

# SWUFE Database 2024-2025-2

## 0. Introduction

### 0.1 What is database

**base:** the main place where a person lives and works, or a place that a company does business from.

**database:** A database is an organized collection of data stored and accessed electronically from a computer

**DBMS:** A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data

**two base goal for a DBMS:** convenient and efficient

### 0.2 database application

- E-commerce
- Enterprise Organizations
- Standalone Application: SQLite

#### 0.2.1 application categories

- online transaction processing
  - low delay
- data analysis
  - e.g. beers and diapers

### 0.3 Role of the database

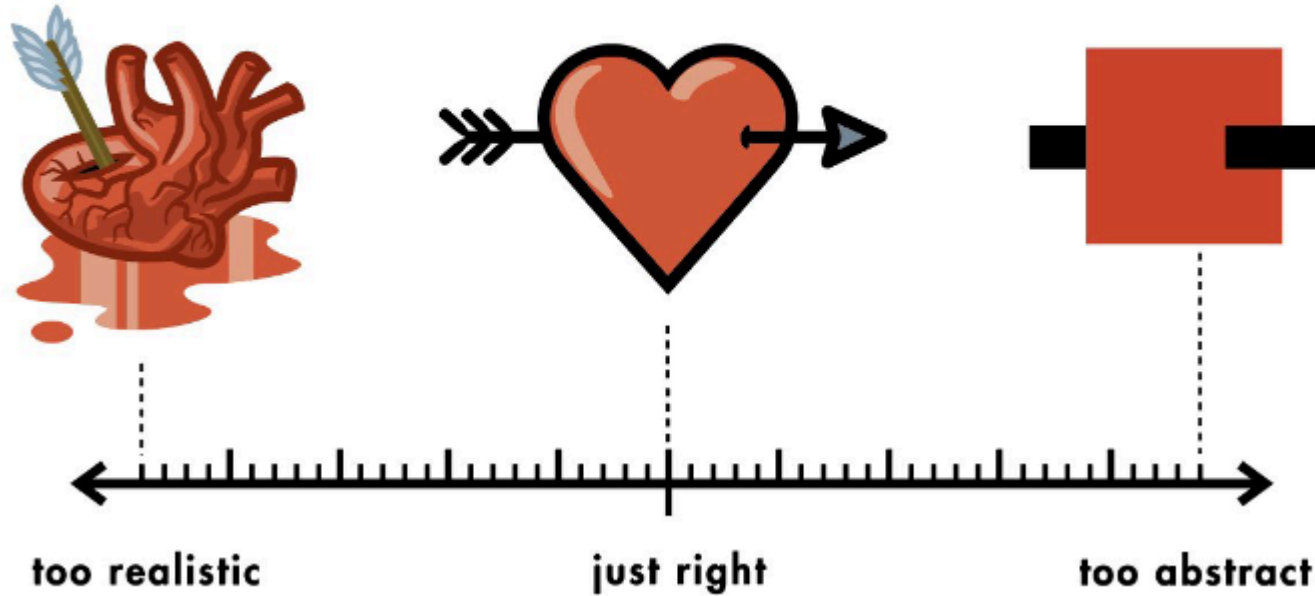
In other words, why file-based database is not as good as expected?

- **File processing system's disadvantage - data redundancy and inconsistency** - Data redundancy means higher storage costs, and multiple copies of the same data can lead to data inconsistencies. - **data isolation** - The data is scattered in different files and may also use different formats. Therefore, it is difficult to write new programs to access the data. - **difficulty in accessing data - integrity problem** - Certain values may be subject to certain constraints. For example,  $\text{salary} > 0$ . Although it is possible to implement the constraint by adding code to the program, it is not flexible enough. For example, new constraints may be added. - **concurrent-access anomalies** - Assuming you have \$100,000 in your account and two expenditures at the same moment (\$10,000 and \$20,000 respectively), the final result may be incorrect (\$90,000 or \$80,000). - **atomicity problems** - You are in the process of making a payment and the system crashes, at this point it may appear that your money is deducted but the object is not received. **Keep in mind the trade-off concept: a file-processing system and a database system (DBMS) are not completely opposed to each other; use the right system for the right situation.**
- When to use DBMS
  - highly valuable
  - relatively Larger
  - accessed by multiple users and applications

