Introduction to SQL

Research Computing Services

Yun Shen

http://rcs.bu.edu

http://rcs.bu.edu/eval

help@scc.bu.edu

A little bit of about our group and me

- Research Computing services, visit http://rcs.bu.edu for more info
  - Consulting
  - Teaching
  - Graphics
  - Optimization
- Experience:
  - Database programming
  - Software development

Tutorial Outlines

- What is SQL
- SQL History
- Terminology By Examples
- SQL Syntax By Examples
- SQL Category

- Small yet worth noting points
- Tutorial sample db overview
- Schema of the sample db
- Data of the sample db
- Hands on Tutorial Setup

BOSTON UNIVERSITY

3

What is SQL ('Structured Query Language')?

- SQL stands for 'Structured Query Language'

- SQL is domain-specific language, NOT a general programming language

  - SQL is specialized to handle 'structured data' that follows relational model – data that incorporates relations among entities and variables.

  - Used to interact with relational databases to manage data: create, populate, modify, or destroy data. Also can manage data access

BOSTON
UNIVERSITY

SQL is a standard language

- Nevertheless, SQL is a 'language'. It has its language specification – a set of language elements, rules and syntax

- Rigid and structural – have both advantages and disadvantages

  - Since the underlying data model is structural, SQL is very 'structural' too - requiring rigid predefined *schema* as compared with those of 'noSQL'

  - Syntax and grammar is also strict

- SQL specific features – triggers, stored procedures

History of SQL

- First developed in 1970s by two scientists at IBM following a theory of 'relational algebra' by Edgar F. Codd, who was also an IBM scientist.

- First commercial implementation of SQL-based RDBMS was Oracle's V2.

- First adopted by ANSI in 1986, and ISO in 1987 as standard.

- The latest version of the SQL standard is from 2016. There have been very many versions in between.

- Though standardized, this does not necessarily mean SQL code can be migrated between different RDBMS seamlessly (Why?)

Terminology - Structure

- Database
- Table
- Column
- Row

- Relation
- Primary key
- Foreign key

BOSTON
UNIVERSITY

Take *sample_ecomm.db* as an Example - schema
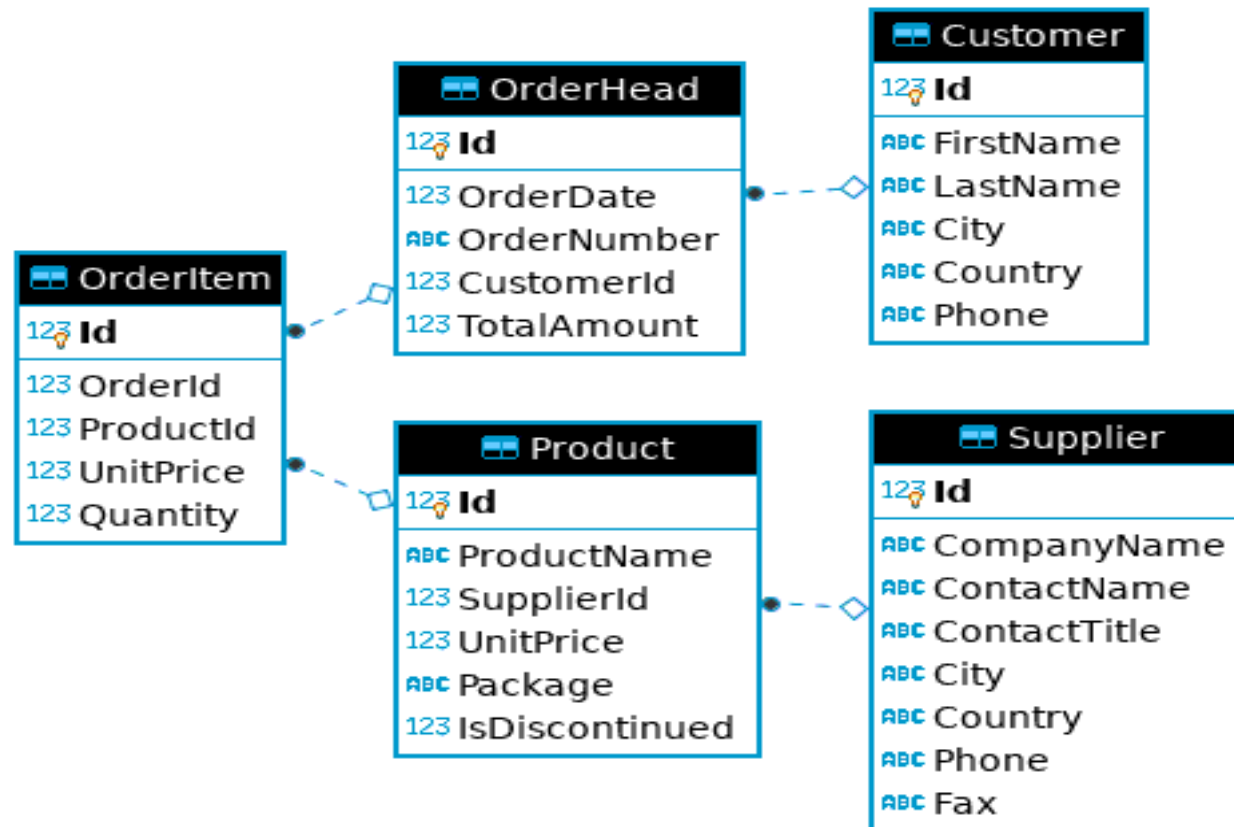
Customer

