

Table des Temps

Création et Utilisation d'une Table des Temps



Dans beaucoup de cas de programmation, il est nécessaire d'avoir à disposition une table de référence pour tous les jours de l'année et différentes informations concernant cette journée.

Lors de développement d'applications liées à la gestion du temps ou en rapport avec les Ressources Humaines, il est très courant d'avoir recours à une table de référence traitant des dates avec des informations diverses intégrées. Ainsi savoir qu'à une date donnée on est en saison Hivernale ou Estivale peut aider à comprendre plus précisément les raisons d'une augmentation des absences (dans le cas d'une application de gestion des absences). On peut ainsi prévoir plus finement la disponibilité de l'ensemble du personnel durant une période donnée.

D'autres cas peuvent justifier ce besoin, comme par exemple une application de gestion de commandes de produits saisonniers (un grand classique est la consommation de produit de rafraîchissement l'été).

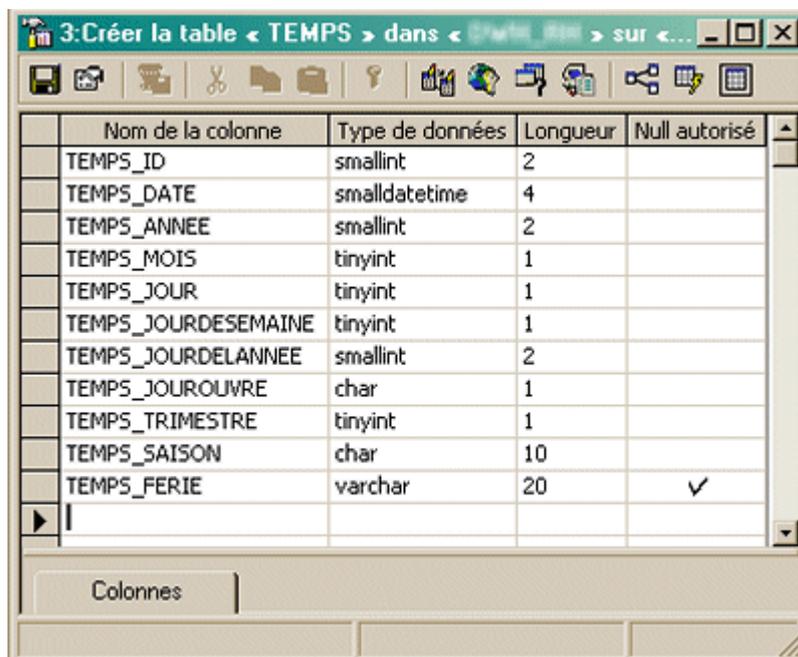
Ainsi, j'ai créé un script SQL qui permet de générer une table et une procédure stockée associée destinée à alimenter cette table. Ce script est spécifique à **SQL Server**, mais peut aisément être adapté pour un autre SGBD (supportant les procédures Stockées) comme ORACLE, ACCESS ou PostgreSQL.

Le cas de MySQL est spécifique, car seule la version 5 supportera les procédures stockées, ainsi dans le cas des versions précédentes il faudra adapter le script SQL et exécuter directement celui-ci pour alimenter la table.

La Table des Temps

Cette Table a pour but de nous stocker différentes informations en plus de la simple Date. On trouvera donc bien sur le mois ou l'année, mais aussi la saison ou encore le Numéro du Trimestre.

Voyons le schéma de la Table ci-dessous :



Nom de la colonne	Type de données	Longueur	Null autorisé
TEMPS_ID	smallint	2	
TEMPS_DATE	smalldatetime	4	
TEMPS_ANNEE	smallint	2	
TEMPS_MOIS	tinyint	1	
TEMPS_JOUR	tinyint	1	
TEMPS_JOURDESEMAINE	tinyint	1	
TEMPS_JOURDELANNEE	smallint	2	
TEMPS_JOUROUVRE	char	1	
TEMPS_TRIMESTRE	tinyint	1	
TEMPS_SAISON	char	10	
TEMPS_FERIE	varchar	20	✓

Ainsi la liste des Champs qui seront remplis lors de l'alimentation de la Table est la suivante :

- **TEMPS_ID** (Type : smallInt, Propriété : NOT NULL) : ID de la Date en cours
- **TEMPS_DATE** (Type : smalldatetime, Propriété : NOT NULL) : Date en cours
- **TEMPS_ANNEE** (Type : smallInt, Propriété : NOT NULL) : Année de la Date en cours
- **TEMPS_MOIS** (Type : tinyInt, Propriété : NOT NULL) : Mois de la Date en cours
- **TEMPS_JOUR** (Type : tinyInt, Propriété : NOT NULL) : Jour de la Date en cours
- **TEMPS_JOURDESEMAINE** (Type : tinyInt, Propriété : NOT NULL) : Numéro du jour de la Semaine de la Date en cours
- **TEMPS_JOURDELANNEE** (Type : smallInt, Propriété : NOT NULL) : Numéro du jour de l'Année de la Date en cours
- **TEMPS_JOUROUVRE** (Type : char(1), Propriété : NOT NULL) : Flag pour savoir si la Date en cours est ouverte ou non
- **TEMPS_TRIMESTRE** (Type : tinyInt, Propriété : NOT NULL) : Numéro du Trimestre de la Date en cours
- **TEMPS_SAISON** (Type : char(10), Propriété : NOT NULL) : Saison de la Date en cours
- **TEMPS_FERIE** (Type : varchar(20), Propriété : NULL) : Si la Date en cours est Férié, Quel est-il ?

