Information Systems

WS 2005, JKU Linz
Course 3: The Relational Model

Gábor Bodnár

URL: `http://www.risc.uni-linz.ac.at/education/courses/ws2005/is/`

Overview

- The relational structure.
- The relational algebra.

The Relational Model

The relational model was formally introduced by E. F. Codd in 1970 and became the dominant modeling technique in the IT sector.

The relational model represents data in the form of two-dimensional tables.

A relational database is a collection of such tables each having a unique name.

area-code	area-name
0732	Linz
0662	Salzburg
0512	Innsbruck
05574	Bregenz
01	Wien
⋮	

ACodes

phone-number	owner-first-name	owner-last-name	area-code
243486	Thomas	Muster	0732
875678	Hans	Schmidt	0662
875678	Hans	Schmidt	0732
34452	John	Example	05574
4554323	Monika	Beispiel	01
⋮			

APhone

Relational Structure

Let $D_1, \ldots, D_n$ be the domains for attributes $A_1, \ldots, A_n$ respectively. Then an n-ary *relation* over the given attributes with the given domains can be represented as

$$R \subseteq D_1 \times \cdots \times D_n.$$

In practice R is a finite set, so it can be described as a table with $m = \#R$ rows and n columns:

$$\begin{array}{ccc} d_{11} & \ldots & d_{1n} \\ d_{21} & \ldots & d_{2n} \\ \vdots & & \\ d_{m1} & \ldots & d_{mn}. \end{array}$$

Relational Structure (continued)

Remark: In the literature the word "table" is used as a synonym for "relation" and "row" is used as a synonym for "tuple".

From the viewpoint of the modeling process, we consider the tables with atomic cells.

A table does not contain two identical rows.

The order of rows in a table is unspecified.

The order of columns can also be arbitrary, as long as the attribute values of each row go into the corresponding column.

Remarks

We call a list of attributes with their corresponding domains a *relation scheme*.

In the relational model, a row corresponds to an entity and a table to an entity set.

The notion of superkey, candidate key and primary key applies to the relational model.

A nonempty set K of attributes is a superkey for R if for any $r, s \in R$, $r \neq s$ implies $r|_K \neq s|_K$ (where $r|_K$ denotes the tuple of attribute values of r belonging to attributes in K).

In certain cases it is unavoidable to use *null values* as some attribute values (usually denoted by NULL).

Relational Algebra

The relational algebra is a query language to execute several kinds of operations on a relational database.

The fundamental operations are the following: select (unary), project (unary), rename (unary), Cartesian product (binary), set theoretic union (binary), set theoretic difference.

Additional operations are defined in terms of the fundamental ones are: set theoretic intersection (binary), natural join (binary), division (binary).

The operations may have additional parameters beside their input arguments; they are put in the subscript of the notation.

Example Tables

We will use the following small database to illustrate the operations.

Student

sid	fname	lname
0251563	Werner	Schmidt
0245654	Andrea	Ritter
0245675	Daniela	Schmidt

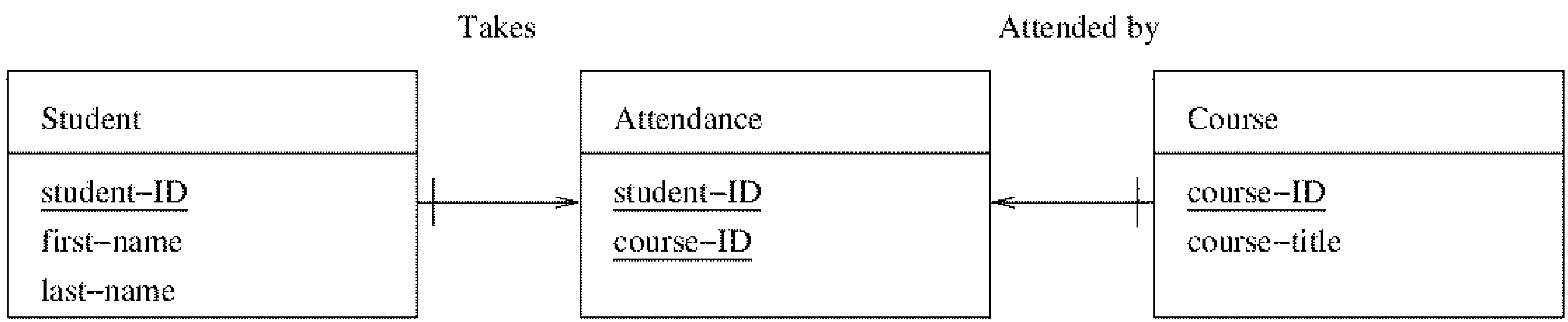
Attendance

sid	cid
0251563	327456
0251563	327564
0245654	327456

Course

cid	title
327456	Analysis I
327564	Algebra I

Recall from the last lecture.



