

# SQL Queries

Chapter 5,  
Sections 5.1-5.6

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 1

---

---

---

---

---

---

---

---

## SQL Query Language

- Implements relational algebra...
  - Select, Project, Join, Set operators
- And so much more...
  - Correlated subqueries
  - Ordering of results
  - Aggregate queries (e.g., SUM, MAX, AVG)
  - Three-valued logic for NULL values
  - Etc.

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 2

---

---

---

---

---

---

---

---

## Basic SQL Query

Optional
Attributes from input relations
List of relations

**SELECT** [DISTINCT] attr-list  
**FROM** relation-list  
**WHERE** qualification

Attr1 op Attr2  
OPS: <, >, =, <=, >=, <>  
Combine using AND, OR, NOT

*(Conceptual)* Evaluation:

1. Take cross-product of relation-list
2. Select rows satisfying qualification
3. Project columns in attr-list (eliminate duplicates only if DISTINCT)

Optimizer chooses efficient plan!!

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 3

---

---

---

---

---

---

---

---

### Example of Basic Query

- Schema:
  - Sailors (sid, sname, rating, age)
  - Boats (bid, bname, color)
  - Reserves (sid, bid, day)
- Find the names of sailors who have reserved boat #103

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

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 4

---

---

---

---

---

---

---

---

### Example of Basic Query

Reserves			Sailors			
sid	bid	day	sid	sname	rating	age
22	101	10/10	22	dustin	7	45
58	103	11/12	58	rusty	10	35
			31	lubber	8	55

  

Reserves x Sailors						
sid	bid	day	sid	sname	rating	age
22	101	10/10	22	dustin	7	45
22	101	10/10	58	rusty	10	35
22	101	10/10	31	lubber	8	55
58	103	11/12	22	dustin	7	45
58	103	11/12	58	rusty	10	35
58	103	11/12	31	lubber	8	55

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 5

---

---

---

---

---

---

---

---

### Example of Basic Query

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

What's the effect of adding DISTINCT?

What's the equivalent relational algebra?

$$\pi_{\text{name}} (\sigma_{\text{bid}=103}(\text{Sailors} \bowtie \text{Reserves}))$$

RA project removes duplicates!

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 6

---

---

---

---

---

---

---

---

## Another Example

- Schema:
  - Sailors (sid, sname, rating, age)
  - Boats (bid, bname, color)
  - Reserves (sid, bid, day)
- Find the colors of boats reserved by a sailor named rusty

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

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 7

---

---

---

---

---

---

---

---

## Note on Range Variables

- Needed when same relation appears twice in FROM clause

```
SELECT S1.sname, S2.sname
FROM Sailors S1, Sailors S2
WHERE S1.age > S2.age
```

**What does this Query compute?**

*Good style to always use range variables anyway...*

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 8

---

---

---

---

---

---

---

---

## ORDER BY clause

- In relational algebra, cannot specify ordering of results
- SQL has an extra ORDER BY clause

**Attribute(s) in ORDER BY clause must be in SELECT list.**

<i>Find the names and ages of all sailors, in increasing order of age</i>	<i>Find the names and ages of all sailors, in decreasing order of age</i>
SELECT S.sname, S.age FROM Sailors S ORDER BY S.age [ASC]	SELECT S.sname, S.age FROM Sailors S ORDER BY S.age DESC

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 9

---

---

---

---

---

---

---

---

## ORDER BY clause

```
SELECT S.sname, S.age, S.rating
FROM Sailors S
ORDER BY S.age ASC, S.rating DESC
```

**What does this query compute?**

Find the names, ages, and rankings of all sailors.  
Sort the result in increasing order of age.  
If there is a tie, sort those tuples in decreasing order of rating.

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 10

---

---

---

---

---

---

---

---

## Union

- Find names of sailors who have reserved a red or a green boat

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

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

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 11

---

---

---

---

---

---

---

---

## Intersect

- Find sids of sailors who have reserved a red and a green boat

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

**What if we replace sid with sname (non-key)?**

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 12

---

---

---

---

---

---

---

---

## More Set Operators

- Also EXCEPT (set difference)
- (Painful) SQL oddities
  - For UNION, default is to eliminate duplicates!
  - To keep duplicates, must specify **UNION ALL**

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 13

---

---

---

---

---

---

---

---

## Nested Queries

- Query with another query embedded inside

**What does this Compute?**

```

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

**Conceptual evaluation:** *For each row of Sailors, evaluate the subquery over reserves*

To find sailors who have not reserved 103, use NOT IN

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 14

---

---

---

---

---

---

---

---

## Nested Queries with Correlation

*Find names of sailors who have reserved exactly one boat.*

