

GESTION DE SYNDIC

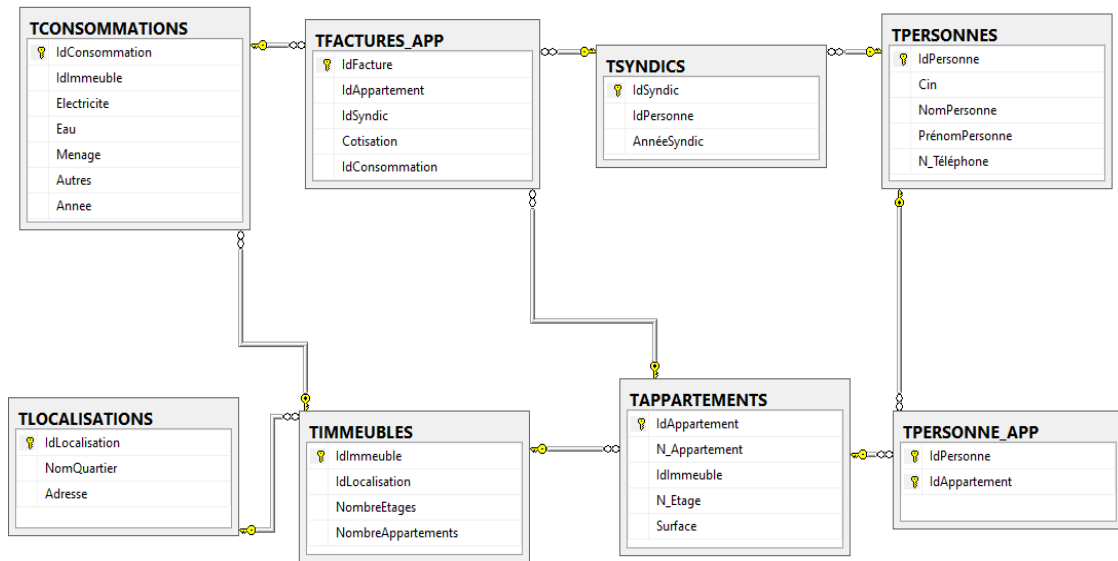
Une chaîne des immeubles - localisée au Maroc - souhaite créer un système d'information permettant de centraliser et de gérer les différentes prestations pour ses syndic. Ci-dessous, la liste de règles de gestion à prendre en considération :

- ✓ Une seule localisation peut avoir un immeuble ou plusieurs.
- ✓ Un immeuble peut être dans une seule localisation.
- ✓ Un immeuble peut contenir plusieurs appartements et plusieurs étages.
- ✓ Un appartement peut être dans un seul immeuble et chaque appartement peut prendre des caractéristiques différentes.
- ✓ Une consommation peut être relative à un seul immeuble.
- ✓ Une consommation peut être relative à une seule année.
- ✓ Un immeuble peut avoir une consommation ou plusieurs.
- ✓ Un personne (propriétaire) peut prendre un appartement ou plusieurs.
- ✓ Un appartement peut avoir un propriétaire ou plusieurs.
- ✓ Un appartement peut payer une facture ou plusieurs.
- ✓ Une facture peut payer par un seul appartement.
- ✓ Un syndic peut prendre une facture ou plusieurs.
- ✓ Une facture peut donner à un seul syndic.
- ✓ une facture peut payer une partie d'une seule consommation.
- ✓ La consommation peut payer par une seule facture ou plusieurs.

On base sur les données dessus et on propose ce modèle logique des données :

- TLOCALISATIONS (IdLocalisation , NomQuartier , Adresse)
- TIMMEUBLES (IdImmeuble , IdLocalisation , NombreEtages , NombreAppartements)
- TAPPARTEMENTS (IdAppartement , N_Appartement , N_Etage , Surface , #IdImmeuble)
- TPERSONNES (IdPersonne , Cin , NomPersonne , PrénomPersonne , N_Téléphone)
- TSYNDICS (IdSyndic , AnnéeSyndic , #IdPersonne)
- TCONSOMMATION (IdConsommation , Electricite , Eau , Menage , Autres , Année , #IdImmeuble)
- TFACTURES (IdFacture , Cotisation , #IdAppartement , #IdSyndic , #IdConsommation)
- TPERSONNE_APP (#IdPersonne , #IdAppartement)

Diagramme de la Base de Données



1. Les personnes qui ne sont pas syndic

• Version 1

```

CREATE PROCEDURE personne_jamais_syndic_v1
AS
BEGIN
    SELECT
        IdPersonne, NomPersonne, PrénomPersonne
    FROM
        TPERSONNES
    WHERE IdPersonne NOT IN
        (
            SELECT
                IdPersonne
            FROM
                TSYNDICS
        )
END
    
```

• Version 2

```

CREATE PROCEDURE personne_jamais_syndic_v2
AS
BEGIN
    SELECT
        IdPersonne, NomPersonne, PrénomPersonne
    FROM
        TPERSONNES P
    WHERE NOT EXISTS
    
```

```

        (
            SELECT
                IdPersonne
            FROM
                TSYNDICS S
            WHERE
                P.IdPersonne=S.IdPersonne
        )
END

