In the Lecture Series Introduction to Database Systems

SQL

Presented by Stéphane Bressan

Structured Query Language

- Originally developed in the System-R project of IBM (1974)
- Industry standard for relational databases (SQL92 is an ANSI/ISO standard)

Structured Query Language

- Data Definition Language for defining relations, views, integrity constraints, triggers
- Data Manipulation Language for updating, and querying
- Database Control Language for defining access rights, concurrency control, etc….

Querying One Table

Find all the information about students

```
SELECT *
FROM student;
```

```
SELECT name, email, year, faculty, department, graduate,
FROM student;
```

name	email	year	faculty	department	graduate
XIE XIN	xiexin2011@gmail.com	1/1/2007	Faculty of Science	Chemistry	
HUANG RAN	huangran1991@yahoo.com	1/8/2007	Faculty of Science	Biology	
GOH ENG CHYE	gohengchye1992@msn.com	1/8/2007	School of Computing	CS	
GOH HUI YING	gohhuiying1989@gmail.com	1/1/2008	Faculty of Science	Biology	
…					

Selecting Columns

Find the names and emails of students

```
SELECT name, email
FROM student;
```

name	email
XIE XIN	xiexin2011@gmail.com
HUANG RAN	huangran1991@yahoo.com
GOH ENG CHYE	gohengchye1992@msn.com
GOH HUI YING	gohhuiying1989@gmail.com
…	

Selecting Rows

Find the names and emails of computer science students

```
SELECT name, email
FROM student
WHERE department='CS';
```

name	email
GOH ENG CHYE	gohengchye1992@msn.com
ZHOU HUICHAN	zhouhuichan1990@msn.com
LIU SHAOJUN	liushaojun2010@msn.com
QIN YIYANG	qinyiyang2010@hotmail.com
…	

Selecting Rows

Find the names and emails of students who graduated after '01/08/10'

```
SELECT name, email, graduate
FROM student
WHERE graduate>= '01/08/10';
```

name	email	graduate
GERALDINE LEE	glee@msn.com	01/08/10
ADELINE WONG	awong007@msn.com	01/08/10
TANG CHEE YONG	tcy@hotmail.com	01/08/10

The condition must be TRUE (not FALSE, not UNKNOWN)

Querying Multiple Tables

Find the names of students and the titles of the books they own

```
SELECT student.name, book.title
FROM student, copy, book
WHERE student.email=copy.owner
AND copy.book=book.ISBN13;
```

name	title
XIE XIN	The Law
XIE XIN	Microsoft Office Access 2007: Comprehensive Concepts and Techniques (Shelly Cashman)
XIE XIN	Student Atlas of World Geography
XIE XIN	Fire from Ice: Searching for the Truth Behind the Cold Fusion Furor (Wiley Science Editions)
…	

Tuple Variables

Find the names of students and the titles of the books they own

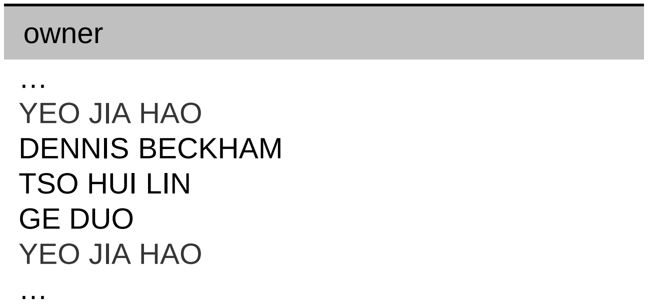
```
SELECT s.name, b.title
FROM student s, copy c, book b
WHERE s.email=c.owner
AND c.book=b.ISBN13;
```

name	title
XIE XIN	The Law
XIE XIN	Microsoft Office Access 2007: Comprehensive Concepts and Techniques (Shelly Cashman)
XIE XIN	Student Atlas of World Geography
XIE XIN	Fire from Ice: Searching for the Truth Behind the Cold Fusion Furor (Wiley Science Editions)
…	