Id	FirstName	LastName	City	Country	Phone
1	Maria	Anders	Berlin	Germany	030-0074321
2	Ana	Trujillo	México D.F.	Mexico	(5); 555-4729
3	Antonio	Moreno	México D.F.	Mexico	(5); 555-3932
4	Thomas	Hardy	London	UK	(171); 555-7788
5	Christina	Berglund	Luleå	Sweden	0921-12 34 65

Terminology - SQL Language Elements

- Clause
- Statement
- Query
- Function
- Stored Procedure

- Predicate
- Expression
- Keyword
- Identifier

A SQL Example From Wikipedia

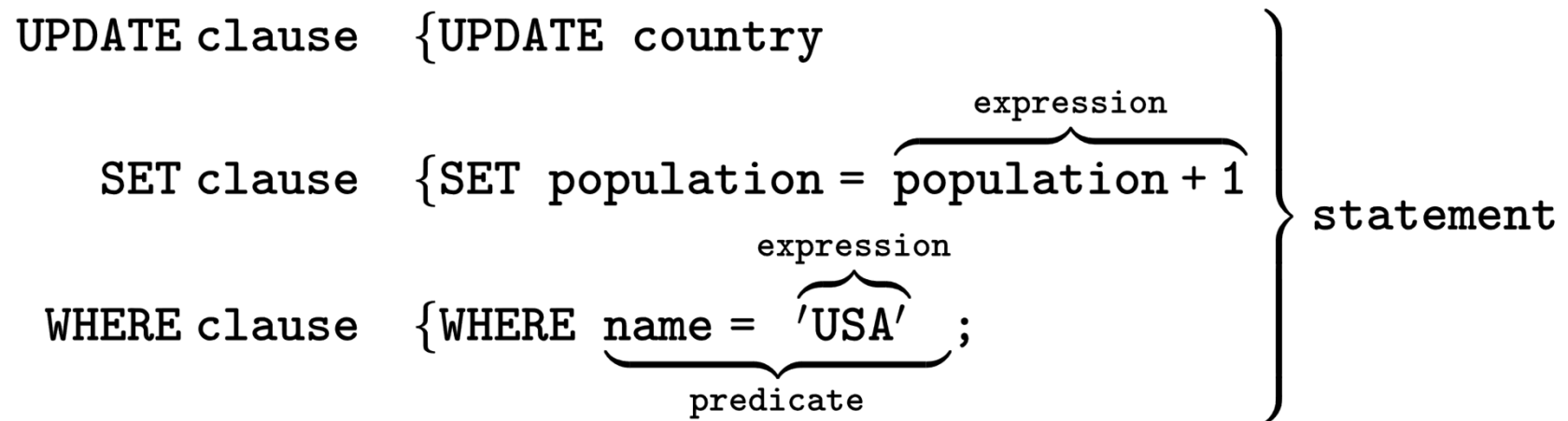Let's take the following SQL UPDATE statement as an example:

