# SEEM3550 Fundamentalsof Information System

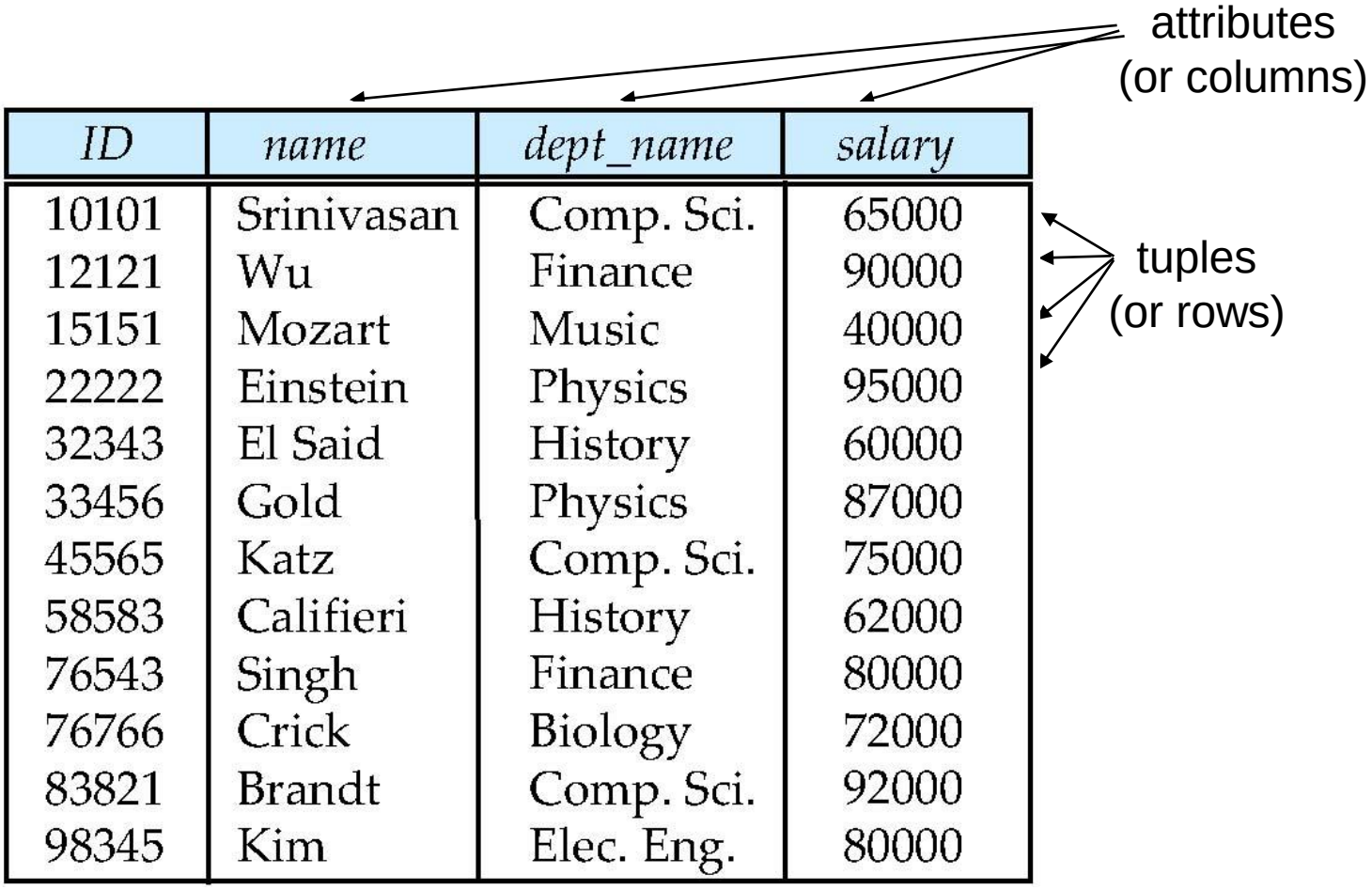
Tutorial 1

Zijin Feng

# Database Management System

- Why not file systems
  - Data redundancy and inconsistency
  - Difficulty in accessing data
  - Data isolation
  - Integrity problems
  - Atomicity of updates
  - Concurrent access
  - Security

# Relational Model




| *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 |

# Relation Schema and Instance

- Each table is a relation ***r*** on the relation schema R: r(R)
- Schema R=$\{ A_1, A_2, \ldots, A_n \}$
- Each attribute $A_i$ has a domain $D_i$
- Attribute values are required to be atomic
- A relation ***r*** is a set of n-tuples ($a_1$, $a_2$, …, $a_n$) where each $a_i$ in $D_i$

# DDL and DML

- Data Definition Language (DDL)
  - **create table** *instructor* (

    *ID* **char**(5),

    *name* **char**(20),

    *dept_name* **char**(20),

    *salary* **numeric**(8,2))

| *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 |

# DDL and DML

- Data Manipulation Language (DML)
  - **select** *name*
    **from** *instructor*
    **where** *instructor.ID* = '22222'