Tous ces champs sont des données très basiques concernant une date donnée, mais on pourrait très bien rajouter d'autres informations comme les Ephémérides ou encore d'autres plus spécifiques.

Pour les Ephémérides, une petite base se trouve ici :

- [Script d'utilisation des Ephémérides en ASP sur Base ACCESS \(FR\)](#)

Maintenant que nous avons la table des Temps, voyons la Procédure Stockée qui permettra de l'alimenter.

La Procédure Stockée

Une fois cette table faite, il reste à l'alimenter afin de lui fournir les données nécessaires pour travailler avec cette table. Ainsi, on peut exécuter un script SQL directement qui se chargera de la charger ou alors créer une Procédure Stockée qui s'occupera de cette tâche et que l'on aura plus qu'à exécuter lorsque l'on veut.

J'ai choisi la seconde solution pour mon cas précis, je vais donc vous fournir le script et expliquer la méthode de chargement utilisée.

Dans cette procédure stockée, il faut pouvoir définir la saison en cours à la date donnée, la précédente méthode était d'utiliser les dates 'standard' de changement de saison, mais dans certains cas, il faut pouvoir être très précis. Ainsi à cette fin, Frédéric BROUARD (bien connu dans le monde SQL Server) m'a fourni des fonctions SQL Server permettant d'obtenir ces dates avec précision.

Il s'agit de 4 fonctions faisant appel à une cinquième sous fonction, vous avez le script de création de ces fonctions à l'adresse suivante :

- [CALCUL DES SAISONS - basé sur l'algorithme de Meeus \(FR\)](#)

De plus je vous invite (si vous ne le connaissez pas déjà) à consulter le site dédié au SQL de Frédéric Brouard :

- [Site web consacré au langage SQL \(FR\)](#)

Ainsi je fixe tout d'abord 2 bornes, la date de début et celle de fin, afin de pouvoir recharger la table à tout moment avec des nouveaux éléments (bien sur en la vidant au préalable).

Ensuite le chargement se fait avec une simple boucle While et pour chaque date, je vais tester les différentes propriétés afin de les stocker en mémoire avant l'insertion de l'item (la saison, le jour de la semaine, si le jour est ouvert ou non, ...).

Enfin je vais chercher à savoir si la date en question est un jour férié et dans ce cas lequel est-ce.

