In the Lecture Series Introduction to Database Systems

SQL Advanced

Presented by Stéphane Bressan

SQL DML, Arithmetic

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

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

```
SELECT price * 1.5
FROM mgh
WHERE title LIKE '%Schaum s Outline%'
```

(No column name)
15
18

Aggregate Queries

Find the total number of (different) books

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

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

(No column name)
311

Aggregate Queries

Find the total number of titles

```
SELECT COUNT(title)
FROM book
```

(No column name)
311

```
SELECT COUNT(ALL title)
FROM book
```

Find the total number of different titles

(No column name)
301

```
SELECT COUNT(DISTINCT title)
FROM book
```

Aggregate Queries

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

Find the average price of a book from McGrawHill

```
SELECT AVG(price)
FROM mgh
```

(No column name)
15.66

Aggregate Queries

Find, for each day, the number of books borrowed

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

Why no zero?

(No column name)
2
1
1
2
...

Aggregate Queries

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

borrowed	(No column name)
2010-01-01	2
2010-01-02	1
2010-01-04	1
2010-02-23	2
…	

Aggregate Queries

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

borrower	borrowed	(No column name)
anniechapman1991@yahoo.com	2010-01-01	2
anniechapman1991@yahoo.com	2010-01-02	1
anniechapman1991@yahoo.com	2010-01-04	1
anniechapman1991@yahoo.com	2010-02-23	2
…		

Aggregate Queries

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

borrower	borrowed	(No column name)
anniechapman1991@yahoo.com	2010-01-01	2
anniechapman1991@yahoo.com	2010-01-02	1
anniechapman1991@yahoo.com	2010-01-04	1
anniechapman1991@yahoo.com	2010-02-23	2
…		

Aggregate Queries

Find for each student and for each day the number of books that the student borrowed on that day

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

borrower	borrowed	(No column name)
angjiayi1990@hotmail.com	2010-01-01	1
anniechapman1991@yahoo.com	2010-01-01	2
davidhall1992@yahoo.com	2010-01-01	1
dennispalmer1992@yahoo.com	2010-01-01	1
dingyang1989@gmail.com	2010-01-01	1
...		

Aggregate Queries

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

NOT INTERESTING!

Aggregate Queries

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

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

borrower
anniechapman1991@yahoo.com
liuyihui1990@hotmail.com
liuyiyang1992@msn.com
vargheseaneja1992@msn.com
xuhuajun1990@msn.com
anniechapman1991@yahoo.com
fengmeng1990@gmail.com
…

“Inco HERE'.”

Aggregate Queries

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

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

borrower
anniechapman1991@yahoo.com
liuyihui1990@hotmail.com
liuyiyang1992@msn.com
vargheseaneja1992@msn.com
xuhuajun1990@msn.com
anniechapman1991@yahoo.com
fengmeng1990@gmail.com
…

Aggregate Queries

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

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

borrower
anniechapman1991@yahoo.com
liuyihui1990@hotmail.com
liuyiyang1992@msn.com
vargheseaneja1992@msn.com
xuhuajun1990@msn.com
fengmeng1990@gmail.com
…

Nested Queries

Find the names of the students from whom the student anniechapman1991@yahoo.com borrowed a book

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

Nested Queries

Find the names of the students from whom the student anniechapman1991@yahoo.com borrowed a book

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

22 rows

Nested Queries

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

22 rows

Nested Queries

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

27 rows

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

Nested Queries

Find the different students from whomanniechapman1991@yahoo.com never borrowed

```
SELECT    DISTINCT email
FROM      loan, student
WHERE     NOT (email=owner
          AND returned  > '2010-03-04'
          AND borrower = 'anniechapman1991@yahoo.com')
```

100 rows

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

Nested Queries

```
SELECT      email
FROM        student
WHERE email  NOT IN (SELECT owner
            FROM  loan
            WHERE
            returned  > '2010-03-04'
            AND borrower = 'anniechapman1991@yahoo.com')
```

78 rows

Nested Queries

```
SELECT      email
FROM        student
WHERE email  <> ALL (SELECT owner
            FROM  loan
            WHERE
            returned  > '2010-03-04'
            AND borrower ='anniechapman1991@yahoo.com')
```

78 rows

Nested Queries

```
SELECT      email
FROM        student
WHERE NOT EXISTS (SELECT owner
            FROM  loan
            WHERE
            email = owner
            AND returned  > '2010-03-04'
            AND borrower = 'anniechapman1991@yahoo.com')
```

78 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 nested 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

Nested Queries

- Nested queries sometimes increase the readability of queries
- Nested queries increase the expressive power of SQL
- However nested queries can fool the optimizer

Algebraic Queries: Set Difference

```
SELECT      email
FROM        student
EXCEPT
SELECT owner
FROM  loan
WHERE
returned  > '2010-03-04'
AND borrower = 'anniechapman1991@yahoo.com'
```