UPDATE country
SET population=population+1
WHERE name='USA'

A chart showing several of the SQL language elements that compose a single statement
(source: https://wikimedia.org/api/rest_v1/media/math/render/svg/b83ad563285f7b0ebb325226d91f25ca0bffa7cd)

A SQL Example From Wikipedia

$$\left.\begin{array}{rl} \text{UPDATE clause} & \{\texttt{UPDATE country} \\ \text{SET clause} & \{\overset{\text{expression}}{\texttt{SET population = }\overbrace{\texttt{population + 1}}} \\ \text{WHERE clause} & \{\texttt{WHERE }\underbrace{\texttt{name = }\overbrace{\texttt{'USA'}}^{\text{expression}}}_{\text{predicate}}\texttt{ ;} \end{array}\right\} \text{statement}$$

A chart showing several of the SQL language elements that compose a single statement
(source: https://wikimedia.org/api/rest_v1/media/math/render/svg/b83ad563285f7b0ebb325226d91f25ca0bffa7cd)

BOSTON
UNIVERSITY

Our Own Query Example

SELECT FirstName, LastName -- SELECT clause
FROM Customer -- FROM clause
WHERE Id=1 – WHERE Clause

Clean way:

SELECT FirstName, LastName FROM Customer WHERE Id=1

Complete Query Statement Syntax – Order Matters !

Clause	Priority	Required?	Covered In Tutorial?
SELECT *<columns>*	5.	Mandatory	✓
FROM *<table>*	1.	Mandatory	✓
WHERE *<predicate on rows>*	2.	Optional	✓
GROUP BY *<columns>*	3.	Optional	✓
HAVING *<predicate on groups>*	4.	Optional, work with GROUP BY	
ORDER BY *<columns>*	6.	Optional	✓
OFFSET	7.	Optional	
FETCH FIRST	8.	Optional	

SQL Category

1. Data Query Language (DQL) - used to query data
2. Data Manipulation Language (DML) – used to create/modify/destroy data
3. Data Definition Language (DDL) – used to define database schema
4. Data Control Language (DCL) – used for security and access control

Most Important SQL Statements

- **SELECT -** extracts data from a database (DQL)

- **UPDATE -** updates data in a database (DML)

- **DELETE -** deletes data from a database (DML)

- **INSERT** - inserts new data into a database (DML)

- **CREATE DATABASE -** creates a new database (DDL)

- **CREATE TABLE -** creates a new table (DDL)

- **DROP TABLE -** deletes a table (DDL)

BOSTON UNIVERSITY

16

Attention Please !

1. SQL keywords and table/column names are **NOT case sensitive**: 'select' and 'SELECT' are the same
2. values stored in a table can be **case-sensitive** – depending on configuration
3. Usually single quotes ('') or double quotes ("") don't matter, but could be configured otherwise
4. Semicolon ';' is the standard way to separate SQL statements. It can be required in some DBMS. So always end each statement with a ';' even after a single statement
5. Comments can be used to make SQL more readable. Usually '--' for single line comment, and '/*' and '*/' for multiline comments. Add '--' at the beginning to indicate a comment line
6. Use alias to make query clear to understand. "AS" keyword can be omitted sometimes.

Standard is NOT STANDARD!!

