Database Designs

Section 5: Database Design Theory

5.1 Practical Tips on Database Design?

A good database design is when a database can withstand the test of time. If the data structure does not change that much even when the data grows and expands into new scenarios/areas of information then this can signify a good good database design. This is a hard topic and so much can be said about this topic alone. Below are a few important principles of designing a good schema for a database.

Designing the tables of a database is called designing the schema of a database.The list below is not a comprehensive list but will help when setting up a database to withstand the test of time.

- Identify the problem you are trying to model/capture from the real worlds
- Have a good idea of the types of queries (i.e. information to drive business decisions) that you want to run on the database
- Break up the information into logical units where each logical unit is one table in your database
- Each logical component will probably connect with other logical components. Capture the relationships either via a constraint or other tables devoted to holding relationship specific data

Tables typically hold information about an entity or a relationship. Think of whether you need a way to uniquely identify each row in a table. If the answer to this question is yes (in most circumstances this will be true) you need a Primary Key.

- A Primary Key can never have a NULL value
- A Primary Key will be used to represent that row and other tables can reference that row by specifying the Primary Key value
- Do not use data that can change as a Primary Key value
- If there are no such values in the fields you want to store then you can generate a value which will be used as a unique identifier i.e. DBMS have auto-increment to generate unique identifiers for you
- Numeric values tend to be more efficient in representation and lookup and should be preferred for Primary Key values
- Having an identifier whose job is to be a key value is standard practice
- Remember, most DBMS will automatically create an index on the Primary Key for faster lookup and retrieval

Do columns have other constraints which they need to satisfy?

- Are NULL values allowed for any column? If the answer is no then the column should be specified as NOT NULL
- Should columns have a default value? If the answer is yes then the column should be setup with the DEFAULT keyword followed by what the default value for that column should be

- Ensure the column datatype are setup correctly e.g. don’t use a string for storing date information (even though it is allowed) as this will limit the features applicable to date and time values for example comparing date difference between two date values
- Think about the number of characters/numbers a column will hold e.g. if a data is a single character don’t use a CHAR(30) to store the string. Database occupy space and space that are not used up are a waste of resource and storage resource is expensive