1222·2022 800 ANNI

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Extended Relational Algebra

Basi di Dati

Bachelor's Degree in Computer Engineering
Academic Year 2024/2025

Ornella Irrera

Intelligent Interactive Information Access (IIIA) Hub
Department of Information Engineering
University of Padua

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Outline

- Extended relational algebra

- Some examples

Introduction

- In traditional relational algebra, operations evaluate one tuple at time, and **independently** of all the other tuples in the relation.
- When we query some information to the database, we may look for some information which is not necessarily stored as a value of an attribute.
- **Selections, projections, joins** may not be enough.

Introduction

Pizzeria

Pizza	Prezzo	Quantità
Margherita	8	10
Capricciosa	11	20
Funghi	9	2
Marinara	6	13

- How can we compute the **total income** for each pizza?

Introduction

Pizzeria

Pizza	Prezzo	Quantità
Margherita	8	10
Capricciosa	11	20
Funghi	9	2
Marinara	6	13

- How can we compute the **income** for each pizza?
- We need to be able to compute operations between the attributes of our relations and, in general, compute **aggregation functions on grouped data.**

Introduction

Pizzeria

ID	Sede	Pizza	Prezzo	Quantità
1	Vicenza	Margherita	8	10
2	Padova	Capricciosa	11	20
3	Padova	Margherita	9	2
4	Vicenza	Marinara	6	13

- How can we compute the **income** for each location?
- We need to be able to compute **aggregation functions on grouped data.**

Generalized projection

The generalized projection extends the standard projection by allowing transformations or computations on attributes.

- Instead of just selecting columns, it can apply functions or expressions to generate new values in the result set.
- The projection will be characterized by:
  - the attributes declared in the projection, and belonging to the relation R;
  - one or more functions on the attributes of R.

Generalized projection

Pizzeria

Pizza	Prezzo	Quantità
Margherita	8	10
Capricciosa	10	20
Funghi	9	2
Marinara	6	13

- We want to compute the income related to each type of pizza.

$$\pi_{pizza,prezzo*quantita'}(Pizzeria)$$

Generalized projection

Pizzeria

Pizza	Prezzo	Quantità
Margherita	8	10
Capricciosa	10	20
Funghi	9	2
Marinara	6	13

$$\pi_{pizza,prezzo*quantita'}(Pizzeria)$$

Pizza	
Margherita	80
Capricciosa	200
Funghi	18
Marinara	78

Generalized projection

Pizzeria

Pizza	
Margherita	80
Capricciosa	200
Funghi	18
Marinara	78

- $\rho_{totale \leftarrow prezzo*quantita'}(\pi_{pizza,prezzo*quantita'}(Pizzeria))$

Pizza	Totale
Margherita	80
Capricciosa	200
Funghi	18
Marinara	78

Aggregation functions

- The aggregation functions are not included in the basic operations and serve as an extension to perform computations on (groups of) attributes of a relation.

- In the context of relational algebra, the aggregation functions are functions used to perform operations on the attributes of a relation

Aggregation functions

- Aggregation functions:
  - **SUM:** sum values contained in a column of a table
  - **MAX:** find the max value within the values of an attribute
  - **MIN:** find the min value within the values of an attribute
  - **AVERAGE:** Compute the average of the values of an attribute
  - **COUNT:** count the number of tuples in a relation. This operation is a bit different as it counts **tuples** and not the values of an attribute.

Aggregation Operation

Formally, given a relation $R(X)$ and given n aggregation functions $f_1(A_i), \ldots, f_n(A_i, A_j \ldots)$ with $A_i, A_j \in X$, the aggregation operation is:

$$\mathscr{F}_{f_1(A_i),\ldots,f_n(A_i,A_j,\ldots)}(R(X))$$

Where:

- $\mathscr{F}$ is the aggregation operator

The result is a **new relation** with n attributes (as many as the aggregation functions), with exactly **one tuple** which is the result of the operations.

