

Michael Sanzo

Database Management

2. Explain the distinctions among the terms primary key, candidate key, and superkey.

Primary Key: Any column or set of columns that can uniquely identify a record in the table.

(There can only be one)

Candidate Key: Any column or set of columns that are candidate to become primary keys. (There can be more than one)

Super Key: If you add any other column/attribute to a primary key then it becomes a super key.

3. Write a short essay on data types. Select a topic for which you might create a table. Name the table and list its fields (columns). For each field, give its data type and whether or not it is nullable.

A data type, in reference to computer programming is a classification identifying one of various types of data. It typically depends on what information you are going to be gathering to determine what type of data you are looking for. With that said, there are multiple of these that exist which consist of syntactic, representation, behavior, and value space. In addition to types, there are many different data type classes which are of primitive, composite, and some others.

Primitive data types are the building blocks of data manipulation. Examples of primitive data types would be byte, short, double, Boolean, char, float, long, and sink. Being a fundamental item, such types serve just one purpose which is to comprehend clean, simple values of a data set.

A composite data type is considered to be the structure along with the use of primitive data types to return a value. In databases, a composite data type would be one that contains multiple primitive data sets that are arranged in arrays or lists. With composite data types we are capable of spectating many portions of similar data as a single datum. The composite data types help bridge the gap between newly added members and previous variables that are prone to being lost upon the complexity of the database.

I believe that a very good example of a database would be a stock screener to track performance through pricing and graphical analysis. There are many examples of databases that provide this information and one in particular that I like is finviz.com. I would like to relate the project we have in class to stock performance and technical analysis tracking. The set would have the price, along with percent changes every day for 1 week, as well as every week thereafter. The database however should be capable of either doing calculations or high frequency trading using algorithms.

4. Explain the following relational “rules” with examples and reasons why they are important.

a. The “All Fields Atomic” rule: There is when there is no internal structure in between a row and column. The best solution to us if your database is running into the problem of disobeying rule 1, then the best solution would be to create a new table. In the example below the best solution would be to make a new data set with all the superpowers. This is important because the objective of the database is to turn data into information. In order to get perfect information, it is vital that the data should be organized as efficient as possible, hence the rule. As you can see below, there can be many problems with altering the set of data, however if you were to just create a new table there is now a directory for all the superpowers.

James Bond Characters

Name	Superpower 1	Superpower 2
Sean	Fly	
Pierce	Hair and Strength	Strength
Piece		

b. The “access rows by content only” rule: This rule states that you are only capable of gathering rows by their content, the attribute values that are in each row. There is no way that a query in a relational language can ask, for example, to retrieve the 3rd row of the table. Instead the query would need to refer to the row with the value of x, in this column, this unique row. Needless to say this rule is important because it is imperative to be specific so the data can be precise.

c. The “all rows must be unique” rule: Two tuples in a relation (rows in a table) cannot be identical in all column values at once. A relation can be thought of as a set of tuples, and of course a set never contains two identical elements; each type of the set must be unique. Since a pure relational query language is able to distinguish rows only in terms of their column values, this is another way of saying that there must be a way of distinguishing any single row from all others in terms of these values, so that a query language statement can retrieve it uniquely.