



BASES DE DONNÉES :

DDL: MODIFIER LA STRUCTURE DE LA BD (ALTER)

**Bases de données: Avancé
420-465-CH**

**Sana El Bahloul
Hiver 2018**

MODIFIER LA STRUCTURE D'UNE TABLE

- Il est possible de modifier la structure et les contraintes d'une table déjà créée par la commande **ALTER TABLE**.
- Voici ce qu'il est possible de réaliser :
 - ajouter/supprimer une colonne
 - créer/supprimer une clé primaire
 - créer/supprimer une clé étrangère
 - changer la valeur par défaut d'un champ
 - ajouter une contrainte d'unicité (interdire les doublons)
 - modifier un champ (type de données, longueur, contraintes)
 - changer totalement la définition d'un champ



AJOUTER UN CHAMP

- Il est possible d'ajouter une colonne à une table après sa création.

Syntaxe:

```
ALTER TABLE tblNomTable ADD NomColonne typededonnées;
```

Exemple : ajout d'une colonne Tel qui est une chaîne de 10 caractères

```
ALTER TABLE tblPersonne ADD telPers char(10);
```

- **Attention:** il est généralement préférable de modifier le script de création à moins que cette modification survienne alors que la bd contient des données autres que celles du chargement initial.



SUPPRIMER UN CHAMP

- Attention, supprimer un attribut implique la suppression des valeurs qui se trouvent dans la colonne qui correspond à cet attribut.

Syntaxe:

```
ALTER TABLE tblNomTable DROP NomChamp;
```

Exemple : enlevez la colonne Tel

```
ALTER TABLE tblPersonne DROP telPers;
```



AJOUTER/SUPPRIMER UNE CLÉ PRIMAIRE

- Il est possible d'ajouter une contrainte clé primaire à une colonne existante

```
ALTER TABLE tblNomTable ADD PRIMARY KEY (NomColonne,...);
```

Exemple:

```
ALTER TABLE tblPersonne ADD PRIMARY KEY (noPers);
```

- Il est aussi possible de retirer la contrainte de clé primaire. Attention cette commande retire la contrainte mais ne détruit pas la colonne

```
ALTER TABLE tblNomTable DROP PRIMARY KEY;
```

Exemple:

```
ALTER TABLE tblPersonne DROP PRIMARY KEY;
```

SUPPRIMER UNE CLÉ ÉTRANGÈRE

- Ajout d'une clé étrangère

```
ALTER TABLE tblNomTable ADD CONSTRAINT FOREIGN KEY  
(colonne) REFERENCES tblNomTable_ref(col_ref)
```

Exemple:

```
ALTER TABLE tblPersonne ADD CONSTRAINT FOREIGNKEY  
fk_noEmploi (noEmploi) REFERENCES tblEmploi (noEmploi);
```

- Il est également possible de supprimer une clé étrangère

```
ALTER TABLE tblNomTable DROP FOREIGN KEY (NomColonne);
```

Exemple:

```
ALTER TABLE tblPersonne DROP FOREIGN KEY(noEmploi) ;
```

AJOUTER/SUPPRIMER UNE VALEUR PAR DÉFAUT

- Ajouter une valeur par défaut

```
ALTER TABLE tblNomTable ALTER NomColonne SET DEFAULT valeur;
```

Exemple:

```
ALTER TABLE tblPersonne ALTER telPers SET DEFAULT ' ';
```

- Supprimer une valeur par défaut

```
ALTER TABLE tblNomTable ALTER NomColonne DROP DEFAULT;
```

Exemple:

```
ALTER TABLE tblPersonne ALTER telPers DROP DEFAULT;
```



AJOUTER/SUPPRIMER UNE CONTRAINTE D'UNICITÉ

- Ajouter une contrainte d'unicité

```
ALTER TABLE tblNomTable ADD UNIQUE (NomColonne);
```

Exemple:

```
ALTER TABLE tblPersonne ADD UNIQUE (nomPers);
```

- Supprimer une contrainte d'unicité

```
ALTER TABLE tblNomTable DROP UNIQUE (NomColonne);
```

Exemple:

```
ALTER TABLE tblPersonne DROP UNIQUE (nomPers);
```



MODIFIER UN TYPE DE DONNÉES

- Modifier seulement le type de données

```
ALTER TABLE tblNomTable MODIFY NomColonne nouveautype;
```

Exemple:

```
ALTER TABLE tblPersonne MODIFY nomPers varchar(40);
```

- Modifier un type de données et une contrainte

```
ALTER TABLE tblNomTable MODIFY Nomcolonne nouveautype contrainte;
```

Exemple:

```
ALTER TABLE tblPersonne MODIFY nomPers varchar(40) NOT NULL;
```



MODIFIER COMPLÈTEMENT UN CHAMP

- Modifier du même coup le nom, le type et les contraintes.

```
ALTER TABLE tblNomTable CHANGE AncienNomColonne  
NouveauNomColonne nouveautype nouvellecontrainte;
```

Exemple:

```
ALTER TABLE tblPersonne  
CHANGE prenom prenPers varchar(40) NOT NULL;
```



RENOMMER UNE COLONNE

- Renommer une colonne

```
ALTER TABLE tblNomTable  
    CHANGE AncienNomColonne NouveauNomColonne;
```

Exemple:

```
ALTER TABLE tblPersonne  
    CHANGE prenom prenPers;
```



RENOMMER UNE TABLE

- Comme nous avons vu, la commande **ALTER TABLE** permet de modifier la structure et les contraintes d'une table. Nous pouvons aussi modifier le nom d'une table à l'aide de la commande **RENAME**

```
RENAME tblNomTable TO tblNouveauNom;
```

Exemple:

```
RENAME Personne TO tblPersonne;
```

