

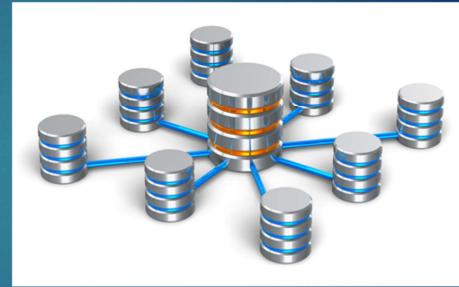
Logols Learning

WEEKEND WEB DEVELOPMENT BOOT CAMP

TRAINING: SQL

MySql

- ▶ Database
- ▶ Tables



- MySql has an open source and a proprietary enterprise version. It is owned by Oracle.
- It has competitors in this realm such as Oracle and Sql Server
- The first thing that you normally do is to create the database.
- There can be multiple databases on a server.
- The server will have security and settings around it and the database itself will as well.
- Tables are created to store data in the database.

MySQL Workbench

- Let's take a tour of MySQL Workbench
 - Create a database using the MySQL client.
 - Connect to the database.
 - Create a query window using the File menu selecting the New Query Tab option.
 - Save the Script using the File menu selecting the Save Script As option.
 - Open the Script using the File menu, select the Open SQL Script option.
 - Create a table using the create table script.
 - View tables within the Navigator.
 - Right click a table and select Send to Sql Editor and then Create Statement.
 - The create statement for the table will be available.

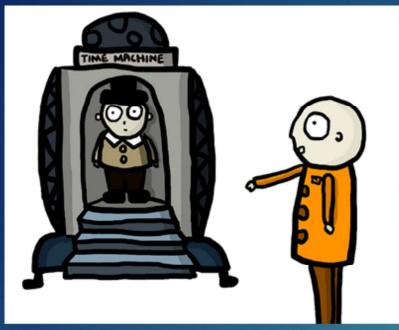
Select Statements

- ▶ Components in Logical Order
- ▶ SELECT
- ▶ FROM
- ▶ WHERE
- ▶ GROUP BY
- ▶ HAVING
- ▶ ORDER BY

▶ Example:

```
SELECT FirstName,  
LastName  
FROM dbo.Person  
WHERE personId = 2
```

- These are the components of a select statement
- The statement starts with the word SELECT then you list the columns you would like to select
- Then you list the word FROM and the name of the table that you want to select from
- Then you list the word WHERE and list all of the condition predicates that you want
- Group By can be used if you want to do an aggregate query
- Having are condition predicates for the aggregate
- Finally you can list the word Order By followed by columns that you want to order by

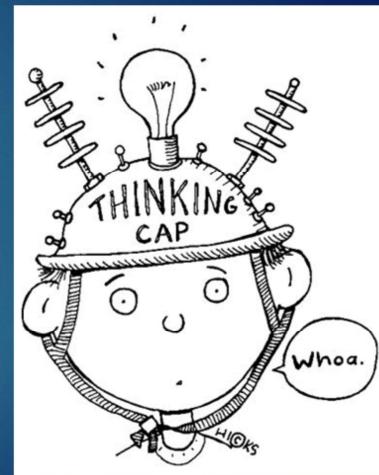


EXAMPLE

SELECT QUERIES

ASSESSMENT

SELECT QUERIES



- Write on the board a select statement that selects FirstName and LastName columns of a table named employee.
- Write on the board a select statement that select FirstName and LastName columns of a table named employee where the EmployeeID equals 5.
- Write on the board a select statement that select FirstName and LastName columns of a table named employee where the FirstName equals Joe and LastName equals Smith.

Insert Statements

► Components

- INSERT
- INTO
- VALUES

► Example:

```
INSERT INTO
dbo.Person
(FirstName, LastName)
VALUES ('Joe',
'Mackie')
```

- These are the components of an insert statement
- The statement starts with the words INSERT INTO then you list the table name followed by columns in parentheses
- Then you list the word VALUES followed by values in parentheses

Update Statements

► Components

► UPDATE

► SET

► WHERE

► Example:

```
UPDATE dbo.Person  
SET FirstName = 'Joe'  
WHERE LastName =  
'Mackie'
```

- These are the components of an update statement
- The statement starts with the word UPDATE then you list the table name
- Then you list the word SET and set the column name = to what you want to update
- Then you list the word WHERE and list all of the condition predicates that you want

