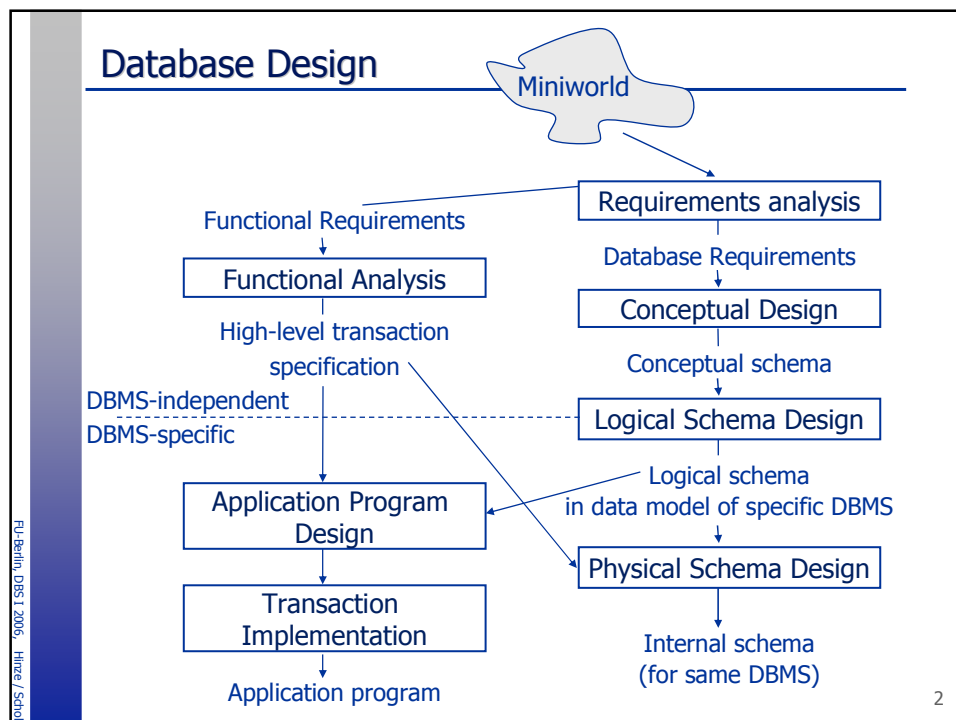


Relational Languages: Relational Algebra & Relational calculus

Relational Algebra
Relational Calculus Languages
 Tuple Calculus
 Domain Calculus
Relational Languages Equivalence



Relational Languages

- ▶ Data model:
 - Collection of concepts to describe structure of a DB ✓
 - Basic operations for specifying retrieval and update

- ▶ Formal Languages for handling data
 - Relational Algebra: procedural language, specifies *how* to evaluate queries

 - Relation calculus: predicate logic interpretation of data and queries, declarative language, specifies *which* data to select

 - Both are closed languages: results of queries on relations are relations

Relational Languages: Example Queries

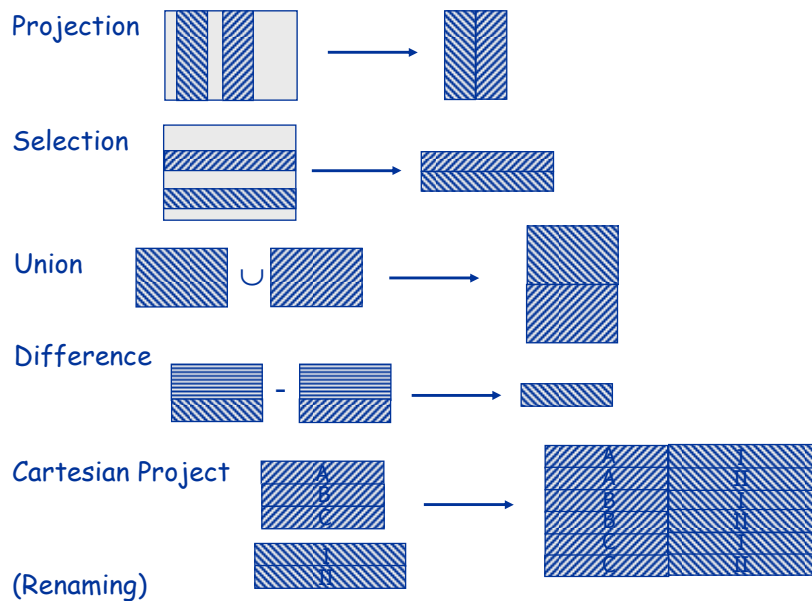
- ▶ Queries at Video shop DB
 1. Names of all customers
 2. All customers named Anna
 3. All movies by George Lucas from 1999 or later
 4. All tapes and their corresponding movie
 5. All customers who have rented at least one science fiction film
 6. All customers whose rented movies all have category "suspense"
 7. Customers that had rented all movies
 8. All movies no copy of which are currently on loan
 9. Number of tapes for each movie
 10. Total receipts of each movie within the last year

Relational Algebra: Basics

- ▶ Procedural query language
- ▶ Consider two relations
 - $R(A_1:C_1, A_2:C_2, \dots, A_r:C_r)$, degree r
 - $S(B_1:D_1, B_2:D_2, \dots, B_s:D_s)$, degree s
- ▶ 5 (+1) Basic Operations
- ▶ Other operations can be derived

