

## Insertion, suppression, modification en SQL

- L'insertion de données dans une table se fait à l'aide de INSERT

## Insertion, suppression, modification en SQL

- L'insertion de données dans une table se fait à l'aide de INSERT  
`INSERT INTO nom_table (col1, col2, col3, ...) VALUES val1, val2, val3, ...`

### Insertion, suppression, modification en SQL

- L'insertion de données dans une table se fait à l'aide de INSERT  
`INSERT INTO nom_table (col1, col2, col3, ...) VALUES val1, val2, val3, ...`

On peut éviter de spécifier les noms de colonnes si on insère toutes les données

### Insertion, suppression, modification en SQL

- L'insertion de données dans une table se fait à l'aide de `INSERT`  
`INSERT INTO nom_table (col1, col2, col3, ...) VALUES val1, val2, val3, ...`  
On peut éviter de spécifier les noms de colonnes si on insère toutes les données
- La suppression de données dans une table se fait à l'aide de `DELETE`

### Insertion, suppression, modification en SQL

- L'insertion de données dans une table se fait à l'aide de INSERT  
`INSERT INTO nom_table (col1, col2, col3, ...) VALUES val1, val2, val3, ...`  
On peut éviter de spécifier les noms de colonnes si on insère toutes les données
- La suppression de données dans une table se fait à l'aide de DELETE  
`DELETE FROM nom_table WHERE condition`

### Insertion, suppression, modification en SQL

- L'insertion de données dans une table se fait à l'aide de `INSERT`  
`INSERT INTO nom_table (col1, col2, col3, ...) VALUES val1, val2, val3, ...`  
On peut éviter de spécifier les noms de colonnes si on insère toutes les données
- La suppression de données dans une table se fait à l'aide de `DELETE`  
`DELETE FROM nom_table WHERE condition`  
La `condition` indiquée permet de sélectionner les données à effacer et s'écrit de la même façon que les conditions d'une requête `SELECT`

### Insertion, suppression, modification en SQL

- L'insertion de données dans une table se fait à l'aide de `INSERT`  
`INSERT INTO nom_table (col1, col2, col3, ...) VALUES val1, val2, val3, ...`  
On peut éviter de spécifier les noms de colonnes si on insère toutes les données
- La suppression de données dans une table se fait à l'aide de `DELETE`  
`DELETE FROM nom_table WHERE condition`  
La `condition` indiquée permet de sélectionner les données à effacer et s'écrit de la même façon que les conditions d'une requête `SELECT`
- La modification de données dans une table se fait à l'aide de `UPDATE`

### Insertion, suppression, modification en SQL

- L'insertion de données dans une table se fait à l'aide de `INSERT`  
`INSERT INTO nom_table (col1, col2, col3, ...) VALUES val1, val2, val3, ...`  
On peut éviter de spécifier les noms de colonnes si on insère toutes les données
- La suppression de données dans une table se fait à l'aide de `DELETE`  
`DELETE FROM nom_table WHERE condition`  
La `condition` indiquée permet de sélectionner les données à effacer et s'écrit de la même façon que les conditions d'une requête `SELECT`
- La modification de données dans une table se fait à l'aide de `UPDATE`  
`UPDATE table SET col1=val1, col2=val2, ... WHERE condition`

## Exemples

On prend l'exemple de la table personnes :

id	Nom	Prénom	Naissance
1	Pascal	Blaise	1623
2	Lovelace	Ada	1815
3	Boole	George	1815

- Ecrire une requête permettant d'insérer Alan Turing né en 1912 dans cette table en incrémentant le champ id

### Exemples

On prend l'exemple de la table personnes :

id	Nom	Prénom	Naissance
1	Pascal	Blaise	1623
2	Lovelace	Ada	1815
3	Boole	George	1815

- Ecrire une requête permettant d'insérer Alan Turing né en 1912 dans cette table en incrémentant le champ id
- Insérer John Doe dans cette table, sans préciser sa date de naissance ni le champ id

## Exemples

On prend l'exemple de la table personnes :

id	Nom	Prénom	Naissance
1	Pascal	Blaise	1623
2	Lovelace	Ada	1815
3	Boole	George	1815

- Ecrire une requête permettant d'insérer Alan Turing né en 1912 dans cette table en incrémentant le champ id
- Insérer John Doe dans cette table, sans préciser sa date de naissance ni le champ id
- Modifier l'enregistrement de John Doe en donnant son année de naissance : 2021 et son id : 13.