Delete Statements

- ▶ Components

- ▶ DELETE

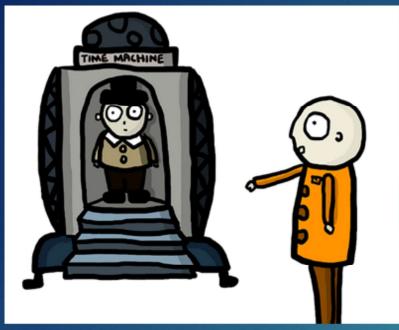
- ▶ FROM

- ▶ WHERE

- ▶ Example:

```
DELETE  
FROM dbo.Person  
WHERE LastName =  
'Mackie'
```

- These are the components of a delete statement
- The statement starts with the words DELETE FROM then you list the table name
- Then you list the word WHERE and list all of the condition predicates that you want



EXAMPLE

INSERT, UPDATE, AND DELETE QUERIES

ASSESSMENT

INSERT, UPDATE, AND DELETE QUERIES



- Write on the board an insert statement for a table named employee with columns EmployeeID, FirstName, and LastName. EmployeeID is auto incrementing.
- Write on the board an insert statement for a table named employee with columns EmployeeID, FirstName, and LastName. EmployeeID is auto incrementing. Insert multiple records.
- Write on the board an update statement that updates FirstName equal to Tim and LastName equal to Grant columns of a table named employee where the FirstName equals Joe and LastName equals Smith.
- Write on the board a delete statement that deletes records of a table named employee where the EmployeeID equals 5.

MySQL Data Types

- ▶ int, smallint, tinyint, bigint
- ▶ bit
- ▶ numeric, decimal
- ▶ date, datetime
- ▶ char, text, varchar

- These are data types for MySQL. These are associated with each column and are specified for variables.
- bigint, smallint, tinyint are different sizes of integers meaning whole numbers.
- bit is a Boolean represented by 1 meaning true and 0 meaning false
- numeric and decimal are synonyms both are given a fixed precision and scale
- Then there are different date and time fields. date stores the date only. Datetime stores both the date and time
- There are a few options for storing text. char is for text with a known number of characters. varchar is for variable amount of text with a limit set. text is deprecated so don't use this.

Create Table

- ▶ Components

- ▶ CREATE TABLE

- ▶ (

- ▶ Column1 data type,
 - ▶ Column2 data type

- ▶);

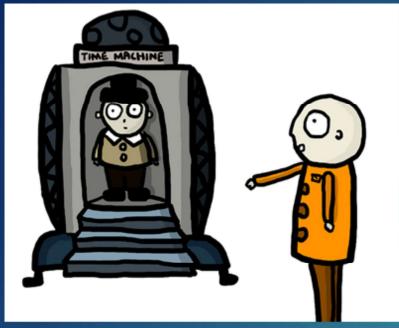
```
CREATE TABLE People
```

```
(
```

```
PersonID int,  
LastName varchar(255),  
FirstName varchar(255),  
Address varchar(255),  
City varchar(255)
```

```
);
```

- This is the basic syntax for creating tables.
- We are currently ignoring details such as primary and foreign keys.
- Use the Create table syntax and specify columns separated by commas surrounded by curly braces.



EXAMPLE

DATABASE AND TABLES

- Let's go through our assessment application and create some tables

https://www.w3schools.com/sql/trysql.asp?filename=trysql_create_table

```
CREATE TABLE People
(
PersonID int,
LastName varchar(255),
FirstName varchar(255),
Address varchar(255),
City varchar(255)
);
```

ASSESSMENT

CREATE TABLES



- Name a MySql data type for a number with no decimals
- Name a MySql data type for a number with decimals
- Name a MySql data type for variable strings
- Write on the board a create statement that creates a table named Employee with an EmployeeID, FirstName, and LastName column with an auto incrementing primary key for the EmployeeID.

Assignment

- ▶ Create a table named Person with the following columns: PersonId, FirstName, LastName, and PersonStatusId.
- ▶ Create a table named PersonStatus with the following columns: PersonStatusId and StatusDescription.



Assignment

- ▶ Insert the following statuses into the PersonStatus table:
 - ▶ 1: Alive, 2: Zombie, 3: Dead, 4: Unknown
- ▶ Insert people into the Person table.



