

**Chapitre 2 et 3 – Objectifs**

**|9:41| : (Connaitre)**

But et origine de l’architecture a trois niveaux des BD

Contenus des niveaux externe, conceptuel et interne

Utilité correspondances externe-conceptuelle / conceptuelle-interne

Signification de l’indépendance physique et logique des données

Une classification des modèles de données

Le but et l’importance de la modélisation conceptuelle

Fonctions, services types et composants logiciels d’un SGBD

Signification et avantages de l’architecture client-serveur

Architecture 2-tiers, 3-tiers et n-tiers

Rôle et usage des moniteurs de traitement de transactions

Infonuagique (cloud)

**Objectifs de l’architecture èa trois niveaux ANSI-SPARC**

Utilisateurs doivent accéder mêmes informations

|9:42| : Tout les utilisateur doivent

Vue \_\_immunisée\_\_ aux changements autres vues

|9:42| : Ce qu’ils voient ne devrait pas être affecté parce que est modifié ailleurs.

L’Utilisateur n’est pas obligé de connaître les détails du stockage physique

|9:42| : L’utilisateur il a besoin que sa l’aide a travailler, il a pas besoin de savoir comment c’est structuré.

DBA peut changer structure \_\_stockage\_\_\_ BD sans affecter vues

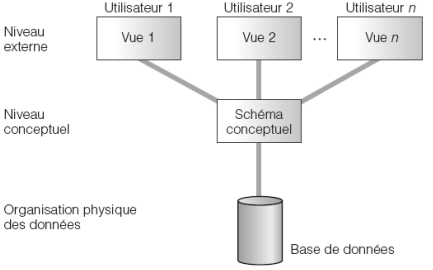
Structure interne pas affectée par changements niveau physique

|9:43| : Donc le niveau le plus si il y a changement fait a la structure physique le niveau interne n’est pas touché par cela.

DBA peut faire changements niveau \_\_\_conceptuel\_\_\_ sans affecter utilisateurs

|9:43| : Pourquoi on fait de l’orienté-objet pour faire abstraction des détails.

Architecture trois niveaux ANSI-SPARC



|9:44| : Chacun des utilisateurs voit quelque chose de la base de donnée mais ils voient seulement ce qu’ils ont besoin, pas plus pas moins. (Niveau Externe)

|9:45| : Au niveau conceptuel, c’est la que toute les données au niveau logique sont située

|9:45| : Organisation physique des données : Comment sont stocké les données au niveau physique. Des fois dans le niveau physique on va fusionner des tables ensembles. Sa va être des produits, des factures. Mais ils seront tous stocké au même endroit.

Indépendance des données

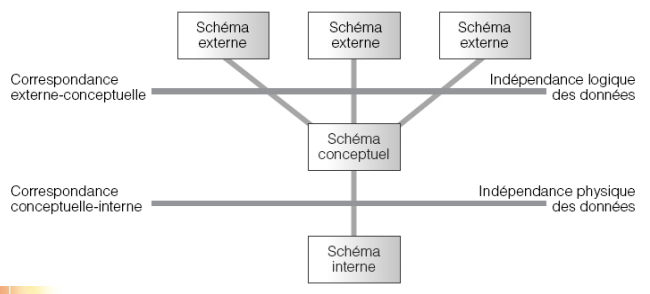
|9:46| : Ma vue ne devrait pas être brisé, changé juste parce qu’il y a un changement au niveau conceptuel. (Schéna conceptuel)

|9:47| : Si un utilisateur demande une information additionnel, le changement est voulu,

|9:47| : Si l’utilisateur 2 n’a pas besoin de ce changement, celui-ci ne l’aura pas.

|9:48| : Indépendance physique des données

|9:49| : Un client sa reste un client une facture sa reste une facture, même si un développeur fait de la séparation table, améliore la performance, il n’y aucun changement pour l’utilisateur on voit encore des factures et des clients.



Modèle de données

|9:50| : Ici faut le voir comme étant des types de modélisation de données

|9:50| : On parle au sens de général

Collection intégrée de \_\_\_concepts\_\_\_ qui décrivent et manipulent les données, les associations entre les données et les contraintes qui s’appliquent aux données dans une organisation

|9:50| : Modèle de classe, des modèle logique. On parle pas d’aller dans Visio et faire un modèle de donnée. On parle ici de différent type de modèle de données. Et partir de ceux-ci on peut créer des diagrammes. On parler de modélisation

Comprend :

Partie : \_\_\_Structurelle\_\_\_

Partie manipulation

Ensemble de règles d’intégrité

Fonction

Représenter données de façcon \_\_\_\_compréhensible\_\_\_\_

|9:51| : Tu a différent type de modèle de donnée et a partir de ceux-ci on veux éventuellement créer une base de données.

