Structured Query Language

Seminar Session

(Monday, Sept. 7, 2015)

General Overview: Relational Model

- Formal Query Languages
  - Relational Algebra, Relational Calculus
- Commercial Query Languages
  - **SQL: Structured Query Language**
    - DML: Data Manipulation Language
      - Select, From, Where
      - Set Operations
      - Ordering, Aggregate Functions
      - Nested Subqueries
    - DDL: Data Definition Language
  - QBE (QUEL): Query By Example

Today

AGENDA

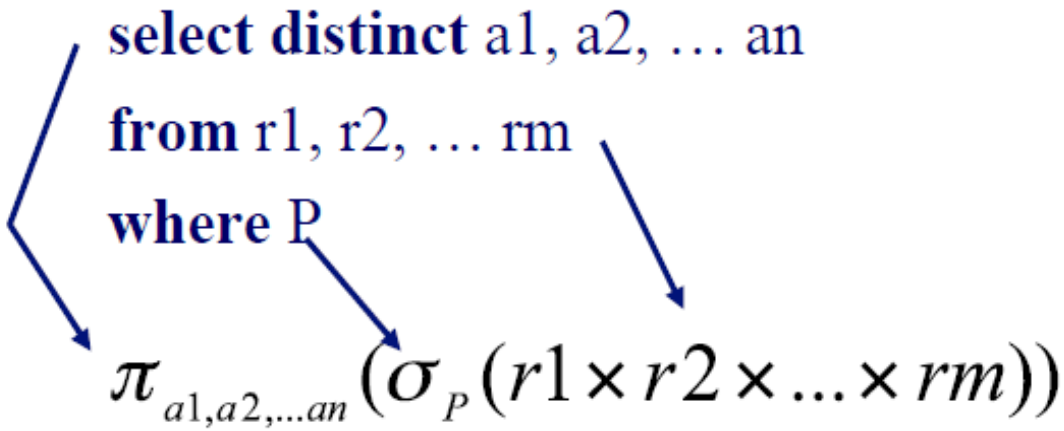
- Relational Algebra to SQL Statement
  - Basic Operations and clauses, e.g., Selection, Projection, Join etc.
- Query to SQL Statement
  - Special operations and clauses, e.g., group by, having, all
  - Other Data Manipulation Language (DML) Queries
  - Views

Data Manipulation Language

- SELECT Query and Relational Algebra

```
SELECT a1, a2, … an
FROM r1, r2, … rm
WHERE P
[order by ….]
[group by …]
[having …]
```

General Form



Mapping to Relational Algebra

Exercise 1: Select and Project

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which are the last names of the readers in Kazan?

  $\Pi_{Surname}(\sigma_{City=Kazan}(READER))$

- SQL Statement?

  1) SELECT Surname FROM Reader WHERE city='Kazan'
  2) SELECT DISTINCT Surname FROM Reader WHERE City = 'Kazan'

What if we have multiple Readers with the same Surname in Kazan?

Exercise 2: Select with JOIN

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which books (Author, Title) are from publishers in Moscow, London or New York?

  $\Pi_{Author,Title}(BOOK \bowtie (\sigma_{City=Moscow\ OR\ City=London\ OR\ City=NewYork}(PUBLISHER)))$

- SQL Statement?

  What if we replace OR operator with AND in this query?

```
Select Author, Title
from Book b, PUBLISHER p
where b.PublisherName=p.PublisherName AND
(p.city=Moscow OR p.city=London OR p.city=NewYork)
```

Exercise 3: Multiple JOINs

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which books (Author, Title) has the reader Jooyoung Lee borrowed?

  $\Pi_{Author,Title}(BOOK \bowtie LOAN \bowtie_{ReaderNr=RDNR} (\sigma_{Surname=Lee\ AND\ Firstname=Jooyoung}(READER)))$

- SQL Statement?

```
SELECT B.Author, B.Title FROM Reader R, Loan L, Book B
WHERE R.Surname = 'Lee' AND R.Firstname = 'Jooyoung'
AND R.RDNR = L.ReaderNr AND L.ISBN = B.ISBN
```

Exercise 3: Multiple JOINs…

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which readers (Surname, Firstname) have borrowed books that were published in their home town?

