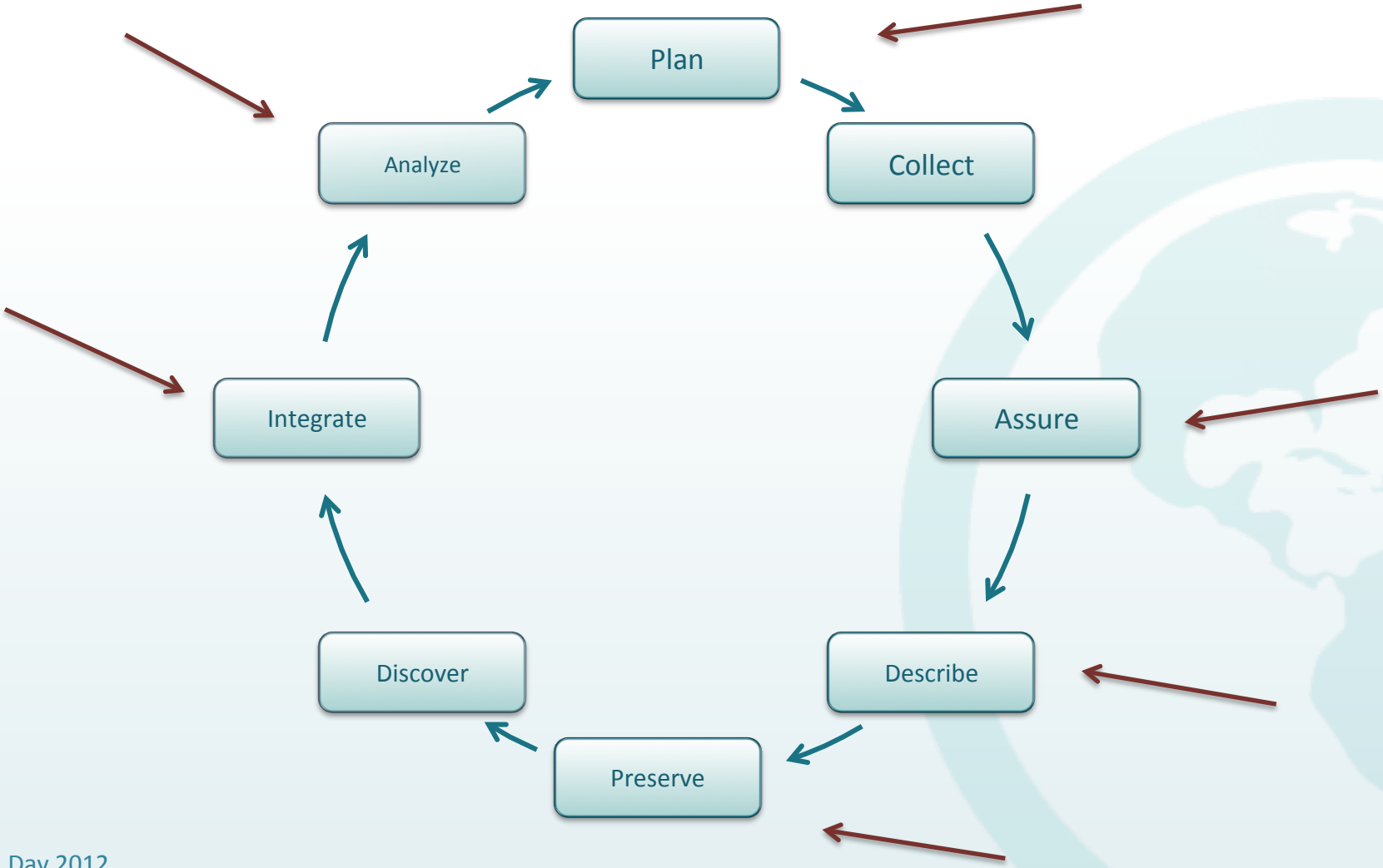
Database Management Systems

Rebecca Koskela
DataONE
University of New Mexico

DataONE

Databases and the Data Life Cycle



Database

- Collection of data stored for a specific purpose
- Structured data
  - —Typically organized by records
  - —And relationships among the records

Example Database Systems

Microsoft Access

FileMaker

Microsoft SQL Server

Oracle

IBM DB2

MySQL

PostgreSQL

SQLite

Comparison

RDBMS	Vendor	Start	Windows	Mac OS	Linux
Access	Microsoft	1992	x		
FileMaker	FileMaker/ Apple	~1983	x	x	
SQL Server	Microsoft	1989	x		
Oracle	Oracle	1979	x	x	x
DB2	IBM	1983	x	x	x
mySQL	Sun/Oracle	1995	x	x	x
postgreSQL	Open source	1989	x	x	x
SQLite	Open source	2000	x	x	X

Comparison (cont.)

RDBMS	Max DB Size
Access	2GB
FileMaker	8 TB
SQL Server	524 258 TB
Oracle	Unlimited
DB2	512 TB
mySQL	Unlimited
postgreSQL	Unlimited
SQLite	32 TB

Advantages of Databases

- Each piece of data is stored only once
- Data integration
  - Field data and lab data
- Maintains the relationships among data
- Easy to subset data
- Automated reports

Spreadsheets vs. Databases

Why spreadsheets?

Familiarity

Drawbacks of spreadsheets:

- Little or no protection against data corruption
- Little or no data validation
- Size limitations

Spreadsheets vs. Databases

Advantages of databases:

- Multi-user access
- Data integrity and data validation
- Protect data from inadvertent corruption
- Reduce data duplication
- Easy to generate reports
- Easy to subset data

Database Design

Goals:

- Supports required and ad hoc information
- Proper and efficient tables
- Data integrity imposed at field, table, and relationship levels
- Supports business rules
- Extensible

Schema

Structure of the database that defines the objects in the database including tables, fields, relationships

Conceptual schema

Logical schema

Physical schema

Conceptual/Logical Schema

- Description of a particular collection of data, using a given data model
- For a relational data model,
  - Define entities (tables)
  - Define attributes (columns) for each entity
  - Specify relationships between entities

Physical Schema

- Physical organization of the data
  - —SQLite, file-based database
  - —Reads and writes to ordinary disk files

Relationships

One-to-one: Record in parent table is related to one and only one record in child table

One-to-many: Record in parent table is related to zero or more records in child table

Many-to-many: Multiple records in parent table are related to multiple records in the child table

Maintaining Relationships

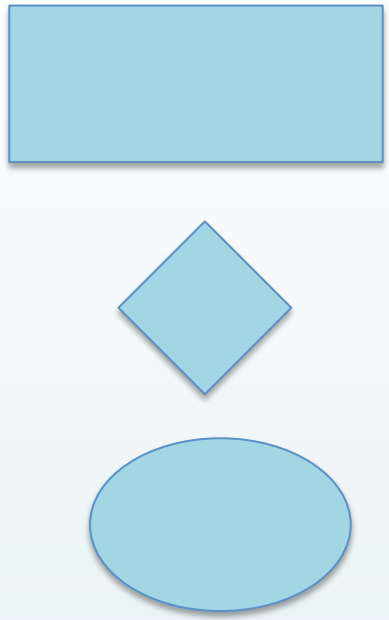
Primary key

- No intrinsic meaning – used to uniquely define a tuple (row)

