

Database Systems

Relational Algebra

H. Turgut Uyar Şule Öğüdücü

2002-2010

1 / 115

License



©2002-2010 T. Uyar, Ş. Öğüdücü

You are free:

- ▶ to Share — to copy, distribute and transmit the work
- ▶ to Remix — to adapt the work

Under the following conditions:

- ▶ Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- ▶ Noncommercial — You may not use this work for commercial purposes.
- ▶ Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Legal code (the full license):

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

2 / 115

Topics

Relational Algebra

Introduction
Selection
Joining
Set Operations

Queries

Introduction
Joining
Subqueries
Set Operations

3 / 115

Closure

Definition

closure:

the input and output of all operations are relations

- ▶ the output of one operation can be the input of another operation
- ▶ operations can be nested

4 / 115

Leap

- ▶ operator first, operands later:
operator (operand₁) (operand₂) ...
- ▶ data types:
 - ▶ string, integer, boolean
 - ▶ no real numbers
- ▶ dotted notation to distinguish attributes with identical names

5 / 115

Data Definition

Creating Relations

```
relation (relation_name) (  
    (attribute_name , type , length)  
    [ , ... ]  
)
```

Deleting Relations

```
delrel (relation_name)
```

6 / 115

Relation Creation Example

Example (creating the movie relation)

```
relation (movie) (  
  (id , integer , 4) ,  
  (title , string , 80) ,  
  (yr , integer , 4) ,  
  (score , integer , 3) ,  
  (votes , integer , 5) ,  
  (directorid , integer , 4)  
)
```

7 / 115

Data Manipulation

Adding Tuples

```
add (relation_name) (value [, ...])
```

- ▶ values must be given in the order as they were defined
- ▶ no quotes around values

Deleting Tuples

```
delete (relation_name) (condition)
```

- ▶ all values in quotes when writing conditions

8 / 115

Data Manipulation Examples

Example (adding a movie)

```
add (movie) (  
  6 ,  
  Usual Suspects ,  
  1995 ,  
  87 ,  
  35027 ,  
  639  
)
```

Example (deleting movies)

```
delete (movie)  
  (score < '30')
```

9 / 115

Example Relations

Example (MOVIE Relation)

ID	TITLE	YR	SCORE	VOTES	DIRECTORID
6	Usual Suspects	1995	87	35027	639
70	Being John Malkovich	1999	83	13809	1485
107	Batman & Robin	1997	35	10577	105
110	Sleepy Hollow	1999	75	10514	148
112	Three Kings	1999	77	10319	0
151	Gattaca	1997	74	8388	2020
213	Blade	1998	67	6885	0
228	Ed Wood	1994	78	6587	148
251	End of Days	1999	55	6095	103
281	Dangerous Liaisons	1988	77	5651	292
373	Fear and Loathing in Las Vegas	1998	65	4658	59
432	Stigmata	1999	61	4141	0
433	eXistenZ	1999	69	4130	97
573	Dead Man	1995	74	3333	175
1468	Europa	1991	76	1042	615
1512	Suspiria	1977	71	1004	2259
1539	Cry-Baby	1990	59	972	364

10 / 115

Example Relations

Example (PERSON Relation)

ID	NAME
9	Arnold Schwarzenegger
26	Johnny Depp
59	Terry Gilliam
97	David Cronenberg
103	Peter Hyams
105	Joel Schumacher
138	George Clooney
148	Tim Burton
175	Jim Jarmusch
187	Christina Ricci
243	Uma Thurman
282	Cameron Diaz
292	Stephen Frears
302	Benicio Del Toro

308	Gabriel Byrne
350	Jennifer Jason Leigh
364	John Waters
406	Patricia Arquette
503	John Malkovich
615	Lars von Trier
639	Bryan Singer
745	Udo Kier
793	Jude Law
1485	Spike Jonze
1641	Iggy Pop
2020	Andrew Niccol
2259	Dario Argento
3578	Traci Lords

11 / 115

Example Relations

Example (CASTING Relation)

MOVIEID	ACTORID	ORD
6	308	2
6	302	3
70	282	2
70	503	14
107	9	1
107	138	2
107	243	4
110	26	1
110	187	2
112	138	1
112	1485	4
151	243	2

151	793	3
213	745	6
213	3578	8
228	26	1
228	406	4
251	9	1
251	308	2
251	745	10
281	243	7
281	503	2
373	26	1
373	187	6
373	282	8

373	302	2
432	308	2
432	406	1
433	350	1
433	793	2
573	26	1
573	308	12
573	1641	6
1468	745	3
1512	745	9
1539	26	1
1539	1641	5
1539	3578	7

12 / 115

Selection

Definition

selection:

creating a new relation from all the tuples that satisfy a condition

Statement

select (relation_name) (condition)

