Database Systems
(CS 355 / CE 373)

Dr. Umer Tariq
Assistant Professor,
Dhanani School of Science & Engineering,
Habib University

Acknowledgements

- Many slides have been borrowed from the official lecture slides accompanying the textbook:

  Database System Concepts, (2019), Seventh Edition,

  Avi Silberschatz, Henry F. Korth, S. Sudarshan

  McGraw-Hill, ISBN 9780078022159

  The original lecture slides are available at:

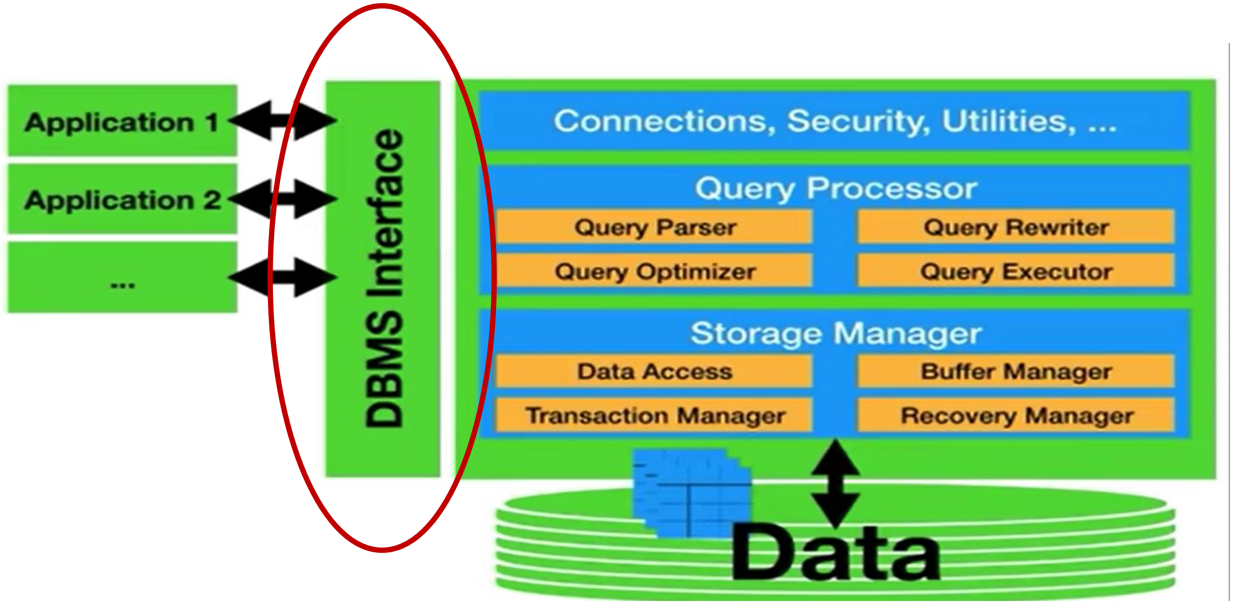
  https://www.db-book.com/

- Some of the slides have been borrowed from the lectures by Dr. Immanuel Trummer (Cornell University). Available at: (www.itrummer.org)

Outline: Week 5

- Query Languages
- Relational Algebra
- Relational Algebra Operations

What should be the DBMS Interface?



- Data Model
  - A collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.

The Relational Model

- The relational model uses <u>a collection of tables</u> to represent both data and the relationships among those data.

Table/ **Relation**
instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Column / Attribute
dept_name

Row / Tuple
(83821, Brandt, Comp. Sci., 92000)

Figure 2.1 The *instructor* relation.

Schema Design 101

- Identify Tables
- Identify Columns/Attributes associated with each table
- Identify Primary Keys
- Identify relationships among tables through Foreign Keys

A Sample Relational Model

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

"Database Languages" Supported by DBMS Interface

- The DBMS interface supports the following types of languages:

- Data Definition Language (DDL)
  - Used to define the Database Schema

- Data Manipulation Language (DML)
  - Used to Retrieve / manipulate data

“Database Languages” Supported by DBMS Interface

- The DBMS interface supports the following types of languages:

- Data Definition Language (DDL)
  - Used to define the Database Schema

- Data Manipulation Language (DML)
  - Used to Retrieve / manipulate data

- In practice, the DDL and DML are not two separate languages: instead they simply form parts of a <u>single database language</u>.

