FA設備技術勉強会

第10回 22年9月17日

データベース プログラム入門

トムライン FX
CHALLENGE WITH PASSION !!

自己紹介

Twitter

- 名前：tomlinefx
- 仕事：エンジニア/投資家
- 理想：吉田松陰
  「志」「共に学ぶ」「知行合一」
- 好き：新技術・お酒・旅行・ネコ
- 得意：装置制御、画像処理、統計

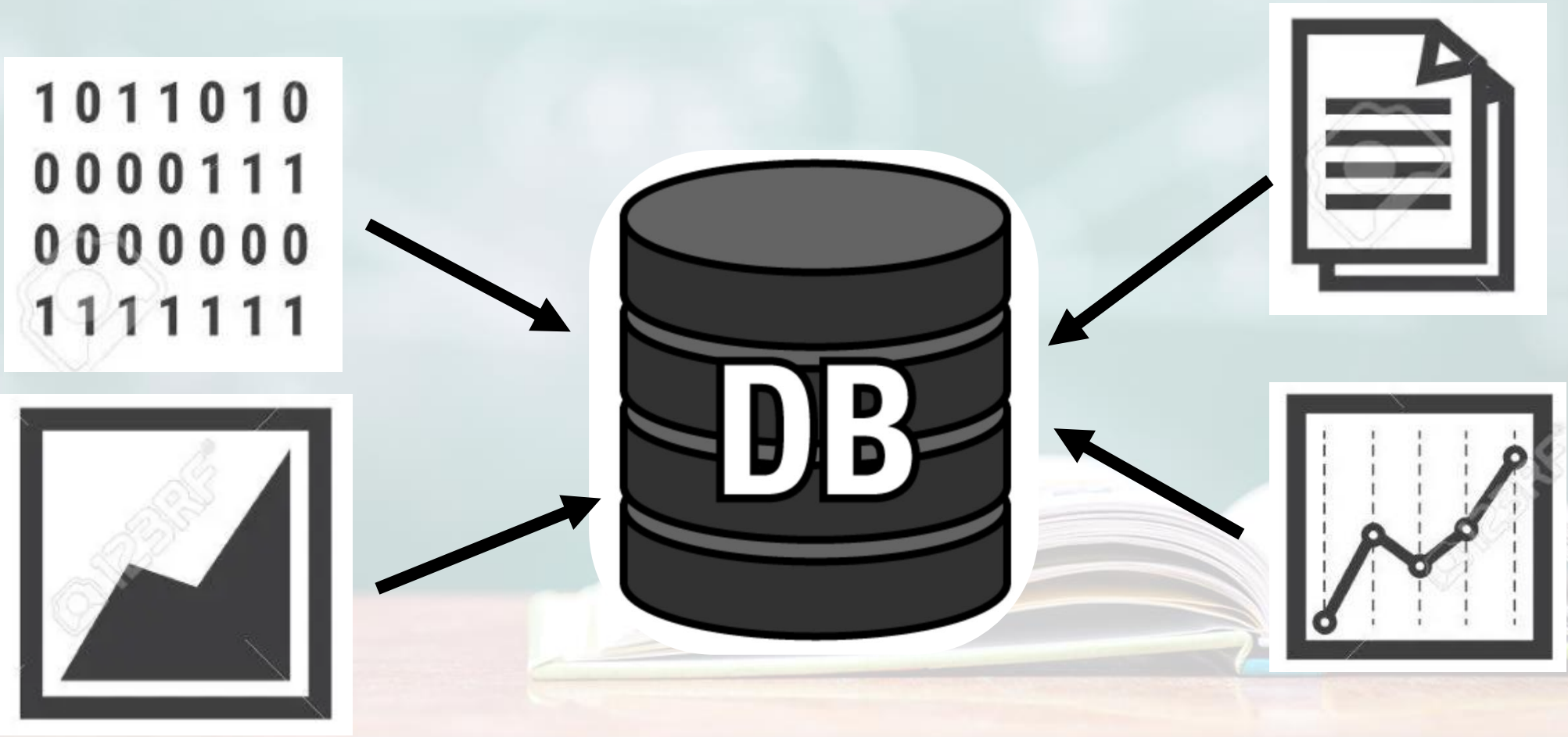
FA設備技術勉強会

No	日付	テーマ
第6回	210903	IEC61131-3 PLC制御
第7回	211205	外観検査装置の光学設計
第8回	220305	PlantUMLﾄﾞｷｭﾒﾝﾄ管理
第9回	220618	**最新FW Webアプリ開発**

