

SQL Basics

1 December 2009
Lecture 7

Topics for Today

- Overview of SQL
- Form of a Basic SQL Query
- Set operations

December 1, 2009

ISE 322: Database Systems

2

Overview of SQL

- Based on Relational Algebra (by Codd)
- Merge of SEQUEL-XRM and System-R by IBM
- ANSI/ISO Standards
 - SQL-92
 - SQL:1999
 - Vendor specific variants

December 1, 2009

ISE 322: Database Systems

3

Overview of SQL

- Not all vendors implement SQL:1999
 - Some not even SQL-92
- Different flavors, variants, key words
 - Commonly used parts are similar

December 1, 2009

ISE 322: Database Systems

4

Overview of SQL

- Various parts
 - Data Manipulation Language
 - Data Definition Language
 - Triggers and Advanced Integrity Constraints
 - Client-Server Execution and Remote Database Access
 - Security

December 1, 2009

ISE 322: Database Systems

5

Overview of SQL

- Advanced features
 - OOP – next semester
 - Recursive queries
 - Decision support
 - Data mining
 - Spatial data
 - XML

December 1, 2009

ISE 322: Database Systems

6

Sample Schemas

Sailors(sid:integer, sname:string, rating:integer, age:real)
 Boats(bid:integer, bname:string, color:string)
 Reserves(sid:integer, bid:integer, day:date)

December 1, 2009

ISE 322: Database Systems

7

Sample Tables

• 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	10/10/98
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

• Boats

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

December 1, 2009

ISE 322: Database Systems

8

Basic SQL Queries

- Each SQL query must be evaluated
 - We will talk about an evaluation strategy
 - Optimizers will convert to algebra and try to save
- Simple queries:
 - Select list (filter **columns**)
 - From list (tables)
 - Optional qualification phrase (filter **rows**)
 - Eliminate doubles with DISTINCT

```
SELECT [DISTINCT] select-list
FROM from-list
WHERE qualification
```

December 1, 2009

ISE 322: Database Systems

9

Using DISTINCT

- Example: Find the names and ages of all Sailors using S3

```
SELECT DISTINCT S.sname, S.age FROM Sailors S
```

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

sname	age
Dustin	45.0
Brutus	33.0
Lubber	55.5
Andy	25.5
Rusty	35.0
Horatio	35.0
Zorba	16.0
Horatio	35.0
Art	25.5
Bob	63.5

sname	age
Dustin	45.0
Brutus	33.0
Lubber	55.5
Andy	25.5
Rusty	35.0
Horatio	35.0
Zorba	16.0
Art	25.5
Bob	63.5

December 1, 2009

ISE 322: Database Systems

10

Basic SQL Queries

- Add a filter
- Find all sailors with a rating equal to or above 7

```
SELECT S.sid, S.sname, S.rating, S.age FROM Sailors S
WHERE S.rating >= 7
```

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

sid	sname	rating	age
22	Dustin	7	45.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
74	Horatio	9	35.0

December 1, 2009

ISE 322: Database Systems

11

Basic SQL Queries

- Taking it apart
 - FROM-list is where data comes from
 - SELECT-list chooses the columns to display
 - QUALIFICATION-list compares values to allow rows in the answer
 - DISTINCT removes doubles

December 1, 2009

ISE 322: Database Systems

12

Relational Algebra to SQL

Relational Algebra Symbol	SQL Term
$\pi_{x,y}$	SELECT x, y
$R \times S$	FROM R, S
$\sigma_{x>y}$	WHERE $x > y$
$R \bowtie_c S$	(R JOIN S) WHERE c , or FROM R, S WHERE c
$R \bowtie S$	S NATURAL JOIN R
$\rho(New(p1 \rightarrow o_1, p2 \rightarrow o_2), E)$	SELECT p_1 AS o_1, p_2 AS o_2 FROM E AS New WHERE ...

December 1, 2009

ISE 322: Database Systems

13

Conceptual Evaluation Strategy

1. Compute cross product in FROM list
2. Delete all rows which do not satisfy WHERE
3. Delete columns not in SELECT
4. If DISTINCT given, eliminate duplicates

December 1, 2009

ISE 322: Database Systems

14

Conceptual Evaluation Strategy

- Find the names of sailors who have reserved boat number 103

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

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(bid:integer, bname:string, color:string)
Reserves(sid:integer, bid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

15

Conceptual Evaluation Strategy

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

sid	bid	day
22	58	10/10/96
58	103	11/12/96

sid	name	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(bid:integer, bname:string, color:string)
Reserves(sid:integer, bid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

16

Conceptual Evaluation Strategy

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

sid	sname	rating	age	sid	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(bid:integer, bname:string, color:string)
Reserves(sid:integer, bid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

17

More SQL Examples

- Find the sids of sailors who have reserved a red boat

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

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(bid:integer, bname:string, color:string)
Reserves(sid:integer, bid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

