

Bases de données

XPOSE – 13 Décembre 2011

SOMMAIRE

- ◉ Qu'est-ce qu'une base de données ?
- ◉ Comment modéliser une base de données ?
- ◉ Quelles sont les problématiques liées aux bases de données ?

Qu'est-ce qu'une base de données ?

Définition

« Une base de données est un ensemble structuré et cohérent de données enregistrées avec le minimum de redondance pour satisfaire simultanément plusieurs utilisateurs de manière sélective et dans un temps opportun. »

Olivier CURE

Pourquoi une base de données ?

- ◎ Intégration et corrélation
 - > Redondance
 - > Risque d'incohérences

- ◎ Flexibilité / Indépendance
 - > Indépendance physique
 - > Indépendance logique
 - > Indépendance vis-à-vis des stratégies d'accès

Pourquoi une base de données ?

- ◎ Disponibilité
 - > Concurrence
 - > Temps de réponse
- ◎ Sécurité
 - > Intégrité
 - > Confidentialité

Concept

- ◉ Microcosme
- ◉ Données non calculées
- ◉ Un langage de requêtes
- ◉ Dépendances fonctionnelles
- ◉ Transactionnel
- ◉ Décisionnel

Concept

- ◉ Tables
- ◉ Entités
- ◉ Colonnes
- ◉ Identifiant primaire
- ◉ Identifiant secondaire
- ◉ Identifiant minimal
- ◉ Références

Concept - ACID

- ◉ Atomique :
 - > Tout ou rien
- ◉ Consistante
 - > Etat stable et cohérent
- ◉ Isolée
 - > Pas d'interférences entre les transactions
- ◉ Durable
 - > Protégé contre les crash, les erreurs et les pertes

Pour quels types de base données?

SQL :

- ◉ Relationnel
- ◉ Objet
- ◉ XML
- ◉ Réseau
- ◉ Hiérarchique

Pour quels types de base données?

NoSQL ou NOSQL :

- ◉ *Clé / Valeur*
- ◉ *Graphe*
- ◉ *Document*
- ◉ *« Column family »*

Histoire et évolution

70 – 80 :
premières
bases de
données
relationnelles



80-90 : Mise en
place des
standards de
modélisation



90-2000 :
fusion du
décisionnel et
du
transactionnel



2000 –
Aujourd'hui :
NoSQL et
CLOUD

Comment modéliser une base de données ?

Catégories des modèles de données

- ◎ Haut niveau
 - > Modèles de données conceptuels
- ◎ Bas niveau
 - > Modèles de données physiques
- ◎ Intermédiaires
 - > Modèles de données représentationnels
 - > Modèles de données d'implémentation

Les étapes de la modélisation

- ◉ Cahier des charges
- ◉ Schéma conceptuel
- ◉ Schéma logique
- ◉ Ajustement du schéma
- ◉ Schéma physique
- ◉ Sécurisation