DataBase概要

[DB (DataBase) とは]

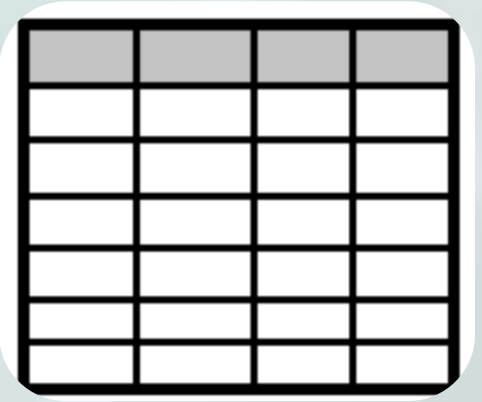
・データを入れる箱



[DBMS（Management Systemu）とは]

- DBを扱うための道具
- テーブル生成/データ抽出
- DBサーバー側ソフト

[RDB（RelationalDB）とは]



- DBの種類の１つ
  ※階層型/ネットワーク型/NoSQL型
- データ表形式で管理
- Relation=表同士の関連付け
  →整理されたエクセル表！

[RDBMS種類]

- **SQL Server：MicroSoft社**
- OracleDataBase：Oracle社
- DB2：IBM社
- PostgreSQL：オープン
- MySQL：オープン／Oracle

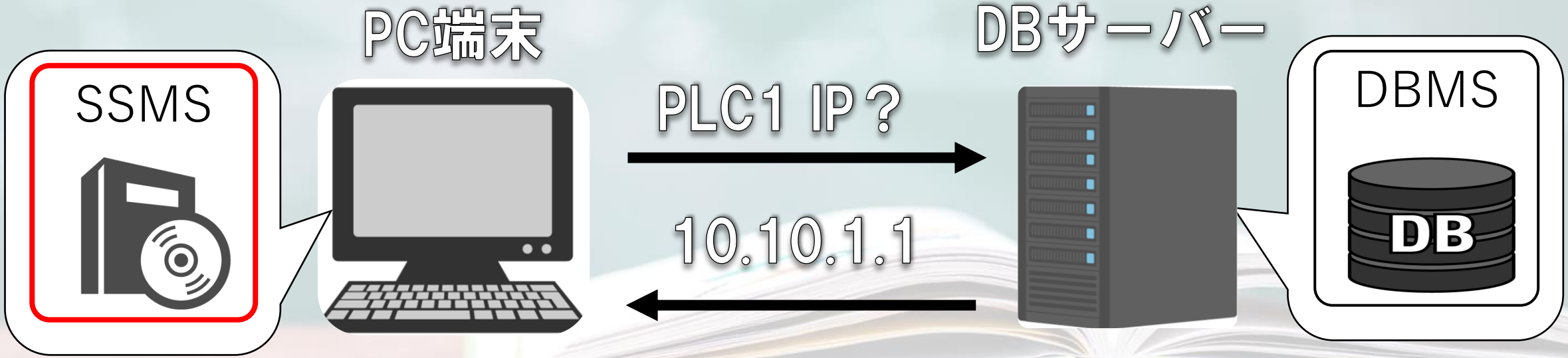
[RDB開発環境]

MicroSoft社ソフト2つ紹介

- SSMS（SQL Server Managemet Studio）
  ※SQL DB統合開発環境
- Visual Studio
  ※アプリ開発環境+SSMS機能

[SQL（Structured Query Language）とは]

- DB管理/操作の専用言語
- SSMSにて操作



モデル仕様
前回Webアプリ

[モデルケース]

