UNIVERSITÀ DEGLI STUDI DI PADOVA

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

- Update operation (insertion, modification, and deletion)
- Relational Algebra (query operations)

Update Operations

Operations

- We can classify the relational model operations into
  - **update**
  - **query**

- Given a database schema and a database instance
  - an **update** is a function which maps the given **database instance** into a **new database instance** which is **valid** with respect to all the integrity constraints
  - a **query** is a function which maps the given **database instance** into a **relation**

Insertion

- It may violate
  - domain constraints
  - tuple constraints
  - key constraints
  - referential integrity constraints
- Alternatives
  - **prevent the insertion**
  - try to correct the cause of the constraint violation

Update

- It may violate
  - domain constraints
  - tuple constraints
  - key constraints
  - referential integrity constraints
- Alternatives
  - prevent the update (***restrict***)
  - propagate the update (***cascade***)

Update: Cascade

Student

BadgeNumber	Name	Surname	BirthDate
6554	Mario	Rossi	05/12/1982
8765	Paolo	Neri	07/04/1979
9283	Luisa	Verdi	22/07/1983
3456	Marta	Rossi	14/02/1981

Exam

Student	Mark	Course
3456	30	03
3456	24	02
9283	28	01
6554	26	01

Course

Code	Title	Teacher
01	Analysis	Bianchi
02	Chemistry	Bruni
03	Physics	Rossi

Update: Cascade

Student

BadgeNumber	Name	Surname	BirthDate
6554	Mario	Rossi	05/12/1982
8765	Paolo	Neri	07/04/1979
9283	Luisa	Verdi	22/07/1983
3457	Marta	Rossi	14/02/1981

Exam

Student	Mark	Course
3456	30	03
3456	24	02
9283	28	01
6554	26	01

Course

Code	Title	Teacher
01	Analysis	Bianchi
02	Chemistry	Bruni
03	Physics	Rossi

Update: Cascade

Student

BadgeNumber	Name	Surname	BirthDate
6554	Mario	Rossi	05/12/1982
8765	Paolo	Neri	07/04/1979
9283	Luisa	Verdi	22/07/1983
3457	Marta	Rossi	14/02/1981

Exam

Student	Mark	Course
3457	30	03
3457	24	02
9283	28	01
6554	26	01

Course

Code	Title	Teacher
01	Analysis	Bianchi
02	Chemistry	Bruni
03	Physics	Rossi

Deletion

- It may violate
  - referential integrity constraints
- Alternatives
  - prevent the deletion (***restrict***)
  - propagate the deletion (***cascade***)
  - modify the values of the referenced attributes setting them to **NULL** (***set null***) or to a default value (***set default***)
    - note that, if the referenced attributes are part of the primary key, it is not possible to set them to NULL

Delete: Cascade

Student

BadgeNumber	Name	Surname	BirthDate
6554	Mario	Rossi	05/12/1982
8765	Paolo	Neri	07/04/1979
9283	Luisa	Verdi	22/07/1983
3456	Marta	Rossi	14/02/1981

Exam

Student	Mark	Course
3456	30	03
3456	24	02
9283	28	01
6554	26	01

Course

Code	Title	Teacher
01	Analysis	Bianchi
02	Chemistry	Bruni
03	Physics	Rossi

Update: Cascade

Student

BadgeNumber	Name	Surname	BirthDate
6554	Mario	Rossi	05/12/1982
8765	Paolo	Neri	07/04/1979
9283	Luisa	Verdi	22/07/1983

Exam

Student	Mark	Course
9283	28	01
6554	26	01

Course

Code	Title	Teacher
01	Analysis	Bianchi
02	Chemistry	Bruni
03	Physics	Rossi

Relational Algebra (query operations)

Relational Algebra

The relational algebra is a set of operators to manipulate whole relations: each operator processes one (or more) relations and produces as output a new relation

- Relational algebra operators can be **composed** to create **expressions**
- The relational algebra is a **procedural language**

Relational Algebra Operators

- Primitive operators
  - $\cup$ union
  - $-$ difference
  - $\times$ product
  - ρ rename
  - σ selection
  - π projection
- Derived operators
  - $\cap$ intersection
  - $\bowtie$ natural join
  - $\bowtie_\Theta$ theta join
  - $\overset{\leftarrow}{\bowtie}_\Theta$ $\overset{\rightarrow}{\bowtie}_\Theta$ $\overset{\leftrightarrow}{\bowtie}_\Theta$ outer join
- Set operators
  - $\cup$ union
  - $-$ difference
  - $\times$ product
  - $\cap$ intersection
- Relational operators
  - ρ rename
  - σ selection
  - π projection
  - $\bowtie$ natural join
  - $\bowtie_\Theta$ theta join
  - $\overset{\leftarrow}{\bowtie}_\Theta$ $\overset{\rightarrow}{\bowtie}_\Theta$ $\overset{\leftrightarrow}{\bowtie}_\Theta$ outer join