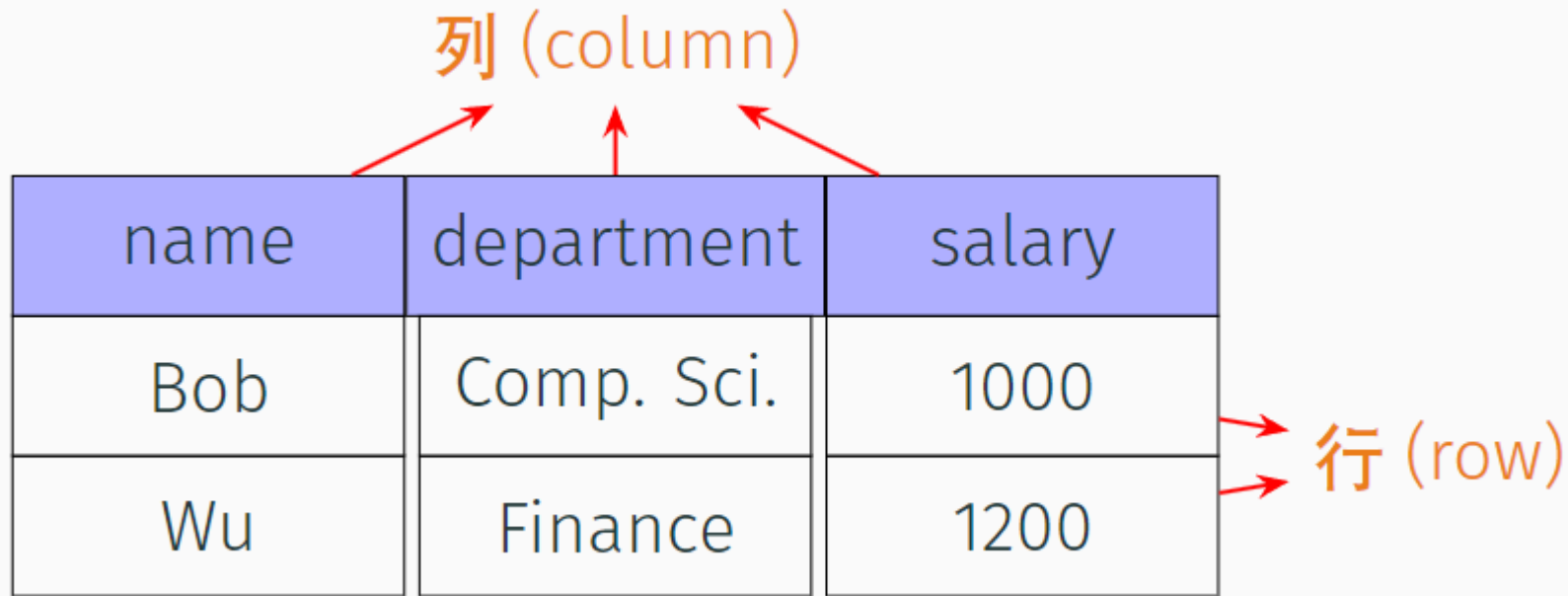
## 0.4 view of data

- Abstraction: help you to Ignore irrelevant details

# THE ABSTRACT-O-METER



- data model: A collection of conceptualization tools that describe data, data relationships, data semantics, and consistency constraints.
  - relational model: table is relation ← Most commonly used models



The diagram shows a table with three columns and two rows. The columns are labeled 'name', 'department', and 'salary'. The rows contain data for 'Bob' and 'Wu'. Red arrows point from the Chinese label '列 (column)' to the column headers, and from '行 (row)' to the data rows.

name	department	salary
Bob	Comp. Sci.	1000
Wu	Finance	1200

- entity-relationship model: the (E-R) data model uses a collection of basic objects, called entities, and relationships among these objects. ← Widely used in database design.
- semi-structured data model: individual data items of the same type may have different sets of attributes. ← Widely used in internet and big data scenarios
- object-based data model
- network model/ hierarchy model
- schema and instance
  - schema: The overall design of the database (translated as "綱要" in Taiwan) can be categorized into physical schema, logical schema, and sub-schema according to the different levels of data abstraction.
  - Instance: A collection of information stored in the database at a specific moment in time.