Foreign key

- Reference to a key in another table – implements relationships between tables

Entity-Relationship Diagrams (ERD)



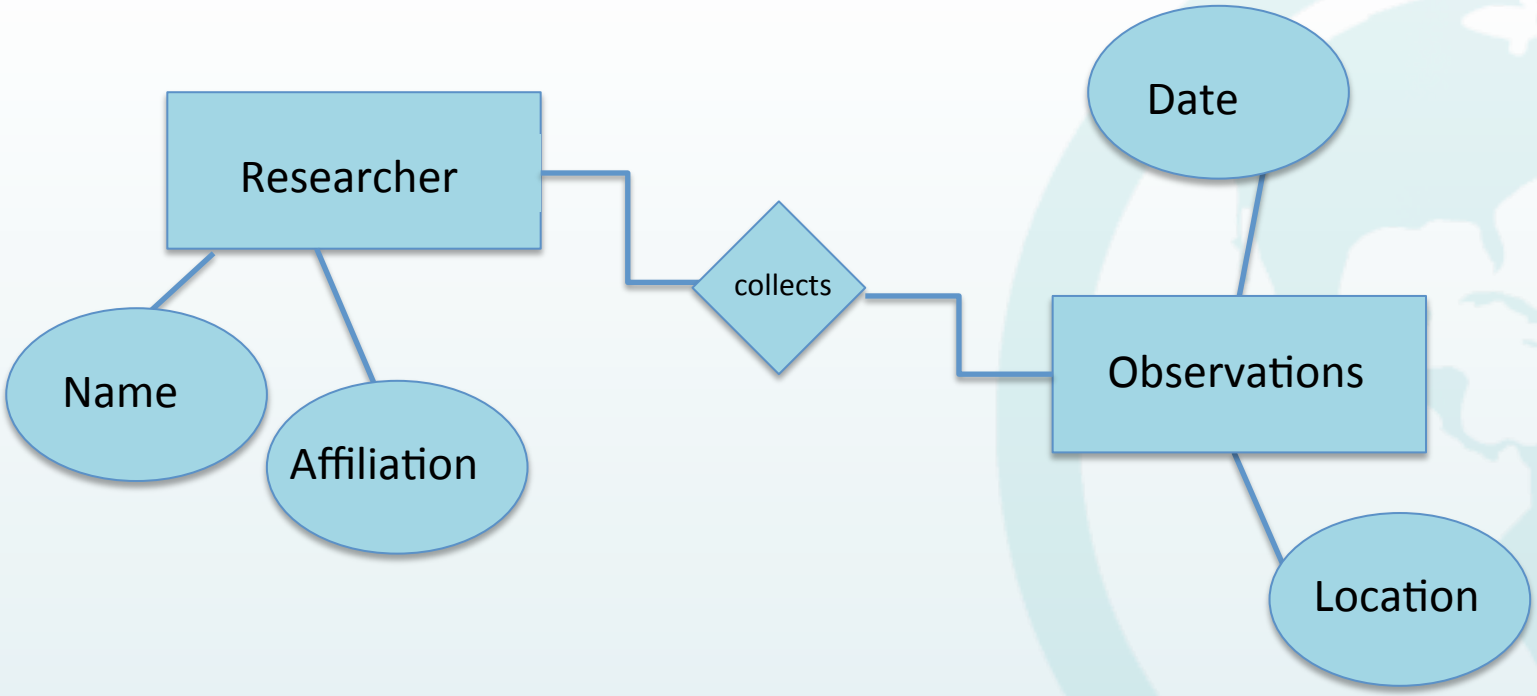
Entity

Relationship

Attribute

Creating an ERD

Researcher collects field data



Identifying Relationships

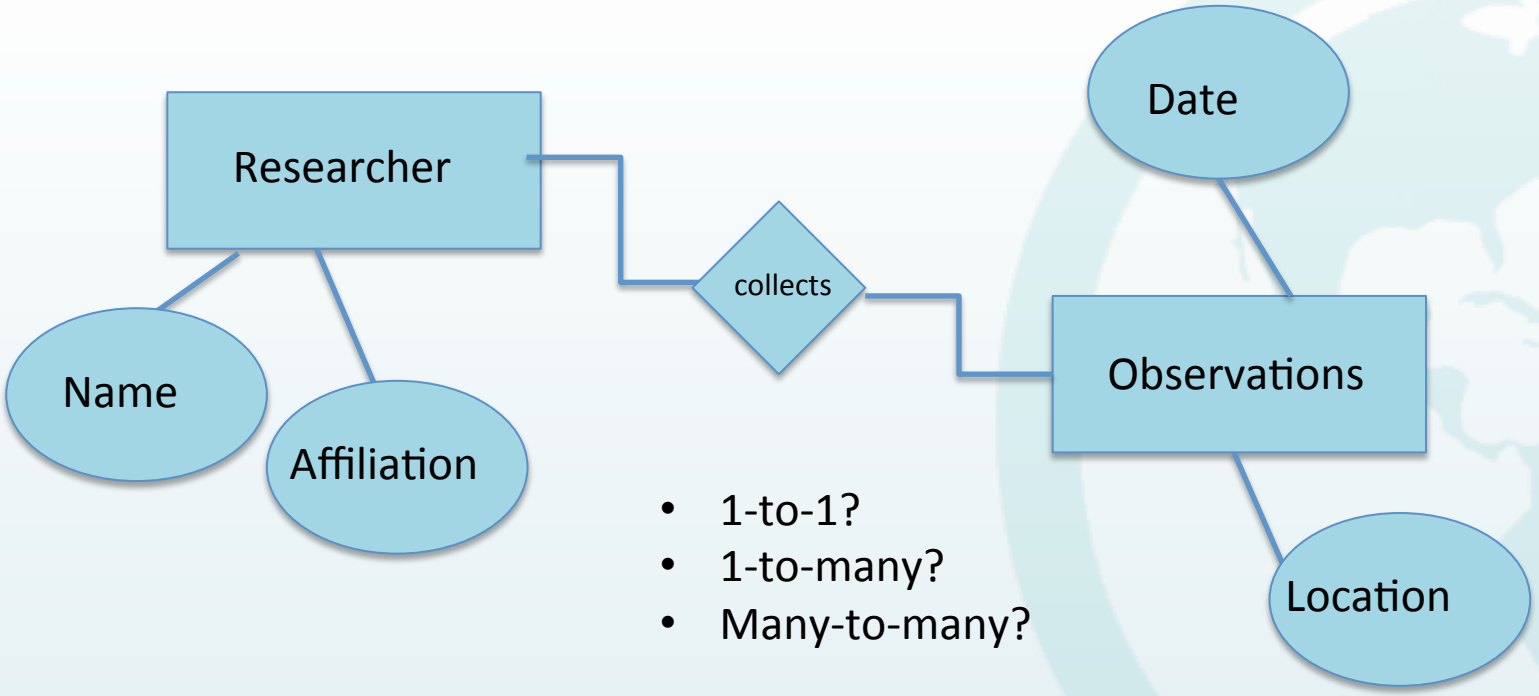
If “entity” relationship is 1-to-1, use one table

If relationship is 1-to-many, use 2 tables with the primary key of the 1-entity as the foreign key of the many-entity

If relationship is many-to-many, use 3 tables; 1 for each entity and 1 more (FK1,FK2) for mapping the many-to-many relationship

Creating an ERD

Researcher collects field data



- 1-to-1?
- 1-to-many?
- Many-to-many?

