Data 301 Data Analytics
Relational Databases

Dr. Irene Vrbik

University of British Columbia Okanagan
irene.vrbik@ubc.ca

Why Relational Databases?

Relational databases allow for the storage and analysis of large amounts of data.

Relational databases are the most common form of database used by companies and organizations for data management.

Since a significant amount of data is stored in relational databases, understanding how to create and query these databases using the SQL standard is a very valuable skill.

What is a database?

A **database** is a collection of logically related data for a particular domain.

To manage (query, monitor, etc.) your database you need a tool...

A **database management system (DBMS)** is software designed for the creation and management of databases.

- e.g. Oracle, DB2, Microsoft Access, MySQL, SQL Server, MongoDB, LibreOffice Base, ...

Bottom line: A **database** is the *data* stored and a **database system** is the *software* that manages the data.

Databases in the Real World

Databases are everywhere in the real-world even though you do not often interact with them directly. $40 billion dollar annual industry

Examples:

- Retailers manage their products and sales using a database. e.g. Wal-Mart has one of the largest databases in the world!
- Online web sites such as Amazon, eBay, and Expedia track orders, shipments, and customers using databases.
- The university maintains all your registration information and marks in a database that is accessible over the Internet.

Apart from these large enterprise-grade data stores, we could also create and store small database on our person computers.

Database System Properties

A database system provides efficient, convenient, and safe multi-user storage and access to massive amounts of persistent data.

Efficient: Able to handle large data sets and complex queries without searching all files and data items.

Convenient: Easy to write queries to retrieve data.

Safe: Protects data from system failures and hackers.

Massive: Database sizes in gigabytes, terabytes and petabytes.

Persistent: Data exists even if have a power failure.

Multi-user: More than one user can access and update data at the same time while preserving consistency.

The Relational Model: Terminology

The **relational model** organizes data into tables called relations.

Developed by E. F. Codd in 1970 and used by most database systems.

Terminology:

- A **relation** is a table with columns and rows.
- An **attribute** is a named column of a relation.
- A **tuple** is a row of a relation (a single record in a table).
- A **domain** is a set of allowable values for one or more attributes.
- The **degree** of a relation is the number of attributes it contains.
- The **cardinality** of a relation is the number of tuples it contains.

Relation Example

The name of this relation is Emp.

Side note: Sometimes, the term field and attribute are used interchangeably, and for most purposes, they are the same thing. However, field describes a particular cell in a table found on any row, while attribute describes an entity characteristic in a design sense. [Source]. Eg. `A. Lee` is a *field* while `ename` is the *attribute*.

Relation Practice Example

WorksOn

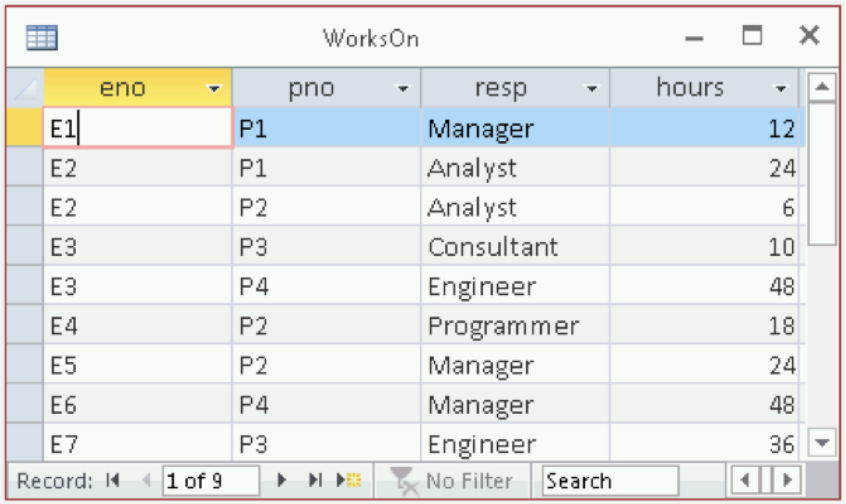
eno	pno	resp	hours
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36

Record: 1 of 9 No Filter Search

Example 5.1

1. What is the name of the relation?
2. What is the cardinality of the relation?
3. What is the degree of the relation?
4. What is the domain of resp? What is the domain of hours?

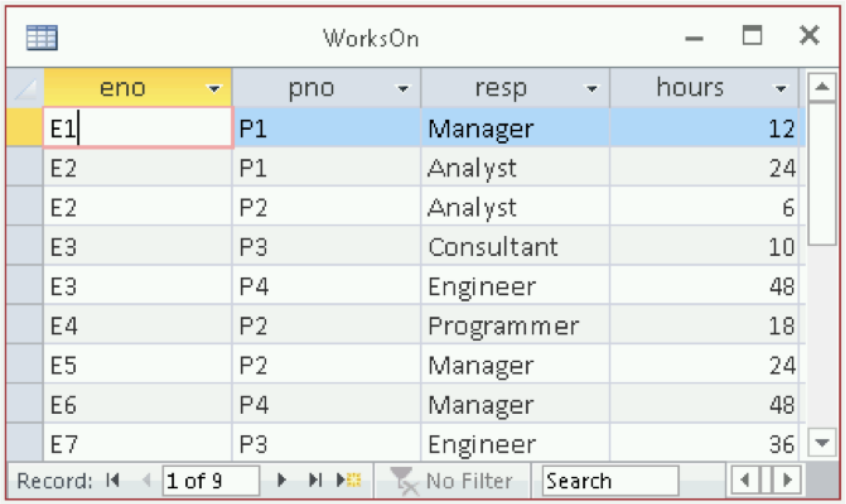
Relation Practice Example



Answer

1. What is the name of the relation? *WorksOn*
2. What is the cardinality of the relation?
3. What is the degree of the relation?
4. What is the domain of `resp`?
5. What is the domain of hours?

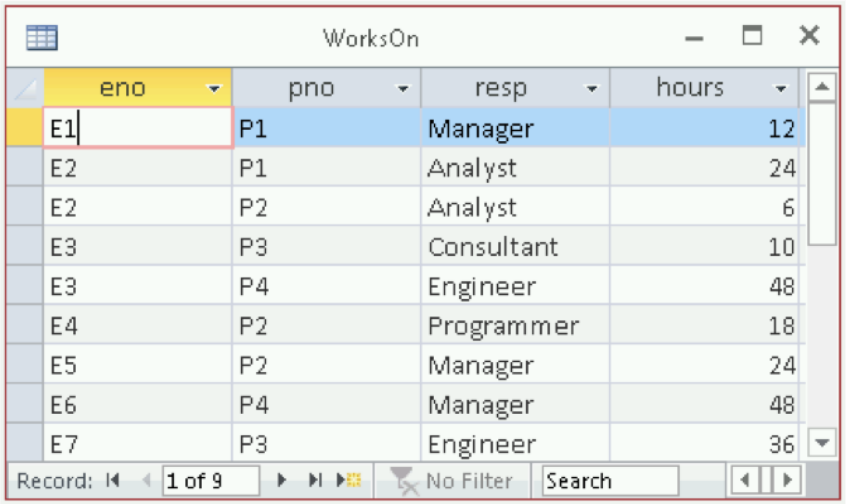
Relation Practice Example



Answer

1. What is the name of the relation? *WorksOn*
2. What is the cardinality of the relation? *9*
3. What is the degree of the relation?
4. What is the domain of `resp`?
5. What is the domain of hours?

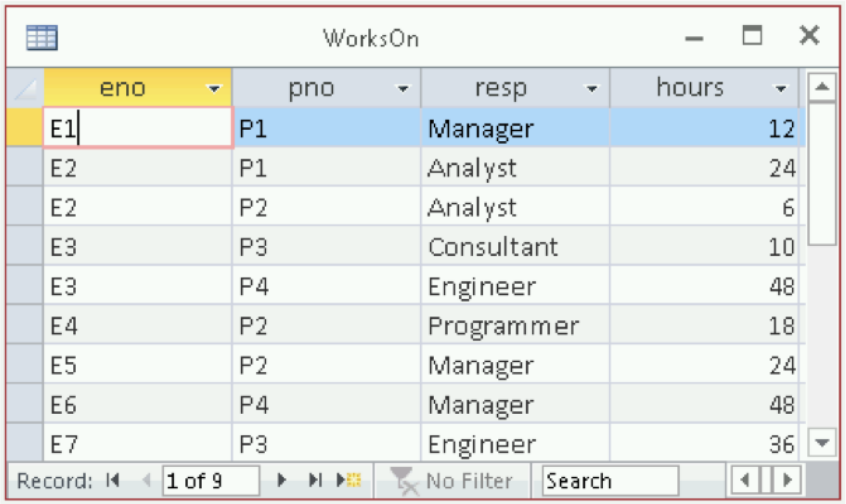
Relation Practice Example



Answer

1. What is the name of the relation? *WorksOn*
2. What is the cardinality of the relation? *9*
3. What is the degree of the relation? *4*
4. What is the domain of `resp`?
5. What is the domain of hours?

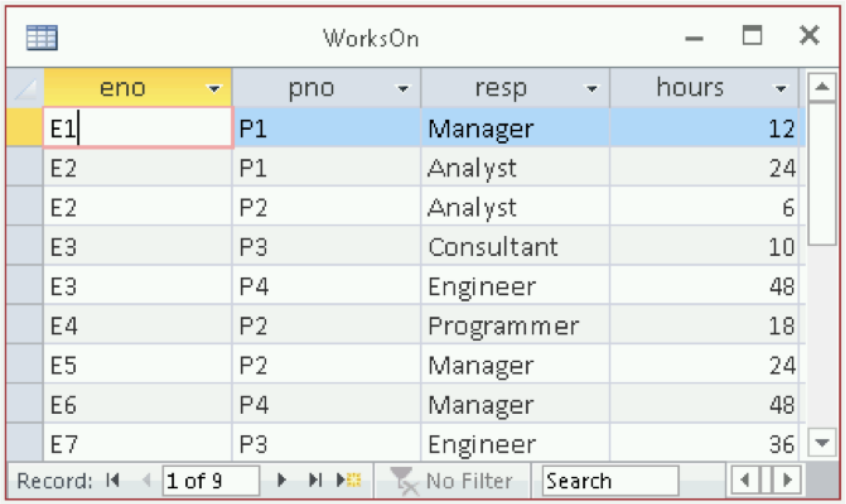
Relation Practice Example



Answer

1. What is the name of the relation? *WorksOn*
2. What is the cardinality of the relation? *9*
3. What is the degree of the relation? *4*
4. What is the domain of `resp`? *string*
5. What is the domain of hours?

Relation Practice Example



Answer

1. What is the name of the relation? *WorksOn*
2. What is the cardinality of the relation? *9*
3. What is the degree of the relation? *4*
4. What is the domain of `resp`? *string*
5. What is the domain of hours? *integer*

Database Definition Question

Question:

How many of the following statements are TRUE?

1. A database refers to the stored data.
2. A database system refers to the software.
3. A database system will lose the data stored when the power is turned off.

A) 0 B) 1 C) 2 D) 3

Database Definition Question

Answer:

How many of the following statements are TRUE?

1. A database refers to the stored data.
2. A database system refers to the software.
3. A database system will lose the data stored when the power is turned off.

A) 0 B) 1 C) 2 D) 3

Database Definition Question

Answer:

How many of the following statements are TRUE?

1. A database refers to the stored data.
2. A database system refers to the software.
3. A database system will lose the data stored when the power is turned off.

A) 0 B) 1 C) 2 D) 3

Database Definition Question

Answer:

How many of the following statements are TRUE?

1. A database refers to the stored data.
2. A database system refers to the software.
3. A database system will lose the data stored when the power is turned off.

A) 0 B) 1 C) 2 D) 3

Database Definition Question

Question:

How many of the following statements are TRUE?

1. Usually, more than one user can use a database system at a time.
2. The cardinality is the number of rows in a relation.
3. A relation's cardinality is always bigger than its degree.

A) 0 B) 1 C) 2 D) 3

Database Definition Question

Answer:

How many of the following statements are TRUE?

1. Usually, more than one user can use a database system at a time.
2. The cardinality is the number of rows in a relation.
3. A relation's cardinality is always bigger than its degree.

A) 0 B) 1 C) 2 D) 3

Database Definition Question

Answer:

How many of the following statements are TRUE?

1. Usually, more than one user can use a database system at a time.
2. The cardinality is the number of rows in a relation.
3. A relation's cardinality is always bigger than its degree.

A) 0 B) 1 C) 2 D) 3

Database Definition Question

Answer:

How many of the following statements are TRUE?

1. Usually, more than one user can use a database system at a time.
2. The cardinality is the number of rows in a relation.
3. A relation's cardinality is always bigger than its degree.

A) 0 B) 1 C) *2* D) 3

Database Definition Matching Question

Question:

Given the three definitions, select the ordering that contains their related definitions to the following three terms: relation, tuple, attribute

A) column, row, table

B) row, column, table

C) table, row, column

D) table, column, row

Database Definition Matching Question

Answer:

Given the three definitions, Select the ordering that contains their related definitions to the following three terms: relation, tuple, attribute

A) column, row, table

B) row, column, table

C) *table, row, column*

D) table, column, row

Cardinality and Degree Question

Question:

A database table has 5 rows and 10 columns. Select one true statement.

A) The table's degree is 50.

B) The table's cardinality is 5.

C) The table's degree is 5.

D) The table's cardinality is 10.

Cardinality and Degree Question

Question:

A database table has 5 rows and 10 columns. Select one true statement.

A) The table's degree is 50.

B) *The table's cardinality is 5.*

C) The table's degree is 5.

D) The table's cardinality is 10.

Creating and Using Databases

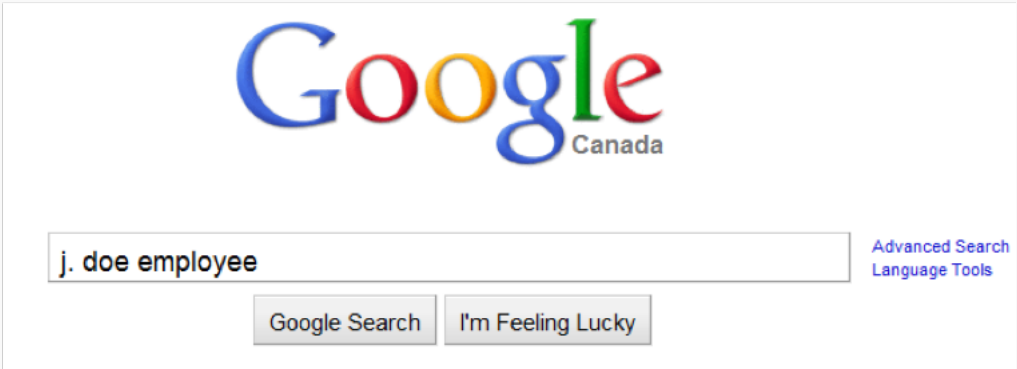
Typically, a data analyst will use an existing database. The database will already be created on a database system and contain data that was inserted and updated previously.