## 0.5 database language

The database system provides:

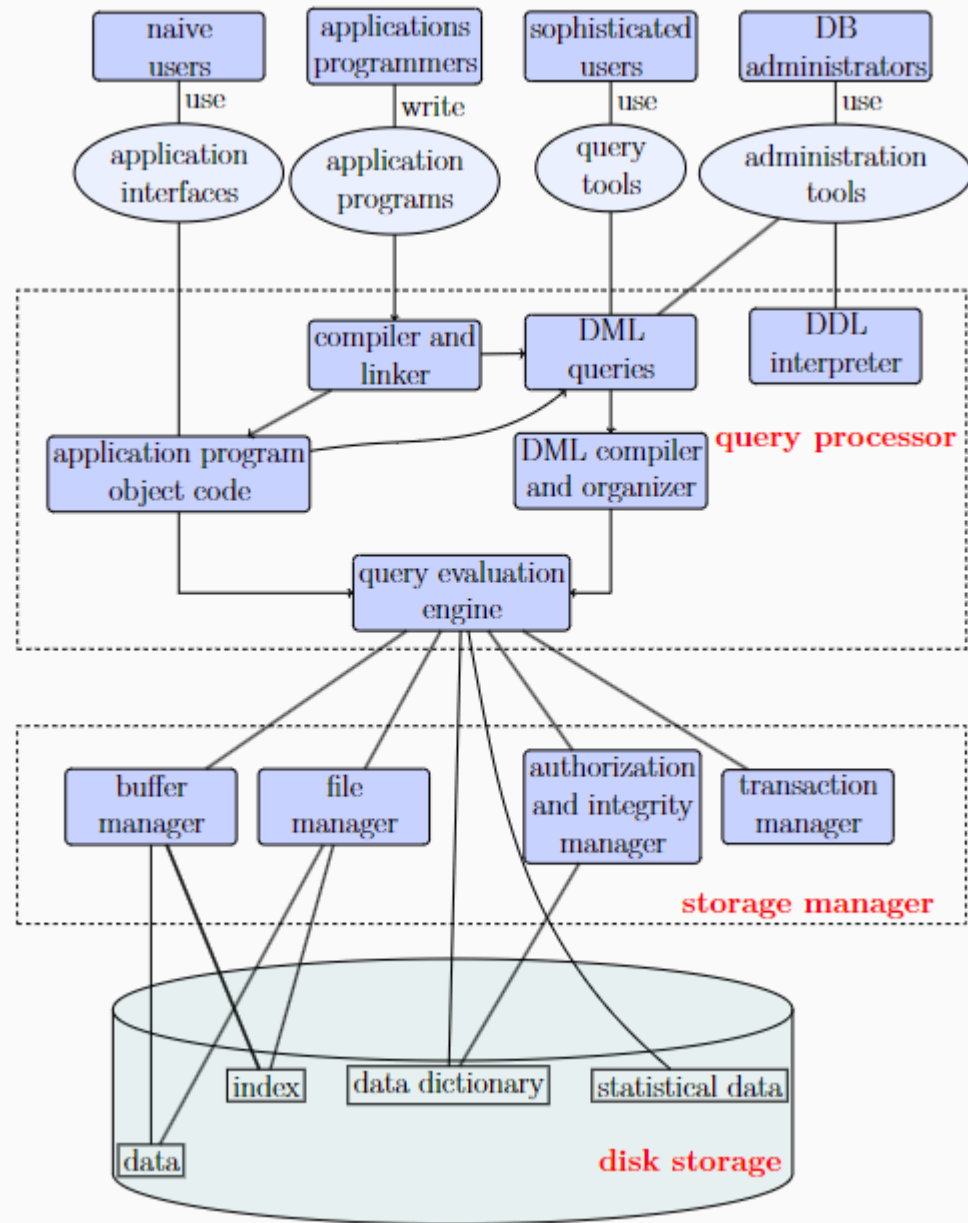
- data-definition language: defines the database schema
- data-manipulation language: expresses database queries and updates.

