Relational databases: usage principles

Mathematical Preliminaries, Relational Model

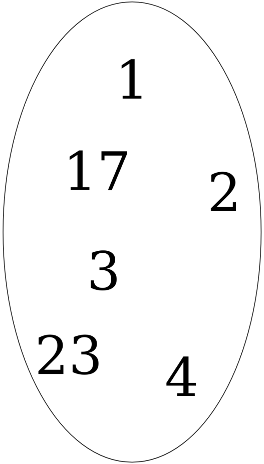
Hiba ALQASIR

2021-2022

télécom saint-étienne
école d'ingénieurs / nouvelles technologies

Preliminaries

A set is an unordered list of instances without multiple duplicates allowed.



In the relational model, the only accepted structure to represent data is the **relation**.

Relation

Between two sets

Given the sets A, B

A relation $\mathcal{R}$ on A, B can be defined as:
$\mathcal{R} \subset A \times B$

$\mathcal{R}$ is a subset of the Cartesian product $A \times B$

Relation

Among three sets

Given the sets A, B, C

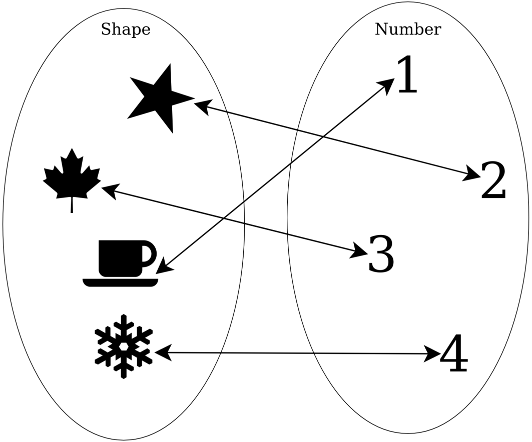
A relation $\mathcal{R}$ on A, B, C can be defined as:
$\mathcal{R} \subset A \times B \times C$

$\mathcal{R}$ is a subset of the Cartesian product $A \times B \times C$

A relation of degree n on the domains $D_1, D_2, \ldots, D_n$
is a *finite* subset of the Cartesian product $D_1 \times D_2 \times \cdots \times D_n$

Relation

Representation / Graph



Relation

Representation / Table

Shape	Number
	1
	2
	3
	4

Tuples

- An element of a relation of dimension n is a tuple $(a_1, a_2, \ldots, a_n)$.
- In the table representation, a tuple is a row, also referred to as a record.
- A tuple is a single entry in the table having the attributes.

Tuples

The definition of a relation as a set has some important implications:

- The order of the tuples is indifferent because there is no order in a set.
- The same tuple cannot be found twice because there are no duplicates in a set.
- There is no ‘empty cell’ in the relation (theoretically).

Schema

We can describe a relation by:

1. The name of the relation.
2. A (distinct) name for each dimension, called attribute name, noted A_i .
3. The domain (type) of the value of each dimension, noted D_i .

Schema: $\mathcal{R}(A_1 : D_1, A_2 : D_2, \ldots, A_n : D_n)$

Concepts and vocabulary

	Model term	Table representation term
$\mathcal{R}$	Relation	Table
T	Tuple	Row
A	Attribute name	Column name
a	Attribute value	Cell
D	Domain	Type

Keys

A key of a relation $\mathcal{R}$ is a minimal subset K of the attributes such that any attribute of $\mathcal{R}$ depends functionally on K.

$\forall u, v \in R$, if $u.K = v.K$, then $u = v$

We can have several potential keys: **candidate keys**.
We must choose only one key: **primary key**.

A **foreign key** is an attribute or group of attributes in one table that are primary keys in another table

Recap

1. A relation of degree n on the domains $D_1, D_2, \ldots, D_n$
   is a finite subset of the Cartesian product $D_1 \times D_2 \times \cdots \times D_n$.
2. The schema of a relation is written $\mathcal{R}(A_1 : D_1, A_2 : D_2, \ldots, A_n : D_n)$,
   where $\mathcal{R}$ is the name of the relation and the A_i are the names of attributes.
3. An element of this relation is a tuple $(a_1, a_2, \ldots, a_n)$,
   where the a_i are the values of the attributes.