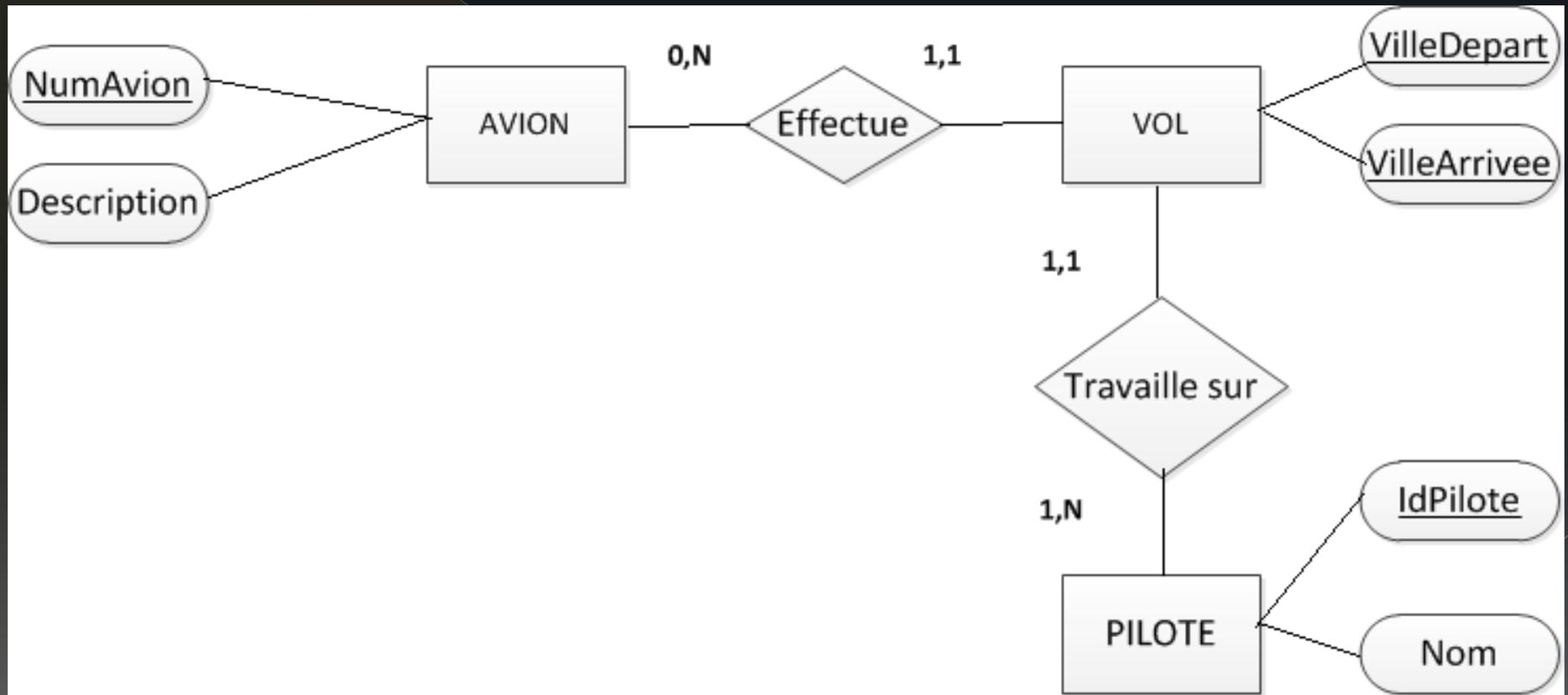
Quels sont les modèles les plus utilisés ?

- ◉ Modèle entités - associations
- ◉ Modèle de données relationnels

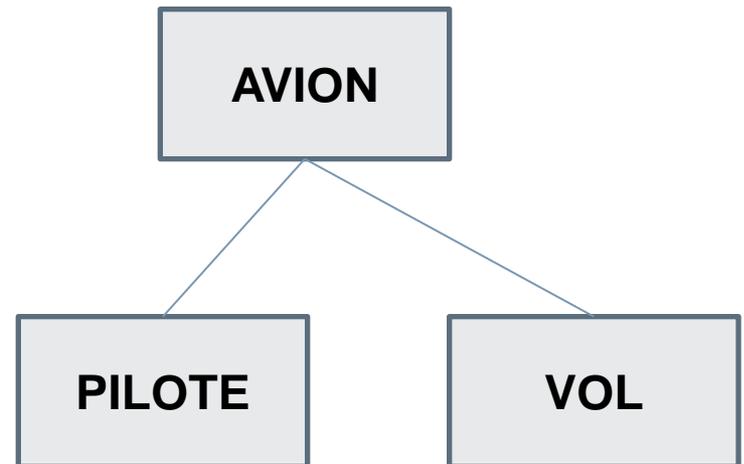
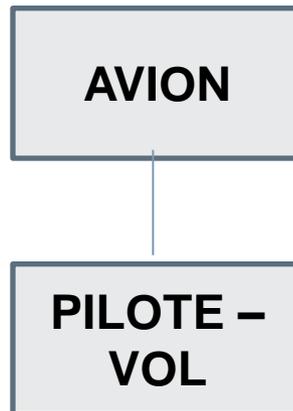
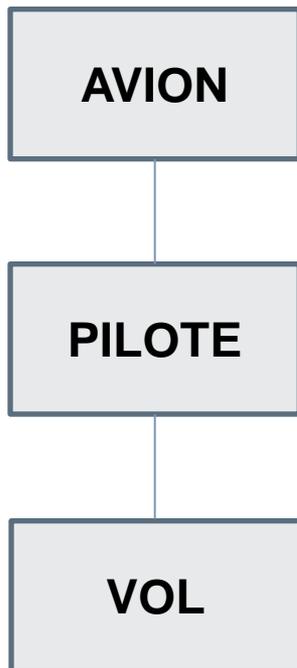
- ◉ Modèles hérités :
 - > Modèle hiérarchique
 - > Modèles en réseau

- ◉ Modèle de données par enregistrement
- ◉ Modèle de données objet

Modèle Entités - Associations



Modèle hiérarchique



Modèle relationnel

- AVION (NumAV, Description VilleArrivee)
- PILOTE (IDPilote, NOMPilote)
- VOL (VilleDepart, VilleArrivee, #IDPilote, #NumAV)

Forme Normale 1

- ⦿ Tous les attributs de la relation doivent contenir une valeur atomique.
- ⦿ Pilote(*ID*, Nom, Avions)
« 456, BOB, [A330, A380]

N'est pas une relation FN1.