Hence:

- degree = n
- cardinality = 1

Exercise

Find the total income and the max totale.

$$INCASSO \leftarrow \rho_{totale \leftarrow prezzo*quantita'}(\pi_{pizza,prezzo*quantita'}(Pizzeria))$$

Incasso

Pizza	Totale
Margherita	80
Capricciosa	200
Funghi	18
Marinara	78

$$\mathscr{F}_{SUM(totale),MAX(totale)}(INCASSO)$$

SUM(totale)	MAX(totale)
376	200

Exercise

Find the total income and the max totale.

$$\rho_{total\_income,max\_income}(\mathscr{F}_{SUM(totale),MAX(totale)}(INCASSO))$$

Total income	Max income
376	200

Grouping

So far, the aggregation operations involved the entire set of tuples in the relation.

However, we may want to compute aggregation operations on sets of tuples decided on one or more attributes.

Grouping

So far, the aggregation operations involved the entire set of tuples in the relation.

However, we may want to compute aggregation operations on sets of tuples decided on one or more attributes.

— **grouping** is an operation that divides a relation into groups based on the values of one or more attributes and then applies an aggregation function to each group.

Grouping Operation

Formally, given a relation $R(X)$, given n aggregation functions $f_1(A_i), \ldots, f_n(A_i, A_j \ldots)$ with $A_i, \ldots, A_j \in X$, given m grouping attributes the grouping operation is:

$$_{(A_1,\ldots,A_m)}\mathscr{F}_{f_1(A_1),\ldots,f_n(A_i,A_j,\ldots)}(R(X))$$

Where:

- $\mathscr{F}$ is the aggregation operator

- The attributes before the $\mathscr{F}$ are the grouping attributes, that are used to group the tuples in the relation $R(X)$

The result is a **new relation** with $n + m$ attributes (as many as the aggregation functions, plus the attributes used in the aggregation), with exactly **as many tuples as the number of groups**.

Hence:

- degree = $n + m$
- cardinality = number of groups

Exercise

Ristorante

Nome	Data	Prezzo_menu	Coperti
Ristorante_1	10/12/24	80	100
Ristorante_2	10/12/24	90	200
Ristorante_2	11/12/24	70	300
Ristorante_1	11/12/24	100	150
Ristorante_3	11/12/24	60	70

We collect the total of menus ordered in three restaurants in two days. We want to get, for each restaurant, the total and the average incomes.

Exercise

Ristorante

Nome	Data	Prezzo_menu	Coperti
Ristorante_1	10/12/24	80	100
Ristorante_2	10/12/24	90	200
Ristorante_2	11/12/24	70	300
Ristorante_1	11/12/24	100	150
Ristorante_3	11/12/24	60	70

$$\rho_{Ricavo \leftarrow Prezzo\_menu * coperto}((\pi_{Nome, Data, Prezzo\_menu * coperti}(Ristorante)))$$

Nome	Data	Ricavo
Ristorante_1	10/12/24	800
Ristorante_2	10/12/24	1800
Ristorante_2	11/12/24	2100
Ristorante_1	11/12/24	1500
Ristorante_3	11/12/24	4200

Exercise

$$INCASSO \leftarrow \rho_{Ricavo \leftarrow Prezzo\_menu*coperto}((\pi_{Nome,Data,Prezzo\_menu*coperti}(Ristorante)))$$

Incasso

Nome	Data	Ricavo
Ristorante_1	10/12/24	800
Ristorante_2	10/12/24	1800
Ristorante_2	11/12/24	2100
Ristorante_1	11/12/24	1500
Ristorante_3	11/12/24	4200

Exercise

$$INCASSO \leftarrow \rho_{Ricavo \leftarrow Prezzo\_menu*coperto}((\pi_{Nome,Data,Prezzo\_menu*coperti}(Ristorante)))$$

Incasso

Nome	Data	Ricavo
Ristorante_1	10/12/24	800
Ristorante_2	10/12/24	1800
Ristorante_2	11/12/24	2100
Ristorante_1	11/12/24	1500
Ristorante_3	11/12/24	4200

$$_{(nome)}\mathscr{F}_{SUM(Ricavo),AVERAGE(Ricavo)}(INCASSO)$$

Nome	SUM(Ricavo)	AVERAGE(Ricavo)
Ristorante_1	2300	1150
Ristorante_2	3900	1950
Ristorante_3	4200	4200

Exercise

$$INCASSO \leftarrow \rho_{Ricavo \leftarrow Prezzo\_menu*coperto}((\pi_{Nome,Data,Prezzo\_menu*coperti}(Ristorante)))$$