Example of a relation

Courses(courseId:string, name:string, semester:string)

courseId	name	semester
c11	Fighting sleep	autumn
c23	Combat bad mood	winter
c34	Seasonal sneezing	spring

télécom saint-étienne

Example of a relation

Students2020(studentId:string, name:string, promotion:string)

studentId	name	promotion
grumpy2020	Grumpy	2020
dopey2020	Dopey	2020
sneezy2020	Sneezy	2020
sleepy2020	Sleepy	2020
sleepy2019	Sleepy	2019

Students2019(studentId:string, name:string, promotion:string)

studentId	name	promotion
grumpy2019	Grumpy	2019
sneezy2019	Sneezy	2019
sleepy2019	Sleepy	2019

Operations on relations

A relation is a set of tuples.
We can apply the set operations and some unary and binary operations.

Unary operations

- Selection
- Projection
- Rename

Selection

$\mathcal{R}_1 := \sigma_C(\mathcal{R}_2)$,
where C is a selection criteria:

- Comparison between an attribute of the relation, A, and a constant a.
- Comparison between two attributes A_1 and A_2

Selection returns all those tuples of $\mathcal{R}_2$ that satisfy C (extracts a subset of tuples).

télécom saint-étienne

Selection

courseId	name	semester
c11	Fighting sleep	autumn
c23	Combat bad mood	winter
c34	Seasonal sneezing	spring

Courses

$$\text{SpringCourses} = \sigma_{\text{semester}=\text{spring}}(Courses)$$

courseId	name	semester
c34	Seasonal sneezing	spring

SpringCourses

Projection

$\mathcal{R}_1 := \prod_S(\mathcal{R}_2)$,
where S is a subset of the attributes in the relation $\mathcal{R}_2$.

Projection returns all tuples with the given attributes only (extracts a subset of attributes).

Note: A projection returns the distinct tuples (after removing duplicates) only.

Projection

courseId	name	semester
c11	Fighting sleep	autumn
c23	Combat bad mood	winter
c34	Seasonal sneezing	spring

Courses

$$\text{CoursesNames} = \prod_{\text{courseId,name}}(Courses)$$

courseId	name
c11	Fighting sleep
c23	Combat bad mood
c34	Seasonal sneezing

CoursesNames

Rename

$\mathcal{R}_1 := \rho_N(\mathcal{R}_2)$,
where N is the new schema for the result relation $\mathcal{R}_1$.

$\mathcal{R}_1 = \rho_{\mathcal{R}_1(A_1,\ldots,A_n)}(\mathcal{R}_2)$
$\mathcal{R}_1$ is a relation with attributes $A_1, \ldots, A_n$ and the same tuples as $\mathcal{R}_2$.

télécom saint-étienne

Rename

courseId	name	semester
c11	Fighting sleep	autumn
c23	Combat bad mood	winter
c34	Seasonal sneezing	spring

Courses

$$\text{NewCourses} = \rho_{\text{NewCourses(id, courseName, semester)}}(Courses)$$

id	courseName	semester
c11	Fighting sleep	autumn
c23	Combat bad mood	winter
c34	Seasonal sneezing	spring

NewCourses

Binary operations

- Union
- Intersection
- Difference
- Cartesian product

Union

$\mathcal{R} := \mathcal{R}_1 \cup \mathcal{R}_2$

The union operation returns **tuples** that appear in one or both relations.

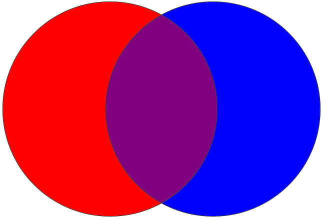
Number of tuples in the resulting relation: at most $T(\mathcal{R}_1) + T(\mathcal{R}_2)$ tuples.

Union operation is valid if and only if the attributes of $\mathcal{R}_1$,$\mathcal{R}_2$ are the same, *i.e.* $A(\mathcal{R}_1) = A(\mathcal{R}_2)$.

Union

studentId	name	promotion
grumpy2020	Grumpy	2020
dopey2020	Dopey	2020
sneezy2020	Sneezy	2020
sleepy2020	Sleepy	2020
grumpy2019	Grumpy	2019
sneezy2019	Sneezy	2019
sleepy2019	Sleepy	2019

Students2019 $\cup$ Students2020



Intersection

$\mathcal{R} := \mathcal{R}_1 \cap \mathcal{R}_2$

Intersection operation returns **tuples** that appear in both the relations.

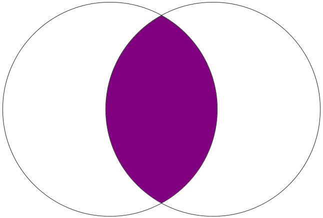
Number of tuples in the resulting relation: at most $min(T(\mathcal{R}_1), T(\mathcal{R}_2))$ tuples.