Pour les jours fériés, j'ai paramétré les jours fériés génériques (1er Janvier, 25 Décembre, ...) et ensuite une liste de jours fériés plus spécifiques pour lesquelles la règle ne m'est pas connue (jour de l'ascension, Lundi de Paques, ...). J'ai enregistré dans la procédure ces jours spécifiques que de 2000 à 2010.

```
-----  
-- CREATION DE LA PROCEDURE STOCKEE D'ALIMENTATION DE LA TABLE DES TEMPS
```

```
CREATE PROCEDURE dbo.TEMPS_ChargeTableTemps
```

```
AS
```

```
DECLARE @TEMPS_DATE    smalldatetime,  
        @TEMPS_FIN      smalldatetime,  
        @TEMPS_ID       smallInt,  
        @TEMPS_JOUROUVRE char(1),  
        @TEMPS_JOURDESEMAINE tinyInt,  
        @TEMPS_TRIMESTRE tinyInt,  
        @TEMPS_ANNEE    char(4),  
        @TEMPS_MOIS     smallInt,  
        @TEMPS_JOUR     smallInt,  
        @TEMPS_SAISON   char(10),  
        @TEMPS_FERIE    varchar(20),  
        @PRINTEMPS      smalldatetime,  
        @ETE             smalldatetime,  
        @AUTOMNE        smalldatetime,  
        @HIVER          smalldatetime
```

```
-- Initialisation du premier Jour de la semaine (1=Lundi)  
SET DATEFIRST 1
```

```
-- Initialisation du format de la date (Année MOis Jour)  
SET DATEFORMAT ymd
```

```

-- Initialisation des Variables (Premiere valeur de l'ID, Date de début et date de fin)
SET @TEMPS_DATE = '2000-01-01'
SET @TEMPS_FIN = '2010-12-31'
SET @TEMPS_ID = 1000

-- Début du Chargement
WHILE @TEMPS_DATE < @TEMPS_FIN
BEGIN

    BEGIN TRANSACTION
    -- Creation et Chargement des variables
    SET @TEMPS_ANNEE = CAST(YEAR(@TEMPS_DATE) AS char(4))
    SET @TEMPS_MOIS = MONTH(@TEMPS_DATE)
    SET @TEMPS_JOUR = DAY(@TEMPS_DATE)
    SET @TEMPS_JOURDESEMAINE = datepart(weekday,@TEMPS_DATE)

    -- Chargement du Week-End
    IF @TEMPS_JOURDESEMAINE IN (6,7)
        SET @TEMPS_JOUROUVRE = 'N'
    ELSE
        SET @TEMPS_JOUROUVRE = 'O'

    -- Chargement du Trimestre
    IF MONTH(@TEMPS_DATE) IN (6,7,8)
        SET @TEMPS_TRIMESTRE = 1
    ELSE IF MONTH(@TEMPS_DATE) IN (9,10,11)
        SET @TEMPS_TRIMESTRE = 2
    ELSE IF MONTH(@TEMPS_DATE) IN (12,1,2)
        SET @TEMPS_TRIMESTRE = 3
    ELSE IF MONTH(@TEMPS_DATE) IN (3,4,5)
        SET @TEMPS_TRIMESTRE = 4

/*
-- Chargement des Dates de début des Saisons : Version Simple ne cherchant pas la précision des dates
SET @PRINTEMPS = CAST(@TEMPS_ANNEE+'-03-21' AS smalldatetime)
SET @ETE = CAST(@TEMPS_ANNEE+'-06-21' AS smalldatetime)
SET @AUTOMNE = CAST(@TEMPS_ANNEE+'-09-21' AS smalldatetime)
SET @HIVER = CAST(@TEMPS_ANNEE+'-12-21' AS smalldatetime)
*/
-- Chargement des Dates de début des Saisons : Version Précise fournis par Frédéric BROUARD (MVP SQL Server 2004)
SET @PRINTEMPS = dbo.FN_SPRING_DATE(@TEMPS_ANNEE)
SET @ETE = dbo.FN_SUMMER_DATE(@TEMPS_ANNEE)
SET @AUTOMNE = dbo.FN_AUTUMN_DATE(@TEMPS_ANNEE)
SET @HIVER = dbo.FN_WINTER_DATE(@TEMPS_ANNEE)

/*
PRINT 'PRINTEMPS :'+ CAST(@PRINTEMPS AS varchar(100))
PRINT 'ETE :'+ CAST(@ETE AS varchar(100))
PRINT 'AUTOMNE :'+ CAST(@AUTOMNE AS varchar(100))
PRINT 'HIVER :'+ CAST(@HIVER AS varchar(100))
*/
*/
IF @TEMPS_DATE >= DATEADD(YEAR,-1,@TEMPS_DATE) AND @TEMPS_DATE < @PRINTEMPS
    SET @TEMPS_SAISON = 'HIVER'
ELSE IF @TEMPS_DATE >= @PRINTEMPS AND @TEMPS_DATE < @ETE
    SET @TEMPS_SAISON = 'PRINTEMPS'
ELSE IF @TEMPS_DATE >= @ETE AND @TEMPS_DATE < @AUTOMNE
    SET @TEMPS_SAISON = 'ETE'
ELSE IF @TEMPS_DATE >= @AUTOMNE AND @TEMPS_DATE < @HIVER
    SET @TEMPS_SAISON = 'AUTOMNE'
ELSE IF @TEMPS_DATE >= @HIVER AND @TEMPS_DATE < dateadd(year,1,@PRINTEMPS)
    SET @TEMPS_SAISON = 'HIVER'

-- CHARGEMENT DES JOURS FERIES GENERIQUES
IF @TEMPS_MOIS = 1 AND @TEMPS_JOUR = 1
    SET @TEMPS_FERIE = 'Jour de l"An'
ELSE IF @TEMPS_MOIS = 5 AND @TEMPS_JOUR = 1
    SET @TEMPS_FERIE = 'Fête du Travail'
ELSE IF @TEMPS_MOIS = 5 AND @TEMPS_JOUR = 8
    SET @TEMPS_FERIE = 'Victoire 1945'
ELSE IF @TEMPS_MOIS = 7 AND @TEMPS_JOUR = 14
    SET @TEMPS_FERIE = 'Fête Nationale'
ELSE IF @TEMPS_MOIS = 8 AND @TEMPS_JOUR = 15
    SET @TEMPS_FERIE = 'Assomption'
ELSE IF @TEMPS_MOIS = 11 AND @TEMPS_JOUR = 1
    SET @TEMPS_FERIE = 'Toussaint'
ELSE IF @TEMPS_MOIS = 11 AND @TEMPS_JOUR = 11
    SET @TEMPS_FERIE = 'Armistice 1918'
ELSE IF @TEMPS_MOIS = 12 AND @TEMPS_JOUR = 25
    SET @TEMPS_FERIE = 'Noël'

```

```

-- CHARGEMENT DES JOURS FERIES PARTICULIERS AUX ANNEES - DATES SPECIFIQUES

-- ANNEE 2000
-- ELSE IF @TEMPS_DATE = '2000-04-21'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2000-06-01'
--   SET @TEMPS_FERIE = 'Ascension'
-- ELSE IF @TEMPS_DATE = '2000-06-06'
--   SET @TEMPS_FERIE = 'Lundi de Pentecôte'

-- ANNEE 2001
-- ELSE IF @TEMPS_DATE = '2001-04-16'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2001-05-24'
--   SET @TEMPS_FERIE = 'Ascension'
-- ELSE IF @TEMPS_DATE = '2001-06-04'
--   SET @TEMPS_FERIE = 'Lundi de Pentecôte'

-- ANNEE 2002
-- ELSE IF @TEMPS_DATE = '2002-04-01'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2002-05-09'
--   SET @TEMPS_FERIE = 'Ascension'
-- ELSE IF @TEMPS_DATE = '2002-05-20'
--   SET @TEMPS_FERIE = 'Lundi de Pentecôte'

-- ANNEE 2003
-- ELSE IF @TEMPS_DATE = '2003-04-21'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2003-05-29'
--   SET @TEMPS_FERIE = 'Ascension'
-- ELSE IF @TEMPS_DATE = '2003-06-06'
--   SET @TEMPS_FERIE = 'Lundi de Pentecôte'

-- ANNEE 2004
-- ELSE IF @TEMPS_DATE = '2004-04-12'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2004-05-20'
--   SET @TEMPS_FERIE = 'Ascension'
-- ELSE IF @TEMPS_DATE = '2004-05-31'
--   SET @TEMPS_FERIE = 'Lundi de Pentecôte'

-- LE JOUR FERIE DU LUNDI DE PENTECOTE EST SUPPRIME
-- ANNEE 2005
-- ELSE IF @TEMPS_DATE = '2005-03-28'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2005-05-05'
--   SET @TEMPS_FERIE = 'Ascension'

-- ANNEE 2006
-- ELSE IF @TEMPS_DATE = '2006-04-17'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2006-05-25'
--   SET @TEMPS_FERIE = 'Ascension'

-- ANNEE 2007
-- ELSE IF @TEMPS_DATE = '2007-04-09'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2007-05-05'
--   SET @TEMPS_FERIE = 'Ascension'

-- ANNEE 2008
-- ELSE IF @TEMPS_DATE = '2008-03-24'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2008-05-01'
--   SET @TEMPS_FERIE = 'Ascension'

-- ANNEE 2009
-- ELSE IF @TEMPS_DATE = '2009-04-13'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2009-05-21'
--   SET @TEMPS_FERIE = 'Ascension'

-- ANNEE 2010
-- ELSE IF @TEMPS_DATE = '2010-04-05'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2010-05-13'
--   SET @TEMPS_FERIE = 'Ascension'

