

Introduction to databases

The Dull Day of Databases

Database Management Systems

- ▶ Designed for storing, managing, and retrieving information
- ▶ We will be working with Relational Databases



Database Management Systems

- ▶ Data is separate from manipulations of the data
- ▶ Tables – store the data
- ▶ Queries – stores questions about the data



Outline

- ▶ Introduction to Relational Databases
- ▶ Example
- ▶ Alternative database structures used in science
- ▶ Types
- ▶ Primary Keys
- ▶ SQL
- ▶ Examples in our Database Management System



Relational Databases

- ▶ **Data is stored in tables**
 - ▶ One table per type of data
 - ▶ Tables can be linked together to combine information
- ▶ **Each row contains a single record**
 - ▶ A single observation or data point
- ▶ **Each column contains a single attribute**
 - ▶ A single type of information



Example

IndivID	SpeciesID	BodyMass	HindFoot



Example

IndivID	SpeciesID	BodyMass	HindFoot
1	DS	110.5	48
2	NA	201.2	34
3	SH	75.7	28



Example

IndivID	SpeciesID	BodyMass	HindFoot
1	DS	110.5	48
2	NA	201.2	34
3	SH	75.7	28



Multiple tables

- ▶ It is often not efficient to include all information of interest in a single table
- ▶ To solve these problems we store data in multiple tables
- ▶ And connect the data in different tables using Joins or Relationships (hence Relational Database)



One table per data type

IndivID	SpeciesID	BodyMass	HindFoot
1	DS	110.5	48
2	NA	201.2	34
3	SH	75.7	28



One table per data type

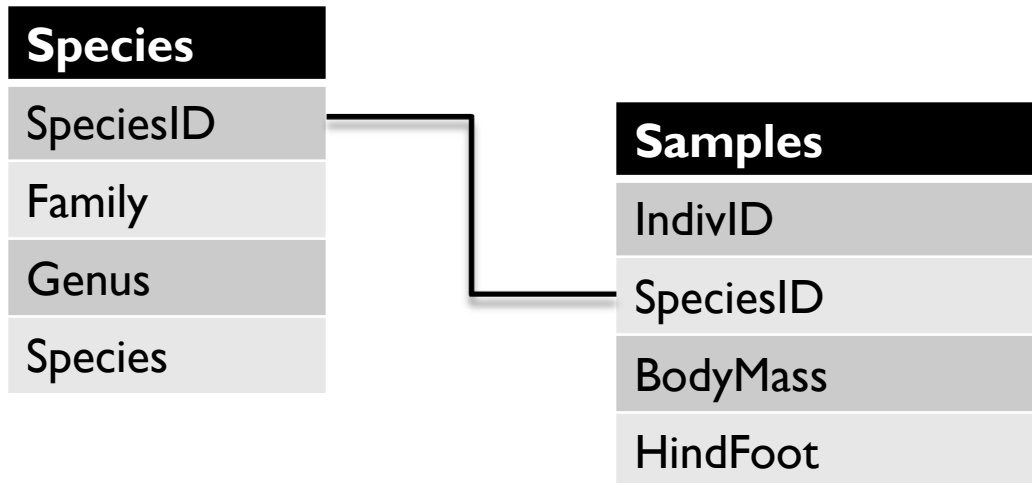
IndivID	SpeciesID	BodyMass	HindFoot
1	DS	110.5	48
2	NA	201.2	34
3	SH	75.7	28

Samples
IndivID
SpeciesID
BodyMass
HindFoot



One table per data type

IndivID	SpeciesID	BodyMass	HindFoot
1	DS	110.5	48
2	NA	201.2	34
3	SH	75.7	28



Alternative structures

	Species 1	Species 2	Species 3
Plot 1	23	5	2
Plot 2	32	10	0
Plot 3	10	20	5



Alternative structures

	Species 1	Species 2	Species 3
Plot 1	23	5	2
Plot 2	32	10	0
Plot 3	10	20	5



Plot	Species	Count
1	1	23
1	2	5
1	3	2
2	1	32
2	2	10
3	1	10



Types

- ▶ Fields in databases have Types just like variables in Python
 - ▶ Types need to be specified in advance

IndivID	SpeciesID	BodyMass	HindFoot



Types

- ▶ Fields in databases have Types just like variables in Python
 - ▶ Types need to be specified in advance

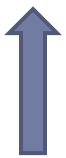
IndivID	SpeciesID	BodyMass	HindFoot
1	DS	110.5	48
2	NA	201.2	34
3	SH	75.7	28



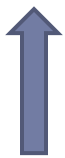
Types

- ▶ Fields in databases have Types just like variables in Python
 - ▶ Types need to be specified in advance

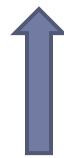
IndivID	SpeciesID	BodyMass	HindFoot
1	DS	110.5	48
2	NA	201.2	34
3	SH	75.7	28



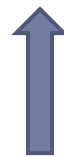
Integer



Text



Double
Decimal



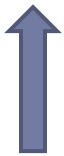
Integer



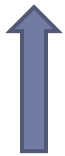
Types

- ▶ Types are highly configurable
 - ▶ Because space is important

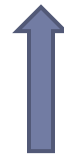
IndivID	SpeciesID	BodyMass	HindFoot
1	DS	110.5	48
2	NA	201.2	34
3	SH	75.7	28



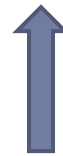
Integer
How long?



Text
How long?



Decimal
How many values
before & after the
decimal place?



Integer
How long?



Primary keys

- ▶ Every table in a database “needs” a column (or a set of columns) that is unique across records
- ▶ This is called the primary key
- ▶ The easiest way to do this is to use an Integer that increments every time a new record is added
- ▶ Many databases that you import will already have a field like this



SQL – structured query language

```
CREATE DATABASE MammalSurveys;  
CREATE TABLE SurveyData (  
    Individ INT,  
    SpeciesID VARCHAR,  
    BodyMass INT,  
    HindFoot INT,  
    PRIMARY KEY (Individ)  
);
```

