Module - I

Relational Databases

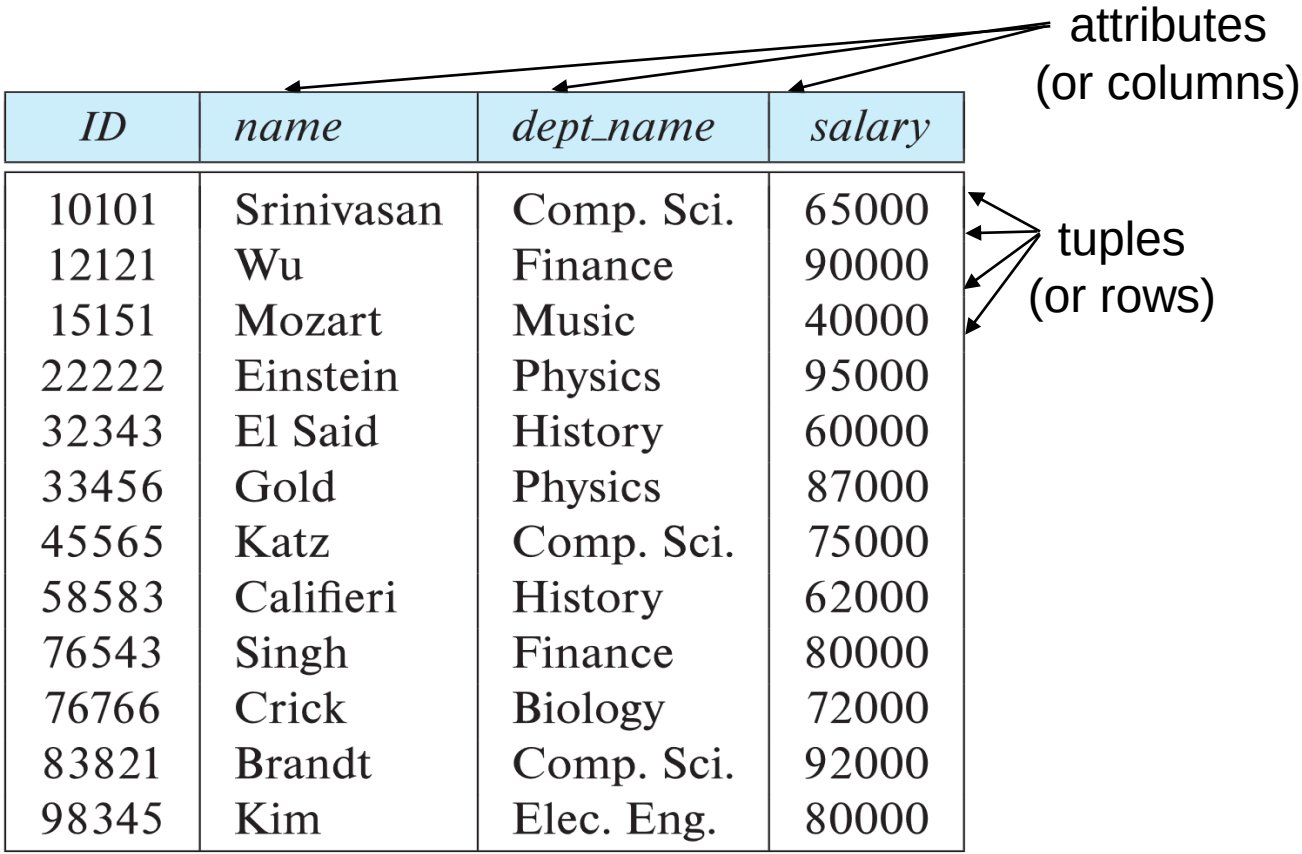
Lecture-08

Introduction to Relational Model

Introduction to Relational Model

- Structure of Relational Databases
- Database Schema
- Keys

Example of a *Instructor* Relation



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

- A_1, A_2, …, A_n are *attributes*
- $R = (A_1, A_2, \ldots, A_n)$ *is a relation schema*

Example:

$instructor = (ID, name, dept\_name, salary)$

- A relation instance r defined over schema R is denoted by $r(R)$.
- The current values in a relation are specified by a table
- An element $\boldsymbol{t}$ of relation $\boldsymbol{r}$ is called a *tuple* and is represented by a *row* in a table

Attributes

- The set of allowed values for each attribute is called the **domain** of the attribute
- Attribute values are (normally) required to be **atomic**; that is, indivisible
- The special value ***null*** is a member of every domain. Indicated that the value is "unknown"
- The null value causes complications in the definition of many operations

Relations are Unordered

- Order of tuples is irrelevant (tuples may be stored in an arbitrary order)
- Example: *instructor* relation with unordered tuples

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

Database Schema

- Database schema - is the logical structure of the database.
- Database instance - is a snapshot of the data in the database at a given instant in time.

