Databases
Relational model & algebra

Hans Philippi

Example database inspired by imdb.com

January 9, 2020

Relational model: glossary

Movie			
movid	title	year	rating
tt0469494	There Will Be Blood	2007	8,1
tt0086879	Amadeus	1984	8,4
tt0102926	The Silence of the Lambs	1991	8,6
tt0110413	Léon	1994	8,7
tt0078788	Apocalypse Now	1979	8,5

The *schema* of this *relation (table)* is:

Movie(movid, title, year, rating)

There are five *tuples (records, rows)*

There are four *attributes (fields)*: movid, title, year, rating

There are four *columns*, identified by an attribute, each containing five values

The *degree* of the relation Movie is four

Relational model: definitions

Definition:
A domain D is a set of atomic values. (For example: integers, chars, strings, floats, dollars, dates)

Definition:
A relation schema R, denoted $R(A_1, A_2, ..., A_n)$,
exists of a relation name R and a set of attributes $A_1, A_2, ..., A_n$;
n is the degree of schema R.
We define $attr(R) = \{A_1, A_2, ..., A_n\}$.

Definition:
A relation r over a schema R, denoted $r(R)$, is a *set* of tuples
$< v_1, v_2, ..., v_n >$, where every $v_i \in D_i \cup \{null\}$.
So every A_i is connected to a specific domain D_i.

Relational model: tuple identification

Movie		
title	year	rating
Invasion of the Body Snatchers	1956	7,8
Amadeus	1984	8,4
Invasion of the Body Snatchers	1978	7,4
Apocalypse Now	1979	8,5

How do we identify movies?

- By title?
- By title + year?

Relational model: tuple identification

Movie			
movid	title	year	rating
tt0049366	Invasion of the Body Snatchers	1956	7,8
tt0086879	Amadeus	1984	8,4
tt0077745	Invasion of the Body Snatchers	1978	7,4
tt0078788	Apocalypse Now	1979	8,5

How do we identify movies?

- By movid: primary key!

Relational model: tuple identification

Movie			
movid	title	year	rating
tt0049366	Invasion of the Body Snatchers	1956	7,8
tt0086879	Amadeus	1984	8,4
tt0077745	Invasion of the Body Snatchers	1978	7,4
tt0078788	Apocalypse Now	1979	8,5

We identify movies by a key, but beware:
tuples are also identified by (`movid, title`) or (`movid, year`) or (`movid, title, rating`) or any combination of attributes containing a key.
Hence the notion of key versus superkey.

Relational model: key constraints

Suppose we have a relation $r(R)$.

Definition:
A set of attributes $K \subseteq attr(R)$ is a superkey if for each valid relation $r(R)$:
$\forall t_1, t_2 \in r : t_1[K] = t_2[K] \Rightarrow t_1 = t_2$

Definition:
A superkey $K \subseteq attr(R)$ is a (candidate) key if there is no $K' \subset K, K' \neq K$, that is also a superkey.

A key K identifies a tuple, because the key values of a tuple are unique in the relation. This requirement is *intensional*;
we call such a requirement a constraint.

Relational model: key constraints

Definition:
One of the candidate keys is chosen (for some reason) as primary key.

Definition:
A foreign key is a set of attributes $K \in attr(R_i)$ that occurs in another relation R_j as a candidate key. We say that $R_i[K]$ references $R_j[K]$.

Relational mode: key constraints

Definition:
The <u>Key integrity constraint</u> requires that in a primary key attribute no null values may occur.

Definition:
The <u>Referential integrity constraint</u> states that

if $R_i[K]$ is a foreign key referring to $R_j[K]$,
then $R_i[K] \subseteq R_j[K] \cup \{null\}$.

Relational model: SQL DDL

```
CREATE TABLE Book (
bookid             integer        not null,
title              varchar(100)   not null,
author             varchar(30)    not null,
price              float,
date_of_purchase   date,
publisher_id       char(6),
CONSTRAINT Book_pk PRIMARY KEY (bookid),
CONSTRAINT Book_fk_Publisher FOREIGN KEY (publisher_id)
          REFERENCES Publisher(publisher_id)
);
```

Relational model: algebra

Relational algebra

- fundamental query language
- based on set theory
- *yardstick*: relational completeness
- *procedural*: ordering of operations
- *compositional*: a query may be composed from subqueries
- *concise*: only five basic operators
- *practical use*: query optimization

Relational model: algebra

Unary operator: **selection** σ_p $\qquad$ p is selection predicate

Actor		
personid	name	birth_year
nm0000204	Natalie Portman	1982
nm0000288	Christian Bale	1974
nm0000358	Daniel Day-Lewis	1957
nm0000201	Michelle Pfeiffer	1958

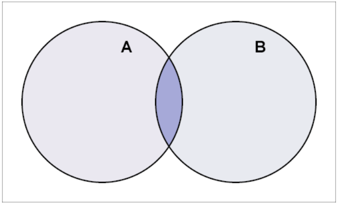
$\sigma_{birth\_year<1960}$(Actor)		
personid	name	birth_year
nm0000358	Daniel Day-Lewis	1957
nm0000201	Michelle Pfeiffer	1958