  $\Pi_{Firstname,Surname}(\sigma_{City=PublisherCity}$ (PUBLISHER ⋈ BOOK ⋈ LOAN ⋈ ReaderNr=RDNR READER))

- SQL Statement?

  SELECT R.Firstname, R.Surname FROM Reader R, Loans L, Book B, Publisher P WHERE R.RDNR = L.ReaderNr AND L.ISBN = B.ISBN AND B.PublisherName = P.PublisherName AND R.City = P.PublisherCity

Exercise 4: Outer JOIN

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- List all the publishers and their respective books.
- SQL Statement?

```
SELECT PublisherName, Title
FROM Book B
RIGHT OUTER JOIN Publisher P ON B.PublisherName =
P.PublisherName
( ORDER BY B.PublisherName ASC, Title ASC);
```

Exercise 5: Set Operation

- Consider the following relational schema:

TAKES (SSN, c-id, grade)

TAKES		
SSN	**c-id**	**grade**
123	15-413	A
234	15-413	B

- Find ssn of people taking both 15-415 and 15-413?

```
select ssn
from takes
where c-id="15-415" and
c-id="15-413"
```

```
(select ssn from takes where c-id="15-415" )
intersect
(select ssn from takes where c-id="15-413" )
```

Can we use DISTINCT here?

Exercise 5: Set Operation…

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which books in the category “Alps” do not belong to the category “Switzerland”?

  $(\Pi_{ISBN}\ (\sigma_{CategoryName=Alps}(BookCategory)))- (\Pi_{ISBN}\ (\sigma_{CategoryName=Switzerland}(BookCategory)))$

- SQL Statement?

```
(SELECT ISBN FROM BookCategory WHERE CategoryName = 'Alps') EXCEPT
(SELECT ISBN FROM BookCategory WHERE CategoryName = 'Switzerland')
```

Exercise 6: Aggregate Functions

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which book has the maximum number of pages?
- SQL Statement?

```
SELECT Title
FROM Book
WHERE  NoPages =
( SELECT MAX(NoPages) FROM Book);
```

```
SELECT Title
FROM Book
WHERE  NoPages IN
( SELECT MAX(NoPages) FROM Book);
```

Exercise 6: Aggregate Functions…

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which book has more pages than twice the average of the number of pages of all books?
- SQL Statement?

```
SELECT Title
FROM Book
WHERE NoPages >= 2* (SELECT AVG(NoPages) FROM Book);
```

Exercise 7: Grouping

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which authors have written more than 5 books?
- SQL Statement?

```
SELECT Author, COUNT(Title) AS number_books
FROM Book
GROUP BY Author
HAVING number_books > 5;
```

Exercise 7: Grouping…

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which author has written more books?
- SQL Statement?

```
SELECT Author, COUNT(Title) AS number_books
FROM Book
GROUP BY Author HAVING number books >= ALL
        (SELECT COUNT(Title) FROM Book GROUP BY Author);
```

Exercise 8: Ordering

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which are the ten oldest books?
- SQL Statement?

```
SELECT ISBN, Author, Title
FROM Book
ORDER BY PubYear
TOP 10 –– (SQL Server)
LIMIT 10 –– (MySQL, PostgreSQL)
FETCH FIRST 10 ROWS –– (DB2)
```

Unfortunately, there is no single syntax.

Exercise 9: Recursion

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Which readers (Surname, Firstname) have borrowed at least a book that has been borrowed also by the reader Sadegh Nobari ?
  - Note: the reader Sadegh Nobari should not be included in the results?
- SQL Statement?