Example:

Schema: *instructor(ID, name, dept_name, salary)*

Instance:

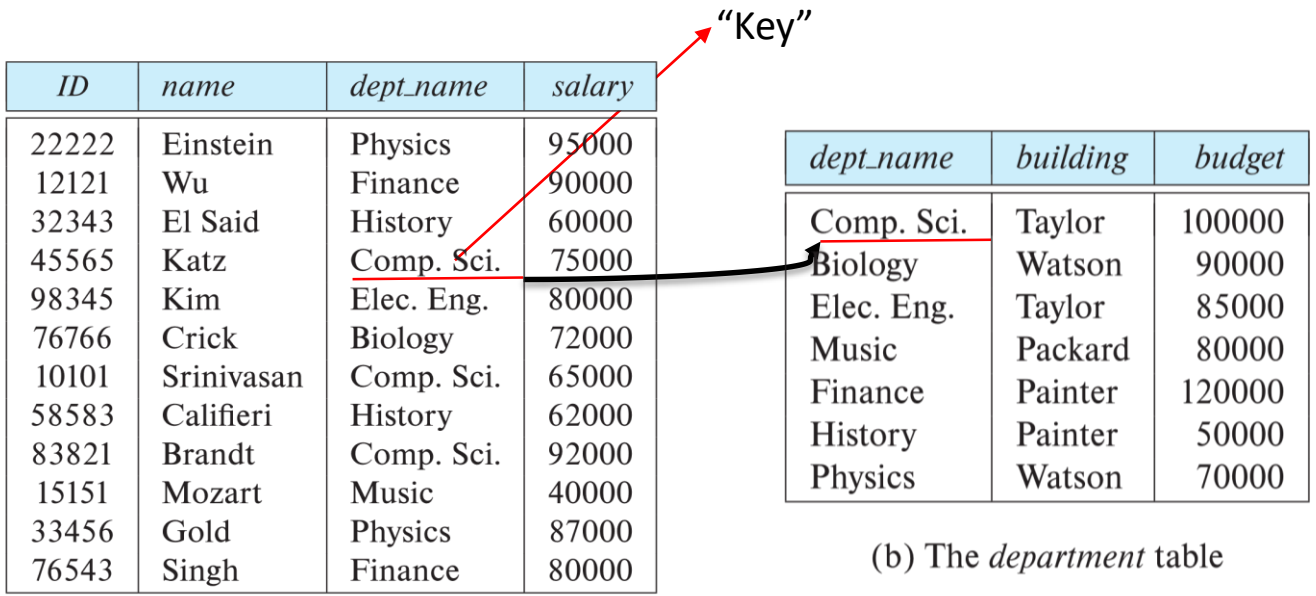
ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music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

Keys

- In a relational database, **keys** are used to establish relationships between tables and ensure data integrity.
- **Key** also help uniquely identify relationships, and thus *distinguish* relationships from each other.

Keys

Example: Consider the following two relations *instructor* and *department*. In the relational model related records are linked together with a "**key**".



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

Types of Keys

- There are several types of keys, each serving a specific purpose:
  - Primary Key
  - Super key
  - Candidate Key
  - Foreign Key

Types of Keys

Primary Key:

- A primary key uniquely identifies each record in a table.
- It must contain unique values, and no two records can have the same primary key value.
- Primary keys are typically used as the main reference for relationships between tables.
- Example: In a “`Instructor`" table, the "`ID`" field can serve as the primary key.

Types of Keys

Super Key:

- A super key is a set of one or more attributes that allow identifying uniquely an entity in the entity set.
- If anyone adds additional attributes to a primary key, the resulting combination would still uniquely identify an instance of the entity set. Such augmented keys are called **super key**.
- A primary key is therefore a minimum **super key**.
- Example: The *Roll_No* attribute of the entity set "Student" distinguishes one student entity from another.

Types of Keys

Candidate Key:

- A Candidate Key can be any column or a combination of columns that can qualify as unique key in database.
- There can be multiple Candidate Keys in one table.
- Each Candidate Key can qualify as Primary Key.

Example: *Student_name* and *Student_street*, are sufficient to uniquely identify one particular student. Hence *Roll_No*., and {*Student_name*, *Student_street*} are candidate keys. Although the attributes *Roll_No* and *Student_name* together can distinguish *Customer* entities, their combination does not form a candidate key. Since the attribute *Roll_No* alone is a candidate key.

Types of Keys

- Let $K \subseteq R$
- K is a **superkey** of R if values for K are sufficient to identify a unique tuple of each possible relation $r(R)$

  Example:

  $\{ID\}$ and $\{ID, name\}$ are both **superkeys** of *instructor*.

- **Superkey** is K is a **candidate key** if K is minimal

  Example:

  $\{ID\}$ is a **candidate key** of *instructor*

- One of the **candidate keys** is selected to be the **primary key**.

Types of Keys

Foreign Key:

- A Foreign Key is a field in one table that refers to the Primary Key in another table.
- It establishes relationships between tables, enforcing referential integrity.
- Example: In an "*Orders*" table, the "*CustomerID*" field can be a **Foreign Key** referencing the "*CustomerID*" **Primary Key** in the "*Customers*" table.

Types of Keys

- **Foreign key** constraint: Value in one relation must appear in another
  - **Referencing** relation
  - **Referenced** relation

Example:

dept_name in *instructor* is a *foreign key* from *instructor* referencing *department*

Types of Keys

- **Foreign key** constraint:

Example:

dept_name in *instructor* is a *foreign key* from *instructor* referencing *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

(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