Compatibility to Union

Two relations $R_1(X_1, X_2, \ldots, X_n)$ and $R_2(Y_1, Y_2, \ldots, Y_n)$ are **compatible to union** if they have the **same degree** n and

$$\begin{cases} X_i = Y_i \\ \mathrm{dom}(X_i) = \mathrm{dom}(Y_i) \end{cases} \qquad 1 \leq i \leq n$$

- Two relations are compatible to union when they have the same number of attributes and each pair of attributes has the same name and domain
- For the intersection, union, and difference operators, the two involved relations must be compatible to union

Intersection

Given two relations $R_1(X)$ and $R_2(Y)$ compatible to union, their **intersection** is the relation

$$R_1 \cap R_2 = \{t \mid t \in R_1 \wedge t \in R_2\}$$

Graduated

Badge	Surname	Age
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Manager

Badge	Surname	Age
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Graduated ∩ Manager

Badge	Surname	Age
7432	Neri	54
9824	Verdi	45

Union

Given two relations $R_1(X)$ and $R_2(Y)$ compatible to union, their **union** is the relation

$$R_1 \cup R_2 = \{t \mid t \in R_1 \vee t \in R_2\}$$

Graduated

Badge	Surname	Age
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Manager

Badge	Surname	Age
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Graduated ∪ Manager

Badge	Surname	Age
7274	Rossi	42
7432	Neri	54
9824	Verdi	45
9297	Neri	33

Difference

Given two relations $R_1(X)$ and $R_2(Y)$ compatible to union, their difference is the relation

$$R_1 - R_2 = \{t \mid t \in R_1 \wedge t \notin R_2\}$$

Graduated

Badge	Surname	Age
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Manager

Badge	Surname	Age
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Graduated − Manager

Badge	Surname	Age
7274	Rossi	42

Difference

Given two relations $R_1(X)$ and $R_2(Y)$ compatible to union, their difference is the relation

$$R_1 - R_2 = \{t \mid t \in R_1 \wedge t \notin R_2\}$$

Graduated

Badge	Surname	Age
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Manager

Badge	Surname	Age
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Manager − Graduated

Badge	Surname	Age
9297	Neri	33

Difference

Given two relations $R_1(X)$ and $R_2(Y)$ compatible to union, their difference is the relation

$$R_1 - R_2 = \{t \mid t \in R_1 \wedge t \notin R_2\}$$

Graduated

Badge	Surname	Age
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Manager

Badge	Surname	Age
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Manager − Graduated

Badge	Surname	Age
9297	Neri	33

!=

Graduated − Manager

Badge	Surname	Age
7274	Rossi	42

NOT COMMUTATIVE!!!

Product (cartesian)

Given two relations $R_1(X)$ and $R_2(Y)$, their product (cartesian) is the relation

$$R_1 \times R_2 = \{xy \mid x \in R_1 \wedge y \in R_2\}$$

- Given the two relations $R_1(X_1, X_2, \ldots, X_n)$ and $R_2(Y_1, Y_2, \ldots, Y_m)$, the cartesian product

$$Q = R_1 \times R_2 = (X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_m)$$

- has degree $q = n + m$
- Let $|R_1| = n_{R_1}$ and $|R_2| = n_{R_2}$ be the cardinalities of the two relations, then the cardinality of the cartesian product is

$$|Q| = n_{R_1} * n_{R_2} = n_Q$$

- Note that the use of the term “cartesian product” is somehow improper because the result of this operation is not a pair of tuples but the set of tuples obtained concatenating each tuple of the first relation with each tuple of the second relation

Product (cartesian): example

Product (cartesian): example

Graduated

Badge	Surname	Age
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Manager

Badge	Surname	Age
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Graduated × Manager

Badge	Surname	Age	Badge	Surname	Age

Product (cartesian): example

Graduated

Badge	Surname	Age
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

Manager

Badge	Surname	Age
9297	Neri	33
7432	Neri	54
9824	Verdi	45

Graduated × Manager

Badge	Surname	Age	Badge	Surname	Age
7274	Rossi	42	9297	Neri	33
7274	Rossi	42	7432	Neri	54
7274	Rossi	42	9824	Verdi	45
7432	Neri	54	9297	Neri	33
7432	Neri	54	7432	Neri	54
7432	Neri	54	9824	Verdi	45
9824	Verdi	45	9297	Neri	33
9824	Verdi	45	7432	Neri	54
9824	Verdi	45	9824	Verdi	45