Standard is NOT STANDARD – none of SQL standard is fully implemented by all vendors. Pay attention to the differences that each vendor's implementation has from the SQL 'standard'

BOSTON
UNIVERSITY

In this Tutorial

- We will use all upper-case for all keywords
- We will use double quotes "" to indicate strings
- We will end each SQL statement with a ';'

Keywords Used in this Tutorial

- INNER JOIN
- SELECT
- *
- FROM
- ORDER BY
- ASC
- DESC

- AND
- OR
- NOT
- WHERE
- LIMIT
- DISTINCT
- AS

- GROUP BY
- ON
- !=
- INSERT
- UPDATE
- DELETE
- CREATE

- TABLE
- LIKE
- %
- INTO
- VALUES
- DROP
- NULL

Functions Used in this Tutorial

- COUNT()
- MIN()
- MAX()
- AVG()
- SUM()
- REPLACE()

Hands On Demo

Basic	SELECT + WHERE
Aggregation	SELECT + GROUP BY
JOIN	SELECT + JOIN
WRITE Queries	INSERT/UPDATE/DELETE

Read Only

Tutorial Tools and Files Overview
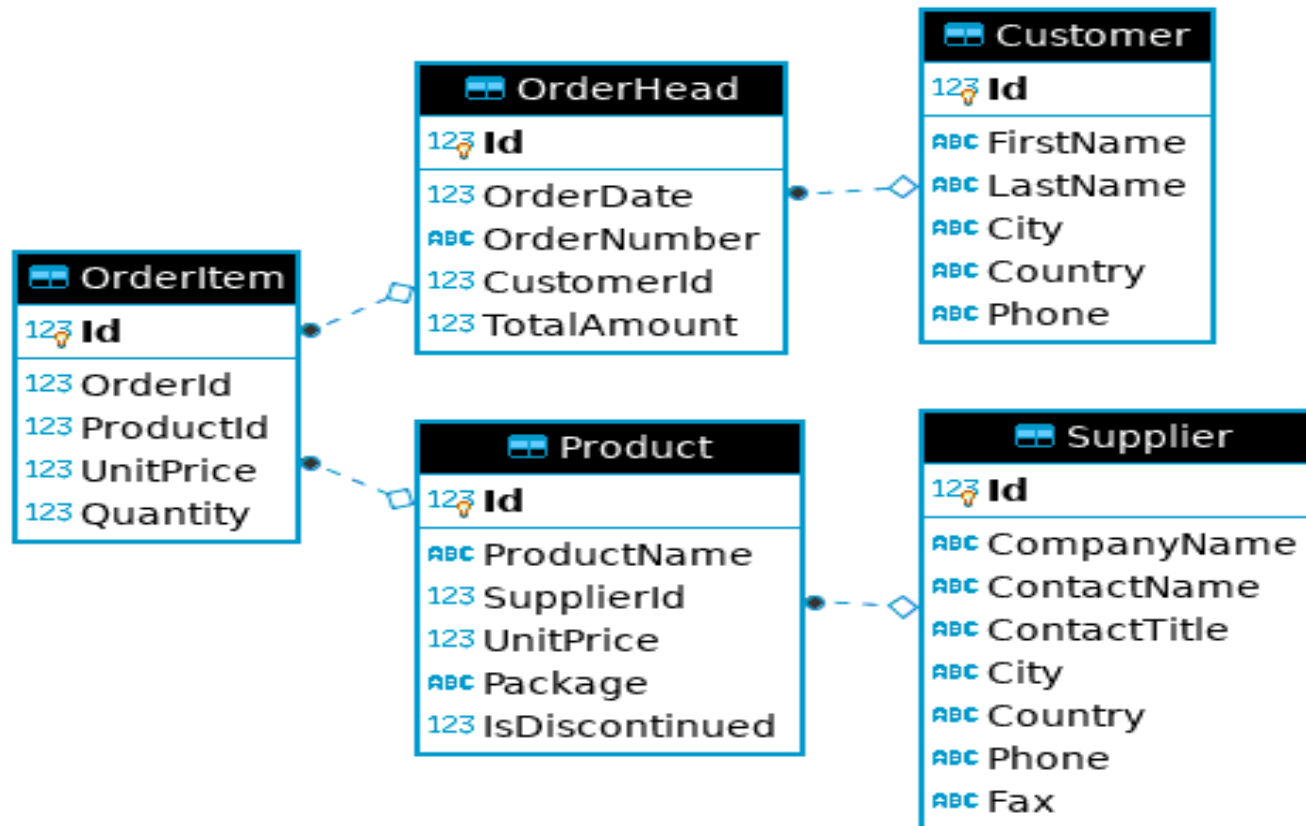
DB GUI : DB Browser for SQLite

