پوهنتون کابل

پوهنحی کمپیوترساینس

Introduction to Database and Data Models

Lectures 07

تهیه کننده : پوهنیار محمد شعیب "زرین خیل"
سال : 1389

Introduction to Database and Data Models - Relational Model

07

By: M Shuaib Zarinkhail

2010

Keys

- Each relation has one or more attributes that uniquely define a row
- This set of attribute(s) is called a 'Primary key'
  - Primary key is a type of a key explained later
- Example:

  STUDENT(NID, Name, Major)

What is the key in this relations?

Points to Set Keys

- An atomic key is better than a compound key
- A numeric attribute is better than a text attribute
- Short and simple key is better than long and complicated keys
- A key should not change over time

Key Symbols

- Primary keys are noted by underlining them:
  - STUDENT(NID, Name, Major)
- Foreign keys are noted by italicizing them:
  - CLASS(ClassID, Credit, *NID*)
- Both are noted by underlining and italicizing them:
  - DORM(DormName, Address, *NID*)

Key Types

- Primary keys (PK)
- Composite keys
- Candidate keys
- Surrogate keys
- Foreign keys (FK)
- Alternate keys
- Compound keys

Primary keys (PK)

- No repeated data
- Indexed fields in a relation
- Declares records in a relation
- Short and best data
- Examples:
  - STUD (Name, Department, DoB, Salary)
  - STORE (StoreName, Address, StoreNumber)

Composite keys

- Constructed from more than one attributes

- Examples:
  - STUD (Name, Department, DoB, Salary)
  - STORE (StoreName, Address, StoreNumber)

Candidate keys

- It is possible for a relation to have several possible attributes where each one could individually be a primary key
- Example:
  - EMP (Emp#, Name, NID, DoB, …)
  - The possible key attributes are called candidate keys

Candidate keys

- Similar to primary keys
- No repeated data
- Declares records within a relation
- No symbol is using for candidate keys
- Every PK is candidate key
- Every candidate key is not a PK but can be a PK

Candidate key Examples

- STUDENT (ID, Name, Department, DoB, Salary)
- STUDENT (ID, Name, Department, DoB, Salary)
- STUDENT (ID, Name, Department, DoB, Salary)
- STUDENT (ID, Name, Department, DoB, Salary)

Surrogate keys

- Replaces with composite keys and becomes primary or candidate keys
- Reduces size of indexed fields (PK)
- Example:
  - STUDENT (Name, FatherName, Department, DoB, Salary)
  - STUDENT (ID, Name, FatherName, Department, DoB, Salary)

Foreign keys (FK)

- Represent one relation in another
- Indicate a relationship between relations
- Use copy of PK of one relation in another
- Attribute, or set of attributes, within one relation that matches candidate key of some other relation (s)

Foreign keys - Examples

- Branch (BranchNo, Street, City)
- Staff (StaffNo, Name, Position, DoB, Salary, *BranchNo*)
  - *BranchNo* is a FK references BRANCH(BranchNo)
- STUDENT (Name, FatherName, Department, DoB, Salary)
- CLASS (ClassID, Credit, *Name, FatherName, Department*)
  - (*Name, FatherName, Department*) is a FK references STUDENT(Name, FatherName, Department)

Foreign Key - Example

A student lives in a dorm room

Foreign key - Example

- STUDENT (NID, Name, Major)
- DORMROOM (Building, Room, Phone)

How do we relate a student to a dorm room?

- STUDENT (NID, Name, Major, *Building*, *Room*)

Foreign key - Example

- Copy of PK of one relation add as foreign key to another relation
- Cardinality of a relationship determines which entity should contain the copy of PK (foreign key)
  - One relation should treat as parent or super type and another relation should be child or sub class in each relationship

Foreign key - Example

- There are three cases
- One to One (1:1)
  - Either relation can be parent
- One to Many / Many to One (1:N or N:1)
  - Only the one-side relation treated as parent
- Many to Many (N:M)
  - Not applicable in practical

1:1 Foreign Keys

- A (<u>Akey</u>, Attr2, Attr3)
- B (Bkey, Attr2, Attr3, *<u>Akey</u>*)

OR

- A (Akey, Attr2, Attr3, *<u>Bkey</u>*)
- B (<u>Bkey</u>, Attr2, Attr3)