## Exemples

On prend l'exemple de la table personnes :

id	Nom	Prénom	Naissance
1	Pascal	Blaise	1623
2	Lovelace	Ada	1815
3	Boole	George	1815

- Ecrire une requête permettant d'insérer Alan Turing né en 1912 dans cette table en incrémentant le champ id
- Insérer John Doe dans cette table, sans préciser sa date de naissance ni le champ id
- Modifier l'enregistrement de John Doe en donnant son année de naissance : 2021 et son id : 13.
- Supprimer l'enregistrement de John Doe de la table

## C8 Schéma relationnel d'une base de données

### Schéma relationnel

Le schéma relationnel d'une base de données présente les tables de cette base sous la forme de liste ou de tableau. Dans les deux cas, on précise la clé primaire de la table en soulignant l'attribut. On indique aussi parfois le type des attributs.

### Exemple

## C8 Schéma relationnel d'une base de données

### Schéma relationnel

Le schéma relationnel d'une base de données présente les tables de cette base sous la forme de liste ou de tableau. Dans les deux cas, on précise la clé primaire de la table en soulignant l'attribut. On indique aussi parfois le type des attributs.

### Exemple

Le schéma relationnel de la table `personne` peut s'écrire :

## C8 Schéma relationnel d'une base de données

### Schéma relationnel

Le schéma relationnel d'une base de données présente les tables de cette base sous la forme de liste ou de tableau. Dans les deux cas, on précise la clé primaire de la table en soulignant l'attribut. On indique aussi parfois le type des attributs.

### Exemple

Le schéma relationnel de la table `personne` peut s'écrire :

- Sous forme de liste :

## C8 Schéma relationnel d'une base de données

### Schéma relationnel

Le schéma relationnel d'une base de données présente les tables de cette base sous la forme de liste ou de tableau. Dans les deux cas, on précise la clé primaire de la table en soulignant l'attribut. On indique aussi parfois le type des attributs.

### Exemple

Le schéma relationnel de la table `personne` peut s'écrire :

- Sous forme de liste :

**personnes** (id : INT, Nom : TEXT, Prenom : TEXT, Naissance : INT)

## C8 Schéma relationnel d'une base de données

### Schéma relationnel

Le schéma relationnel d'une base de données présente les tables de cette base sous la forme de liste ou de tableau. Dans les deux cas, on précise la clé primaire de la table en soulignant l'attribut. On indique aussi parfois le type des attributs.

### Exemple

Le schéma relationnel de la table `personne` peut s'écrire :

- Sous forme de liste :

**personnes** (id : INT, Nom : TEXT, Prenom : TEXT, Naissance : INT)

- Sous forme de tableau :

<b>personnes</b>	
<u>id</u>	: INT
Nom	: TEXT
Prenom	: TEXT
Naissance	: INT

## C8 Schéma relationnel d'une base de données

### Duplication de l'information

On évite pour de multiples raisons (espace occupé, efficacité pour les recherches ou les modifications, ...) de dupliquer l'information présente dans une base de données. Cela conduit à l'utilisation de plusieurs tables **liées** entre elles, c'est à dire que certains attributs d'une table sont les clés primaires d'une autre table. On dit que ce sont des **clés étrangères**. Les clés étrangères sont précédées de # dans le schéma relationnel.

### Exemple

## C8 Schéma relationnel d'une base de données

### Duplication de l'information

On évite pour de multiples raisons (espace occupé, efficacité pour les recherches ou les modifications, ...) de dupliquer l'information présente dans une base de données. Cela conduit à l'utilisation de plusieurs tables **liées** entre elles, c'est à dire que certains attributs d'une table sont les clés primaires d'une autre table. On dit que ce sont des **clés étrangères**. Les clés étrangères sont précédées de # dans le schéma relationnel.

### Exemple

On souhaite modéliser un relevé de notes sur lequel figure :

## C8 Schéma relationnel d'une base de données

### Duplication de l'information

On évite pour de multiples raisons (espace occupé, efficacité pour les recherches ou les modifications, ...) de dupliquer l'information présente dans une base de données. Cela conduit à l'utilisation de plusieurs tables **liées** entre elles, c'est à dire que certains attributs d'une table sont les clés primaires d'une autre table. On dit que ce sont des **clés étrangères**. Les clés étrangères sont précédées de # dans le schéma relationnel.

