

Khoi Duong

Prof. VB

CS457L

9/17/2022

Ex 5.13:

In the relation CLASS, candidate keys can be *Univ_section#* since the value of that attribute is unique in the relation CLASS. Besides, if *Univ_section#* is not unique in the relation, we can have various candidate keys under different conditions:

- *Course#* if every class has one unique course number.
- *Instructor_name* if every teacher/instructor only teaches one class
- *Room#* if the same room cannot be used by more than one class
- (*Semester, Weekdays*) if for a given semester, each weekday has only one class
- (*Building_code, Time_period, Credit_hours*) if in a given building (e.g. Building A, B, or C, etc.) and a given credit-hours value, every class has its unique time period value (e.g. 30 mins, 45 mins, 75 mins, etc.)

Ex 5.15:

For a database that keeps track of business trips of salespersons in a sales office, and we know from the problem that a trip can be charged to one or more accounts. Furthermore, we know that a foreign key is a field of a table pointing to the primary key of another table. Thus, for the 3 sections SALEPERSON, TRIP, and EXPENSE, we will have the foreign key below:

- *Ssn* in TRIP section is the foreign key which points to the primary key of SALEPERSON section (which is also *Ssn*)
- *Trip_id* is the foreign key in both TRIP and EXPENSE section since it is the primary key of both sections. This means it can point to each table primary key.

Assume that there will be more sections in the relations of the database, we will have different foreign keys based on different conditions:

- If every salesperson has a unique ID value in DEPARTMENT section, the *Dept_no* will be the foreign key in SALEPERSON relation.
- *Account#* can be a foreign key in EXPENSE relation if it points to the primary key of another table (e.g. CUSTOMER table with an unique account number)

Ex 5.19:

a.

The *Local_phone* and *Cell_phone* attributes are missing the area code (or state, province, city code), which is a very important information. If there is no area code in the phone number field, there is no way we can call someone who live in a different state or province.

b.

We should add new attributes to the schema for STUDENT since the area code is different based on the province or state the student lives in. Furthermore, the area code should be in a new attribute in order to classify students based on their area code number or to make easy for people living outside the state/province to call the student.

c.

When splitting this field from one attribute into three attributes, the advantages are:

- Sort and group data based on attribute *first_name*, *last_name*, or *middle_name*
- Easier to create a find query for the students (e.g. students may enter only first, last, or middle name in order to immediately find their name since the query will give out only several results on the basis of the attributes' values)
- The name value is now separated into 3 attributes which means the data will be more precise and clearer for the administrator to organize and manage.

However, the disadvantages of splitting this field from 1 to 3 attributes are:

- If students don't have middle name, the database will take extra spaces just to contain NULL values.
- The number of NULL values may increase in the database since there will be students not having middle name
- Assume that every student has a unique name, splitting this field from 1 to 3 attributes means that we have to take all 3 attributes together if we want to make the name attribute become the primary key

d.

General guideline for deciding when to store information in a single attribute and when to split the information:

- When splitting the information will increase drastically the number of NULL values without any more useful advantages, it is better to keep all information in a single attribute.

- If the piece of information is too complex or the information is divisible, we can split up the data into 2 or more attributes
- If the data can be much more easier to classify or manage in detail, smaller pieces of information, it is better to split up the data
- When the data needs to be sorted or grouped based on the sub element of an attribute, splitting that attribute into more attributes will help the finding query become easier.

e.

Suppose the student can have between 0 and 5 phones. Two different designs that allow this type of information will be:

- All of the phone numbers can be put in a same attribute, creating a query by separating them with commas.
- Duplicate records (e.g. create multiple same student ID values or name) in order to store different phone numbers. In this case, the phone number can be chosen to be the primary key as it is unique while the student ID or name attribute may have more than one same value (assume that the student can have between 0 and 5 phones).