SQL (Structured Query Language) is the current mainstream. It is a declarative language, i.e., it focuses on What, not How.

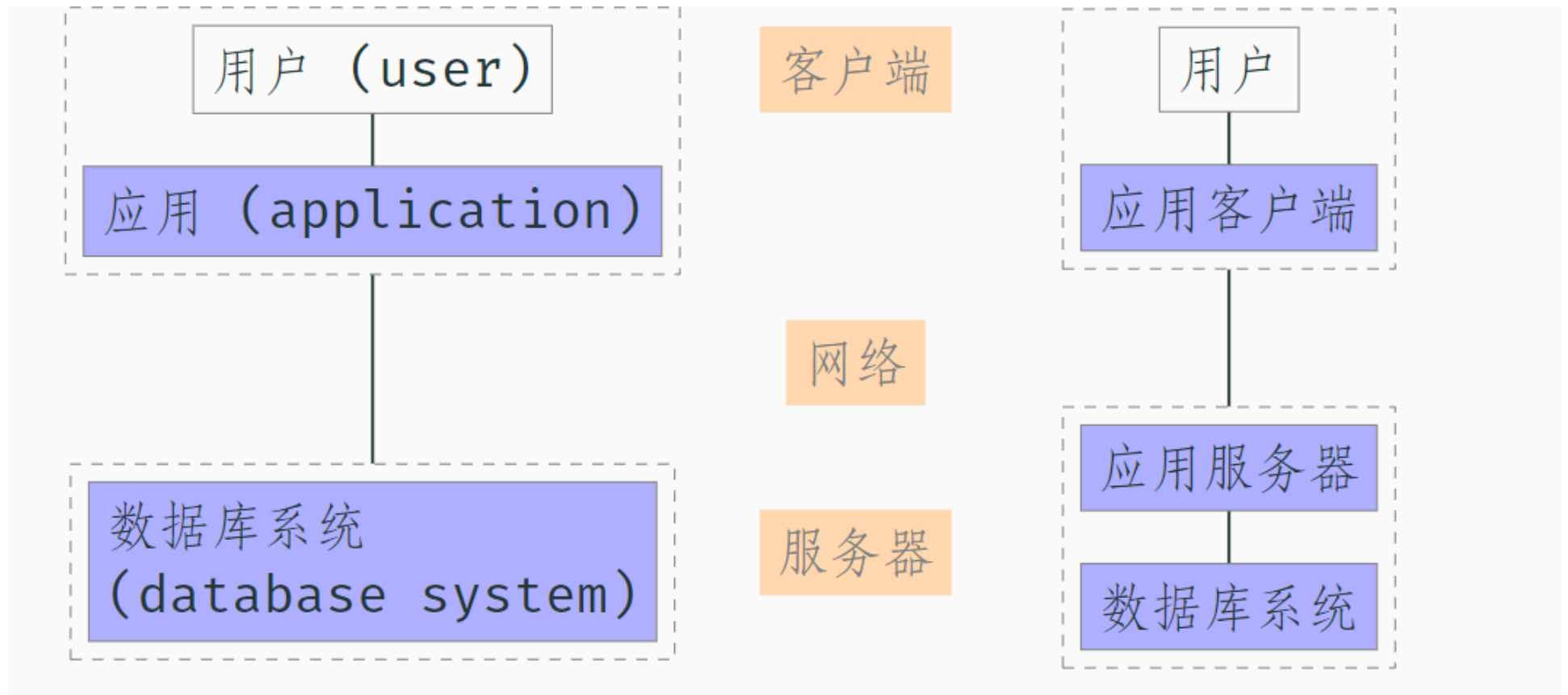
- DDL: Defines database structure (e.g., CREATE, ALTER, DROP).
- DML: Manipulates data within the database (e.g., SELECT, INSERT, UPDATE, DELETE).

## **0.6 Database System Architecture**

too complex to understand, and meaningless as well.