### Exemple

On souhaite modéliser un relevé de notes sur lequel figure :

- Un élève (nom, prénom, date de naissance, et identifiant unique)

## C8 Schéma relationnel d'une base de données

### Duplication de l'information

On évite pour de multiples raisons (espace occupé, efficacité pour les recherches ou les modifications, ...) de dupliquer l'information présente dans une base de données. Cela conduit à l'utilisation de plusieurs tables **liées** entre elles, c'est à dire que certains attributs d'une table sont les clés primaires d'une autre table. On dit que ce sont des **clés étrangères**. Les clés étrangères sont précédées de # dans le schéma relationnel.

### Exemple

On souhaite modéliser un relevé de notes sur lequel figure :

- Un élève (nom, prénom, date de naissance, et identifiant unique)
- Un ensemble de matières fixées

## C8 Schéma relationnel d'une base de données

### Duplication de l'information

On évite pour de multiples raisons (espace occupé, efficacité pour les recherches ou les modifications, . . .) de dupliquer l'information présente dans une base de données. Cela conduit à l'utilisation de plusieurs tables **liées** entre elles, c'est à dire que certains attributs d'une table sont les clés primaires d'une autre table. On dit que ce sont des **clés étrangères**. Les clés étrangères sont précédées de # dans le schéma relationnel.

### Exemple

On souhaite modéliser un relevé de notes sur lequel figure :

- Un élève (nom, prénom, date de naissance, et identifiant unique)
- Un ensemble de matières fixées
- Au plus une note par matière et par élève

## C8 Schéma relationnel d'une base de données

### Duplication de l'information

On évite pour de multiples raisons (espace occupé, efficacité pour les recherches ou les modifications, ...) de dupliquer l'information présente dans une base de données. Cela conduit à l'utilisation de plusieurs tables **liées** entre elles, c'est à dire que certains attributs d'une table sont les clés primaires d'une autre table. On dit que ce sont des **clés étrangères**. Les clés étrangères sont précédées de # dans le schéma relationnel.

### Exemple

On souhaite modéliser un relevé de notes sur lequel figure :

- Un élève (nom, prénom, date de naissance, et identifiant unique)
- Un ensemble de matières fixées
- Au plus une note par matière et par élève
- Expliquer pourquoi un schéma relationnel d'une seule table notes n'est pas satisfaisant.

## C8 Schéma relationnel d'une base de données

### Duplication de l'information

On évite pour de multiples raisons (espace occupé, efficacité pour les recherches ou les modifications, ...) de dupliquer l'information présente dans une base de données. Cela conduit à l'utilisation de plusieurs tables **liées** entre elles, c'est à dire que certains attributs d'une table sont les clés primaires d'une autre table. On dit que ce sont des **clés étrangères**. Les clés étrangères sont précédées de # dans le schéma relationnel.

### Exemple

On souhaite modéliser un relevé de notes sur lequel figure :

- Un élève (nom, prénom, date de naissance, et identifiant unique)
- Un ensemble de matières fixées
- Au plus une note par matière et par élève
- Expliquer pourquoi un schéma relationnel d'une seule table notes n'est pas satisfaisant.
- Proposer un schéma relationnel constitué de 3 tables

## Intégrité référentielle

Dans une base de données, l'**intégrité référentielle**, assure qu'une clé étrangère est toujours présente en tant que clé primaire dans la table qu'elle référence. Cela assure la cohérence des données et empêche des suppressions par erreur de la part des utilisateurs.

## Exemple

### Intégrité référentielle

Dans une base de données, l'**intégrité référentielle**, assure qu'une clé étrangère est toujours présente en tant que clé primaire dans la table qu'elle référence. Cela assure la cohérence des données et empêche des suppressions par erreur de la part des utilisateurs.

### Exemple

Dans l'exemple précédent, la table des notes doit toujours faire référence à un élève et à une matière. L'intégrité référentielle préserve automatiquement la relation "une note appartient forcément à un élève" et "une note est donnée dans une matière existante"

## Jointure

- La **jointure** est l'opération permettant de récupérer des informations fragmentées sur plusieurs tables.

### Jointure

- La **jointure** est l'opération permettant de récupérer des informations fragmentées sur plusieurs tables.
- La syntaxe est la suivante :

### Jointure

- La **jointure** est l'opération permettant de récupérer des informations fragmentées sur plusieurs tables.

- La syntaxe est la suivante :

```
SELECT attr1,attr2, ... FROM table1 JOIN table2 ON table1.attr  
= table2.attr
```