HALMSTAD UNIVERSITY

DS4001 Databases (7.5 credits)

Lecture 3 – Introduction to Relational Algebra

Yuantao Fan

yuantao.fan@hh.se

Halmstad University

Overview

- Review on Sets
  - Definition
  - Operations
  - Properties
- Introduction to Relational Algebra
  - Selection σ
  - Projection π
  - Union $\cup$
  - Intersection $\cap$
  - Product x
  - Join $\bowtie$

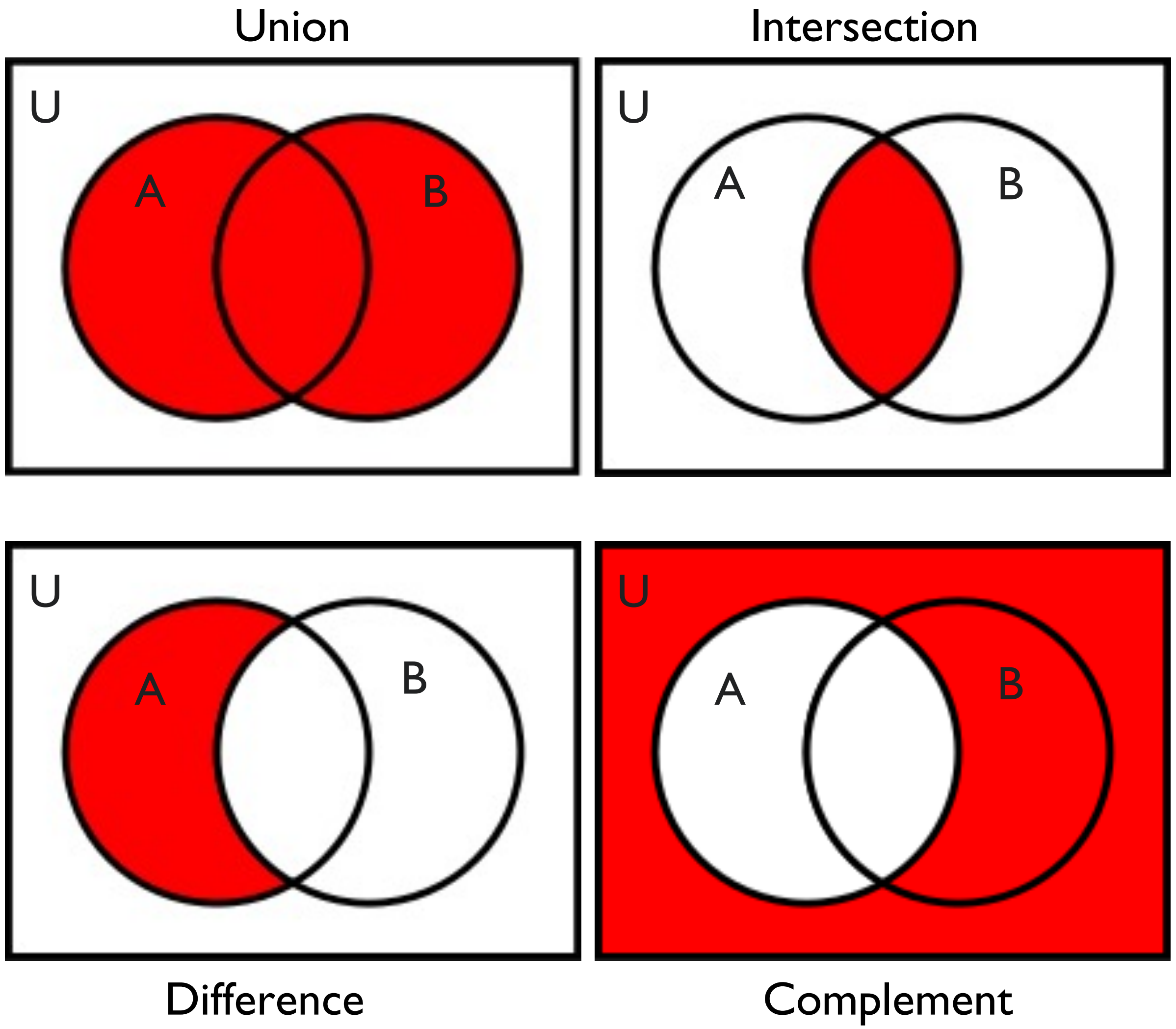
What is a set?