78 rows

Algebraic Queries: Union and Intersection

```
SELECT   name
FROM     student
WHERE    year ='2009-08-01'
UNION
SELECT   name
FROM     student
WHERE    year='2009-01-01'
```

```
SELECT   name
FROM     student
WHERE    year ='2009-08-01'
INTERSECT
SELECT   name
FROM     student
WHERE    year='2009-01-01'
```

Try and find out what happens if there are duplicates

Algebraic Queries: Join

```
SELECT    book, name
FROM      loan INNER JOIN student
ON email=owner
```

```
SELECT    book, name
FROM      loan (LEFT | RIGHT | FULL) OUTER JOIN student
ON email=owner
```

(natural, cross)

Views

CREATE VIEW vmgh

AS SELECT ISBN13, title, authors

FROM book

WHERE publisher='McGraw-Hill'

SELECT * FROM vmgh

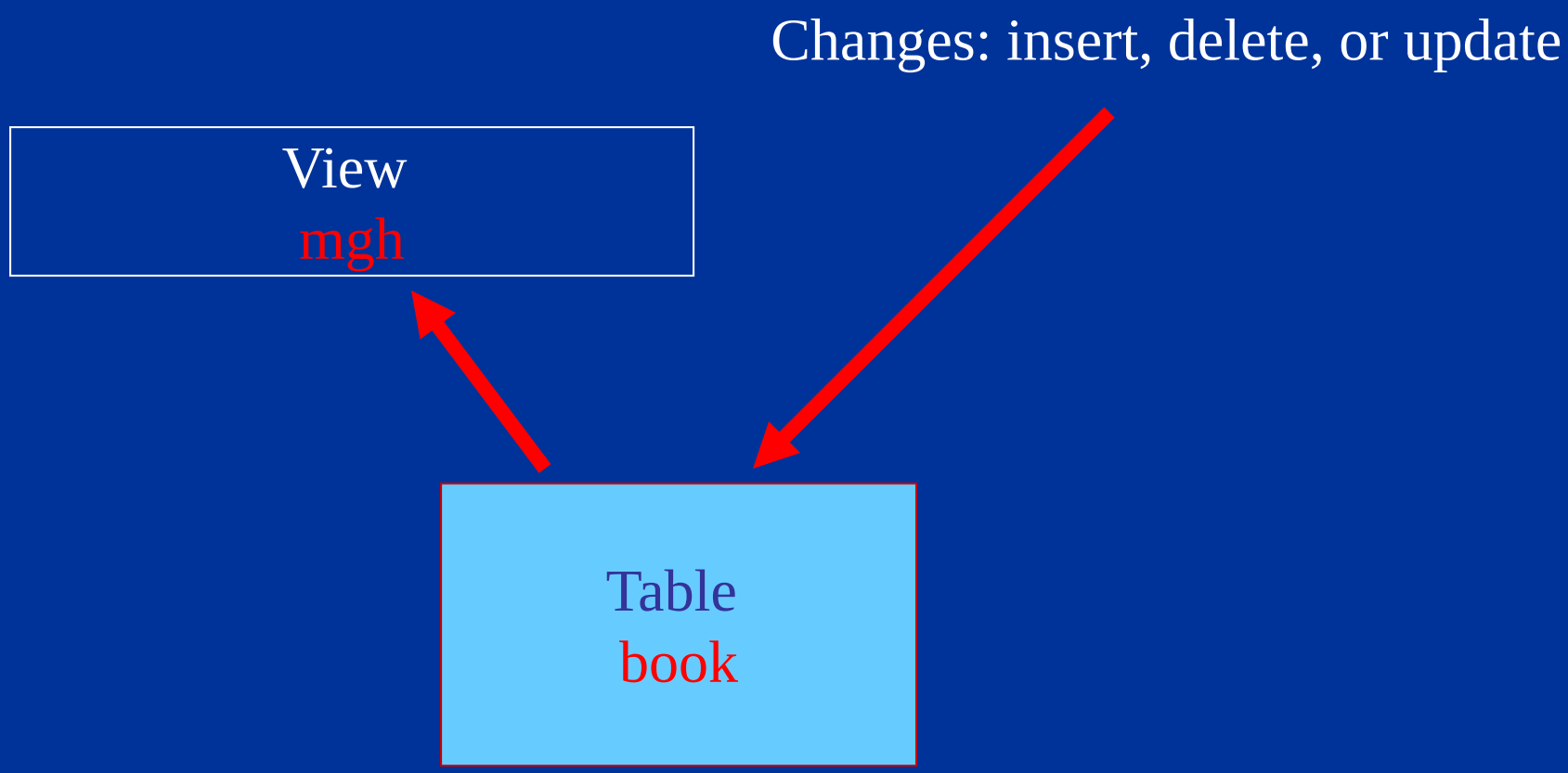
ISBN13	title	authors
978-0071508612	Schaum s Outline of Calculus	Frank Ayres, Elliott Mendelson
978-0071635264	Schaum s Outline of Chinese Grammar	Claudia Ross
978-0071639309	Practice Makes Perfect Spanish Verb Tenses	Dorothy Richmond