18

More SQL Examples

- Find the names of sailors who have reserved a red 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'
```

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(hid:integer, hname:string, color:string)
Reserves(sid:integer, hid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

19

More SQL Examples

- Find the colors of boats reserved by Lubber

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

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(hid:integer, hname:string, color:string)
Reserves(sid:integer, hid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

20

More SQL Examples

- Find the names of sailors who have reserved at least one boat

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

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(hid:integer, hname:string, color:string)
Reserves(sid:integer, hid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

21

More complex

- You can put anything in the column comparisons

```
SELECT S1.sname AS name1, S2.sname AS name2
FROM Sailors S1, Sailors S2
WHERE S2.rating < S1.rating - 1
```

December 1, 2009

ISE 322: Database Systems

22

More Complex

- Compute increments for the ratings of persons who have sailed two different boats on the same day

```
SELECT S.sname, S.rating + 1 as rating
FROM Sailors S, Reserves R1, Reserves R2
WHERE S.sid = R1.sid AND S.sid = R2.sid
AND R1.day = R2.day AND R1.bid <> R2.bid
```

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(hid:integer, hname:string, color:string)
Reserves(sid:integer, hid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

23

Strings

- LIKE
- %
 - 'Begin' LIKE 'Be%'
- _
 - 'Begin' LIKE 'Begi_'

December 1, 2009

ISE 322: Database Systems

24

Strings

- Find the ages of all sailors whose name begins and ends with B and has at least three characters

```
SELECT S.age
FROM Sailors S
WHERE S.name LIKE 'B_%B'
```

December 1, 2009

ISE 322: Database Systems

25

So far

- Overview of SQL
- Form of a Basic SQL Query
- Set operations

December 1, 2009

ISE 322: Database Systems

26

Set Operations

- UNION, INTERSECT, EXCEPT
 - MINUS same as EXCEPT
 - Similar to Relational Algebra
- IN, NOT IN
- op ANY, op ALL
 - op can be: <, <=, >, >=, =, <>
- EXISTS, NOT EXISTS
- UNIQUE, NOT UNIQUE
 - UNIQUE true if there are not duplicates (or empty)
 - False otherwise

December 1, 2009

ISE 322: Database Systems

27

Using OR

- Query: Find the 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')
```

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(bid:integer, bname:string, color:string)
Reserves(sid:integer, bid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

28

Using Union

- Query: Find the 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'
UNION
SELECT S2.sname FROM Sailors S2, Reserves R2, Boats B2
WHERE S2.sid = R2.sid AND R2.bid = B2.bid
AND B2.color = 'green'
```

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(bid:integer, bname:string, color:string)
Reserves(sid:integer, bid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

29

Using Intersect

- Find the names of sailors who have reserved a red and 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'
INTERSECT
SELECT S2.sname FROM Sailors S2, Reserves R2, Boats B2
WHERE S2.sid = R2.sid AND R2.bid = B2.bid
AND B2.color = 'green'
```

```
Sailors(sid:integer, sname:string, rating:integer, age:real)
Boats(bid:integer, bname:string, color:string)
Reserves(sid:integer, bid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

30

Using Except/Minus

- Find the sids of all sailors who have reserved red boats but not green boats

```
SELECT S.sid FROM Sailors S, Reserves R, Boats B
WHERE S.sid = R.sid AND R.bid = B.bid AND B.color =
'red'
EXCEPT
SELECT S2.sid FROM Sailors S2, Reserves R2, Boats B2
WHERE S2.sid = R2.sid AND R2.bid = B2.bid AND
B2.color = 'green'
```

```
Sailors(sid:integer, name:string, rating:integer, age:real)
Boats(bid:integer, name:string, color:string)
Reserves(sid:integer, bid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

31

Using Except/Minus

- Find the sids of all sailors who have reserved red boats but not green boats

```
SELECT R.sid FROM Reserves R, Boats B WHERE R.bid =
B.bid AND B.color = 'red'
EXCEPT
SELECT R2.sid FROM Reserves R2, Boats B2 WHERE
R2.bid = B2.bid AND B2.color = 'green'
```

```
Sailors(sid:integer, name:string, rating:integer, age:real)
Boats(bid:integer, name:string, color:string)
Reserves(sid:integer, bid:integer, day:date)
```

December 1, 2009

ISE 322: Database Systems

32

About duplicates

- Using UNION, INTERSECT, EXCEPT automatically removes duplicates
 - Just like Relational Algebra
- You can leave them using ALL
 - UNION ALL
 - INTERSECT ALL
 - EXCEPT ALL

December 1, 2009

ISE 322: Database Systems

33

Conclusion

- Overview of SQL
- Form of a Basic SQL Query
- Set operations

December 1, 2009

ISE 322: Database Systems

34