- A set is a well defined collection of distinct objects, called the "elements" or "members" of the set
  - A set might be finite or infinite
  - x is an element of set A is denoted by $x \in A$
  - Negated by writing $\mathrm{x} \notin \mathrm{A}$

- Notations
  - Enumeration – enumerate all elements (explicitly or implicitly)
    - Continent = {Asia, Africa, NorthAmerica, SouthAmerica, Antarctica, Europe, and Australia}
    - M = {2, 4, 6, 8, …, 100}
  - Set builder notation
    - { variable | descriptive statement }
    - E.g. Even numbers = { $2k$ | $k \in \mathbb{Z}$ }
    - Q ={ x | x = p / q where p and q are integers and q $\neq$0 }
  - Commonly used sets
    - N - set of all natural numbers
    - Z - set of all integers
    - Q - set of all rational numbers
    - …

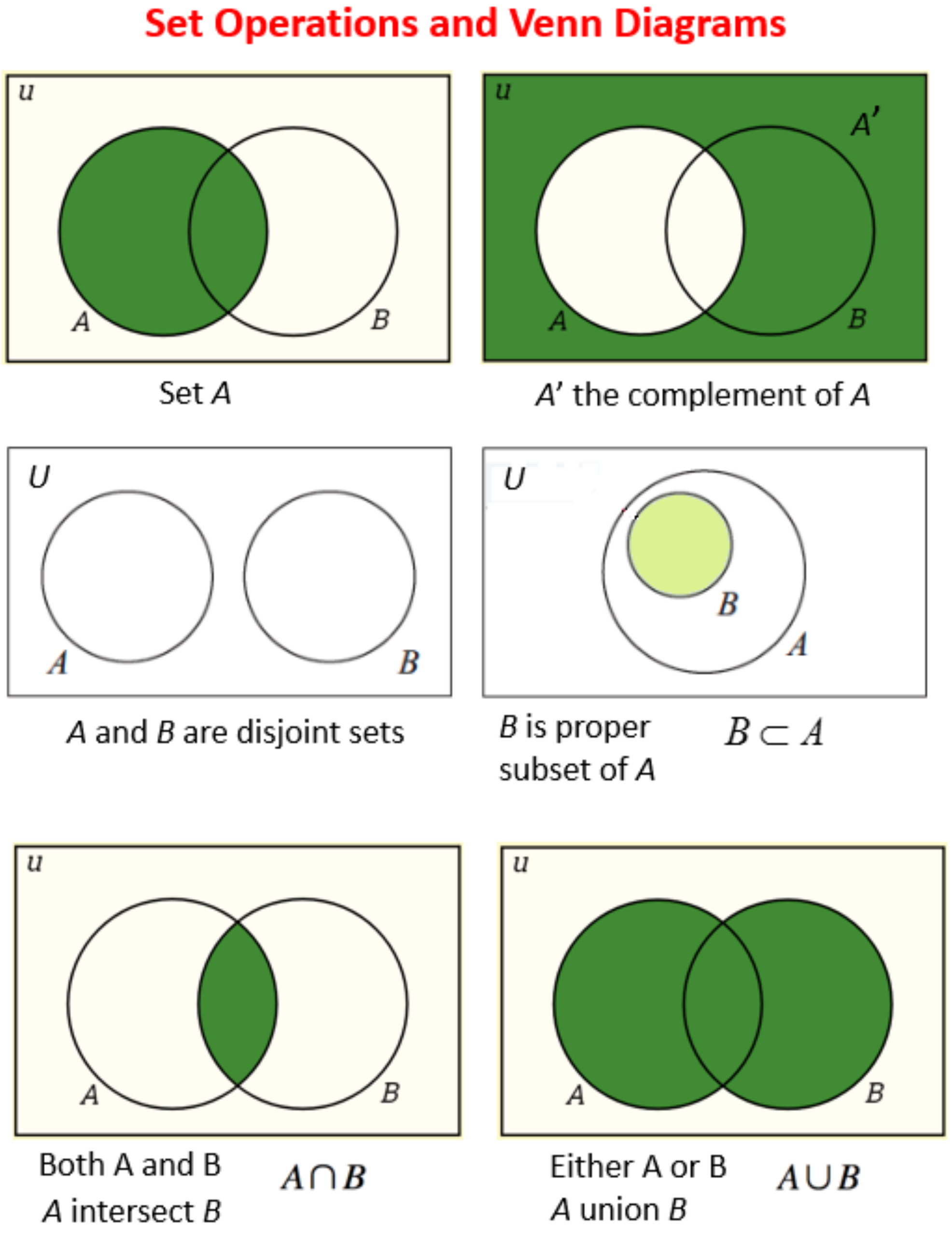
Properties and operations

- Union: $A \cup B = \{ x \mid x \in A \text{ or } x \in B \}$
- Intersection: $A \cap B = \{ x \mid x \in A \text{ and } x \in B \}$
- Difference: $U \setminus A = \{ x \mid x \in U \text{ and } x \notin A \}$
  - Also written as U - A
- Complement
  - Given a universal set U, U - A is also written as $\bar{A}$



Properties and operations

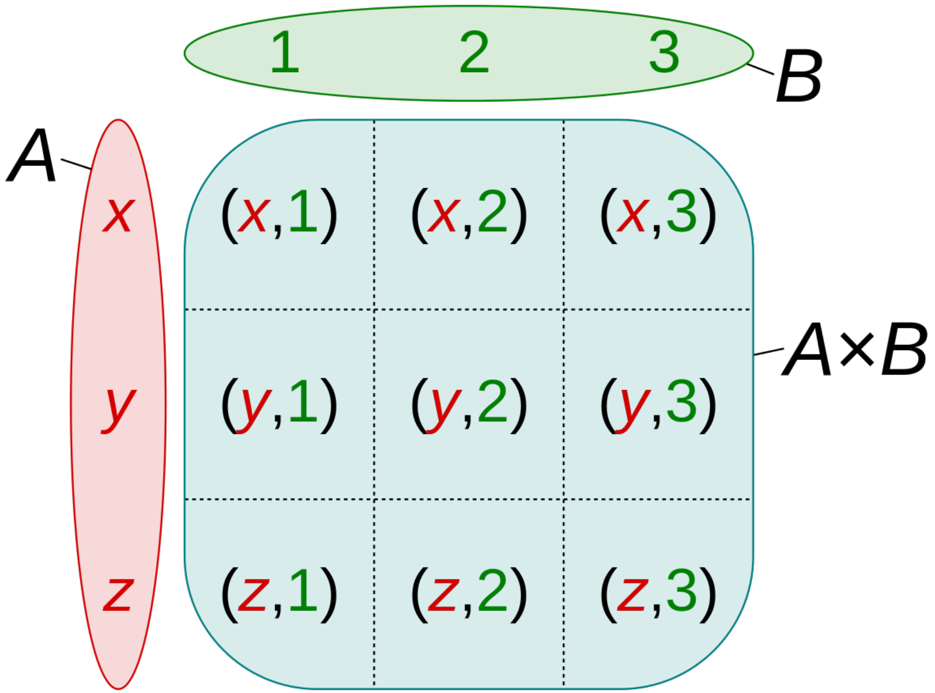
- Subset:
  - $A \subseteq B$ if for all $x \in A$ then $x \in B$
- Propoer subset:
  - $A \subset B$ if $A \subseteq B$ and $A \neq B$
- Equality:
  - $A = B$ if $A \subseteq B$ and $B \subseteq A$



