Lecture 7

The Relational Data Model

Week4

Overview

- Relations and attributes
- Key attributes (candidate, primary)
- Attribute domains
- Foreign key constraints

Relations and attributes

- The central concept of the relational data model is the *relation*
- A relation has one or more *attributes*, and each attribute may have an atomic value
- The set of all possibble values of an attribute is called the attribute’s *value domain*

An example relation

Employee (ssn, name, address, dno)

Relation schema

attribute domains:

- ssn is an Integer
- name is a String
- address is a String
- etc.

Relation schema
with atribute domains

Employee (int:ssn, string: name, string: address, int: dno)

Relation schema, relation instance

- A relation is a *set of tuples*
- Each tuple consists of a list of values
- Each value in the tuple corresponds to one of the attributes of the relation

Employee =

$$\text{A relation instance} \left\{ \begin{array}{l} \{ <69, \text{john}, \text{11HighSt}, \quad 2> \\ \quad <72, \text{jane}, \text{22LowSt}, \quad 3> \\ \quad <99, \text{sue} , \text{77MiddleSt}, 3> \} \end{array} \right.$$

- Tuples in a relation are unique (no repetition)
- Relations can be thought of intuitively as tables, attributes describing colum names and tuples corresponding to rows of the table

ssn,	name,	address,	dno
69,	john,	11 HighSt,	2
72,	jane,	22 LowSt,	3
99,	sue ,	77 MiddleSt,	3

Relation schema

- A relation schema describes the relation name, the name of the attributes in the relation, and the attribute domains for each attribute
- In addition, a relation schema describes various *constraints* that must hold true of any instance of the relation.
- Primary key attributes underlined

E.g. Employee ($\underline{\text{ssn}}$, name, address, dno)

- A set of attributes which are known to have a unique value for each possible tuple in a relation is called a *candidate key*.
- Candidate keys may be represented as below (e.g. tax file number, abbreviated as ‘tfn’, is also unique for Employee, so the Employee relation has two candidate keys, ssn and tfn).

E.g. Employee (<u>ssn</u>, name, address, tfn, dno)
ck: {ssn} and {tfn}

Key attributes (candidate-, primary-)

Any relation must have a set of attributes which has a unique value for each tuple in that relation. This ensures that two tuples from the same schema are always distinguishable. I.e., every relation must have at least one key - where one is *designated* to be the primary key.

E.g. Employee (ssn, name, address, dno)

Example relation with *composite* primary key

SAILOR (Name, MothersName, Bdate, Bplace, Address, Phone)

Here the primary key is a composite of four attributes

pk = {Name, MothersName, Bdate, Bplace}

Similarly, a relation may have other composite candidate keys.

Arbitrary constraints

- Constraints which have no special graphical notation may be described using logic expressions that must evaluate to true over the set of tuples for all relationship instances

- This however, often needs *operations* on the relationship instances, so we need to introduce relational algebra

- Alternatively, relational calculus can also be used for the same purpose (to be discussed later)

The end