Normalization

Process of organizing the fields and tables of a relational database to minimize redundancy and dependency

Usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them

Objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table

Example

Customer ID	First Name	Surname	Telephone
123	Robert	Smith	505-861-2025
456	Jane	Wright	505-403-1659
789	Maria	Garcia	505-808-9633

Added requirement: multiple phone numbers

One possibility

Customer ID	First Name	Surname	Telephone 1	Telephone 2
123	Robert	Smith	505-861-2025	
456	Jane	Wright	505-403-1659	505-776-4100
789	Maria	Garcia	505-808-9633	

- Difficulty in querying the table.
  - "Which customers have telephone number *X*?" and
  - "Which pairs of customers share a telephone number?"
- Inability to enforce uniqueness of Customer-to-Telephone Number links through the RDBMS.
- Database design is imposing constraints on the business process, rather than (as should ideally be the case) vice-versa.

Instead

CustomerName

Customer ID	First Name	Surname
123	Robert	Smith
456	Jane	Wright
789	Maria	Garcia

Customer ID	Telephone
123	505-861-2025
456	505-403-1659
789	505-808-9633
456	505-776-4100

CustomerTelephone

Data Types for Attributes

- Numeric
  - Integer
  - Real, float
- Text
  - String (varchar)
  - Character
- Date
  - Stored as text("YY-MM-DD HH:MM:SS.SSS), real (Julian), or integer (Unix time)
- Boolean
  - stored as integers, 0=false, 1=true

Null

Represents missing or unknown value

- Not zero
- Not blank (‘ ‘)
- Not a zero-length string (‘’)

Creating a Database

Tables

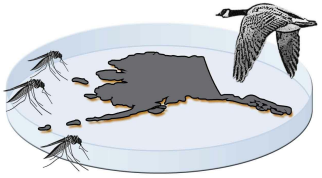
- Start by identifying all the nouns

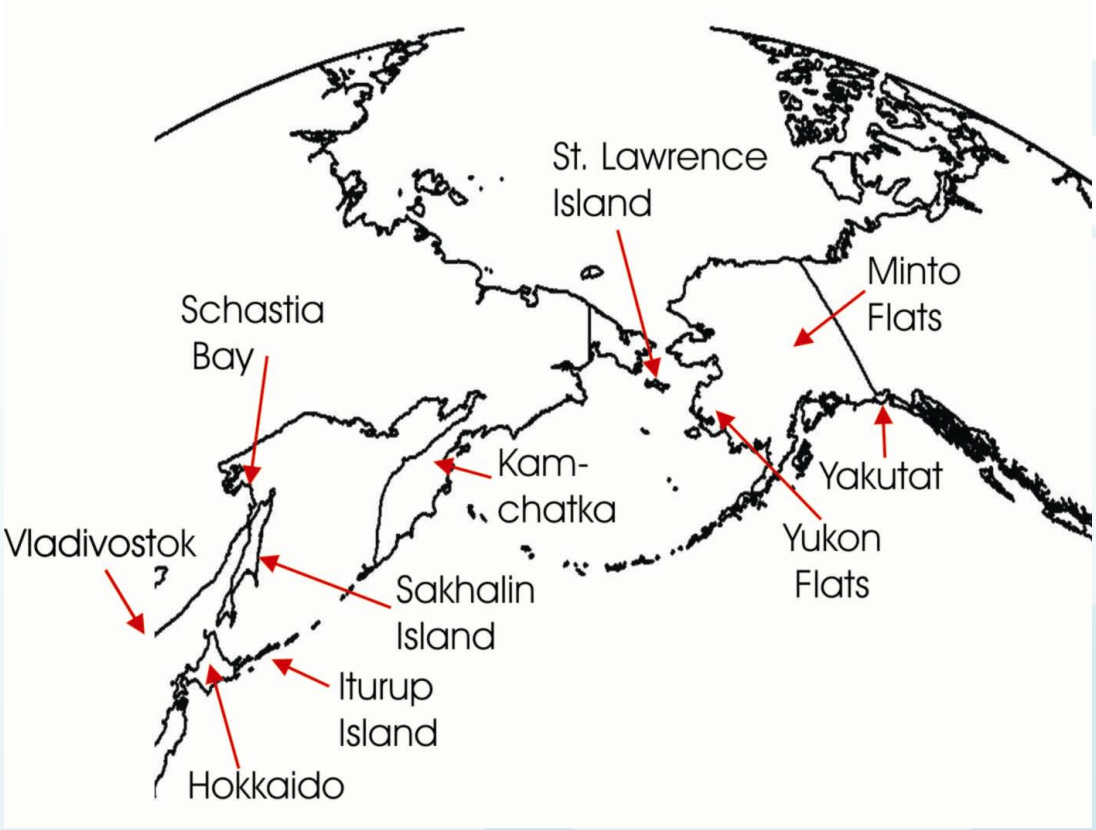
Attributes

Relationships among the tables

Alaska Asian Avian Influenza Research

Alaska Zoonotic Disease Center





Center Tasks

- Collect samples for avian influenza surveillance in Alaska, Russia, Mongolia, and Japan
- Screen samples by RT-PCR and/or virus isolation for influenza viruses
- Genetic characterization of positive samples
- Archive samples

Conceptual Schema

Semantics

Avian Influenza project:

—Birds are captured and swabbed to collect sample that is put in vial with barcode

