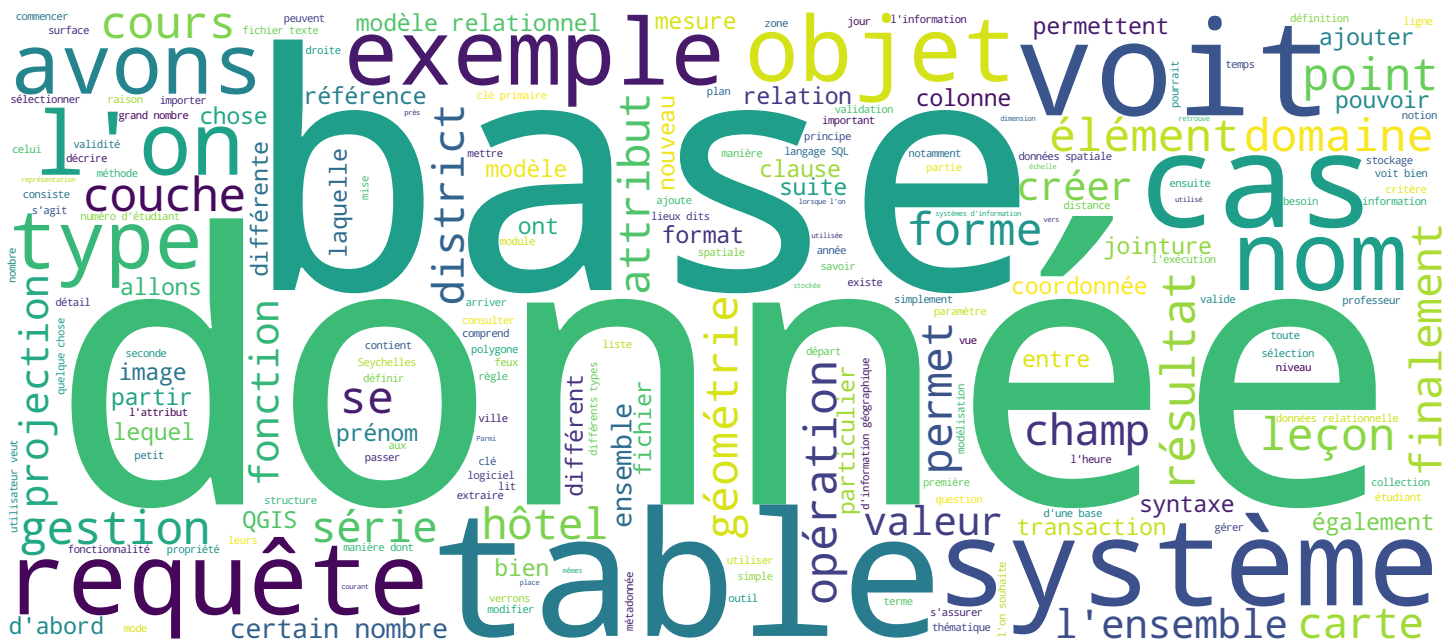


Bases de données relationnelles

Introduction aux systèmes d'information géographique

Stéphane Joost, Marc Soutter, Fernand Kouamé, Amadou Sall



Search MOOC



Video



Bases de données relationnelles

Objectifs de la leçon

- Découvrir les bases de données, et en particulier les bases de données relationnelles et spatiales

Après cette leçon vous serez capables

- de décrire comment une base de données relationnelle est organisée et à quoi correspondent les objets qu'elle contient