- ▶ also known as: *restrict*
- ▶ output relation header = input relation header

13 / 115

Selection Examples - 1

Example (movies with more than 10000 votes)

s1 = **select** (movie) (votes > '10000')

ID	TITLE	YR	SCORE	VOTES	DIRECTORID
6	Usual Suspects	1995	87	35027	639
70	Being John Malkovich	1999	83	13809	1485
107	Batman & Robin	1997	35	10577	105
110	Sleepy Hollow	1999	75	10514	148
112	Three Kings	1999	77	10319	0

14 / 115

Selection Examples - 2

Example (movies older than 1992, with scores higher than 75)

s2 = **select** (movie)
((yr < '1992') **and** (score > '75'))

ID	TITLE	YR	SCORE	VOTES	DIRECTORID
281	Dangerous Liaisons	1988	77	5651	292
1468	Europa	1991	76	1042	615

15 / 115

Projection

Definition

projection:

creating a new relation using the given attributes

Statement

project (relation_name) (attribute [, ...])

- ▶ output relation header = attribute list

16 / 115

Projection Examples - 1

Example (titles of all movies)

p1 = **project** (movie) (title)

TITLE
Usual Suspects
Being John Malkovich
Batman & Robin
Sleepy Hollow
Three Kings
Gattaca
Blade
Ed Wood
End of Days

Dangerous Liaisons
Fear and Loathing in Las Vegas
Stigmata
eXistenZ
Dead Man
Europa
Suspiria
Cry-Baby

17 / 115

Projection Examples - 2

Example (titles and years of all movies)

p2 = **project** (movie) (title , yr)

TITLE	YR
Usual Suspects	1995
Being John Malkovich	1999
Batman & Robin	1997
Sleepy Hollow	1999
Three Kings	1999
Gattaca	1997
Blade	1998
Ed Wood	1994
End of Days	1999

Dangerous Liaisons	1988
Fear and Loathing in Las Vegas	1998
Stigmata	1999
eXistenZ	1999
Dead Man	1995
Europa	1991
Suspiria	1977
Cry-Baby	1990

18 / 115

Projection Examples - 3

Example (years of all movies)

p3 = **project** (movie) (yr)

YR
1995
1999
1997
1998
1994
1988
1991
1977
1990

19 / 115

Projection Examples - 4

Example (titles of all movies with votes more than 5000 and scores higher than 70)

1. all movies with votes more than 5000 and scores higher than 70
2. titles of all movies with votes more than 5000 and scores higher than 70

20 / 115

Projection Examples - 4

Example (all movies with votes more than 5000 and scores higher than 70)

p4a = **select** (movie)
((votes > '5000') **and** (score > '70'))

ID	TITLE	YR	SCORE	VOTES	DIRECTORID
6	Usual Suspects	1995	87	35027	639
70	Being John Malkovich	1999	83	13809	1485
110	Sleepy Hollow	1999	75	10514	148
112	Three Kings	1999	77	10319	0
151	Gattaca	1997	74	8388	2020
228	Ed Wood	1994	78	6587	148
281	Dangerous Liaisons	1988	77	5651	292

21 / 115

Projection Examples - 4

Example (titles of all movies with votes more than 5000 and scores higher than 70)

p4 = **project** (p4a) (title)

TITLE
Usual Suspects
Being John Malkovich
Sleepy Hollow
Three Kings
Gattaca
Ed Wood
Dangerous Liaisons

22 / 115

Projection Examples - 4

Example (titles of all movies with votes more than 5000 and scores higher than 70)

p4 = **project** (**select** (movie)
((votes > '5000') **and** (score > '70'))
(title))

23 / 115

Product

Definition

product: relation₁ × relation₂
Cartesian product of two relations

Statement

product (relation₁) (relation₂)

- output relation header = relation₁ header + relation₂ header

24 / 115

Product Example

Example ($s_2 \times p_4$)

$x_1 = \text{product} (s_2) (p_4)$

ID	TITLE	YR	SCORE	VOTES	DIRECTORID	p4.TITLE
281	Dangerous Liaisons	1988	77	5651	292	Usual Suspects
281	Dangerous Liaisons	1988	77	5651	292	Being John Malkovich
281	Dangerous Liaisons	1988	77	5651	292	Sleepy Hollow
281	Dangerous Liaisons	1988	77	5651	292	Three Kings
281	Dangerous Liaisons	1988	77	5651	292	Gattaca
281	Dangerous Liaisons	1988	77	5651	292	Ed Wood
281	Dangerous Liaisons	1988	77	5651	292	Dangerous Liaisons
1468	Europa	1991	76	1042	615	Usual Suspects
1468	Europa	1991	76	1042	615	Being John Malkovich
1468	Europa	1991	76	1042	615	Sleepy Hollow
1468	Europa	1991	76	1042	615	Three Kings
1468	Europa	1991	76	1042	615	Gattaca
1468	Europa	1991	76	1042	615	Ed Wood
1468	Europa	1991	76	1042	615	Dangerous Liaisons