- ▶ Schema-compatible relations
 - = identical schemas (with renaming)
 - same degree of relations [$r=s$]
 - domains of R and S are pair-wise compatible
 - \exists Permutation φ of indices $\{1, \dots, r\}: \forall i, 1 \leq i \leq r: C_i = D_{\varphi(i)}$

Relational Algebra : Basic Operations

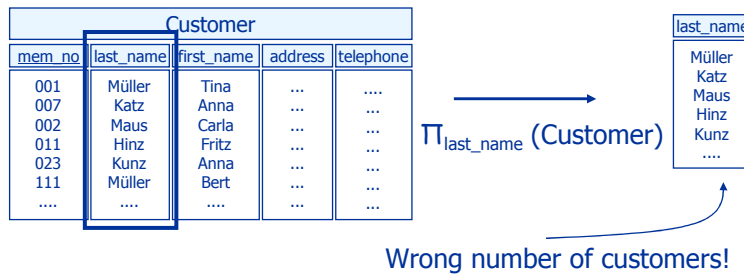


Relational Algebra : Basic Operations

► Projection: $\Pi_{\langle \text{List of attributes} \rangle}(\text{Relation})$ Important concept

- Reduction of relation schema to selected attributes
- Reduction of relation tuple to respective attributes
- Removal of duplicate tuples (set condition!)
- $\Pi_B(R) = \{t[B] \mid t \in R\}, B \subseteq R(R)$

Example: Names of all customers



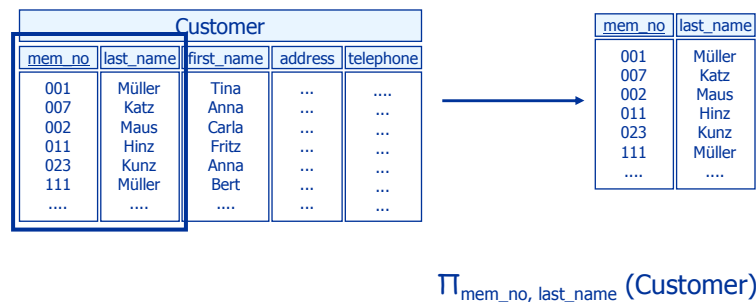
FU-Berlin, DBS I 2006, Hinz / Scholz

7

Relational Algebra: Basic Operations

Example: Names of all customers

Include key attribute into projection:



FU-Berlin, DBS I 2006, Hinz / Scholz

8

Relational Algebra: Basic Operations

▶ Selection: $\sigma_{\langle \text{condition} \rangle}(\text{Relation})$

Important concept

- New relation schema = old relation schema
- All tuples $t \in R$ that satisfy condition
- Condition: qualifier-free Boolean predicate P
- $\sigma_p(R) = \{ t \mid P(t) \text{ true}, t \in R \}$

Example: All customers named Anna

| Customer | | | | |
|----------|-----------|------------|---------|-----------|
| mem_no | last_name | first_name | address | telephone |
| 001 | Müller | Tina | ... | ... |
| 007 | Katz | Anna | ... | ... |
| 002 | Maus | Carla | ... | ... |
| 011 | Hinz | Fritz | ... | ... |
| 023 | Kunz | Anna | ... | ... |
| 111 | Bla | Bert | ... | ... |
| ... | ... | ... | ... | ... |

| mem_no | last_name | first_name | address | telephone |
|--------|-----------|------------|---------|-----------|
| 007 | Katz | Anna | ... | ... |
| 023 | Kunz | Anna | ... | ... |
| ... | ... | ... | ... | ... |

$\sigma_{\text{first_name}=\text{"Anna"}}(\text{Customer})$

Relational Algebra: Selection predicates

▶ Primitive Predicates

- Compare an attribute and a value or two attributes
- Operands: Attribute names, constants
- Operators: =, ≠, ≤, ≥, <, >

▶ Boolean row predicates

- Combine primitive predicates by AND, OR, NOT and parenthesis
- Boolean operators: \wedge, \vee, \neg
- Operator preference and brackets as usual

▶ No "second order" predicates

Example: "Rows which have NULL as value of all attributes"

Relational Algebra: Basic Operations

Example: All movies by Lucas from 1999 or later

| Movie | | | | | | |
|-------|--------------|----------|------|-----------|-------|--------|
| id | title | category | year | director | price | length |
| 095 | Psycho | suspense | 1960 | Hitchcock | 2.00 | ... |
| 112 | ET | comedy | 1982 | Spielberg | 1.50 | ... |
| 345 | Star Wars I | SciFi | 1999 | Lucas | 2.00 | ... |
| 222 | Psycho | suspense | 1998 | Van Sant | 2.20 | ... |
| 290 | Star Wars IV | SciFi | 1997 | Lucas | 2.00 | ... |
| 100 | Jaws | horror | 1975 | Spielberg | 1.50 | ... |
| ... | ... | ... | ... | ... | ... | ... |