```

```

ELSE
  SET @TEMPS_FERIE = NULL

-- Ajout des Valeurs dans la Table
INSERT TEMPS
SELECT
  @TEMPS_ID,
  @TEMPS_DATE,
  YEAR(@TEMPS_DATE),
  @TEMPS_MOIS,
  @TEMPS_JOUR,
  @TEMPS_JOURDESEMAINE,
  datepart(dayofyear,@TEMPS_DATE),
  @TEMPS_JOUROUVRE,
  @TEMPS_TRIMESTRE,
  @TEMPS_SAISON,
  @TEMPS_FERIE

SET @TEMPS_DATE = @TEMPS_DATE + 1
SET @TEMPS_ID = @TEMPS_ID + 1

COMMIT TRANSACTION
END
GO
-----
-- MAINTENANT IL FAUT EXECUTER LA PROCEDURE STOCKEE
-- POUR CHARGER LA TABLE AVEC LES DONNEES
-----
-- exec dbo.TEMPS ChargeTableTemps

```

Dans ce script, ne connaissant pas les dates des jours fériés pour l'année 2000 du 'Lundi de Paques' et du 'Lundi de Pentecôte', si quelqu'un peut me fournir un lien ou trouver cette information et même les années précédentes ou même la règle pour trouver ces dates, je suis preneur.

Il faut aussi noter qu'à partir de l'année 2005, le 'Lundi de Pentecôte' ne sera plus férié en France, ce n'est donc pas un oubli dans le script SQL.

Maintenant, juste pour ceux qui veulent le script dans la totalité, je le fourni dans la page suivante.

Le Script SQL Entier

Je rappelle que ce script dans la présentation fournie est spécifique à SQL Server, pour les autres SGBD, il faut le retravailler.

J'ai ajouté dans celui-ci la suppression des objets existants (la table et la procédure stockée) afin de ne pas avoir d'erreur à la création de ceux-ci.