25 / 115

Joining

Definition

joining:

creating a new relation by matching tuples of two relations over common values of one or more attributes

Statement

join (relation_1) (relation_2) (condition)

- ▶ **select** (product (relation_1) (relation_2)) (condition)
- ▶ output relation header = relation₁ header + relation₂ header

26 / 115

Join Examples - 1

Example (titles of all movies and the names of their directors)

1. all movies and their directors
2. titles of all movies and the names of their directors

27 / 115

Join Examples - 1

Example (all movies and their directors)

$j1a = \text{join} (\text{movie}) (\text{person})$
(movie.directorid=person.id)

ID	TITLE	...	DIRECTORID	PERSON.ID	NAME
6	Usual Suspects	...	639	639	Bryan Singer
70	Being John Malkovich	...	1485	1485	Spike Jonze
107	Batman & Robin	...	105	105	Joel Schumacher
110	Sleepy Hollow	...	148	148	Tim Burton
151	Gattaca	...	2020	2020	Andrew Niccol
...
433	eXistenZ	...	97	97	David Cronenberg
573	Dead Man	...	175	175	Jim Jarmusch
1468	Europa	...	615	615	Lars von Trier
1512	Suspiria	...	2259	2259	Dario Argento
1539	Cry-Baby	...	364	364	John Waters

28 / 115

Join Examples - 1

Example (titles of all movies and the names of their directors)

$j1 = \text{project} (j1a) (\text{title}, \text{name})$

TITLE	NAME
Usual Suspects	Bryan Singer
Being John Malkovich	Spike Jonze
Batman & Robin	Joel Schumacher
Sleepy Hollow	Tim Burton
Gattaca	Andrew Niccol
...	...
eXistenZ	David Cronenberg
Dead Man	Jim Jarmusch
Europa	Lars von Trier
Suspiria	Dario Argento
Cry-Baby	John Waters

29 / 115

Join Examples - 2

Example (titles of all movies and the names of their actors along with their ordinal numbers)

1. all movies and the identities of their actors
2. all movies and their actors
3. titles of all movies and the names of their actors along with their ordinal numbers

30 / 115

Join Examples - 2

Example (all movies and the identities of their actors)

```
j2a = join (movie) (casting)
      (movie.id=casting.movieid)
```

ID	TITLE	...	MOVIEID	ACTORID	ORD
6	Usual Suspects	...	6	308	2
6	Usual Suspects	...	6	302	3
70	Being John Malkovich	...	70	282	2
70	Being John Malkovich	...	70	503	14
...
1512	Suspiria	...	1512	745	9
1539	Cry-Baby	...	1539	26	1
1539	Cry-Baby	...	1539	1641	5
1539	Cry-Baby	...	1539	3578	7

31 / 115

Join Examples - 2

Example (all movies and their actors)

```
j2b = join (j2a) (person)
      (j2a.actorid=person.id)
```

ID	TITLE	...	ACTORID	ORD	PERSON.ID	NAME
6	Usual Suspects	...	308	2	308	Gabriel Byrne
6	Usual Suspects	...	302	3	302	Benicio Del Toro
70	Being John Malkovich	...	282	2	282	Cameron Diaz
70	Being John Malkovich	...	503	14	503	John Malkovich
...
1512	Suspiria	...	745	9	745	Udo Kier
1539	Cry-Baby	...	26	1	26	Johnny Depp
1539	Cry-Baby	...	1641	5	1641	Iggy Pop
1539	Cry-Baby	...	3578	7	3578	Traci Lords

32 / 115

Joining Examples - 2

Example (titles of all movies and the names of their actors along with their ordinal numbers)

```
j2 = project (j2b) (title, name, ord)
```

TITLE	NAME	ORD
Usual Suspects	Gabriel Byrne	2
Usual Suspects	Benicio Del Toro	3
Being John Malkovich	Cameron Diaz	2
Being John Malkovich	John Malkovich	14
...
Suspiria	Udo Kier	9
Cry-Baby	Johnny Depp	1
Cry-Baby	Iggy Pop	5
Cry-Baby	Traci Lords	7

33 / 115

Join Examples - 3

Example (names of actors who played with Johnny Depp)

1. Johnny Depp's identity
2. identities of movies Johnny Depp was in
3. identities of actors who were in the movies Johnny Depp was in
4. names of actors who played with Johnny Depp

34 / 115

Join Examples - 3

Example (Johnny Depp's identity)

```
j3a = project (select (person)
                    (name='Johnny Depp'))
            (id)
```

ID
26

35 / 115

Join Examples - 3

Example (identities of movies Johnny Depp was in)

```
j3b = project (join (j3a) (casting)
               (j3a.id=casting.actorid))
            (movieid)
```