Intersection operation is valid if and only if the attributes of $\mathcal{R}_1$,$\mathcal{R}_2$ are the same, *i.e.* $A(\mathcal{R}_1) = A(\mathcal{R}_2)$.

Intersection

studentId	name	promotion
sleepy2019	Sleepy	2019

Students2019 ∩ Students2020



Difference

$\mathcal{R} := \mathcal{R}_1 - \mathcal{R}_2$

Difference operation returns **tuples** that appear in one relation (first one) but not in the other (second one).

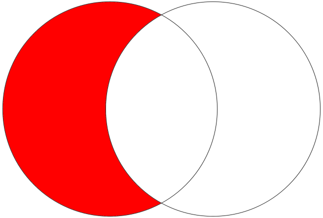
Number of tuples in the resulting relation: at most $T(\mathcal{R}_1)$ tuples.

Difference operation is valid if and only if the attributes of $\mathcal{R}_1$,$\mathcal{R}_2$ are the same, *i.e.* $A(\mathcal{R}_1) = A(\mathcal{R}_2)$.

Difference

studentId	name	promotion
grumpy2019	Grumpy	2019
sneezy2019	Sneezy	2019

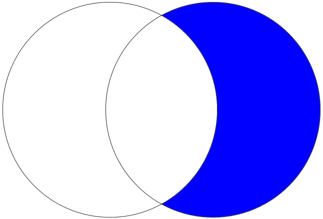
Students2019 − Students2020



Difference

studentId	name	promotion
grumpy2020	Grumpy	2020
dopey2020	Dopey	2020
sneezy2020	Sneezy	2020
sleepy2020	Sleepy	2020

Students2020 − Students2019



Cartesian product

$\mathcal{R} := \mathcal{R}_1 \times \mathcal{R}_2$

Pair each tuple t_1 of $\mathcal{R}_1$ with each tuple t_2 of $\mathcal{R}_2$.

Number of tuples in the resulting relation: $T(\mathcal{R}_1) * T(\mathcal{R}_2)$ tuples.
Schema of $\mathcal{R}$ is the attributes of $\mathcal{R}_1$ and then $\mathcal{R}_2$, in order.
But beware attribute A of the same name in $\mathcal{R}_1$ and $\mathcal{R}_2$: use $\mathcal{R}_1.A$ and $\mathcal{R}_2.A$.

No validity constraint.

Cartesian product

A	B
a	b
x	y

$\mathcal{R}_1$

C	D
c	d
u	v
x	y

$\mathcal{R}_2$

A	B	C	D
a	b	c	d
a	b	u	v
a	b	x	y
x	y	c	d
x	y	u	v
x	y	x	y

$\mathcal{R}_1 \times \mathcal{R}_2$

Join operations

- Theta join
- Natural
- Inner
- Outer

Theta join

$\mathcal{R} := \mathcal{R}_1 \bowtie_C \mathcal{R}_2$
where C is a condition (as in 'if' statements) that refers to attributes of $\mathcal{R}_2$.

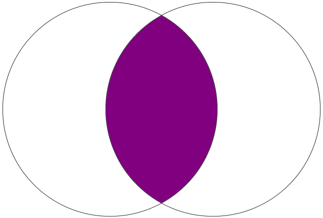
- Take the product $\mathcal{R} := \mathcal{R}_1 \times \mathcal{R}_2$
- Then apply σ_C to $\mathcal{R}$.

télécom saint-étienne

Natural join

$\mathcal{R} := \mathcal{R}_1 \bowtie \mathcal{R}_2$

Cartesian product of two relations followed by the removal of duplicate attributes.



Natural join

Capital	State
Paris	France
Madrid	Spain
Berlin	Germany

Capitals

State	President
France	Emmanuel Macron
Germany	Frank-Walter Steinmeier
Italy	Sergio Mattarella

Presidents

Capital	State	President
Paris	France	Emmanuel Macron
Berlin	Germany	Frank-Walter Steinmeier

Capitals $\bowtie$ Presidents

The information about *Italy* and *Spain* are lost.

Outer Join

Outer join has been extended from the natural join operation for avoiding information loss.

Left Outer Join / Left Join

$\mathcal{R} := \mathcal{R}_1 ⟕ \mathcal{R}_2$

Left Join makes a natural join but adds extra tuples to the result padded with NULL values to deal with missing information in the first relation.