Dans le cadre de remarques constructives faites par M. Brouard, j'ai modifié certaines parties de ce script afin de le rendre plus sûr et performant (même s'il n'est pas destiné à être exécuté tout le temps). Dans le désordre, on trouve maintenant l'utilisation plus rapide des SET en lieu et place des SELECT, l'ajout de l'exécution dans une transaction, la fixation du type de format de la date (afin d'éviter les plantages), l'utilisation des fonctions de recherche des dates de début des saisons, l'utilisation des CAST au lieu des CONVERT, ...

```
----- SPECIFIQUE SQL SERVER -----
-- Suppression de la Table si elle existe déjà
IF EXISTS (select * from dbo.sysobjects where id = object_id(N'[dbo].[TEMPS]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
    DROP table [dbo].[TEMPS]
GO

-- Suppression de la Procédure stockée de chargement
IF EXISTS (select * from dbo.sysobjects where id = object_id(N'[dbo].[TEMPS_ChargeTableTemps]') and OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP procedure [dbo].[TEMPS_ChargeTableTemps]
GO

-- Suppression de la fonction de recherche de la date du début de l'Automne
IF EXISTS (select * from dbo.sysobjects where id = object_id(N'[dbo].[FN_AUTUMN_DATE]') and xtype in (N'FN', N'IF', N'TF'))
    DROP function [dbo].[FN_AUTUMN_DATE]
GO

-- Suppression de la fonction de recherche de la date du début du Printemps
IF EXISTS (select * from dbo.sysobjects where id = object_id(N'[dbo].[FN_SPRING_DATE]') and xtype in (N'FN', N'IF', N'TF'))
    DROP function [dbo].[FN_SPRING_DATE]
GO

-- Suppression de la fonction de recherche de la date du début de l'Eté
IF EXISTS (select * from dbo.sysobjects where id = object_id(N'[dbo].[FN_SUMMER_DATE]') and xtype in (N'FN', N'IF', N'TF'))
    DROP function [dbo].[FN_SUMMER_DATE]
GO

-- Suppression de la fonction de recherche de la date du début de l'Hiver
IF EXISTS (select * from dbo.sysobjects where id = object_id(N'[dbo].[FN_WINTER_DATE]') and xtype in (N'FN', N'IF', N'TF'))
    DROP function [dbo].[FN_WINTER_DATE]
GO

-- Suppression de la sous-fonction de recherche des Saisons
IF EXISTS (select * from dbo.sysobjects where id = object_id(N'[dbo].[FN_SUB_SEASONS]') and xtype in (N'FN', N'IF', N'TF'))
    DROP function [dbo].[FN_SUB_SEASONS]
GO

-----
-- CREATION DE LA TABLE DES TEMPS AVEC TOUS LES CHAMPS NECESSAIRES AU FONCTIONNEMENT

CREATE TABLE dbo.TEMPS
(
    TEMPS_ID          smallInt    NOT NULL,
    TEMPS_DATE        smalldatetime NOT NULL,
    TEMPS_ANNEE       smallInt    NOT NULL,
    TEMPS_MOIS        tinyInt     NOT NULL,
    TEMPS_JOUR        tinyInt     NOT NULL,
    TEMPS_JOURDESEMAINE tinyInt   NOT NULL,
    TEMPS_JOURDELANNEE smallInt   NOT NULL,
    TEMPS_JOUROUVRE   char(1)    NOT NULL,
    TEMPS_TRIMESTRE   tinyInt     NOT NULL,
    TEMPS_SAISON      char(10)   NOT NULL,
    TEMPS_FERIE       varchar(20) NULL
)
GO
```

```

-----
-- CREATION DE LA PROCEDURE STOCKEE D'ALIMENTATION DE LA TABLE DES TEMPS

CREATE PROCEDURE dbo.TEMPS_ChargeTableTemps

AS

DECLARE @TEMPS_DATE    smalldatetime,
        @TEMPS_FIN      smalldatetime,
        @TEMPS_ID       smallInt,
        @TEMPS_JOUROUVRE char(1),
        @TEMPS_JOURDESEMAINE tinyInt,
        @TEMPS_TRIMESTRE tinyInt,
        @TEMPS_ANNEE    char(4),
        @TEMPS_MOIS     smallInt,
        @TEMPS_JOUR     smallInt,
        @TEMPS_SAISON   char(10),
        @TEMPS_FERIE    varchar(20),
        @PRINTEMPS     smalldatetime,
        @ETE            smalldatetime,
        @AUTOMNE        smalldatetime,
        @HIVER          smalldatetime

-- Initialisation du premier Jour de la semaine (1=Lundi)
SET DATEFIRST 1

-- Initialisation du format de la date (Année MOis Jour)
SET DATEFORMAT ymd

-- Initialisation des Variables (Premiere valeur de l'ID, Date de début et date de fin)
SET @TEMPS_DATE = '2000-01-01'
SET @TEMPS_FIN = '2010-12-31'
SET @TEMPS_ID = 1000

-- Début du Chargement
WHILE @TEMPS_DATE < @TEMPS_FIN
BEGIN

    BEGIN TRANSACTION
    -- Creation et Chargement des variables
    SET @TEMPS_ANNEE = CAST(YEAR(@TEMPS_DATE) AS char(4))
    SET @TEMPS_MOIS = MONTH(@TEMPS_DATE)
    SET @TEMPS_JOUR = DAY(@TEMPS_DATE)
    SET @TEMPS_JOURDESEMAINE = datepart(weekday,@TEMPS_DATE)

    -- Chargement du Week-End
    IF @TEMPS_JOURDESEMAINE IN (6,7)
        SET @TEMPS_JOUROUVRE = 'N'
    ELSE
        SET @TEMPS_JOUROUVRE = 'O'

    -- Chargement du Trimestre
    IF MONTH(@TEMPS_DATE) IN (6,7,8)
        SET @TEMPS_TRIMESTRE = 1
    ELSE IF MONTH(@TEMPS_DATE) IN (9,10,11)
        SET @TEMPS_TRIMESTRE = 2
    ELSE IF MONTH(@TEMPS_DATE) IN (12,1,2)
        SET @TEMPS_TRIMESTRE = 3
    ELSE IF MONTH(@TEMPS_DATE) IN (3,4,5)
        SET @TEMPS_TRIMESTRE = 4

/*
    -- Chargement des Dates de début des Saisons : Version Simple ne cherchant pas la précision des dates
    SET @PRINTEMPS = CAST(@TEMPS_ANNEE+'-03-21' AS smalldatetime)
    SET @ETE = CAST(@TEMPS_ANNEE+'-06-21' AS smalldatetime)
    SET @AUTOMNE = CAST(@TEMPS_ANNEE+'-09-21' AS smalldatetime)
    SET @HIVER = CAST(@TEMPS_ANNEE+'-12-21' AS smalldatetime)
*/
    -- Chargement des Dates de début des Saisons : Version Précise fournis par Frédéric BROUARD (MVP SQL Server 2004)
    SET @PRINTEMPS = dbo.FN_SPRING_DATE(@TEMPS_ANNEE)
    SET @ETE = dbo.FN_SUMMER_DATE(@TEMPS_ANNEE)
    SET @AUTOMNE = dbo.FN_AUTUMN_DATE(@TEMPS_ANNEE)
    SET @HIVER = dbo.FN_WINTER_DATE(@TEMPS_ANNEE)

/*
    PRINT 'PRINTEMPS :'+ CAST(@PRINTEMPS AS varchar(100))
    PRINT 'ETE :'+ CAST(@ETE AS varchar(100))
    PRINT 'AUTOMNE :'+ CAST(@AUTOMNE AS varchar(100))
    PRINT 'HIVER :'+ CAST(@HIVER AS varchar(100))
*/
*/