Introduction aux systèmes d'information géographique

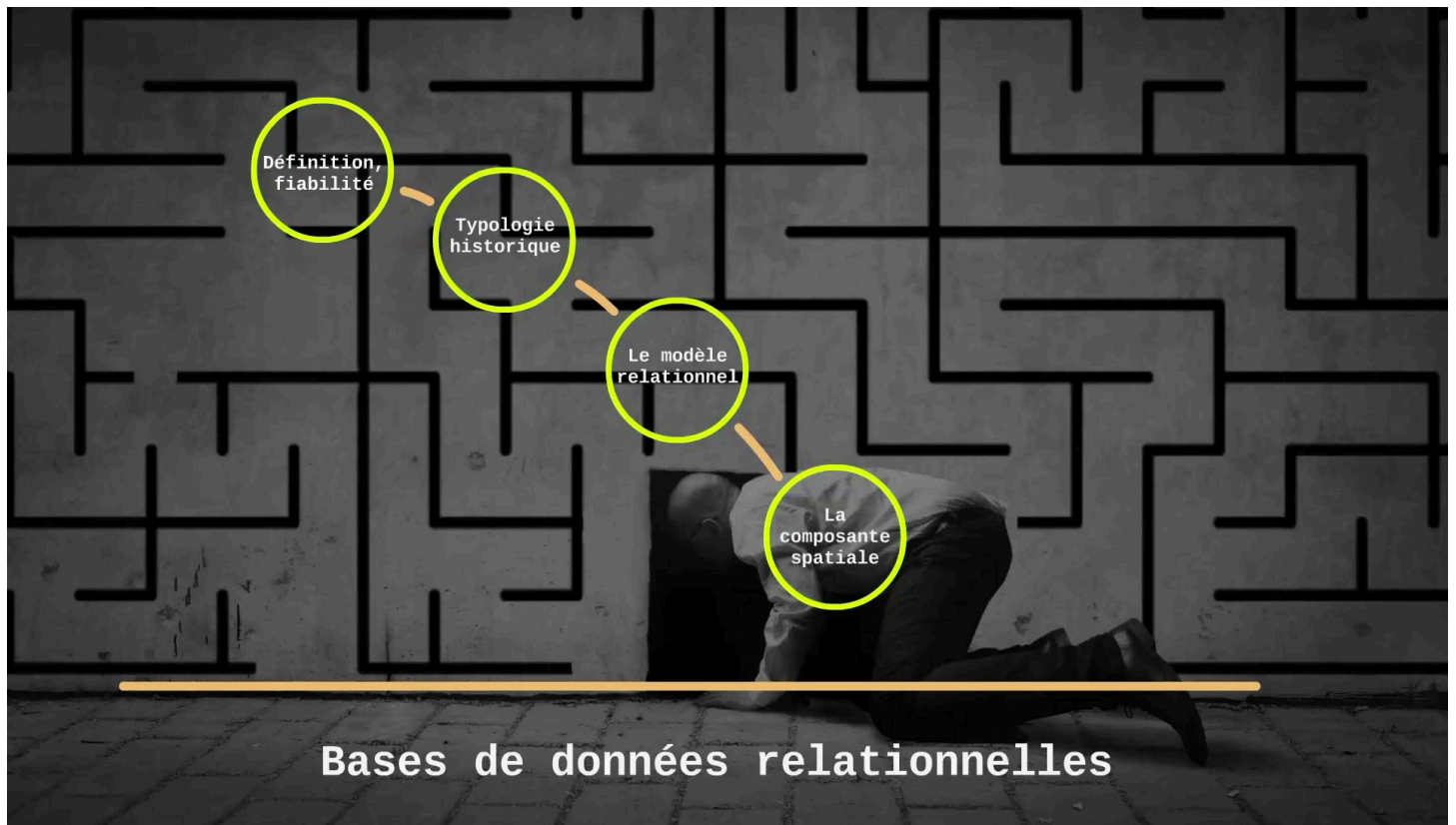
Bienvenue dans ce cours qui va porter sur les bases de données relationnelles qui sont la forme la plus largement répandue de base de données et que l'on retrouve dans absolument tous les domaines de la vie courante.