Assignment

- ▶ Perform a select of all people.
- ▶ Select all people that have an unknown status.
- ▶ Select all people that are alive or have an unknown status.
- ▶ Select all people that are alive and have first name Tom.



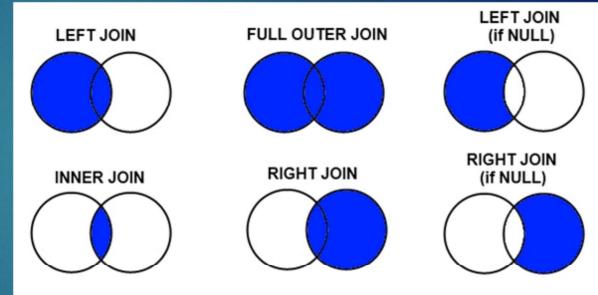
Assignment

- ▶ Update the status to Zombie for a given person based upon their name that currently has a status of Alive.
- ▶ Delete every person that is dead.



Join Types

- ▶ Inner Join
- ▶ Left Outer Join
- ▶ Right Outer Join



- There are several ANSI join types. These include cross join, inner join, left outer join, right outer join, and full outer join.
- Each of these join types produce different results when you join tables together.
- We will go through only Inner, Left Outer, and Right Outer Joins

Inner Join

- ▶ Every row from the first table will be matched with every row from the second table based upon the on conditions specified.

```
SELECT c.ClassId, c.ClassName, t.TeacherId,  
t.FirstName, t.LastName  
FROM dbo.Class c  
INNER JOIN dbo.Teacher t  
ON c.TeacherId = t.TeacherId
```

- The inner join is maybe the most common join you will use in a relational database.
- Similar to the cross join, it matches each row from the first table to every table in the second, but only based upon the condition specified.
- So, in the example you see we have added the on condition.
- In this instance we are joining on the TeacherId of the first table to the TeacherId of the second table.
- So, the results will only return when the TeacherId in the first table is in the second table and they match.

Left Outer Join

- ▶ Every row from the first table will be returned and results from the second table will be included only if the condition matches.

```
SELECT c.ClassId, c.ClassName, t.TeacherId,  
t.FirstName, t.LastName  
FROM dbo.Class c  
LEFT OUTER JOIN dbo.Teacher t  
ON c.TeacherId = t.TeacherId
```

- The outer joins are all related to each other. We are going to start with the Left Outer Join.
- With left outer join, every record from the first table is returned no matter what.
- Results will be returned from the second table if there is a match based upon the condition.
- If there are multiple results in the second table that match, then a row will be returned for each.
- If there is not a match in the second table then nulls will be returned for each column reference to the second table.
- The syntax is very similar to the inner join, just replace inner with left outer.

Right Outer Join

- ▶ Every row from the second table will be returned and results from the first table will be included only if the condition matches.

```
SELECT c.ClassId, c.ClassName, t.TeacherId,  
t.FirstName, t.LastName  
FROM dbo.Class c  
RIGHT OUTER JOIN dbo.Teacher t  
ON c.TeacherId = t.TeacherId
```

- The Right Outer Join is the opposite of the Left Outer Join.
- With right outer join, every record from the second table is returned no matter what.
- Results will be returned from the first table if there is a match based upon the condition.
- If there are multiple results in the first table that match, then a row will be returned for each.
- If there is not a match in the first table then nulls will be returned for each column reference to the first table.
- The syntax is very similar to the left outer join, just replace left with right.

Table Alias

- ▶ Short name that can be given to a table
- ▶ Example:

```
SELECT c.ClassId, c.ClassName, t.TeacherId,  
t.FirstName, t.LastName  
FROM dbo.Class c  
INNER JOIN dbo.Teacher t  
ON c.TeacherId = t.TeacherId
```

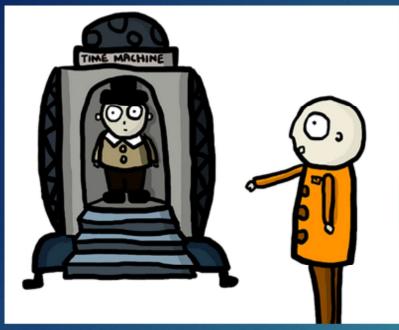
- You may have noticed that we were not using the full name of the table to identify where columns were coming from
- This is because we have placed an alias right after the table.
- This alias can then be used to reference that table
- If we were not using an alias you could write out the full name of the table, but this would take more time.