- Tips:



- It's silly to connect a database system directly to an application like this on the left

## 1. Relational model

consists of a collection of tables

- Relation:
  - Given sets  $X$  and  $Y$ , the Cartesian product  $X \times Y$  is defined as  $\{(x, y) | x \in X, y \in Y\}$  and its elements called ordered pairs
  - A binary relation  $R$  over sets  $X$  and  $Y$  is a subset of  $X \times Y$ .

## 1.1 the structure of relational model

database	Excel
relation	table
tuple	row
attribute	column

**A row in a table represents a relationship among a set of values.**



<i>course_id</i>	<i>title</i>	<i>dept_name</i>	<i>credits</i>
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4

The *course* table

<i>course_id</i>	<i>prereq_id</i>
BIO-301	BIO-101
BIO-399	BIO-101
CS-190	CS-101
CS-315	CS-101

Two courses are **related**.

The *prereq* table

The tuples here are not in order, but can be repeated

## 1.2 relation schema

- database schema: Logical design of the database
- database instance: A snapshot of the data in the database at a given moment Similarly, there is a relation schema and a relation instance.

- relation schema: The name of a relation and the set of attributes for a relation
  - e.g. The relation schema of this table

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
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

is **instructor**(ID, name, dept\_name, salary)

**all schema during this course can check in [reference/schema.pdf](#)**

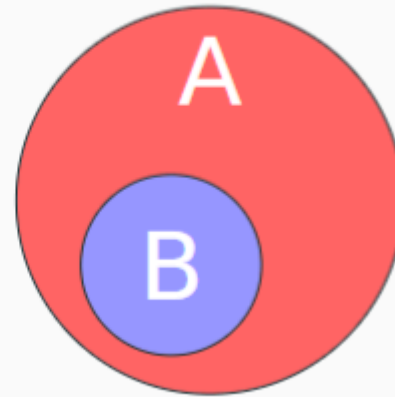
## 2. key

A way to **distinguish** between **different tuples** in a given relation.

- super key: A collection of **one or more attributes**, such that the combination of attributes allows us to **uniquely identify** a tuple in a relation.

people(name, age, origin, nationality, id)

- (id)
- (id, name)
- (id, age)
- (id, name, age)
- ...

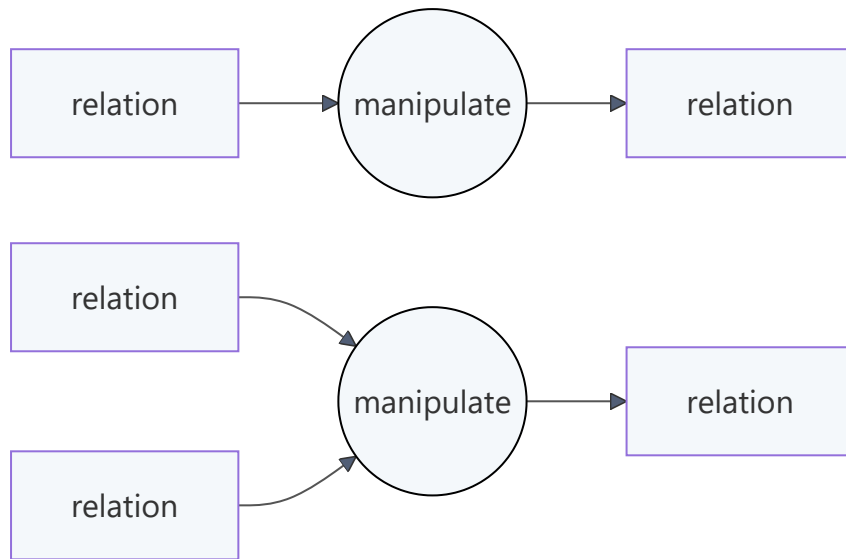


- candidate key: Its true subset cannot form a super key of a super key (also called "minimal super key").
  - Also not unique
- primary key: Candidate key that is selected by the database designer.
  - use **underline** to identify
  - **atomicity**
- foreign key:
  - e.g. Attribute **dept\_name** is a **foreign key** from instructor, **referencing** department. instructor(ID, name, dept\_name, salary); department(dept\_name, building, budget)
  - referencing relation and referenced relation
  - **Note: The foreign key does not necessarily have the same name as the primary key to which it is referentially related.**

### 3. relation algebra

Relational algebra is the theoretical foundation of SQL

Relational algebra is defined over **relations, tuples and attributes**.



