

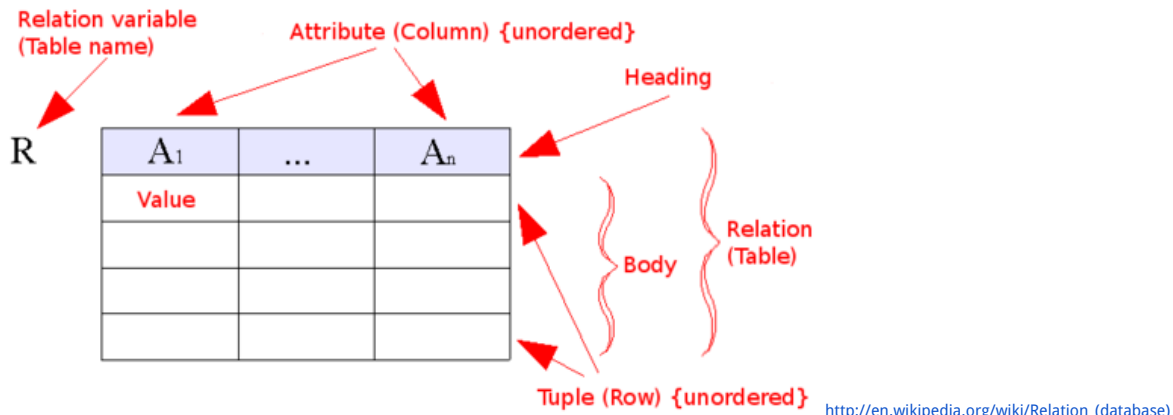
# Relational Database

Definition:

A database that conforms to relational model theory.

The purpose of a relational model is to provide a declarative method for specifying data and queries.

Relational Database uses a set of mathematical terms.



Relation (Table): Data structure which consists of a heading and an unordered set of tuples which share the same type.

Relation value: An instance of a relation.

Relation variable: A variable which has a relation value

Heading: unordered set of certain attributes. Column headers.

Body: unordered set of tuples

Tuple (Row): Represents an unordered set of attributes which includes an object and information of that object.

Attribute (Column): A pair of attribute name and domain name. Domain can be considered data type.

Degree of relation: the number of attributes which constitute the heading. An n-ary relation is a relation value in which its degree is n.

Cardinality of a relation: the number of tuples which constitute a relation value.

Example of a of a heading with 3 attributes:

Attribute name: Domain name
Emp ID: Integer
Emp Name: String
Department: String

Example of a relation value with the above heading and tuples which conforms to the attributes:

Emp ID: Integer	Emp Name: String	Department: String
001	A B C	Sales
555	D E F	HR
110	G H I	IT

Tuples are unordered. It is hence not correct to say that:

Emp name is to the right of Emp ID

Tuple containing Emp ID 001 is above tuple containing Emp ID 555

### **Base relation variable versus Derived relation variable**

Base relation variable is a source relational variable that is not derived from any other relation variable. Base table.

Example of a base relation variable:

Store ID	Store Location	Sales
0125	USA	20000
0126	Singapore	15000
0127	UK	10000

Derived relation variable is a relation variable that is derived from the base relation variable by applying relational algebra or relational calculus expression. Design restrictions on base table and display.

Example of a derived relation variable (with the restriction Sales  $\geq$  15000):

Store ID	Store Location	Sales
0125	USA	20000
0126	Singapore	15000

Tuple with Store ID 0127 not displayed because the Sales attribute of this tuple does not meet the stated restriction.

### **Primary Key**

A primary key uniquely defines the relationship within a database. It must not repeat. Natural attributes, attributes that mean something to the database are good primary keys but in certain cases they may be repeated, e.g. name. Hence a surrogate key is introduced, mostly in the form of ID e.g. Employee ID. This key has no intrinsic meaning and its purpose is the unique identification of each tuple.

### **Foreign Key**

A field in a relation value that matches the candidate key of another table. Consider a STORE relation and a SALES relation. A particular sales with its unique ID will only be associated with a single store. Hence a foreign key will be present in SALES which matches the primary key (ID) of STORE.

### **Superkey**

Superkey is a set of attributes of a relation value for which a not two tuples have the same value for these attributes in the set. Set of attributes within a table which values can be used to uniquely identify a tuple.

Example of a superkey in a table:

Emp ID: Integer	Emp Name: String	Department: String
001	A B C	Sales
555	D E F	HR
110	G H I	IT
896	A B C	Sales
987	J K L	Sales

{Emp Name, Department} **cannot** be a superkey because 001 and 896 have exact same attribute values of the set even when they are two distinct tuples (as seen from Emp ID).

Suitable superkeys: {Emp ID}, {Emp ID, Emp Name}, {Emp ID, Department}, {Emp ID, Emp Name, Department}

{Emp ID, Emp Name, Department} is a trivial superkey because it is the set of all attributes of this relation value.

{Emp ID} is a candidate key, a minimal superkey, the minimal set of attribute/s that is required to identify a single tuple.

### **Integrity Rules**

Entity Integrity: every relation must have a primary key, and primary key is unique, no 2 rows can have the same primary key.

Domain integrity: restricting data to predefined data types

Referential integrity: every value of attribute to exist as a value of another attribute in a different or same relation. A foreign key declared in a relation can only contain values found as a primary/candidate key of another table.