Souce: https://www.onlinemathlearning.com/venn-diagrams.html

Cartesian Product

- The Cartesian product of two sets A and B is denoted as $A \times B$
  - $A \times B = \{\ (\mathrm{a}, b) \mid \mathrm{a} \in \mathrm{A} \text{ and } \mathrm{b} \in \mathrm{B}\ \}$
  - Note that (a, b) is ordered pairs
- Cardinality - numer of elements of the set
  - |A x B| = |A| x |B|
- Example
  - A deck of cards: 4 suits and 13-element cards



Source: https://en.wikipedia.org/wiki/Cartesian_product

Projection

- Given $(\mathrm{a}, b) \in \mathrm{A} \times \mathrm{B}$ the projections are defined as
  - $\pi_A(a, b) = a$
  - $\pi_B(a, b) = b$

- Example
  - Points in 2D-space: $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$
  - If point $\mathrm{r} \in \mathbb{R}^2$, then
  - $\pi_x\, r$ is the x coordinate
  - $\pi_y\, r$ is the y coordinate

Relations

- A relation R is a subset of the cartesian product of two or more sets
  - $R \subseteq A_1 \times A_2 \times \ldots \times A_n$
  - Equivalent notations
    - $(a_1, a_2, \ldots, a_n)$ satisfies R
    - $(a_1, a_2, \ldots, a_n) \in R$
    - $R\ (a_1, a_2, \ldots, a_n)$

- A binary relation $R \subseteq A \times B$ consists of a set of pairs
  - Denoted using aRb

Relations

- Let R be a relation on a set A, that is $R \subseteq A \times A$
  - R is Reflexive if for all $x \in A$. xRx
  - R is Symmetric if for all $x, y \in A$. xRy –> yRx
  - R is Transitive if for all $x, y, z \in A$. xRy and yRz -> xRz

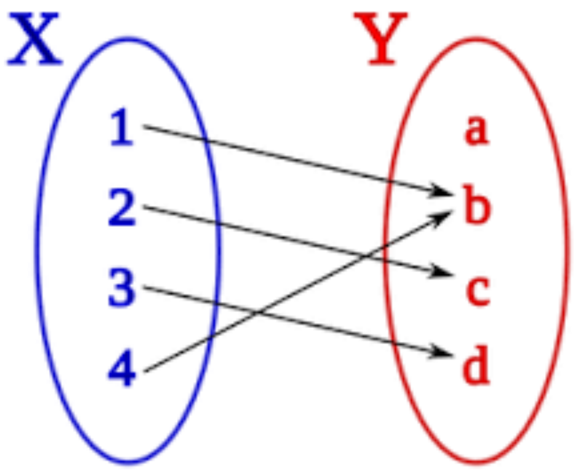
- Example - Let A = {0, 1, 2, 3} and a relation R on A be given by
  - R={ (0,0), (0,1), (0,3), (1,0), (1,1), (2,2), (3,0), (3,3) }
  - Is R reflexive? Symmetric? Transitive?

Relations

- Let R be a relation on a set A, that is $R \subseteq A \times A$
  - R is Reflexive if for all $x \in A$. xRx
  - R is Symmetric if for all x, $y \in A$. xRy –> yRx
  - R is Transitive if for all x, y, $z \in A$. xRy and yRz -> xRz
- Since relations are also sets
  - Set operations also apply

Functions

- A function from a set X to a set of Y, assigns to each element of X exactly on element of Y
  - X is the domain of the function
  - Denoted as $x \mapsto f(x)$
  - or f(x) =y
  - A function from A to B is also a relation f ⊆ A x B

