SEPTEMBER 16, 2024

HOMEWORK 2 A

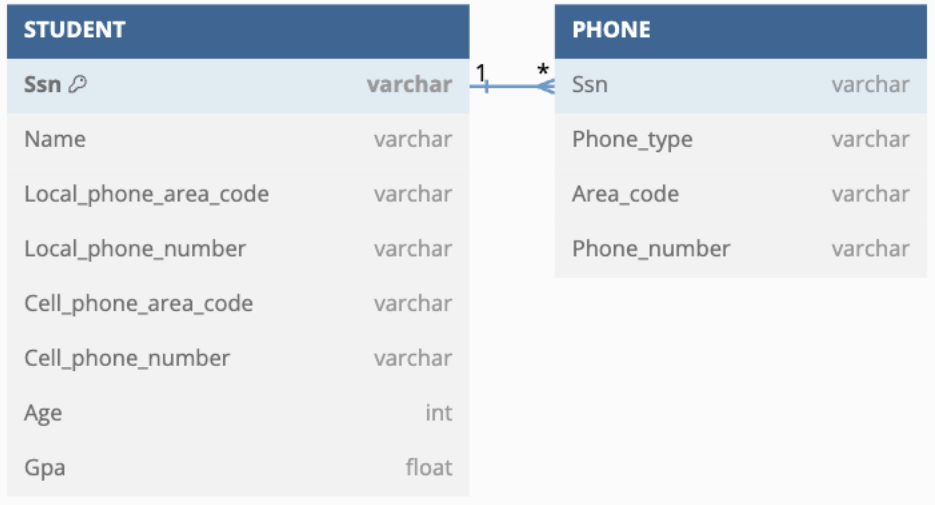
CS 457 B

UMAIR DADA
2O281 - SFBU

1. 5.13
    a. In the relation `CLASS(Course#, Univ_Section#, Instructor_name, Semester, Building_code, Room#, Time_period, Weekdays, Credit_hours)`, candidate keys could be:
        i. **Univ_Setion:** It can be a unique section number for all the classes, in accordance that there are no repetition between the semesters or the classes.
        ii. **Course + Semester + Time_period + Weekdays + Room:** Uniquely identifies a class if there no multiple classes scheduled at the same time and place.
        iii. **Course + Instructor_name + Semester + Time_period + Weekdays:** Bases on the instructor that they won't teaching multiple classes
        iv. **Building_code + Room# + Time_period + Weekdays + Semester:** This is as per basis of that no room is double booked or has the same name.

2. 5.15
    a. Foreign Keys
        i. **TRIP(San) → SALESPERSON(Ssn):** Assuming if each of the trips are associated with a single salesperson like there won't be more than one salesperson for one trip.
        ii. **EXPENSE(Trip_id) → TRIP(Trip_id):** All the expenses in trip together, identified by Trip_id. In this way the foreign key can be created.

3. 5.19
    a. Critical Mission Information for both Local_phone and Cell_phone.
        i. The missing field could is the area code. If the area code is missing the call cannot be made as the area code is required to make sure the number is dialed properly, specifically different state or areas.
    b. Storing Additional Information.
        i. To store the additional information, instead of modifying the Local_phone and Cell_phone attributes to store the area code, it would be advisable to create a new attribute named for example area_code to store it. This way it would save the database faster not by much but noticeable enough and make it easier for query manipulation.
        ii. An example of the schema



iii. An alternative situation would be to store full phone number for both Cell_phone and Local_phone, As a recommendation, it can may be used only if the backend of an application that connected to the database cannot be altered for the updated queries.

c. Advantages and Disadvantages of splitting the name attribute.

i. **Advantages**

1. Easier to search or sort by first or last name.
2. A cleaner organized database in case of middle initials or name
3. A flexibility in formatting in case of official documents.

ii. **Disadvantages**

1. Some students may not have a middle initial or name which could lead to empty values.
2. More columns making the database larger.

d. General guidling for storing or splitting information.

i. Separate Attributes

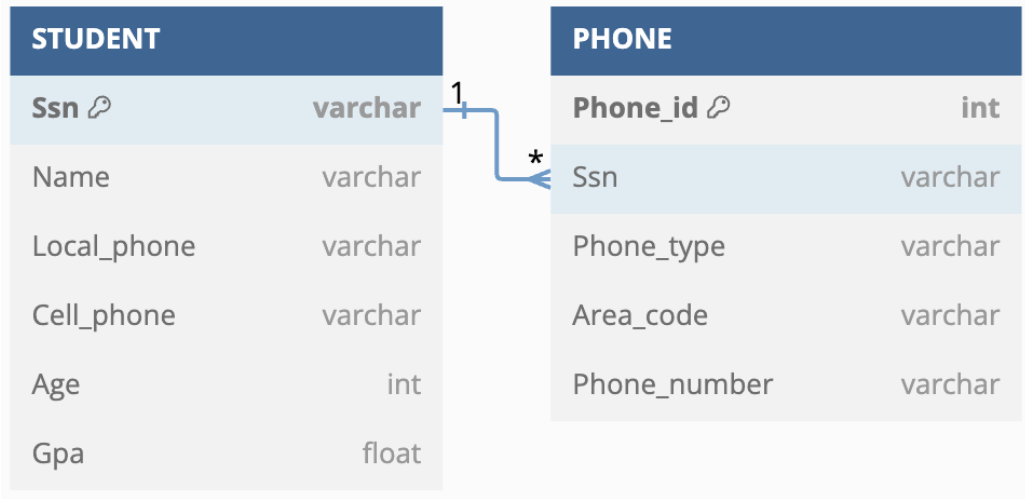
1. Individual components are constantly being used separately like first name, last name or area code.
2. It improves data integrity and querying.

ii. Single Attributes

1. Components that are constantly being used together have to be joined on most queries like full name (in some application the name split is quite rare) or if a single phone number is used then splitting with area code wouldn't be advised.

e. Design to handle 0 to 5 phone numbers for a student.

i. Design 1 (Phone Number Table)

1. 

2. In this design, the phone table is created and referenced to the student's SSN and its one of the values that won't change. If the student has multiple numbers, then a single query gets all the number linked to the student.

ii. Design 2 (Single Table) NOT RECOMMENDED

1.

STUDENT	
Ssn	**varchar**
Name	varchar
Local_phone1_area_code	varchar
Local_phone1_number	varchar
Local_phone2_area_code	varchar
Local_phone2_number	varchar
Cell_phone1_area_code	varchar
Cell_phone1_number	varchar
Cell_phone2_area_code	varchar
Cell_phone2_number	varchar
Cell_phone3_area_code	varchar
Cell_phone3_number	varchar
Age	int
Gpa	float

2. In this design, multiple fields are added to the `student` table, which allows a maximum number of local and cellphone numbers. In the example design above we have 2 local phones and 3 cellphones: each with its own area code. This way all the data is right there and allows simpler queries.

3. I don't recommend this design at all, this design makes the table unnecessarily big, even though simpler queries are helpful, is most cases the primary phone number is required without having need of all 5 numbers. When the query is run a lot of data is considered waste and could possibly take up more storage (empty field still take up storage), reduce performance as it's a large table.