Relational model: algebra

Unary operator: **selection** with complex predicates

$$\sigma_{(birth\_year>1996)\wedge(gender='female')}(\texttt{Actor})$$

Logical *and* corresponds to set intersection

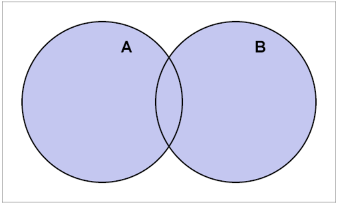


Relational model: algebra

Unary operator: **selection** with complex predicates

$$\sigma_{(country='Netherlands')\vee(country='Belgium')}(\texttt{Actor})$$

Logical *or* corresponds to set union



Relational model: algebra

Unary operator: **projection** π_L L is projection list

Movie			
movid	title	year	rating
tt0049366	Invasion of the Body Snatchers	1956	7,8
tt0086879	Amadeus	1984	8,4
tt0077745	Invasion of the Body Snatchers	1978	7,4
tt0078788	Apocalypse Now	1979	8,5

$\pi_{title,year}$(Movie)	
title	year
Invasion of the Body Snatchers	1956
Amadeus	1984
Invasion of the Body Snatchers	1978
Apocalypse Now	1979

Relational model: algebra

Composition of projection and selection

Actor		
personid	name	birth_year
nm0000204	Natalie Portman	1982
nm0000288	Christian Bale	1974
nm0000358	Daniel Day-Lewis	1957
nm0000201	Michelle Pfeiffer	1958

$\pi_{name,birth\_year}(\sigma_{birth\_year<1960}(\texttt{Actor}))$	
name	birth_year
Daniel Day-Lewis	1957
Michelle Pfeiffer	1958

Relational model: algebra

Binary operator: **union** $\cup$ schema compatibility

Actor		
personid	name	birth_year
nm0000154	Mel Gibson	1956
nm0000354	Matt Damon	1970
nm0000358	Daniel Day-Lewis	1957

Director		
personid	name	birth_year
nm0000759	Paul Thomas Anderson	1970
nm0000154	Mel Gibson	1956
nm0000338	Francis Ford Coppola	1939

Relational model: algebra

Binary operator: **union** $\cup$ schema compatibility

Actor ∪ Director		
personid	name	birth_year
nm0000154	Mel Gibson	1956
nm0000354	Matt Damon	1970
nm0000358	Daniel Day-Lewis	1957
nm0000759	Paul Thomas Anderson	1970
nm0000338	Francis Ford Coppola	1939

Binary operator: **difference** − schema compatibility

Actor − Director		
personid	name	birth_year
nm0000354	Matt Damon	1970
nm0000358	Daniel Day-Lewis	1957

Binary operator: **intersection** $\cap$ schema compatibility

Actor ∩ Director		
personid	name	birth_year
nm0000154	Mel Gibson	1956

Relational model: algebra

Binary operator: **Cartesian product** $\times$

R	
A	B
a	11
b	43

S	
C	D
b	25
c	41
b	21

R × S			
A	B	C	D
a	11	b	25
a	11	c	41
a	11	b	21
b	43	b	25
b	43	c	41
b	43	b	21

Binary operator: **theta-join** $\bowtie_\theta$

θ is matching condition

R	
A	B
a	11
b	43
c	37

S	
C	D
b	55
c	31
b	21

R $\bowtie_\theta$ S			
A	B	C	D
b	43	b	21
c	37	c	31

$$\theta : (R.A = S.C) \wedge (R.B > S.D)$$

Binary operator: **natural join** $\bowtie$ default matching condition

R	
A	B
a	11
b	43
c	37

S	
A	D
b	55
c	31
b	21
d	17

R $\bowtie$ S		
A	B	D
b	43	55
b	43	21
c	37	31

Relational model: algebra

Binary operator: **division** $\div$

T	
A	B
1	1
1	3
1	4
2	2
2	4
6	1
6	3
8	1
8	3
8	4
8	7

U
B
1
3
4

$T \div U$
A
1
8

Unary operators: **assignment & renaming**

$$T[A_1, ..., A_n] := < alg\_expr >$$

renaming in expressions (variations)

$$\rho(T)(< alg\_expr >)$$

$$\rho(T, A_1, ..., A_n)(< alg\_expr >)$$

Relational model: algebra

assignment & renaming examples:

$$Oldmovies[omid, omtitle] := \pi_{movid,title}\ (\sigma_{year<1930}(Movie))$$

on the fly renaming within an expression:

$$\ldots \bowtie \rho(Oldmovies, omid, omtitle)(\pi_{movid,title}\ (\sigma_{year<1930}(Movie)))$$

MonetDB: DBMS using MAL, a dialect of relational algebra

- developed at CWI, Amsterdam
- platform for analytical databases
- outperforms several commercial systems
- main-memory approach
- MAL is intermediate language for query processing
- SQL queries are translated to MAL and optimized
- *Pathfinder:* XQUERY queries are translated to MAL and optimized

Relational model: overview of algebra

Overview unary operators

- selection $\sigma_p(R)$ — p is selection predicate
- projection $\pi_L(R)$ — L is projection list
- renaming $\rho(R)$ — or using assignment

Overview binary operators

- union $R \cup S$ — schema compatibility
- difference $R - S$ — schema compatibility
- intersection $R \cap S$ — schema compatibility
- cartesian product $R \times S$
- theta-join $R \bowtie_\theta S$ — θ is matching condition
- natural join $R \bowtie S$
- division $R \div S$ — schema requirements