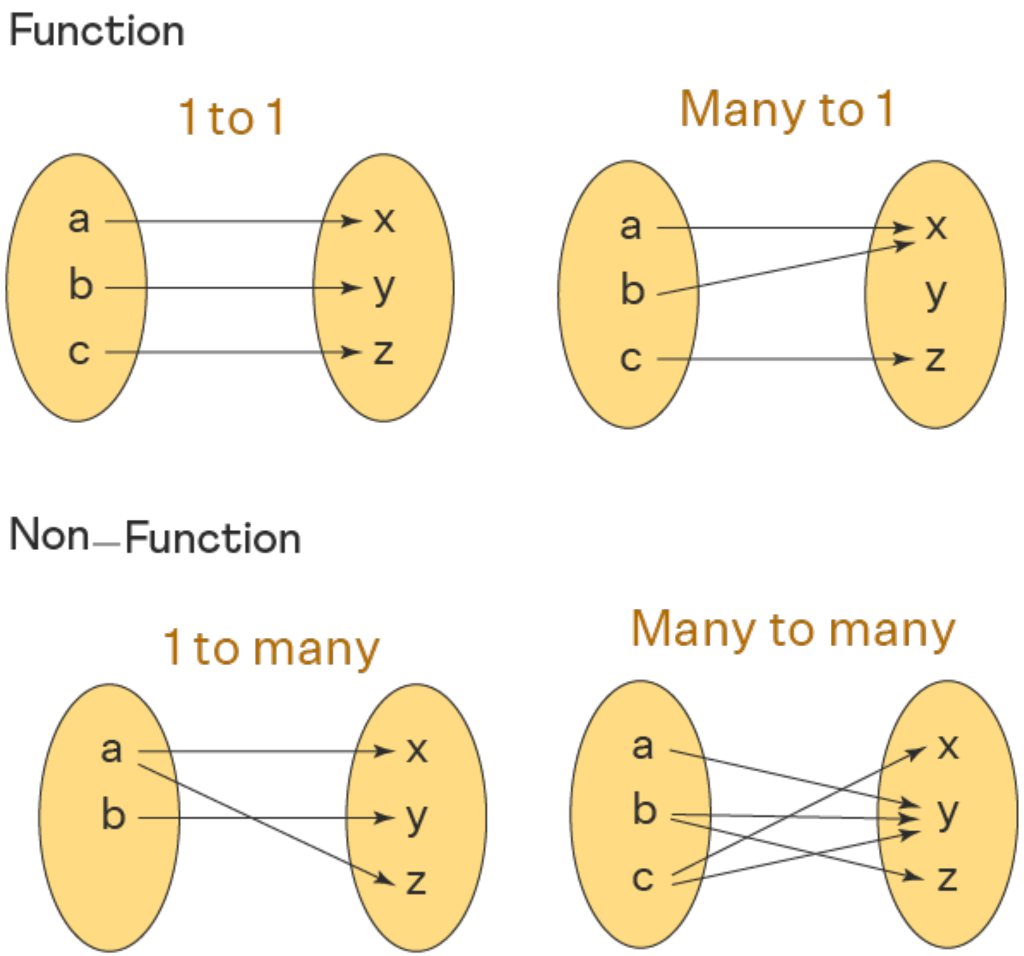


Function as mapping

- Example
  - Rsquare ⊆ Z x N such that Rsquare(n) = n^2

HALMSTAD UNIVERSITY

Identifying a function



Source: https://www.cuemath.com/calculus/What-are-functions/

Relational Model

- A relation is unordered set that contain the relation ship of attributes that represent entities
- A tuple is a set of attrubute values (domain) in the relation
- Example
  - (Intro to Computer Science, CS1310, 4, CS)

Course(Course_name, Course_code, Credit_hours, Teacher)

Course_name	Course_code	Credit_hours	Teacher
Intro to Computer Science	CS1310	4	11
Data Structures	CS3320	4	22
Discrete Mathematics	MATH2410	3	33
Database	CS3380	3	44

Relational Model - Primary Key

- Primary key is unique
  - identifies a single tuple
- If not specified
  - Some DBMS may create an internal primary key by default

Course(<u>id</u>, Course_name, Course_code, Credit_hours, Teacher)

id	Course_name	Course_code	Credit_hours	Teacher
1	Intro to Computer Science	CS1310	4	11
2	Data Structures	CS3320	4	22
3	Discrete Mathematics	MATH2410	3	33
4	Database	CS3380	3	44

Relational Model – Foreign Key

Teacher(id, First_name, Last_name)

id	First_name	Last_name
11	Walter	White
22	John	Smith

Course(id, Course_name, Course_code, Credit_hours, Teacher)

id	Course_name	Course_code	Credit_hours	Teacher
1	Intro to Computer Science	CS1310	4	11
2	Data Structures	CS3320	4	22
3	Discrete Mathematics	MATH2410	3	33
4	Database	CS3380	3	44