- 設備100台の運用中
- 品質条件：温度/湿度/圧力異常
- 設備指示計値を毎日手書き記録

［問題］

設備データアナログ管理

※現場手書/エクセル手入力

［課題］

設備管理システム内製化

※生産技術主導で全部やる！

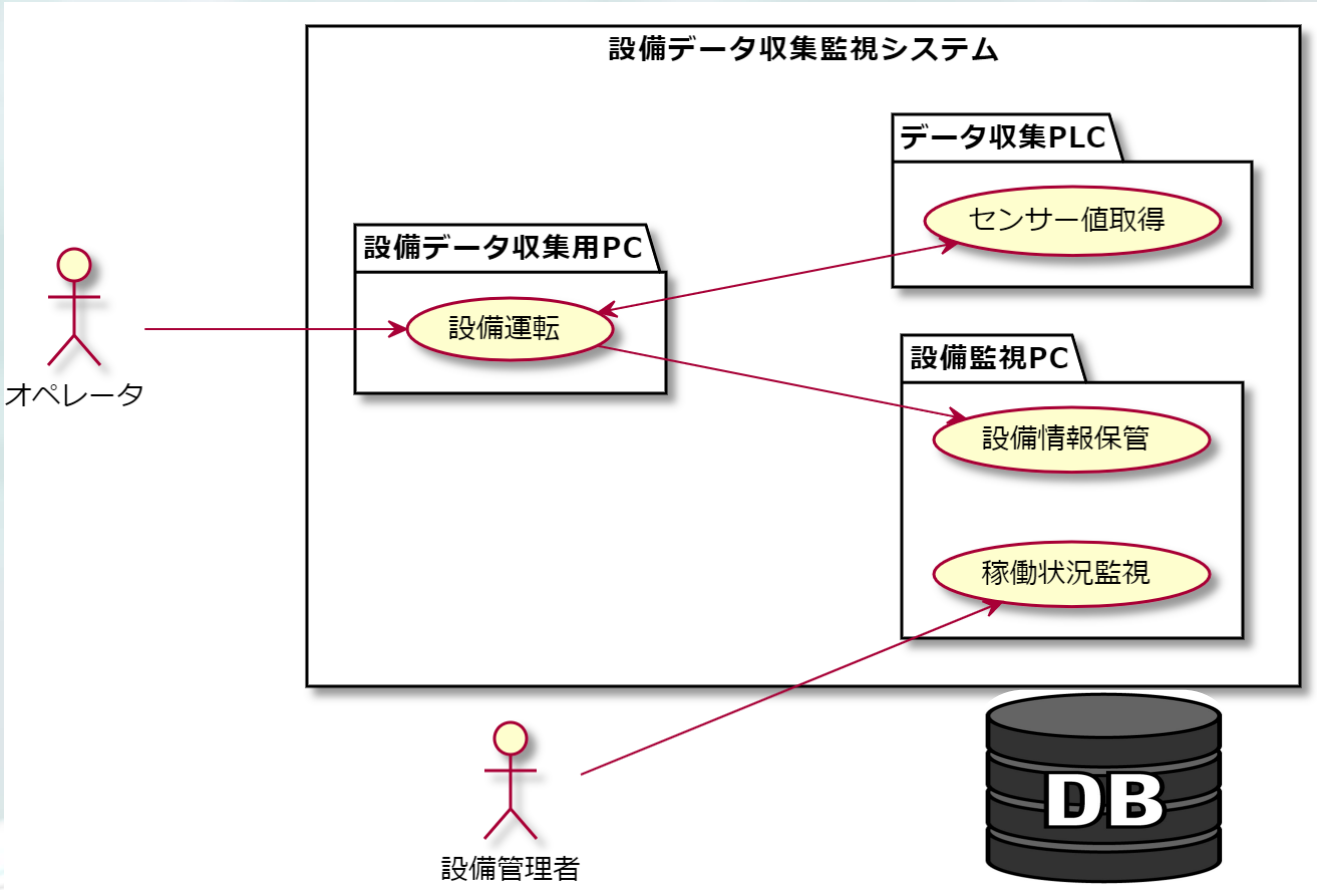
※リスキリング→自ら道を創る！

[システム要件]

- 各種工程設備100台 PLC制御
- 対象項目：温度/湿度/圧力
- 稼働状況リアルタイム監視
  ※設備前確認→遠隔確認/監視
- 異常設備抽出
  ※温度/湿度/圧力閾値NG