Renaming

Find the names of students who lent a book returned after 2010-03-04 to anniechapman1991@yahoo.com

```
SELECT s.name AS owner
FROM loan l, student s
WHERE  s.email=l.owner
   AND l.returned  > '2010-03-04'
   AND l.borrower = 'anniechapman1991@yahoo.com';
```



```
owner
…
YEO JIA HAO
DENNIS BECKHAM
TSO HUI LIN
GE DUO
YEO JIA HAO
…
```

Duplicates

Find the different names of students who lent a book returned after 2010-03-04 to anniechapman1991@yahoo.com

```
SELECT DISTINCT s.name AS owner
FROM loan l, student s
WHERE  s.email=l.owner
   AND l.returned  > '2010-03-04'
   AND l.borrower = 'anniechapman1991@yahoo.com';
```

owner
…
DENNIS BECKHAM
GE DUO
TSO HUI LIN
YEO JIA HAO
…

Ordering Rows

Find the names of students who lent a book returned after 2010-03-04 to anniechapman1991@yahoo.com in descending alpha-numerical order

```
SELECT s.name AS owner
FROM loan l, student s
WHERE s.email=l.owner
   AND l.returned  > '2010-03-04'
   AND l.borrower = 'anniechapman1991@yahoo.com'
   ORDER BY name DESC;
```

owner
…
ZENG YIHUI
YEO JIA HAO
YEO JIA HAO
…

Ordering Rows

Find the ISBN14 of the books that have been borrowed by anniechapman1991@yahoo.com , their borrowing and return dates in ascending order of the borrowing and return dates

```
SELECT l.book, l.borrowed, l.returned
FROM loan l
WHERE l.borrower='anniechapman1991@yahoo.com'
ORDER BY l.borrowed, l.returned;
```

book	borrowed	returned
978-1405072878	1/1/2010	17/3/2010
978-0596007126	1/1/2010	14/4/2010
978-9562913621	2/1/2010	3/1/2010
978-0470526705	4/1/2010	11/1/2010
…		

Arithmetic

Find the price of the Schaum s Outline books and add 50% to it

```
SELECT m.price * 1.5
FROM mgh m
WHERE m.title LIKE '%Schaum s Outline%’;
```

mgh

ISBN13	title	authors	price
978-0071508612	Schaum s Outline of Calculus	Frank Ayres, Elliott Mendelson	10
978-0071635264	Schaum s Outline of Chinese Grammar	Claudia Ross	12
978-0071639309	Practice Makes Perfect Spanish Verb Tenses	Dorothy Richmond	25

(No column name)
15
18

Aggregate Queries: Counting Rows

Find the total number of (different) books

```
SELECT COUNT(*)
   FROM book b;
```

```
SELECT COUNT(DISTINCT *)
   FROM book b;
```

(No column name)
311

Aggregate Queries: Counting Rows

Find the total number of titles

```
SELECT COUNT(b.title)
FROM book b;

SELECT COUNT(ALL b.title)
FROM book b;
```

(No column name)
311

Aggregate Queries: Counting Rows

Find the total number of different titles

```
SELECT COUNT(DISTINCT b.title)
FROM book b;
```

(No column name)
301

Aggregate Queries: Average, Minimum, etc.

Find the average price of a book from McGrawHill;

```
SELECT AVG(m.price)
FROM mgh m;
```

mgh

ISBN13	title	authors	price
978-0071508612	Schaum s Outline of Calculus	Frank Ayres, Elliott Mendelson	10
978-0071635264	Schaum s Outline of Chinese Grammar	Claudia Ross	12
978-0071639309	Practice Makes Perfect Spanish Verb Tenses	Dorothy Richmond	25

(No column name)
15.66

Aggregate Queries: Grouping

Find, for each day, the number of books borrowed by anniechapman1991@yahoo.com

```
SELECT COUNT(l.book)
FROM loan l
WHERE l.borrower='anniechapman1991@yahoo.com'
GROUP BY l.borrowed
```

Can we get 0?