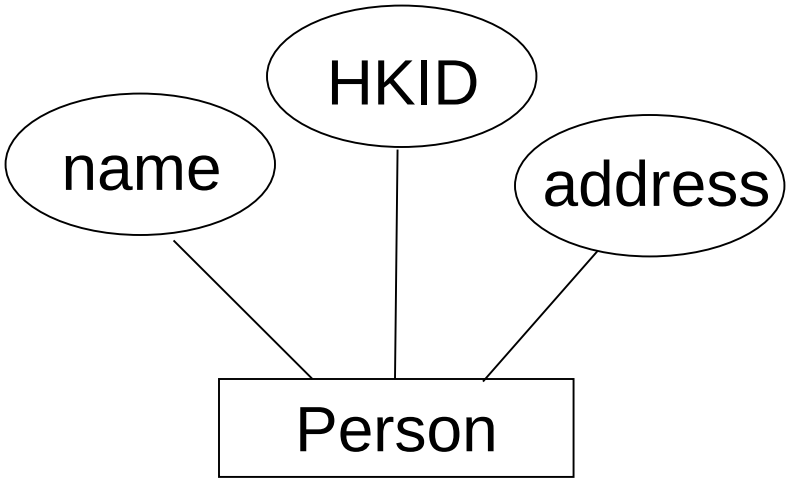
In plain English:

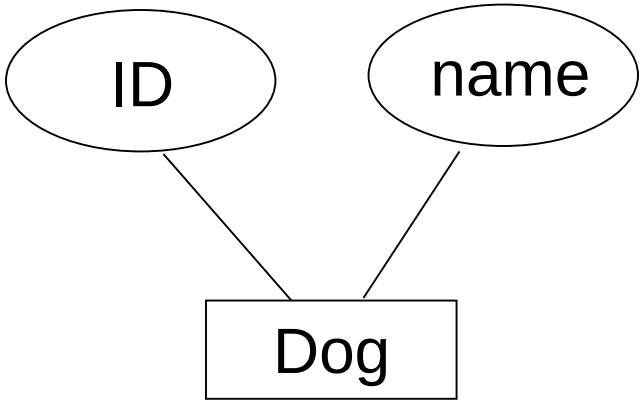
Find all instructors' name whose ID is equal to 22222

| *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 |

# The Entity-Relationship Model

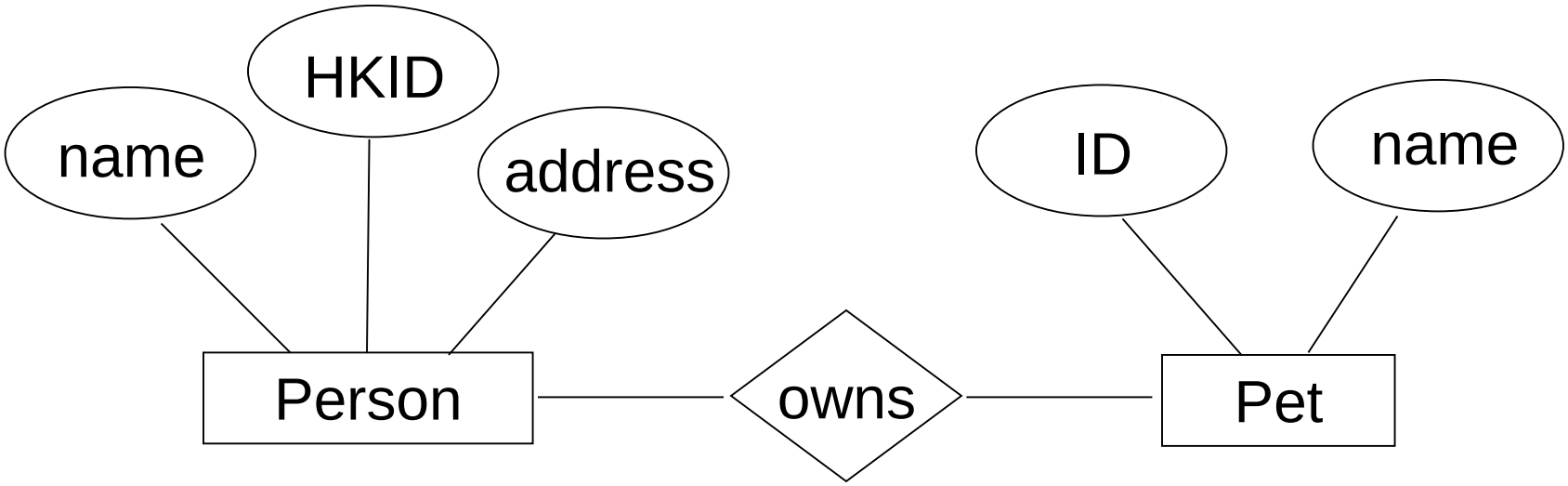
- **Entity:** An entity is a real-world object distinguishable from other objects.
  - e.g. A person, a dog, a CD, a …