$\sigma_{\text{director}=\text{"Lucas"} \wedge \text{year} \geq 1999}$ (Customer)

| id | title | category | year | director | price | length |
|-----|-------------|----------|------|----------|-------|--------|
| 345 | Star Wars I | SciFi | 1999 | Lucas | 2.00 | ... |
| ... | ... | ... | ... | ... | ... | ... |

Relational Algebra: Basic Operations

▶ Selection properties

- $\sigma_P(\sigma_Q(R)) = \sigma_Q(\sigma_P(R))$
- $\sigma_P(\sigma_P(R)) = \sigma_P(R)$
- $\sigma_{Q \wedge P}(R) = \sigma_Q(\sigma_P(R))$
- $\sigma_{Q \vee P}(R) = \sigma_Q(R) \cup \sigma_P(R)$
- $\sigma_{\neg P}(R) = R - \sigma_P(R)$

▶ Projection properties

- $X \subseteq Y \subseteq R(R) \rightarrow \Pi_X(\Pi_Y(R)) = \Pi_X(R)$
- $X, Y \subseteq R(R) \rightarrow \Pi_X(\Pi_Y(R)) = \Pi_{X \cap Y}(R) = \Pi_Y(\Pi_X(R))$

▶ Selection and Projection Property

- $\text{attr}(P) \subseteq X \subseteq R(R) \rightarrow \Pi_X(\sigma_P(R)) = \sigma_P(\Pi_X(R))$
where $\text{attr}(P)$ denotes the set of attributes used in P

Relational Algebra: Basic Operations

▶ Union

- R and S are schema-compatible
- $R \cup S = \{t \mid t \in R \vee t \in S\}$

▶ Difference

- ☞ R and S are schema-compatible
- ☞ $R - S = \{t \mid t \in R \wedge t \notin S\} = R \setminus S$

| R | |
|---|---|
| A | B |
| a | c |
| b | c |
| a | d |
| b | e |
| a | a |
| b | d |
| a | e |

-

| S | |
|---|---|
| A | B |
| a | c |
| b | c |
| a | a |

=

| A | B |
|---|---|
| a | e |
| b | d |
| a | d |
| b | e |

13

FU-Berlin, DBS I 2006, Hintze / Scholz

Relational Algebra: Basic Operations

▶ Cartesian Product $T = R \times S$

Important concept

- $T(A_1:C_1, A_2:C_2, \dots, A_r:C_r, B_1:D_1, B_2:D_2, \dots, B_s:D_s)$
- $R \times S = \{t \bullet u \mid t \in R, u \in S\}$
- Tupel concatenation of $t = (v_1, \dots, v_r), u = (w_1, \dots, w_s)$
 $t \bullet u := (v_1, \dots, v_r, w_1, \dots, w_s), \text{ degree } r + s$

▶ Rename:

- ☞ Changes Schema, no new relation
- ☞ Necessary for using relation or attribute more than once in one query
- ☞ Rename relation R in S $\rho_S(R)$
- ☞ Rename attribute B in A $\rho_{A \leftarrow B}(R)$

14

FU-Berlin, DBS I 2006, Hintze / Scholz

Relational Algebra: Basic Operations

Example: All Tapes and their corresponding movie

| Tape | | | Movie | | | | | | |
|------|--------|----------|-------|--------------|----------|------|-----------|-------|--------|
| id | format | movie_id | id | title | category | year | director | price | length |
| 0001 | DVD | 095 | 095 | Psycho | suspense | 1960 | Hitchcock | 2.00 | ... |
| 0002 | DVD | 112 | 112 | ET | comedy | 1982 | Spielberg | 1.50 | ... |
| 0003 | VHS | 222 | 345 | Star Wars I | SciFi | 1999 | Lucas | 2.00 | ... |
| 0004 | DVD | 345 | 222 | Psycho | suspense | 1998 | Van Sant | 2.20 | ... |
| 0005 | VHS | 345 | 290 | Star Wars IV | ScFi | 1997 | Lucas | 2.00 | ... |
| 0009 | VHS | 345 | 100 | Jaws | horror | 1975 | Spielberg | 1.50 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

$\sigma_{\text{Tape.movie_id}=\text{Movie.id}}(\text{Tape} \times \text{Movie}) =$

$\sigma_{\text{t.movie_id}=\text{m.id}}(\rho_{\text{t}}(\text{Tape}) \times \rho_{\text{m}}(\text{Movie}))$

| t.id | t.format | t.movie_id | m.id | m.title | m.category | m.year | m.director | m.price | m.length |
|------|----------|------------|------|-------------|------------|--------|------------|---------|----------|
| 0001 | DVD | 095 | 095 | Psycho | suspense | 1960 | ... | 2.00 | ... |
| 0002 | DVD | 112 | 112 | ET | comedy | 1982 | ... | 1.50 | ... |
| 0003 | VHS | 222 | 222 | Psycho | suspense | 1998 | ... | 2.20 | ... |
| 0004 | DVD | 345 | 345 | Star Wars I | SciFi | 1999 | ... | 2.00 | ... |
| 0005 | VHS | 345 | 345 | Star Wars I | SciFi | 1999 | ... | 2.00 | ... |
| 0009 | VHS | 345 | 345 | Star Wars I | SciFi | 1999 | ... | 2.00 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