Rename

Given a relation $R(X)$ and two sets of attributes $A_1, A_2, \ldots, A_m \in X$ and $B_1, B_2, \ldots, B_m \notin X$, the rename is the relation with attributes $X - \{A_1, A_2, \ldots, A_m\} \cup \{B_1, B_2, \ldots, B_m\}$ such that

$$\rho_{B_1,B_2,\ldots,B_m \leftarrow A_1,A_2,\ldots,A_m}(R) = \{t \mid \text{exists } x \in R \text{ such that } t[B_1, B_2, \ldots, B_m] = x[A_1, A_2, \ldots, A_m] \wedge t[C] = x[C] \text{ if } C \neq A_1, A_2, \ldots, A_m\}$$

- This definition states that we are changing only the attribute names but not their domain or the associated values
  - we are modifying the schema but not the instance

Rename: Example (1/2)

Paternity

Father	Child
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternity

Mother	Child
Eva	Abele
Eva	Set
Sara	Isacco

Paternity ∪ Maternity ?

Rename: Example (1/2)

$\rho_{Parent \leftarrow Father}(Paternity)$

Parent	Child
Adamo	Abele
Adamo	Caino
Abramo	Isacco

Maternity

Mother	Child
Eva	Abele
Eva	Set
Sara	Isacco

Rename: Example (1/2)

$\rho_{Parent \leftarrow Father}(Paternity)$

Parent	Child
Adamo	Abele
Adamo	Caino
Abramo	Isacco

$\rho_{Parent \leftarrow Mother}(Maternity)$

Parent	Child
Eva	Abele
Eva	Set
Sara	Isacco

Rename: Example (1/2)

$\rho_{Parent \leftarrow Father}(Paternity)$

Parent	Child
Adamo	Abele
Adamo	Caino
Abramo	Isacco

$\rho_{Parent \leftarrow Mother}(Maternity)$

Parent	Child
Eva	Abele
Eva	Set
Sara	Isacco

$\rho_{Parent \leftarrow Father}(Paternity) \cup \rho_{Parent \leftarrow Mother}(Maternity)$

Parent	Child
Adamo	Abele
Adamo	Caino
Abramo	Isacco
Eva	Abele
Eva	Set
Sara	Isacco

Rename: Example (2/2)

Clerk

Surname	Office	Salary
Rossi	Roma	55
Neri	Milano	64

Worker

Surname	Factory	Wages
Bruni	Monza	45
Verdi	Latina	55

$\rho_{Site,\ Pay \leftarrow Office,\ Salary}(Clerk)$

$\cup$

$\rho_{Site,\ Pay \leftarrow Factory,\ Wages}(Worker)$

Surname	Site	Pay
Rossi	Roma	55
Neri	Milano	64
Bruni	Monza	45
Verdi	Latina	55

Selection

Given a relation $R(X)$ and a proposition Θ, the selection is the relation:

$$\sigma_{\Theta}(R) = \{t \mid t \in R \wedge \Theta\}$$

- The selection produces a result which
  - has the same schema as the operand
  - contains a subset of the tuples of the operand, i.e. those tuples which match the condition expressed by the proposition
- The proposition Θ can be defined as follows
  - $X_i\ \theta\ X_j$ with X_i and X_j attributes of R on the same domain and $\theta \in \{<, >, =, \neq, \leq, \geq\}$ comparison operator
  - $X_i\ \theta\ c$ with X_i attribute of R, θ comparison operator and $c \in \mathrm{dom}(X_i)$ constant
  - if ϕ and ψ are propositions, then also $\phi \wedge \psi,\ \phi \vee \psi,\ \neg\phi$ are propositions

Selection: Exercise (1/4)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find the employees who earn more than 50

Selection: Exercise (1/4)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find the employees who earn more than 50

$\sigma_{Salary>50}(Employee)$

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
5698	Rossi	Roma	64

Selection: Exercise (2/4)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find the employees who earn more than 50 and work in Milan

Selection: Exercise (2/4)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find the employees who earn more than 50 and work in Milan

$\sigma_{Salary>50 \wedge Branch = Milano}$ (Employee)

Badge	Surname	Branch	Salary
5998	Neri	Milano	64

Selection: Exercise (3/4)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find the employees who have the same surname as the branch where they work

Selection: Exercise (3/4)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find the employees who have the same surname as the branch where they work

$\sigma_{Surname = Branch}$ (Employee)

Badge	Surname	Branch	Salary
9553	Milano	Milano	44

Selection: Exercise (4/4)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find the employees who have the same surname as the branch where they work or have salary < 60 and branch Rome

Selection: Exercise (4/4)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find the employees who have the same surname as the branch where they work or have salary < 60 and branch Rome

