Jacob M. Peck
CSC 459 - Databases
Professor James P. Early
January 30, 2012

Assignment 1

1. **Q** Discuss some important differences between a Data Manipulation Language (DML) and a Data Definition Language (DDL). What do each provide to a database?

   A The DML describes methods of how to obtain and manipulate data stored within a particular instance of a database, whereas the DDL describes the format and layout of the data's schema. Some database DMLs function as DDLs as well, such as SQL's `CREATE` keyword. DMLs and DDLs allow simple, uniform access to a database, regardless of size or record count, with results that are consistent and conform to a correct schema.

2. **Q** Discuss the roles of the Physical, Logical, and View levels of a database. What do each provide?

   A The physical layer of a database is the bare-metal aspect of the data. Where and how are individual records stored on in the database? Are they stored as plaintext, or some sort of random access file? The logical layer of the database is the layer where the DML and DDL live. These queries and schema definitions provide coherence and a layer of abstraction away from the physical layer. The view layer lies on top of the logical layer, and provides means for accessing data in a way that allows for clients to know nothing about the underlying schema, by way of data-hiding. These three layers together allow for flexibility and loose coupling—one could, for example, replace the logical layer with a different logical layer that responded to the same queries and stored data in the same format as the previous logical layer, and neither the client nor the datastore know any better.

3. **Q** Consider a small medical clinic in an urban center. Describe (without any formal diagrams) the kind of data that may need to be stored about patients, office staff, and medical personnel. What are some possible relationships among data? What kinds of safeguards should be taken to protect this information?

   A You might need to store quite a bit of information about patients: name, blood type, allergic reaction, current medications, entire medical histories, insurance status, next appointment information, etc. Medical personnel would be much easier: name, position, specialty, salary/wages, address, etc. Office staff would be even easier: name, wages, address, department. From this basis, one could determine a master lookup table, called *persons*, which could map names to person type (patient, medical, or office), and phone numbers and addresses as well, to provide almost guaranteed unique primary keys. This *persons* table would then map to other tables based upon

the type mapped to in the other tables. The *patient* table would hold information applicable to patients, the *medical_personnel* table would hold information about medical personnel, and the *office_personnel* table would hold information about the office personnel. To safeguard this information, the entire database could be encrypted, and stored offsite.

4. **Q** The 'union' relational operator has some requirements of its associated operand relations. What are these requirements, and what is the schema of the resulting relation? Also, how do these compare to the 'set difference' relational operator?

   A The 'union' operator ($\cup$) requires both operands to have compatible schemas— both relations must have the same *arity* and the types of data in the relations must be compatible on an attribute-by-attribute fashion, *i.e.* relations with schemas $\{A : string, B : number\}$ and $\{C : string, D : number\}$ would union just fine, where as a union between relations with schemas $\{A : string, B : number\}$ and $\{C : string, D : boolean\}$ would not be allowed. The 'set difference' operator ($-$ or $\setminus$, variously) has the same requirements as $\cup$. In both cases, the resulting schema would depend upon implementation, but may be safely assumed to be the same schema as the first operand, unless the database engine you are working with is documented to do otherwise.