```

- Version 3

```

CREATE PROCEDURE personne_jamais_syndic_v3
AS
BEGIN
    SELECT
        P.IdPersonne,NomPersonne,PrénomPersonne
    FROM
        TPERSONNES P LEFT JOIN TSYNDICS S
        ON
            P.IdPersonne=S.IdPersonne
    WHERE IdSyndic IS NULL
END

```

- Version 4

```

CREATE PROCEDURE personne_jamais_syndic_v4
AS
BEGIN
    SELECT
        P.IdPersonne,NomPersonne,PrénomPersonne
    FROM
        TSYNDICS S RIGHT JOIN TPERSONNES P
        ON
            P.IdPersonne=S.IdPersonne
    WHERE IdSyndic IS NULL
END

```

2. Les personnes qui ont plus d'un appartement

```

CREATE PROCEDURE personne_admet_plus_app
AS
BEGIN
    SELECT
        IdPersonne,nbapp
    FROM
        (
            SELECT
                IdPersonne,count(IdAppartement) as nbapp
            FROM
                TPERSONNE_APP
            GROUP BY IdPersonne
        ) AS T
    WHERE
        nbapp > 1

```

END

3. Les appartements qui ont plus d'un propriétaire

```
CREATE PROCEDURE appartement_a_plus_per
AS
BEGIN
    SELECT
        IdAppartement, COUNT(IdPersonne) AS NBR_per
    FROM TPERSONNE_APP
    GROUP BY IdAppartement
    HAVING
        COUNT(IdPersonne) > 1
END
```

4. Les immeubles d'une localisation donnée

```
CREATE PROCEDURE liste_immeuble_local
    @IdLocalisation AS INT
AS
BEGIN
    SELECT
        IdImmeuble , TLOCALISATIONS.IdLocalisation , NomQuartier
    FROM
        TIMMEUBLES, TLOCALISATIONS
    WHERE
        TIMMEUBLES.IdLocalisation=TLOCALISATIONS.IdLocalisation AND
        TLOCALISATIONS.IdLocalisation= @IdLocalisation
END
```

5. Calculer la cotisation de chaque appartement

- Version 1

```
CREATE PROCEDURE cotisation_chaque_app
AS
BEGIN
    SELECT
        IdFacture , IdAppartement ,
        (Electricite+Eau+Menage+Autres)/NombreAppartements AS Cotisation , Annee
    FROM
        TFACTURES_APP F , TCONSOMMATIONS C , TIMMEUBLES I
    WHERE
        F.IdConsommation = C.IdConsommation
        AND
        C.IdImmeuble = I.IdImmeuble
END
```

- Version 2

```
CREATE PROCEDURE cotisation_chaque_app_sous_req
```

```

AS
BEGIN
    SELECT IdFacture , IdAppartement ,
    (
    SELECT
        (Electricite+Eau+Menage+Autres)/NombreAppartements AS Cotisation
    FROM
        TCONSOMMATIONS C , TIMMEUBLES I
    WHERE
        TFACTURES_APP.IdConsommation = C.IdConsommation
        AND
        C.IdImmeuble = I.IdImmeuble
    ) AS Cotisation , Annee
    FROM
        TFACTURES_APP , TCONSOMMATIONS
END

```

6. Calculer la cotisation d'un appartement donnée et une année donnée

```

CREATE PROCEDURE cotisation_annee_app
    @annee AS INT,
    @idappartement AS INT
AS
BEGIN
    SELECT
        IdFacture , IdAppartement ,
        (Electricite+Eau+Menage+Autres)/NombreAppartements AS Cotisation , Annee
    FROM
        TFACTURES_APP F , TCONSOMMATIONS C , TIMMEUBLES I
    WHERE
        IdAppartement=@idappartement
        AND
        annee=@annee
        AND
        F.IdConsommation = C.IdConsommation
        AND
        C.IdImmeuble = I.IdImmeuble
END

```

7. la consommation moyenne des immeubles dans chaque année

```

CREATE PROCEDURE consommation_moy_par_imm_annee
AS
BEGIN
    SELECT
        IdConsommation , I.IdImmeuble,
        (Electricite+Eau+Menage+Autres)/NombreAppartements AS Consomation , Annee
    FROM
        TCONSOMMATIONS C , TIMMEUBLES I
    WHERE
        C.IdImmeuble = I.IdImmeuble
END

```

8. Calculer la consommation max des immeubles d'une année donnée

- Version 1

```

CREATE PROCEDURE cons_max_immeuble
  @Année AS INT
AS
BEGIN
    SELECT
        MAX(consom) AS con_max
    FROM
        (
            SELECT
                (Electricite+Eau+Menage+Autres) AS consom , IdImmeuble
            FROM
                TCONSOMMATIONS
            WHERE
                Annee=@Année
        ) AS T

END