MOVIEID
110
228
373
573
1539

36 / 115

Join Examples - 3

Example (identities of actors who were in the movies Johnny Depp was in)

```
j3c = project (join (j3b) (casting)
                (j3b.movieid=casting.movieid))
                (actorid)
```

ACTORID
26
187
406
282
302
308
1641
3578

37 / 115

Join Examples - 3

Example (names of actors who played with Johnny Depp)

```
j3 = project (join (j3c) (person)
                 (j3c.actorid=person.id))
                 (name)
```

NAME
Johnny Depp
Christina Ricci
Patricia Arquette
Cameron Diaz
Benicio Del Toro
Gabriel Byrne
Iggy Pop
Traci Lords

38 / 115

Natural Join

Definition

natural join:

joining two relations over the attributes which have the same name

Statement

```
natjoin (relation_1) (relation_2)
```

- ▶ output relation header = relation₁ header \cup relation₂ header

39 / 115

Natural Join

- ▶ in Leap, a condition must be predefined for a natural join

```
add (relship) (frelation , prelation ,
               fkey1 , fkey2 , fkey3 ,
               pkey1 , pkey2 , pkey3)
```

Example (movie.directorid \rightarrow person.id)

```
add (relship) (movie , person ,
               directorid , -, -,
               id , -, -)
```

40 / 115

Natural Join Example

Example (all movies and their directors)

```
n1 = natjoin (movie) (person)
```

ID	TITLE	...	DIRECTORID PERSON.ID	NAME
6	Usual Suspects	...	639	Bryan Singer
70	Being John Malkovich	...	1485	Spike Jonze
107	Batman & Robin	...	105	Joel Schumacher
110	Sleepy Hollow	...	148	Tim Burton
151	Gattaca	...	2020	Andrew Niccol
228	Ed Wood	...	148	Tim Burton
251	End of Days	...	103	Peter Hyams
281	Dangerous Liaisons	...	292	Stephen Frears
373	Fear and Loathing in Las Vegas	...	59	Terry Gilliam
433	eXistenZ	...	97	David Cronenberg
573	Dead Man	...	175	Jim Jarmusch
1468	Europa	...	615	Lars von Trier
1512	Suspiria	...	2259	Dario Argento
1539	Cry-Baby	...	364	John Waters

41 / 115

Division

Definition

division: relation₁ / relation₂

creating a new relation from the tuples of the first relation that contain all the tuples in the second relation

Statement

```
divide (relation_1) (relation_2)
```

- ▶ relation₂ header \subseteq relation₁ header
- ▶ output relation header = relation₁ header - relation₂ header

42 / 115

Division Example

Example (titles of movies in which Johnny Depp and Christina Ricci both played)

1. identities of Johnny Depp and Christina Ricci
2. identities of all movies and actors
3. identities of movies in which Johnny Depp and Christina Ricci both played
4. titles of movies in which Johnny Depp and Christina Ricci both played

43 / 115

Division Example

Example (identities of Johnny Depp and Christina Ricci)

```
v1a = project (select (person)
  ((name='Johnny Depp')
   or (name='Christina Ricci'))
  (id))
```

ID
26
187

44 / 115

Division Example

Example (identities of all movies and actors)

```
v1b = project (casting) (movieid, actorid)
```

MOVIEID	ACTORID
6	308
6	302
70	282
70	503
...	...
110	26
110	187
...	...
228	26
...	...

...	...
373	26
373	187
...	...
573	26
...	...
1512	745
1539	26
1539	1641
1539	3578

45 / 115

Division Example

Example (identities of movies in which Johnny Depp and Christina Ricci both played)

```
v1c = project (divide (v1b) (v1a)) (movieid)
```

MOVIEID
110
373

46 / 115

Division Example

Example (titles of movies in which Johnny Depp and Christina Ricci both played)

```
v1 = project (join (v1c) (movie)
  (v1c.movieid=movie.id))
  (title)
```

TITLE
Sleepy Hollow
Fear and Loathing in Las Vegas

47 / 115

Division Example

Example (product - division relationship)

```
product (v1c) (v1a)
```

MOVIEID	ACTORID
110	26
110	187
373	26
373	187

48 / 115

Intersection

Definition

intersection: $\text{relation}_1 \cap \text{relation}_2$

creating a new relation from the tuples found in both relations

Statement

```
intersect (relation_1) (relation_2)
```

► output relation header = relation₁ header = relation₂ header

49 / 115

Intersect Example

Example (names of all directors who also acted)

1. identities of all directors
2. identities of all actors
3. identities of all directors who also acted
4. names of all directors who also acted

50 / 115

Intersection Example

Example (identities of all directors)

```
i1a = project (movie) (directorid)
```