- For relational databases, the most popular database language is **Structured Query Language (SQL)**

Structured Query Language (SQL)

- The standard to access/retrieve/manipulate data in a **relational database**

- Examples of a Data Definition Language (DDL) Component

```
create table department
   (dept_name   char (20),
    building    char (15),
    budget      numeric (12,2));
```

- Examples of a Data Manipulation Language (DML) Component

```
select instructor.name
from instructor
where instructor.dept_name = 'History';
```

```
select instructor.ID, department.dept_name
from instructor, department
where instructor.dept_name= department.dept_name and
      department.budget > 95000;
```

Relational Model

$2x + 3y = 5$

$x + y$ $\quad 5 = 2 + 3$

Table3 = Table1 $\bowtie$ Table2

Relational "Algebra"

- A formal query language for the relational model.
- Provides a theoretical foundation for the Structured Query Language (SQL)

Relational Algebra: Basic Operations

- The relational algebra consists of a set of operations that take one or two relations as input and produce a new relation as their result.
    - Select (σ) ↔ WHERE
    - Project (Π) ↔ SELECT
    - Cartesian product (x) ↔ FROM Table1, Table2
    - Join ($\bowtie$) ↔ FROM TABLE1 inner join Table2 on ___ = ___
    - Union ($\cup$)
    - Intersection ($\cap$)
    - Set difference (-)
    - Rename (ρ)

- These operations enable a user to specify retrieval requests (or queries) as basic relational algebra expressions.

Basic Operations of Relational Algebra: 1) **Select (σ)**

- The **select** operation selects tuples that satisfy a given predicate.
  - Notation: $\sigma_p(r)$
  - p is called the selection "predicate"

σ_p (instructor)

- Example:
  - Select those tuples of the instructor relation where the instructor is in the "Physics" department.

instructor

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

- Query:
  - $\sigma_{dept\_name=\text{"Physics"}}(instructor)$

ID	name	dept_name	salary
22222	Ein...	Physics	95000
33456	Gold	Physics	87000

Basic Operations of Relational Algebra: **Select (σ)**

- The **select** operation selects tuples that satisfy a given predicate.
  - Notation: $\sigma_p(r)$
  - p is called the selection predicate

- Example:
  - Select those tuples of the instructor relation where the instructor is in the "Physics" department.

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

- Query:
  - $\sigma_{\text{dept\_name="Physics"}}(\textit{instructor})$

- Result:

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

Basic Operations of Relational Algebra: **Select (σ)**

- In the selection predicate, we allow comparisons using

  =, ≠, ≥, >, ≤, <.

- We can combine several predicates into a larger predicate by using the connectives:

  $\wedge$ (and), $\vee$ (or), $\neg$ (not)

- Example:
  - Form a query to find the instructors in Physics Department with a salary greater than $90,000
- Query:
  - $\sigma_{(\text{dept\_name="Physics"} \wedge \text{salary} > 90{,}000)}$ (Instructor)
- Result:

Instructor

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Basic Operations of Relational Algebra: **Select (σ)**

- In the selection predicate, we allow comparisons using

  $=, \neq, \geq, >, \leq, <$.

- We can combine several predicates into a larger predicate by using the connectives:

  $\wedge$ (and), $\vee$ (or), $\neg$ (not)

- Example:
  - Form a query to find the instructors in Physics Department with a salary greater than $90,000
- Query:
  - $\sigma_{\text{(dept\_name="Physics"} \wedge \text{ salary > 90,000)}}$ (Instructor)
- Result:

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Basic Operations of Relational Algebra: **Select (σ)**

- The select predicate may include comparisons between two attributes.

- Example:
  - Find all departments whose name is the same as their building name

- Query:
  - $\sigma_{\text{dept\_name = building}}$ (department)

- Result: ?

dept-nm	buildg	budget

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Basic Operations of Relational Algebra: 2) **Project (Π)**

- The **<u>project</u>** operation is a unary operation that returns its argument relation, with certain attributes left out.
  - Notation: $\Pi_{A1, A2, A3 \ldots, Ak}(r)$

    where $A1$, $A2$, …, Ak are attribute names and r is a relation name.