Notes

Summary



0m 22s



Les objectifs de cette leçon consistent donc à découvrir les bases de données en particulier les bases de données relationnelles et la manière dont la dimension spatiale est intégrée dans ces bases de données. Au terme de cette leçon vous devriez être capables de décrire les concepts qui fondent les bases de données relationnelles ainsi que les divers types d'objets qu'elles contiennent. Dans la leçon précédente nous avons vu plusieurs formes de stockage de données sous forme de fichiers simples ou de fichiers semi-structurés. Aujourd'hui, nous allons passer en revue la suite des possibilités de stockage de données à savoir les bases de données qui permettent un stockage plus structuré et la plupart du temps un stockage centralisé dans une architecture client-serveur. Dans cette leçon nous allons aborder la question de la définition d'une base de données les notions de fiabilité de ces bases de données. Nous ferons un bref survol historique de l'évolution des différents types de base de données. Nous nous concentrerons sur le modèle relationnel que nous verrons assez en détail. Nous parlerons ensuite de la manière dont la composante spatiale est gérée dans ces bases de données.

Notes

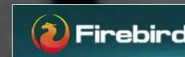
Summary



0m 34s

Définitions – base de données et SGBD

- Une **base de données** est une collection de données **persistantes**, centralisées ou non, servant pour les besoins d'une ou plusieurs applications, interrogeables et modifiables par un groupe d'utilisateurs travaillant en parallèle.
- Un **Système de gestion de bases de données** (SGBD) est un logiciel permettant de créer, structurer, alimenter, mettre à jour et consulter une base de données; c'est, en fait, l'interface entre la base de données et les utilisateurs ou leurs programmes.



Introduction aux systèmes d'information géographique

Et pour finir, un premier bref aperçu des logiciels SGBD, donc des logiciels Système de Gestion de Bases de Données, que nous verrons aussi à plusieurs reprises dans les cours suivants. Une base de données peut être définie comme une collection de données persistantes éventuellement centralisée utilisée par plusieurs applications et par plusieurs groupes d'utilisateurs qui peuvent éventuellement travailler en parallèle. On voit que avec cette définition très générale les fichiers d'hébergement de données que nous avons vu lors de la précédente leçon peuvent aussi être considérés comme des bases de données dans la mesure où elles offrent une certaine persistance au stockage d'information. Les systèmes de gestion de bases de données sont par contre des logiciels qui permettent de créer, de structurer, de documenter, de consulter les bases de données. Il s'agit en fait de l'interface qui existe entre la base de données et les utilisateurs ou les applications qui utilisent les bases de données. Comme le montre cette image il en existe un très grand nombre autant dans le domaine commercial que dans le domaine du logiciel libre.

Notes

Summary



1m 50s

Transactions et fiabilité

- Les modifications d'une BD sont le fruit de **transactions**
- Une **transaction** est une suite d'opérations, qui fait passer le BD d'un **état initial** à un **état final**
- Pour qu'une BD passe d'un état **cohérent** et **intègre** à un autre état **cohérent** et **intègre**, la transaction doit respecter 4 règles

Atomicité

Isolation

Cohérence

Durabilité

Propriétés ACID

Introduction aux systèmes d'information géographique

Les modifications d'une base de données sont le fruit de transactions. Les transactions étant une suite d'opérations qui font passer la base de données d'un état initial à un état final. Pour qu'une base de données passe d'un état initial intègre et cohérent à un état final intègre et cohérent, les transactions doivent respecter un certain nombre de critères qui sont l'atomicité, la cohérence l'isolation et la durabilité, critères que l'on résume souvent sous le nom de "propriété ACID" et que nous allons voir un peu plus en détail à présent.

Notes

Summary



3m 03s

ACID - Cohérence

T1 A: ↓5, B: ↑10, condition de validité $A + B = 100$



- Règles de cohérence = contraintes d'intégrité, p.ex $A+B = 100$ ou $A > 30$ etc.
- Si une règle de cohérence n'est pas respectée, la transaction est annulée

Introduction aux systèmes d'information géographique

Le principe d'atomicité stipule que la transaction se fait complètement ou pas du tout. Dans le cas ici d'une transaction qui comprend deux opérations, la première consistant à retrancher 10 à un ensemble A et la seconde à ajouter 10 à un ensemble B, avec pour condition de validité que $A + B = 100$, on voit que l'opération se fait en deux étapes pour arriver à un résultat qui est ensuite enregistré dans la base de données par une opération qui s'appelle un commit, une opération de validation. Il peut arriver que l'une des ces opérations échoue et dans ce cas-là le principe d'atomicité veut que la première opération de la transaction puisse être réversible et annulée donc qu'on puisse retourner en fait à l'état initial de la base de données. Les bases de données doivent vérifier des principes de cohérence. Si nous avons ici l'exemple d'une transaction qui consiste à retrancher 5 d'un ensemble A et ajouter 10 à un ensemble B avec une condition de validité $A + B = 100$ comme tout à l'heure, on voit bien qu'au terme de la transaction le résultat ne respecte plus la condition de validité si bien que la validation doit être empêchée et l'opération dans son ensemble annulée.