Column Alias

- ▶ Short name that can be given to a column
- ▶ Example:

```
SELECT c.ClassId, c.ClassName, t.TeacherId,  
t.FirstName AS TeacherFirstName, t.LastName AS  
TeacherLastName  
FROM dbo.Class c  
INNER JOIN dbo.Teacher t  
ON c.TeacherId = t.TeacherId
```

- Column alias' allow you to give a more descriptive name of a result
- Use the AS syntax followed by the alias that you want to provide
- In this example we are specifying the t.FirstName as TeacherFirstName for clarification.
- There could be tables with the same column names and alias' allow us to return both in the same result and name them something different.



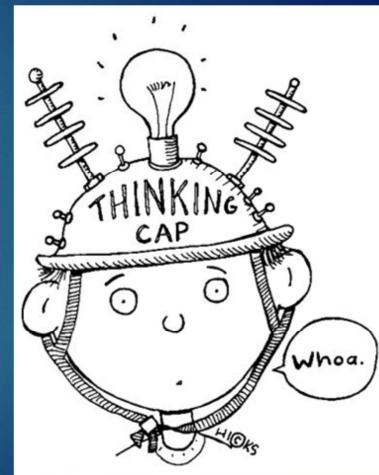
EXAMPLE

JOINS

- So far we have worked with sql that manipulated only a single table at a time.
- There are many cases with relational databases where we need to use or join multiple tables together at once.

ASSESSMENT

JOINS



- Name three joins types
- Write on the board a select statement with an inner join between an Employee table and a Team table. Both tables include a TeamID column.
- Write on the board a select statement with a left outer join between an Employee table and a Team table. Both tables include a TeamID column.
- Write on the board a select statement with a right outer join between an Employee table and a Team table. Both tables include a TeamID column.

Assignment

- ▶ Select all statuses joined with people to get a full list of everyone and their status.
- ▶ Change the select so that it shows all statuses regardless of whether the status is currently associated with any person.



What is an ORM?

- ▶ Object Relational Mapper
- ▶ Mismatch between Object Model and Relational Model
- ▶ ORM converts between the two
- ▶ Provides Query and Persistence Capability

- ORM stands for object relational mapper.
- There is often a mismatch between the relational database model that we create and the object model that we want to work with in code.
- Because of this there is some work for us to translate the data coming back from tables into objects.
- There are a number of design patterns thought up to deal with this scenario.
- An ORM usually implements some of those design patterns to take care of most of the heavy lifting of the mapping.
- They also often provide a way to query and persist the data.

Entity Framework

- ▶ Implementation of an ORM
- ▶ Created by Microsoft
- ▶ Linq Syntax used for Queries
- ▶ Data Model First vs. Code First

- Microsoft's recommended ORM implementation is Entity Framework.
- It uses Linq syntax for querying the data.
- There is a way to use Entity Framework by writing code first and another way for setting up the data model first.
- We are going to focus on creating the data model first.

Dapper

- ▶ Micro ORM
- ▶ Performs only mapping and nothing else
- ▶ Fast
- ▶ Why are we using it?
 - ▶ Allows practice with database queries
 - ▶ Visibility into database and application interaction

- Dapper is considered a Micro ORM
- All it does is map data to an object of a class.
- It's much faster than entity framework because it's such a thin layer
- We are using it for a few reasons:
 - I don't want to completely abstract away the database.
 - I want you to be able to practice queries to the database with sql.
 - I want you to be able to see the actual interaction between the database and the application.

Base Repository

```
Private string connectionString =  
    @"server=localhost;port=3306;database=Assessment;user=[user];  
    password=[password]";  
  
public IDbConnection Connection  
{  
    get  
    {  
        return new MySqlConnection(connectionString);  
    }  
}
```

- We can setup a base repository for other repositories to use the same connection.

Repository

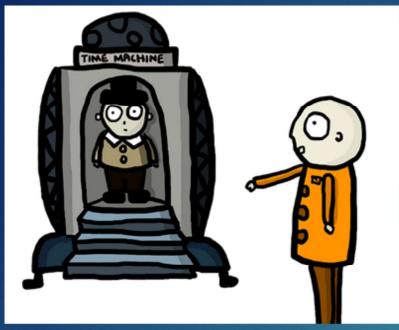
```
public Subject Get(int subjectID)
{
    using ( IDbConnection dbConnection = Connection)
    {
        dbConnection.Open();
        return dbConnection.Query<Subject>("Select SubjectId, Name,
Description From Subject Where SubjectId = @SubjectId", new {
SubjectId = subjectID }, commandType:
CommandType.Text).FirstOrDefault();
    }
}
```

- We can setup methods in the repository to retrieve data using Dapper.