- sufficient yet simple/clean interface for demo purpose
- SQLite engine is already embedded in this tool

Sample DB: sample_ecomm.db
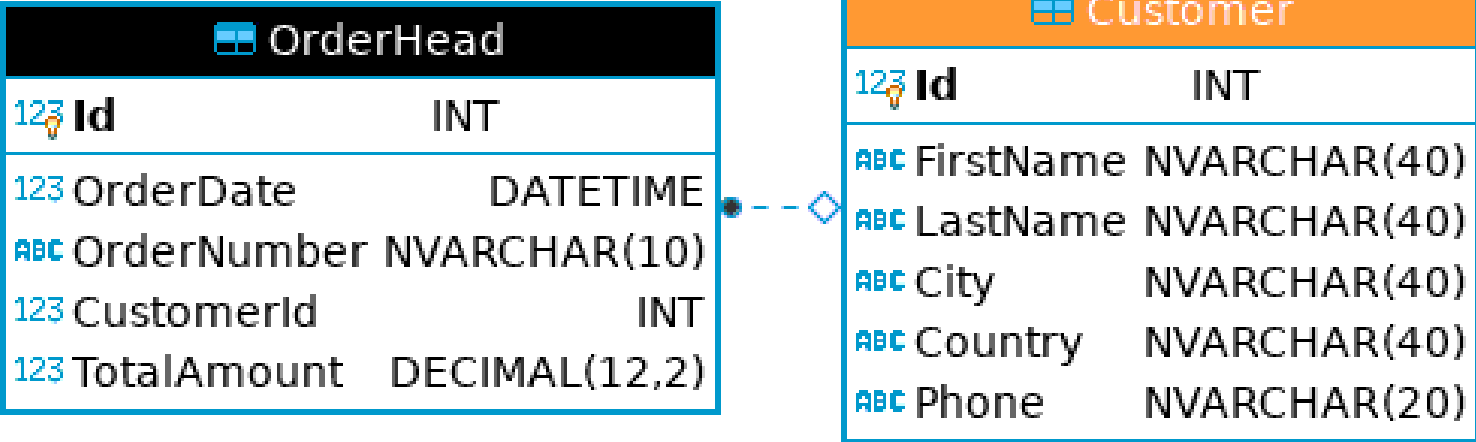
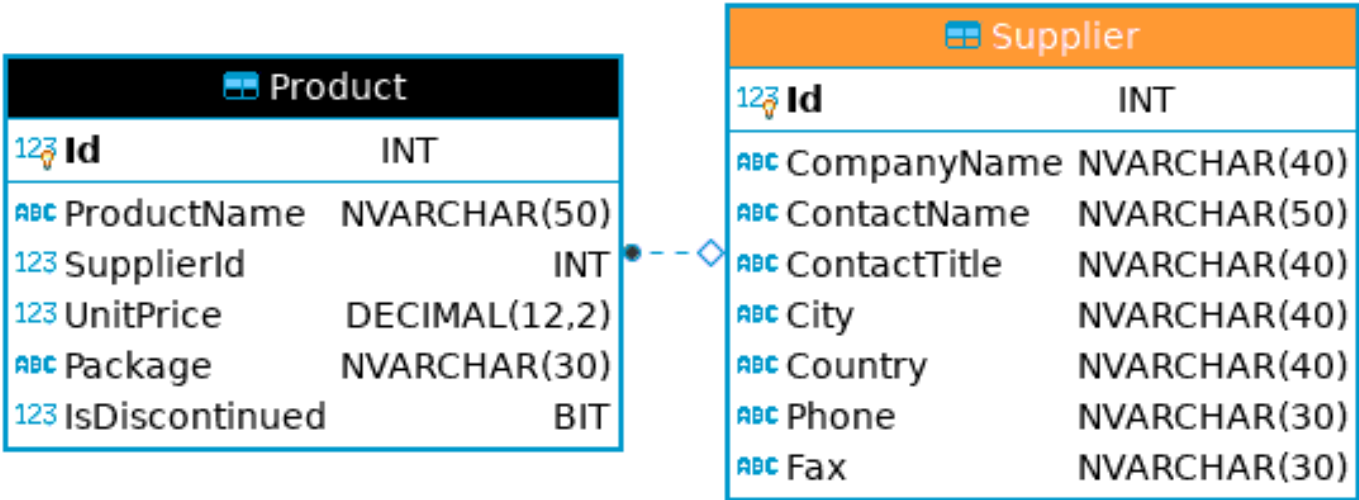- a simple example e-commerce db. We will explore it a bit more …