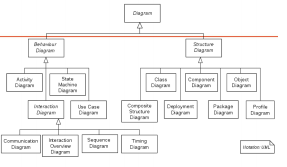
Types modèles de données

Modèles de données basés \_\_\_objet\_\_\_

|9:52| : On travaille avec des objets (des regroupements d’information)



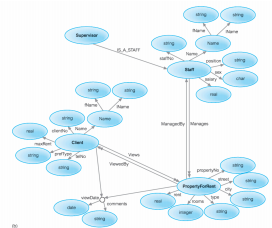
|9:52| : Ce modèle est un l’ensemble entité-association



|9:54| : Diagramme UML, ça reste des classes regroupés abrégés.



|9:53| : Modèle sémantique, par des dessins, représente des objet réel.

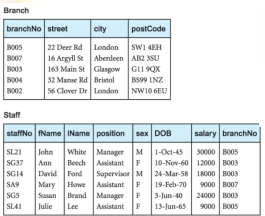


|9:53| : Modèle fonctionnel ou on vois les informations mais les liens entre eux, on voit des fonctions et ce qui catégorise les employés. On en parle pas plus que ça. C’est regroupé par fonctions.

Types modèles de données

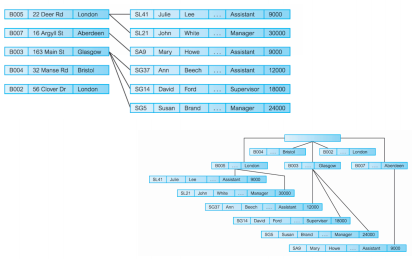
Modèle de données basés \_\_\_\_Enregistrement\_\_\_\_

|9:55| : Modèle relationnelle



|9:55| : Ce sont des modèles reliées.

|9:55| : On se retrouve plus au niveau de la base de données.



|9:56| : La succursale B005 elle a deux employés Julie et John, pour arriver a se resultat il faut suivre les pointeurs. C’est le modèle de données en réseau.

Modèles de données \_\_\_physiques\_\_\_

|9:57| : Il y en a pas beaucoup, c’est comment les données sont structuré sur le disque, Tu a frame memory et unified, Mais cela dépasse l’objectif du cours.

Modélisation conceptuelle

Schéma conceptuel

Cœur supportant \_\_\_\_vues\_\_\_\_ utilisateurs

Représentation exacte et complète besoins en informations

|10:0| : Faut qu’il contiennent toutes l’information qu’on a besoin.

Modélisation conceptuelle

Processus développement modèle d’utilisation information

\_\_\_\_indépendant\_\_\_\_ détails implémentation

|10:1| : Même si c’est indépendant, le tout fini presque toujours par devenir un modèle relationnelle, mais on devrait pas faire ça, on doit parler au client de façon conceptuel. Le client veut pas savoir c’est quoi une clé étrangère ou clé primaire.

Résulte modèle de données \_\_\_\_conceptuel\_\_\_\_

Fonctions d’un SGBD

|10:3| : un SGBD doit rempleir ses besoins :

Stockage, obtention et mise a jour

Founir un Catalogue

|10:3| : On parle de l’information sur l’information, les meta-données. La façon que l’information est structurée. c’est la définition des données.

Support \_\_\_\_transactions\_\_\_\_

|10:4| : c’est des qu’on a eux étapes pour faire une opération, ex : un virement bancaire, si tu fais un virement bancaire et tu fais juste un retrait ce n’est plus un virement bancaire. Il faut absolument que les deux opérations se fassent pour que la transaction ait lieu.

Services

Contrôle \_\_\_\_concurrence\_\_\_\_

|10:5| : Permettre l’accès concurrent sans briser les données

Récupération

\_\_\_\_Autorisation\_\_\_\_

|10:6| : Ce que vous le droit de faire et pas faire.

Communication données

|10:7| : Communication avec n serveur web pour envoyer les page générées. Ex : Requête PHP pour accéder a une base de données. Le SGBD doit fournir le service de communication.

Intégrité

|10:7| : Faut que les regle soient respecté.