DIRECTORID
639
1485
105
148
0
2020
103
292
59
97
175
615
2259
364

51 / 115

Intersect Example

Example (identities of all actors)

```
i1b = project (casting) (actorid)
```

ACTORID
308
302
282
503
9
138
243
26

187
1485
793
745
3578
406
350
1641

52 / 115

Intersect Example

Example (identities of all directors who also acted)

```
i1c = intersect (i1a) (i1b)
```

ACTORID
1485

53 / 115

Intersect Example

Example (names of all directors who also acted)

```
i1 = project (join (i1c) (person)  
              (actorid=person.id))  
      (name)
```

NAME
Spike Jonze

54 / 115

Union

Definition

union: $\text{relation}_1 \cup \text{relation}_2$

creating a new relation from the tuples which are found in at least one of the relations

Statement

union (relation_1) (relation_2)

- output relation header = relation_1 header = relation_2 header

55 / 115

Union Example

Example (names of directors and actors of all movies newer than 1997)

1. all movies newer than 1997
2. all movies newer than 1997 and their actors
3. identities of directors of all movies newer than 1997
4. identities of actors of all movies newer than 1997
5. identities of directors and actors of all movies newer than 1997
6. names of directors and actors of all movies newer than 1997

56 / 115

Union Example

Example (all movies newer than 1997)

$u1a = \text{select (movie) (yr > '1997')}$

ID	TITLE	YR	SCORE	VOTES	DIRECTORID
70	Being John Malkovich	1999	83	13809	1485
110	Sleepy Hollow	1999	75	10514	148
112	Three Kings	1999	77	10319	0
213	Blade	1998	67	6885	0
251	End of Days	1999	55	6095	103
373	Fear and Loathing in Las Vegas	1998	65	4658	59
432	Stigmata	1999	61	4141	0
433	eXistenZ	1999	69	4130	97

57 / 115

Union Example

Example (all movies newer than 1997 and their actors)

$u1b = \text{join (u1a) (casting)}$
 $(u1a.id = \text{casting.movieid})$

ID	TITLE	...	DIRECTORID	MOVIEID	ACTORID	ORD
70	Being John Malkovich	...	1485	70	282	2
70	Being John Malkovich	...	1485	70	503	14
110	Sleepy Hollow	...	148	110	26	1
110	Sleepy Hollow	...	148	110	187	2
112	Three Kings	...	0	112	138	1
...
373	Fear and Loathing in Las Vegas	...	59	373	302	2
432	Stigmata	...	0	432	308	2
432	Stigmata	...	0	432	406	1
433	eXistenZ	...	97	433	350	1
433	eXistenZ	...	97	433	793	2

58 / 115

Union Example

Example (identities of directors of all movies newer than 1997)

$u1c = \text{project (u1b) (directorid)}$

DIRECTORID
1485
148
0
103
59
97

59 / 115

Union Example

Example (identities of actors of all movies newer than 1997)

$u1d = \text{project (u1b) (actorid)}$

ACTORID
282
503
26
187
138
1485
745

ACTORID
3578
9
308
302
406
350
793

60 / 115

Union Example

Example (identities of directors and actors of all movies newer than 1997)

```
u1e = union (u1c) (u1d)
```

ACTORID
1485
148
0
103
59
97
282
503
26
187

138
745
3578
9
308
302
406
350
793

61 / 115

Union Example

Example (names of directors and actors of all movies newer than 1997)

```
u1 = project (join (u1e) (person)  
              (u1e.actorid=person.id)) (name)
```

NAME
Spike Jonze
Tim Burton
Peter Hyams
Terry Gilliam
David Cronenberg
Cameron Diaz
John Malkovich
Johnny Depp
Christina Ricci

George Clooney
Udo Kier
Traci Lords
Arnold Schwarzenegger
Gabriel Byrne
Benicio Del Toro
Patricia Arquette
Jennifer Jason Leigh
Jude Law

62 / 115

Difference

Definition

difference: relation₁ - relation₂

creating a new relation from the tuples which are found in the first relation but not in the second

Statement

```
difference (relation_1) (relation_2)
```

- output relation header = relation₁ header = relation₂ header

63 / 115

Difference Example

Example (names of actors who have not played with Johnny Depp)

1. identities of actors who played with Johnny Depp
2. identities of all actors
3. identities of actors who have not played with Johnny Depp
4. names of actors who have not played with Johnny Depp

64 / 115

Difference Example

Example (identities of actors who have not played with Johnny Depp)

```
d1a = difference (i1b) (j3c)
```

ACTORID
503
9
138
243
1485
793
745
350

65 / 115

Difference Example

Example (names of actors who have not played with Johnny Depp)

```
d1 = project (join (d1a) (person)  
               (d1a.actorid=person.id))  
           (name)
```