Operations are applied either to a single relation or to two relations. **The result is always a single relation.**

### 3.1 SELECT

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
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

e.g1: Select all teachers whose **dept\_name** is **Physics**.

$$\sigma_{dept\_name='physics'}(instructor)$$

e.g2: Select all teachers with 'salary greater than 90000'.

$$\sigma_{salary>90000}(instructor)$$

Comparison operations on predicates (predicate) can use the symbols:  $>$ ,  $<$ ,  $=$ ,  $\geq$ ,  $\leq$ ,  $\neq$ . In addition, multiple predicates can be combined by **logical connectives**:

- and:  $\wedge$
- or:  $\vee$

- not:  $\neg$

e.g3: Select all teachers in the Physics Academy with a salary greater than 90000.

$$\sigma_{\text{physics}=\text{"Physics"} \wedge \text{salary} > 90000}(\text{instructor})$$

## 3.2 PROJECT

e.g1: Returns the name, dept\_name, and salary of all teachers.

$$\Pi_{\text{ID}, \text{dept\_name}, \text{salary}}(\text{instructor})$$

- $\text{select}(\sigma)$  : Intercepts relationships horizontally, affecting [line].
- $\text{project}(\Pi)$  : Intercepts relationships vertically, affects [columns].

Moreover, the generic projection operation allows simple arithmetic on attributes:

$$\Pi_{\text{ID}, \text{dept\_name}, \text{salary}/12}(\text{instructor})$$

## 3.3 Combination of relational operations

e.g4: Find the **names** of **all Physics** faculty members.

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

e.g5: Find information on all faculty members belonging to the **Physics' or** Chemistry' department.

$$\sigma_{\text{dept\_name}=\text{"Physics"} \vee \text{dept\_name}=\text{"Chemistry"}}(\text{instructor})$$

e.g6: Find name and salary of the teacher with ID 10101

$$\Pi_{\text{name}, \text{salary}}(\sigma_{\text{ID}=10101}(\text{instructor}))$$

## 3.4 Cartesian product

$$A \times B = \{(a, b) : a \in A, b \in B\}$$

- instructor(ID, name, dept\_name, salary)
- teaches(ID, course\_id, sec\_id, semester, year)
- The schema of  $r = \text{instructor} \times \text{teaches}$  :  $r(\text{instructor.ID, name, dept\_name, salary, teaches.ID, course\_id, sec\_id, semester, year})$

<i>instructor.ID</i>	<i>name</i>	<i>dept.name</i>	<i>salary</i>	<i>teaches.ID</i>	<i>course_id</i>	<i>sec_id</i>	<i>semester</i>	<i>year</i>
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

$r = \text{instructor} \times \text{teaches}$

Find information about all the teachers and their classes:

### 3.4.1 JOIN

JOIN is a combination of Cartesian product and selection:

$\text{instructor} \bowtie_{\text{instructor.ID=teaches.ID}} \text{teaches}$

- natural join If the  $\theta$  condition is that **attribute values of the same name are equal**, it can be omitted .In this case, it is called a **natural join**.

$\text{instructor} \bowtie \text{teaches}$

### 3.5 UNION, INTERSECT, DIFFERENCE

e.g.  $\text{section}(\text{course\_id}, \text{sec\_id}, \text{semester}, \text{year}, \text{building}, \text{room\_number}, \text{time\_slot\_id})$ ; Find all courses that are in both the Fall 2017 semester and Spring 2018 semester

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

**The precondition for two relations to perform the parallelism operation is that they are compatible, i.e., the two relations have the same arity and each corresponding attribute is of the same type.**

e.g. Find the names of all employees with a salary greater than 10,000  
 $\text{employee}(\underline{\text{ID}}, \text{person\_name}, \text{street}, \text{city})$   
 $\text{works}(\underline{\text{ID}}, \text{company\_name}, \text{salary})$   
 $\text{company}(\underline{\text{company\_name}}, \text{city})$

$$\Pi_{\text{person\_name}}(\sigma_{\text{salary}>10000}(\text{works} \bowtie \text{employee}))$$