—Location, date, species, age, sex, bird band, recapture status are recorded

—Morphological measurements are taken

Logical Schema

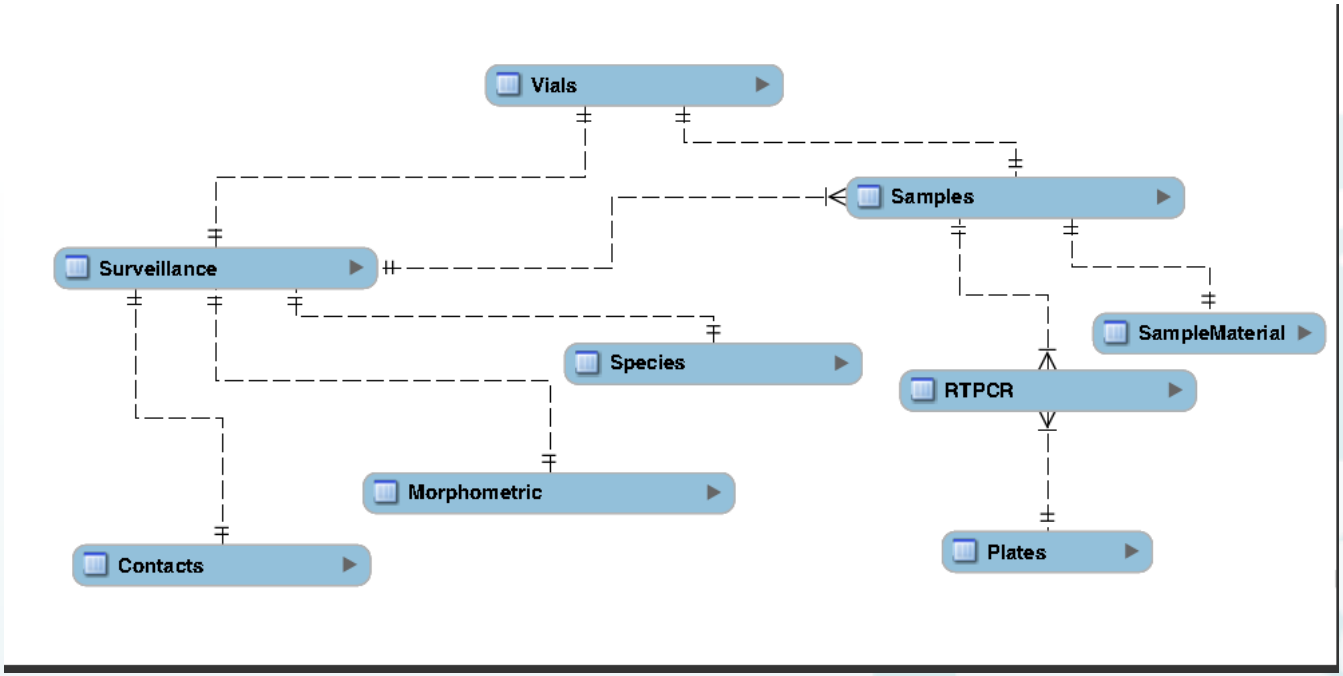
Data model

1. Define entities (tables)
2. Define attributes (columns) for each entity
3. Specify relationships between entities

Field Data

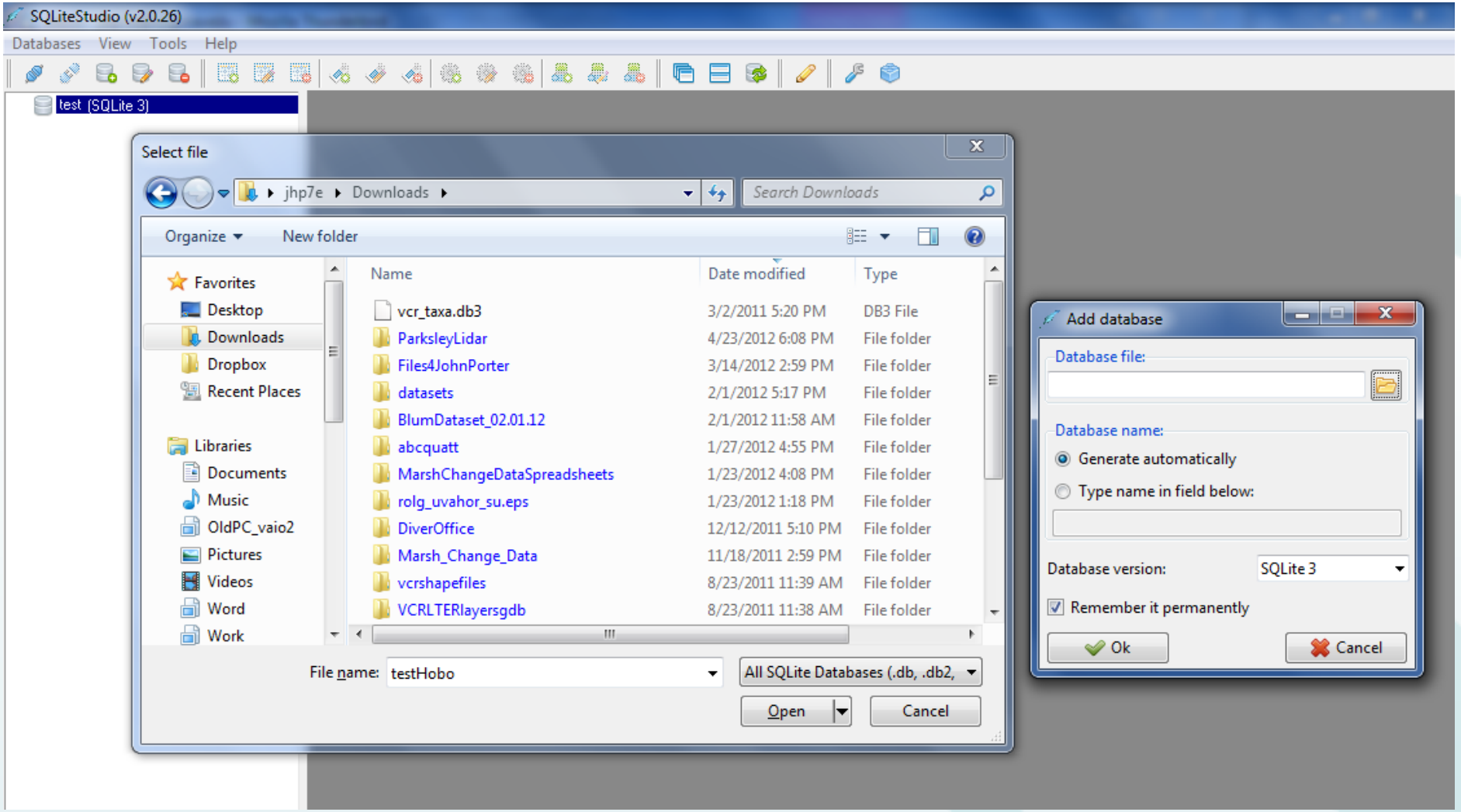
Barcode	Bird_ID	Species	Sex/Age	Type	Location/GPS	Date	Notes
8BM3001	1837-22186	MALL	HYF	C	Y Lakes N	8/17/08	Recap
8BM3002	934-93127	AGWT	AHYM	C	Y Lakes N	8/17/08	Recap
8BM3003	934-93115	AGWT	AHYM	C	Y Lakes N	8/17/08	Recap
8BM3004	1837-22317	MALL	AHYM	C	Main Channel	8/17/08	
8BM3005	1837-22318	MALL	AHYM	C	Main Channel	8/17/08	
8BM3006	1837-22319	MALL	HYM	C	Main Channel	8/17/08	
8BM3007	1837-22320	MALL	AHYF	C	Main Channel	8/17/08	
8BM3008	1837-22321	MALL	AHYM	C	Main Channel	8/17/08	
8BM3009	1837-22021	MALL	HYF	C	Main Channel	8/17/08	
8BM3010	1096-79463	NOPI	AHYM	C	Main Channel	8/17/08	
8BM3011	1146-17396	NOPI	AHYM	C	Cabin	8/17/08	
8BM3012	1146-17397	NOPI	AHYM	C	Cabin	8/17/08	
8BM3013	1146-17398	NOPI	AHYM	C	Cabin	8/17/08	
8BM3014	1146-17399	NOPI	HYF	C	Cabin	8/17/08	
8BM3015	1146-17400	NOPI	AHYM	C	Cabin	8/17/08	
8BM3016	1146-17401	NOPI	HYF	C	Cabin	8/17/08	
8BM3017	1146-17402	NOPI	AHYM	C	Cabin	8/17/08	
8BM3018	1146-17403	NOPI	HYM	C	Cabin	8/17/08	
8BM3019	1146-17404	NOPI	AHYM	C	Cabin	8/17/08	
8BM3020							
8BM3021	1146-17405	NOPI	AHYF	C	Cabin	8/17/08	
8BM3022	1146-17406	NOPI	AHYM	C	Cabin	8/17/08	
8BM3023	1146-17186	NOPI	HYM	C	Cabin	8/17/08	Recap
8BM3024	1096-79466	NOPI	HYM	C	Cabin	8/17/08	Recap
8BM3025	1146-17278	NOPI	AHYF	C	Cabin	8/17/08	Recap

