Exploratory Data Analysis with R

Database Creation

Xuemao Zhang
East Stroudsburg University

November 28, 2022

Outline

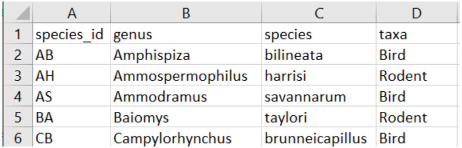
- Database structure
- Table import
- SQL syntax

Database structure

- We first create an empty database `Database1`, and then import three tables to the database.
  - `species.csv`
  - `surveys.csv`
  - `plots.csv`
- Open each of these csv files and explore them. What information is contained in each file? How are the three tables related?
- The database is copied from Introducing Databases and SQL
- Dataset Description: The data set is a time-series for a small mammal community in southern Arizona. This is part of a project studying the effects of rodents and ants on the plant community that has been running for almost 40 years. The rodents are sampled on a series of 24 plots, with different experimental manipulations controlling which rodents are allowed to access which plots.

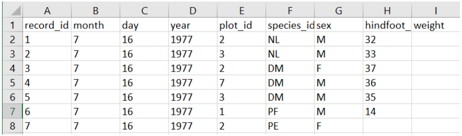
Database structure

species.csv



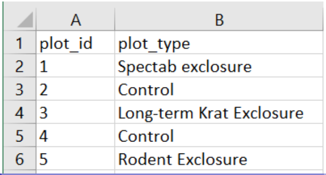
	A	B	C	D
1	species_id	genus	species	taxa
2	AB	Amphispiza	bilineata	Bird
3	AH	Ammospermophilus	harrisi	Rodent
4	AS	Ammodramus	savannarum	Bird
5	BA	Baiomys	taylori	Rodent
6	CB	Campylorhynchus	brunneicapillus	Bird

surveys.csv



	A	B	C	D	E	F	G	H	I
1	record_id	month	day	year	plot_id	species_id	sex	hindfoot_	weight
2	1	7	16	1977	2	NL	M	32	
3	2	7	16	1977	3	NL	M	33	
4	3	7	16	1977	2	DM	F	37	
5	4	7	16	1977	7	DM	M	36	
6	5	7	16	1977	3	DM	M	35	
7	6	7	16	1977	1	PF	M	14	
8	7	7	16	1977	2	PE	F		

plots.csv



	A	B
1	plot_id	plot_type
2	1	Spectab exclosure
3	2	Control
4	3	Long-term Krat Exclosure
5	4	Control
6	5	Rodent Exclosure

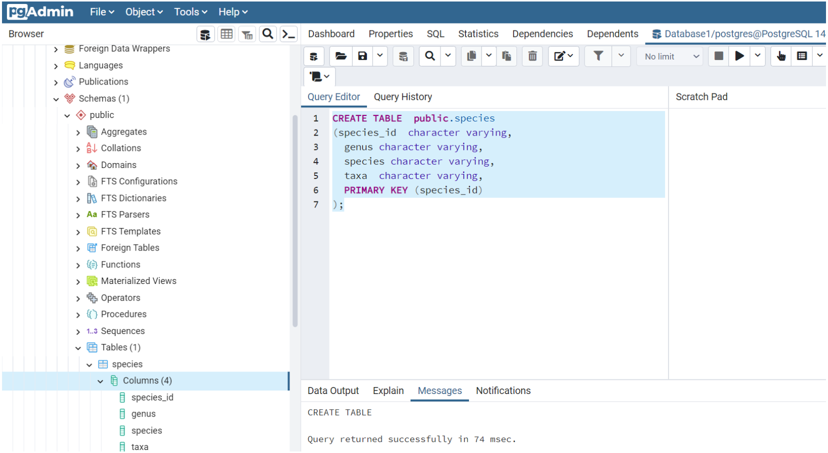
Table import

- Let's import the three tables to the database Database1.
  - Import the parent tables species and plots first, and then import table surveys
- **Step 1**: Create a table in PgAdmin and match the column headers in the table to the columns in your CSV.
  - Write down the following query in the Query Editor

```
CREATE TABLE  public.species
(species_id  character varying,
  genus character varying,
  species character varying,
  taxa  character varying,
  PRIMARY KEY (species_id)
);
```

Table import

- Now a table named `species` has been created.