[システム概要]

- PC2台
  - ①データ収集
  - ②運用/保守
- PLC100台
  - ※国内主要メーカ



[システム開発環境]

- RDBMS：SQL Server2019
  ※Express版10GBまで無料
- アプリ：Visual Studio2019
  ※開発言語 C# / Express版無料
  ※ [WebApp] Blazor @.NET5
  ※ [NativeApp] Form @.NET4.72

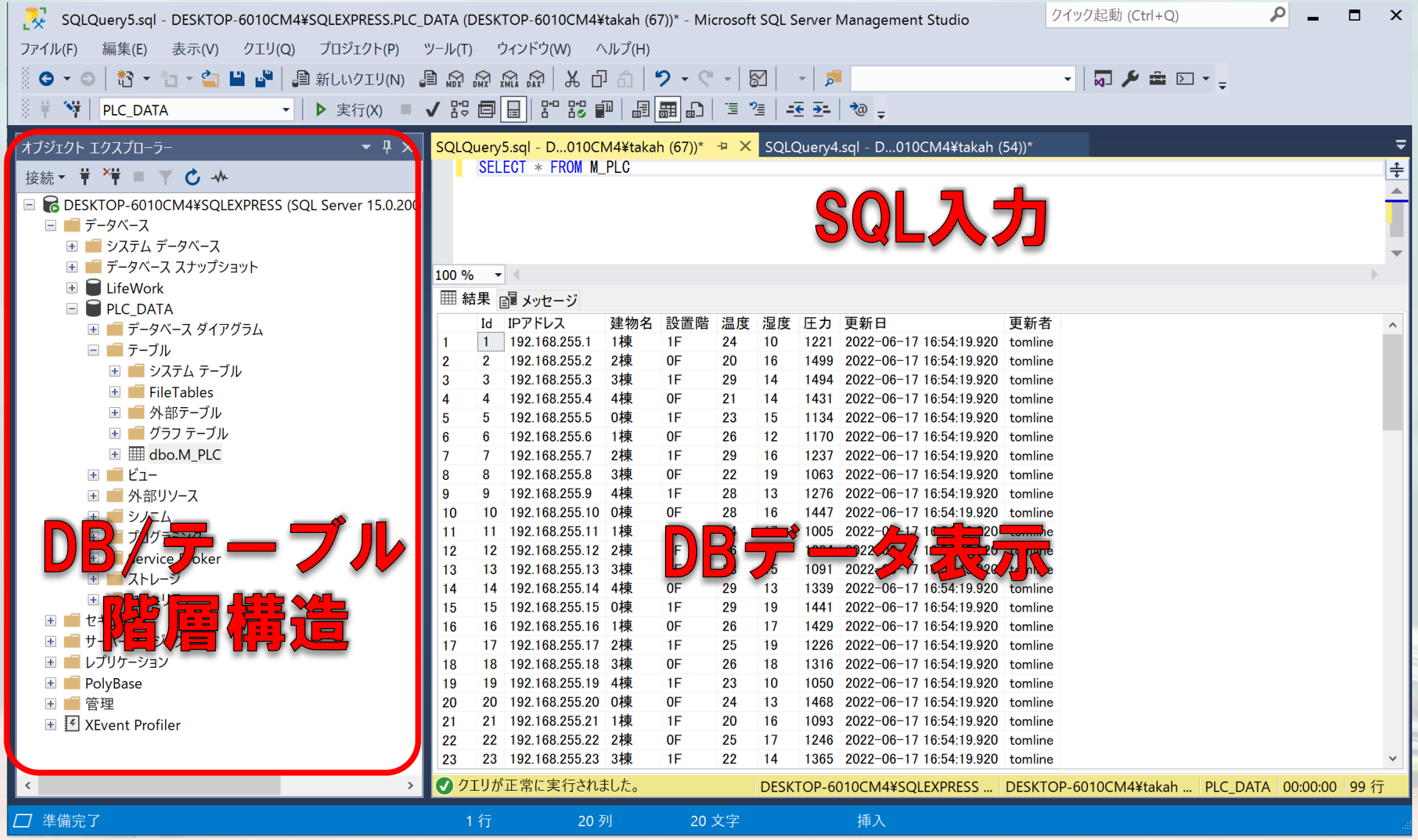
DB設計＆SQL基礎

[DBテーブル設計]

