Relational Algebra

A *relation* is a set of tuples. In CS, this is called a database or file. The items of the relation can be seen as **attribute value pairs**, with the values being atomic ground forms (ie not composite terms and not pointers). Other terms:

Domain:	the set of possible values for a relation
Relation:	a subset of the cartesian product of domains
Attribute:	column of a relation
Item:	row of a relation, a tuple
Key:	a minimal set of attributes which identify a unique tuple

Since a relational database contains the same information as a relation table, the database must:

1. have no duplicates
2. have values from the same domain
3. have simple attribute structures (not composite)
4. an attribute must be accessed by a single key

It is always possible to express multiple argument relations using only binary relations. E.g.:

PERSON[name, age, sex] = PERSON[name, age] and PERSON[name, sex]

Operators in a Relational Algebra:

Selection: Reduce the number of rows in a table (horizontal cut)

Projection: Reduce the number of columns (vertical cut), and remove duplicates.

Restriction: Make a relation consisting of all rows which meet a functional test

Union: Combine rows of two tables (same attributes). Same as file-merge.
aka: Or, Append

Difference: The rows of relation 1 with duplicates in relation2 removed
aka: -, remove, minus

Join: Make a table with the items common to both relations.
aka: intersection

Generalized Join: For relations with unequal number of attributes, carry along the extra attributes in the new table. If the relations have no common attributes, form the cartesian product.

Hierarchy is formed in two ways:

1. *generalization* of subtypes (standard oo inheritance)
2. *aggregation* of components, making new relations of existing fields

Relational Knowledge-base Example

Vocabulary:

(father X Y)
(mother X Y)
(male Y)
(female Y)
(parent X Y)
(sibling X Y)
(brother X Y)
(sister X Y)
(uncle X Y)
(aunt X Y)
(gfather X Y)
(gmother X Y)
(ancestor X Y)
(cousin X Y)

Knowledge Base:

(if (father A B) (parent A B))
(if (mother A B) (parent A B))
(if (and (parent A C) (parent A B) (not (= B C))) (sibling B C))
(if (and (sibling A B) (male A)) (brother A B))
(if (and (sibling A B) (female A)) (sister A B))
(if (and (parent B C) (brother A B)) (uncle A C))
(if (and (parent B C) (sister A B)) (aunt A C))
(if (and (parent B C) (father A B)) (gfather A C))
(if (and (parent B C) (mother A B)) (gmother A C))
(if (parent A B) (ancestor A B))
(if (and (parent A B) (ancestor B C)) (ancestor A C))
(if (and (parent A C) (parent B D) (sibling A B)) (cousin C D))
(if (father A B) (male A))
(if (mother A B) (female A))

Facts:

(father arthur bertram)
(father arthur bailey)
(father bertram cornish)
(father bertram carey)
(mother beatrice cornish)
(mother beatrice carey)
(father bailey carleton)
(father bailey cassandra)
(mother bessie carleton)
(mother bessie cassandra)
(male cornish)
(male carey)
(male carleton)
(female cassandra)

Example questions:

(gfather arthur ?)
(cousin ? cassandra)