15

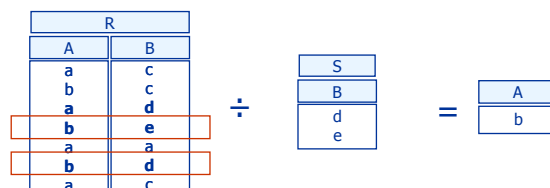
Relational Algebra: Additional Operations

Intersection $R \cap S$

- R and S are schema-compatible
- $R \cap S = R - (R - S)$

Division $R \div S$

- For queries with all-quantifier
- $R(A_1, \dots, A_r, B_1, \dots, B_k), S(B_1, \dots, B_k)$
- $T = R \div S, T(A_1, \dots, A_r)$
- $R \div S = \Pi_{R-S}(R) - \Pi_{R-S}((\Pi_{R-S}(R) \times S) - R)$



16

Relational Algebra: Additional Operations

▶ Theta- Join: $R \bowtie_{\theta} S$

Important concept

- Consider $R(A_1, \dots, A_r), S(B_1, \dots, B_s)$
- $T = R \bowtie_{\theta} S = \{(A_1, \dots, A_r, B_1, \dots, B_s) \mid \theta(A_1, \dots, A_r, B_1, \dots, B_s) \text{ true}\}$
- Boolean predicate θ formed of primitive predicates $a \text{ op } b$, $a \in R(R), b \in R(S), \text{op} \in \{=, \neq, \leq, \geq, <, >\}$
- $R \bowtie_{\theta} S \equiv \sigma_{\theta}(R \times S)$

▶ Equijoin: theta-join form $R \bowtie_{R.A_i=S.B_j} S$

17

Relational Algebra: Additional Operations

Example: All Tapes and their corresponding movie

Tape $\bowtie_{\text{Tape.movie_id=Movie.id}}$ Movie

| tape. id | tape. format | tape. movie_id | movie. id | movie. title | movie. category | movie. year | movie. director | movie. price | movie. length |
|-------------|-----------------|-------------------|--------------|-----------------|--------------------|----------------|--------------------|-----------------|------------------|
| 0001 | DVD | 095 | 095 | Psycho | suspense | 1960 | ... | 2.00 | ... |
| 0002 | DVD | 112 | 112 | ET | comedy | 1982 | ... | 1.50 | ... |
| 0003 | VHS | 222 | 222 | Psycho | suspense | 1998 | ... | 2.20 | ... |
| 0004 | DVD | 345 | 345 | Star Wars I | SciFi | 1999 | ... | 2.00 | ... |
| 0005 | VHS | 345 | 345 | Star Wars I | SciFi | 1999 | ... | 2.00 | ... |
| 0009 | VHS | 345 | 345 | Star Wars I | SciFi | 1999 | ... | 2.00 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

18

Relational Algebra: Additional Operations

► Natural join: $R \bowtie S$

Important concept

- Consider $R(A_1, \dots, A_r, C_1, \dots, C_k), S(B_1, \dots, B_s, C_1, \dots, C_k)$
- $\text{dom}(C_j)$ equal in R and $S, \forall 1 \leq j \leq k$
- $\forall 1 \leq i \leq r \forall 1 \leq j \leq s : A_i \neq B_j$
- $R \bowtie S = \Pi_{A_1, \dots, A_r, C_1, \dots, C_k, B_1, \dots, B_s} (\sigma_{R.C_1=S.C_1 \wedge \dots \wedge R.C_k=S.C_k} (R \times S))$
- $R \bowtie S$ has degree $r + s + k$

Relational Algebra: Additional Operations

Example: All Tapes and their corresponding movie

Tape \bowtie ($\rho_{\text{movie_id} \leftarrow \text{id}}$ (Movie))

| id | format | movie_id | title | category | year | director | price | length |
|------|--------|----------|-------------|----------|------|----------|-------|--------|
| 0001 | DVD | 095 | Psycho | suspense | 1960 | ... | 2.00 | ... |
| 0002 | DVD | 112 | ET | comedy | 1982 | ... | 1.50 | ... |
| 0003 | VHS | 222 | Psycho | suspense | 1998 | ... | 2.20 | ... |
| 0004 | DVD | 345 | Star Wars I | SciFi | 1999 | ... | 2.00 | ... |
| 0005 | VHS | 345 | Star Wars I | SciFi | 1999 | ... | 2.00 | ... |
| 0009 | VHS | 345 | Star Wars I | SciFi | 1999 | ... | 2.00 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

Relational Algebra: Additional Operations

▶ Inner joins (Natural join, Theta-join)

- Non-matching tuples are lost
- Associative and commutative

▶ Outer join $R \bowtie S$

- Non-matching tuples included
- Right outer join: \bowtie
 $(r, \text{NULL}, \dots, \text{NULL})$ - rows for each $r \in R$ which does not have join row in S - according to join predicate P

| R | | | S | | = | A1 | A2 | B1 | B2 |
|---|---|-----------------------|---|---|---|----|----|----|----|
| a | c | $\bowtie_{R.A2=S.B1}$ | d | f | | b | d | d | f |
| b | d | | e | g | | - | - | e | g |