NAME
John Malkovich
Arnold Schwarzenegger
George Clooney
Uma Thurman
Spike Jonze
Jude Law
Udo Kier
Jennifer Jason Leigh

66 / 115

References

Required text: Date

- ▶ Chapter 7: Relational Algebra
 - 7.1. Introduction
 - 7.2. Closure Revisited
 - 7.4. The Original Algebra: Semantics

67 / 115

Simple Queries

Statement

```
SELECT [ ALL | DISTINCT ] column_name [ , ... ]  
FROM table_name
```

- ▶ identical rows are allowed
 - ▶ **ALL**: preserve identical rows (default)
 - ▶ **DISTINCT**: multiple rows only once
- ▶ *: all columns

68 / 115

Query Examples

Example (all data of all movies)

```
SELECT * FROM MOVIE
```

Example (titles and years of all movies)

```
SELECT TITLE, YR FROM MOVIE
```

Example (years when movies were filmed)

```
SELECT DISTINCT YR FROM MOVIE
```

69 / 115

Sorting Results

Statement

```
SELECT [ ALL | DISTINCT ] column_name [ , ... ]  
FROM table_name  
[ ORDER BY { column_name [ ASC | DESC ] }  
[ , ... ] ]
```

- ▶ sort order:
 - ▶ **ASC**: ascending (default)
 - ▶ **DESC**: descending

70 / 115

Query Examples

Example (years when movies were filmed, in ascending order)

```
SELECT DISTINCT YR FROM MOVIE  
ORDER BY YR
```

Example (years when movies were filmed, in descending order)

```
SELECT DISTINCT YR FROM MOVIE  
ORDER BY YR DESC
```

71 / 115

Expressions

Statement

```
SELECT [ ALL | DISTINCT ]  
{ expression [ AS column_name ] } [ , ... ]  
FROM table_name  
[ ORDER BY { column_name [ ASC | DESC ] }  
[ , ... ] ]
```

- ▶ the new column can be named
- ▶ the name or order of the column can be used for sorting

72 / 115

Query Examples

Example (titles and total scores of all movies)

```
SELECT TITLE, SCORE * VOTES
FROM MOVIE
```

73 / 115

Query Examples

Example (titles and total scores of all movies, in descending order of total scores)

```
SELECT TITLE, SCORE * VOTES AS POINTS
FROM MOVIE
ORDER BY POINTS DESC
```

```
SELECT TITLE, SCORE * VOTES
FROM MOVIE
ORDER BY 2 DESC
```

74 / 115

Selecting Rows

Statement

```
SELECT [ ALL | DISTINCT ]
{ expression [ AS column_name ] } [, ...]
FROM table_name
[ WHERE condition ]
[ ORDER BY { column_name [ ASC | DESC ] }
[, ...] ]
```

► comparison operators:

= < > <= >= <>

► connectives:

NOT AND OR

75 / 115

Condition Expressions

► whether a column is empty or not:

column_name IS { NULL | NOT NULL }

► set membership:

column_name IN (value_set)

► string comparison

column_name LIKE pattern

- in the pattern, % can substitute any symbol group

76 / 115

Query Examples

Example (year of "Citizen Kane")

```
SELECT YR FROM MOVIE
WHERE (TITLE = 'Citizen_Kane')
```

Example (titles of movies with scores less than 3 and votes more than 10)

```
SELECT TITLE FROM MOVIE
WHERE ((SCORE < 3) AND (VOTES > 10))
```

77 / 115

Query Examples

Example (titles of movies with unknown year)

```
SELECT TITLE FROM MOVIE
WHERE (YR IS NULL)
```

Example (titles of movies in years 1967, 1954 and 1988)

```
SELECT TITLE, YR FROM MOVIE
WHERE (YR IN (1967, 1954, 1988))
```

78 / 115

Query Examples

Example (titles and scores of "Police Academy" movies)

```
SELECT TITLE, SCORE FROM MOVIE
WHERE (TITLE LIKE 'Police_Academy%')
```

79 / 115

Grouping

Statement

```
SELECT [ ALL | DISTINCT ]
{ expression [ AS column_name ] } [ , ... ]
FROM table_name
[ WHERE condition ]
[ GROUP BY column_name [ , ... ] ]
[ HAVING condition ]
[ ORDER BY { column_name [ ASC | DESC ] }
[ , ... ] ]
```

- ▶ selected rows can be grouped
- ▶ groups can be filtered

80 / 115

Processing Order

- ▶ the rows that satisfy the **WHERE** condition are selected
- ▶ selected rows are grouped using the columns specified in the **GROUP BY** clause
 - ▶ if no group, the entire result is one group
- ▶ the groups that satisfy the **HAVING** condition are selected
- ▶ the expressions given in the column list are calculated
- ▶ the result is sorted on the column list given in the **ORDER BY** clause