To use an existing database, the data analyst must be able to use the tools and languages to query the database. The standard is SQL.

Creating a large database is outside of the scope of this class, but we will learn how to create individual tables and load data into them which is a common data analysis task.

A Simple Query Language: Keyword Seaching

A query on Google, for example, allows a user to type keywords or phrases and returns a best answer estimate.



This works fairly well for web searches, although we lack precision. Precision is required for many applications.

- Example: How would you return all employees with salary greater than 30,000 using keyword search?

SQL Overview

Structured **Q**uery **L**anguage or **SQL** is the standard database query language to retrieve *exact answers*.

- Using simple, declarative *statements* (i.e. a valid command recognized by the database), SQL queries specifies *what to* retrieve while preserving the accuracy, security and integrity of the database.
- SQL is used by Microsoft Access, LibreOffice Base, and almost all other database systems.
- A knowledge of SQL commands will give you the power to access and explore data stored or relational databases.

SQL Overview

- There is a set of *reserved/key words* that cannot be used as names for database fields and tables. e.g. `SELECT`, `FROM`, `WHERE`, etc.
- SQL is generally *case-insensitive*.
  - `SELECT` will be treated the same as `Select` and `select`[1]
  - Some setups are case-sensitive for table and column, for example, 'ENAME' not the same as 'ename'.
  - Linux MySQL: usually defaults to case-sensitive for table and column names
  - Windows: usually defaults to case-insensitive for table and column names
  - Note that LibreOffice Base converts unquoted fields and table names to upper case; more on this here.
- SQL is *free-format* and white-space is ignored.
- Statements always end in a semicolon ;

[1] standard convention usually will capitalize keywords

SQL Create Table

The `CREATE TABLE` **command/clause** is used to create a table in the database.

- A **clause/command** performs a specific tasks in SQL. While clause will work with lowercase letters, the convention is to write command in capital letters.

A table consists of a table name (eg. `emp`) and a set of fields/attributes with their names and data types (i.e. domain).

Example:

```
CREATE TABLE emp (
    eno       CHAR(5),
    ename     VARCHAR(30) NOT NULL,
    bdate     DATE,
    title     CHAR(2),
    salary    DECIMAL(9,2),
    supereno  CHAR(5),
    dno       CHAR(5),
    PRIMARY KEY (eno)
)
```

field must always have a value

Data Types:
CHAR(5) – always 5 chars long
VARCHAR(30) – up to 30 chars long
DECIMAL(9,2) – e.g. 1234567.99
DATE – e.g. 1998/01/18

SQL Create Table - Constraints

- The `NOT NULL` is a *constraint* which indicates that we can not enter a tuple into this relation without the field `ename`
- To put another way, all employees in the employee table must be entered with an employee name (makes sense!)
- Other constraints include:

  `UNIQUE` forces all rows to have a different value.
  `DEFAULT` sets the field to a certain value if it is not specified,

  eg. `DEFAULT 0` or `DEFAULT 'unknown'`

What is a key?

A key is a set of attributes that *uniquely* identifies a tuple in a relation. Read more about this constrain here

- A *primary key* uniquely identifies a record in the table.
- A table can have only one primary key; however it can consist of single or multiple columns (fields).
- A *foreign key* is a field in a table that is primary key in another table.

Although keys are not required, they can help to identify a particular row (data item) and find it faster.

In the `emp` table, the key was `eno`. It was called the `primary key` because it was the main key used to find an employee in the table.

Question:

What is a key to identify a student in this class?

CREATE TABLE Syntax

The general syntax to create a new table in a database:

```
CREATE TABLE table_name (
    attribute1 datatype,
    attribute2 datatype,
    ...
);
```

We specify the names of the columns of our table within the parenthesis

- notice that they are separated by commas
- end of lines have a semicolons
- We can choose from a number of data types (e.g. varchar, integer, date, etc.).

CREATE TABLE Syntax in LibreOffice Base

In LibreOffice Base, unquoted fields and table names are converted to upper case. read more here.

To prevent this, you can use quoted attribute and tables names, eg.

```
CREATE TABLE "table_name" (
    "attribute1" datatype,
    "attribute2" datatype,
   ....
);
```

Try It: CREATE TABLE

Question:

Create a table called `mydata` that has three fields:

- `num` – that will store a number (use `int` as data type)
- `message` – that will store a string up to 50 characters (`varchar` data type)
- `amount` – that stores a decimal number with 8 total digits and 2 decimal digits (decimal data type)

Use the web site sqlfiddle.com or DB-fiddle to try your table creation.

SQL fiddle is an online SQL database (no need to download anything) where we can test, debug and share SQL snippets. Sharing is as easy as pasting the url. click me!

Relational Database Management Systems

In this lecture we will be introducing a couple relational database management systems.

- UBC qualify for a free Office 365 subscription, which includes free downloads of: Word, Excel, PowerPoint, and more.
- Students running Windows can also install **Microsoft Access**.
- Microsoft Access is a Database Management System (DBMS).
- Access can work directly with data from many SQL databases on the desktop, on servers, on minicomputers, or on mainframes, and with data stored on web servers.

TutorialsPoint is a great resource for help on MS Access.

Relational Database Management Systems

For students *not* running Windows, a nice alternative is *LibreOffice*

- *LibreOffice* is a free and open-source office suite, a project of The Document Foundation.
- **LibreOffice Base** is a free and open-source relational database management system that is part of the LibreOffice office suite; download here
- Like Access, it can be used to create and manage databases either locally or on servers eg. **mysql,** an **access database**.

As I am using a Mac, I will be conducting demonstrations in LibreOffice Base. The details for Microsoft Access are provided as snap shots throughout and will be very similar to LibreOffice Base.

- Here is a useful webpage for help on LibreOffice Base along with the handbook.

Relational Database Management Systems

- We will learn how to code SQL queries.
- Queries offer the ability to retrieve and filter data, calculate summaries (totals), and update/move/delete records in bulk.
- We also may want to take advantage of the graphical user interface (GUI) frontend for data manipulation and queries.
- These visual representations are easy to use and hides the complexity of writing SQL commands while still providing access to powerful and advanced analysis
- We'll see how we can easy switch between the graphical query design and SQL syntax.
- Bonus: this easy back and forth may be useful as you are trying to learn SQL.

CREATE TABLE in Microsoft Access

In Microsoft Access, start by creating a “Blank desktop database" (give it whatever name you want).

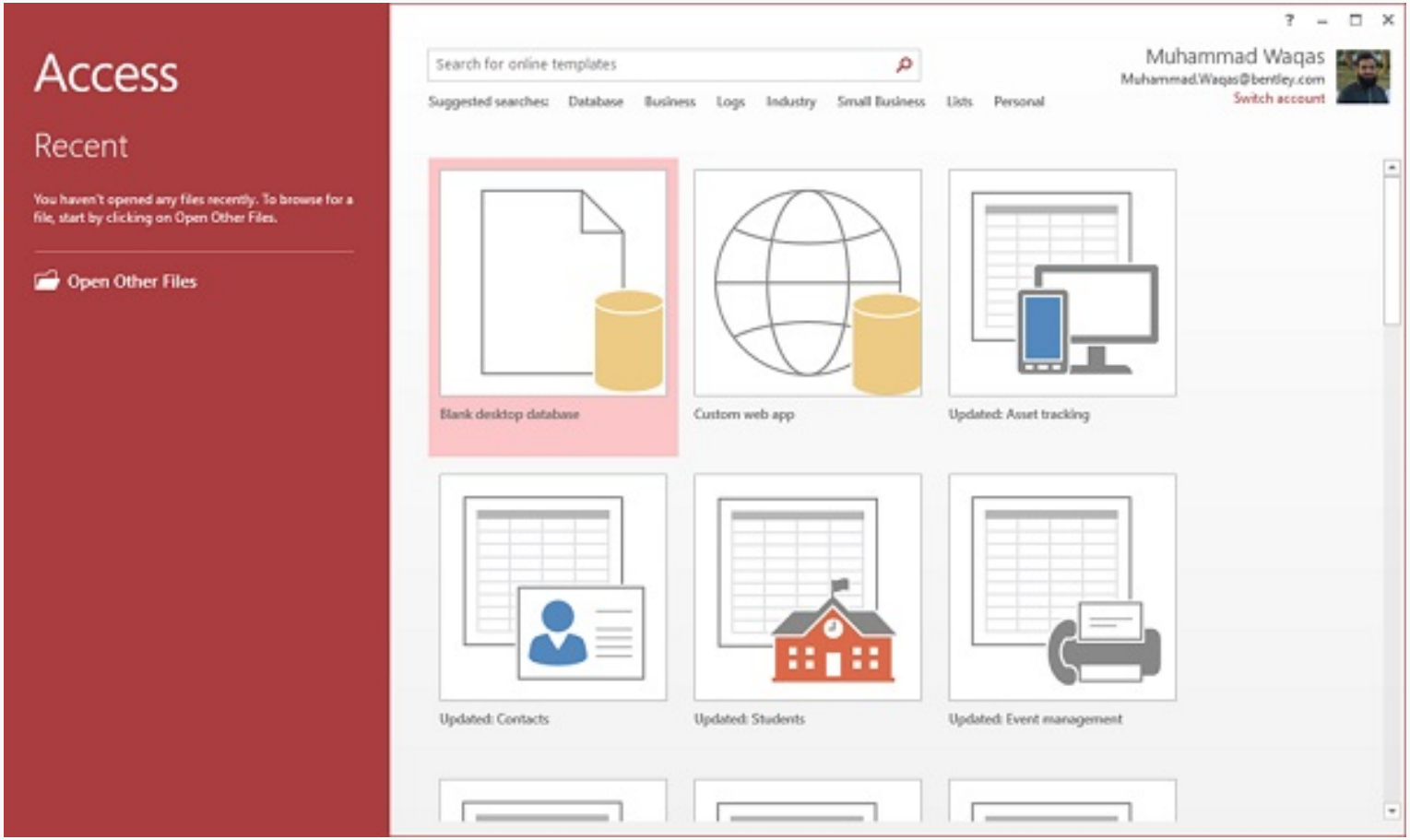
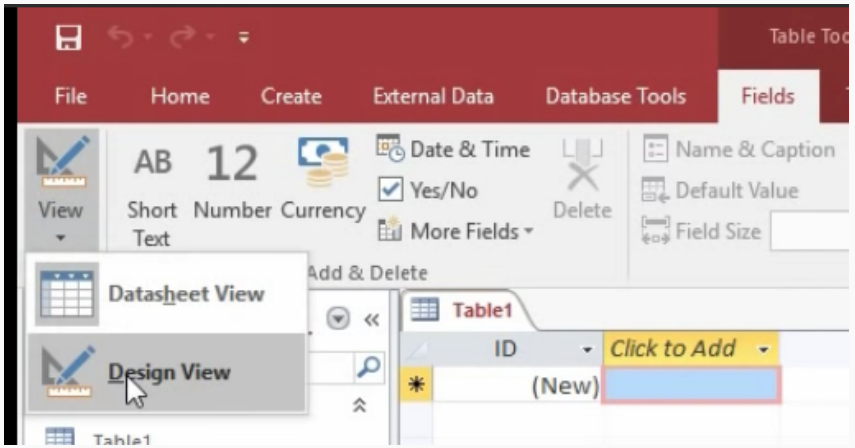


Image source

CREATE TABLE in Microsoft Access

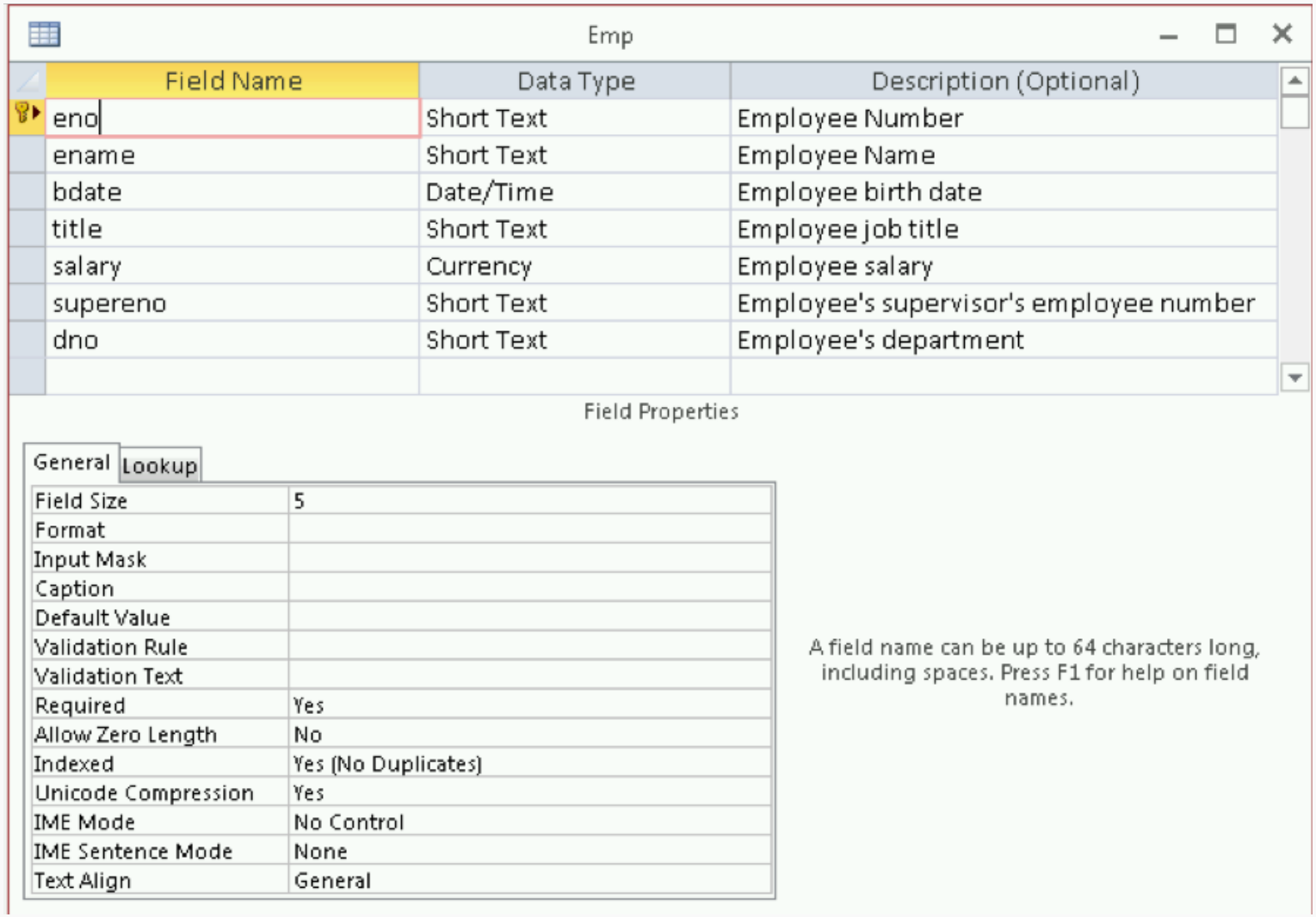
Next, navigate to the **Create** tab in the Ribbon and select the **Table** button 

to build a table. To get to the "Design View" as seen on the next page, go to **View** > **Design View** in the **Fields** tab:



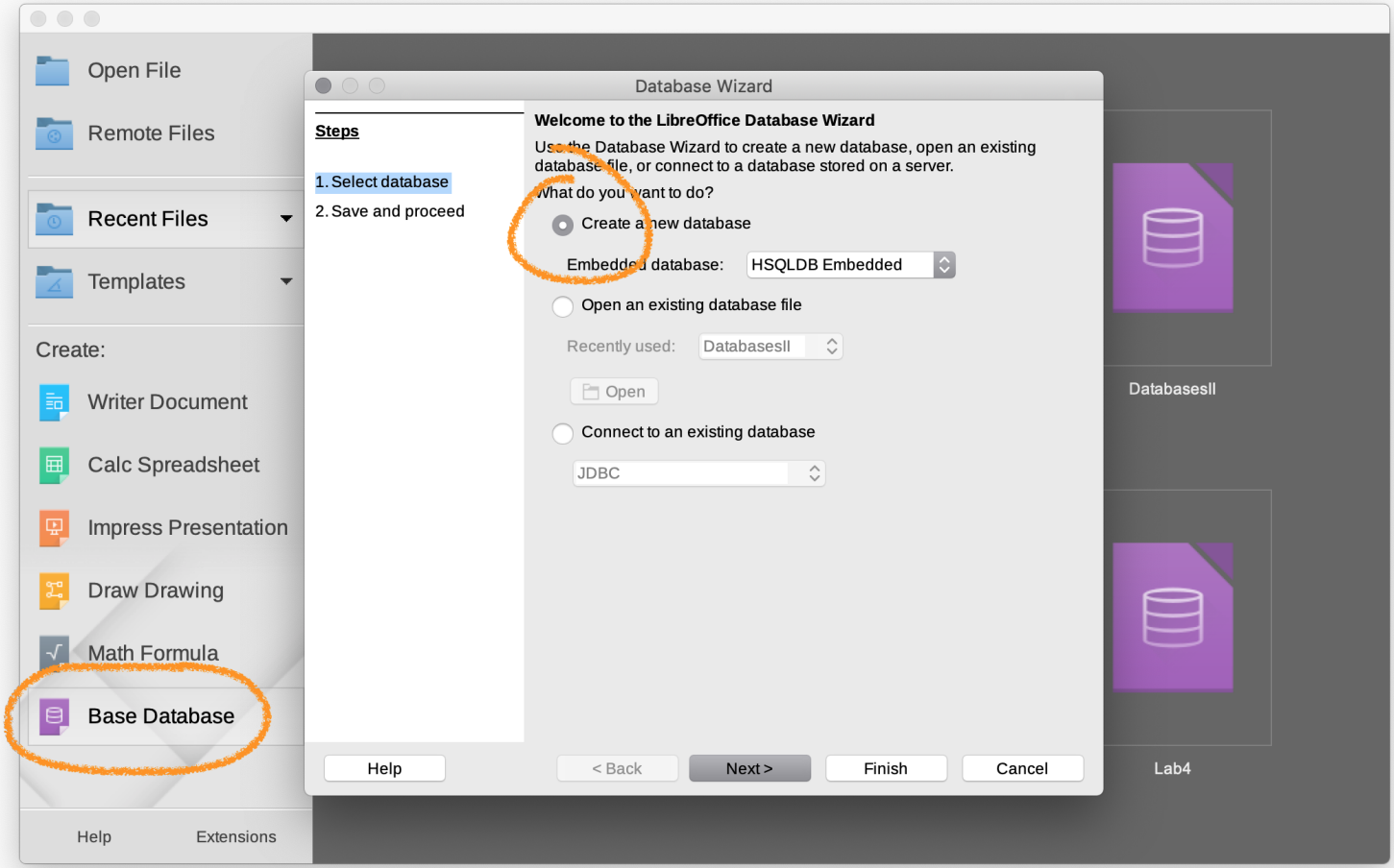
For a walk through of this, see this demo on YouTube.

CREATE TABLE in Microsoft Access



CREATE TABLE in LibreOffice Base

Upon opening Base, a database Wizard will pop-up. While we could access an existing database from a server, today we will be creating a new internal database:



CREATE TABLE in LibreOffice Base

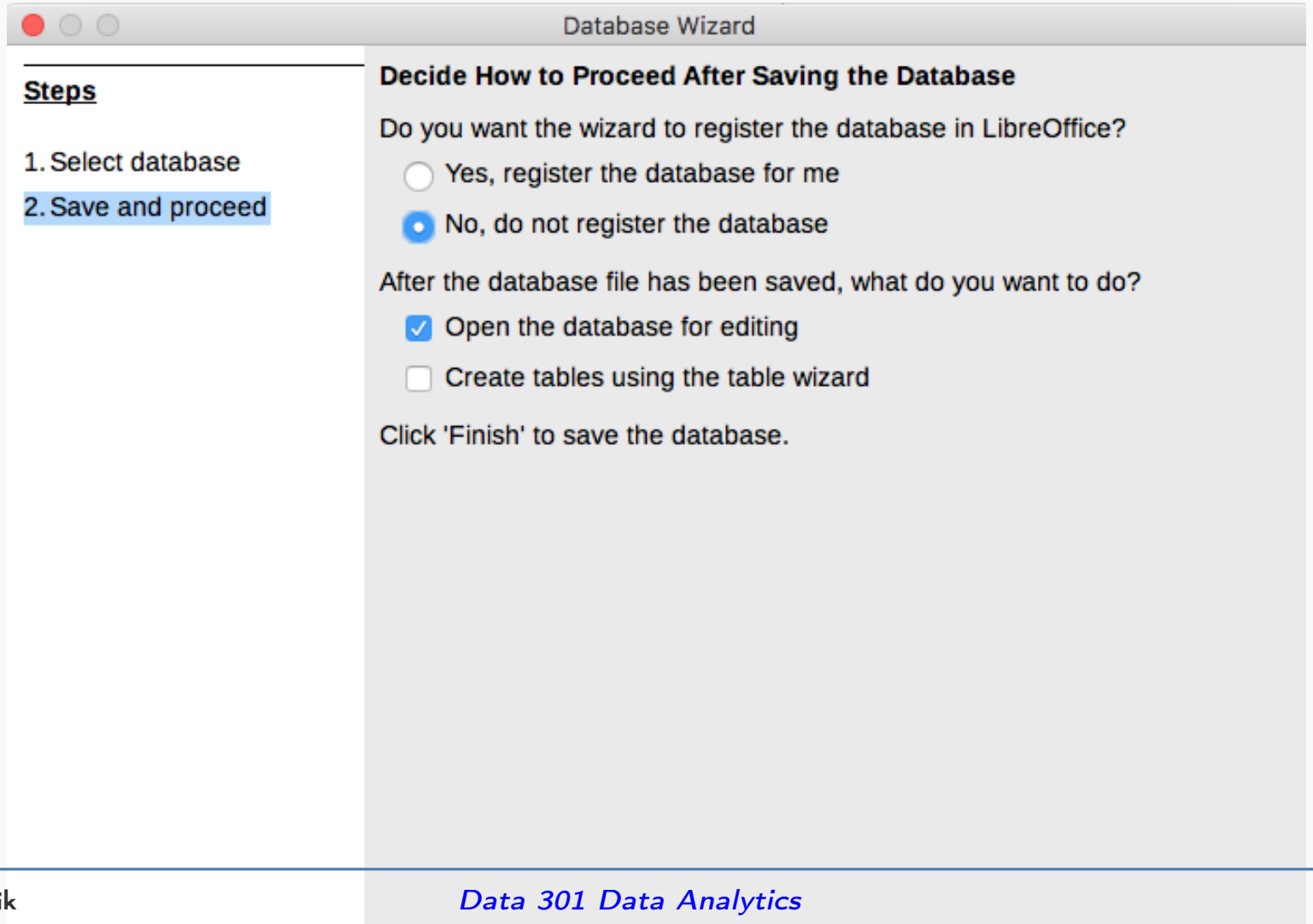
- If you don't see the Database Wizard, choose **File** > **New** > **Database**. (Shortcut Cmnd + N).
- To create a new database file.
  1. select the type of database (leave the HSQLDB environment default),
  2. you may or may not choose to register you database[2]. For now choose no (we can always register it later).
  3. click the **Finish** button.

N.B. The Table Wizard helps you to add a sample tables to the new database file upon which we can build; see *Using the Wizard to create a table*.

[2] registered databases be accessed by other program components as a data source (e.g. mail merge

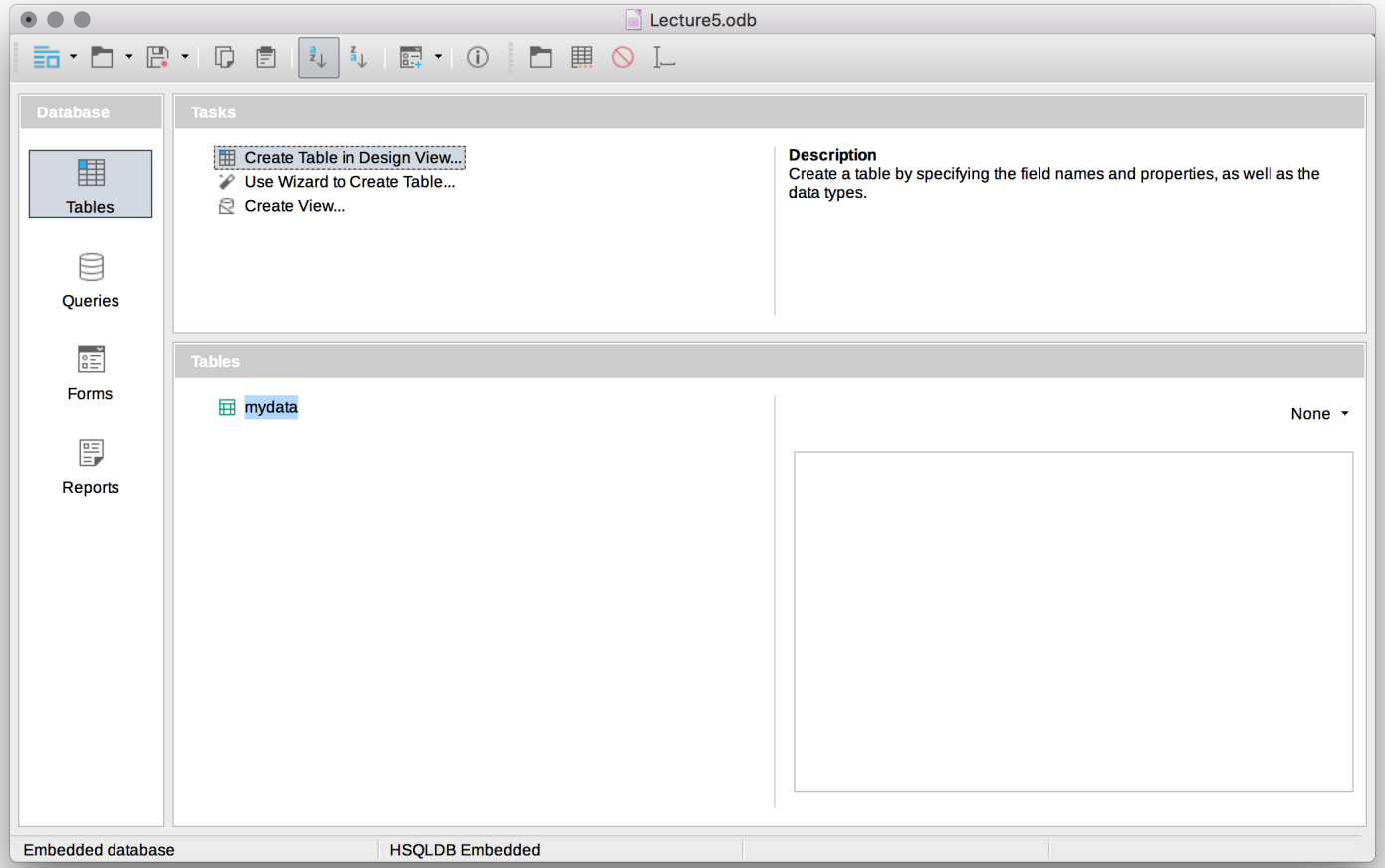
CREATE TABLE in LibreOffice Base

It doesn't make any functional difference to us if we register the database for what we will be doing. Either leave it at the default or change it to "No, do not register the database".