CLI Commands

- ▶ mkdir – Create Directory
- ▶ cd – Change Directory
- ▶ Add project:
 - ▶ dotnet new classlib
 - ▶ dotnet new webapi
- ▶ Add reference:
 - ▶ dotnet add reference [path]/[name.csproj]
 - ▶ dotnet add package Dapper
 - ▶ dotnet add package MySql.Data

- In order to make projects for our main components we will need to use CLI commands.
- mkdir creates a new directory on your computer.
- cd changes the directory that you are currently in.
- dotnet new classlib – creates a new class library project. This is what we use to create a new .Net project that is code only.
- dotnet new webapi – creates a new web api project. This is what we use to create a new .Net project that will contain our web api.
- dotnet add reference – allows us to reference code from another project. One project does not know about the other until a reference is added.
- dotnet add package – allows us to add a package to our code. A package is created by someone and can be downloaded over the internet for use in your project.
- We will be using two packages from the Nuget package manager which are Dapper and MySql.Data.

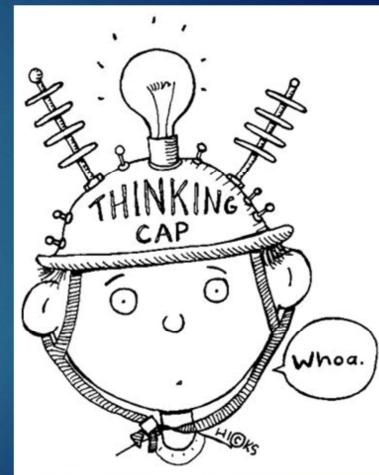


EXAMPLE

DAPPER

ASSESSMENT

DAPPER



- What is the pattern used to map relational database tables to classes?
- What does ORM stand for?
- Name two ORM options.

Assignment

- ▶ A status report is needed of all government employees. Statuses are:
 - ▶ 1: Alive, 2: Zombie, 3: Dead, 4: Unknown
- ▶ Retrieve the data from the tables we created with a join.
- ▶ Loop through each record to display the status of each person in the database.



QUICK REVIEW

SQL



- Write on the board a select statement that selects FirstName and LastName columns of a table named employee.
- Write on the board a select statement that select FirstName and LastName columns of a table named employee where the EmployeeID equals 5.
- Write on the board a select statement that select FirstName and LastName columns of a table named employee where the FirstName equals Joe and LastName equals Smith.
- Write on the board an insert statement for a table named employee with columns EmployeeID, FirstName, and LastName. EmployeeID is auto incrementing.
- Write on the board an insert statement for a table named employee with columns EmployeeID, FirstName, and LastName. EmployeeID is auto incrementing. Insert multiple records.
- Write on the board an update statement that updates FirstName equal to Tim and LastName equal to Grant columns of a table named employee where the FirstName equals Joe and LastName equals Smith.
- Write on the board a delete statement that deletes records of a table named employee where the EmployeeID equals 5.

- Write on the board a create statement that creates a table named Employee with an EmployeeID, FirstName, and LastName column with an auto incrementing primary key for the EmployeeID.
- Write on the board a select statement with an inner join between an Employee table and a Team table. Both tables include a TeamID column.
- Write on the board a select statement with a left outer join between an Employee table and a Team table. Both tables include a TeamID column.
- Write on the board a select statement with a right outer join between an Employee table and a Team table. Both tables include a TeamID column.

Additional Resources

- ▶ Code Katas
 - ▶ <https://www.codewars.com/>
- ▶ UDacity
 - ▶ <https://www.udacity.com/course/intro-to-relational-databases--ud197>
- ▶ MySql
 - ▶ <https://www.mysql.com/>
- ▶ Sql Bolt
 - ▶ <https://sqlbolt.com>

Keep Practicing!

- ▶ Try creating more tables.
- ▶ Try different selects, inserts, deletes, and updates.
- ▶ Try different joins.
- ▶ Try connecting your new tables to an application using Dapper.