[**基于Dapper的分页实现，支持筛选，排序，结果集总数，多表查询，非存储过程**](https://www.cnblogs.com/jinweijie/p/dapper_pagination.html)

**简介**

之前事先搜索了下博客园上关于Dapper分页的实现，有是有，但要么是基于存储过程，要么支持分页，而不支持排序，或者搜索条件不是那么容易维护。

**代码**

首先先上代码： <https://github.com/jinweijie/Dapper.PagingSample>

**方法定义**

以下是我的一个分页的实现，虽然不是泛型（因为考虑到where条件以及sql语句的搭配），但是应该可以算是比较通用的了，方法定义如下：

public Tuple<IEnumerable<Log>, int> Find(LogSearchCriteria criteria

, int pageIndex

, int pageSize

, string[] asc

, string[] desc);

以上函数定义是一个查询Log的示例，返回结果中，Tuple的第一个值是结果集，第二个值是总行数（例如，总共有100条记录，每页10条，当前第一页，那么第一个值是10条记录，第二个值是100）

在示例项目中，我用两种方法实现了分页：

1. 第一种是基于2此查询，第一次得到总数，第二次查询得到结果集。

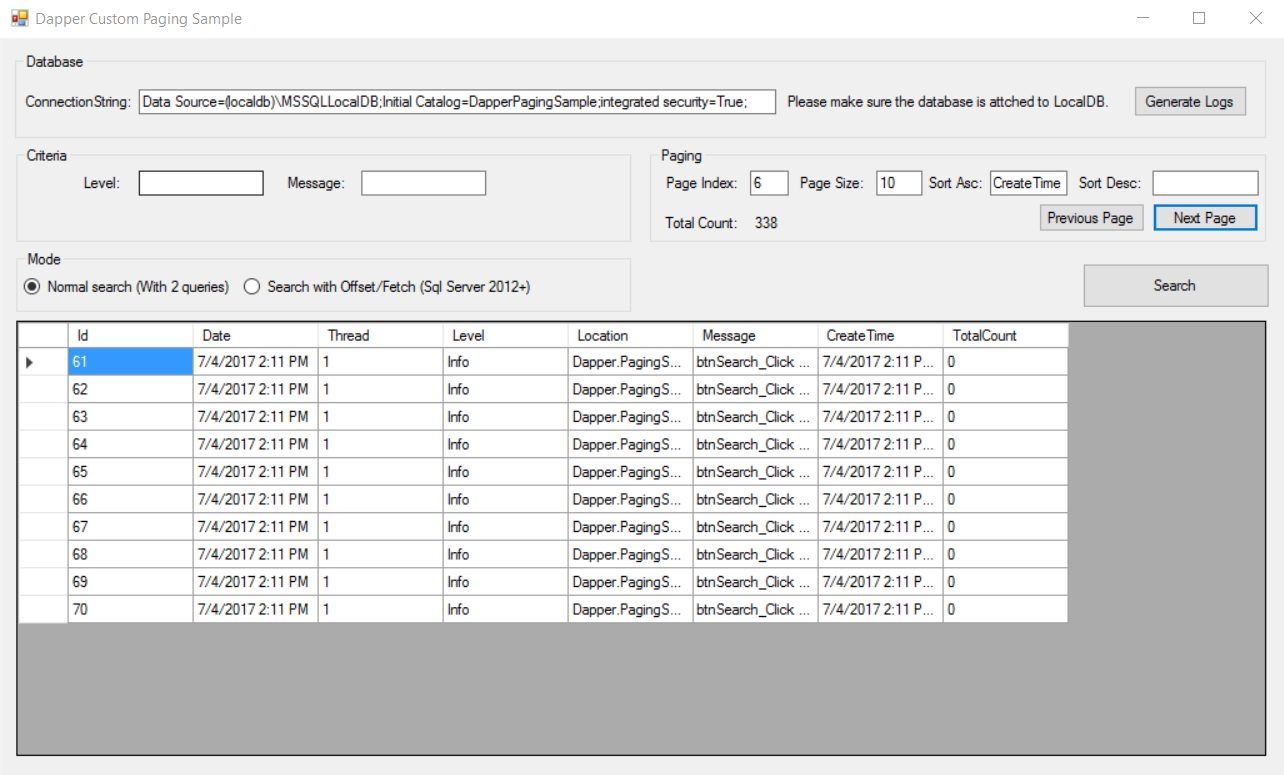
2. 第二种是基于1此查询，用了SqlServer 的Offest/Fetch，所以只支持Sql Server 2012+，所以大家根据自己用的Sql Server版本选择不同的实现，这里当然是第二种实现效率更高一点。

**运行示例**

1. 将Github的Repo下载或者Clone到本地以后，到Database目录下，解压缩Database.7z

2. Attach到Sql Server上。默认我使用Sql Server LocalDB，连接字符串是 Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=DapperPagingSample;integrated security=True;   如果你用的不是LocalDB，请酌情修改App.Config的连接字符串。