81 / 115

Group Values

- ▶ one value for each group
 - ▶ the value of the grouping column
 - ▶ the result of an aggregate function
- ▶ aggregate functions:
COUNT SUM AVG MAX MIN
 - ▶ column name as parameter
 - ▶ null values are ignored

82 / 115

Query Examples

Example (for every year, the number of movies with score greater than 8.5 in that year)

```
SELECT YR, COUNT(*) FROM MOVIE
WHERE (SCORE > 8.5)
GROUP BY YR
```

83 / 115

Query Examples

Example (score of the favorite movie of every year, in ascending order of years)

```
SELECT YR, MAX(SCORE) FROM MOVIE
GROUP BY YR
ORDER BY YR
```

84 / 115

Query Examples

Example (total number of votes)

```
SELECT SUM(VOTES) FROM MOVIE
```

85 / 115

Query Examples

Example (averages of movie scores which were filmed in the years where there are at least 25 movies for which more than 40 people have voted, in ascending order of years)

```
SELECT YR, AVG(SCORE)
FROM MOVIE
WHERE (VOTES > 40)
GROUP BY YR
HAVING (COUNT(ID) >= 25)
ORDER BY YR
```

86 / 115

Joining

- ▶ joining can be done using **WHERE** conditions
 - ▶ list the tables to join in the table list
 - ▶ use the dotted notation for columns with identical names
- ▶ processing order:
 - ▶ the Cartesian product of tables is obtained
 - ▶ the rows that satisfy the **WHERE** condition are selected
 - ▶ ...

87 / 115

Query Examples

Example (name of the director of "Star Wars")

```
SELECT NAME
FROM MOVIE, PERSON
WHERE ((DIRECTORID = PERSON.ID)
AND (TITLE = 'Star_Wars'))
```

88 / 115

Query Examples

Example (names of all the cast of "Alien")

```
SELECT NAME
FROM MOVIE, PERSON, CASTING
WHERE ((TITLE = 'Alien')
AND (MOVIEID = MOVIE.ID)
AND (ACTORID = PERSON.ID))
```

89 / 115

Query Examples

Example (titles of all movies Harrison Ford was in)

```
SELECT TITLE
FROM MOVIE, PERSON, CASTING
WHERE ((NAME = 'Harrison_Ford')
AND (MOVIEID = MOVIE.ID)
AND (ACTORID = PERSON.ID))
```

90 / 115

Query Examples

Example (titles of all movies where Harrison Ford was not the lead)

```
SELECT TITLE
FROM MOVIE, PERSON, CASTING
WHERE ((NAME = 'Harrison_Ford')
      AND (MOVIEID = MOVIE.ID)
      AND (ACTORID = PERSON.ID)
      AND (ORD > 1))
```

91 / 115

Query Examples

Example (titles and names of lead actors of all movies in 1962)

```
SELECT TITLE, NAME
FROM MOVIE, PERSON, CASTING
WHERE ((YR = 1962)
      AND (MOVIEID = MOVIE.ID)
      AND (ACTORID = PERSON.ID)
      AND (ORD = 1))
```

92 / 115

Table Expressions

- ▶ the join operation can be expressed as a table expression:
 - ▶ by specifying a condition
 - ▶ over columns with the same name
 - ▶ natural join
 - ▶ outer join
 - ▶ product

Statement

```
SELECT ...
FROM table_expression [ AS table_name ]
WHERE selection_condition
...
```

93 / 115

Joining Using Conditions

Expression Syntax

```
table1 JOIN table2
ON join_condition
```

94 / 115

Query Examples

Example (name of the director of "Star Wars")

```
SELECT NAME
FROM MOVIE JOIN PERSON
ON (DIRECTORID = PERSON.ID)
WHERE (TITLE = 'Star_Wars')
```

95 / 115

Joining Over Columns with the Same Name

Expression Statement

```
table1 JOIN table2
USING (column_name [, ...])
```

- ▶ repeated columns are taken once

96 / 115

Natural Join

Expression Syntax

table1 **NATURAL JOIN** table2

97 / 115

Outer Join

- ▶ in inner join, if a row of a table does not match with any row of the other table, it is NOT included in the result
- ▶ in outer join, it IS included where the row from the other table consists of empty values

Statement

```
table1 [ LEFT | RIGHT | FULL ]
       [ OUTER ] JOIN table2
```

98 / 115

Outer Join Examples

Example (left outer join)

Table: T1

NUM	NAME
1	a
2	b
3	c

Table: T2

NUM	VALUE
1	xxx
3	yyy
5	zzz

Table: SELECT * FROM T1 LEFT JOIN T2

NUM	NAME	NUM	VALUE
1	a	1	xxx
2	b		
3	c	3	yyy

99 / 115

Outer Join Examples

Example (right outer join)