```

```

IF @TEMPS_DATE >= DATEADD(YEAR,-1,@TEMPS_DATE) AND @TEMPS_DATE < @PRINTEMPS
  SET @TEMPS_SAISON = 'HIVER'
ELSE IF @TEMPS_DATE >= @PRINTEMPS AND @TEMPS_DATE < @ETE
  SET @TEMPS_SAISON = 'PRINTEMPS'
ELSE IF @TEMPS_DATE >= @ETE AND @TEMPS_DATE < @AUTOMNE
  SET @TEMPS_SAISON = 'ETE'
ELSE IF @TEMPS_DATE >= @AUTOMNE AND @TEMPS_DATE < @HIVER
  SET @TEMPS_SAISON = 'AUTOMNE'
ELSE IF @TEMPS_DATE >= @HIVER AND @TEMPS_DATE < dateadd(year,1,@PRINTEMPS)
  SET @TEMPS_SAISON = 'HIVER'

-- CHARGEMENT DES JOURS FERIES GENERIQUES
IF @TEMPS_MOIS = 1 AND @TEMPS_JOUR = 1
  SET @TEMPS_FERIE = 'Jour de l"An'
ELSE IF @TEMPS_MOIS = 5 AND @TEMPS_JOUR = 1
  SET @TEMPS_FERIE = 'Fête du Travail'
ELSE IF @TEMPS_MOIS = 5 AND @TEMPS_JOUR = 8
  SET @TEMPS_FERIE = 'Victoire 1945'
ELSE IF @TEMPS_MOIS = 7 AND @TEMPS_JOUR = 14
  SET @TEMPS_FERIE = 'Fête Nationale'
ELSE IF @TEMPS_MOIS = 8 AND @TEMPS_JOUR = 15
  SET @TEMPS_FERIE = 'Assomption'
ELSE IF @TEMPS_MOIS = 11 AND @TEMPS_JOUR = 1
  SET @TEMPS_FERIE = 'Toussaint'
ELSE IF @TEMPS_MOIS = 11 AND @TEMPS_JOUR = 11
  SET @TEMPS_FERIE = 'Armistice 1918'
ELSE IF @TEMPS_MOIS = 12 AND @TEMPS_JOUR = 25
  SET @TEMPS_FERIE = 'Noël'

-- CHARGEMENT DES JOURS FERIES PARTICULIERS AUX ANNEES - DATES SPECIFIQUES

-- ANNEE 2000
-- ELSE IF @TEMPS_DATE = '2000-04-21'
--   SET @TEMPS_FERIE = 'Lundi de Paques'
-- ELSE IF @TEMPS_DATE = '2000-06-01'
--   SET @TEMPS_FERIE = 'Ascension'
-- ELSE IF @TEMPS_DATE = '2000-06-06'
--   SET @TEMPS_FERIE = 'Lundi de Pentecôte'

-- ANNEE 2001
ELSE IF @TEMPS_DATE = '2001-04-16'
  SET @TEMPS_FERIE = 'Lundi de Paques'
ELSE IF @TEMPS_DATE = '2001-05-24'
  SET @TEMPS_FERIE = 'Ascension'
ELSE IF @TEMPS_DATE = '2001-06-04'
  SET @TEMPS_FERIE = 'Lundi de Pentecôte'

-- ANNEE 2002
ELSE IF @TEMPS_DATE = '2002-04-01'
  SET @TEMPS_FERIE = 'Lundi de Paques'
ELSE IF @TEMPS_DATE = '2002-05-09'
  SET @TEMPS_FERIE = 'Ascension'
ELSE IF @TEMPS_DATE = '2002-05-20'
  SET @TEMPS_FERIE = 'Lundi de Pentecôte'

-- ANNEE 2003
ELSE IF @TEMPS_DATE = '2003-04-21'
  SET @TEMPS_FERIE = 'Lundi de Paques'
ELSE IF @TEMPS_DATE = '2003-05-29'
  SET @TEMPS_FERIE = 'Ascension'
ELSE IF @TEMPS_DATE = '2003-06-06'
  SET @TEMPS_FERIE = 'Lundi de Pentecôte'

-- ANNEE 2004
ELSE IF @TEMPS_DATE = '2004-04-12'
  SET @TEMPS_FERIE = 'Lundi de Paques'
ELSE IF @TEMPS_DATE = '2004-05-20'
  SET @TEMPS_FERIE = 'Ascension'
ELSE IF @TEMPS_DATE = '2004-05-31'
  SET @TEMPS_FERIE = 'Lundi de Pentecôte'

-- LE JOUR FERIE DU LUNDI DE PENTECOTE EST SUPPRIME
-- ANNEE 2005
ELSE IF @TEMPS_DATE = '2005-03-28'
  SET @TEMPS_FERIE = 'Lundi de Paques'
ELSE IF @TEMPS_DATE = '2005-05-05'
  SET @TEMPS_FERIE = 'Ascension'

