT *
   FROM Reserves AS R1
   WHERE R1.bid = 103
   AND R1.sid = S1.sid
  )
```

EXISTS test whether a set is nonempty

# Correlated Nested Queries

Find the names of sailors who have reserved boat number 103

```
SELECT S1.sname
FROM Sailors AS S1
WHERE EXISTS
  (SELECT *
   FROM Reserves AS R1
   WHERE R1.bid = 103
   AND R1.sid = S1.sid
  )
```

Result: {Dustin, Lubber, Horatio}

Sailors			
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

Is Reserves not an empty set?

Reserves		
sid	bid	day
22	101	10-Oct-2019
22	102	10-Oct-2019
22	103	08-Oct-2019
22	104	07-Oct-2019
31	102	10-Nov-2019
31	103	06-Nov-2019
31	104	12-Nov-2019
64	101	05-Sep-2019
64	102	08-Sep-2019
74	103	08-Sep-2019

# Example 19 - Complex Correlated Nested Queries

Find the names of sailors who have reserved both red and green boat

```
SELECT S1.sname
FROM   Sailors AS S1
WHERE  S1.sid IN
       (SELECT R1.sid
        FROM   Reserves AS R1
        JOIN   Boats AS B1
        ON     R1.bid = B1.bid
        WHERE  B1.color = 'red'
               AND    R1.bid IN
                   (SELECT R1.sid
                    FROM   Reserves AS R1
                    JOIN   Boats AS B1
                    ON     R1.bid = B1.bid
                    WHERE  B1.color = 'green'
                   )
      )
```

# Example 20(a) - Complex Correlated Nested Queries

Find the names of sailors who have reserved all boats

```
SELECT S1.sname
FROM Sailors AS S1
WHERE NOT EXISTS
  (SELECT B1.bid
   FROM Boats AS B1 NOT IN
     (SELECT R1.bid
      FROM Reserves AS R1
      WHERE R1.sid = S1.sid)
  )
```

# Example 20(b) - Complex Correlated Nested Queries

Find the names of sailors who have reserved all boats

```
SELECT S1.sname
FROM Sailors AS S1
WHERE NOT EXISTS
  (SELECT B1.bid
   FROM Boats AS B1
   WHERE NOT EXISTS
     (SELECT R1.bid
      FROM Reserves AS R1
      WHERE R1.sid = S1.sid
      AND R1.bid = B1.bid
    )
  )
```

# Queries on Example Database

Q1: Find the names of the **Sailors** who have reserved **Boat** 103

```
SELECT S.sname  
FROM Sailors AS S, Reserves AS R  
WHERE (S.sid = R.sid AND R.bid = 103)
```

# Queries on Example Database

Q1: Find the names of the **Sailors** who have reserved **Boat 103**

```
SELECT sname  
FROM Sailors  
JOIN Reserves  
ON Sailors.sid = Reserves.sid  
WHERE (Reserves.bid = 103)
```

# Queries on Example Database

Q1: Find the names of the **Sailors** who have reserved **Boat** 103

```
SELECT S.sname  
FROM Sailors AS S  
JOIN Reserves AS R  
ON S.sid = R.sid  
WHERE (R.bid = 103)
```

# Queries on Example Database

Q1: Find the names of the **Sailors** who have reserved **Boat** 103

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

# Queries on Example Database

Q2: Find the names of the **Sailors** who reserved a red boat

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

# Queries on Example Database

Q2: Find the names of the **Sailors** who **reserved** a red boat

```
SELECT S.sname  
FROM Sailors AS S  
JOIN Reserves AS R  
JOIN Boats AS B  
ON (S.sid = R.sid AND R.bid = B.bid)  
WHERE B.color = 'red';
```

# Queries on Example Database

Q3: Find the colors of boats reserved by Lubber

```
SELECT B.color  
FROM Sailors AS S, Reserves AS R, Boats AS B  
WHERE (S.sid = R.sid AND R.bid = B.bid AND  
        S.sname = 'Lubber')
```

# Queries on Example Database

Q3: Find the colors of boats reserved by Lubber