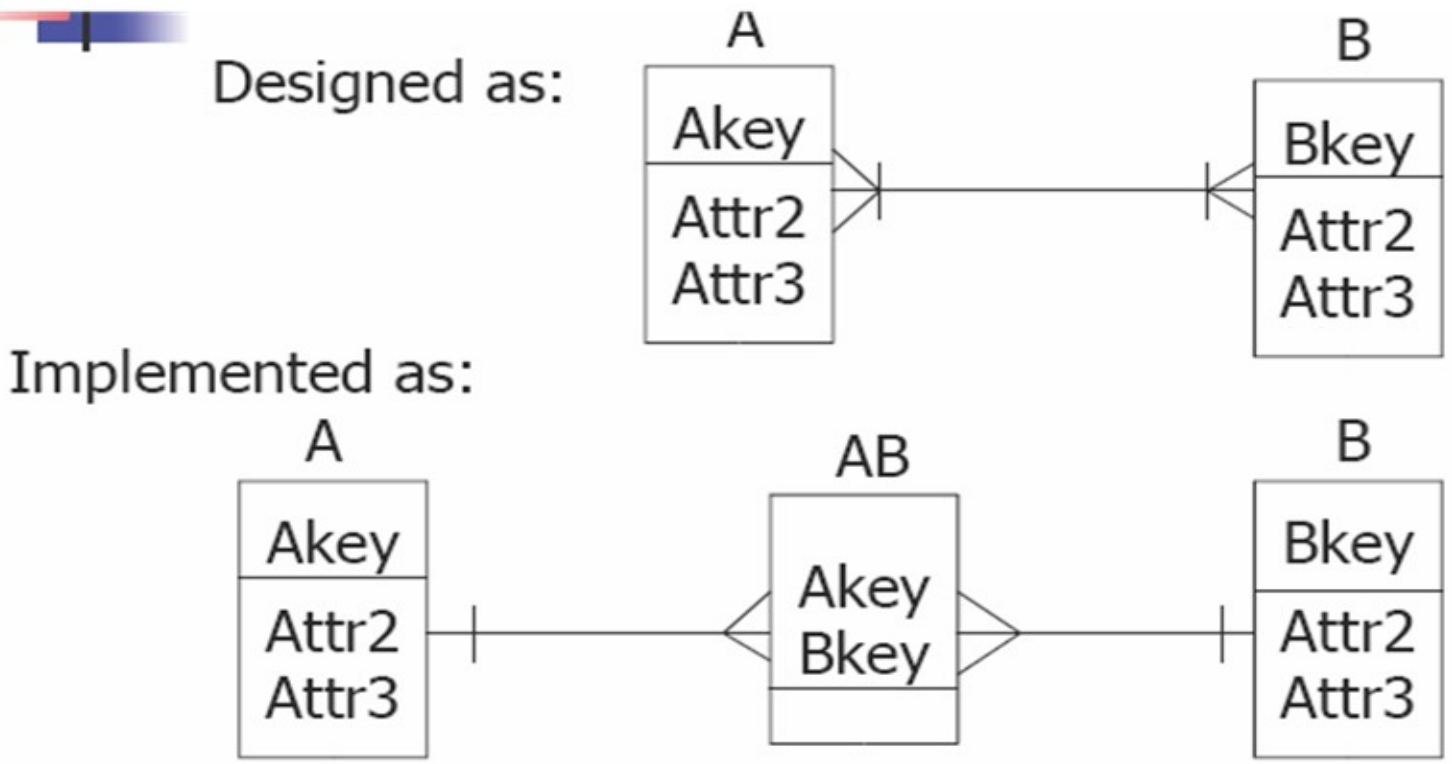
1:N Foreign Keys

- A (Akey, Attr2, Attr3)
- B (Bkey, Attr2, Attr3, *Akey*)