- The result is defined as the relation of k attributes obtained by erasing the attributes that are not listed.
- Duplicate rows removed from result, since relations are sets.
- Example:
  - Eliminate the *dept_name* attribute of *instructor*
- Query:
  - $\Pi_{ID,\ name,\ salary}(instructor)$
- Result:

Instructor

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Π_{dept_name} (department)   {(Physics), (Finance), (Hist) — .. }

dept_name
Physics
Finan
Hist.
Comp Sc
EE
Biology
Comp Sc

Basic Operations of Relational Algebra: **Project (Π)**

- The **project** operation is a unary operation that returns its argument relation, with certain attributes left out.
  - Notation: $\Pi_{A1, A2, A3 \ldots, Ak}(r)$
    where *A1*, *A2*, …, *Ak* are attribute names and *r* is a relation name.

- The result is defined as the relation of *k* attributes obtained by erasing the attributes that are not listed.
- Duplicate rows removed from result, since relations are sets.
- Example:
  - Eliminate the *dept_name* attribute of *instructor*
- Query:
  - $\Pi_{ID,\ name,\ salary}$ (*instructor*)
- Result: →

department

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

ID	*name*	*salary*
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

Relational Algebra: Composition of Relational Operators

- The result of a relational-algebra operation is relation and therefore two or more relational-algebra operations can be composed together into a **relational-algebra expression**.

- Example:
  - Find the names of all instructors in the Physics department.
- Query:

$$\pi_{name}\left(\sigma_{dept\_name = \text{'Physics'}}\left(instructor\right)\right)$$

instruct.

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Relational Algebra: Composition of Relational Operators

- The result of a relational-algebra operation is relation and therefore two or more relational-algebra operations can be composed together into a **relational-algebra expression**.

- Example:
  - Find the names of all instructors in the Physics department.
- Query:

$$\Pi_{name}(\sigma_{dept\_name\ =\ \text{“Physics”}}\ (instructor))$$

- Result:

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Basic Operations of Relational Algebra: 3) **Cartesian Product (x)**

- The **Cartesian-product** operation (denoted by x) allows us to combine information from any two relations.
  - Notation: *instructor* x *teaches*

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure 2.1 The *instructor* relation.

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

Figure 2.7 The *teaches* relation.

instructor x *teaches* →

instructor.ID	name	dept_name	salary	teaches.ID	course_id	sec_id	semester	year
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
10101 ...	...	...	...	98345 ...	EE-181 ...	1 ...	Spring ...	2017 ...
12121	Wu	Finance	90000	10101	CS-101	1	Fall	2017
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2018
12121	Wu	Finance	90000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
12121	Wu	Finance	90000	15151	MU-199	1	Spring	2018
12121	Wu	Finance	90000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
15151	Mozart	Music	40000	10101	CS-101	1	Fall	2017
15151	Mozart	Music	40000	10101	CS-315	1	Spring	2018
15151	Mozart	Music	40000	10101	CS-347	1	Fall	2017
15151	Mozart	Music	40000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
15151	Mozart	Music	40000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
22222	Einstein	Physics	95000	10101	CS-101	1	Fall	2017
22222	Einstein	Physics	95000	10101	CS-315	1	Spring	2018
22222	Einstein	Physics	95000	10101	CS-347	1	Fall	2017
22222	Einstein	Physics	95000	12121	FIN-201	1	Spring	2018
22222	Einstein	Physics	95000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

Basic Operations of Relational Algebra: **Cartesian Product (x)**

- The **Cartesian-product** operation (denoted by x) allows us to combine information from any two relations.
  - Notation: *instructor* x *teaches*

- The Cartesian-Product associates every tuple of *instructor* with every tuple of *teaches.*

- Since the instructor *ID* appears in both relations we distinguish between these attributes by attaching to the attribute the name of the relation from which the attribute originally came.
  - *instructor.ID*
  - *teaches.ID*

