

# Databases Midterm Review

Andres Ponce

April 21, 2019

## 1 Relational Model

For relational models of database systems, there are usually several models. The **entity-relationship model**, which is typically used for conceptual database design, the **relational model**, typically used for logical database design, and the **physical model**, which I guess is used to describe relational systems at their most fundamental level? The universe works in mysterious ways.

The relational model emphasized the separation between the different models discussed above. This might have lead to a more abstracted view of relational models, which would have made it easier for people to adopt this method of storing information.

Why are they called relations? Well, they are based on *mathematical relations*, group items with different fields. They can be thought of as tuples, where they are inseperable connections between different parameters.

A **relation** refers to a subset of the Cartesian products of a certain number of sets. A *Cartesian product* is the set of items where the first element of every tuple comes from the first set in the product, etc...

Thus if there are  $n$  sets, the cartesian product will be all tuples containing  $n$  elements where the  $i$ th element comes from the  $i$ th set.

### 1.1 Keys

This is how we describe relations in databases. However, this presents a very limited view of what we can actually do. We can operate on these elements depending on certain characteristics. We can also order them according to certain fields.

**Keys** are the fields (set of attributes) which we can use to identify different entities (rows in a database). Common examples would be Id or names; these allow us to easily look for unique elements in our relation.

A **superkey** is a set of attributes for which every entity will have a unique value (IDs would be a good example). The **candidate key** is the minimal such set of attributes that allow us to uniquely identify elements.

Keys allow us to ensure datavase integrity, since for a key with a unique attribute, it will not be possible to store two elements with that same key. Thus, an extra layer of security might be added to a database system.

Primary keys that come from a different relation is called a *foreign key*.

## 2 Relational Algebra

By using query languages, we can make requests from a database system. Usually, we will specify a relation we want to query (this might be a subset of an existing relation or a new relation altogether). These queries provide a relation instance as a result. Everything is a relation!

The six basic operators in relational algebra:

1. **select:**  $\sigma$
2. **project:**  $\Pi$
3. **union:**  $\cup$

4. **set different:** -
5. **Cartesian product:**  $\times$
6. **rename:**  $\rho$

## 2.1 selection $\sigma$

The selection operator retrieves multiple tuples out of a relation. We can choose which of the elements in the relation to retrieve by using a boolean condition. We can either check a property of the row itself, or we can compare it to another relation or value. Thus the selection operator could have either 1 or 2 arguments. In query languages such as SQL, we can choose to retrieve either the entire entry in the relation or we can choose certain of the fields.

Because the formal definition of a set, any duplicated rows will be 'selected out' from the final relation; that is, they will be excluded.

Also, since the result of an expression is (almost always) a relation, we can nest relations in each other.

## 2.2 Union $\cup$

The  $\cup$  operator in set notation functions like an 'or'. It retrieves all the rows in the relations that meet the criteria. It can check the relations that occur in each set individually or the ones that occur in both sets (so like an inclusive or).

To ensure the ability to be able to make the unions, both sets have to be compatible, either having the same number of fields or being otherwise comparable.

## 2.3 difference

The difference of two sets A and B can be expressed by the phrase: 'The set of elements that are in A but not in B'. Same as before, we have to be able to compare the two sets in this manner.

## 2.4 Intersection

The intersection of two sets is the set of the elements that are common to all sets. However, intersection is not an elemental operation, since it can be composed of

$$R \text{ and } S = R - (R - S)$$

## 2.5 Cartesian Product

The cartesian product, as mentioned before for n sets, is the set of tuples where the ith element comes from the ith set in the relation. For two sets, the cartesian product will contain n x m elements in it.

To compose a query in relational algebra we require to specify a new set for every query we want to make.

## 2.6 Join $\bowtie$

The join operation denotes the mixture of the cartesian product and selections operation. It is basically the selection from a cross-product of relations(based on a certain criteria). We would say we are joining *on* a certain value or field.

We specify a *join condition*, which we use to join two relations together on the occurrence of that criteria.

## 2.7 Equi-Join

The equi-join is a type of join where we join based on the equality of two fields in certain relations. So joining on the equality of two different fields is what we were doing in the first homework.

## 2.8 Natural Join $\bowtie$