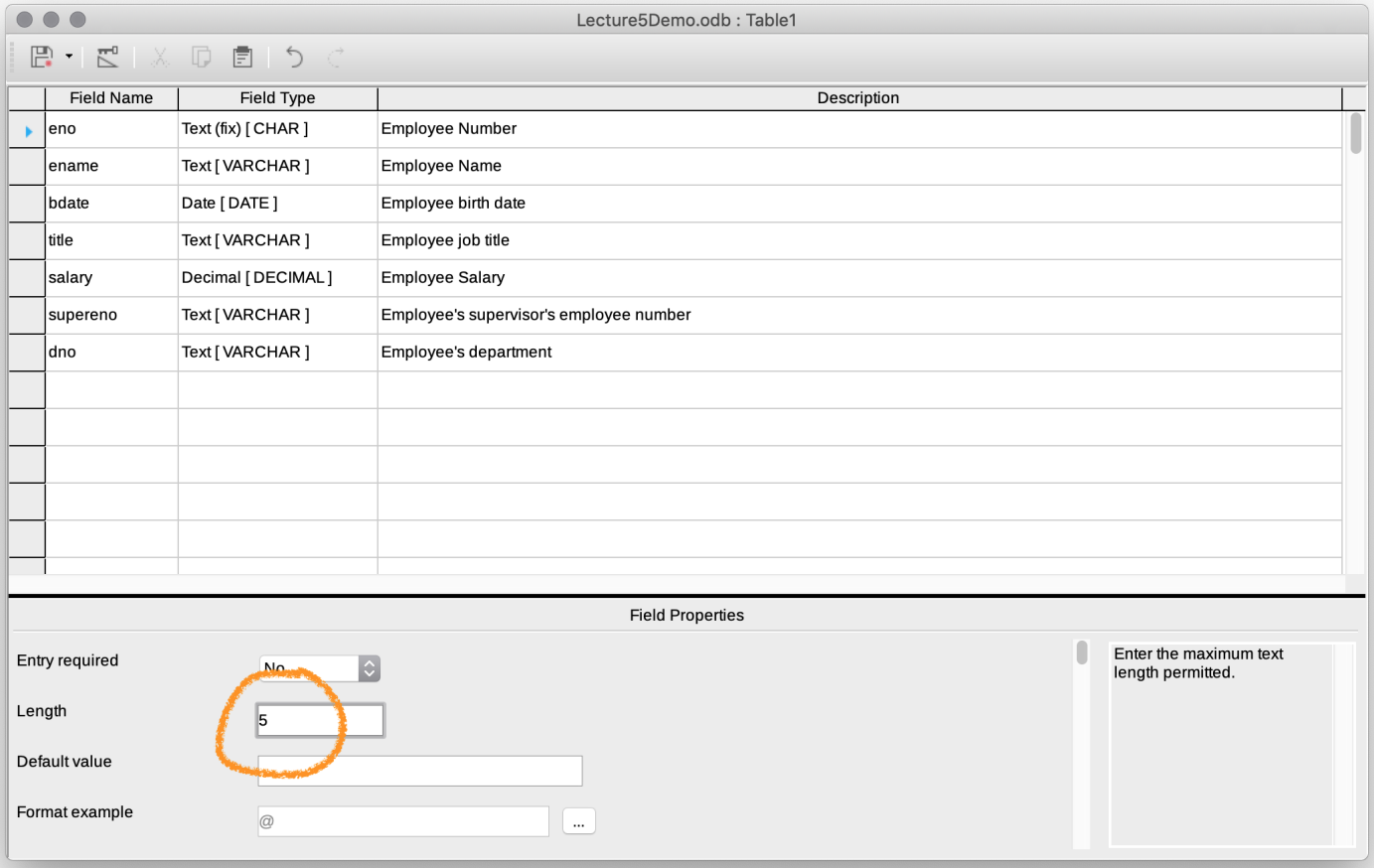
CREATE TABLE in LibreOffice Base

Right now we have an empty database. In LibreOffice Base use "Create a Table in Design View" to add a table.



CREATE TABLE in LibreOffice Base

We can enter our fields and specify the data types from the drop down menu. Any extra parameters can be entered in the boxes provided below.



- To make an attribute a primary key in LibreOffice Base, simply right click on the Field Name and select “Primary Key".
- In MS Access, select the field and press the primary key button

- N.B. Data type options are named slightly different in LibreOffice Base than MS Access).
- For example “Text" vs. “Short Text", LibreOffice base does not have a “Currency" option.
- To view the image from the previous slide, open WorksOn.odb in LibreOffice base, right click on the `emp` table and press Edit

Schemas and Metadata

Typically a database will be a compilation of many tables that are all connected in some way.

The description of the structure of the database is called a schema. It can refer to a visual representation of a database or a set of rules that govern a database.

The schema is a type of *metadata*.

Adding Data using INSERT

Insert a row using the INSERT command:

```
INSERT INTO emp VALUES ('E9','S. Smith',
'1975-03-05', 'SA',60000,'E8','D1');
```

Recall the fields of our table: eno, ename, bdate, title, salary, supereno, dno

If you do not give values for all fields in the order they are in the table, you must list the fields you are providing data for:

```
INSERT INTO emp(eno, ename, salary)
     VALUES ('E9','S. Smith',60000);
```

Note: If any columns are omitted from the list, they are set to NULL. Note that NULL is not the same thing as an empty string ''

Question

Using the `mydata` table insert three rows:

- (1, 'Hello', 99.45)
- (2, 'Goodbye', 55.99)
- (3, 'No Amount')

Some notes on quotes

Notice how numbers do not need to be surrounded in quotes. While we may use single or double quotes around text, single quotes is generally preferred.

Dates

The DATA data type appears in the format YYYY-MM-DD (in quotes) eg, `bdate = '1975-01-15'`.

Continuing from our sqlfiddle example.

SQL Fiddle | MySQL 5.6 | View Sample Fiddle | Clear | Text to DDL | Donate | About

```
1  CREATE TABLE mydata (
2    num int,
3    message varchar(50),
4    amount decimal(8,2)
5  );
6
7  INSERT into mydata VALUES (1, 'Hello', 99.45);
8  INSERT into mydata VALUES (2, 'Goodbye', 55.99);
9  INSERT into mydata(num, message) VALUES (3, 'No Amount');
10
11
```

Build Schema | Edit Fullscreen | Browser | [;]

```
1  SELECT * FROM mydata;
```

Run SQL | Edit Fullscreen | [;]

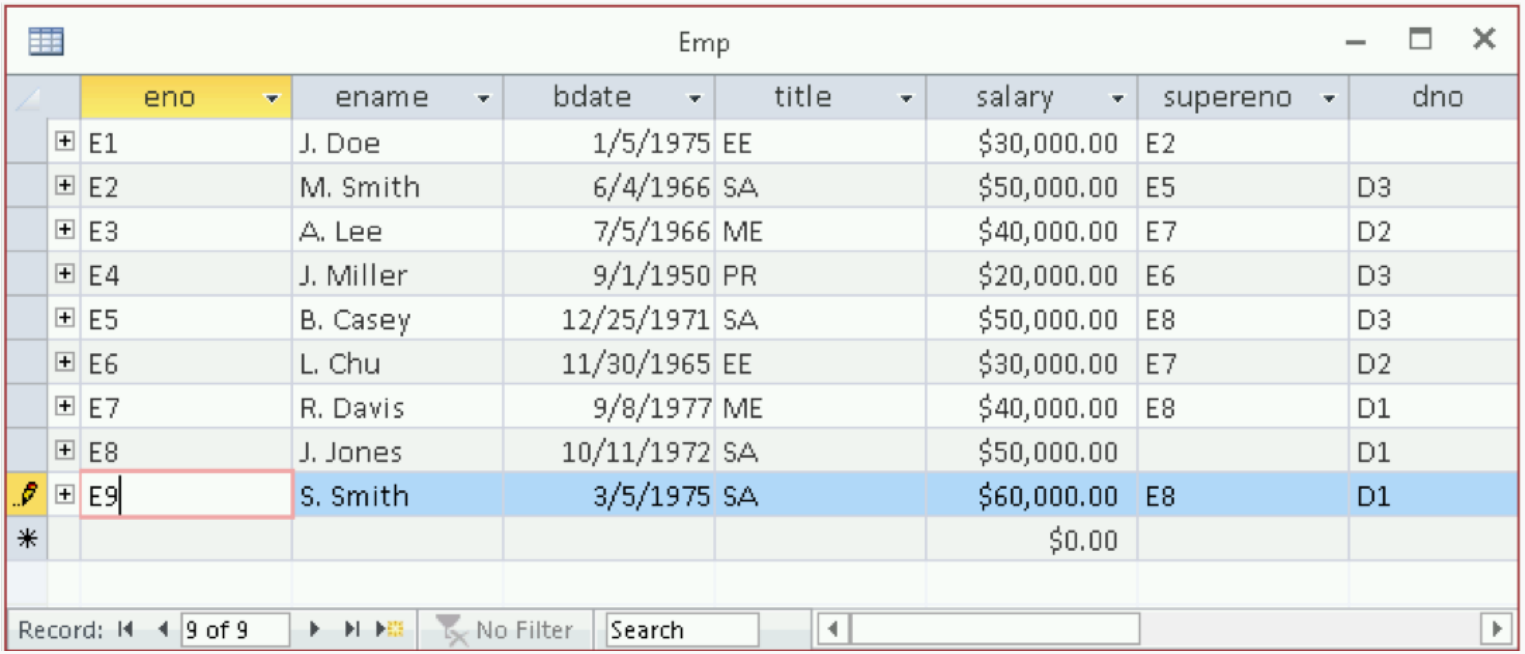
num	message	amount
1	Hello	99.45
2	Goodbye	55.99
3	No Amount	(null)

Record Count: 3; Execution Time: 3ms | View Execution Plan | link

- `SELECT * FROM mydata;` will return the entire `mydata` table (more on this later).
- The **Record Count** located on the green line– this indicates the number of rows returned in our query.

Adding Data in Microsoft Access

Consider the `WorksOn.accdb` database on Canvas. In Microsoft Access, insert a new row by entering data into the last row of the table when in data view.



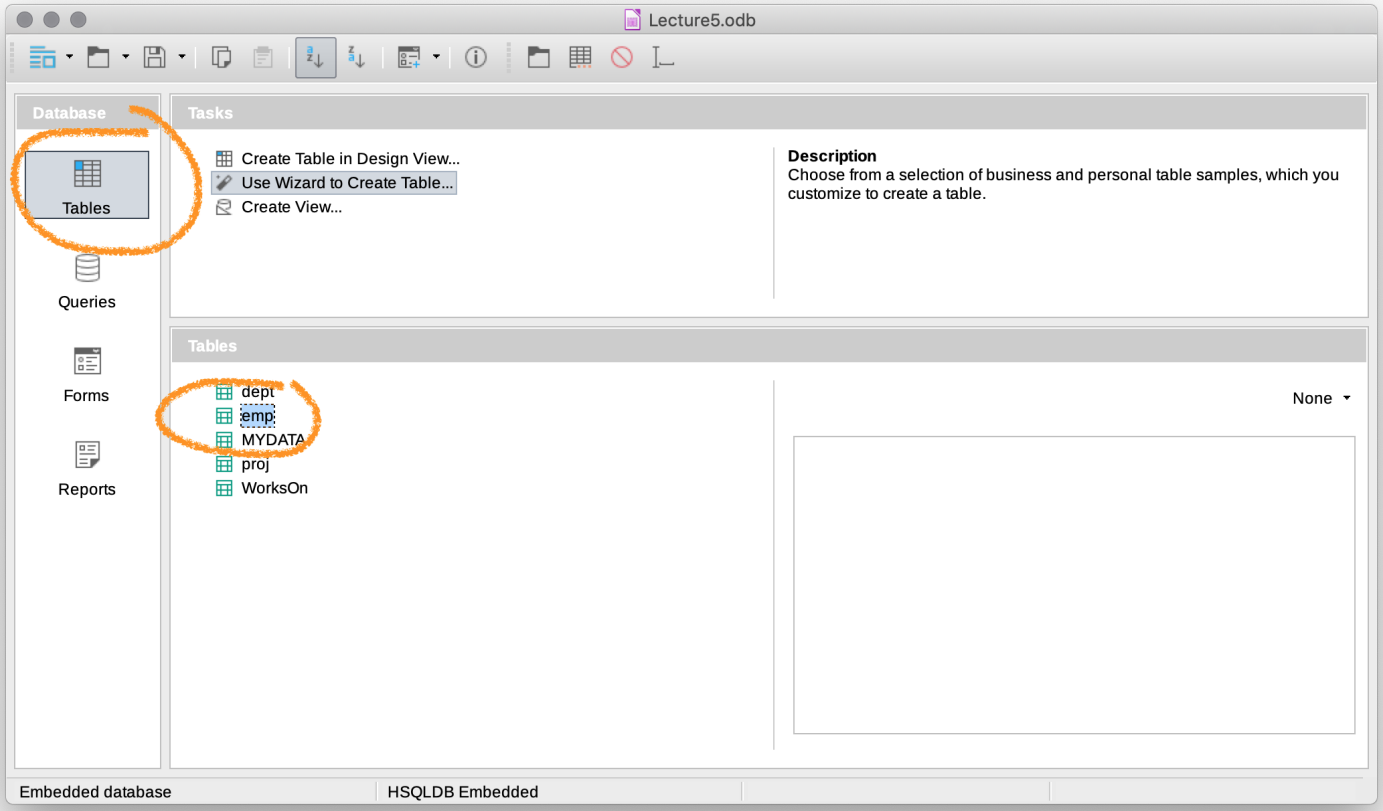
Emp

eno	ename	bdate	title	salary	supereno	dno
E1	J. Doe	1/5/1975	EE	$30,000.00	E2	
E2	M. Smith	6/4/1966	SA	$50,000.00	E5	D3
E3	A. Lee	7/5/1966	ME	$40,000.00	E7	D2
E4	J. Miller	9/1/1950	PR	$20,000.00	E6	D3
E5	B. Casey	12/25/1971	SA	$50,000.00	E8	D3
E6	L. Chu	11/30/1965	EE	$30,000.00	E7	D2
E7	R. Davis	9/8/1977	ME	$40,000.00	E8	D1
E8	J. Jones	10/11/1972	SA	$50,000.00		D1
E9	S. Smith	3/5/1975	SA	$60,000.00	E8	D1
				$0.00		

Record: 9 of 9 No Filter Search

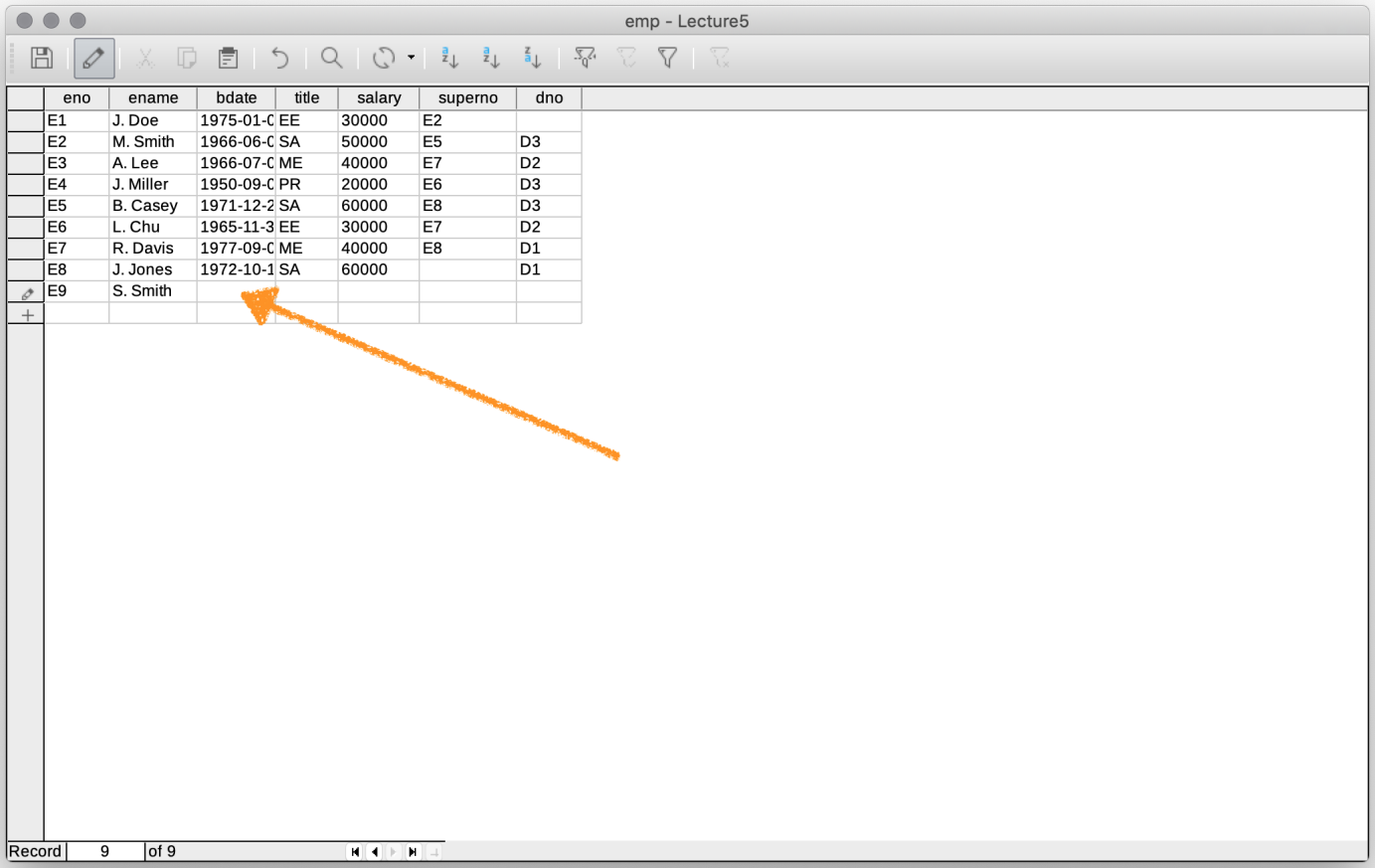
Adding Data in LibreOffice Base

Consider this `WorksOn.obd`, we can add a row to the `emp` table by simply double clicking it in **Tables** and inputting the data as we would in an Excel spreadsheet.