BOSTON
UNIVERSITY
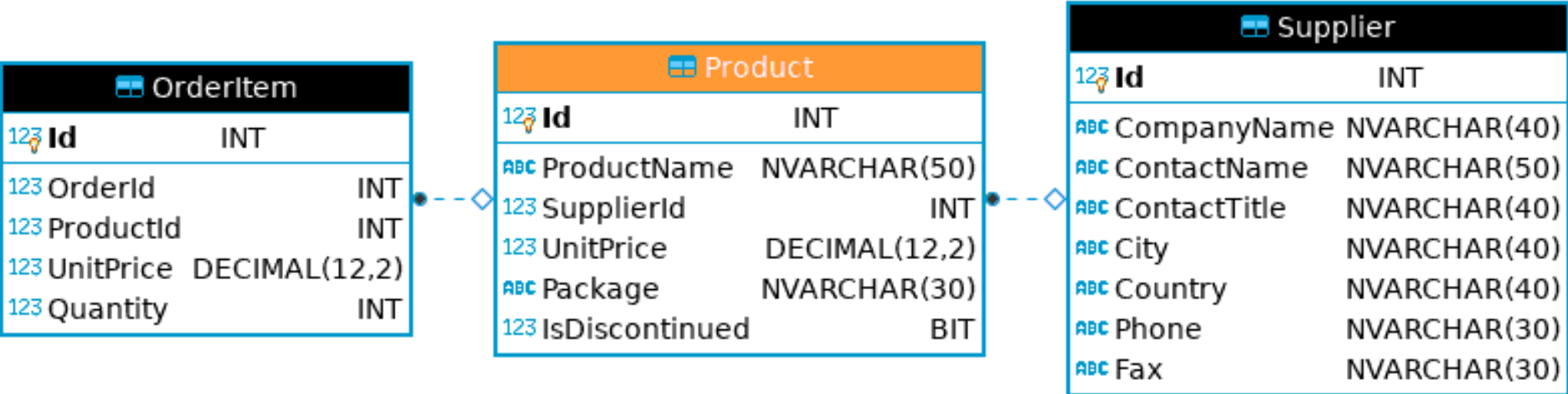
sample_ecomm.db E-R Diagram

Look Into Individual Table – Customer

Look Into Individual Table – Supplier

Look Into Individual Table – Product

Look Into Individual Table – OrderHead

OrderItem

123 **Id**	INT
123 OrderId	INT
123 ProductId	INT
123 UnitPrice	DECIMAL(12,2)
123 Quantity	INT