Relational Model – Foreign Key

Teacher(id, First_name, Last_name)

id	First_name	Last_name
11	Walter	White
22	John	Smith

Course(id, Course_name, Course_code, Credit_hours, Teacher)

id	Course_name	Course_code	Credit_hours	Teacher
1	Intro to Computer Science	CS1310	4	11
2	Data Structures	CS3320	4	22
3	Discrete Mathematics	MATH2410	3	33
4	Database	CS3380	3	44

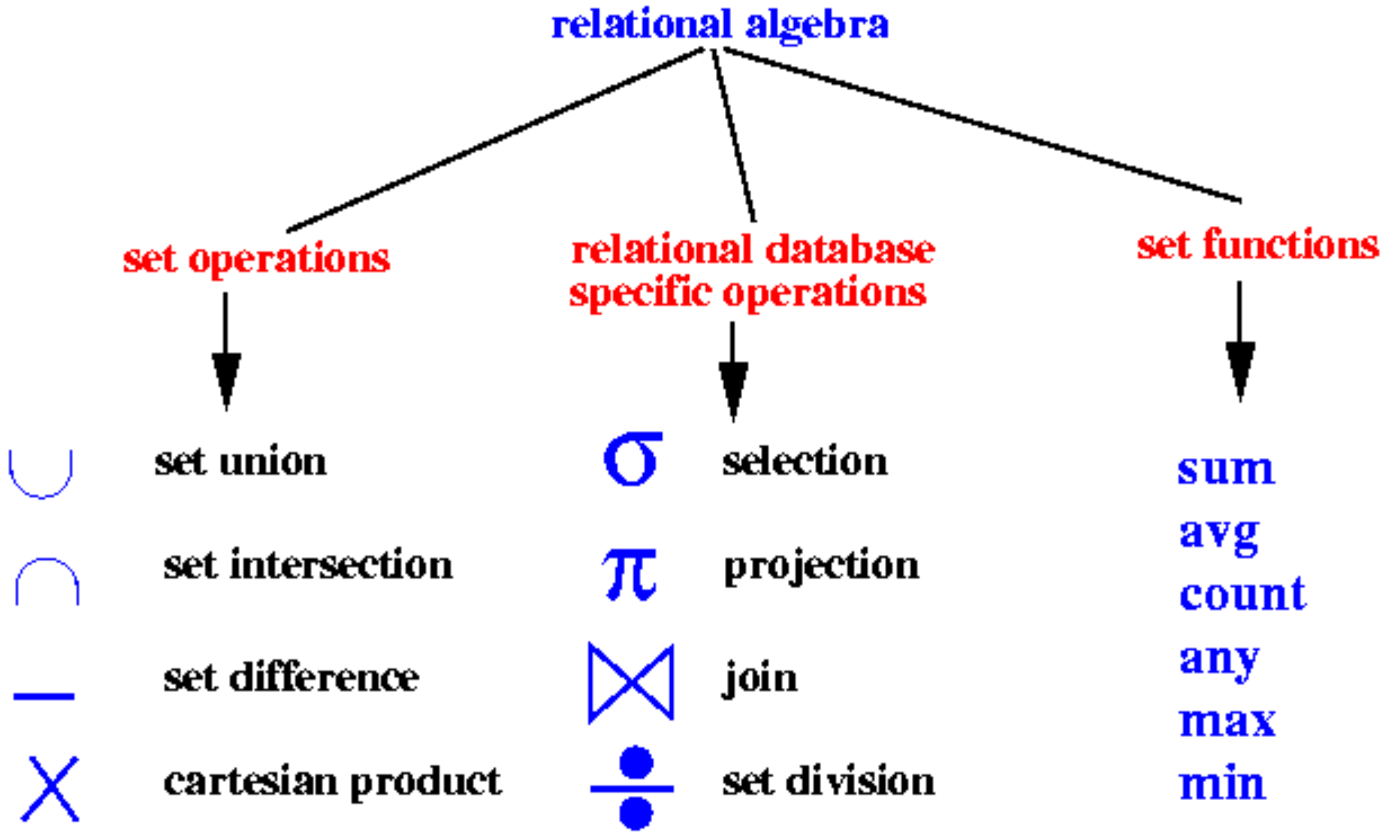
Teaches(Teacher_id, Course_id)