Adding Data in LibreOffice Base

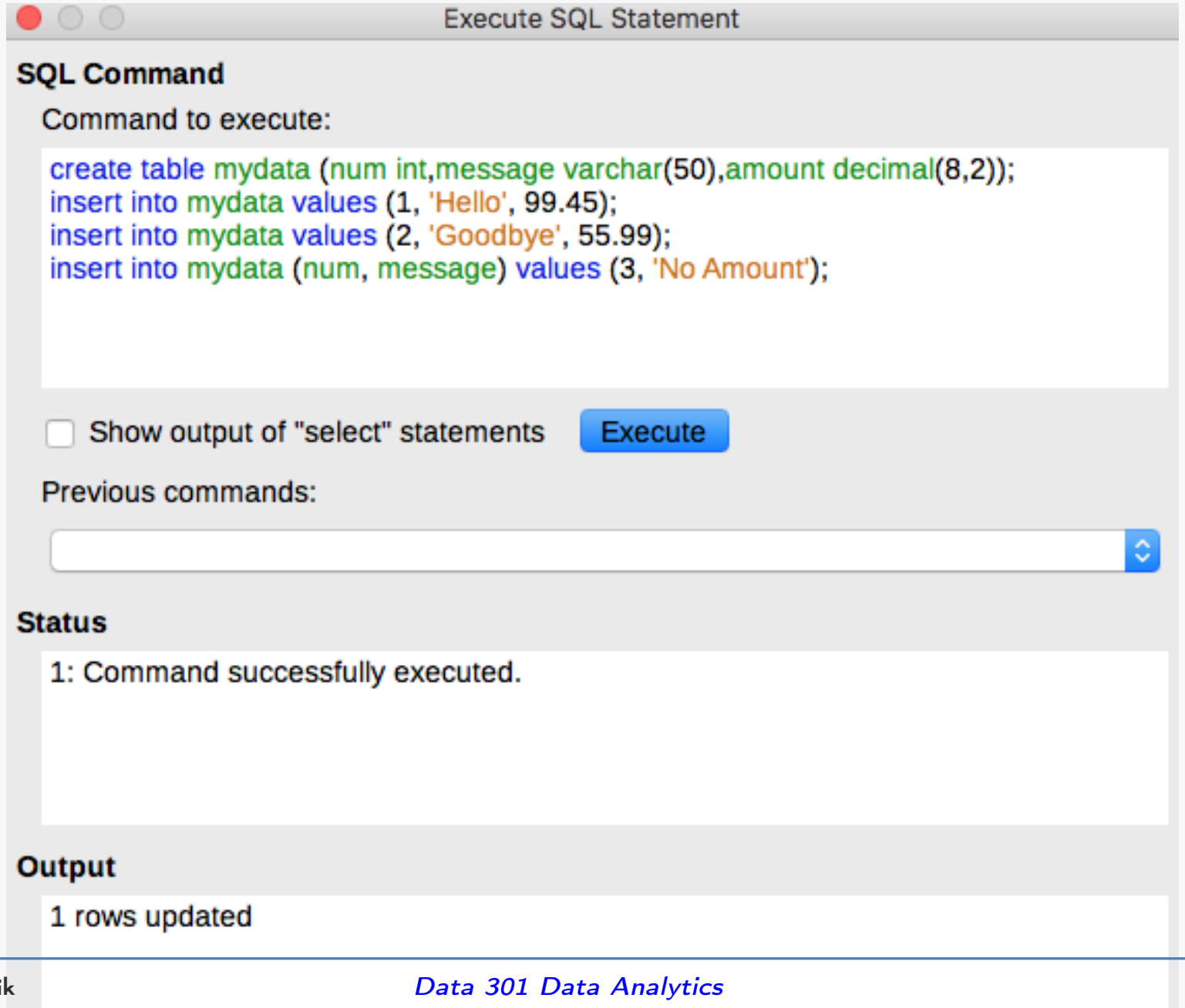
Consider this `WorksOn.obd`, we can add a row to the `emp` table by simply double clicking it in **Tables** and inputting the data as we would in an Excel spreadsheet. N.B. You have to set a primary key before adding records to a table using this feature



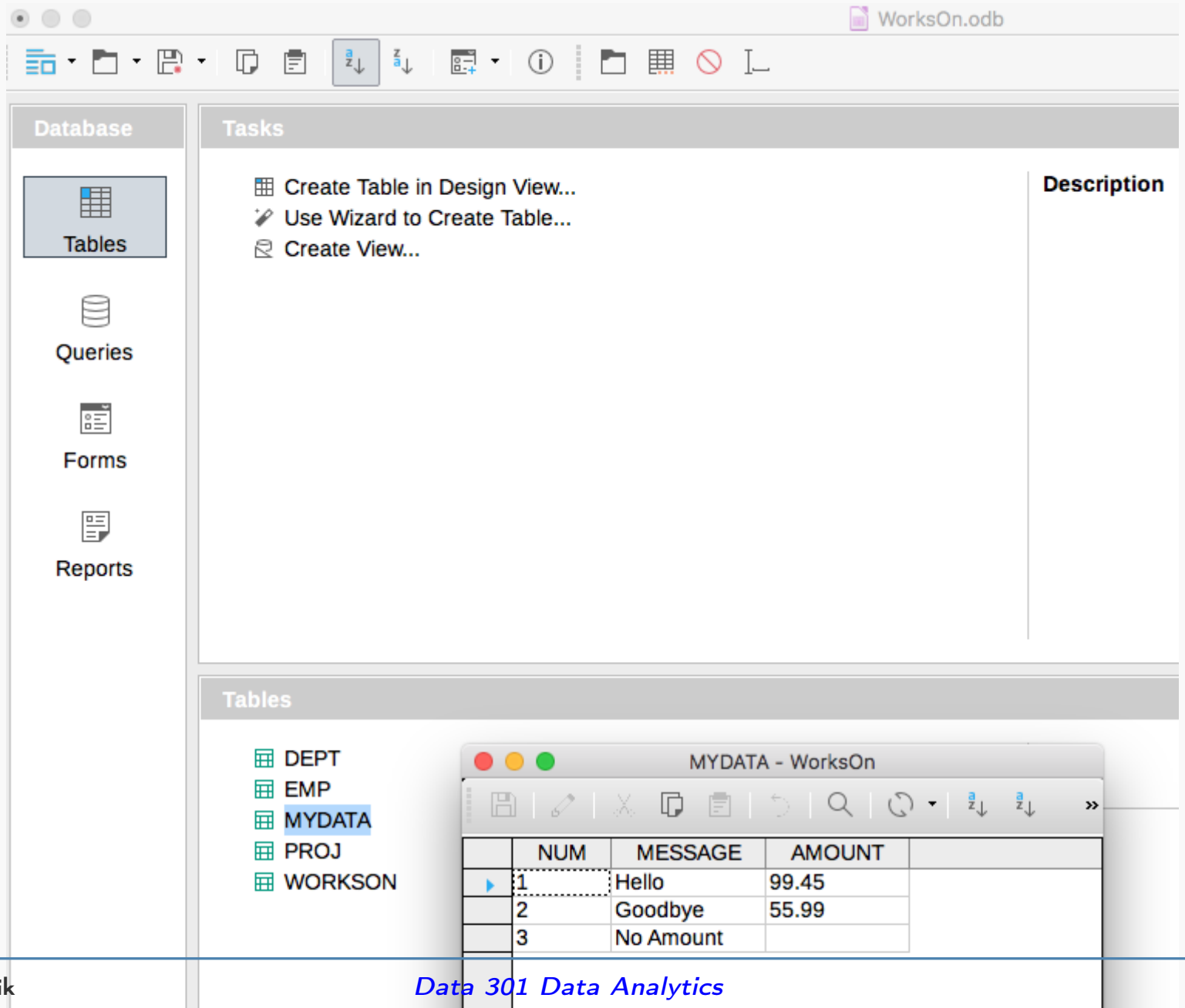
SQL mode

- Rather than using the Design View mode, we can use SQL commands by selection **Tools** > **SQL** in LibreOffice Base.
- If you are using Microsoft Access, you can access SQL view by following this demo or the instructions on this slide.
- After you type in your command and press **Execute**, you will get a status command to see if your command was successful or had any syntax errors:
- N.B. In order to see `mytable` appear in the Tables panel, you may have to refresh by selection **View** > **Refresh Tables**
- Recall since I didn't put my table names and fields in quotes, they get converted to capital letters (specific to LibreOffice Base–may not be the functionality of Access).

SQL mode

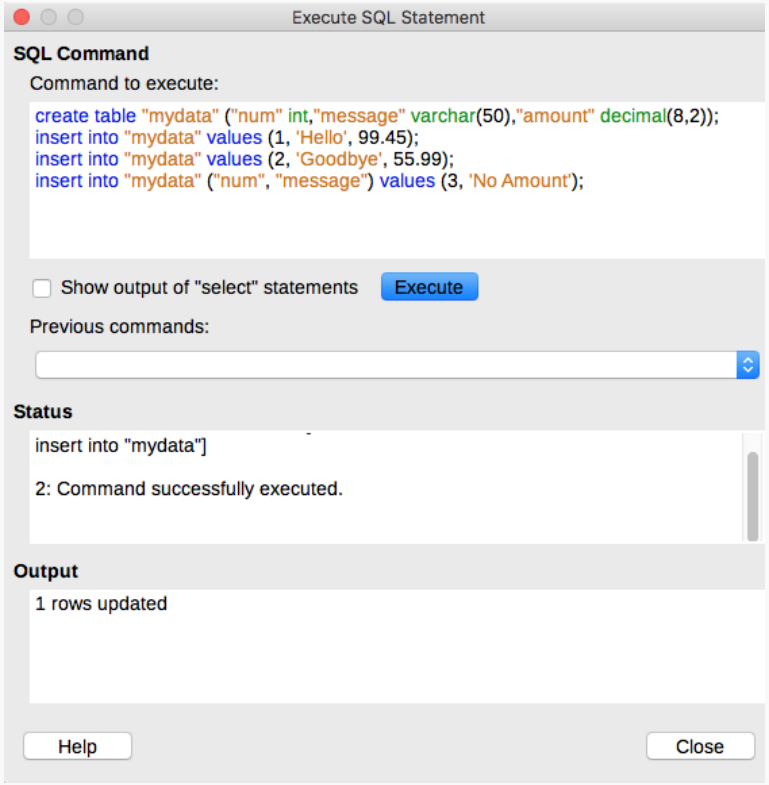

Execute SQL Statement
SQL Command
Command to execute:
create table mydata (num int,message varchar(50),amount decimal(8,2));
insert into mydata values (1, 'Hello', 99.45);
insert into mydata values (2, 'Goodbye', 55.99);
insert into mydata (num, message) values (3, 'No Amount');
Show output of "select" statements
Execute
Previous commands:
Status
1: Command successfully executed.
Output
1 rows updated

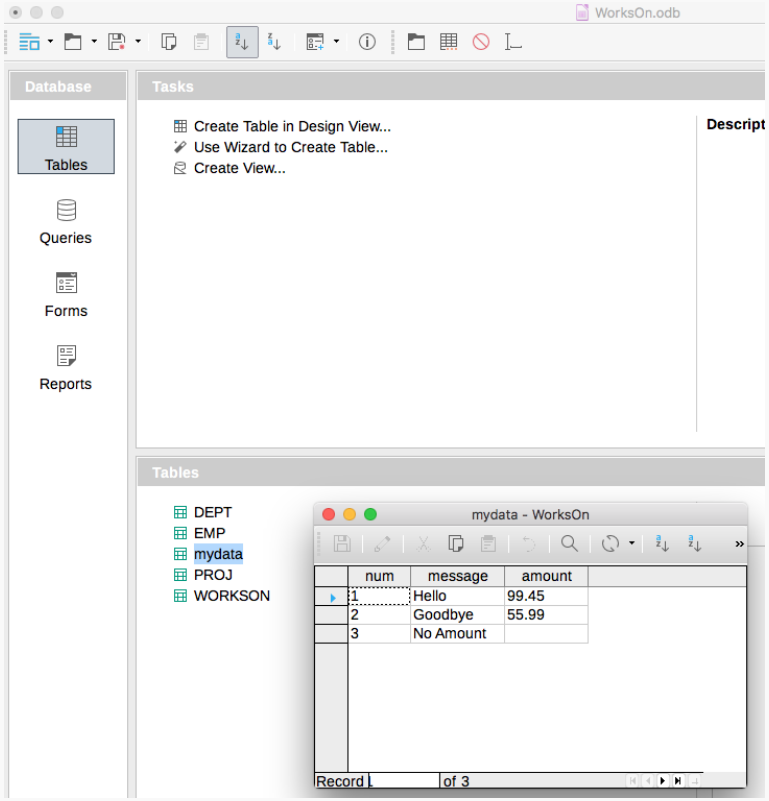
SQL mode


WorksOn.odb
Database
Tables
Queries
Forms
Reports
Tasks
Create Table in Design View...
Use Wizard to Create Table...
Create View...
Description
Tables
DEPT
EMP
MYDATA
PROJ
WORKSON
MYDATA - WorksOn
NUM
MESSAGE
AMOUNT
1
Hello
99.45
2
Goodbye
55.99
3
No Amount