Promotion indépendance données

|10:08| : Grace au catalogue, on va pouvoir promouvoir l’indépendance de données. C’est beaucoup plus facile d’accéder aux données. Si on vous donne un username et un mot de passe, même si je connais pas la structure je vais être capable d’aller chercher l’information. Et de l’afficher.

Utilitaires

|10:9| : Vont être la pour importer des données dans la base de données et vice versa pour les extraires. Des service statistiques qui vont nous aider a savoir ce qui est le plus en demande.

3 Architectures de SGBD multi-utilisateurs

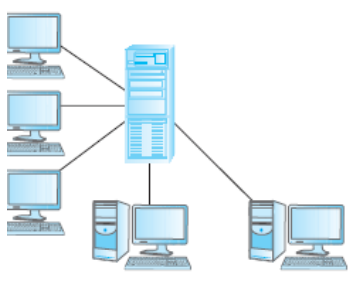
|10:10| : Histoire des SGBD en 3 architecture

\_\_\_\_Télétraitement\_\_\_\_

Architecture \_\_\_\_traditionnelle\_\_\_\_

Mainframe

Terminaux attachées



|10:10| : Un main frame au centre, dans le temps c’était la grosseur de 5 a 6 frigo. L’idée c’est que tout est centralisé. On a tout dans cet architecture. Tout s’exécute sur l’ordinateur principal.

|10:11| : En réalité on a un terminal attaché a un main frame.

|10:11| : Un terminal c’est quoi c’est un clavier et un écran, il n’a pas de RAM il n’y pas de processeur.

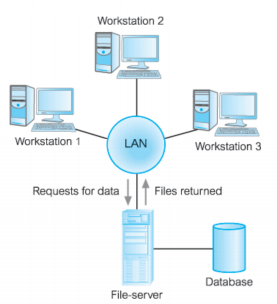
|10:12| : Imaginer que vous êtes 5 personnes tous branché sur la même tour a votre bureau, chacun utilise une partie de la puissance de la machine.

|10:13| : Tout ce passe sur le mainframe,

|10:13| : Avec l’événement des ordinateurs personnel, on s’est mis a simuler des terminals par une fenêtre noire, mais encore la les opérations sont encore fait au main-frame.

3 Architectures de SGBD multi-utilisateurs

Serveur de fichiers



|10:16| : On stocke la base de donnée sur le serveur, avec l’approche serveur de fichier.

|10:17| : On a tous mis chez le client, et la seule chose qu’on a laissé sur le serveur, c’est les fichiers. Les SGBD sont installé sur chaque machine et cela était problématique a cause du trafic de données.

3 Architectures de SGBD multiutilisateurs

Client-serveur

|10:19| : on peut accéder facilement aux bd déjà existante.

Avantages

Plus grand accès BD \_\_\_\_existantes\_\_\_\_

Augmente performance

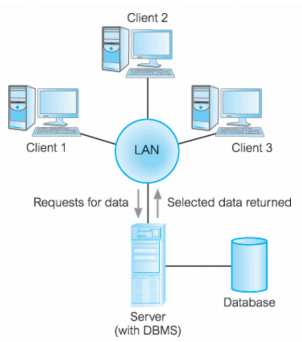
Réduction possible coûts matériel

|10:20| : parce que les client nont pas besoin d’etre plus puissant.