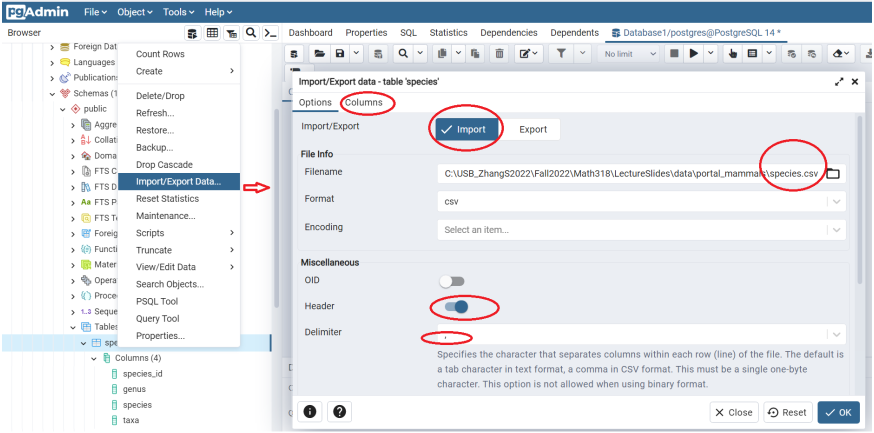
```
CREATE TABLE  public.species
(species_id  character varying,
  genus character varying,
  species character varying,
  taxa  character varying,
  PRIMARY KEY (species_id)
);
```

```
CREATE TABLE

Query returned successfully in 74 msec.
```

Table import

- **Step 2**: Right click the table and choose `Import/Export Data...`. See the detail below



Import/Export data - table 'species'

Options | Columns

Import/Export: [✓ Import] Export

File Info

Filename: C:\USB_ZhangS2022\Fall2022\Math318\LectureSlides\data\portal_mammals\species.csv

Format: csv

Encoding: Select an item...

Miscellaneous

OID: □

Header: ☑

Delimiter: ,

Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

Table import

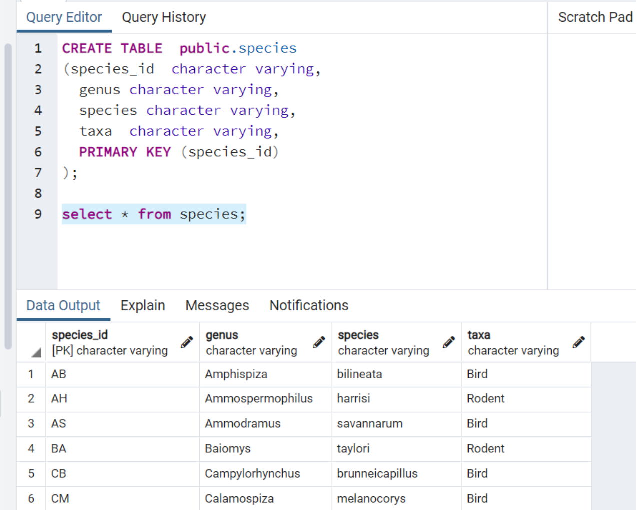
- Another way to complete **Step 2**: Writing an SQL Query that references your CSV file path

```
COPY species -- it refers to the table name in the query above
FROM 'C:\USB_ZhangS2022\Fall2022\Math318\
LectureSlides\data\portal_mammals\species.csv' DELIMITER ','
CSV Header;
```

Table import

- Double check if the data is loaded

```
select * from species;
```



```
CREATE TABLE  public.species
(species_id  character varying,
  genus character varying,
  species character varying,
  taxa  character varying,
  PRIMARY KEY (species_id)
);

select * from species;
```

	species_id [PK] character varying	genus character varying	species character varying	taxa character varying
1	AB	Amphispiza	bilineata	Bird
2	AH	Ammospermophilus	harrisi	Rodent
3	AS	Ammodramus	savannarum	Bird
4	BA	Baiomys	taylori	Rodent
5	CB	Campylorhynchus	brunneicapillus	Bird
6	CM	Calamospiza	melanocorys	Bird