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
12121	Wu	Finance	90000	10101	CS-101	1	Fall	2017
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2018
12121	Wu	Finance	90000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
12121	Wu	Finance	90000	15151	MU-199	1	Spring	2018
12121	Wu	Finance	90000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
15151	Mozart	Music	40000	10101	CS-101	1	Fall	2017
15151	Mozart	Music	40000	10101	CS-315	1	Spring	2018
15151	Mozart	Music	40000	10101	CS-347	1	Fall	2017
15151	Mozart	Music	40000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
15151	Mozart	Music	40000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
22222	Einstein	Physics	95000	10101	CS-101	1	Fall	2017
22222	Einstein	Physics	95000	10101	CS-315	1	Spring	2018
22222	Einstein	Physics	95000	10101	CS-347	1	Fall	2017
22222	Einstein	Physics	95000	12121	FIN-201	1	Spring	2018
22222	Einstein	Physics	95000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

Basic Operations of Relational Algebra: **Cartesian Product (x)**

- The **Cartesian-product** operation (denoted by x) allows us to combine information from any two relations.
  - Notation: *instructor* x *teaches*

- Most of the resulting rows have information about instructors who did *not* teach a particular course.

- To get only those tuples of "*instructor* x *teaches* " that pertain to instructors and the courses that they taught:

$\sigma_{instructor.id = teaches.id}$ (*instructor* x *teaches*)

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
12121	Wu	Finance	90000	10101	CS-101	1	Fall	2017
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2018
12121	Wu	Finance	90000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
12121	Wu	Finance	90000	15151	MU-199	1	Spring	2018
12121	Wu	Finance	90000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
15151	Mozart	Music	40000	10101	CS-101	1	Fall	2017
15151	Mozart	Music	40000	10101	CS-315	1	Spring	2018
15151	Mozart	Music	40000	10101	CS-347	1	Fall	2017
15151	Mozart	Music	40000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
15151	Mozart	Music	40000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
22222	Einstein	Physics	95000	10101	CS-101	1	Fall	2017
22222	Einstein	Physics	95000	10101	CS-315	1	Spring	2018
22222	Einstein	Physics	95000	10101	CS-347	1	Fall	2017
22222	Einstein	Physics	95000	12121	FIN-201	1	Spring	2018
22222	Einstein	Physics	95000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

Basic Operations of Relational Algebra: **Cartesian Product (x)**

- The **Cartesian-product** operation (denoted by x) allows us to combine information from any two relations.
  - Notation: *instructor* x *teaches*

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
12121	Wu	Finance	90000	10101	CS-101	1	Fall	2017
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2018
12121	Wu	Finance	90000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
12121	Wu	Finance	90000	15151	MU-199	1	Spring	2018
12121	Wu	Finance	90000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
15151	Mozart	Music	40000	10101	CS-101	1	Fall	2017
15151	Mozart	Music	40000	10101	CS-315	1	Spring	2018
15151	Mozart	Music	40000	10101	CS-347	1	Fall	2017
15151	Mozart	Music	40000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
15151	Mozart	Music	40000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
22222	Einstein	Physics	95000	10101	CS-101	1	Fall	2017
22222	Einstein	Physics	95000	10101	CS-315	1	Spring	2018
22222	Einstein	Physics	95000	10101	CS-347	1	Fall	2017
22222	Einstein	Physics	95000	12121	FIN-201	1	Spring	2018
22222	Einstein	Physics	95000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

- To get only those tuples of "*instructor* x *teaches*" that pertain to instructors and the courses that they taught:

$$\sigma_{instructor.id = teaches.id} (instructor \times teaches)$$

- Result:

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
32343	El Said	History	60000	32343	HIS-351	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-101	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-319	1	Spring	2018
76766	Crick	Biology	72000	76766	BIO-101	1	Summer	2017
76766	Crick	Biology	72000	76766	BIO-301	1	Summer	2018
83821	Brandt	Comp. Sci.	92000	83821	CS-190	1	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-190	2	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-319	2	Spring	2018
98345	Kim	Elec. Eng.	80000	98345	EE-181	1	Spring	2017

Basic Operations of Relational Algebra: **Join ($\bowtie$)**

- The **join** operation allows us to combine a select operation and a **c**artesian product operation into a single operation.
  - Notation: Consider relations *r*(*R*) and *s*(*S*). Let θ be a predicate on attributes in the schema R ∪ S. The join operation $r \bowtie_\theta s$ is defined as follows:

$$\boxed{r \bowtie_\theta s = \sigma_\theta (r \times s)}$$

- Thus $\sigma_{instructor.id = teaches.id}$ *(instructor x teaches)* can equivalently be written as:

$$instructor \bowtie_{Instructor.id = teaches.id} teaches.$$

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure 2.1 The *instructor* relation.