(No column name)
2
1
1
2
…

Aggregate Queries: Grouping

This one eliminates duplicates

```
SELECT l.book
FROM loan l
GROUP BY l.book;
```

```
SELECT DISTINCT l.book
FROM loan l;
```

Aggregate Queries: Grouping

Find, for each day, the number of books borrowed by anniechapman1991@yahoo.com, print the day and the quantity

```
SELECT l.borrower, l.borrowed, COUNT(l.book)
FROM loan l
WHERE l.borrower='anniechapman1991@yahoo.com'
GROUP BY l.borrowed;
```

borrowed	Expr1001
1/1/2010	2
2/1/2010	1
4/1/2010	1
12/1/2010	1
16/1/2010	1

Aggregate Queries: Grouping

Find, for each day, the number of books borrowed by anniechapman1991@yahoo.com, print the borrower, the day and the quantity

```
SELECT l.borrower, l.borrowed, COUNT(l.book)
FROM loan l
WHERE l.borrower='anniechapman1991@yahoo.com'
GROUP BY l.borrowed
```

"not a GROUP BY expression"

Aggregate Queries: Grouping

Find, for each day, the number of books borrowed by anniechapman1991@yahoo.com, print the borrower, the day and the quantity

```
SELECT l.borrower, l.borrowed, COUNT(l.book)
FROM loan l
WHERE l.borrower='anniechapman1991@yahoo.com'
GROUP BY l.borrowed, l.borrower
```

borrower	borrowed	Expr1002
anniechapman1991@yahoo.com	1/1/2010	2
anniechapman1991@yahoo.com	2/1/2010	1
anniechapman1991@yahoo.com	4/1/2010	1
anniechapman1991@yahoo.com	12/1/2010	1
…		

Aggregate Queries: Grouping

Find, for each day, the number of books borrowed, print the borrower, the day and the quantity

```
SELECT l.borrower, l.borrowed, COUNT(l.book)
FROM loan l
GROUP BY l.borrowed, l.borrower;
```

borrower	borrowed	Expr1002
angjiayi1990@hotmail.com	1/1/2010	1
anniechapman1991@yahoo.com	1/1/2010	2
davidhall1992@yahoo.com	1/1/2010	1
dennispalmer1992@yahoo.com	1/1/2010	1
…		

Aggregate Queries: Grouping

What does this query find?

```
SELECT l.borrower, l.borrowed, COUNT(l.book)
FROM loan l
GROUP BY l.borrowed, l.borrower, l.book;
```

Aggregate Queries: Condition

Find the students who borrowed more that one book on any given day

```
SELECT l.borrower
FROM loan l
GROUP BY l.borrowed, l.borrower
WHERE COUNT(l.book) >1
```

“Incorrect syntax near the keyword 'WHERE'.”

Aggregate Queries: Condition

Find the students who borrowed more that one book on any given day

```
SELECT l.borrower
FROM loan l
GROUP BY l.borrowed, l.borrower
HAVING COUNT(l.book) >1;
```

borrower
angjiayi1990@hotmail.com
angjiayi1990@hotmail.com
anniechapman1991@yahoo.com
anniechapman1991@yahoo.com
…

Aggregate Queries: Condition

Find the different students who borrowed more that one book on any given day

```
SELECT DISTINCT l.borrower
FROM loan l
GROUP BY l.borrowed, l.borrower
HAVING COUNT(l.book) >1;
```

borrower
angjiayi1990@hotmail.com
anniechapman1991@yahoo.com
anupamaanghan2010@yahoo.com
chewsoennam1989@msn.com
…

Nested Queries

Find the names of the students from whom anniechapman1991@yahoo.com borrowed a book and returned it after 2010-03-04

```
SELECT s.name
FROM student s
WHERE s.email = ANY (SELECT l.owner
  FROM loan l
  WHERE l.returned  > '2010-03-04'
  AND l.borrower = 'anniechapman1991@yahoo.com');
```

22 rows

Nested Queries

Find the names of the students from whom anniechapman1991@yahoo.com borrowed a book and returned it after 2010-03-04

```
SELECT s.name
FROM student s
WHERE email IN (SELECT l.owner
   FROM loan l
   WHERE l.returned  > '2010-03-04'
   AND l.borrower = 'anniechapman1991@yahoo.com');
```

22 rows

Nested Queries

Find the names of the students from whom anniechapman1991@yahoo.com borrowed a book and returned it after 2010-03-04

```
SELECT s.name
FROM loan l, student s
WHERE s.email=l.owner
   AND l.returned  > '2010-03-04'
   AND l.borrower = 'anniechapman1991@yahoo.com';
```