# The Entity-Relationship Model

- Relationship: A relationship describes the connection between entities
  - E.g. A person owns a pet, A man marries a woman

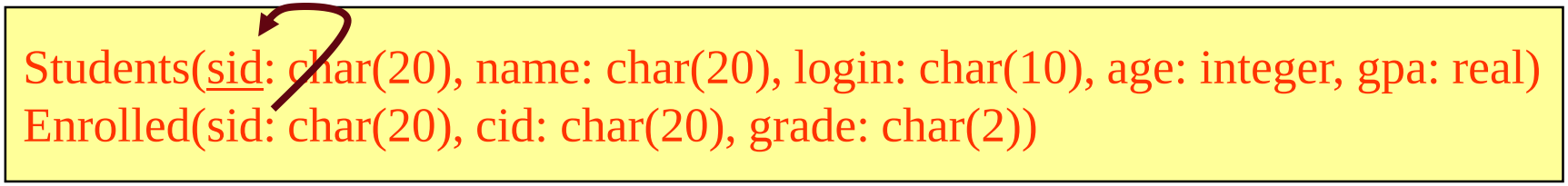


# Key

- Superkey
  - *K* is a **superkey** of *R* if values for *K* are sufficient to identify a unique tuple of each possible relation *r(R)*
- Candidate key
  - Superkey *K* is a **candidate key** if *K* is minimal
- Primary key
  - The one of candidate keys that designated is the primary key

# Key

- Foreign key: Set of attributes in one relation that is used to `refer' to a tuple in another relation
  - E.g., *sid* is a foreign key referring to Students



# Key

| A1 | A2 | A3 | A4 |
|---|---|---|---|
| 1 | a | alpha | x |
| 2 | a | beta | x |
| 3 | b | gamma | y |
| 4 | c | delta | z |

- Superkey
  - A1, (A1, A2), A3, etc.
- Candidate key
  - A1, A3
- Primary key
  - A1 or A3

# Relational Operators

| Symbol (Name) | Example of Use |
|---|---|
| $\sigma$ (Selection) | $\sigma_{\text{salary>=85000}}(instructor)$ |
| | Return rows of the input relation that satisfy the predicate. |
| $\Pi$ (Projection) | $\Pi_{ID,\ salary}(instructor)$ |
| | Output specified attributes from all rows of the input relation. Remove duplicate tuples from the output. |
| $\bowtie$ (Natural Join) | $instructor \bowtie department$ |
| | Output pairs of rows from the two input relations that have the same value on all attributes that have the same name. |
| $\times$ (Cartesian Product) | $instructor \times department$ |
| | Output all pairs of rows from the two input relations (regardless of whether or not they have the same values on common attributes) |
| $\cup$ (Union) | $\Pi_{name}(instructor) \cup \Pi_{name}(student)$ |
| | Output the union of tuples from the two input relations. |

# Exercise

*R*

| sid | bid | day |
|---|---|---|
| 22 | 103 | 10/10/96 |
| 31 | 101 | 11/12/96 |
| 22 | 102 | 11/12/96 |
| 22 | 101 | 10/10/96 |
| 58 | 103 | 11/12/96 |

Reservation

*S*

| sid | sname | rating | age |
|---|---|---|---|
| 22 | Bob | 7 | 45.0 |
| 31 | Alice | 8 | 55.5 |
| 58 | Peter | 10 | 35.0 |

Sailor

*B*

| bid | bname | color |
|---|---|---|
| 101 | Sun | red |
| 102 | Moon | yellow |
| 103 | Mars | green |

Boat

S(*sid*, *sname*, *rating*, *age*)
B(*bid*, *bname*, *color*)
R(*sid*, *bid*, *day*)

# Exercise

S(*sid*, *sname*, *rating*, *age*)
B(*bid*, *bname*, *color*)
R(*sid*, *bid*, *day*)

1. Find the name and ratings of each sailor

2. Find the names of sailors over the age of 40

# Exercise

S(*sid*, *sname*, *rating*, *age*)
B(*bid*, *bname*, *color*)
R(*sid*, *bid*, *day*)

3. Find the names of sailors who have reserved boat 103

4. Find the names of sailors who have reserved a red boat

# Exercise

S(*sid*, *sname*, *rating*, *age*)
B(*bid*, *bname*, *color*)
R(*sid*, *bid*, *day*)

5. Find the names of sailors who have reserved a red boat or a green boat

6. Find the names of sailors who have reserved a red boat and a green boat

# Exercise

S(*sid*, *sname*, *rating*, *age*)
B(*bid*, *bname*, *color*)
R(*sid*, *bid*, *day*)

7. Find the names of sailors who've reserved at least 1 boats

8. Find the sids of sailors with age over 40 who have not reserved a red boat