Left Outer Join / Left Join

Capital	State
Paris	France
Madrid	Spain
Berlin	Germany

Capitals

State	President
France	Emmanuel Macron
Germany	Frank-Walter Steinmeier
Italy	Sergio Mattarella

Presidents

Capital	State	President
Paris	France	Emmanuel Macron
Madrid	Spain	NULL
Berlin	Germany	Frank-Walter Steinmeier

Capitals $⟕$ Presidents

Right Outer Join / Right Join

$\mathcal{R} := \mathcal{R}_1 ⋊ \mathcal{R}_2$

Right Join makes a natural join but adds extra tuples to the result padded with NULL values to deal with missing information in the second relation.

Right Outer Join / Right Join

Capital	State
Paris	France
Madrid	Spain
Berlin	Germany

Capitals

State	President
France	Emmanuel Macron
Germany	Frank-Walter Steinmeier
Italy	Sergio Mattarella

Presidents

Capital	State	President
Paris	France	Emmanuel Macron
Berlin	Germany	Frank-Walter Steinmeier
NULL	Italy	Sergio Mattarella

Capitals $⟖$ Presidents

Full Outer Join / Full Join

$\mathcal{R} := \mathcal{R}_1 ⟗ \mathcal{R}_2$

Full Join makes a natural join but adds extra tuples to the result padded with NULL values to deal with missing information in the first and in the second relations.

Full Outer Join / Full Join

Capital	State
Paris	France
Madrid	Spain
Berlin	Germany

Capitals

State	President
France	Emmanuel Macron
Germany	Frank-Walter Steinmeier
Italy	Sergio Mattarella

Presidents

Capital	State	President
Paris	France	Emmanuel Macron
Madrid	Spain	NULL
Berlin	Germany	Frank-Walter Steinmeier
NULL	Italy	Sergio Mattarella

Capitals ⟗ Presidents

Exercise #1

Suppose there exists a pair of relations $\mathcal{R}_1(X, Y)$ and $\mathcal{R}_2(X, Y)$ having $t_1 > 0$ and $t_2 > 0$ tuples, respectively.
Find out the minimum and maximum possible number of tuples that may appear in the resulting relations provided by the following operations.

- $\mathcal{R}_1 \cup \mathcal{R}_2$
- $\mathcal{R}_1 \cap \mathcal{R}_2$
- $\mathcal{R}_1 - \mathcal{R}_2$
- $\mathcal{R}_1 \times \mathcal{R}_2$

Exercise #1

Suppose there exists a pair of relations $\mathcal{R}_1(X, Y)$ and $\mathcal{R}_2(X, Y)$ having $t_1 > 0$ and $t_2 > 0$ tuples, respectively.

Expression	Minimum tuples	Maximum tuples
$\mathcal{R}_1 \cup \mathcal{R}_2$		
$\mathcal{R}_1 \cap \mathcal{R}_2$		
$\mathcal{R}_1 - \mathcal{R}_2$		
$\mathcal{R}_1 \times \mathcal{R}_2$		

télécom saint-étienne

Exercise #2

A database of a building management syndicate with the schema:

- Building (**id**, name, address)
- Apartment (**id** , no , surface , level , *idBuilding*)
- Person (**id**, first name , last name , profession , *apartmentid*)
- Owner (**idPerson** , **idAppart**, quotePart)

Exersice from: http://sql.bdpedia.fr/

Exercise #2

id	name	address
1	Koudalou	3 rue des Martyrs
2	Barabas	2 allée du Grand Turc

Building

idPerson	idAppart	quotePart
1	100	33
5	100	67
1	101	100
5	102	100
1	202	100
5	201	100
2	103	100

Owner

id	no	surface	level	idBuilding
100	1	150	14	1
101	34	50	15	1
102	51	200	2	1
103	52	50	5	1
104	43	75	3	1
200	1	150	0	2
201	2	250	1	2
202	3	250	2	2

Apartment

Exercise #2

id	first name	last name	profession	apartmentid
1		Prof	Enseignant	202
2	Alice	Grincheux	Cadre	103
3	Léonie	Atchoum	Stagiaire	100
4	Barnabé	Simplet	Acteur	102
5	Alphonsine	Joyeux	Rentier	201
6	Brandon	Timide	Rentier	104
7	Don-Jean	Dormeur	Musicien	200

Person

Exercise #2

Express the following queries in relational algebra:

- Who are the owners of Atchoum's apartment?
- In which buildings does an actor live?
- Who lives in an apartment of less than 70 m^2 ?
- Who owns, at least partially, the apartment he occupies?
- In which buildings are there no musicians?
- Who owns an apartment without occupying it ?

RelaX – relational algebra calculator: https://dbis-uibk.github.io/relax