21

FU-Berlin, DBS I 2006, Hinz / Scholz

Relational Algebra: Additional Operations

▪ Left outer join: \bowtie

$(\text{NULL}, \dots, \text{NULL}, s)$ - rows for each $s \in S$ which does not have join row in R - according to join predicate P

| R | | | S | | = | A1 | A2 | B1 | B2 |
|---|---|-----------------------|---|---|---|----|----|----|----|
| a | c | $\bowtie_{R.A2=S.B1}$ | d | f | | a | c | - | - |
| b | d | | e | g | | b | d | d | f |

▪ Full outer join: \bowtie

tuples of both R and S are included

| R | | | S | | = | A1 | A2 | B1 | B2 |
|---|---|-----------------------|---|---|---|----|----|----|----|
| a | c | $\bowtie_{R.A2=S.B1}$ | d | f | | a | c | - | - |
| b | d | | e | g | | b | d | d | f |
| - | - | | - | - | | - | - | e | g |

- Left and right outer join not commutative

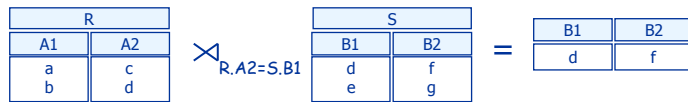
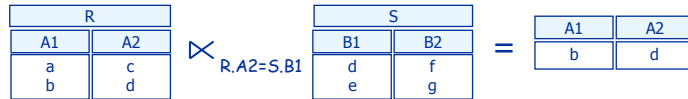
22

FU-Berlin, DBS I 2006, Hinz / Scholz

Relational Algebra: Additional Operations

▶ Semi-Joins

- $R \bowtie S = \Pi_{R(R)}(R \bowtie S) = S \bowtie R$
- All tuples in R that have potential join partners in S (semi-join of R with S)



- Semi-join not commutative

23

FU-Berlin, DBS 1.2006, Hinz / Scholz

Relational Algebra: Examples

All Customers who have rented at least one science fiction film

$$\Pi_{\text{mem_no, last_name}} (\text{customer} \bowtie_{\text{member_no}=\text{member}} \text{rental} \bowtie_{\text{tape_id}=\text{id}} \text{tape} \bowtie_{\text{movie_id}=\text{movie.id}} (\sigma_{\text{category}=\text{"SciFi"}} (\text{movie})))$$

All Customers whose rented movies all have category "suspense"

$$\text{customer} - (\text{customer} \bowtie_{\text{member}=\text{mem_no}} (\text{rental} \bowtie_{\text{Tape_id}=\text{id}} (\text{tape} \bowtie_{\text{movie_id}=\text{movie.id}} (\sigma_{\text{category} \neq \text{"suspense"}} (\text{movie}))))))$$

24

FU-Berlin, DBS 1.2006, Hinz / Scholz

Relational Algebra: Examples

Customers that had rented all movies

$$(\Pi_{\text{mem_no}, \text{movie.id}} (\text{customer} \bowtie_{\text{member_no}=\text{member}} \text{rental} \bowtie_{\text{tape_id}=\text{id}} \text{tape} \bowtie_{\text{movie_id}=\text{movie.id}} \text{movie})) \div \Pi_{\text{movie.id}} (\text{movie}))$$

All movies no copy of which are currently on loan

$$\text{movie} - (\text{movie} \bowtie_{\text{movie_id}=\text{movie.id}} (\sigma_{\text{until}=\text{"NULL"}} (\text{rental}) \bowtie_{\text{tape_id}=\text{id}} \text{tape}))$$

Relational Algebra: Examples

Number of tapes for each movie

No count!

Total receipts of each movie within the last year

No count!

Relational Algebra: Operator Tree

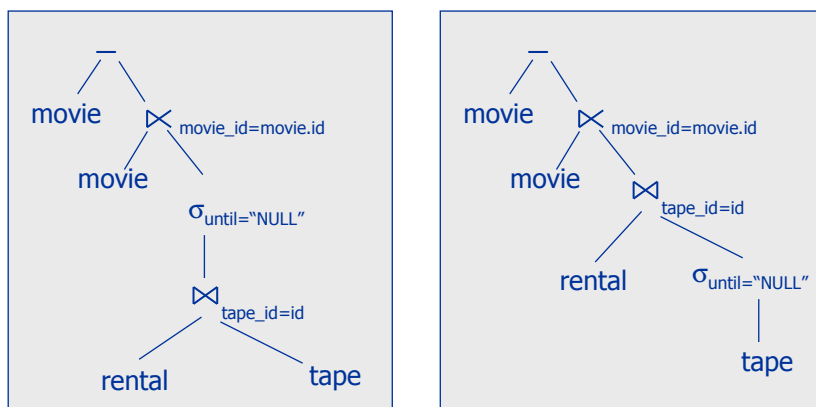
Important concept

- ▶ Data structure representing RA expression
- ▶ More clearly structured
- ▶ Evaluation by recursive evaluation of the tree

- ▶ Basis for algebraic optimization
 - Implementation of Algebraic Optimization by transformation of operator tree
 - Interchange of operations according to the laws of RA e.g., $(R \bowtie_{\sigma_p}(S))$ faster than $(\sigma_p(R \bowtie S))$
 - Reduction of number of tuples to evaluate
 - No change of time complexity in general

