

- ▶ Certaines applications nécessitent :
 - ▶ des langages de requêtes **riches** (jointure, agrégation)
 - ▶ une conformité aux propriétés **ACID**
 - ▶ mais des **performances supérieures** à celles des SGBD classiques

Limites des systèmes
classiques de gestion de bases
de données

SGBD relationnels
classiques : forces et
faiblesses

Systèmes NoSQL

Systèmes NewSQL

Pierre Senellart

NewSQL

- ▶ Certaines applications nécessitent :
 - ▶ des langages de requêtes **riches** (jointure, agrégation)
 - ▶ une conformité aux propriétés **ACID**
 - ▶ mais des **performances supérieures** à celles des SGBD classiques
- ▶ Solutions possibles :
 - ▶ Se débarrasser des **goulots d'étranglement** classiques des SGBD : verrous, journalisation, gestion des caches
 - ▶ Bases de données **en mémoire vive**, avec copie sur disque asynchrone
 - ▶ Une gestion de concurrence **sans verrou** (MVCC)
 - ▶ Une architecture distribuée sans partage d'information (**shared nothing**) et avec **équilibrage de charge** transparent



Limites des systèmes
classiques de gestion de bases
de données

SGBD relationnels
classiques : forces et
faiblesses

Systèmes NoSQL

Systèmes NewSQL

Pierre Senellart

Dans quels cas choisir un SGBD non classique ?

- Quand on a des besoins de **latence** ou de **débit extrêmes**

Limites des systèmes
classiques de gestion de bases
de données

SGBD relationnels
classiques : forces et
faiblesses

Systèmes NoSQL

Systèmes NewSQL

Pierre Senellart

Dans quels cas choisir un SGBD non classique ?

- ▶ Quand on a des besoins de **latence** ou de **débit extrêmes**
- ▶ Quand on a des **volumes** de données **extrêmes**

Limites des systèmes
classiques de gestion de bases
de données

SGBD relationnels
classiques : forces et
faiblesses

Systèmes NoSQL

Systèmes NewSQL

Pierre Senellart

Dans quels cas choisir un SGBD non classique ?

- ▶ Quand on a des besoins de **latence** ou de **débit extrêmes**
- ▶ Quand on a des **volumes** de données **extrêmes**
- ▶ Quand le modèle relationnel et SQL **se prêtent mal** au stockage et à l'accès aux données (pas si fréquent !)

Limites des systèmes
classiques de gestion de bases
de données

SGBD relationnels
classiques : forces et
faiblesses

Systèmes NoSQL

Systèmes NewSQL

Pierre Senellart

Dans quels cas choisir un SGBD non classique ?

- ▶ Quand on a des besoins de **latence** ou de **débit extrêmes**
- ▶ Quand on a des **volumes** de données **extrêmes**
- ▶ Quand le modèle relationnel et SQL **se prêtent mal** au stockage et à l'accès aux données (pas si fréquent !)
- ▶ Quand, après tests détaillés, les performances des SGBD classiques se révèlent **insuffisantes**

Limites des systèmes
classiques de gestion de bases
de données

SGBD relationnels
classiques : forces et
faiblesses

Systèmes NoSQL

Systèmes NewSQL

Pierre Senellart

Dans quels cas choisir un SGBD non classique ?

Limites des systèmes
classiques de gestion de bases
de données

SGBD relationnels
classiques : forces et
faiblesses

Systèmes NoSQL

Systèmes NewSQL

- ▶ Quand on a des besoins de **latence** ou de **débit extrêmes**
- ▶ Quand on a des **volumes** de données **extrêmes**
- ▶ Quand le modèle relationnel et SQL **se prêtent mal** au stockage et à l'accès aux données (pas si fréquent !)
- ▶ Quand, après tests détaillés, les performances des SGBD classiques se révèlent **insuffisantes**
- ▶ **Savoir ce qu'on perd** : ACID (suivant les cas), possibilité d'interrogations complexes, stabilité de logiciels bien établis. . .

Dans quels cas choisir un SGBD non classique ?

Limites des systèmes
classiques de gestion de bases
de données

SGBD relationnels
classiques : forces et
faiblesses

Systèmes NoSQL

Systèmes NewSQL

- ▶ Quand on a des besoins de **latence** ou de **débit extrêmes**
- ▶ Quand on a des **volumes** de données **extrêmes**
- ▶ Quand le modèle relationnel et SQL **se prêtent mal** au stockage et à l'accès aux données (pas si fréquent !)
- ▶ Quand, après tests détaillés, les performances des SGBD classiques se révèlent **insuffisantes**
- ▶ **Savoir ce qu'on perd** : ACID (suivant les cas), possibilité d'interrogations complexes, stabilité de logiciels bien établis. . .
- ▶ Les bases de données NoSQL et NewSQL répondent à de **vrais besoins**. . . mais les besoins sont **souvent surestimés**