OrderHead

123 **Id**	INT
123 OrderDate	DATETIME
ABC OrderNumber	NVARCHAR(10)
123 CustomerId	INT
123 TotalAmount	DECIMAL(12,2)

Customer

123 **Id**	INT
ABC FirstName	NVARCHAR(40)
ABC LastName	NVARCHAR(40)
ABC City	NVARCHAR(40)
ABC Country	NVARCHAR(40)
ABC Phone	NVARCHAR(20)

BOSTON
UNIVERSITY

Look Into Individual Table – OrderItem

Customer - data view and schema view

Customer

🔑	Id	int identity
	FirstName	nvarchar(40)
	LastName	nvarchar(40)
	City	nvarchar(40)
	Country	nvarchar(40)
	Phone	nvarchar(20)

Id	FirstName	LastName	City	Country	Phone
1	Maria	Anders	Berlin	Germany	030-0074321
2	Ana	Trujillo	México D.F.	Mexico	(5); 555-4729
3	Antonio	Moreno	México D.F.	Mexico	(5); 555-3932
4	Thomas	Hardy	London	UK	(171); 555-7788
5	Christina	Berglund	Luleå	Sweden	0921-12 34 65

OrderHead - *data view and schema view*

OrderHead	
Id	int identity
OrderDate	datetime
OrderNumber	nvarchar(10)
CustomerId	int
TotalAmount	decimal(12, 2)

Id	OrderDate	OrderNumber	CustomerId	TotalAmount
1	Jul 4 2012 12:00:00:000AM	542378	85	440
2	Jul 5 2012 12:00:00:000AM	542379	79	1863.4
3	Jul 8 2012 12:00:00:000AM	542380	34	1813
4	Jul 8 2012 12:00:00:000AM	542381	84	670.8
5	Jul 9 2012 12:00:00:000AM	542382	76	3730

OrderItem - *data view and schema view*

OrderItem

Id	int identity
OrderId	int
ProductId	int
UnitPrice	decimal(12, 2)
Quantity	int

Id	OrderId	ProductId	UnitPrice	Quantity
1	1	11	14	12
2	1	42	9.8	10
3	1	72	34.8	5
4	2	14	18.6	9
5	2	51	42.4	40

BOSTON
UNIVERSITY

Product - *data view and schema view*

Product

🔑	Id	int identity
	ProductName	nvarchar(50)
	SupplierId	int
	UnitPrice	decimal(12, 2)
	Package	nvarchar(30)
	IsDiscontinued	bit

Id	ProductName	SupplierId	UnitPrice	Package	IsDiscontinued
1	Chai	1	18	10 boxes x 20 bags	0
2	Chang	1	19	24 - 12 oz bottles	0
3	Aniseed Syrup	1	10	12 - 550 ml bottles	0
4	Chef Anton's Cajun Seasoning	2	22	48 - 6 oz jars	0
5	Chef Anton's Gumbo Mix	2	21.35	36 boxes	1

Supplier - *data view and schema view*

	Supplier	
	Id	int identity
	CompanyName	nvarchar(40)
	ContactName	nvarchar(50)
	ContactTitle	nvarchar(40)
	City	nvarchar(40)
	Country	nvarchar(40)
	Phone	nvarchar(30)
	Fax	nvarchar(30)

Id	CompanyName	ContactName	ContactTitle	City	Country	Phone	Fax
24	G'day, Mate	Wendy Mackenzie	NULL	Sydney	Australia	(02); 555-5914	(02); 555-4873
25	Ma Maison	Jean-Guy Lauzon	NULL	Montréal	Canada	(514); 555-9022	NULL
26	Pasta Buttini s.r.l.	Giovanni Giudici	NULL	Salerno	Italy	(089); 6547665	(089); 6547667
27	Escargots Nouveaux	Marie Delamare	NULL	Montceau	France	85.57.00.07	NULL
28	Gai pâturage	Eliane Noz	NULL	Annecy	France	38.76.98.06	38.76.98.58