Logical Schema Example

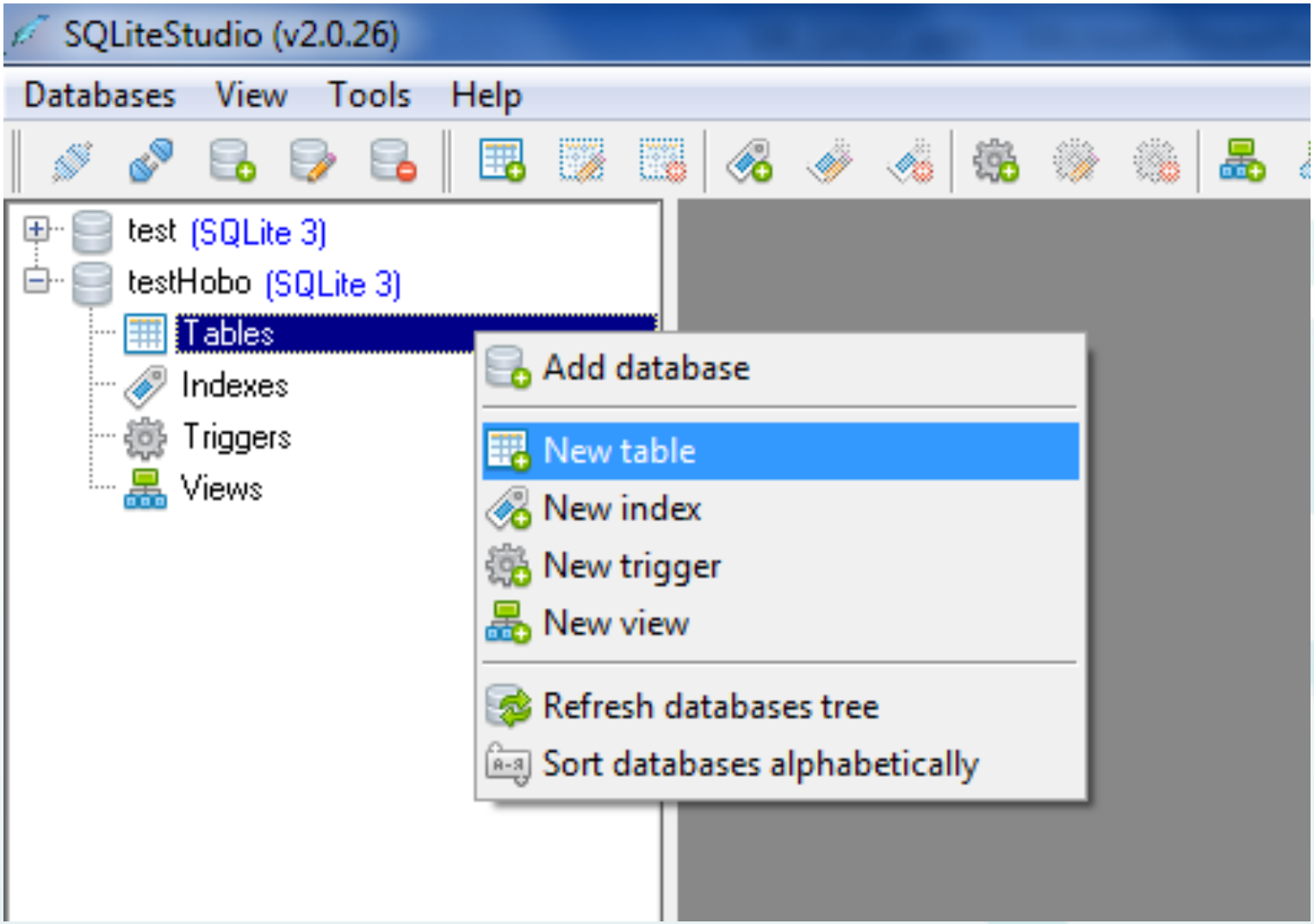


Physical Schema: Depends on which RDBMS you choose

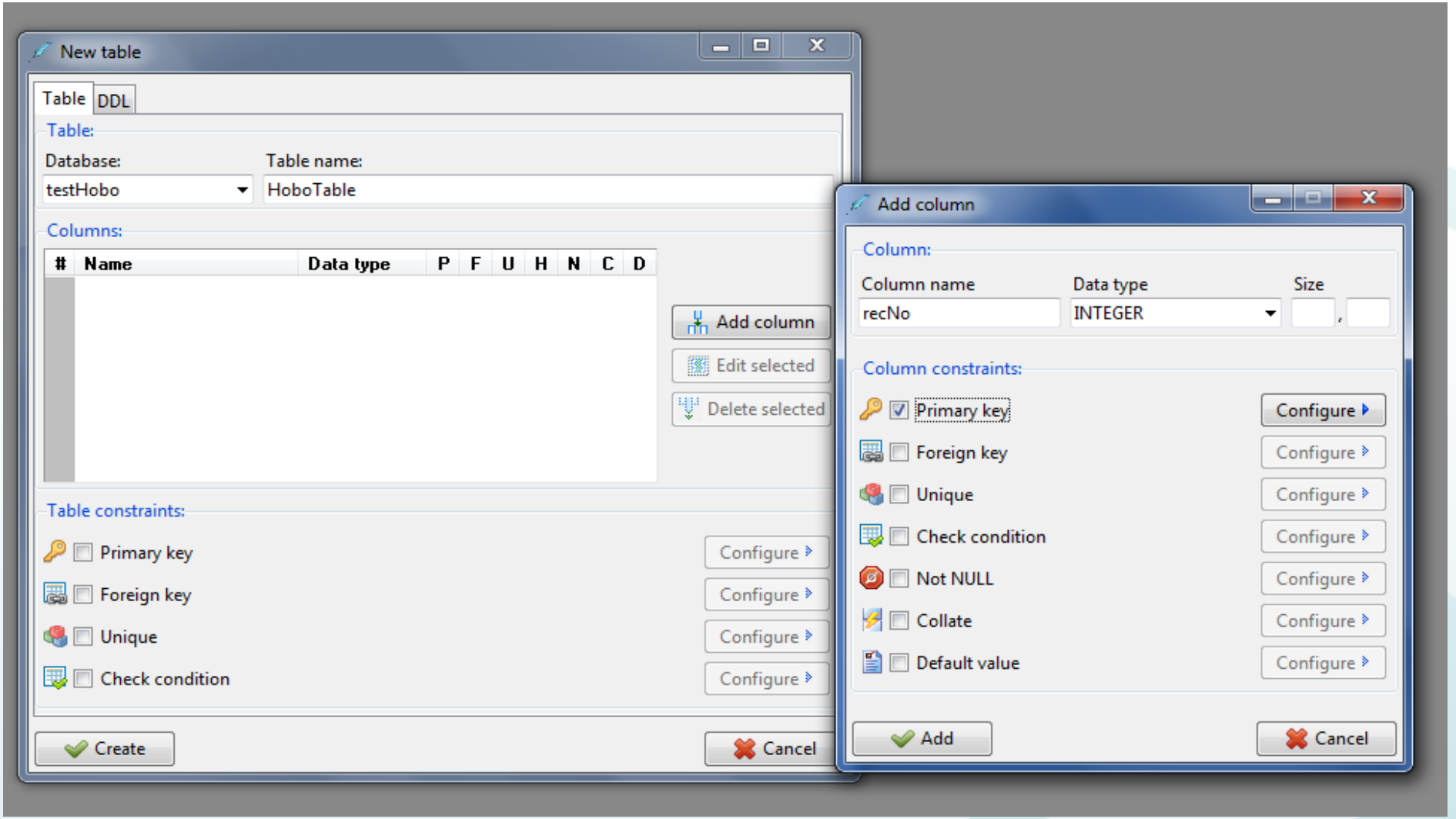
Create Database in SQLite



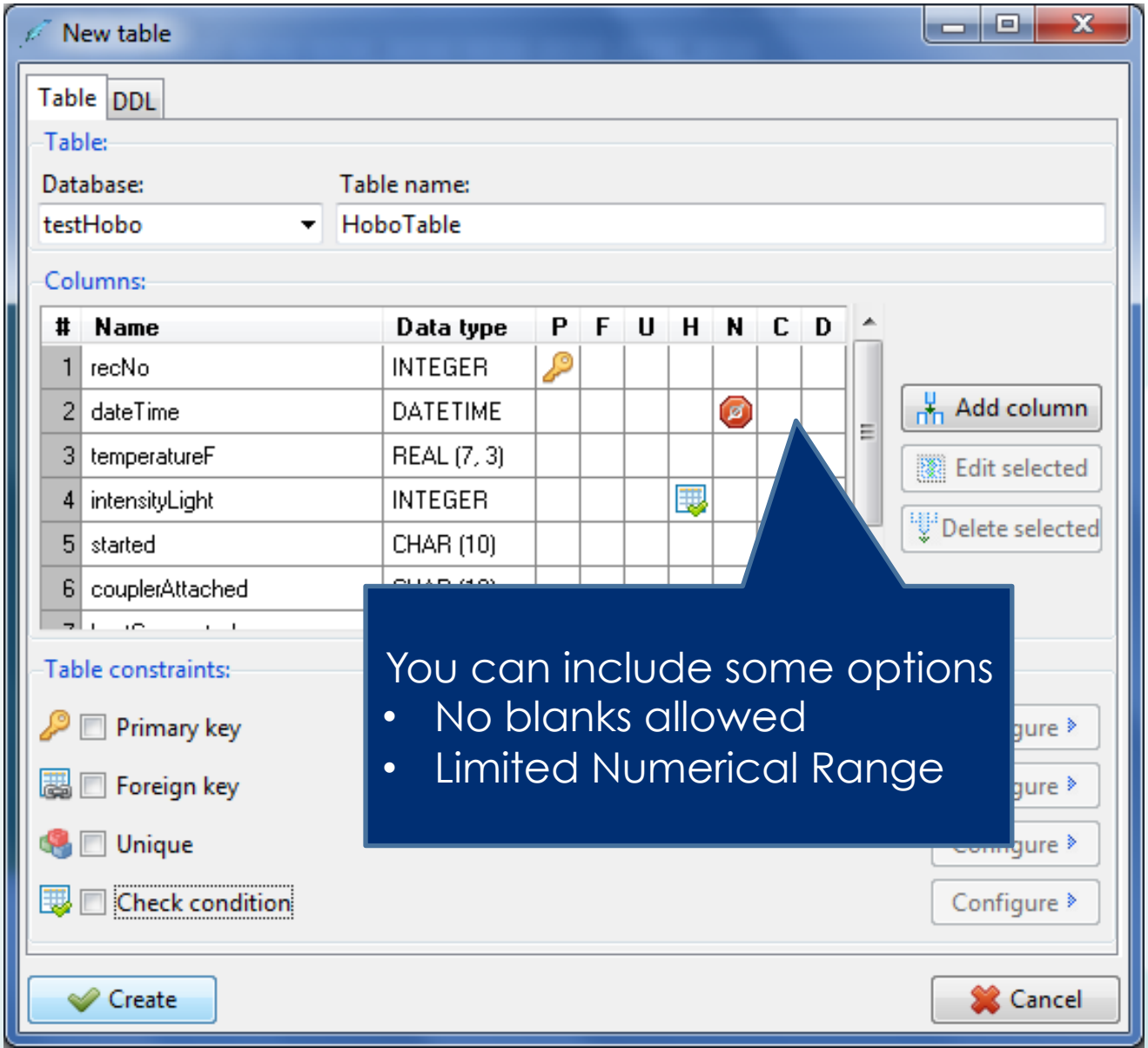
Add a Table


SQLiteStudio (v2.0.26)
Databases
View
Tools
Help
test (SQLite 3)
testHobo (SQLite 3)
Tables
Indexes
Triggers
Views
Add database
New table
New index
New trigger
New view
Refresh databases tree
Sort databases alphabetically

