

# Introduction to T-SQL Queries




Kathi Kellenberger

Redgate Software



# Kathi Kellenberger

**Editor and Advocate**  
**Redgate Software**

-  /Kathikellenberger
-  @aunkathi
- <http://aunkathisql.com>
-  Kathi.Kellenberger@red-gate.com

- Lifelong learner
- Teacher

- 
- Co-leader of Data Platform Women in Technology Group
  - Data Platform MVP

# Agenda

- Class 1
  - Module 1: Introduction
  - Module 2: Simple select statements
  - Module 3: Filtering
- Class 2
  - Module 4: Expressions
  - Module 5: Joining
- Class 3
  - Module 5: Joining (Continued)
  - Module 6: Grouping
- Class 4
  - Module 7: Subqueries
  - Module 8: UNION

# CLASS MATERIALS

- <https://github.com/KathiKellenberger/CoderGirlDataAnalysis>
  - Slides
  - Demos
  - Labs
- Students should install Azure Data Studio and connect to
  - [sqlprojects.com](https://sqlprojects.com),3500
  - Student
  - Madison18\*
  - Instructions will be given in class

# Module 1: Introduction

# What's a database?

## Database

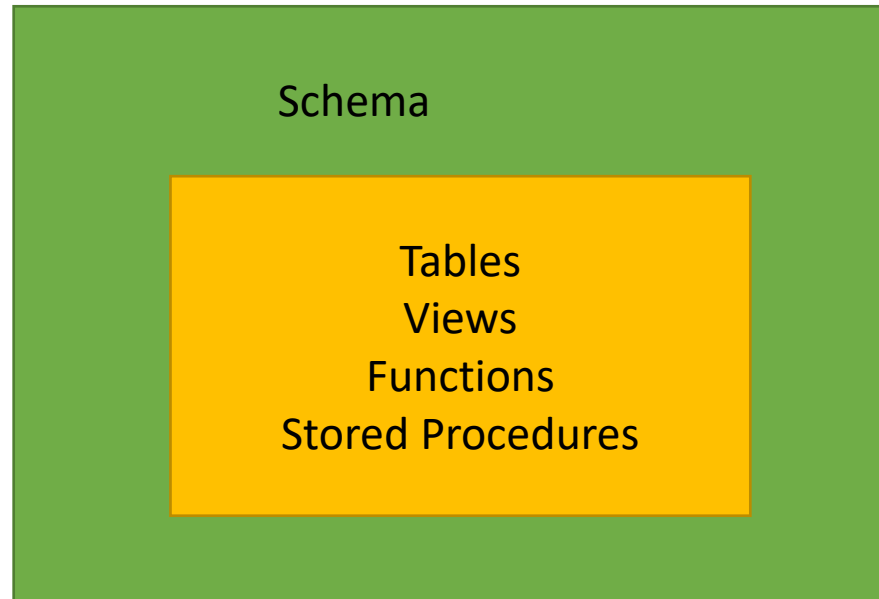
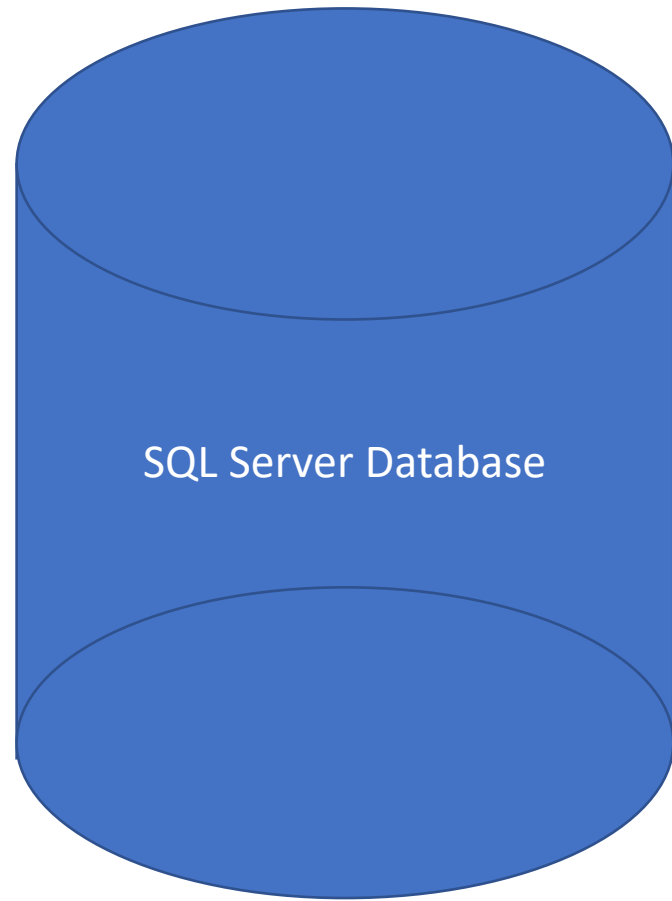


```
target_date | target_time |
-----+-----+
2016-12-26 | 02:29:30 |
2016-12-26 | 02:32:29 |
2016-12-26 | 02:32:29 |
2016-12-26 | 02:32:29 |
2016-12-26 | 02:35:29 |
2016-12-26 | 02:35:29 |
2016-12-26 | 02:38:29 |
2016-12-26 | 02:38:29 |
2016-12-26 | 02:41:30 |
2016-12-26 | 02:41:30 |
2016-12-26 | 02:44:29 |
2016-12-26 | 02:44:29 |
```