```

```

-- ANNEE 2006
ELSE IF @TEMPS_DATE = '2006-04-17'
    SET @TEMPS_FERIE = 'Lundi de Paques'
ELSE IF @TEMPS_DATE = '2006-05-25'
    SET @TEMPS_FERIE = 'Ascension'

-- ANNEE 2007
ELSE IF @TEMPS_DATE = '2007-04-09'
    SET @TEMPS_FERIE = 'Lundi de Paques'
ELSE IF @TEMPS_DATE = '2007-05-05'
    SET @TEMPS_FERIE = 'Ascension'

-- ANNEE 2008
ELSE IF @TEMPS_DATE = '2008-03-24'
    SET @TEMPS_FERIE = 'Lundi de Paques'
ELSE IF @TEMPS_DATE = '2008-05-01'
    SET @TEMPS_FERIE = 'Ascension'

-- ANNEE 2009
ELSE IF @TEMPS_DATE = '2009-04-13'
    SET @TEMPS_FERIE = 'Lundi de Paques'
ELSE IF @TEMPS_DATE = '2009-05-21'
    SET @TEMPS_FERIE = 'Ascension'

-- ANNEE 2010
ELSE IF @TEMPS_DATE = '2010-04-05'
    SET @TEMPS_FERIE = 'Lundi de Paques'
ELSE IF @TEMPS_DATE = '2010-05-13'
    SET @TEMPS_FERIE = 'Ascension'

ELSE
    SET @TEMPS_FERIE = NULL

-- Ajout des Valeurs dans la Table
INSERT TEMPS
SELECT
    @TEMPS_ID,
    @TEMPS_DATE,
    YEAR(@TEMPS_DATE),
    @TEMPS_MOIS,
    @TEMPS_JOUR,
    @TEMPS_JOURDESEMAINE,
    datepart(dayofyear,@TEMPS_DATE),
    @TEMPS_JOUROUVRE,
    @TEMPS_TRIMESTRE,
    @TEMPS_SAISON,
    @TEMPS_FERIE

SET @TEMPS_DATE = @TEMPS_DATE + 1
SET @TEMPS_ID = @TEMPS_ID + 1

COMMIT TRANSACTION
END
GO

-----
/*****
CALCUL DES SAISONS - basé sur l'algorithmes de Meeus
("Astronomical Algorithms", 1991) modifié par Simon Cassidy
-----
Frédéric BROUARD - SQLpro - 2004-06-27 - Transact SQL
http://sqlpro.developpez.com/MSSQLServer_fonctions/UDF.html#FN_SEASONS
*****/
CREATE FUNCTION FN_SPRING_DATE (@Y INT)
RETURNS DATETIME
AS
-- calcul de la date de début du printemps

BEGIN
-- @Y est l'année considérée
-- effets de bord : année absente
IF @Y IS NULL
    RETURN NULL

-- limites de calculs de l'an 1000 à l'an 3000
IF NOT @Y BETWEEN 1000 AND 3000
    RETURN NULL

```

```

DECLARE @M FLOAT(50)
DECLARE @JDME FLOAT(50)
DECLARE @T FLOAT(50)
DECLARE @S FLOAT(50)
DECLARE @W FLOAT(50)
DECLARE @L FLOAT(50)
DECLARE @JD FLOAT(50)
DECLARE @D DATETIME

SET @M = (CAST(@Y AS FLOAT(50)) - 2000.0) / 1000.0
SET @JDME = 2451623.80984
      + 365242.37404 * @M
      + 0.05169 * SQUARE(@M)
      - 0.00411 * POWER (@M,3)
      - 0.00057 * POWER (@M,4)

RETURN dbo.FN_SUB_SEASONS (@JDME)
END
GO

-----
/*****
  CALCUL DES SAISONS - basé sur l'algorithmes de Meeus
  ("Astronomical Algorithms", 1991) modifié par Simon Cassidy
-----
Frédéric BROUARD - SQLpro - 2004-06-27 - Transact SQL
http://sqlpro.developpez.com/MSSQLServer\_fonctions/UDF.html#FN\_SEASONS
*****/
CREATE FUNCTION FN_SUMMER_DATE (@Y INT)
  RETURNS DATETIME
AS
-- calcul de la date de début de l'été

BEGIN
-- @Y est l'année considérée
-- effets de bord : année absente
IF @Y IS NULL
  RETURN NULL

-- limites de calculs de l'an 1000 à l'an 3000
IF NOT @Y BETWEEN 1000 AND 3000
  RETURN NULL

DECLARE @M FLOAT(50)
DECLARE @JDME FLOAT(50)

SET @M = (CAST(@Y AS FLOAT(50)) - 2000.0) / 1000.0
SET @JDME = 2451716.56767
      + 365241.62603 * @M
      + 0.00325 * SQUARE(@M)
      + 0.00888 * POWER (@M, 3)
      - 0.00030 * POWER (@M,4)

RETURN dbo.FN_SUB_SEASONS (@JDME)
END
GO

-----
/*****
  CALCUL DES SAISONS - basé sur l'algorithmes de Meeus
  ("Astronomical Algorithms", 1991) modifié par Simon Cassidy
-----
Frédéric BROUARD - SQLpro - 2004-06-27 - Transact SQL
http://sqlpro.developpez.com/MSSQLServer\_fonctions/UDF.html#FN\_SEASONS
*****/
CREATE FUNCTION FN_AUTUMN_DATE (@Y INT)
  RETURNS DATETIME
AS
-- calcul de la date de début d'automne

BEGIN
-- @Y est l'année considérée
-- effets de bord : année absente

IF @Y IS NULL
  RETURN NULL

-- limites de calculs de l'an 1000 à l'an 3000
IF NOT @Y BETWEEN 1000 AND 3000
  RETURN NULL