Forme Normale 2

Une relation est de 2^{ème} forme normale :

- > si elle est de première forme normale.
- > Tout attribut non clé n'est pas une partie de clé

- Pilote (IDPilote, Nom, Licence, DateObtention) **=> N'est pas FN2**
- Pilote (IDPilote, Nom)
- LicencePilote (IDPilote, Licence, DateObtention)

Forme Normale 3

Une relation de 3^{ème} FN si :

- > Elle est 2^{ème} FN
- > Tout attribut n'appartenant pas à une clé ne dépend pas d'un autre attribut non clé
- Pilote (IDPilote, Nom, NumVol, VilleDepart, Ville Arrivée) => **N'est pas FN3**
- Pilote (IDPilote, Nom)
- Vol(NumVol, VilleDepart, Ville Arrivée)

Formes Normales 4/5/6

- ◉ Evite la redondance d'informations
- ◉ En pratique la FN 3 est suffisante
- ◉ Pour des raisons de performance (Projection, jointures), une dénormalisation est parfois nécessaire

Modèle Objet

- ◉ Historiquement Object Data Management Group
- ◉ Normalisation de l'OQL : Object Query Language
- ◉ Remplacé par le JDOQL : Java Data Object Query Language

JDOQL

- ◉ Même syntaxe que le Java pour les expressions.
- ◉ `Vol.class, Pilote.Nom== « Bob»`
- ◉ http://docs.oracle.com/cd/E13189_01/kodo/docs303/jdo_overview_query_jdoql.html
- ◉ <http://db.apache.org/jdo/jdoql.html>

**Quelles sont les problématiques
liées aux bases de données ?**

La multiplicité des types de données

- ◉ Données multimédias
- ◉ Données géographiques
- ◉ Données semi-structurées
- ◉ Comment gérer ces différents types de données ?

L'information incomplète

- ◉ Problème de qualité
- ◉ Rigueur et précision
- ◉ Comment produire de l'information correcte à partir de ces données ?

Le volume et les performances

- ◎ 500.000 enregistrements en 1970
- ◎ 90 TO ajoutés par jours sur Facebook
- ◎ Comment mémoriser, gérer et exploiter ces données dans des temps de réponse acceptable ?

Maintenance et évolution

- ⦿ Une base de donnée pour plusieurs milliers d'applications
- ⦿ Comment limiter les impacts lors d'une modification des structures de données ?

Les données distribuées et nomades

- ◉ Développement des bases de données distribuées
- ◉ Redondance et recopie des informations
- ◉ Comment garantir la cohérence des données ?

Le Web et les bases de données

- ◉ Le cloud computing
- ◉ Datastore
- ◉ Comment garantir la confidentialité et l'intégrité des données ?

Les bases de données décisionnelles

- ◉ Les DataWarehouse et l'informatique décisionnel.
- ◉ Comment lier le relationnel et le décisionnel ?

Les ORM

- ◉ Object relational mapping
- ◉ Modéliser les besoins utilisateurs en classe java
- ◉ Comment conserver l'évolutivité de la base de données et la qualité des données ?

Avez-vous des questions ?

Bibliographie

- ◉ Conception et architecture des bases de données – R. ELMASRI / S. NAVATHE – *PEARSON EDUCATION*
- ◉ Bases de données, Architectures, modèles relationnels et objets, SQL 3 – S. MIRANDA – *DUNOD*
- ◉ Bases de données, Concepts, utilisation et développement – J.L. HAINAUT – *DUNOD*
- ◉ Bases de données de la modélisation au SQL – L. AUDIBERT – *ELLIPSES*
- ◉ Les bases de données relationnelles – A. FLORY / F. LAFOREST – *ECONOMICA*
- ◉ Data Mining – R. LEFEBURE / G. VENTURI – *EYROLLES*
- ◉ Le Cloud Computing avec Amazon Web Services – J. BARR – *PEARSON*