A database is an organized collection of data. It is the collection of schemas, tables, queries, reports, views and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information, such as modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

Database - Wikipedia

<https://en.wikipedia.org/wiki/Database>



# Tables

UsedCars					
ID	Make	Model	Type	Year	Color
1	Chevrolet	Malibu	Passenger car	2015	Blue
2	Hyundai	Sonata	Passenger car	2011	Silver
3	Chrysler	Pacifica	Minivan	2017	White
4	Toyota	Prius	Hybrid car	2013	White
5	Hyundai	Elantra	Passenger car	2015	Blue
6	Chevrolet	Silverado	Truck	2013	Red

```
SELECT *  
FROM UsedCars  
WHERE Make = 'Hyundai';
```



# T-SQL

- SQL = Structured Query Language
- T-SQL = Transact SQL
- Each vendor has own version
- The basics are the same

# PRINT, GO, USE, and comments

- Print displays a message
- GO is a batch separator
- USE – switch databases (will not work with our Azure dbs)
- -- (two dashes) for a one-line comment
- /\* \*/ for multi line comments
- Use a tick mark aka single quote around strings or dates
- Use a semi-colon at end of statements

# Get started

- Connection
  - sqlprojects.com,3500
  - Student with pw Madison18\*
  - SQL Authentication
- You can continue to use the databases after today.

Connection Details	
Connection type	Microsoft SQL Server
Server	sqlprojects.com,3500
Authentication type	SQL Login
User name	student
Password	.....
	<input checked="" type="checkbox"/> Remember password
Database	<Default>
Server group	<Default>
Name (optional)	

# Demo 1: Getting around in Azure Data Studio

# Lab

- Complete Module 1 Lab 1
- Start back up at

# Module 2:

## Simple SELECT statements

# SELECT

- Keyword for retrieving data from a database
- Return a list of columns or expressions
- Syntax

```
SELECT <expr1>[,<expr2>,<expr3>,...]
```

# FROM

- The table where the data can be found
- Syntax

```
SELECT *  
FROM <schema>.<table>
```

```
SELECT <expr1>[,<expr2>,<expr3>,...]  
FROM <schema>.<table>
```

- The schema is often “dbo”
- You join tables together in the FROM clause, but you’ll learn about that in a later module



# Aliases

- Give a name to an expression or table
- Syntax

```
SELECT <expr1> AS Name1
```

```
FROM <tablename> AS tbl
```

```
SELECT <expr1> AS [The name]
```

```
SELECT <expr1> AS "The name"
```

# TOP

- Return a number of rows or a percent of rows
- Syntax

```
SELECT TOP(n) <expr1>[,<expr2>,<expr3>,...]  
FROM <schema>.<table>
```

```
SELECT TOP(n) PERCENT <expr1>[,<expr2>,<expr3>,...]  
FROM <schema>.<table>
```

# DISTINCT

- Return a unique set of rows

- Syntax

```
SELECT DISTINCT <expr1>[,<expr2>,<expr3>,...]  
FROM <schema>.<table>
```

Demo: SELECT FROM

# Lab

- Complete Module 2 Lab 1
- Start at 9:50
- 13 minutes
- In lab info, label the parts to make it easier

# Ordering data

- Use the ORDER BY clause
- One or more columns or expressions
- Ascending by default
- Use DESC to reverse order

Demo: ORDER BY

# Lab

- Complete Module 2 Lab 2



# Module 3: Filtering

# WHERE

- Basic Syntax

```
SELECT <expr1>[,<expr2>,<expr3>,...]
```

```
FROM <schema>.<table>
```

```
WHERE <expr5> = <expr6>
```

```
ORDER BY <expr1>
```

- Dates example

```
SELECT SalesOrderID, ShipDate
```

```
FROM Sales.SalesOrderHeader
```

```
WHERE ShipDate >= '2011-06-07' and ShipDate < '2011-06-08'
```

# Operators

- =, <>, !=
- <, >, <=, >=
- BETWEEN
- LIKE (with wildcards %, \_ and more)
- IN
- AND, OR for multiple expressions
- NOT
- Parentheses to enforce logic

Demo: The WHERE clause

# Lab

- Complete Module 3 Lab 1

# Working with NULL

- Unknown
- Can't compare anything to NULL
- When trying to compare to NULL, the row is not returned
- Use ISNULL or COALESCE to replace the NULL
- Use IS NULL or IS NOT NULL to compare

Demo: NULL

# Lab

- Complete Module 3 Lab 2