3. Ctrl+F5运行程序，示例项目里，我用了一个简单的WinForm程序，但应该可以比较好的演示分页效果。



**多表支持**

增加了示例，支持多表查询，例如有两个Log表，Level表，Log的LevelId字段引用Level的Id字段，通过以下的查询，可以实现多表查询的分页，排序，过滤：

首先是通过两次查询的示例（基本支持所有版本Sql Server）：

[复制代码](javascript:void(0);)

1 public Tuple<IEnumerable<Log>, int> Find(LogSearchCriteria criteria

2 , int pageIndex

3 , int pageSize

4 , string[] asc

5 , string[] desc)

6 {

7 using (IDbConnection connection = base.OpenConnection())

8 {

9 const string countQuery = @"SELECT COUNT(1)

10 FROM [Log] l

11 INNER JOIN [Level] lv ON l.LevelId = lv.Id

12 /\*\*where\*\*/";

13

14 const string selectQuery = @" SELECT \*

15 FROM ( SELECT ROW\_NUMBER() OVER ( /\*\*orderby\*\*/ ) AS RowNum, l.\*, lv.Name as [Level]

16 FROM [Log] l

17 INNER JOIN [Level] lv ON l.LevelId = lv.Id

18 /\*\*where\*\*/

19 ) AS RowConstrainedResult

20 WHERE RowNum >= (@PageIndex \* @PageSize + 1 )

21 AND RowNum <= (@PageIndex + 1) \* @PageSize

22 ORDER BY RowNum";

23

24 SqlBuilder builder = new SqlBuilder();

25

26 var count = builder.AddTemplate(countQuery);

27 var selector = builder.AddTemplate(selectQuery, new { PageIndex = pageIndex, PageSize = pageSize });

28

29 if (!string.IsNullOrEmpty(criteria.Level))

30 builder.Where("lv.Name= @Level", new { Level = criteria.Level });

31

32 if (!string.IsNullOrEmpty(criteria.Message))

33 {

34 var msg = "%" + criteria.Message + "%";

35 builder.Where("l.Message Like @Message", new { Message = msg });

36 }

37

38 foreach (var a in asc)

39 {

40 if(!string.IsNullOrWhiteSpace(a))

41 builder.OrderBy(a);

42 }

43

44 foreach (var d in desc)

45 {

46 if (!string.IsNullOrWhiteSpace(d))

47 builder.OrderBy(d + " desc");

48 }

49

50 var totalCount = connection.Query<int>(count.RawSql, count.Parameters).Single();

51 var rows = connection.Query<Log>(selector.RawSql, selector.Parameters);

52

53 return new Tuple<IEnumerable<Log>, int>(rows, totalCount);

54 }

55 }

[复制代码](javascript:void(0);)

第二个示例是通过Offset/Fetch查询（支持Sql Server 2012+）

[复制代码](javascript:void(0);)

1 public Tuple<IEnumerable<Log>, int> FindWithOffsetFetch(LogSearchCriteria criteria

2 , int pageIndex

3 , int pageSize

4 , string[] asc

5 , string[] desc)

6 {

7 using (IDbConnection connection = base.OpenConnection())

8 {

9

10 const string selectQuery = @" ;WITH \_data AS (

11 SELECT l.\*, lv.Name AS [Level]

12 FROM [Log] l

13 INNER JOIN [Level] lv ON l.LevelId = lv.Id

14 /\*\*where\*\*/

15 ),

16 \_count AS (

17 SELECT COUNT(1) AS TotalCount FROM \_data

18 )

19 SELECT \* FROM \_data CROSS APPLY \_count /\*\*orderby\*\*/ OFFSET @PageIndex \* @PageSize ROWS FETCH NEXT @PageSize ROWS ONLY";

20

21 SqlBuilder builder = new SqlBuilder();

22

23 var selector = builder.AddTemplate(selectQuery, new { PageIndex = pageIndex, PageSize = pageSize });

24

25 if (!string.IsNullOrEmpty(criteria.Level))

26 builder.Where("lv.Name = @Level", new { Level = criteria.Level });

27

28 if (!string.IsNullOrEmpty(criteria.Message))

29 {

30 var msg = "%" + criteria.Message + "%";

31 builder.Where("l.Message Like @Message", new { Message = msg });

32 }

33

34 foreach (var a in asc)

35 {

36 if (!string.IsNullOrWhiteSpace(a))

37 builder.OrderBy(a);

38 }

39

40 foreach (var d in desc)

41 {

42 if (!string.IsNullOrWhiteSpace(d))

43 builder.OrderBy(d + " desc");

44 }

45

46 var rows = connection.Query<Log>(selector.RawSql, selector.Parameters).ToList();

47

48 if(rows.Count == 0)

49 return new Tuple<IEnumerable<Log>, int>(rows, 0);

50

51

52 return new Tuple<IEnumerable<Log>, int>(rows, rows[0].TotalCount);

53

54 }

55 }

[复制代码](javascript:void(0);)

**谢谢**

希望对大家有帮助:)

最后，我更新了本篇随便，增加了内容，希望不要再被撤下了（上次撤下说是因为篇幅太短。。。），因为个人觉得这个对大家应该还是会有用的。