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

Figure 2.7 The *teaches* relation.

instructor.ID	name	dept_name	salary	teaches.ID	course_id	sec_id	semester	year
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
32343	El Said	History	60000	32343	HIS-351	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-101	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-319	1	Spring	2018
76766	Crick	Biology	72000	76766	BIO-101	1	Summer	2017
76766	Crick	Biology	72000	76766	BIO-301	1	Summer	2018
83821	Brandt	Comp. Sci.	92000	83821	CS-190	1	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-190	2	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-319	2	Spring	2018
98345	Kim	Elec. Eng.	80000	98345	EE-181	1	Spring	2017

Give me the course ids all the courses: $\Pi_{course\_id}(\sigma_{name='Singh'}(instructor \bowtie_{inst.ID = teach.ID} department))$
were taught by Mr. Singh

Basic Operations of Relational Algebra: **Join (⋈)**

- The **join** operation allows us to combine a select operation and a **c**artesian product operation into a single operation.
  - Notation: Consider relations *r*(*R*) and *s*(*S*). Let θ be a predicate on attributes in the schema R ∪ S. The join operation $r \bowtie_\theta s$ is defined as follows:

$$\boxed{r \bowtie_\theta s = \sigma_\theta (r \times s)}$$

- Thus $\sigma_{instructor.id = teaches.id}$ ***(instructor x teaches)*** can equivalently be written as:

instructor ⋈ department
instructor.id = dept_name

$$instructor \bowtie_{Instructor.id = teaches.id} teaches.$$

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure 2.1 The *instructor* relation.

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

Figure 2.7 The *teaches* relation.

instructor.ID	name	dept_name	salary	teaches.ID	course_id	sec_id	semester	year
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
32343	El Said	History	60000	32343	HIS-351	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-101	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-319	1	Spring	2018
76766	Crick	Biology	72000	76766	BIO-101	1	Summer	2017
76766	Crick	Biology	72000	76766	BIO-301	1	Summer	2018
83821	Brandt	Comp. Sci.	92000	83821	CS-190	1	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-190	2	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-319	2	Spring	2018
98345	Kim	Elec. Eng.	80000	98345	EE-181	1	Spring	2017

5)

Basic Operations of Relational Algebra: **Union (∪)**

- The **union** operation allows us to combine two relations.
  - Notation: $r \cup s$

- Example
  - Find all courses taught in the Fall 2017 semester, or in the Spring 2018 semester, or in both:

- Query:

$$\Pi_{course\_id}\,(\sigma_{semester=\text{“Fall”} \wedge year=2017}(section)) \cup$$
$$\Pi_{course\_id}\,(\sigma_{semester=\text{“Spring”} \wedge year=2018}(section))$$

- Result:

Course_id
CS-101
CS-347
PHY-101

Course_id
CS-101
CS-315
CS-319
~~CS-319~~
FIN-201
MU-199

course_id	sec_id	semester	year	building	room_number	time_slot_id
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

Basic Operations of Relational Algebra: **Union ($\cup$)**

- The **union** operation allows us to combine two relations.
  - Notation: $r \cup s$

- Example
  - Find all courses taught in the Fall 2017 semester, or in the Spring 2018 semester, or in both:

- Query:

$$\Pi_{course\_id} (\sigma_{semester=\text{“Fall”} \land year=2017} (section)) \cup$$
$$\Pi_{course\_id} (\sigma_{semester=\text{“Spring”} \land year=2018} (section))$$