Relational Algebra: Operator Tree

Example: All movies no copy of which are currently on loan



Relational Algebra: Operations

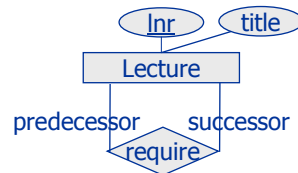
- ▶ Basic set of RA operations $\{\Pi, \sigma, \times, -, \cup\}$
 - RA expressions using different operators of RA can be expressed by the above operators only

- ▶ Missing operations in RA:

- Transitive closure

Example: $\text{Lecture}(\text{Inr}, \dots)$

$\text{Require}(\text{preLNR}, \text{succLNR})$



Query: All lectures required for lecture XYZ

- Predicates on tables, e.g. arithmetic functions: SUM, AVERAGE, MAXIMUM, MINIMUM, or COUNT, grouping

Relational Algebra: Summary

- ▶ Relational Algebra:

- Formal language for handling data in relational model
- Procedural language, *how* to retrieve data
- No practical relevance for querying DB
- Formal basis for query optimization

- ▶ Important terms & concepts

- Union $R \cup S$
- Difference $R - S$
- Selection $\sigma_{\langle \text{predicate} \rangle}(R)$
- Projection $\Pi_{\langle \text{attribute list} \rangle}(R)$
- Cartesian Product $R \times S$
- Joins $R \bowtie_{\langle \text{predicate} \rangle} S$
- Operator tree

Relational Calculus: Languages

- ▶ Non-procedural, declarative query language
- ▶ Two types of languages

- Tuple calculus
Variables in expressions represent a row of a relation (tuple variable)

Example: $\{c.\text{last_name}, c.\text{first_name} \mid c \in \text{Customer}\}$

- Domain calculus
Variables represent domain values of attributes of a relation of the DB (domain variables)

Example: $\{[l,f] \mid \exists m,a,t ([m,l,f,a,t] \in \text{Customer})\}$

31

Tuple Calculus: Introduction

- ▶ Queries in tuple calculus Important concept
 - Form: $\{t \mid F(t)\}$ with tuple variable t
 - F is formula
 - t is the only free variable of formula F
 - Answer of query: set of tuples t from DB with $F(t) = \text{TRUE}$

- ▶ Example:

All customers named Anna

$\{c \mid c \in \text{Customer} \wedge c.\text{first_name} = \text{"Anna"}\}$

- ▶ Free variable: no existence- or all-quantor (\exists, \forall)
- ▶ Formulae made of atoms
- ▶ Atoms evaluate to TRUE or FALSE

32

Tuple Calculus: Atom Forms

- ▶ $t \in R$
 - with relation name R and tuple variable t
 - Alternative notation: $R(t)$
 - Example: $c \in \text{Customer}$

- ▶ $(t.A \text{ operator } s.B)$
 - t, s tuple variables
 - A, B attribute names of relations on which t, s ranges
 - operator $\in \{=, \leq, \geq, \neq, <, >\}$
 - Example: $c.\text{member_no} = r.\text{member}$

- ▶ $(t.A \text{ operator } c)$
 - c constant value
 - Example: $c.\text{first_name} = \text{"Anna"}$

Relational Calculus: Atoms and Formulae

1. Every atom is a formula.
2. If F_1 is a formula so are $\neg F_1$ and (F_1)
3. If F_1, F_2 are formulae so are $F_1 \wedge F_2, F_1 \vee F_2, F_1 \Rightarrow F_2$
4. If F with t free is formula so are $\exists t (F), \forall t (F)$
5. Nothing else is a formula

Tuple Calculus: Examples

▶ Formula properties

- $\forall t \in R(F(t)) = \neg(\exists t \in R(\neg F(t)))$
- $\exists t \in R(F(t)) = \neg(\forall t \in R(\neg F(t)))$

▶ Tuple Calculus Examples:

- Names of all customers
 $\{c.last_name, c.first_name \mid c \in \text{Customer}\}$
- All customers named Anna
 $\{c \mid c \in \text{Customer} \wedge c.first_name = \text{„Anna“}\}$
- All movies by George Lucas from 1999 or later
 $\{m.id, m.title \mid m \in \text{Movie} \wedge m.director = \text{„Lucas“} \\ \wedge m.year \geq 1999\}$

35

Tuple Calculus: Examples

All Tapes and their corresponding movie

$\{t.id, m.title \mid t \in \text{Tape} \wedge m \in \text{Movie} \wedge t.movie_id = m.id\}$

All Customers who have rented at least one science fiction film

$\{c \mid c \in \text{Customer} \\ \wedge \exists r \in \text{Rental}(c.member_no=r.member \\ \wedge \exists t \in \text{Tape}(r.tape_id=t.id \\ \wedge \exists m \in \text{Movie}(t.movie_id=m.id \\ \wedge m.category = \text{„SciFi“}))))\}$