Table: T1

NUM	NAME
1	a
2	b
3	c

Table: T2

NUM	VALUE
1	xxx
3	yyy
5	zzz

Table: SELECT * FROM T1 RIGHT JOIN T2

NUM	NAME	NUM	VALUE
1	a	1	xxx
3	c	3	yyy
		5	zzz

100 / 115

Outer Join Examples

Example (full outer join)

Table: T1

NUM	NAME
1	a
2	b
3	c

Table: T2

NUM	VALUE
1	xxx
3	yyy
5	zzz

Table: SELECT * FROM T1 FULL JOIN T2

NUM	NAME	NUM	VALUE
1	a	1	xxx
2	b		
3	c	3	yyy
		5	zzz

101 / 115

Query Examples

Example (titles of all movies with no known actor)

```
SELECT TITLE
FROM MOVIE LEFT JOIN CASTING
ON (MOVIEID = MOVIE.ID)
WHERE (ACTORID IS NULL)
```

102 / 115

Products

Expression Syntax

`table1 CROSS JOIN table2`

103 / 115

Self Join

- ▶ if the columns to join are in the same table
- ▶ give a new name to the table in the expression

104 / 115

Query Examples

Example (titles of all movies with the same number of votes)

```
SELECT M1.TITLE, M2.TITLE
FROM MOVIE AS M1, MOVIE AS M2
WHERE (M1.VOTES = M2.VOTES)
      AND (M1.ID < M2.ID)
```

105 / 115

Subqueries

Statement

WHERE expression operator
[**ALL** | **ANY**] (subquery)

- ▶ result of a subquery used in a condition expression
 - ▶ *careful*: row and column counts of subquery
 - ▶ **ALL**: for all values from subquery
 - ▶ **ANY**: for at least one value from subquery

106 / 115

Query Examples

Example (titles and scores of all movies more popular than "Star Wars", in descending order of scores)

```
SELECT TITLE, SCORE FROM MOVIE
WHERE ( SCORE >
  ( SELECT SCORE FROM MOVIE
    WHERE (TITLE = 'Star_Wars') )
) ORDER BY SCORE DESC
```

107 / 115

Query Examples

Example (titles of movies with more votes than the sum of all "Police Academy" movies)

```
SELECT TITLE FROM MOVIE
WHERE ( VOTES >
  ( SELECT SUM(VOTES) FROM MOVIE
    WHERE (TITLE LIKE 'Police_Academy%') )
)
```

108 / 115

Query Examples

Example (titles of movies with scores smaller than all "Police Academy" movies)

```
SELECT TITLE FROM MOVIE
WHERE ( SCORE < ALL
( SELECT SCORE FROM MOVIE
  WHERE (TITLE LIKE 'Police_Academy%') )
)
```

109 / 115

Set Operations

- ▶ an operation on two subquery results
- ▶ basic set operations in the relational model:
 - ▶ intersection: **INTERSECT**
 - ▶ union: **UNION**
 - ▶ difference: **EXCEPT**
- ▶ identical rows are not allowed in the result

110 / 115

Query Examples

Example (number of people who both directed and acted)

```
SELECT COUNT(*) FROM (
( SELECT DISTINCT DIRECTORID FROM MOVIE )
INTERSECT
( SELECT DISTINCT ACTORID FROM CASTING )
) AS DIRECTORACTOR
```

111 / 115

Query Examples

Example (number of directors who have not acted)

```
SELECT COUNT(*) FROM (
( SELECT DISTINCT DIRECTORID FROM MOVIE )
EXCEPT
( SELECT DISTINCT ACTORID FROM CASTING )
) AS DIRECTORONLY
```

112 / 115

Query Examples

Example (number of people who worked in movies before 1930)

```
SELECT COUNT(*) FROM (
( SELECT DISTINCT DIRECTORID FROM MOVIE
  WHERE (YR < 1930) )
UNION
( SELECT DISTINCT ACTORID FROM CASTING
  WHERE (MOVIEID IN
    ( SELECT ID FROM MOVIE
      WHERE (YR < 1930) )) )
) AS OLDMOVIES
```

113 / 115

Extra Examples

- ▶ how many movies John Travolta acted in and in which years
- ▶ titles and number of cast members of the movies filmed in 1978, in ascending order of cast member numbers
- ▶ names of actors who played with Johnny Depp
- ▶ titles and names of lead actors of the movies Uma Thurman was in
- ▶ names of actors with at least 10 lead roles

114 / 115

References

Required text: [Date](#)

- ▶ Chapter 8: Relational Calculus
 - ▶ 8.6. [SQL Facilities](#)
- ▶ Appendix B: [SQL Expressions](#)
- ▶ Chapter 19: Missing Information

[SQLzoo](#)

- ▶ A Gentle Introduction to SQL: <http://sqlzoo.net/>