Table import

- Import the table plots similarly
  - Again, we first create an empty table using the Query Editor

```
CREATE TABLE  public.plots
(plot_id  integer,
  plot_type character varying,
  PRIMARY KEY (plot_id)
);
```

```
COPY plots
FROM 'plots.csv' DELIMITER ',' -- you may need to change the path
CSV Header;
```

Table import

- Last, we import the table surveys

```
CREATE TABLE  public.surveys
(record_id integer,
  month integer,
  day integer,
  year integer,
  plot_id integer,
  species_id  character varying,
  sex character(1),
  hindfoot_length integer,
  weight integer,
  PRIMARY KEY (record_id),
  CONSTRAINT fk1_surveys
      FOREIGN KEY(species_id)
      REFERENCES species(species_id),
    CONSTRAINT fk2_surveys
      FOREIGN KEY(plot_id)
      REFERENCES plots(plot_id)
);
```

Table import

- Last, we import the table `surveys`

```
COPY surveys
FROM 'surveys.csv' DELIMITER ',' -- you may need to change the path
CSV Header;
```

- Now, the database `Database1` with three tables have been created

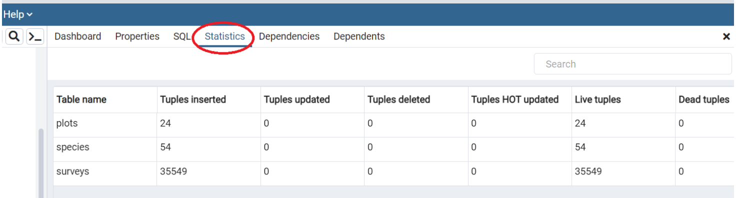


Table name	Tuples inserted	Tuples updated	Tuples deleted	Tuples HOT updated	Live tuples	Dead tuples
plots	24	0	0	0	24	0
species	54	0	0	0	54	0
surveys	35549	0	0	0	35549	0

Table import

- A simpler example with one table only from youtube: SQL SELECT Tutorial |¦| SQL Tutorial |¦| SQL for Beginners
  - Data: https://github.com/socratica/data

```
CREATE TABLE  public.earthquake
(earthquake_id integer,
    occurred_on timestamp without time zone,
    latitude numeric,
    longitude numeric,
    depth numeric,
    magnitude numeric,
    calculation_method character varying,
    network_id character varying,
    place character varying,
    cause character varying,
    PRIMARY KEY (earthquake_id)
);

COPY earthquake
FROM 'earthquake.csv' DELIMITER ','
CSV Header;
```

SQL syntax

- All SQL statements start with an SQL keyword
- All SQL statements end with a semicolon
  - Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.
- SQL keywords are NOT case sensitive: `select` is the same as `SELECT`
- Identifiers (used to name tables, columns, etc.) are not case sensitive
- Tables are defined within a (default) **schema**
  - A database contains one or more named schemas, which in turn contain tables.
  - Unlike databases, schemas are not rigidly separated: a user can access objects in any of the schemas in the database they are connected to, if they have privileges to do so.
  - https://www.postgresql.org/docs/15/ddl-schemas.html
- You can use comments within your SQL statements. There are several ways
  - `-- comment goes here`
  - `/* comment goes here */`
  - could be other ways depending on a specific DBMS vendor

SQL syntax

Some of The Most Important SQL **Key Words**:

- `SELECT` - extracts data from a database
- `UPDATE` - updates data in a database
- `DELETE` - deletes data from a database
- `INSERT INTO` - inserts new data into a database
- `CREATE DATABASE` - creates a new database
- `ALTER DATABASE` - modifies a database
- `CREATE TABLE` - creates a new table
- `ALTER TABLE` - modifies a table
- `DROP TABLE` - deletes a table
- `CREATE INDEX` - creates an index (search key)
- `DROP INDEX` - deletes an index

License