LibreOffice Base using lowercase letters

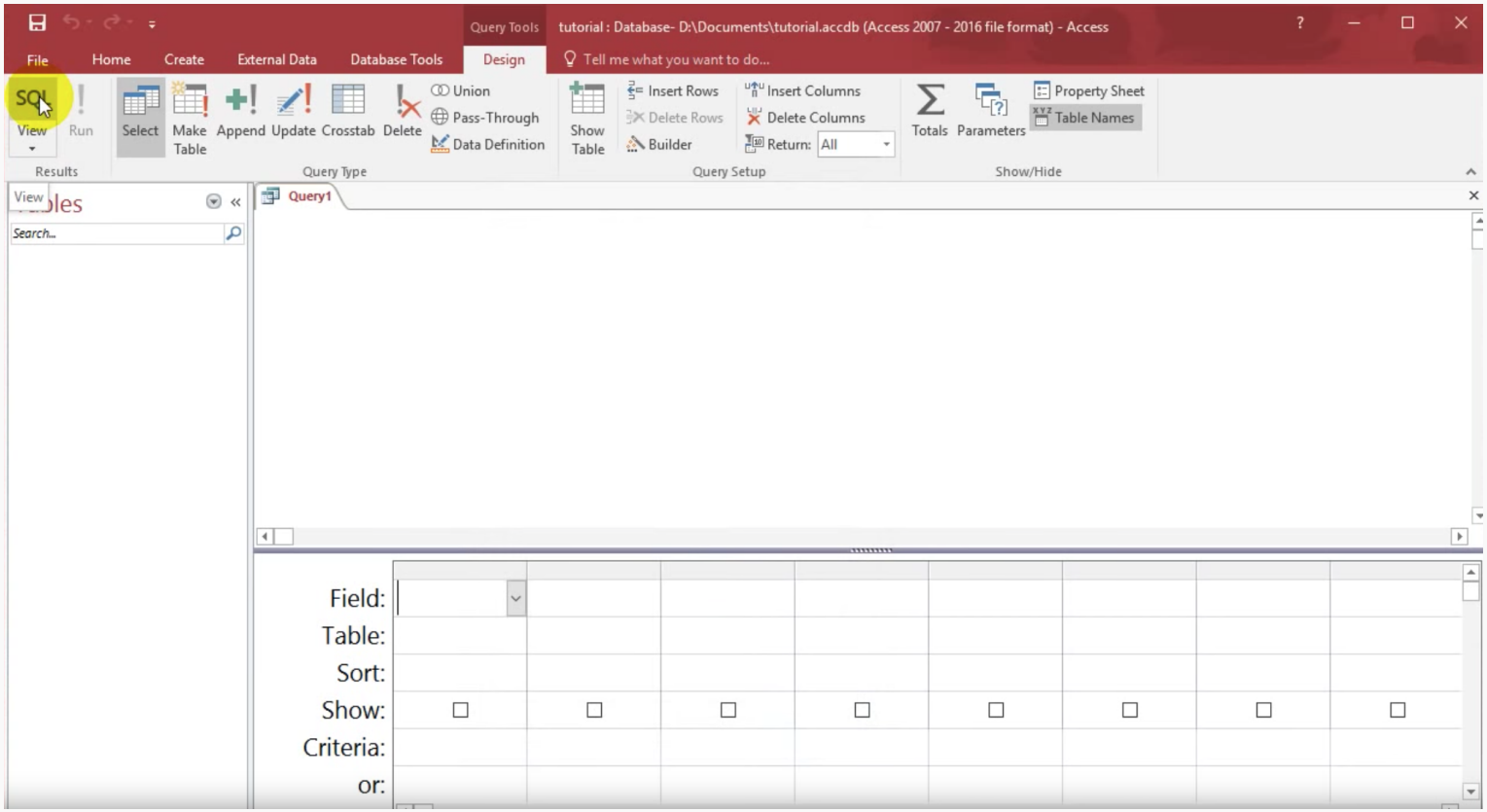
To force the names of tales and fields to be lowercase in LibreOffice base, use double quotes.





SQL view

In Microsoft Access, go to the Design tab in the ribbon and select the SQL View button



Comment

- For monetary values in LibreOffice base we used the data type `DECIMAL(`x`, 2 )`.
- For monetary values in Microsoft Access, use `CURRENCY` (with no arguments).

We can updating existing rows using the UPDATE statement. Examples:

1. Increase all employee salaries by 10%.

```
UPDATE SET salary = salary*1.10
```

2. Increase salary of employee E2 to $1 million and change his name:

```
UPDATE emp
SET salary = 1000000, ename='Rich Guy'
WHERE eno = 'E2';
```

INSERT vs UPDATE

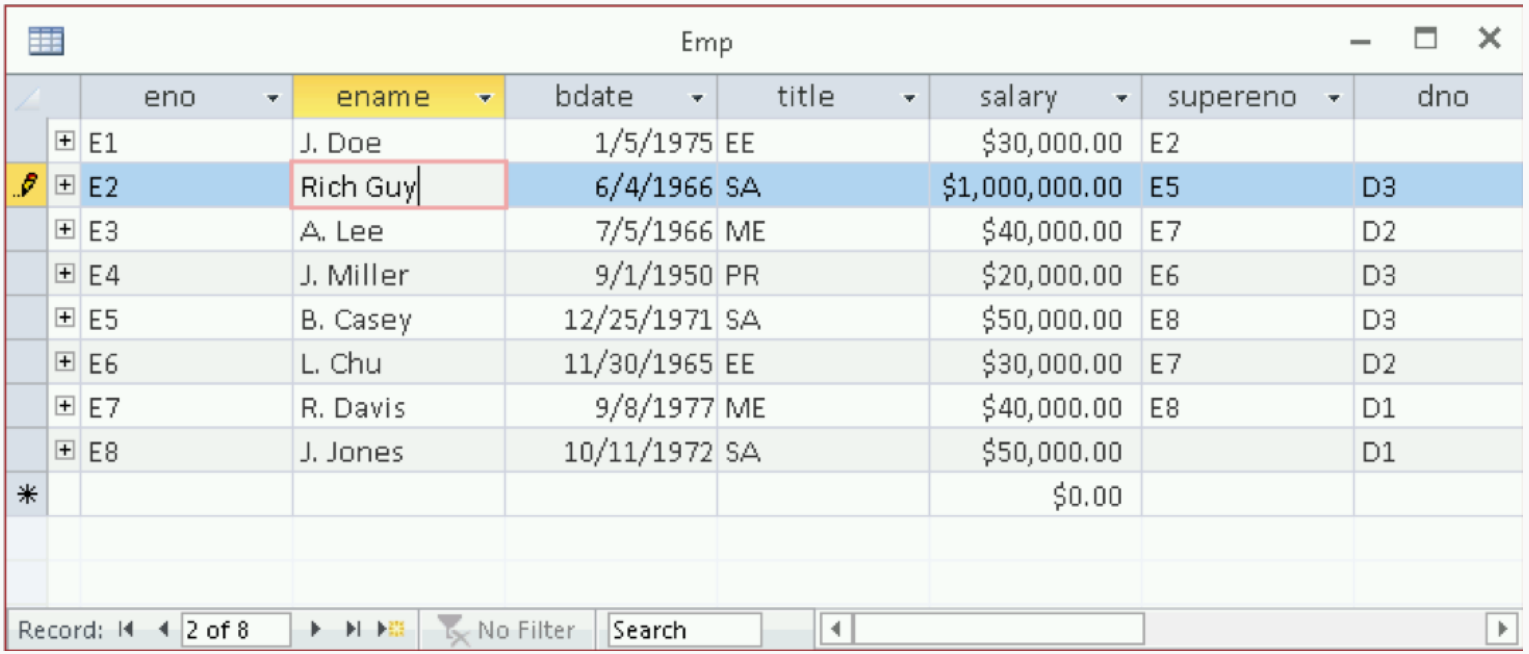
The main difference between INSERT and UPDATE is that INSERT is used to add new records to the table while UPDATE is used to modify the existing records in the table

Notes:

- Use `WHERE` to filter only the rows to update.
- The `WHERE` clause can also be used with other comparisons (eg. `=` , `>`, `<`) with numbers, characters, or dates.
- For example, you can filter on dates passed a certain time using `WHERE date > '2011-12-16'`
  - In Access, you will need to surround your dates with #, eg `WHERE date > #2011-12-16#`
- Notice how we can also change (i.e `SET`) more than one value at a time by separating the fields by commas.
- Notice that there are no commas separating keyword clauses

Updating in Microsoft Access and LibreOffice Base

The UPDATE command is supported by Microsoft Access/LibreOffice Base's GUI. That is, to modify individual data items with SQL code, simply enter the table, select the row and cell to update and change the data. Data is saved when you leave the row.



Example 5.2

Question Using the `mydata` table and the three rows previously inserted do these updates:

- Update all `amount` fields to be 99.99.
- Update the `num` field and set it to 10 for the record with `num = 1`.
- Update the message field to 'Changed' for the record with `num = 2`.

Perform these takes in SQL mode in either Access, Base, or sql-fiddle.com.

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL Donate About

```
1  CREATE TABLE mydata (
2    num int,
3    message varchar(50),
4    amount decimal(8,2)
5   );
6
7   INSERT into mydata VALUES (1, 'Hello', 99.45);
8   INSERT into mydata VALUES (2, 'Goodbye', 55.99);
9  INSERT into mydata (num, message) values (3, 'No Amount');
10
11 UPDATE mydata SET amount = 99.99;
12 UPDATE mydata SET num=10 WHERE num = 1;
13 UPDATE mydata SET message = 'Changed' WHERE num = 2;
14
```

Build Schema | Edit Fullscreen | Browser | [;]

```
1 SELECT * FROM mydata;
```

Run SQL | Edit Fullscreen | [;]

num	message	amount
10	Hello	99.99
2	Changed	99.99
3	No Amount	99.99

Record Count: 3; Execution Time: 4ms View Execution Plan link

Adding/Deleting Attributes

We can add or delete *columns* using the ALTER statement.

1. To add the column notes to our emp table:

```
ALTER TABLE emp
ADD COLUMN notes VARCHAR(50);
```

Alternatively we could just type ADD:

```
ALTER TABLE  emp
ADD notes VARCHAR(50);
```

2. To delete the column notes from our emp table:

```
ALTER TABLE emp
DROP COLUMN notes;
```

Adding/Deleting Attributes
Notes:

- Note that we don't need quotes for field names (just as we didn't need them in the `CREATE TABLE` statement).
- We can not at a column at a specific position in the table (they get placed in the last row by default).
- We can add multiple columns at a time using the syntax:

```
ALTER TABLE table_name
ADD column_1 column_definition, column_2
column_definition;
```

Delete Statements

Rows are deleted using the `DELETE` statement. Examples:

- Fire everyone in the company.

```
DELETE FROM emp;
```

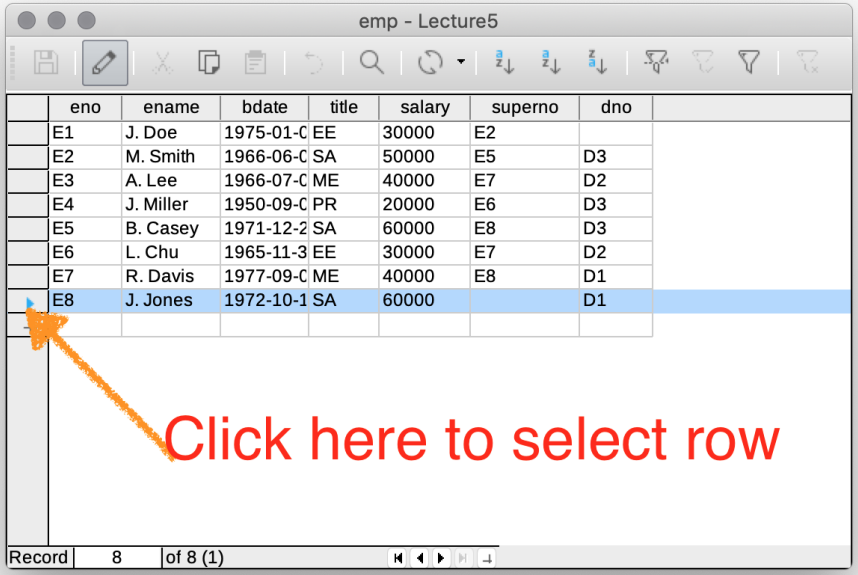
- Fire everyone making over $35,000.

```
DELETE FROM emp WHERE salary > 35000;
```

- You can think of the previous example as deleting all the visible rows in an excel sheet after applying a filter.