- Result:

course_id
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

Basic Operations of Relational Algebra: **Union (∪)**

- The **union** operation allows us to combine two relations.
  – Notation: $r \cup s$
- For $r \cup s$ to be valid
  1. *r, s* must have the *same* **arity** (same number of attributes)
  2. The attribute domains must be **compatible** (example: 2nd column of *r* deals with the same type of values as does the 2nd column of *s*)

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

b)

Basic Operations of Relational Algebra: **Intersection ($\cap$)**

- The **intersection** operation allows us to find tuples that are in both the input relations.
  - Notation: $r \cap s$
- Example
  - Find all courses taught in both the Fall 2017 and the Spring 2018 semesters.
- Query:

$$\Pi_{course\_id}\,(\sigma_{semester=\text{"Fall"}\ \wedge\ year=2017}\,(section))\ \cap$$
$$\Pi_{course\_id}\,(\sigma_{semester=\text{"Spring"}\ \wedge\ year=2018}\,(section))$$

- Result:

course_id
CS-101
CS-347
PHY-101

course_id
CS-101
CS-319
MU-199

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

Basic Operations of Relational Algebra: **Intersection ($\cap$)**

- The **intersection** operation allows us to find tuples that are in both the input relations.
  - Notation: $r \cap s$
- Example
  - Find all courses taught in both the Fall 2017 and the Spring 2018 semesters.
- Query:

$$\Pi_{course\_id} (\sigma_{semester= \text{"Fall"} \land year=2017} (section)) \cap$$
$$\Pi_{course\_id} (\sigma_{semester= \text{"Spring"} \land year=2018} (section))$$

- Result:

course_id
CS-101

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

Basic Operations of Relational Algebra: **Intersection ($\cap$)**

- The **<u>intersection</u>** operation allows us to find tuples that are in both the input relations.
  - Notation: $r \cap s$
- For $r \cap s$ to be valid
  1. *r, s* must have the *same* **arity** (same number of attributes)
  2. The attribute domains must be **compatible** (example: 2nd column of *r* deals with the same type of values as does the 2nd column of *s*)

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

Basic Operations of Relational Algebra: **Set Difference (-)**

- The **set difference** operation allows us to find tuples that are in one relation but are not in another.
  - Notation: $r - s$
- Example
  - Find all courses taught in the Fall 2017 semester, but not in the Spring 2018 semester
- Query:

$$\Pi_{course\_id}\,(\sigma_{semester=\text{“Fall”} \wedge year=2017}\,(section)) - \Pi_{course\_id}\,(\sigma_{semester=\text{“Spring”} \wedge year=2018}\,(section))$$

- Result:

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

Basic Operations of Relational Algebra: **Set Difference (-)**

- The **<u>set difference</u>** operation allows us to find tuples that are in one relation but are not in another.
  - Notation: $r - s$
- Example
  - Find all courses taught in the Fall 2017 semester, but not in the Spring 2018 semester
- Query:

$$\Pi_{course\_id}\,(\sigma_{semester=\text{"Fall"}\ \wedge\ year=2017}\,(section)) - \Pi_{course\_id}\,(\sigma_{semester=\text{"Spring"}\ \wedge\ year=2018}\,(section))$$

- Result:

course_id
CS-347
PHY-101

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

Basic Operations of Relational Algebra: **Set Difference (-)**

- The **set difference** operation allows us to find tuples that are in one relation but are not in another.
  - Notation: *r - s*
- For *r - s* to be valid
  1. *r, s* must have the *same* **arity** (same number of attributes)
  2. The attribute domains must be **compatible** (example: 2nd column of *r* deals with the same type of values as does the 2nd column of *s*)

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

৪)

Basic Operations of Relational Algebra: **Assignment ($\leftarrow$)**

- It is convenient at times to write a relational-algebra expression by **assigning** parts of it to temporary relation variables.
  - Notation: The assignment operation is denoted by $\leftarrow$ and works like assignment in a programming language.

- Example
  - Find all instructor in the "Physics" and "Music" department.