No	項目	型	レコード例
1	制御盤番号	int	1
2	IPアドレス	nvchar(64)	10.10.1.1
3	建物名	nvchar(64)	A棟
4	設置階	nvchar(64)	1F
5	温度	real	25.2[℃]
6	湿度	real	13.8[%]
7	圧力	real	500[kPA]
8	更新日	datetime	2022/9/17 19:00:00

ER図

[SSMS紹介]



	Id	IPアドレス	建物名	設置階	温度	湿度	圧力	更新日	更新者
1	1	192.168.255.1	1棟	1F	24	10	1221	2022-06-17 16:54:19.920	tomline
2	2	192.168.255.2	2棟	0F	20	16	1499	2022-06-17 16:54:19.920	tomline
3	3	192.168.255.3	3棟	1F	29	14	1494	2022-06-17 16:54:19.920	tomline
4	4	192.168.255.4	4棟	0F	21	14	1431	2022-06-17 16:54:19.920	tomline
5	5	192.168.255.5	0棟	1F	23	15	1134	2022-06-17 16:54:19.920	tomline
6	6	192.168.255.6	1棟	0F	26	12	1170	2022-06-17 16:54:19.920	tomline
7	7	192.168.255.7	2棟	1F	29	16	1237	2022-06-17 16:54:19.920	tomline
8	8	192.168.255.8	3棟	0F	22	19	1063	2022-06-17 16:54:19.920	tomline
9	9	192.168.255.9	4棟	1F	28	13	1276	2022-06-17 16:54:19.920	tomline
10	10	192.168.255.10	0棟	0F	28	16	1447	2022-06-17 16:54:19.920	tomline
11	11	192.168.255.11	1棟				1005	2022-06-17 16:54:19.920	tomline
12	12	192.168.255.12	2棟						
13	13	192.168.255.13	3棟				1091		
14	14	192.168.255.14	4棟	0F	29	13	1339	2022-06-17 16:54:19.920	tomline
15	15	192.168.255.15	0棟	1F	29	19	1441	2022-06-17 16:54:19.920	tomline
16	16	192.168.255.16	1棟	0F	26	17	1429	2022-06-17 16:54:19.920	tomline
17	17	192.168.255.17	2棟	1F	25	19	1226	2022-06-17 16:54:19.920	tomline
18	18	192.168.255.18	3棟	0F	26	18	1316	2022-06-17 16:54:19.920	tomline
19	19	192.168.255.19	4棟	1F	23	10	1050	2022-06-17 16:54:19.920	tomline
20	20	192.168.255.20	0棟	0F	24	13	1468	2022-06-17 16:54:19.920	tomline
21	21	192.168.255.21	1棟	1F	20	16	1093	2022-06-17 16:54:19.920	tomline
22	22	192.168.255.22	2棟	0F	25	17	1246	2022-06-17 16:54:19.920	tomline
23	23	192.168.255.23	3棟	1F	22	14	1365	2022-06-17 16:54:19.920	tomline

[テーブル生成]

内容：RDBテーブル生成

SQL： CREATE

書式：

CREATE テーブル名(
カラム名 型 属性,
・・・・
)

サンプル

```
CREATE TABLE M_PLC(
    Id int not null,
    IPアドレス nvarchar(20) not null,
    建物名 nvarchar(10) not null,
    設置階 nvarchar(10) not null,
    温度 real not null,
    湿度 real not null,
    圧力 real not null,
    更新日 datetime not null,
    更新者 nvarchar(10) not null,
    primary key(Id),
)
```

[データ出力]