```

- Version 2

```

CREATE PROCEDURE conso_max_top
  @Année AS INT
AS
BEGIN
    SELECT
        TOP(1) WITH TIES IdImmeuble ,(Electricite+Eau+Menage+Autres) AS
consom_max
    FROM
        TCONSOMMATIONS
    WHERE
        Annee= @Année
    ORDER BY
        consom_max desc

END

```

9. Calculer la consommation min des immeubles d'une année donnée

- Version 1

```

CREATE PROCEDURE cons_min_immeuble
  @Année AS INT
AS
BEGIN
    SELECT
        MIN(consom) AS con_min
    FROM
        (
            SELECT
                (Electricite+Eau+Menage+Autres) AS consom , IdImmeuble
            FROM
                TCONSOMMATIONS
            WHERE
                Annee=@Année
        ) AS T

```

END

- Version 2

```
CREATE PROCEDURE conso_min_top
@Année AS INT
AS
BEGIN
    SELECT
        TOP(1) WITH TIES IdImmeuble ,(Electricite+Eau+Menage+Autres) AS
        consom_min
    FROM
        TCONSOMMATIONS
    WHERE
        Annee=@Année
    ORDER BY
        consom_min

END
```

10. Les appartements qui n'ont pas une facture

- Version 1

```
CREATE PROCEDURE app_pas_facturev1
AS
BEGIN
    SELECT
        IdAppartement ,IdImmeuble
    FROM
        TAPPARTEMENTS
    WHERE
        IdAppartement NOT IN
        (
            SELECT
                IdAppartement
            FROM
                TFACTURES_APP
        )

END
```

- Version 2

```
CREATE PROCEDURE app_pas_facturev2
AS
BEGIN
    SELECT
        IdAppartement, IdImmeuble
    FROM
        TAPPARTEMENTS
    WHERE NOT EXISTS
        (
            SELECT
```

```

        IdAppartement
FROM      TFACTURES_APP
WHERE     TAPPARTEMENTS.IdAppartement = TFACTURES_APP.IdAppartement
)
END

```

- Version 3

```

CREATE PROCEDURE app_pas_facturev3
AS
BEGIN
    SELECT
        IdAppartement
    FROM
        TAPPARTEMENTS
    EXCEPT
    SELECT
        IdAppartement
    FROM
        TFACTURES_APP
END

```

11. Une procédure stockée permettant de mettre à jour le champ de cotisation pour l'ensemble des appartements

- Version 1

```

CREATE PROCEDURE clc_cotisation_curseur
AS
BEGIN
DECLARE
    @idfacture AS INT,
    @Cotisation AS FLOAT

    declare CursCOT CURSOR LOCAL FAST_FORWARD
    FOR
        SELECT
            IdFacture , Cotisation
    FROM
        (
    SELECT
        IdFacture , IdAppartement ,
(Electricite+Eau+Menage+Autres)/NombreAppartements AS Cotisation , Annee
    FROM
        TFACTURES_APP F , TCONSOMMATIONS C , TIMMEUBLES I
    WHERE
        F.IdConsommation = C.IdConsommation
        AND
        C.IdImmeuble = I.IdImmeuble
    ) AS T

```



```

OPEN CursCOT
FETCH NEXT FROM CursCOT INTO @idfacture,@Cotisation
WHILE (@@FETCH_STATUS = 0)
BEGIN
    UPDATE
        TFACTURES_APP
    SET
        Cotisation= @Cotisation
    WHERE
        IdFacture = @idfacture

    FETCH NEXT FROM CursCOT INTO @idfacture,@Cotisation
END
CLOSE CursCOT
DEALLOCATE CurCOT
END
GO

```

- Version 2

Création de la vue :

```

CREATE VIEW view_cotisation_chaque_app
AS
    SELECT
        IdFacture , IdAppartement ,
        (Electricite+Eau+Menage+Autres)/NombreAppartements AS Cotisation , Annee
    FROM
        TFACTURES_APP F , TCONSOMMATIONS C , TIMMEUBLES I
    WHERE
        F.IdConsommation = C.IdConsommation
        AND
        C.IdImmeuble = I.IdImmeuble

```

- ```

CREATE PROCEDURE clc_cotisation_avc_view
AS
BEGIN
 UPDATE
 TFACTURES_APP
 SET
 Cotisation = v.Cotisation
 FROM
 view_cotisation_chaque_app AS v
 WHERE TFACTURES_APP.IdFacture = v.Idfacture
END

```

- Version 3

Création de la fonction:

```

CREATE FUNCTION cotis_par_appv1
(@IdFacture AS INT)

```

```

RETURNS REAL
AS
BEGIN
DECLARE
 @cot AS REAL
SELECT
 @cot =Cotisation
FROM
 view_cotisation_chaque_app
where
 IdFacture = @IdFacture
return @cot
END

```

- ```

CREATE PROCEDURE clc_cotisation_avc_fct
AS
BEGIN

UPDATE
    TFACTURES_APP
SET
    cotisation = dbo.cotis_par_appv1(IdFacture)

END

```