Exercise 9: Recursion…

- Which readers (Surname, Firstname) have borrowed at least a book that has been borrowed also by the reader Sadegh Nobari ?
  - Note: the reader Sadegh Nobari should not be included in the results?
- SQL Statement?

```
SELECT R1.Firstname, R1.Surname
FROM Reader R1, Loan L1, Loan L2, Reader R2
WHERE R2.Firstname='Sadegh'
AND R2.Surname = 'Nobari'
AND L2.ReaderNr = R2.RDNR
AND L1.ISBN = L2.ISBN
AND R1.RDNR = L1.ReaderNr
AND R1.RDNR <> R2.RDNR
```

Data Manipulation Language

- INSERT Query

```
INSERT INTO table
(column1, column2, ... )
VALUES
(expression1, expression2, ... )
```

- UPDATE Query

```
UPDATE table_name
SET column1=value1,column2=value2,...
WHERE some_column=some_value;
```

Exercise 9: DML Queries

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- The reader Qiang Qu borrows the copy with CopyNumber 4 of the book with ISBN 123456.
- SQL Statement?

```
INSERT INTO Loan (ReaderNr, ISBN, Copy)
SELECT RDNR, 123456, 4
FROM Reader
WHERE Firstname = 'Qiang' AND Surname = 'Qu'
```

Exercise 9: DML Queries...

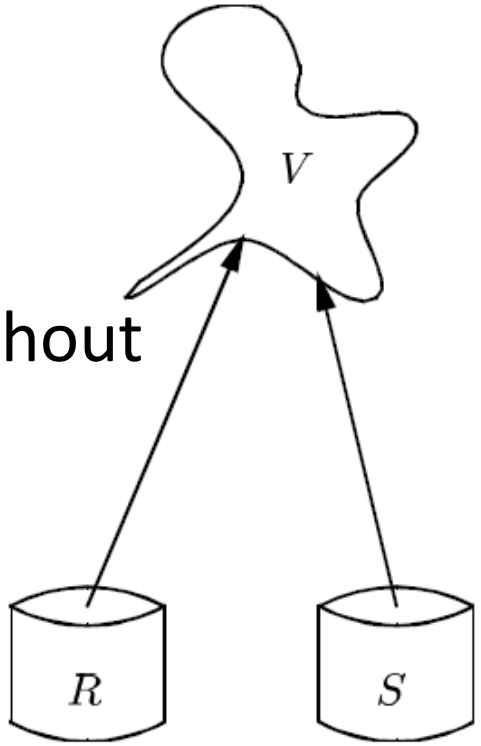
- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthdate)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Change the return date of all the books in the category “Databases” that should be returned before 15.03.2013 so that they can be kept for 30 days longer (Assume that you can add days to dates in SQL).

```
UPDATE Loan SET ReturnDate = ReturnDate + 30 WHERE
ReturnDate < ’2013−03−15’ AND
ISBN IN (SELECT ISBNFROM BookCategory
WHERE CategoryName = ’Databases’)
```

Views



- An expression that describes a table without creating it.
- View definition form is:

```
CREATE VIEW <name> AS <query>;
```

- Example
  - The view CanDrink is the set of drinker-beer pairs such that the drinker frequents at least one bar that serves the beer.

```
CREATE VIEW CanDrink AS
SELECT drinker, beer
FROM Frequents, Sells
WHERE Frequents.bar = Sells.bar;
```

```
SELECT beer
FROM CanDrink
WHERE drinker='Sally';
```

Assignment

- Consider the following relational schema:

Customers (CustID, LastName, FirstName)
Inventory (TapeID, MovieID)
Movies (MovieID, MovieName)
MovieSupplier (SupplierID,MovieID, Price)
Orders (OrderID, SupplierID, MovieID, Copies)
Rentals (CustomerID, TapeID, CkoutDate, Duration)
Suppliers (SupplierID, SupplierName)

- Write SQL statements for the following queries:
  1. Which movies are not supplied by “PIXAR" or “LOINSGATE"?
  2. Which movie was rented for the longest duration (by any customer)?

Assignment…

- Write SQL statements for the following queries:
  3. Which suppliers supply all the movies in the inventory?
  4. How many movies in the inventory does each movie supplier supply?
  5. For which movies have more than 4 copies been ordered?
  6. Which customers rented "Fatal Attraction 1987" or rented a movie supplied by "VWS Video"?

Assignment…

- Write SQL statements for the following queries:
  7. For which movies are there more than 1 copy in our inventory?
  8. Which customers rented movies for 5 days or more?
  9. Which supplier has the cheapest price for the movie "Almost Angels 1962"?
  10. Which movies aren't in the inventory?

Important Note: Configure PostgreSQL on your machine and practice this assignment along with latex submission.

You can populate tables with few random records.