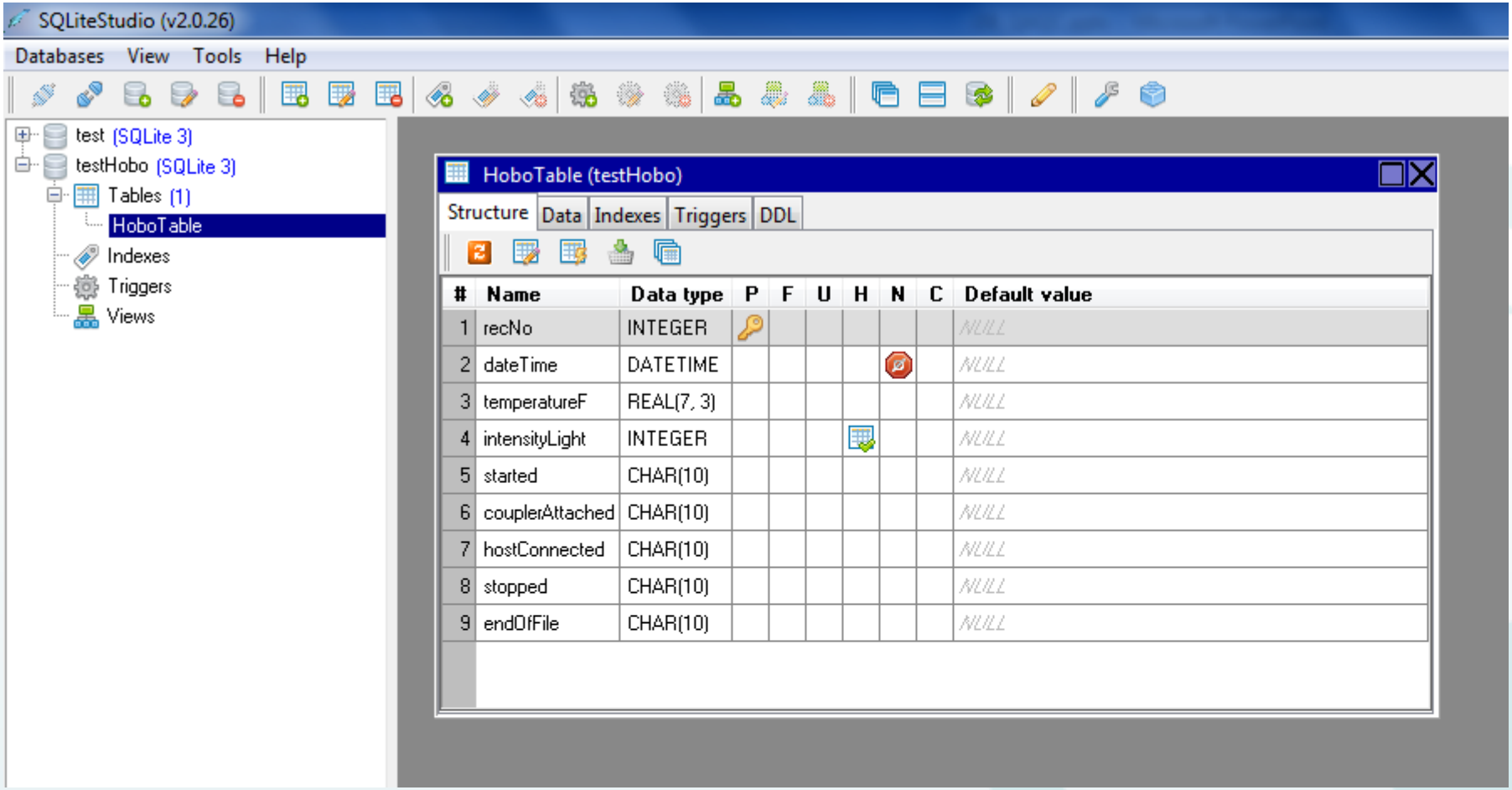
Populate the table fields



Ensure Integrity



Empty Table Ready to Fill



SQL

Structured **Q**uery **L**anguage

Special-purpose programming language

Scope includes data insert, query, update and delete, schema creation and modification

Basic SQL

SELECT [attribute list] (columns)

FROM [relation]

WHERE [condition]

JOIN

SELECT

```
SELECT SpeciesName, CommonName FROM Species

SELECT SpeciesName, CommonName FROM Species
   ORDER BY SpeciesName

SELECT COUNT(SpeciesName)
   FROM Species

SELECT DISTINCT(SpeciesName)
   FROM Species

SELECT DISTINCT(CommonName)
   FROM Species
```

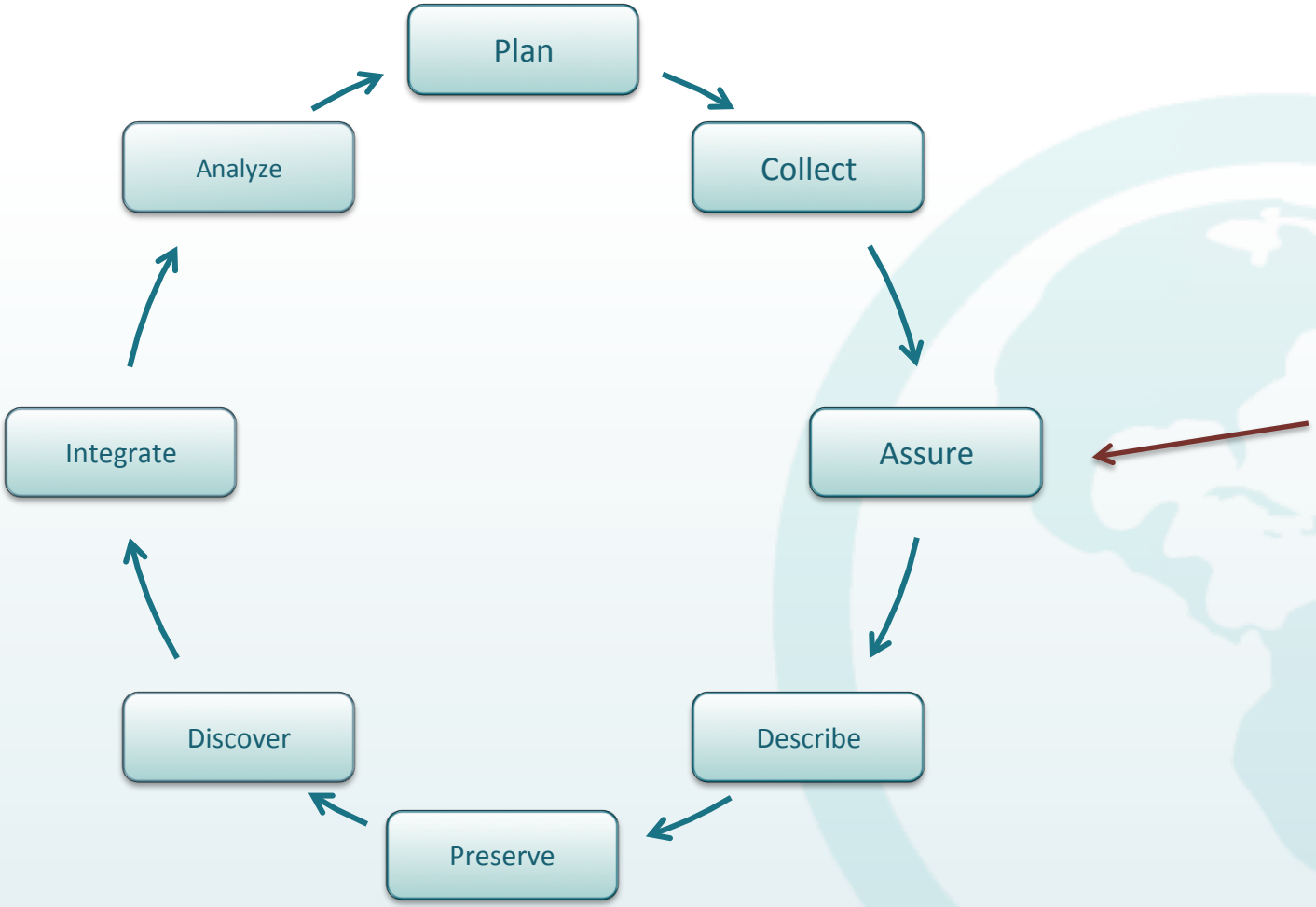
SELECT (cont.)

```
SELECT COUNT(SpeciesName)
   FROM Species

SELECT DISTINCT(SpeciesName)
   FROM Species

SELECT DISTINCT(CommonName)
   FROM Species
```

JOIN

```
SELECT Species.SpeciesName, Location.PlotName
    FROM Species
    JOIN Location
    ON Species.Location = Location.Pkey
```