```

```

DECLARE @M FLOAT(50)
DECLARE @JDME FLOAT(50)

SET @M = (CAST(@Y AS FLOAT(50)) - 2000.0) / 1000.0
SET @JDME = 2451810.21715
      + 365242.01767 * @M
      - 0.11575 * SQUARE(@M)
      + 0.00337 * POWER (@M, 3)
      + 0.00078 * POWER (@M, 4)

RETURN dbo.FN_SUB_SEASONS (@JDME)
END
GO

-----
/*****
  CALCUL DES SAISONS - basé sur l'algorithmes de Meeus
  ("Astronomical Algorithms", 1991) modifié par Simon Cassidy
-----
Frédéric BROUARD - SQLpro - 2004-06-27 - Transact SQL
http://sqlpro.developpez.com/MSSQLServer\_fonctions/UDF.html#FN\_SEASONS
-----/
CREATE FUNCTION FN_WINTER_DATE (@Y INT)
  RETURNS DATETIME
AS
-- calcul de la date de début d'hiver

BEGIN
-- @Y est l'année considérée
-- effets de bord : année absente
IF @Y IS NULL
  RETURN NULL

-- limites de calculs de l'an 1000 à l'an 3000
IF NOT @Y BETWEEN 1000 AND 3000
  RETURN NULL

DECLARE @M FLOAT(50)
DECLARE @JDME FLOAT(50)

SET @M = (CAST(@Y AS FLOAT(50)) - 2000.0) / 1000.0
SET @JDME = 2451900.05952
      + 365242.74049 * @M
      - 0.06223 * SQUARE(@M)
      - 0.00823 * POWER (@M, 3)
      + 0.00032 * POWER (@M, 4)

RETURN dbo.FN_SUB_SEASONS (@JDME)
END
GO

-----
/*****
  CALCUL DES SAISONS - basé sur l'algorithmes de Meeus
  ("Astronomical Algorithms", 1991) modifié par Simon Cassidy
-----
Frédéric BROUARD - SQLpro - 2004-06-27 - Transact SQL
http://sqlpro.developpez.com/MSSQLServer\_fonctions/UDF.html#FN\_SEASONS
-----/
CREATE FUNCTION FN_SUB_SEASONS (@JDME FLOAT(50))
  RETURNS DATETIME
AS
-- sous fonction utilisée par les 4 calculs de saisons
BEGIN

IF @JDME IS NULL
  RETURN NULL

DECLARE @T FLOAT(50)
DECLARE @S FLOAT(50)
DECLARE @W FLOAT(50)
DECLARE @L FLOAT(50)
DECLARE @JD FLOAT(50)
DECLARE @D DATETIME

SET @T = (@JDME - 2451545.0) / 36525

```

```

SET @S =
485 * COS(0.43563418129778464 + 33.757041381353048 * @T) +
203 * COS(0.64978608051748876 + 575.33848531501758 * @T) +
199 * COS(0.73443454923921381 + 0.35231216280757538 * @T) +
182 * COS(0.48607419668042079 + 7771.3771552463541 * @T) +
156 * COS(1.2765338149086527 + 786.04194554533876 * @T) +
136 * COS(1.2482594810263443 + 393.02097277266938 * @T) +
77 * COS(0.39339721339952183 + 1150.6769706300352 * @T) +
74 * COS(1.6880824525289155 + 52.969102188531025 * @T) +
70 * COS(0.76061448801912879 + 157.7343580417903 * @T) +
58 * COS(0.34574972482007665 + 588.4926828214484 * @T) +
52 * COS(1.69593643416289 + 2.6298272103200158 * @T) +
50 * COS(0.366868208769208 + 39.81490468210017 * @T) +
45 * COS(0.82972952639810416 + 522.36940057977904 * @T) +
44 * COS(0.43895030687657388 + 550.75533081445974 * @T) +
29 * COS(1.063429113240145 + 77.552256689088878 * @T) +
18 * COS(0.96202548369927443 + 1179.0629008647159 * @T) +
17 * COS(1.5496778428457652 + 79.629809364200341 * @T) +
16 * COS(1.7111207986552408 + 1097.7078858947966 * @T) +
14 * COS(1.7411404617895434 + 548.67777813934822 * @T) +
12 * COS(1.6648695734773908 + 254.43144545527034 * @T) +
12 * COS(1.5203563114122605 + 557.31427814345443 * @T) +
12 * COS(0.36320301734001997 + 606.97767436883066 * @T) +
9 * COS(0.48397980157802756 + 21.32991313471798 * @T) +
8 * COS(0.2696533694331239 + 294.24635013737048 * @T)

```

```

SET @W = ( 35999.373*@T - 2.47 ) * PI() / 180

```

```

SET @L = 1 + 0.0334*COS(@W) + 0.0007*COS(2*@W)

```

```

SET @JD = @JDME + (0.00001*@S/@L)

```

```

SET @JD = @JD - 2415020.50208142228

```

```

SET @D = CAST(FLOOR(@JD) AS DATETIME)

```

```

RETURN @D
END
GO

```

```

-----
-- MAINTENANT IL FAUT EXECUTER LA PROCEDURE STOCKEE
-- POUR CHARGER LA TABLE AVEC LES DONNEES
-----
-- exec dbo.TEMPS ChargeTableTemps

```

Conclusion

Ce script SQL a été développé dans un besoin spécifique pour établir un référentiel de Temps dans un outil de Ressources Humaines, vous pouvez l'adapter comme vous le souhaitez pour vos besoins spécifiques et le personnaliser à votre guise.

Vous avez aussi une petite base Access sur le lien suivant regroupant les premiers mois pour les Ephémérides :

- [Script d'utilisation des Ephémérides en ASP \(FR\)](#)

Quelques liens utiles sur le sujet :

- [Site web consacré au langage SQL \(FR\)](#)
- [Calendrier, timing et horaires en SQL... \(FR\)](#)
- [Date and Time Manipulation in SQL Server 2000 \(US\)](#)
- [Working with SQL Server Date/Time Variables \(US\)](#)

Vous trouverez le fichier SQL à l'adresse suivante :

- **Fichier SQL ainsi que l'article au format PDF**

En vous souhaitant de bons projets de développement.

Romelard Fabrice (alias F___)