36

Tuple Calculus: Examples

All Customers whose rented movies all have category "suspense"

$$\{c \mid c \in \text{Customer} \\ \wedge \exists r \in \text{Rental}(c.\text{member_no}=r.\text{member} \\ \wedge \exists t \in \text{Tape}(r.\text{tape_id}=t.\text{id} \\ \wedge \forall m \in \text{Movie}(t.\text{movie_id}=m.\text{id} \Rightarrow \\ m.\text{category}=\text{"suspense"}))\}$$
$$\{c \mid c \in \text{Customer} \\ \wedge \exists r \in \text{Rental}(c.\text{member_no}=r.\text{member} \\ \wedge \exists t \in \text{Tape}(r.\text{tape_id}=t.\text{id} \\ \wedge \neg \exists m \in \text{Movie}(t.\text{movie_id} = m.\text{id} \Rightarrow \\ m.\text{category} \neq \text{"suspense"}))\}$$

37

FL-Berlin, DBS I 2006, Hinz / Scholz

Tuple Calculus: Examples

Customers that had rented all movies

$$\{c \mid c \in \text{Customer} \\ \wedge \forall m \in \text{Movie}(\\ \exists t \in \text{Tape}(t.\text{movie_id}=m.\text{id} \\ \wedge \exists r \in \text{Rental}(r.\text{tape_id}=t.\text{id} \\ \wedge c.\text{member_no}=r.\text{member}))\}$$

All movies no copy of which are currently on loan

$$\{m \mid m \in \text{Movie} \\ \wedge \forall t \in \text{Tape}(t.\text{movie_id}=m.\text{id} \\ \Rightarrow \neg \exists r \in \text{Rental}(r.\text{tape_id}=t.\text{id} \\ \wedge r.\text{until}=\text{'NULL'})\}$$

38

FL-Berlin, DBS I 2006, Hinz / Scholz

Tuple Calculus: Examples

Number of tapes for each movie

No count!

Total receipts of each movie within the last year

No count!

Tuple Calculus vs Relational Algebra

▶ Selection

$\sigma_{\langle \text{predicate} \rangle}(R)$ equivalent to $\{ r \mid r \in R \wedge \langle \text{predicate} \rangle \}$

▶ Projection, cross product

$\Pi_{R,a,s,b}(R \times S)$ equivalent to $\{ r.a, s.b \mid r \in R \wedge s \in S \}$

▶ Join

$R \bowtie_p S$ equivalent to $\{ r \mid r \in R \wedge s \in S \wedge P \}$

▶ Union

$R \cup S$ equivalent to $\{ t \mid t \in R \vee t \in S \}$

▶ Difference

$R - S$ equivalent to $\{ t \mid t \in R \wedge \neg t \in S \}$

Tuple Calculus: Safe expression

- ▶ Solution set for $\{ t \mid \neg t \in R \}$? Important concept
 - All tuples NOT belonging to R, infinite set?
- ▶ Formula domain:
 - all attribute data of referenced relations in DB, and
 - constants of the formula
- ▶ Safe expression:
tuple calculus expression is safe if result is subset of domain
 - Idea: safe if all free tuple variables restricted in F
 - Example: $\{x \mid x \in T \wedge \neg x \in R\}$ safe

41

FL-Berlin, DBS I 2006, Hinz / Scholz

Tuple Calculus: Practical Use

- ▶ Tuple calculus basis for DB language QUEL
 - In 70s used in Ingres (University of Berkeley)
 - Commercial INGRES now SQL
- ▶ Examples:
 - All customers named Anna
RANGE of c is Customer
RETRIEVE (c.mem_no, c.last_name, c.first_name)
WHERE c. first_name = "Anna"
 - All movies by George Lucas from 1999 or later
RANGE of m is Movie
RETRIEVE (m.title)
WHERE m.director="Lucas" AND year>=1999

42

FL-Berlin, DBS I 2006, Hinz / Scholz

Domain Calculus: Introduction

▶ Queries in domain calculus

- Form: $\{[a,b,c] \mid F([a,b,c])\}$ with a,b,c domain variables
- F is formula
- a, b, c are the only free variables of formula F
- Answer of query: set of tuples t from DB with $F(t) = \text{TRUE}$

▶ Example:

All customers named Anna

$\{[m,f,l] \mid \exists a,t ([m,f,l,a,t] \in \text{Customer} \wedge f = \text{"Anna"})\}$

- ▶ Domain variables represent sets of possible attribute values (domains)
- ▶ Formulae made of atoms
- ▶ Atoms evaluate to TRUE or FALSE

43

Domain Calculus: Atom Forms

▶ $[a_1, \dots, a_n] \in R$

- with relation R of grade n
- Domain variables a_1, \dots, a_n according to order in schema of R
- Example: $[m,f,l,a,t] \in \text{Customer}$

▶ $(a_i \text{ operator } a_j)$

- operator $\in \{=, \leq, \geq, \neq, <, >\}$
- a_i, a_j domain variables
- Example: $mn = r$

▶ $(a_i \text{ operator } c)$

- c constant value
- Example: $f = \text{"Anna"}$

44

Domain Calculus: Examples

Names of all customers

$$\{[l,f] \mid \exists m,a,t ([m,f,l,a,t] \in \text{Customer})\}$$

All customers named Anna

$$\{[m,f,l] \mid \exists a,t ([m,f,l,a,t] \in \text{Customer} \wedge f = \text{"Anna"})\}$$