Select

The operations selects tuples of the relation that satisfy the predicate which is an additional parameter to it. It is denoted by σ.

The atomic predicates allowed in the parameter of σ are $=, \neq, <, \leq, <, \geq$, moreover with the logical connectives $\wedge, \vee$ compound formulas can be build.

Example

The operation $\sigma_{\texttt{lname}=\text{"Schmidt"}}(\texttt{Student})$ results

0251563	Werner	Schmidt
0245675	Daniela	Schmidt

.

Project

The operation projects the relation onto the set of specified attributes and additionally eliminates multiple rows. It is denoted by Π.

Example

$\Pi_{\texttt{fname},\texttt{lname}}(\texttt{Student})$

Werner	Schmidt
Andrea	Ritter
Daniela	Schmidt

$\Pi_{\texttt{lname}}(\texttt{Student})$

Schmidt
Ritter

Cartesian Product

The operation combines each tuple from its first argument with each tuple from the second. The result is a relation whose attributes are the disjoint union of the attributes of the arguments.

The number of rows of the Cartesian product is the product of the numbers of rows of the arguments. It is denoted by $\times$.

Since the operation leads to exponential blowup of the number of rows, it is avoided in practice.

The operations which are formally defined via Cartesian products (e.g. joins) are implemented using more efficient algorithms that do not generate rows that would not satisfy the predicate of the operation anyway.

Example

$$\texttt{Student} \times \texttt{Course}$$

0251563	Werner	Schmidt	327456	Analysis I
0251563	Werner	Schmidt	327564	Algebra I
0245654	Andrea	Ritter	327456	Analysis I
0245654	Andrea	Ritter	327564	Algebra I
0245675	Daniela	Schmidt	327456	Analysis I
0245675	Daniela	Schmidt	327564	Algebra I

.

Rename

The operation just changes the name of its argument to the one given to it in its parameter. It is denoted by ρ.

Example

We want to select the students with identical last names

$$\Pi_{\texttt{Student.fname},\texttt{Student.lname}}(\sigma_P(\texttt{Student} \times \rho_{\texttt{Student2}}(\texttt{Student})))$$

where P is

$$\texttt{Student.lname} = \texttt{Student2.lname} \wedge \texttt{Student.sid} \neq \texttt{Student2.sid}.$$

Werner	Schmidt
Daniela	Schmidt

.

Union

The operation takes the set theoretic union of two compatible relations, where compatible means that the sets of attributes and the corresponding domains are the same. It is denoted by $\cup$.

$\mathtt{Teacher}$

fname	lname
Wolfgang	Schiffer
Sabrina	Kaiser

$$\Pi_{\mathtt{fname},\mathtt{lname}}(\mathtt{Student}) \cup \mathtt{Teacher}$$

Wolfgang	Schiffer
Werner	Schmidt
Sabrina	Kaiser
Daniela	Schmidt
Andrea	Ritter

.

Difference and Intersection

The difference operation takes the set theoretic difference of two compatible relations. It is denoted by $-$.

The intersection operation takes the set theoretic intersection of two compatible relations. It is denoted by $\cap$ and it can be defined as

$$A \cap B = A - (A - B).$$

Natural Join

The operation combines a Cartesian product a selection and a projection operation into a single operation. It is denoted by $\bowtie$.

Let A and B be relations with attribute sets R and S respectively and let $\{A_1, \ldots, A_n\} = R \cap S$, then

$$A \bowtie B = \Pi_{R \cup S}(\sigma_{A.A_1 = B.A_1 \wedge \cdots \wedge A.A_n = B.A_n}(A \times B)).$$

If $R \cap S = \emptyset$ then $A \bowtie B = A \times B$; we also have $A \bowtie B = B \bowtie A$, and the operation binds from left to right.

Example

`Student` $\bowtie$ `Attendance`

sid	fname	lname	cid
0251563	Werner	Schmidt	327456
0251563	Werner	Schmidt	327564
0245654	Andrea	Ritter	327456

`Student` $\bowtie$ `Attendance` $\bowtie$ `Course`

sid	fname	lname	cid	title
0251563	Werner	Schmidt	327456	Analysis I
0251563	Werner	Schmidt	327564	Algebra I
0245654	Andrea	Ritter	327456	Analysis I

Division

The division operation requires that the set of attributes of the divisor is a subset of the set of attributes of the dividend.

It leaves those tuples of the dividend in the result (projected to the difference attribute set) which have an occurrence for each tuple in the divisor over the common attributes.