```
SELECT B.color  
FROM Sailors AS S  
JOIN Reserves AS R  
JOIN Boats AS B  
ON (S.sid = R.sid AND R.bid = B.bid)  
WHERE S.sname = 'Lubber';
```

# Queries on Example Database

Q4: Find the names of Sailors who have reserved at least one boat

```
SELECT S.sname  
FROM Sailors AS S, Reserves AS R  
WHERE (S.sid = R.sid);
```

# Queries on Example Database

Q4: Find the names of **Sailors** who have **reserved** at least one boat

```
SELECT S.sname  
FROM Sailors AS S  
JOIN Reserves AS R  
ON (S.sid = R.sid);
```

# Queries on Example Database

Q5: Find the names of **Sailors** who have reserved a red or a green **Boat**

```
SELECT S.sname  
FROM Sailors AS S, Reserves AS R, Boats AS B  
WHERE (S.sid = R.sid AND R.bid = B.bid AND B.color = 'red');
```

**UNION**

```
SELECT S.sname  
FROM Sailors AS S, Reserves AS R, Boats AS B  
WHERE (S.sid = R.sid AND R.bid = B.bid AND B.color = 'green');
```

# Queries on Example Database

Q5: Find the names of Sailors who have reserved a red or a green Boat

```
(SELECT S.sname  
FROM   Sailors AS S  
JOIN   Reserves AS R  
JOIN   Boats AS B  
ON     (S.sid = R.sid AND R.bid = B.bid)  
WHERE  B.color = 'red');
```

**UNION**

```
(SELECT S.sname  
FROM   Sailors AS S  
JOIN   Reserves AS R  
JOIN   Boats AS B  
ON     (S.sid = R.sid AND R.bid = B.bid)
```

# Queries on Example Database

Q6: Find the names of Sailors who have reserved a red AND a green Boat

```
(SELECT R.sid  
FROM Reserves AS R  
JOIN Boats AS B  
ON (R.bid = B.bid)  
WHERE B.color = 'red' AND R.sid
```

IN

```
(SELECT R.sid  
FROM Reserves AS R  
JOIN Boats AS B  
ON (R.bid = B.bid)  
WHERE B.color = 'green');
```

# Queries on Example Database

Q6: Find the names of Sailors who have reserved a red AND a green Boat

```
SELECT S1.sname  
FROM Sailors AS S1  
JOIN (SELECT R.sid  
      FROM Reserves AS R  
      JOIN Boats AS B  
      ON (R.bid = B.bid)  
WHERE B.color = 'red' AND R.sid  
      IN  
      (SELECT R.sid  
      FROM Reserves AS R  
      JOIN Boats AS B  
      ON (R.bid = B.bid)  
WHERE B.color = 'green')) AS S2  
ON S1.sid = S2.sid;
```

# Queries on Example Database

Q7: Find the names of Sailors who have reserved at least two boats

```
CREATE TABLE Temp1 AS
  (SELECT S.sid , S.sname , R.bid
   FROM   Sailors AS S , Reserves AS R
   WHERE S.sid = R.sid );
```

```
SELECT T1.sname
FROM   Temp1 AS T1
JOIN   Temp1 AS T2
ON     T1.sid = T2.sid
WHERE  (T1.bid <> T2.bid)
```

# Queries on Example Database

Q7: Find the names of Sailors who have reserved at least two boats

```
SELECT T1.sname  
FROM (SELECT S.sid , S.sname , R.bid  
      FROM Sailors AS S, Reserves AS R  
     WHERE S.sid = R.sid) AS T1  
  JOIN (SELECT S.sid , S.sname , R.bid  
        FROM Sailors AS S, Reserves AS R  
       WHERE S.sid = R.sid) AS T2  
ON T1.sid = T2.sid  
WHERE (T1.bid <> T2.bid);
```

# Queries on Example Database

Q8: Find the sids of Sailors with age over 20 who have not reserved a red boat

```
SELECT S1.sid  
FROM      Sailors AS S1  
WHERE    S1.age >= 20 AND S1.sid  
           NOT IN  
(  
  SELECT S2.sid  
  FROM      Sailors AS S2  
  JOIN      Reserves AS R2  
  JOIN      Boats As B2  
  ON        (S2.sid = R2.sid AND R2.bid = B2.bid)  
  WHERE    (B.color = 'red'))  
)
```