Views

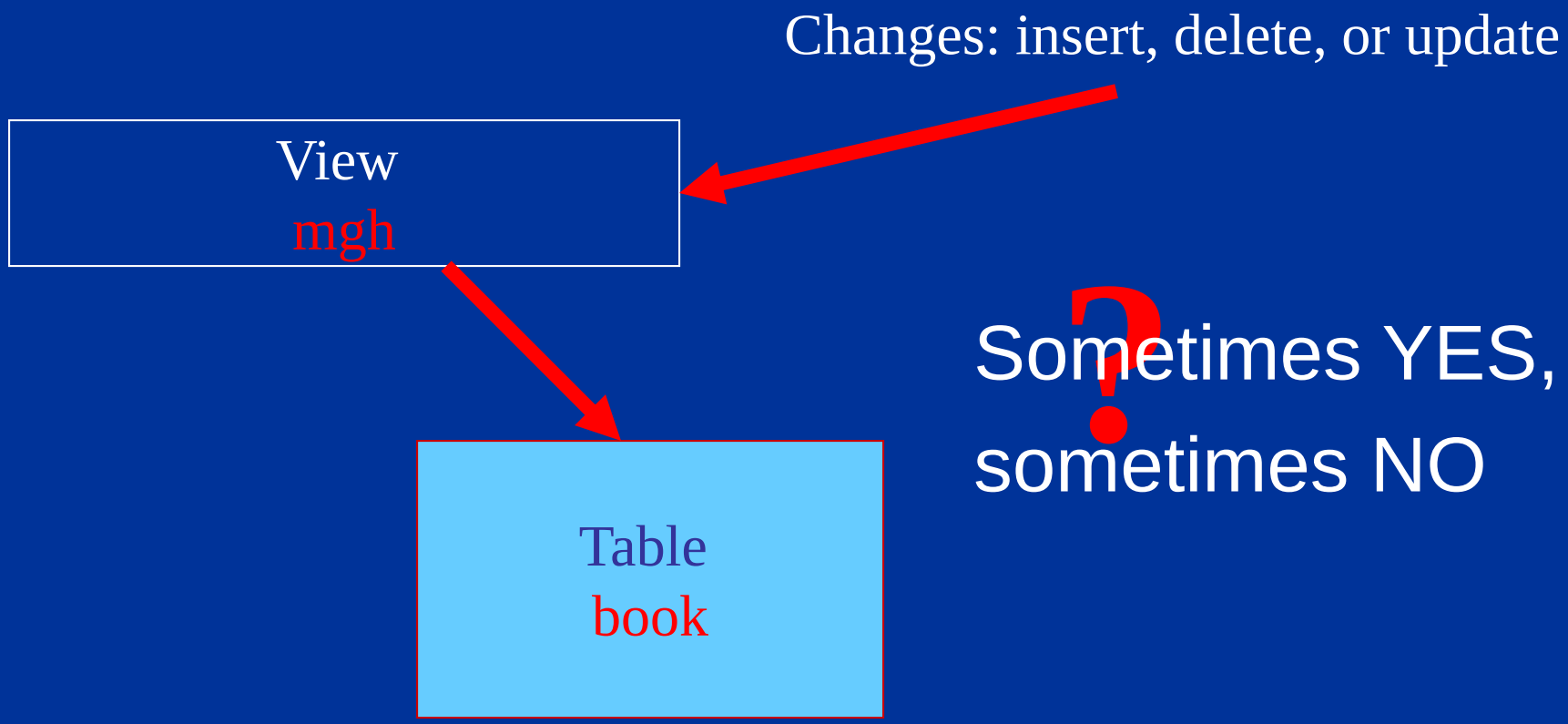
```
CREATE VIEW name [schema]
AS sql_query
```

- A view is a query with a name
- A view can be used exactly as a table
- The contents of the view is always up-to-date

Changes in the base tables are propagated



Can we update the views?



Views

```
insert into vmgh VALUES ('978-0077338497',
        'Readings in the Western Humanities Volume 2',
        'Roy Matthews, Dewitt Platt')
```

"Cannot insert the value NULL into column 'ISBN10', table 'tutorials.dbo.book'; column does not allow nulls. INSERT fails."

Can we update the views?

```
CREATE VIEW bb (borrower, borrowed, quantity)
AS
SELECT    borrower, borrowed, COUNT(book)
FROM      loan
GROUP BY borrowed, borrower
```

There is no deterministic way to propagate changes of COUNT(book) to book!

Views

- Logical Data Independence is achieved by means of views
- Views can be pre-compiled
- However views may fool the optimizer

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