```

SELECT S.sname
FROM Sailors S
WHERE EXISTS (SELECT UNIQUE R.bid
              FROM Reserves R
              WHERE S.sid=R.sid)
  
```

True if value appears once

**Illustrates why subquery must be evaluated for each tuple in Sailors.**

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 15

---

---

---

---

---

---

---

---

## More Set Comparison Operators

- We've already seen IN, EXISTS and UNIQUE. Can also use NOT IN, NOT EXISTS and NOT UNIQUE.
- Also available: *op ANY, op ALL, op is <, ≤, >, ≥, =, ≠*
- Find sailors whose rating is greater than that of all sailors called Horatio:
 

```
SELECT *
FROM Sailors S
WHERE S.rating > ALL (SELECT S2.rating
                     FROM Sailors S2
                     WHERE S2.sname='Horatio')
```

**What if we replace ALL with ANY?**

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 16

---

---

---

---

---

---

---

---

## Aggregate Operators

```
SELECT COUNT (*) FROM Sailors S
```

```
SELECT COUNT (DISTINCT S.name)
FROM Sailors S
```

```
SELECT AVG (S.age)
FROM Sailors S
WHERE S.rating=10
```

```
SELECT AVG (DISTINCT S.age)
FROM Sailors S
WHERE S.rating=10
```

```
SELECT S.sname FROM Sailors S
WHERE S.rating= (SELECT MAX(S2.rating) FROM Sailors S2)
```

COUNT (\*)  
COUNT ( [DISTINCT] A)  
SUM ( [DISTINCT] A)  
AVG ( [DISTINCT] A)  
MAX (A) *Can use Distinct*  
MIN (A) *Can use Distinct*

*single column*

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 17

---

---

---

---

---

---

---

---

## Aggregate Query - Example

- Find name and age of oldest sailor(s)

```
SELECT S.sname, MAX (S.age)
FROM Sailors S
```

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

How many tuples in the result?

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 18

---

---

---

---

---

---

---

---

## GROUP BY

- Conceptual evaluation
  - Partition data into groups according to some criterion
  - Evaluate the aggregate for each group

**Example:** For each rating level, find the age of the youngest sailor

```
SELECT MIN (S.age), S.rating
FROM Sailors S
GROUP BY S.rating
```

How many tuples in the result?

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 19

---

---

---

---

---

---

---

---

---

---

## GROUP BY and HAVING

```
SELECT [DISTINCT] target-list
FROM relation-list
WHERE qualification
GROUP BY grouping-list
HAVING group-qualification
```

**Target-list contains:**

- Attribute names (subset of grouping-list)
- Aggregate operations (e.g., min(age))

**Conceptual Evaluation:**

1. Eliminate tuples that don't satisfy qualification
2. Partition remaining data into groups
3. Eliminate groups according to group-qualification
4. Evaluate aggregate operation(s) for each group

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 20

---

---

---

---

---

---

---

---

---

---

**Find the age of the youngest sailor with age >= 18, for each rating with at least 2 such sailors**

```
SELECT S.rating, MIN (S.age)
FROM Sailors S
WHERE S.age >= 18
GROUP BY S.rating
HAVING COUNT (*) > 2
```

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
71	zorba	10	16.0
64	horatio	7	35.0
29	brutus	1	33.0
58	rusty	10	35.0

rating	age
1	33.0
7	45.0
7	35.0
8	55.5
10	35.0

rating	age
7	35.0

Answer relation

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 20

---

---

---

---

---

---

---

---

---

---

## NULL Values in SQL

- NULL represents 'unknown' or 'inapplicable'
- Query evaluation complications
  - Q: Is (rating > 10) true when rating is NULL?
  - A: Condition evaluates to 'unknown' (not T or F)
- What about AND, OR connectives?
  - Need 3-valued logic
- WHERE clause eliminates rows that don't evaluate to true

p	q	p AND q	p OR q
T	T	T	T
T	F	F	T
T	U	U	T
F	T	F	T
F	F	F	F
F	U	F	U
U	T	U	T
U	F	F	U
U	U	U	U

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 22

---

---

---

---

---

---

---

---

---

---

## NULL Values Example

What does this query return?

```
SELECT sname
FROM sailors
WHERE age > 45
      OR age <= 45
```

sid	sname	rating	age
22	dustin	7	45
58	rusty	10	NULL
31	lubber	8	55

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 23

---

---

---

---

---

---

---

---

---

---

## NULL Values in Aggregates

- NULL values generally ignored when computing aggregates
  - Modulo some special cases (see textbook)

```
SELECT AVG(age)
FROM sailors
```

**Returns 50!**

sid	sname	rating	age
22	dustin	7	45
58	rusty	10	NULL
31	lubber	8	55

1/27/09 EECS 484: Database Management Systems, Kristen LeFevre 24

---

---

---

---

---

---

---

---

---

---