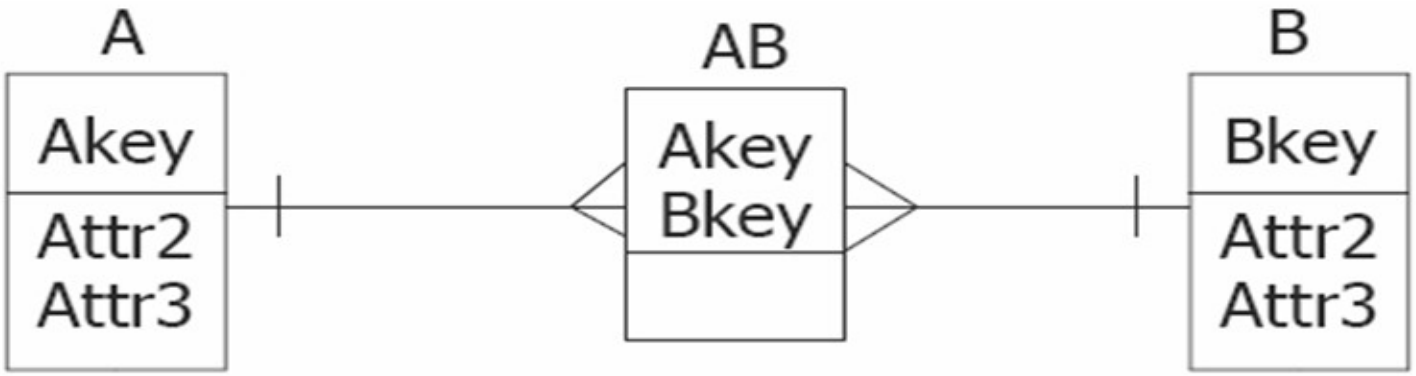
N:1 Foreign Keys

- A (Akey, Attr2, Attr3, *Bkey*)
- B (Bkey, Attr2, Attr3,)

N:M Foreign Keys



N:M Foreign Keys



A(Akey, Attr2, Attr3)

B(Bkey, Attr2, Attr3)

AB(Akey, Bkey) ← Intersection relation

Alternate keys

- Any candidate key which is not selected as Primary key
- An alternate key is a function of all candidate keys minus the primary key

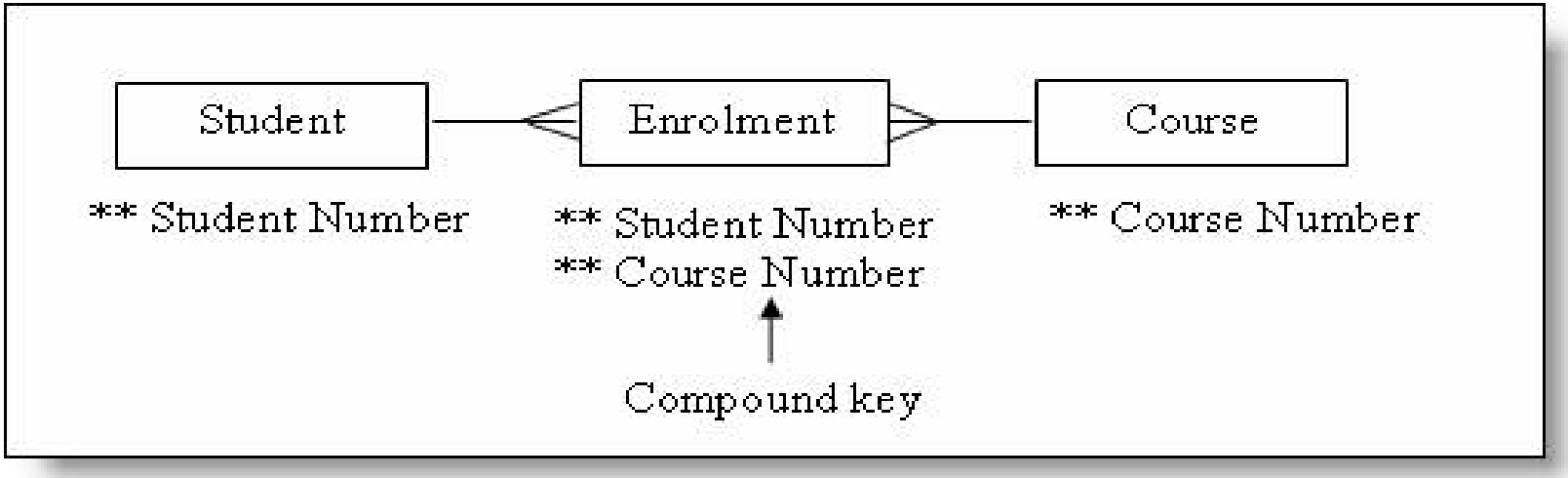
Compound keys

- A compound key consists of more than one attribute to uniquely identify an entity occurrence
- Compound key is different from Composite key
  - Each attribute, which makes up the key, is also a simple key in its own right in Compound key
  - While a composite key may consist from any attribute
- Each Compound key is a Composite key
- Each Composite key is not a Compound key, but can be

Compound keys - Example

- We have an entity named ENROLMENT, which holds the courses on which a student is enrolled
- A student is allowed to enroll in more than one course
  - This has a compound key of both student number and course number, which is required to uniquely identify a student on a particular course

Compound keys - Example



- Student number and course number combined is a compound primary key for the ENROLMENT entity

Compound keys - Example

- Student number in the ENROLMENT entity is a simple key in its own right, which is used as a FK to link to the STUDENT entity
- Course number in the ENROLMENT entity is a simple key in its own right, which is used as a FK to link to the COURSE entity