Teacher_id	Course_id
11	1
22	2
22	3
22	4

HALMSTAD UNIVERSITY

Relational Algebra

- Fundamental operations to retrieve and manipulate tupels in a relation



Source: https://medium.com/@sahirnambiar/relational-algebra-the-underpinnings-of-sql-74959481231a

Relational Algebra - Selection

- Syntax
  - $\sigma_{predicate}(R)$

- Choose a subset of the tuples from a relation that satisfies a selection predicate
  - Filtering using predicates
  - Combine multiple predicates

T(Tid, Cid)

Tid	Cid
a1	10
a2	11
a3	12
a3	13

$\sigma_{\text{Tid='a1'}}(T)$

Tid	Cid
a1	10

SELECT *
FROM T
WHERE Tid='a1';

$\sigma_{\text{Tid='a3'} \wedge \text{Cid='12'}}(T)$

Tid	Cid
a3	12

SELECT *
FROM T
WHERE Tid='a3' AND Cid='12';

Relational Algebra - Projection

- Syntax
  - $\pi_{A1,A2,\ldots An}(R)$
- Generate a relation with tuples that contains attributes specified
  - Can manipulate the values
  - Can rearrange the order

T(Tid, Cid)

Tid	Cid
a1	10
a2	11
a3	12
a3	13

$\pi_{\text{Tid,Cid*2}}\ (\sigma_{\text{Tid='a2'}}(\text{T}))$

Tid	Cid
a2	24

```
SELECT Tid, Cid*2
FROM T
WHERE Tid='a2';
```

Relational Algebra - Union

- Syntax
  - (A ∪ B)
- Collect all tuples in both relations

A(Tid, Cid)

Tid	Cid
a1	10
a2	11

B(Tid, Cid)

Tid	Cid
a2	11
a3	13

(A ∪ B)

Tid	Cid
a1	10
a2	11
a2	11
a3	13

```
(SELECT * FROM A)
UNION ALL
(SELECT * FROM B);
```

Relational Algebra - Intersection

- Syntax
  - (A ∩ B)

- Collect tuples that appear in both relations

A(Tid, Cid)

Tid	Cid
a1	10
a2	11

B(Tid, Cid)

Tid	Cid
a2	11
a3	13

(A ∩ B)

Tid	Cid
a2	11

```
(SELECT * FROM A)
INTERSECT
(SELECT * FROM B);
```

Relational Algebra - Product

- Syntax
  - (A x B)

- Generate a relation that contains all possible combinations of the tuples from both relations

SELECT * FROM A, B;

SELECT * FROM A CROSS JOIN B;

A(Tid, Cid)

Tid	Cid
a1	10
a2	11

B(Tid, Cid)

Tid	Cid
a2	11
a3	13

(A x B)

A.Tid	A.Cid	B.Tid	B.Cid
a1	10	a2	11
a1	10	a3	13
a2	11	a2	11
a2	11	a3	13

Relational Algebra - Join

- Syntax
  - (A $\bowtie$ B)

- Generate a relation that contains all tuples with a common value(s) of one (or more) attrubute(s)

A(Tid, Cid)

Tid	Cid
a1	10
a2	11

B(Tid, Cid)

Tid	Cid
a2	11
a3	13

(A $\bowtie$ B)

Tid	Cid
a2	11

SELECT * FROM A NATURAL JOIN B;

Next Lecture on Wednesday

- Some useful In-build functions & exercises
  - Ordering
  - Aggregation
  - Grouping
  - Having

- DML & DDL exercises

- Lab 1 intoduction

- Project – finding teammates (3 person per team)