It is denoted by $\div$. Let A and B be relations with attribute sets R and S respectively and let $S \subseteq R$, then

$$A \div B = \Pi_{R-S}(A) - \Pi_{R-S}((\Pi_{R-S}(A) \times B) - A).$$

Example

$\mathsf{J} = \texttt{Student} \bowtie \texttt{Attendance} \bowtie \texttt{Course}$

sid	fname	lname	cid	title
0251563	Werner	Schmidt	327456	Analysis I
0251563	Werner	Schmidt	327564	Algebra I
0245654	Andrea	Ritter	327456	Analysis I

Course

cid	title
327456	Analysis I
327564	Algebra I

$\mathsf{J} \div \texttt{Course}$

0251563	Werner	Schmidt

.

Assignment

It is denoted by $\leftarrow$, and works as the assignment operator in any imperative programming languages.

If a name of a relation of the database stands on the left hand side of the operator, the statement results the modification of the database.

If a new name gets assigned to a relation, it does not mean to introduce a new relation into the database.

Example

$\texttt{Student} \leftarrow \texttt{Student} - \sigma_{\texttt{sid}=\text{"}0245675\text{"}}(\texttt{Student})$

This deletes the student with ID '0245675' from the `Students` table.

Generalized Projection

This operation is analogous to projection with the additional feature that arithmetic expressions formed by the attributes and constants from the corresponding domains are allowed as parameters for the operation.

Example

Let the `Analysis-result` table be described by the attributes `sid`, `pts1`, `pts2`, `pts3`, where the pts'es are the points the students reached on the first, second and third exam of a course. Then the generalized projection

$$\Pi_{\mathtt{sid},(\mathtt{pts1}+\mathtt{pts2}+\mathtt{pts3})/3}(\mathtt{Analysis\text{-}result})$$

will describe the average score of the students.

Outer Join

It is an extension of Join to deal with missing information.

- *left outer-join* the result will contain also those tuples from the first (left) argument of the operation that did not match any tuple from the second (right) argument, and pads the missing attributes with null values.
- *right outer-join* analogously to the previous case, the tuples of the second (right) argument are carried over and padded with null values as necessary.
- *full outer-join* this operation, analogously to the previous cases, carries over every tuples from its arguments and pads them with null values ("on the left" or "on the right") as necessary.

Example

A left outer-join `Student` ⟕ `Attendance` ⟕ `Course` results

sid	fname	lname	cid	title
0251563	Werner	Schmidt	327456	Analysis I
0251563	Werner	Schmidt	327564	Algebra I
0245654	Andrea	Ritter	327456	Analysis I
0245675	Daniela	Schmidt	NULL	NULL

Aggregate operations

The general form is $\quad {}_{G_1,\ldots,G_n}\mathcal{G}_{F_1 A_1,\ldots,F_m A_m}(E)$,

where E is a relational algebra expression, the G_i are attributes to group on, F_i is an aggregate function to be applied on attribute A_i.

The semantic of the operation is the following:

- The tuples in E are grouped by the G_i, such that in each group the tuples have the same G_i-value for each i and tuples in different groups have at least one difference in the G_i-values.

- A tuple of the resulting relation is $(g_1, \ldots, g_n, a_1, \ldots, a_m)$, where the $g_1, \ldots, g_n$ identifies a group and $a_1, \ldots, a_m$ are the results of the corresponding aggregate functions on the attribute values of the tuples in the group.

Example

Let us consider a relation `Credits`,

`Credits`

sid	cid	credit
0251563	327456	3
0251563	327564	2
0245654	327456	3

The aggregate operation

$${}_{\texttt{sid}}\mathrm{Creditsum}_{\textbf{sum}\ \texttt{credit}}(\texttt{Credits})$$

results

0251563	5
0245654	3

.

Database Modification

Let E be a relational algebra expression resulting a relation compatible to a relation R.

delete $R \leftarrow R - E$.

insert $R \leftarrow R \cup E$.

update It is denoted by δ. General form: $\delta_{A \leftarrow E}(R)$,

where R is a relation, A is an attribute and E is an arithmetic expression with constants and attributes in R, resulting a value in the domain of A.

If one wants to apply the update only on a subset of tuples fulfilling the predicate P, one can use $\delta_{A \leftarrow E}(\sigma_P(R))$.

Views

With the view construction one can create virtual tables in a database which are not included in the logical scheme.

A *view* is defined via a new command

$$\texttt{create view } V \texttt{ as } Q$$

where V is a new name (identifier) for the relation to be created and Q is a query of the relational algebra.

After the definition we allow V to appear anywhere in a relational algebra statement where a relation of the database can appear.

Views–Caveats

- Whenever a view is used to modify the database the resolution of the operation may require extensive usage of NULL values.
- As views only represent relations that are derived from the relations of the database via queries, they have to be recomputed after each modification of the database. The null values can be a real problem.
- From the definition it also follows that views can be used to define new views, hence resolution of views can require several steps.

Summary

- Relational structure (columns, rows, tables)

Relational algebra:

- Select, Project, Cartesian product, Rename
- Union, Difference, Intersection
- Natural join, Division
- Assignment, Generalized Projection, Outer joins
- Aggregate operations

Database modification

Views