Deleting Data in Microsoft Access/LibreOffice Base

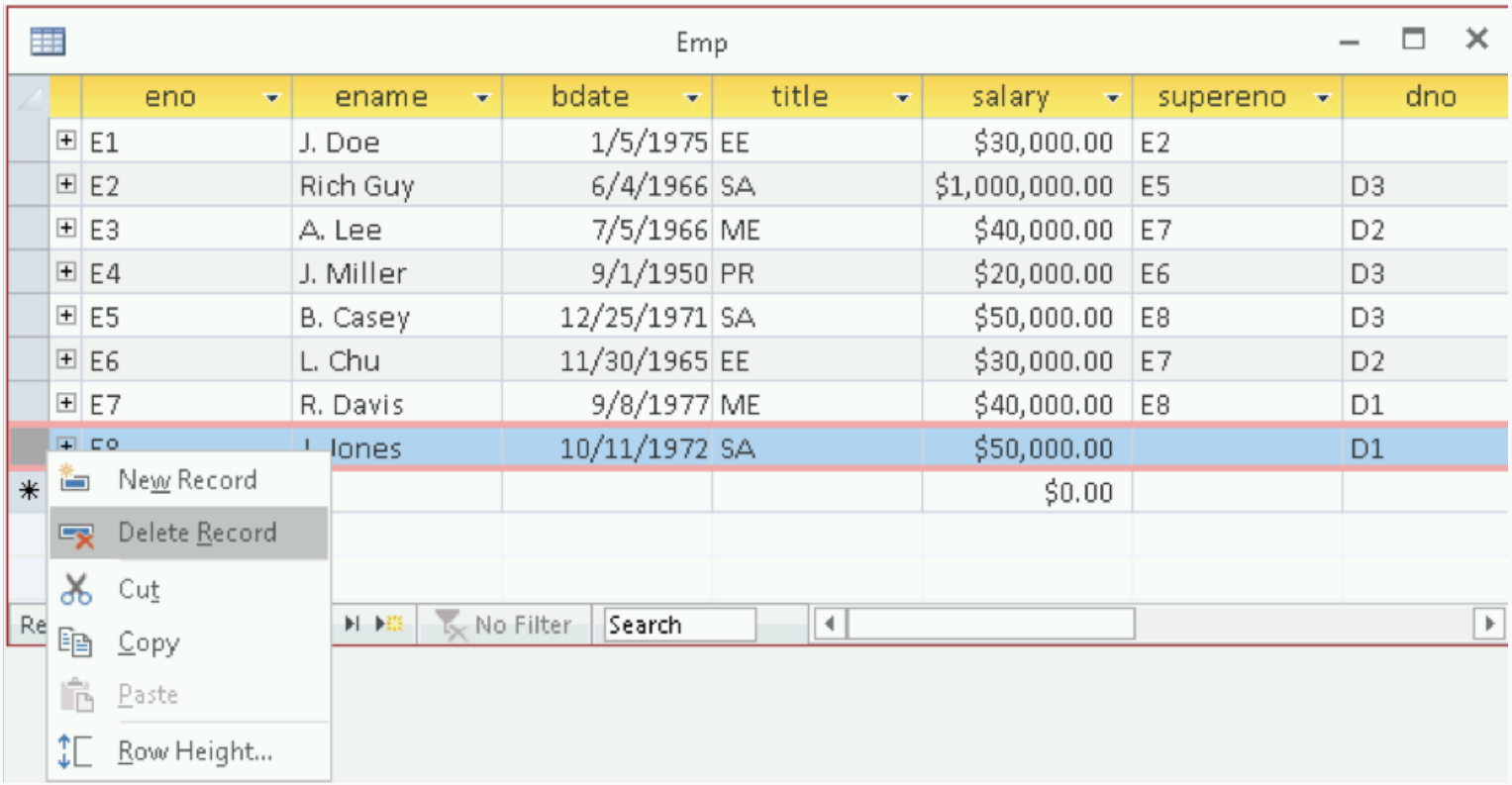
The DELETE command is supported by Microsoft Access/LibreOffice Base GUI. To delete an individual row, select the row (by clicking the gray space to the left of the row), then press Alt + Delete key or right click and select **Delete Row** from pop-up menu.[3]



[3] I sometimes get weird errors when trying to use this feature

Deleting Data in Microsoft Access/LibreOffice Base

The DELETE command is supported by Microsoft Access/LibreOffice Base. To delete an individual row, select the row (by clicking the gray space to the left of the row), then press Delete key or right click select **Delete Record** from pop-up menu.



Example 5.3 (Delete)

Using the `mydata` table and the three rows previously inserted do these deletes:

- Delete the row with num = 10.
- Delete the row(s) with message > 'D'.
- Delete all rows.

You can use Access, LibreOffice Base, or sqlfiddle.com.

Let's have a look at the current table mydata

SQL Fiddle | MySQL 5.6 | View Sample Fiddle | Clear | Text to DDL | Donate

```
1  CREATE TABLE mydata (
2    num int,
3    message varchar(50),
4    amount decimal(8,2)
5   );
6
7   INSERT into mydata VALUES (1, 'Hello', 99.45);
8   INSERT into mydata VALUES (2, 'Goodbye', 55.99);
9  INSERT into mydata (num, message) values (3, 'No Amount');
10
11 UPDATE mydata SET amount = 99.99;
12 UPDATE mydata SET num=10 WHERE num = 1;
13 UPDATE mydata SET message = 'Changed' WHERE num = 2;
14
15
```

Build Schema | Edit Fullscreen | Browser | [;]

```
1 SELECT * FROM mydata;
```

Run SQL | Edit Fullscreen | [;]

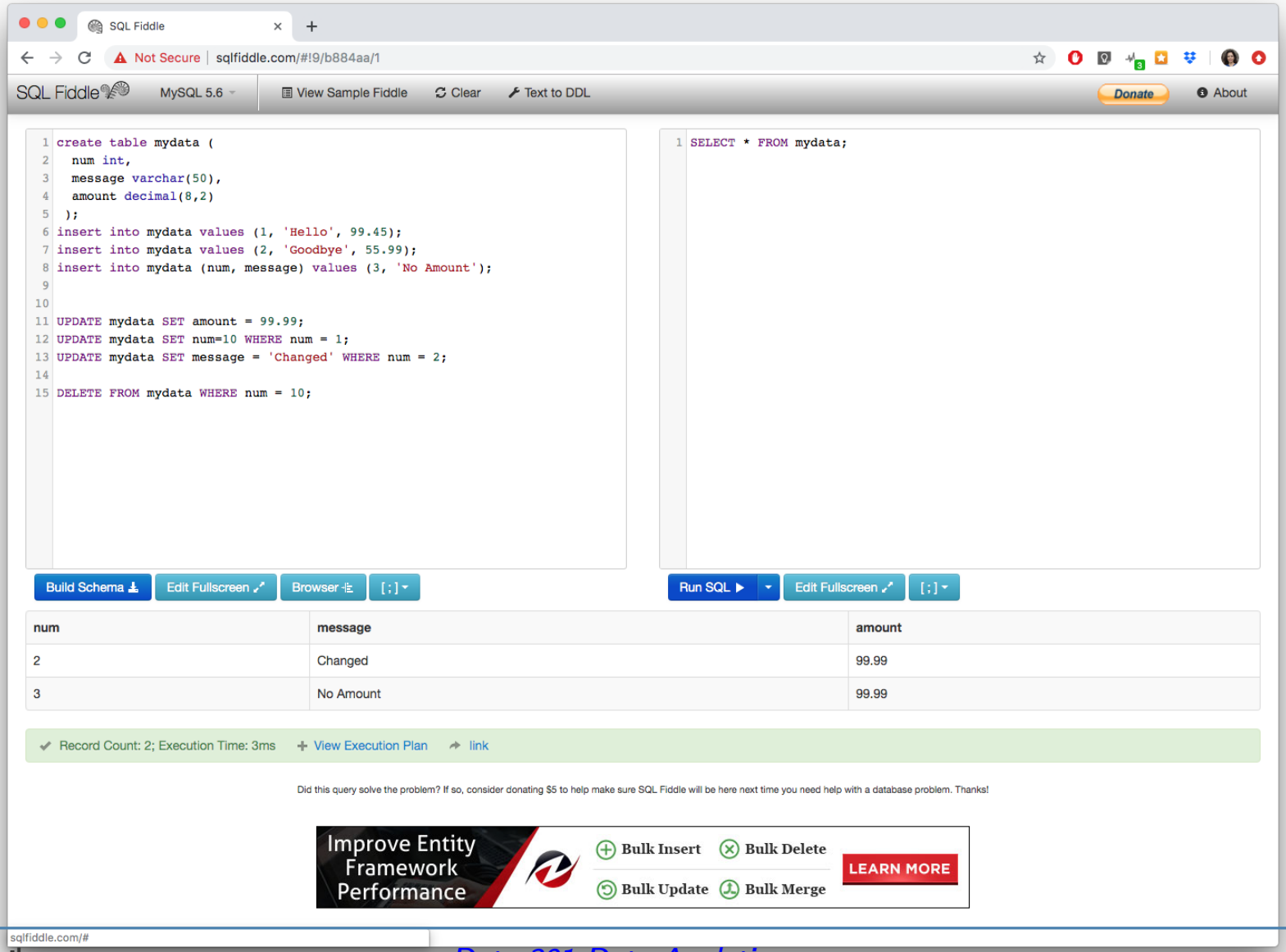
num	message	amount
10	Hello	99.99
2	Changed	99.99
3	No Amount	99.99

Record Count: 3; Execution Time: 4ms | View Execution Plan | link

Did this query solve the problem? If so, consider donating $5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!

Delete the row with num = 10

```
DELETE FROM mydata WHERE num = 10;
```



Delete the row(s) with message > 'D'.

Let's look at what > 'D' looks like anyhow:

SQL Fiddle | MySQL 5.6 | View Sample Fiddle | Clear | Text to DDL | Donate | About

```
1  CREATE TABLE mydata (
2    num int,
3    message varchar(50),
4    amount decimal(8,2)
5   );
6
7   INSERT into mydata VALUES (1, 'Hello', 99.45);
8   INSERT into mydata VALUES (2, 'Goodbye', 55.99);
9  INSERT into mydata (num, message) values (3, 'No Amount');
10
11 UPDATE mydata SET amount = 99.99;
12 UPDATE mydata SET num=10 WHERE num = 1;
13 UPDATE mydata SET message = 'Changed' WHERE num = 2;
14
15
```

Build Schema | Edit Fullscreen | Browser | [;]

```
1 SELECT * FROM mydata WHERE message > "D";
```

Run SQL | Edit Fullscreen | [;]

num	message	amount
10	Hello	99.99
3	No Amount	99.99

Record Count: 2; Execution Time: 7ms | View Execution Plan | link

Delete the row(s) with message > 'D'.

So removing the rows greater than D yeilds:

SQL Fiddle | MySQL 5.6 | View Sample Fiddle | Clear | Text to DDL | Donate | About

```
1  CREATE TABLE mydata (
2    num int,
3    message varchar(50),
4    amount decimal(8,2)
5   );
6
7   INSERT into mydata VALUES (1, 'Hello', 99.45);
8   INSERT into mydata VALUES (2, 'Goodbye', 55.99);
9  INSERT into mydata (num, message) values (3, 'No Amount');
10
11 UPDATE mydata SET amount = 99.99;
12 UPDATE mydata SET num=10 WHERE num = 1;
13 UPDATE mydata SET message = 'Changed' WHERE num = 2;
14
15 DELETE FROM mydata WHERE message > 'D';
```

Build Schema | Edit Fullscreen | Browser | [;]

```
1 SELECT * FROM mydata;
```

Run SQL | Edit Fullscreen | [;]

num	message	amount
2	Changed	99.99

Record Count: 1; Execution Time: 4ms | View Execution Plan | link

Comparison operators on strings

- Comparison operators on strings uses lexicographical ordering.

- Warning: This is similar to the alphabetical order except sometimes all the lowercase letters come before all the uppercase letters and other times all the uppercase letters come before all the lowercase letters.

Comparison operators on strings

On DB-fiddle $C < c < Changed < change$ in lexicographical order, hence:

Schema SQL
CREATE TABLE, INSERT, UPDATE etc.

```
CREATE TABLE LEXO(
                STRS VARCHAR(30)
);
--- And we populate it with the following strings

INSERT INTO LEXO VALUES('C');
INSERT INTO LEXO VALUES('c');
INSERT INTO LEXO VALUES('Changed');
INSERT INTO LEXO VALUES('changed');
```