All movies by George Lucas from 1999 or later

$$\{[i,t] \mid \exists c,y,d,p,l ([i,t,c,y,d,p,l] \in \text{Movie} \wedge \\ d = \text{"Lucas"} \wedge y \geq 1999)\}$$

All Tapes and their corresponding movie

$$\{[ti,t] \mid \exists tf,mi ([ti,tf,mi] \in \text{Tape} \wedge \\ \exists i,c,y,d,p,l ([i,t,c,y,d,p,l] \in \text{Movie} \wedge mi = i))\}$$

Domain Calculus: Examples

All Customers who have rented at least one science fiction film

$$\{[m,f,l] \mid \exists a,t ([m,f,l,a,t] \in \text{Customer} \\ \wedge \exists rti,rm,rf,ru ([rti,rm,rf,ru] \in \text{Rental} \wedge m = rm \\ \wedge \exists ti,tf,tmi ([ti,tf,tmi] \in \text{Tape} \wedge rti = ti \\ \wedge \exists mi,mt,mc,my,md,mp,ml ([mi,mt,mc,my,md,mp,ml] \in \text{Movie} \\ \wedge tmi = mi \wedge mc = \text{"SciFi"}))))\}$$

All Customers whose rented movies all have category "suspense"

$$\{[m,f,l] \mid \exists a,t ([m,f,l,a,t] \in \text{Customer} \\ \wedge \exists rti,rm,rf,ru ([rti,rm,rf,ru] \in \text{Rental} \wedge m = rm \\ \wedge \exists ti,tf,tmi ([ti,tf,tmi] \in \text{Tape} \wedge rti = ti \\ \wedge \forall mi,mt,mc,my,md,mp,ml ([mi,mt,mc,my,md,mp,ml] \in \\ \text{Movie} \\ \wedge tmi = mi \Rightarrow mc = \text{"suspense"}))))\}$$

Domain Calculus: Examples

Customers that had rented all movies

$$\{[m,f,l] \mid \exists a,t ([m,f,l,a,t] \in \text{Costumer} \\ \wedge \forall mi,mt,mc,my,md,mp,ml ([mi,mt,mc,my,md,mp,ml] \in \text{Movie} \\ \wedge \exists ti,tf,tmi ([ti,tf,tmi] \in \text{Tape} \\ \wedge tmi=mi \\ \wedge \exists rti,rm,rf,ru ([rti,rm,rf,ru] \in \text{Rental} \wedge m=rm \\ \wedge rti=ti))))\}$$

All movies no copy of which are currently on loan

$$\{[i,t,d] \mid \exists c,y,p,l ([i,t,c,y,d,p,l] \in \text{Movie} \\ \wedge \forall ti,tf,tmi ([ti,tf,tmi] \in \text{Tape} \wedge tmi=i \Rightarrow \\ \neg \exists rti,rm,rf,ru ([rti,rm,rf,ru] \in \text{Rental} \wedge rti=ti))))\}$$

47

FI-Berlin, DSS 1.2006, Hinz / Scholz

Domain Calculus: Examples

Number of tapes for each movie

No count!

Total receipts of each movie within the last year

No count!

48

FI-Berlin, DSS 1.2006, Hinz / Scholz

Domain Calculus

▶ Safe expression $\{[a_1, \dots, a_n] \mid F(a_1, \dots, a_n)\}$:

1. Constants $c_i (1 \leq i \leq n)$ in domain of F if $[c_1, \dots, c_n]$ in solution
2. For all $\exists a_1 (F(a_1))$ F true only for elements of domain of F
3. For all $\forall a_1 (F(a_1))$ F true only iff true for all elements of domain of F

2.+3. necessary since domain variables not bound to relations

▶ Formula domain:

- all attribute data of referenced relations in DB, and
- constants of the formula

Domain Calculus: Practical Use

▶ Domain calculus basis for DB language QBE

- QBE = query by example
- One of the first graphical query languages
- Interface option for DB2 (IBM)
- templates of relations on screen
- Users fill in constants (...), examples (____), output (P. ...)
- Example: All movies by George Lucas from 1999 or later

| Movie | <u>id</u> | title | category | year | director | price | length |
|-------|-----------|-------|----------|-------|----------|-------|--------|
| | P.1 | P.bla | | ≥1999 | Lucas | | |

each column in template as implicit domain variable

$\{[i,t] \mid \exists c,y,d,p,l ([i,t,c,y,d,p,l] \in \text{Movie} \wedge d = \text{"Lucas"} \wedge y \geq 1999)\}$

Relational Calculus: Summary

▶ Relational Calculus:

- Formal languages for handling data in relational model
- Declarative language, *which* data to retrieve
- Basis for QUEL, QBE, SQL

▶ Important terms & concepts

- Tuple Calculus
- Domain Calculus
- Bound and free variables
- Safe expression

- Relational Completeness

Relational Languages: Conclusion

▶ Relational completeness:

Important concept

- query language for relational model is complete if at least as expressive as relational algebra
- base line for DB query languages: every query language should be as expressive as relational algebra

▶ Equivalent expressivenesses:

- Relational Algebra
- Tuple calculus restricted to safe expressions
- Domain calculus restricted to safe expressions

- Proof using induction:
RA expressions \rightarrow TC expressions \rightarrow
DC expressions \rightarrow RA expressions