Databases and the Data Life Cycle

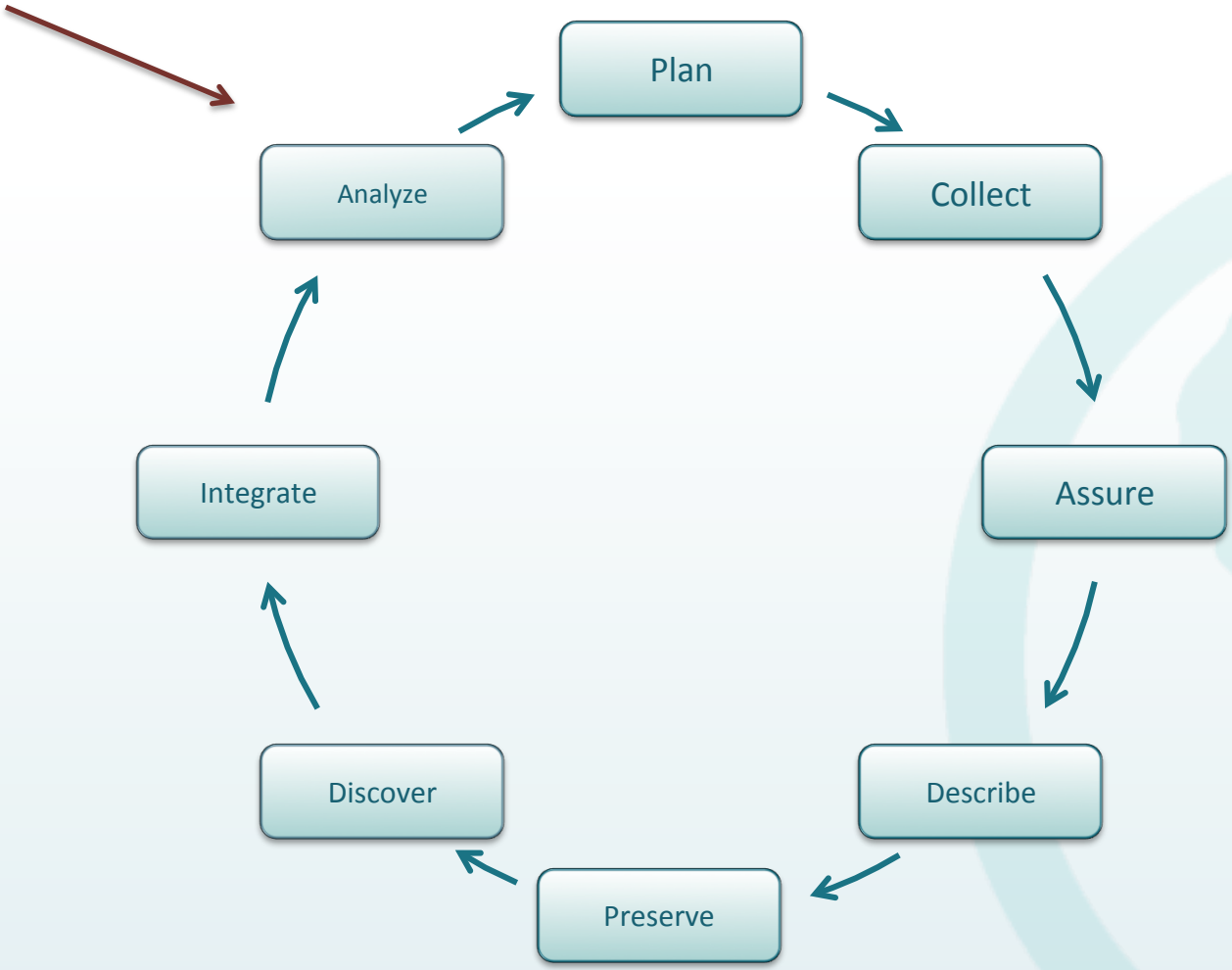


QA/QC Data

Examples:

- Check the range of measurements to see if any are out of the expected range
- Changes in adjacent measurements that are greater than expected
- Check for duplicate records
- Check the date and time range of the data

Databases and the Data Life Cycle

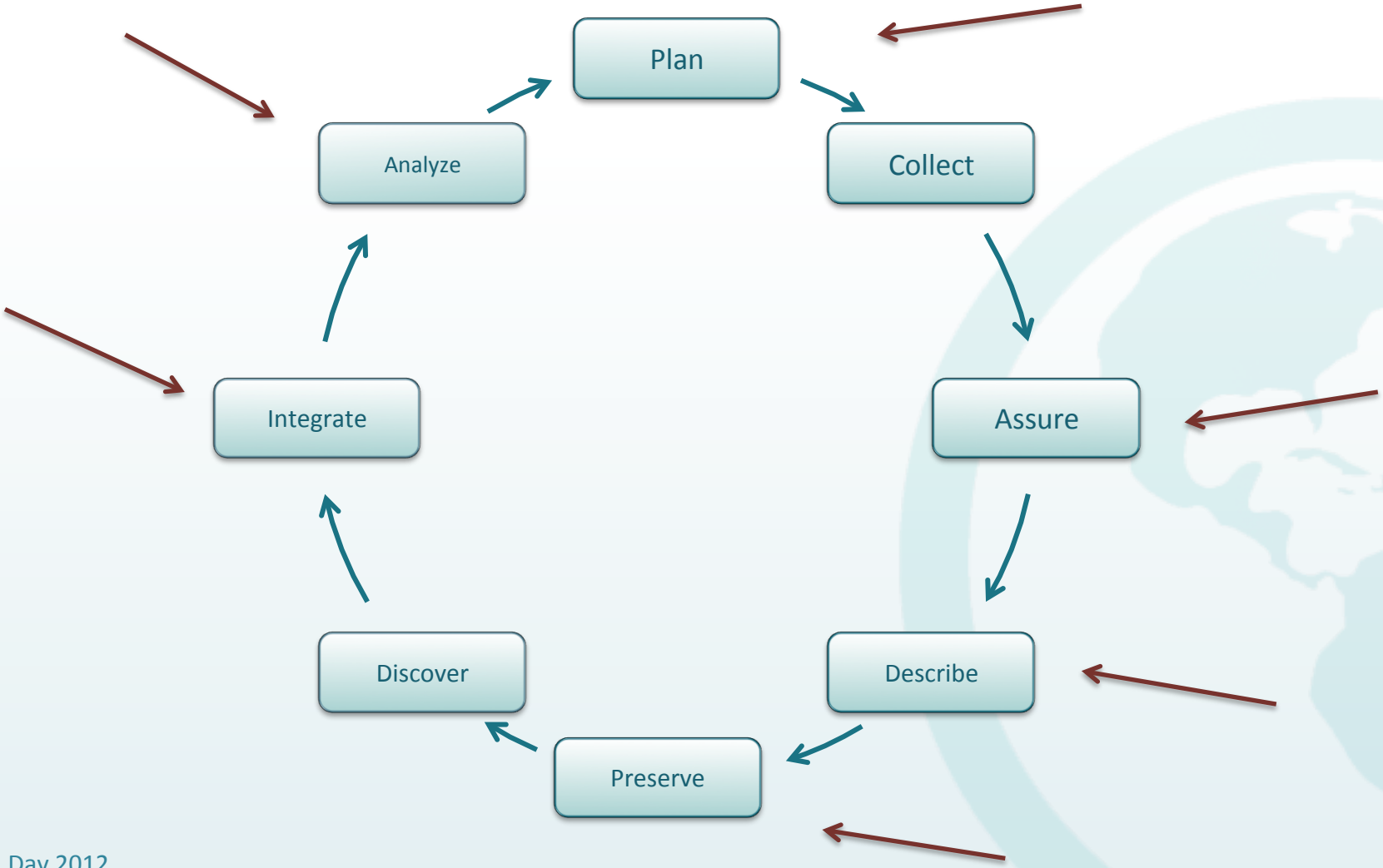


Other SQL Functions

Functions for descriptive statistics:

- COUNT()
- MAX(), MIN(), AVE()
- DISTINCT
- ORDER BY
- GROUP BY

Databases and the Data Life Cycle



DataONE

UNM UNIVERSITY LIBRARIES

Made possible by generous support from Dr. Walter E. Dean

Walter E. Dean Environmental Information Management Institute

June 3 through June 21, 2013
University of New Mexico