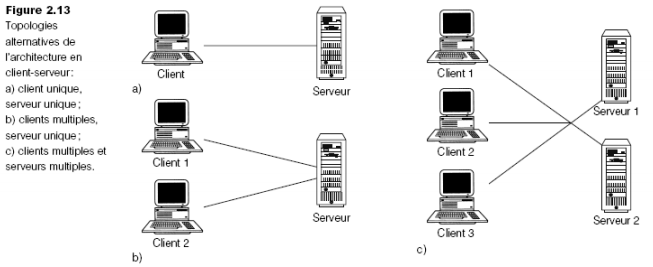
Réduction coûts communications

|10:20| : comme tu communique avec le SGBD,

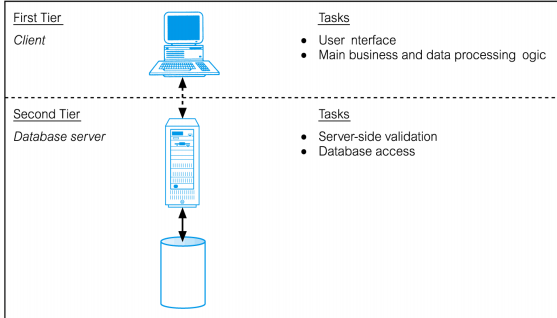
Augmente \_\_\_\_la cohérence\_\_\_\_



Topologies client-serveur alternatives



Client-serveur traditionnel 2-tier



Client-serveur 3-tier

Besoin \_\_\_\_a\_\_\_\_

Client –> 2 problèmes extensibilité

1995 –> 3 couches proposées

Avantages

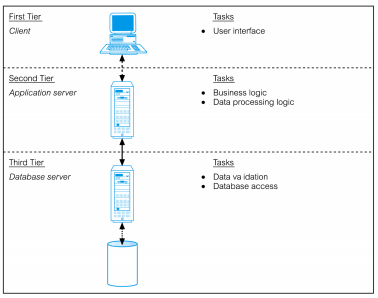
Client \_\_\_\_a\_\_\_\_

Maintenance application centralisée

Facilité remplacer 1 \_\_\_\_a\_\_\_\_

Séparation logique enteprise et fonctions BD

Appariement naturel \_\_\_\_a\_\_\_\_

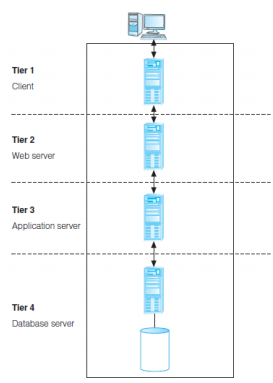


Client-serveur n-Tier (4-Tier)

3-tiers peut devenir n-tiers

 + \_\_\_\_a\_\_\_\_

Serveurs applications hébergent API de logique et processus d’affaires utilisable par autres applications.



Exercices

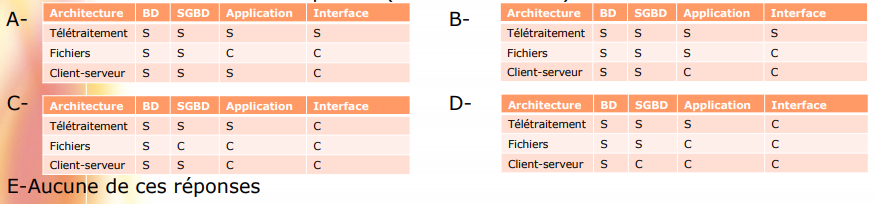
Cette fonction du SGBD permet de conserver des informations sur la structure des BD

A – Service communciation des données B – Service de l’indépendance des données

C – Support des transactions D – Catalogue

E – Toutes ces réponses F – Aucune de ces réponses

Ou se retrouve les différentes parties (Serveur ou Client) ?



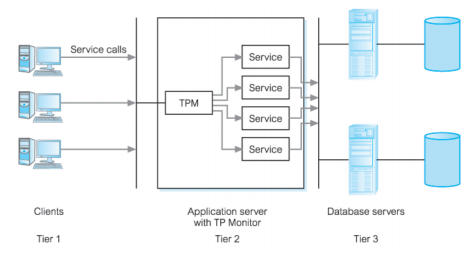
Middleware

Logiciel permettant communications netre applications disparates

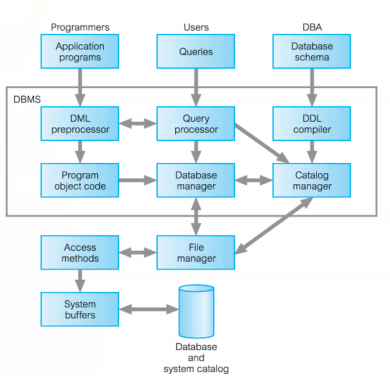
Par exemple MTT

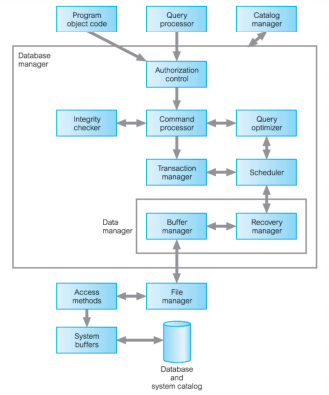
Moniteur de traitement de transaction

Programme contrôle \_\_\_\_a\_\_\_\_ données entre clients et serveurs fournissant environnement cohérent (Online Transaction Processing)