27 rows

Nested Queries

Find the different students from whom anniechapman1991@yahoo.com never borrowed

```
SELECT s.email
FROM student s
WHERE s.email NOT IN (SELECT l.owner
   FROM loan l
   WHERE l.borrower = 'anniechapman1991@yahoo.com');
```

72 rows

Nested Queries

Find the different students from whom anniechapman1991@yahoo.com never borrowed

```
SELECT s.email
FROM student s
WHERE s.email  <> ALL (SELECT l.owner
   FROM  loan l
   WHERE l.borrower ='anniechapman1991@yahoo.com');
```

72 rows

Nested Queries

Find the different students from whom anniechapman1991@yahoo.com never borrowed

```
SELECT s.email
FROM student s
WHERE NOT EXISTS (SELECT l.owner
   FROM loan l
   WHERE s.email = l.owner
   AND l.borrower = 'anniechapman1991@yahoo.com');
```

72 rows

Nested Queries (Scope)

An attribute can only be used within the SELECT and WHERE clauses of the query in which its relation is declared (FROM clause) and within sub-queries queries

Nested Queries

- There can be multiple nested queries and multiple levels of nested queries
- Nested queries can appear in the WHERE but also the HAVING clauses

Union

```
SELECT *
FROM student T
WHERE T.department=‘CS’
UNION
SELECT *
FROM student T
WHERE T.department=‘IS’
```

Find all the information about students in the computer science department or in the information systems department

Intersection

```
SELECT T1.email
FROM student T
WHERE T1.department=‘CS’
INTERSECT
SELECT T2.owner AS email
FROM copy T2
WHERE T2.book='978-0262033848’
```

Find the emails of students in the computer science department owning a book with ISBN14 ‘978-0262033848’

(Non-Symmetric) Difference

```
SELECT T1.email
FROM student T1
WHERE T1.department=‘CS’
MINUS
SELECT T2.owner AS email
FROM copy T2
WHERE T2.book='978-0262033848’
```

Find all the mails of students in the computer science department but those owning a book with ISBN14 ‘978-0262033848’

Join

```
SELECT T1.email
FROM student T1, copy T2
WHERE T1.department=‘CS’
AND T2.owner=T1.email
AND T2.book='978-0262033848’
```

Find all the mails of students in the computer science department owning a book with ISBN14 ‘978-0262033848’

Inner Join

```
SELECT T1.email
FROM student T1 INNER JOIN copy T2
ON T2.owner=T1.email
WHERE T1.department=‘CS’
AND T2.book='978-0262033848’
```

Find all the mails of students in the computer science department owning a book with ISBN14 ‘978-0262033848’

Left Outer Join

```
Select T1.name,T2.book
FROM student T1, copy T2
WHERE T1.email=T2.owner
UNION
SELECT T3.name, CAST(NULL AS CHAR(14)AS book
FROM student T3
WHERE NOT EXISTS (SELECT *
                  FROM copy T4
                  WHERE T3.email=T4.owner)
```

Find the names of the students and the titles of the books they own. If a student does not own any book, print a NULL value

Left Outer Join

```
Select DISTINCT T1.name,T2.book
FROM student T1 LEFT OUTER JOIN copy T2
ON T1.email=T2.owner
```

Find the names of the students and the titles of the books they own. If a student does not own any book, print a NULL value

Right Outer Join

```
Select DISTINCT T2.title, T1.owner
FROM copy T1 RIGHT OUTER JOIN book T2
ON T1.book=T2.ISBN14
```

Find the title of books and the emails of their owner. If a book does not have an owner , print a NULL value

Full Outer Join

```
Select DISTINCT T2.a, T1.c
FROM table1 T1 FULL OUTER JOIN table T2
ON T1.b=T2.b
```

Other Join

- (EQUI) JOIN
- NATURAL JOIN USING
- CROSS JOIN

Summary

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. ORDER BY
6. SELECT

Credits

The content of this lecture is based on chapter 5 of the book "Introduction to database Systems"
By
S. Bressan and B. Catania,
McGraw Hill publisher

Animated characters are animated using VocaliseTTS under license from Digital Curiosty

Clipart and media are licensed from Microsoft Office Online Clipart and Media