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

- Query:

$$Physics \leftarrow \sigma_{dept\_name=\text{“Physics”}}(instructor)$$

$$Music \leftarrow \sigma_{dept\_name=\text{“Music”}}(instructor)$$

$$Physics \cup Music$$

- Result:

Basic Operations of Relational Algebra: **Rename (ρ)**

a)

- The results of relational-algebra expressions do not have a name that we can use to refer to them. The **rename** operator, ρ, is provided for that purpose
  - Notation: $\rho_x(E)$ returns the result of expression E under the name x
  - Another form of the rename operation:

$$\rho_{x(A1,A2,\,..\,An)}(E)$$

$\rho_W(\text{instructor})$

$\rho_I(\text{instructor})$

$W \times I$

instructor

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Equivalent Queries

- There is more than one way to write a query in relational algebra.

- Example
  - Find ~~information about courses taught by~~ instructors in the Physics department with salary greater than 90,000
- Query1

$$\sigma_{dept\text{-}name='Physics' \wedge salary>90{,}000}(Instructor)$$

- Query2

$$\sigma_{salary>90{,}000}\left(\sigma_{dept\text{-}name='Physics'}(Instructor)\right)$$

- The two queries are not identical; they are, however, equivalent -- they give the same result on any database.

Instructor

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Equivalent Queries

- There is more than one way to write a query in relational algebra.

- Example
  - Find information about courses taught by instructors in the Physics department with salary greater than 90,000
- Query1

  $\sigma_{dept\_name=\text{“Physics”} \wedge salary > 90{,}000}\,(instructor)$

- Query2

  $\sigma_{dept\_name=\text{“Physics”}}(\sigma_{salary > 90.000}\,(instructor))$

- The two queries are not identical; they are, however, equivalent -- they give the same result on any database.

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Summary

- Relational Algebra Operators are useful for formulating various queries in order to retrieve information from relations.
- Operators covered in this set of slides are: select, project, join, rename, union, intersection, cartesian product and set difference.
- Operators are both unary and binary.
- Operators can also be composed to form new operations.
- There are several ways to state the same query, which can be termed as equivalent.

Relational Algebra: Example

Consider the following relations for a database that keeps track of employees, customers, orders, products, and their details for a company:

- Customers (CustomerID, CustomerName, Phone, City)
- Orders (OrderID, CustomerID ,EmployeeID, OrderDate, ShipmentDestinationCity)
- Employees (EmployeeID, Name, Title, Phone, City)
- Products (ProductID, ProductName, UnitPrice, UnitsInStock)
- OrderDetails (OrderID , ProductID, Quantity, DiscountPercentage)

Specify the following queries in **relational algebra** on the database schema above:

(a) List the **IDs** and **names** of all customers that are based in "Karachi"

(b) List the **IDs** and **dates** of all orders that were shipped to 'Islamabad' and handled by the Employee with ID "1010"

(c) List **IDs** and **names** of all customers that have placed an order on "01-01-2023"

(d) List **IDs** and **names** of all customers that have placed an order on "01-01-2023", but have not placed an order on "01-01-22"

(e) List **IDs** of all the products that have ever been sold at a discount of greater than 15%

Relational Algebra: Example

Solution:

(a)	$\Pi_{CustomerID, CustomerName}(\sigma_{City="Karachi"}(Customers))$
(b)	$\Pi_{OrderID, OrderDate}(\sigma_{EmployeeID=1010 \wedge ShipmentDestinationCity="Islamabad"}(Orders))$
(c)	$\Pi_{Customers.CustomerID, Customers.CustomerName}(\sigma_{OrderDate=01-01-23}$ $(Customers \bowtie_{Customers.CustomerID=Orders.CustomerID} Orders))$
(d)	$a \leftarrow \Pi_{Customers.CustomerID, Customers.CustomerName}(\sigma_{OrderDate=01-01-23}$ $(Customers \bowtie_{Customers.CustomerID=Orders.CustomerID} Orders))$ $b \leftarrow \Pi_{Customers.CustomerID, Customers.CustomerName}(\sigma_{OrderDate=01-01-22}$ $(Customers \bowtie_{Customers.CustomerID=Orders.CustomerID} Orders))$ $a - b$
(e)	$\Pi_{Products.ProductID}(\sigma_{DiscountPercentage>15}$ $(Products \bowtie_{Products.ProductID=OrderDetails.ProductID} OrderDetails))$