内容：DBからデータ出力

SQL：SELECT

書式：

SELECT カラム名
FROM DB名
SELECT ＊ FROM DB名

サンプル

```
SELECT
M_PLC.IPアドレス,
M_PLC.圧力,
M_PLC.温度
FROM M_PLC
```

［データ抽出］

内容：DBからデータ条件付き出力

SQL：WHERE

書式：

SELECT カラム名

FROM DB名

WHERE 条件式

サンプル

```
SELECT *
FROM [PLC_DATA].[dbo].[M_PLC]
WHERE dbo.M_PLC.温度 > 25
```

［データ並び替え］

内容：DBからデータ並び替え出力

SQL：ORDER BY（ASC）

書式：

SELECT ＊

FROM DB名

ORDER BY カラム名

※降順 ASC追加

サンプル

```
SELECT TOP (10) *
FROM [PLC_DATA].[dbo].[M_PLC]
WHERE dbo.M_PLC.温度 > 25
ORDER BY dbo.M_PLC.温度 ASC
```

［データ並び替え］

内容：DBからデータ並び替え出力

SQL：ORDER BY（ASC）

書式：

SELECT ＊

FROM DB名

ORDER BY カラム名

※降順 ASC追加

サンプル

```
SELECT TOP (10) *
FROM [PLC_DATA].[dbo].[M_PLC]
WHERE dbo.M_PLC.温度 > 25
ORDER BY dbo.M_PLC.温度 ASC
```

Visual Studio
プログラム紹介

[C#言語 SQL命令]

```
public static void CreateTable()
{
    var buf = new StringBuilder();
    buf.Append("CREATE TABLE M_PLC(");
    buf.Append("    Id int not null,");
    buf.Append("    IPアドレス nvarchar(20) not null,");
    buf.Append("    建物名 nvarchar(10) not null,");
    buf.Append("    設置階 nvarchar(10) not null,");
    buf.Append("    温度 real not null,");
    buf.Append("    湿度 real not null,");
    buf.Append("    圧力 real not null,");
    buf.Append("    更新日 datetime not null,");
    buf.Append("    更新者 nvarchar(10) not null,");
    buf.Append("    primary key(Id),");
    buf.Append(")");

    Function.Action($"{buf}");
}
```

```
public static class Function
{
    2 個の参照
    public static bool Action(string query)
    {
        var result = false;
        using (var con = DBConnection.CreateIbstance())
        {
            try
            {
                var com = con.GetSqlCommand();
                com.CommandText = query;
                result = com.ExecuteNonQuery().Equals(1);
                result = true;
            }
            catch(Exception ex)
            {

            }
        }

        return result;
    }
}
```

[C#言語 専用クエリ]

```
private void On初期表示()
{
    // 表示内容リセット
    using (var db = new Models.PLC_DATAContext())
    {
        MPlcs = db.MPlcs.ToList();
        currentCount = MPlcs.Count();
    }
}

private void UpdateTable()
{
    using (var db = new Models.PLC_DATAContext())
    {
        MPlcs = db.MPlcs.ToList();
        currentCount = MPlcs.Count();
    }
}
```

```
#region DBレコード抽出クエリ
private void On温度25以上()
{
    using (var db = new Models.PLC_DATAContext())
    {
        MPlcs = db.MPlcs.Where(x=>x.温度 >= 25).OrderByDescending(x => x.温度).ToList();
        StateHasChanged();
        currentCount = MPlcs.Count();
    }
}

private void On湿度15未満()
{
    using (var db = new Models.PLC_DATAContext())
    {
        MPlcs = db.MPlcs.Where(x => x.湿度 < 15).OrderBy(x => x.湿度).ToList();
        StateHasChanged();
        currentCount = MPlcs.Count();
    }
}

private void On圧力1300以上()
{
    using (var db = new Models.PLC_DATAContext())
    {
        MPlcs = db.MPlcs.Where(x => x.圧力 >= 1300).Take(15).ToList();
        StateHasChanged();
        currentCount = MPlcs.Count();
    }
}
#endregion
```

[VisualStudio環境メリット]

- アプリ開発とDB管理統合
  ※SSMS機能統合
- GitHubによる履歴管理
- 最新技術にて開発効率改善
  ※EntityFramework/Blazor/WebAPI

SQL Server DEMO

まとめ

- RDBデータ管理便利
- SQLにてDB操作簡単
- SQL Server2019無料枠有

↓

皆さんも是非お試し下さい！

ご清聴ありがとうございました！

世界一の生産技術

ご質問・ご依頼・ご要望はこちら↓

tomlinefx@gmai.com

Blog

推奨書籍

①DB設計

［目的］

②SQL