1. How many different ways (considering orders of tuples and attributes) are there to represent a relation $r$ if $r$ has:

(a) Three attributes and three tuples?
(b) Four attributes and five tuples?
(c) $n$ attributes and $m$ tuples?

2. Reading and understanding formal definitions is a good skill to acquire. Consider the following definition for relations in a relational database.

A relation scheme $R$ is a non-empty set of attribute names $R = \{A_1, A_2, ..., A_n\}$. To avoid having to continuously say “names,” we say “attributes” to mean “attribute names.” Further, because we use relation schemes so much, when we have single-letter attributes, we reduce the set notation by dropping the braces, commas, and spaces. Thus, we write the set $R$ above as $A_1A_2...A_n$. Each attribute $A$ has a domain, denoted $\text{dom}(A)$, which is a set of values.

A relation is always defined with respect to a relation scheme. To denote that a relation $r$ is defined with respect to a relation scheme $R$, we write $r(R)$, read “$r$ is a relation on scheme $R$” or just “$r$ is a relation on $R$” or even just “$r$ on $R$” if we know that $r$ is a relation. A relation is a set of $n$-tuples, $\{t_1, ..., t_k\}$, where $n$ is $|R|$, the cardinality of $R$.

Let scheme $R$ be $A_1A_2...A_n$. Then, an $n$-tuple $t$ for a relation $r(R)$ is a function, from $R$ to the union of domains $D = \text{dom}(A_1) \cup \text{dom}(A_2) \cup ... \cup \text{dom}(A_n)$, with the restriction that $t(A_i) \in \text{dom}(A_i), 1 \leq i \leq n$.

We usually write a relation as a table, for example,

$$
\begin{array}{cc}
A & B \\
ar & 1 \\
b & 1 \\
b & 2 \\
\end{array}
$$

where the domains of the attributes are either implied or given. Here, we let $\text{dom}(A)$ be $\{a, b, c\}$, and $\text{dom}(B)$ be $\{0, ..., 9\}$. Show that $r$ satisfies the definition for a relation by answering the following questions.

(a) What is the relation scheme for $r$? Write this scheme using both regular set notation and the relation-scheme notation described above. (Note that this is the domain for the tuple functions.)
(b) Give the union of domains $D$ for $A$ and $B$. (Note that this is the codomain for the tuple functions.)
(c) Let the three tuples in the table be $t_1$ for the first, $t_2$ for the second, and $t_3$ for the third. Write the three functions, $t_1, t_2,$ and $t_3$ as sets of ordered pairs. Then form a set from these three sets of ordered pairs. (Observe that the set that contains the sets of ordered pairs satisfies the definition of a relation. Observe also that the transformation of a relation from a table form to a function form is straightforward.)
(d) Let $s = \{u_1, u_2\}$, where $u_1 = \{(\text{Name}, \text{Nancy}), (\text{Address}, 12 \text{Oak}), (\text{Age}, 21)\}$ and $u_2 = \{(\text{Name}, \text{Zak}), (\text{Address}, 12 \text{Oak}), (\text{Age}, 15)\}$. Write relation $s$ as a table. (Observe that the transformation of a relation from a function form to a table form is straightforward.)
(e) How many different equivalent layouts are there for table $r$, where different means a different ordering of attributes or a different ordering of tuples? (Observe that although there are many layouts, there is really only one table because rearranging the order for the elements of a set means nothing.)

(f) Evaluate: $t_1(A)$ and $t_2(B)$ for $r$ and $u_1(\text{Name})$ and $u_2(\text{Age})$ for $s$. (Observe that these are standard function evaluations.)

(g) As a shorthand, when we have a scheme, we often use the order of the attributes and write $t_1 = <a, 1>$ for $t_1 = \{(A, a), (B, 1)\}$. We also extend this idea and write $t_1(AB)$ as $<a, 1>$. Using this simplified tuple form, write $t_2(AB)$ for $r$ and $t_3(X)$ for $r$, where $X = AB$, and $u_1(Y)$ for $s$, where $Y = \text{Name Age}$. When there is only one value in this tuple form, we drop the angle brackets. Thus, $t_3(A) = <a> = a$, which is the same result obtained by simply evaluating the function $t_3$ in a standard way.

3. The following definition is the standard math definition of a relation (not a relational database relation). We want to observe that this definition of a relation differs from the relational database definition.

Let $D_1, D_2, \ldots, D_n$ be domains (sets of values). A relation $r$ is a subset of the cross product of the domains ($r \subseteq D_1 \times D_2 \times \ldots \times D_n$).

For this definition the database relation $r$ in Problem (2) is $\{<a, 1>, <b, 1>, <b, 2>\}$. Note that this is a set of ordered pairs. (Recall: $<a, 1> \neq <1, a>$, $\{a, 1\} = \{1, a\}$, and $\{<a, 1>\} \neq \{<1, a>\}$.)

(a) Observe that there are no attributes. Explain how to add them.

(b) Assuming attribute $A$ has been added for the first elements in the ordered pairs of $r$ and $B$ has been added for the second elements, we can write the relation $r$ as

$$
\begin{array}{c|c|c}
A & B \\
\hline
a & 1 \\
b & 1 \\
b & 2
\end{array}
$$

which looks identical to the relation $r$ in Problem (2). According to the definitions are the two relations equal? Explain.

(c) How many different layouts are there for table $r$ according to the definition in this problem?

4. (Thought question—no need to turn in an answer) When does it matter whether you know the particulars of detailed, formal definitions, and when does it not matter?