$\sigma_{Surname = Branch \vee (Salary < 60 \wedge Branch = 'Roma')}$ (Employee)

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
9553	Milano	Milano	44

Projection

Given a relation $R(X)$ and a list of its attributes $A_1, A_2, \ldots, A_m$, the projection is the relation:

$$\pi_{A_1,A_2,\ldots,A_m}(R) = \{t[A_1, A_2, \ldots, A_m] \mid t \in R\}$$

- The projection produces a result which
  - has a subset of the attributes of the operand
  - contains (a subset of) the tuples of the operand
- The degree of a projection is m
- The cardinality of a projection is
  - less than or equal to the cardinality of the input relation; less than if the projection contains duplicated tuples which are removed because of the definition of relation
  - equal to the cardinality of the input relation if the attributes $A_1, A_2, \ldots, A_m$ are a super-key

Projection: Exercise (1/2)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find the badge number and the surname of all the employees

Projection: Exercise (1/2)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find the badge number and the surname of all the employees

$\pi_{Badge, Surname}$(Employee)

Badge	Surname
7309	Rossi
5998	Neri
9553	Milano
5698	Rossi

Projection: Exercise (2/2)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Projection: Exercise (2/2)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find surname and branch of all the employees

$\pi_{Surname, Branch}(Employee)$

Surname	Branch
Rossi	Roma
Neri	Milano
Milano	Milano

Selection and Projection

Employee

Badge	Surnamy	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

- Selection and projection are two “orthogonal”, i.e. complementary operators
- Selection performs a **horizontal decomposition**
- Projection performs a **vertical decomposition**

Selection and Projection

Employee

Badge	Surnamy	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

σ

- Selection and projection are two “orthogonal”, i.e. complementary operators
- Selection performs a **horizontal decomposition**
- Projection performs a **vertical decomposition**

Selection and Projection

Employee

Badge	Surnamy	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

σ

π

- Selection and projection are two “orthogonal”, i.e. complementary operators
- Selection performs a **horizontal decomposition**
- Projection performs a **vertical decomposition**

Selection and Projection: Exercise (1/2)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find badge number and surname of all the employees who earn more than 50

Selection and Projection: Exercise (1/2)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find badge number and surname of all the employees who earn more than 50

$\pi_{Badge,\ Surname}(\sigma_{Salary>50}(Employee))$

Badge	Surname
7309	Rossi
5998	Neri
5698	Rossi

Selection and Projection: Exercise (1/2)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find badge number and surname of all the employees who earn more than 50

$\pi_{Badge,\ Surname}(\sigma_{Salary>50}(Employee))$

=

$\sigma_{Sakary>50}(\pi_{Badge,\ Surname}(Employee))$

Selection and Projection: Exercise (1/2)

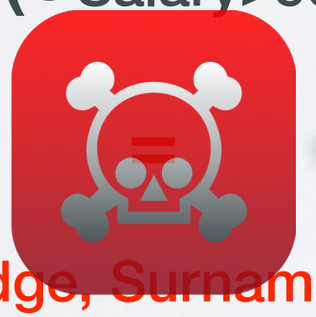
Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

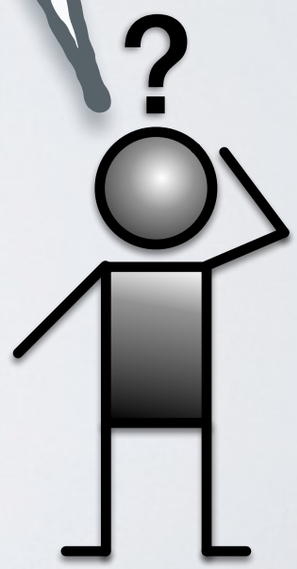
Find badge number and surname of all the employees who earn more than 50

$\pi_{Badge, Surname}(\sigma_{Salary>50}(Employee))$

In general, selection and projection are **not commutative**. They are commutative only when the attributes of the selection are a subset of the attributes of the projection



$\sigma_{Sakary>50}(\pi_{Badge, Surname}(Employee))$



Selection and Projection: Exercise (2/2)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find surname and branch of all the employees with salary < 50 **or** branch = 'Rome'

Selection and Projection: Exercise (2/2)

Employee

Badge	Surname	Branch	Salary
7309	Rossi	Roma	55
5998	Neri	Milano	64
9553	Milano	Milano	44
5698	Rossi	Roma	64

Find surname and branch of all the employees with salary < 50 **or** branch = 'Rome'

$\pi_{Surname,\ Branch}(\sigma_{Salary<50\ \vee\ Branch = 'Roma'}(Employee))$

Surname	Branch
Milano	Milano
Rossi	Roma