Composant SGBD





Infonuagique

Définition National Institute of Standards and Technology \_\_\_\_a\_\_\_\_

Modèle permettant accès réseau sur demande, pratique et omniprésent à un ensemble partagé de ressources informatiques configurables (réseaux, serveurs, stockages, application et services) qui peut rpaidement être provisionné et relâcheé avec effort de gestion/interaction fournisseur \_\_\_\_a\_\_\_\_

Caractéristique clés :

\_\_\_\_a\_\_\_\_ Service à la demande

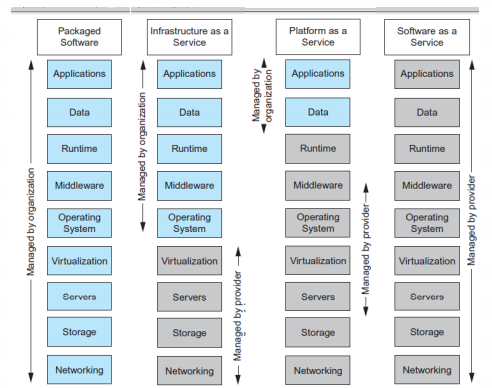
Acessible totalité du réseau

\_\_\_\_a\_\_\_\_ ressources

Rapidement \_\_\_\_a\_\_\_\_

Service \_\_\_\_a\_\_\_\_

Infonuagique – Modèle de service



Infonuagique

**Avantages**

* $
* \_\_\_\_a\_\_\_\_
* Sécurité améliorée
* \_\_\_\_a\_\_\_\_ améliorée
* Accès nouvelles \_\_\_\_a\_\_\_\_
* Développement + rapide
* Tests à grande \_\_\_\_a\_\_\_\_
* \_\_\_\_a\_\_\_\_ (mobile)
* Augmente compétivité

**Risques**

* Dépendance \_\_\_\_a\_\_\_\_
* Dépendance système
* Dépendance \_\_\_\_a\_\_\_\_ nuage
* Manque contrôle
* Manque information transparence traitement

Infonuagique – Solution BD

Saas

Data as a Service (DaaS)

Services permettent définition données dans nuage

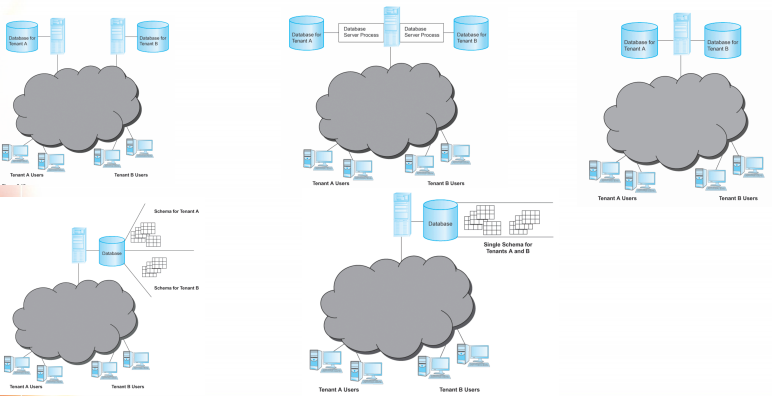
Implémente \_\_\_\_a\_\_\_\_ interfaces typiques SGBD (SQEL)  organisations ou public

Database as a Services (DbaaS)

\_\_\_\_a\_\_\_\_ BD complète développeurs

Fournit couche gestion optimisation

DbaaS – architectures



Exercice

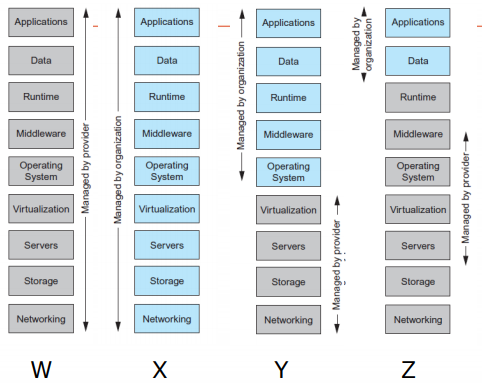
Associer les types de service infonuagique

1 – Sofware as a service

2 – Infrastructure as a service

3 – Platform as a srevice

4 – Package Sofware



A – 1W, 2Z, 3X, 4Y B – 1W, 2Y, 3Z, 4X

C – 1Y, 2W, 3X, 4Z D – 1Y, 2W, 3Z, 4X E – Aucune de ces réponses

Références

Images modèles de données page 7 tirées de Wikipédia

Autres images tirées du manuel de Connolly et Begg