Incasso

Nome	Data	Ricavo
Ristorante_1	10/12/24	800
Ristorante_2	10/12/24	1800
Ristorante_2	11/12/24	2100
Ristorante_1	11/12/24	1500
Ristorante_3	11/12/24	4200

$${}_{(nome)}\mathcal{F}_{SUM(Ricavo),AVERAGE(Ricavo)}(INCASSO)$$

$$\rho_{Nome,Totale,Media}({}_{(nome)}\mathcal{F}_{SUM(Ricavo),AVERAGE(Ricavo)}(INCASSO))$$

Nome	Totale	Media
Ristorante_1	2300	1150
Ristorante_2	3900	1950
Ristorante_3	4200	4200

NULL Values and Aggregation operation

Student	Data	Mark	Course
123456	10/12/24	20	Analysis
123456	10/12/24	25	Databases
384752	11/12/24	NULL	Analysis
384752	11/12/24	18	Physics
3841345	11/12/24	NULL	Physics

The aggregation operations such as AVERAGE, MAX, MIN, SUM does not consider NULL values, and the result is computed considering exclusively the tuples whose values for the attributes in aggregation are **not null.**

NULL Values and Aggregation operation

Student	Data	Mark	Course
123456	10/12/24	20	Analysis
123456	10/12/24	25	Databases
384752	11/12/24	NULL	Analysis
384752	11/12/24	18	Physics
3841345	11/12/24	NULL	Physics

The aggregation operations such as AVERAGE, MAX, MIN, SUM do not consider NULL values, and the result is computed considering exclusively the tuples whose values for the attributes in aggregation are **not null.**

!!!! NULL DOES NOT MEAN 0!!!

NULL Values and Aggregation operation

Badge	Nome	Cognome	Car
123456	Mario	Rossi	Volvo
384762	Matteo	Verdi	NULL
382320	Marco	Verdi	NULL
596830	John	Rossi	NULL
192092	Mary	Bianchi	Fiat
384135	Isabel	Bianchi	Fiat

$$\mathscr{F}_{COUNT(*)}(Worker) = 6$$

$$\mathscr{F}_{COUNT(badge,cognome)}(Worker) = 6$$

$$\mathscr{F}_{COUNT(cognome,car)}(Worker) = 4$$

$$\mathscr{F}_{COUNT(car)}(Worker) = 2$$

NULL Values and Aggregation operation

Badge	Nome	Cognome	Car
123456	Mario	Rossi	Volvo
384762	Matteo	Verdi	NULL
382320	Marco	Verdi	NULL
596830	John	Rossi	NULL
192092	Mary	Bianchi	Fiat
384135	Isabel	Bianchi	Fiat

$$|_{(badge,Cognome)}\mathscr{F}_{COUNT(*)}(Worker)| = 6$$

Badge	Cognome	COUNT(*)
123456	Rossi	1
384762	Verdi	1
382320	Verdi	1
596830	Rossi	1
192092	Bianchi	1
384135	Bianchi	1

NULL Values and Aggregation operation

Badge	Nome	Cognome	Car
123456	Mario	Rossi	Volvo
384762	Matteo	Verdi	NULL
382320	Marco	Verdi	NULL
596830	John	Rossi	NULL
192092	Mary	Bianchi	Fiat
384135	Isabel	Bianchi	Fiat

$$\left|{}_{(Cognome,Car)}\mathscr{F}_{COUNT(*)}(Worker)\right| = 4$$

Cognome	Car	COUNT(*)
Rossi	Volvo	1
Verdi	NULL	2
Rossi	NULL	1
Bianchi	Fiat	2

NULL Values and Grouping operation

Badge	Nome	Cognome	Car
123456	Mario	Rossi	Volvo
384762	Matteo	Verdi	NULL
382320	Marco	Verdi	NULL
596830	John	Rossi	NULL
192092	Mary	Bianchi	Fiat
384135	Isabel	Bianchi	Fiat

$$\left|{}_{(Car)}\mathcal{F}_{COUNT(*)}(Worker)\right| = 3$$

Car	COUNT(*)
Volvo	1
Fiat	2
NULL	3