Text to DDL

Query SQL

```
SELECT *
FROM LEXO
ORDER BY STRS ASC;
```

Results

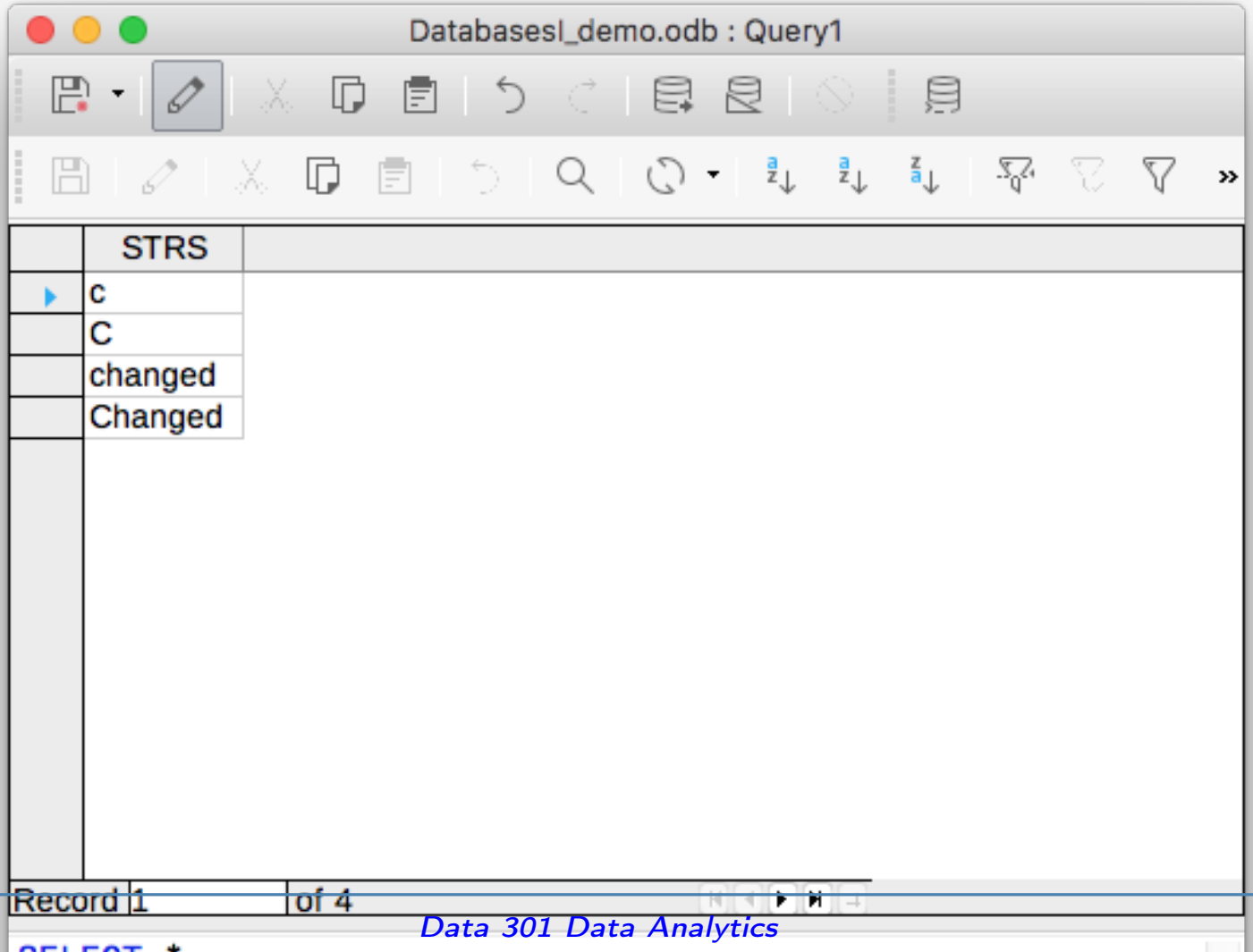
Copy as Markdown

Query #1 Execution time: 1ms

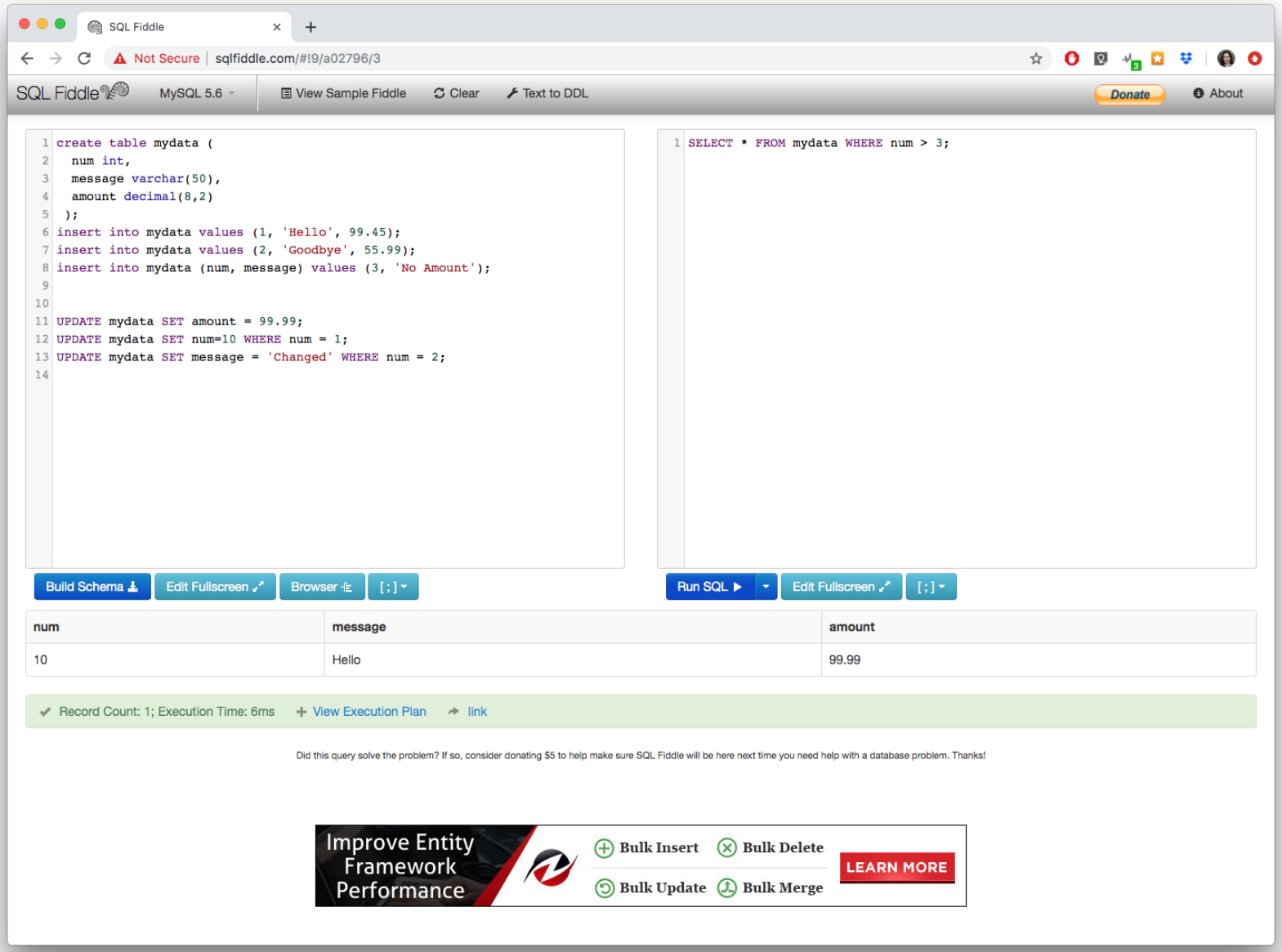
STRS
C
c
Changed
changed

Comparison operators on strings

On LibreOffice Base $c < C < changed < Changed$ in lexicographical order, hence:

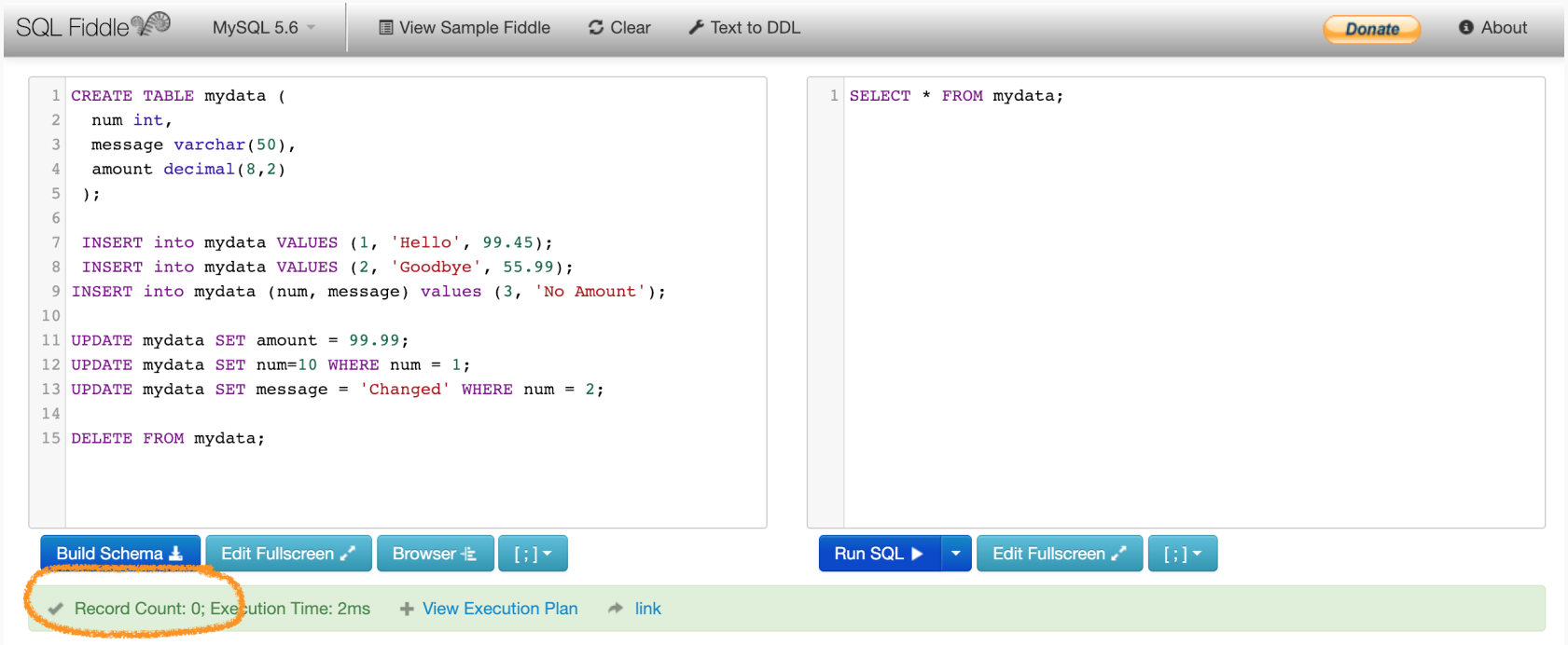


Using > on numbers:



Delete all rows

Removing all rows and querying all the data from `mydata` will return zero records.



DROP TABLE

When we delete all the records from a table, that table still exists (its just empty).

The command `DROP TABLE` is used to delete the table and *all its data* from the database. An example of usage:

```
DROP TABLE emp;
```

Warning

The database does not confirm if you really want to drop the table and delete its data. The effect of the command is immediate.

Create Tables Question

Example 5.4

How many of the following statements are TRUE?

1. Each field in the CREATE TABLE statement is separated by a comma.
2. The data type for a field is optional.
3. You can create two tables in a database with the same name.
4. A table will not be dropped (with DROP TABLE) if it contains data.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Create Tables Question

Answer

How many of the following statements are TRUE?

1. Each field in the CREATE TABLE statement is separated by a comma.
2. The data type for a field is optional.
3. You can create two tables in a database with the same name.
4. A table will not be dropped (with DROP TABLE) if it contains data.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Create Tables Question

Answer

How many of the following statements are TRUE?

1. Each field in the CREATE TABLE statement is separated by a comma.
2. The data type for a field is optional.
3. You can create two tables in a database with the same name.
4. A table will not be dropped (with DROP TABLE) if it contains data.

A) 0 B) 1 C) 2 D) 3 E) 4

Create Tables Question

Answer

How many of the following statements are TRUE?

1. Each field in the CREATE TABLE statement is separated by a comma.
2. The data type for a field is optional.
3. You can create two tables in a database with the same name.
4. A table will not be dropped (with DROP TABLE) if it contains data.

A) 0 B) 1 C) 2 D) 3 E) 4

Create Tables Question

Answer

How many of the following statements are TRUE?

1. Each field in the CREATE TABLE statement is separated by a comma.
2. The data type for a field is optional.
3. You can create two tables in a database with the same name.
4. A table will not be dropped (with DROP TABLE) if it contains data.

A) 0 B) 1 C) 2 D) 3 E) 4

Create Tables Question

Answer

How many of the following statements are TRUE?

1. Each field in the CREATE TABLE statement is separated by a comma.
2. The data type for a field is optional.
3. You can create two tables in a database with the same name.
4. A table will not be dropped (with DROP TABLE) if it contains data.

A) 0 B) *1* C) 2 D) 3 E) 4

Example 5.5 (Insert Question)

How many of the following statements are TRUE?

1. You must always specify the fields being inserted with INSERT statement.
2. If you list the fields, the fields must be in the same order as the table.
3. If you do not provide a value for a number field, it will default to 1.
4. Number data items are enclosed in single quotes.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Answer

How many of the following statements are TRUE?

1. You must always specify the fields being inserted with INSERT statement.
2. If you list the fields, the fields must be in the same order as the table.
3. If you do not provide a value for a number field, it will default to 1.
4. Number data items are enclosed in single quotes.

A) 0 B) 1 C) 2 D) 3 E) 4

Answer

How many of the following statements are TRUE?

1. You must always specify the fields being inserted with INSERT statement.
2. If you list the fields, the fields must be in the same order as the table.
3. If you do not provide a value for a number field, it will default to 1.
4. Number data items are enclosed in single quotes.

A) 0 B) 1 C) 2 D) 3 E) 4

Answer

How many of the following statements are TRUE?

1. You must always specify the fields being inserted with INSERT statement.
2. If you list the fields, the fields must be in the same order as the table.
3. If you do not provide a value for a number field, it will default to 1.
4. Number data items are enclosed in single quotes.

A) 0 B) 1 C) 2 D) 3 E) 4

Answer

How many of the following statements are TRUE?

1. You must always specify the fields being inserted with INSERT statement.
2. If you list the fields, the fields must be in the same order as the table.
3. If you do not provide a value for a number field, it will default to 1.
4. Number data items are enclosed in single quotes.

A) 0 B) 1 C) 2 D) 3 E) 4

Answer

How many of the following statements are TRUE?

1. You must always specify the fields being inserted with INSERT statement.
2. If you list the fields, the fields must be in the same order as the table.
3. If you do not provide a value for a number field, it will default to 1.
4. Number data items are enclosed in single quotes.

A) *0* B) 1 C) 2 D) 3 E) 4

Example 5.6 (Update Question)

How many of the following statements are TRUE?

1. You may update more than one row at a time.
2. If the UPDATE has no WHERE clause, it updates all rows.
3. You may update zero or more rows using a UPDATE statement.
4. UPDATE may change more than one data value (column) in a row.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Answer

How many of the following statements are TRUE?

1. You may update more than one row at a time.
2. If the UPDATE has no WHERE clause, it updates all rows.
3. You may update zero or more rows using a UPDATE statement.
4. UPDATE may change more than one data value (column) in a row.

A) 0 B) 1 C) 2 D) 3 E) 4

Answer

How many of the following statements are TRUE?

1. You may update more than one row at a time.
2. If the UPDATE has no WHERE clause, it updates all rows.
3. You may update zero or more rows using a UPDATE statement.
4. UPDATE may change more than one data value (column) in a row.

A) 0 B) 1 C) 2 D) 3 E) 4

Answer

How many of the following statements are TRUE?

1. You may update more than one row at a time.
2. If the UPDATE has no WHERE clause, it updates all rows.
3. You may update zero or more rows using a UPDATE statement.
4. UPDATE may change more than one data value (column) in a row.

A) 0 B) 1 C) 2 D) 3 E) 4

Answer

How many of the following statements are TRUE?

1. You may update more than one row at a time.
2. If the UPDATE has no WHERE clause, it updates all rows.
3. You may update zero or more rows using a UPDATE statement.
4. UPDATE may change more than one data value (column) in a row. (use commas to separate parameters)

A) 0 B) 1 C) 2 D) 3 E) *4*

Example 5.7 (Delete Question)

How many of the following statements are TRUE?

1. A DELETE with no WHERE clause will delete all rows.
2. The DELETE keyword is case-sensitive.
3. It is possible to DELETE zero or more rows using a WHERE clause.
4. `DELETE mydata` will delete the entire table and its data.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Answer

How many of the following statements are TRUE?

1. A DELETE with no WHERE clause will delete all rows.
2. The DELETE keyword is case-sensitive.
3. It is possible to DELETE zero or more rows using a WHERE clause.
4. `DELETE mydata` will delete the entire table and its data.

A) 0 B) 1 C) 2 D) 3 E) 4

Answer

How many of the following statements are TRUE?

1. A DELETE with no WHERE clause will delete all rows.
2. The DELETE keyword is case-sensitive.
3. It is possible to DELETE zero or more rows using a WHERE clause.
4. `DELETE mydata` will delete the entire table and its data.

A) 0 B) 1 C) 2 D) 3 E) 4

Answer

How many of the following statements are TRUE?

1. A DELETE with no WHERE clause will delete all rows.
2. The DELETE keyword is case-sensitive.
3. It is possible to DELETE zero or more rows using a WHERE clause.
4. `DELETE mydata` will delete the entire table and its data.

A) 0 B) 1 C) 2 D) 3 E) 4

Answer

How many of the following statements are TRUE?

1. A DELETE with no WHERE clause will delete all rows.
2. The DELETE keyword is case-sensitive.
3. It is possible to DELETE zero or more rows using a WHERE clause.
4. `DELETE mydata` will delete the entire table and its data. for this we need DROP table

A) 0 B) 1 C) *2* D) 3 E) 4