

Nick Carrozza

CMPT 308

Lab 2 – CAP Database

14 September 2016

1. Refer to corresponding PDF files.
2. A primary key is a key that uniquely identifies all table rows in a relational database table. On the other hand, a candidate key is a key that can technically serve as a primary key because it can still uniquely identify all table rows in a relational database table, however it is not the primary key of that table. A super key specifies that no two sets of attributes in a table are alike, and a set of attributes can in fact be a super key if the set is unique. In addition, the primary key of a table is always then a super key since a primary key is required to be completely unique.
3. Data types can include strings of text, strings of bits, boolean values, integers, floating point numbers, and dates/times. It is important that all attributes have a data type. "SQL permits reasonable coercions between values of character-string types (Ullman, 30)." This basically supports the notion that a character-string value can exist in a given row and column in the relational data model. For strings of bits, the same rules apply, however the only difference is that their values are made up of strings of bits not characters. In addition, the boolean values denote three possible values: true, false, and unknown (Ullman, 30). The integer data type simply denotes an integer value. Data types FLOAT or REAL are often used and there is a particular notation required in order to represent specific numbers in the relational data model. Likewise, dates and times can be expressed using a specific notation by the data types DATE and TIME (Ullman, 31). These are considered character strings of a special form (Ullman, 31). Correctly expressing these data types is critical in the declaration of a data table.

One may create a data table by using the key-words CREATE TABLE followed by the name of the table and a parenthesized list, separated by commas, that lists the attributes and their corresponding data types (Ullman, 31). For example, one may create a table for the elements on an individual's driver's license, with different attributes that include first name, last name, age, height, ID number, expiration date etc. It can be declared by the following:

```

CREATE TABLE LicenseInfo (
    firstName    CHAR(100),
    lastName     CHAR(100),
    age          INT,
    height       INT,
    IDNumber     INT,
    ExpDate      DATE
);

```

4. a) The first relational rule ensures that all fields must be atomic, meaning that no two pieces of data can exist in the same row and column. For example, one column could contain values for favorite color, with one entity in a row having more than one favorite color, like two favorite colors. This cannot exist, as only one favorite color is allowed to exist in that column and row.

	FavoriteColor	FavoriteFood
PersonA	Blue	Bread
PersonB	Green, Red	Cereal

In the above table, green and red cannot exist in the PersonB row and FavoriteColor column. Only one ascribed data value can exist in that square, either green **or** red.

- b) The second relational rule upholds the notion that one is never allowed to access data based on location in the table. For example, one cannot query something “give me the fourth row, second column”. Queries must be made based on the attributes and ascribed values that exist in the table only.
- c) Finally, the third rules states that all rows must be unique with no repetition.

	FavoriteColor	FavoriteFood
PersonA	Blue	Bread
PersonB	Green	Cereal
PersonB	Red	Bread

In the above table, two “PersonB” columns cannot exist, as ambiguity is then established when one queries the database and which row the user is actually trying to access is indeterminate. It is worth noting that two identical values can exist in one column, as with the FavoriteFood column since the same attribute can exist as long as they apply to different primary key values.