Alan Thomas

CSI215-01CA

Dr. Dulaney

Assignment#16: 5.2, 5.3, 5.5, 5.11, 5.19

November 13, 2016

5.2) Why are tuples in a relation not ordered?

Tuples are not in a particular order when in a relationship due to its foundation in mathematics. A set, which is a relationship, has no distinguishable order, thus a relationship in a database also inherits this trait, as in its most basic form is a set.

5.3) Why are duplicate tuples not allowed in a relationship?

Duplicate tuples are not allowed because it is a key constraint which states that no two tuples can or will share the same values for their attributes. Basically, it is a method to reduce redundancy, which is an objective when designing a database.

5.5) Why do we designate one of the candidate keys of the relationship to be the primary key?

It is considered best practice to choose one of the candidate keys that is the smallest, that is to say, that the candidate key with the least amount of attributes, in general, should be the chosen key. By following this principle, it allows for a database that is easier to work with than one that utilizes overcomplex keys.

5.11)

1. There are no constraints violated in this operation.
2. This is a violation due to the fact that the department table does not have a value of 2 in the dnumber field. This could be resolved by assigning a number to the inserted information that correlates with an existing dnumber so that the relationship is valid.
3. This would result in an error due to the fact that the department table already contains a dnumber with a value of 4. This could be fixed by designating a different number that isn’t located within the dnumber information. Additionally, assuming we did not insert the information from case a, we could say that there is not manager with the social of 943775543. This could be resolved by assigning the relationship to a different manager social security number, or we could simply insert the information found in case a.
4. This case contains errors as well. Pno cannot be null as it is a key within the works\_on table. This could be resolved by providing an integer. Also, an employee would have to be present in the employee table with the provided social security number in order to add them to the works\_on link table. This could be resolved by inserting the employee into the employee table prior to inserting their information into the works\_on table.
5. No errors.
6. No errors.
7. This case would result in issues. When removing the employee with the social 987654321 from the employee table, you would nullify many relations that have been established in subsequent tables, for example, the dependents table, and works\_on table. This could be resolved by removing these references prior to removing the employee from the employee table.
8. This case would result in errors. Similar to case g, removing the product from the project table would result in references being broken and thus create errors. This could be remedied by removing the references located in the works\_on table.
9. No errors.
10. Assuming case a was not performed; this operation would result in errors. Errors would be thrown due to the fact that there are no matching social security numbers with the information provided. This could be rectified by inserting an employee into the employee table which matches the given social security number.
11. This would not result in errors. However, the value of the essn that they are attempting to modify would not be found.

5.19)

* 1. Information which would be useful to have in the database would be an area code in the local phone field, along with the cell phone field. A country code may be useful as well when considering there may be exchange students.
  2. Personally, I would alter the phone number fields to require an area code and country code. This is also assuming this database has not gone into production yet.
  3. The advantages of splitting the name field to consist of a first, middle and last would be that the usability and efficiency of the database would increase. The database would be using a lot of overhead when users would perform queries on the names considering how many false results they received. By allowing the user to specify searches with a greater degree of accuracy, the database could see performance improvements. The cons of this are that the database would require more storage space. However, this seems insignificant when considering the benefits that changing the name field would provide.
  4. A guideline I would use would be if it can increase the performance of the database. Additionally, if it makes using the database easier to use for the end user, I would think it was something worth considering. After all, it is the end user who will be utilizing the database. Another factor I would consider is the budget of institution creating the database. If they are building a database on a shoestring budget and do not have the funds to provide for storage, I may keep fields limited.
  5. If students could have up to 5 phones, the designer could create a separate table that supports this. This link table would then have a relationship with the student. Additionally, you could simply have fields named phone-1 – phone5 and allow the information to be null within these fields.