

CST363 Database Assignment 8 on DDL (Data Definition Language)

Name ____ Christopher Holmes _____

Normalization : Read Chapter 10 in murach textbook about normalization and denormalization of database design.

Read Chapter 11 discussion on using SQL DDL statements to create tables, indexes

1. In your own words, describe the nature and purpose of the normalization process.

The purpose of normalization is to make the data the most efficient for access, storage, and maintenance of the database. This is accomplished by going through different levels of normalization that eventually result in the data being in a form that helps to eliminate duplicate data and ensure that each item has an index by being in a table where it relates to the primary key.

2. What is *denormalization*? When it is used?

Denormalization is when you purposely do not follow the steps involved in normalization. In general, denormalization is used when the benefits of efficiency outweigh the risk of having redundancy errors and storage errors.

3. Explain the types of insert, update and delete problems that can occur if tables are not normalized?

The types of problems that you can run into with insert, update, and delete when a table is not normalized are data inconsistencies, such as data not being present when trying to do an insert and causing a failure, eliminate redundancies, such as duplicate data, and when deleting, too much information being deleted because of an improper relationship that is defined.

4. What is the difference between weak and strong entities? How are weak entities represented in a relational database?

A strong entity in a database is an entity that does not depend on any other entity in the entire schema. A strong entity will always have a primary key associated with it. A weak entity is an entity that does depend on another entity, usually a strong

CST363 Database Assignment 8 on DDL (Data Definition Language)

entity, to be able to exist. A weak entity do not have a primary key, but instead they have a partial key.

5. List the three types of binary relationships and give an example of each (do not use examples from the textbook).

The three types of binary relationships are:

1:1 Employee to Computer

1:N Cell phone to cell phone manufacturer

N:M Cell phone to feature

6. Taking your example of a 1:1 relationship from question #5, write a sql statement that uses a join that returns all data from both tables.

```
Select *  
  
From employee  
  
Join computer on employee.computerNum = computer.computerNum
```

7. For a 1:N relationship, give the key of the parent entity, write an sql select statement that returns all the data from all the child entities.

```
Select *  
  
From cell_phone cp  
  
Join cell_manufacturer cm  
  
on cp.cell_manufacturer_id = cm.cell_manufacturer_id  
  
where cp.cell_id = 98765
```

CST363 Database Assignment 8 on DDL (Data Definition Language)

8. For the Many:Many relationship example you wrote in question #5, write an sql select that given a key of one entity, will return all data from all if the related entities.

```
Select *  
  
From cell_phone cp  
  
Where (cp.has_4G_LTE = 'Y' or cp.has_5G = 'Y')  
  
And cp.has_touch_screen = 'Y'
```