Tutorial Setup

Option 1: Use BU Common App - https://rdweb.wvd.microsoft.com/arm/webclient/index.html
Other options: install local copy

All the tutorial files can be accessed from:
http://rcs.bu.edu/examples/db/tutorials/intro_to_SQL/

Download Tutorial materials as one single package (After download, please unzip the above zip file before you move forward !!):
http://rcs.bu.edu/examples/db/tutorials/intro_to_SQL/intro_to_SQL.zip

Tutorial Setup Instructions are at the 'presentation/' subdirectory:
presentation/instr_tutorialSetup_local.docx   (install on PC/Windows)
presentation/instr_tutorialSetup_local_mac.docx   (install on Apple Mac OS 10.12+)
presentation/instr_tutorialSetup_vdi.docx   (Connect to BU Common Apps via BU Login)

DB Browser software tool : tutfiles/software/
Tutorial Sample DB : tutfiles/db/sample_ecomm.db
Demo SQL script: tutfiles/sql/ecomm_demo.sql

Some Extra Info

The following pages are some extra information you may be interested in

BOSTON UNIVERSITY

Point #1: GUI tool is not the only way!

A GUI tool like DB Browser is not the only way to access databases!

There could be many other ways! The following are the two ways:

SQLite Programming Interface - Python

```
[yshen16@scc-wi2 codesnippet]$ module load python3/3.8.10
[yshen16@scc-wi2 codesnippet]$ ipython
Python 3.8.10 (default, May  3 2021, 17:15:02)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.23.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import sqlite3
   ...: ecomm = sqlite3.connect('../db/sample_ecomm.db')
   ...: c = ecomm.cursor()
   ...: id = ('10',)
   ...: c.execute('SELECT * FROM customer WHERE id=?', id)
   ...: print(c.fetchone())
   ...: ecomm.close()
(10, 'Elizabeth', 'Lincoln', 'Tsawassen', 'Canada', '(604); 555-4729')

In [2]: exit()
[yshen16@scc-wi2 codesnippet]$ 
```

SQLite Programming Interface - Python

- Just run as normal python script:

```
[yshen16@scc-wi2 codesnippet]$ python sample_ecomm_python.py
(10, 'Elizabeth', 'Lincoln', 'Tsawassen', 'Canada', '(604); 555-4729')
[yshen16@scc-wi2 codesnippet]$
```

SQLite Programming Interface - R

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> #load library
> library(RSQLite)
>
> # create connection
> ecomm <- dbConnect(RSQLite::SQLite(), "sample_ecomm.db")
>
> # query data
> result <- dbGetQuery(ecomm, "SELECT * FROM customer WHERE id=?", params=c(10))
> result
  Id FirstName LastName      City Country              Phone
1 10 Elizabeth  Lincoln Tsawassen  Canada (604); 555-4729
>
> # disconnect db
> dbDisconnect(ecomm)
> q()
```

Point #2: How to Choose Database Tool

- 1. understand the differences among Database Management Systems (DBMSs)
- 2. Analyze Data
  - a. writing DB/reading DB
  - b. frequency
  - c. application domain (real time/transactional) ?
- 3. Budget and Cost
  - a. initial cost
  - b. maintainence

BOSTON UNIVERSITY

41

Useful Resources:

- This tutorial materials:

  http://rcs.bu.edu/examples/db/tutorials/intro2SQL/

- W3Schools SQL tutorial: https://www.w3schools.com/sql/

- Khan Academy:

  https://www.khanacademy.org/computing/computer-programming/sql

BOSTON UNIVERSITY

42

Thank You !!

Please don't forget to spend some time to give me some feedback at

http://rcs.bu.edu/eval

BOSTON UNIVERSITY

43