Notes

Summary



3m 41s

ACID - Durabilité

T1 A: ↓10, B: ↑10, condition de validité $A + B = 100$



- Enregistrement en mémoire de la transaction au moment de la validation
- Pas de passage par une mémoire tampon qui pourrait s'effacer en cas de coupure de courant par exemple

Introduction aux systèmes d'information géographique

Le principe d'isolation. On a ici le cas de deux transactions la première consistant à retrancher 10 d'un ensemble A et rajouter 10 à un ensemble B avec toujours la même condition de validité et une deuxième transaction qui fait à peu près la même opération. On retranche 5 de l'ensemble B et on ajoute 5 à l'ensemble A. Et en fait le principe d'isolation veut que l'exécution de ces transactions l'une après l'autre donne le même résultat que si les transactions sont exécutées en même temps c'est-à-dire en série. On voit que si l'une de ces opérations échoue, la dernière par exemple, il faut s'assurer que le principe d'annulation nous ramène la base de données dans un état valide ce qui est le cas dans le cas séquentiel. Mais par contre lorsque les opérations se font en série on voit qu'il faut qu'une procédure spéciale soit mise en place au niveau du système de gestion de bases de données pour s'assurer de l'exécution en série en cas de pépin pour remettre la base dans un état valide et cohérent. Finalement le principe de durabilité stipule que l'enregistrement en mémoire de la transaction au moment de la validation ne puisse pas être empêché ou interrompu par un événement extérieur.

Notes

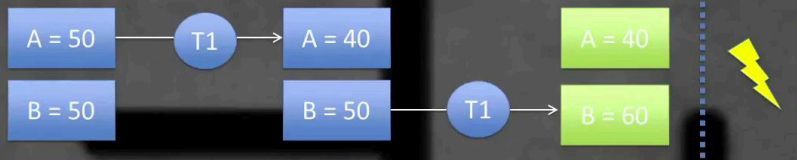
Summary



4m 59s

ACID - Durabilité

T1 A: ↓10, B: ↑10, condition de validité $A + B = 100$



- Enregistrement en mémoire de la transaction au moment de la validation
- Pas de passage par une mémoire tampon qui pourrait s'effacer en cas de coupure de courant par exemple

Introduction aux systèmes d'information géographique

Par exemple, une coupure de courant un tremblement de terre ou que sais-je. Cela signifie techniquement en fait qu'il faut éviter de passer par une mémoire tampon qui pourrait s'effacer en cas de coupure de courant avant la validation proprement dite.

Notes

Summary



Transaction et fiabilité

Propriétés ACID

- **Atomicité**
Tout les opérations ou aucune
- **Cohérence**
Vérification des contraintes d'intégrité
- **Isolation**
Exécution simultanée = exécution séquentielle
- **Durabilité**
Enregistrement direct sans passage par une mémoire-tampon

Introduction aux systèmes d'information géographique

En résumé, les principes ACID stipulent donc pour l'atomicité qu'une transaction doit être effectuée complètement ou pas du tout, pour la cohérence que l'on va vérifier au niveau de la base de données, les conditions de validité, le principe d'isolation, qu'une exécution en série donne le même résultat qu'une exécution séquentielle des différentes opérations d'une transaction. Et finalement le principe de durabilité stipule que des incidents externes, coupure de courant ou autre, n'affectent pas la manière dont l'information est stockée et validée dans la base de données.

Notes

Summary



6m 37s

Typologie historique des bases de données

dès années 1950-60,

- Stockage en **fichiers**

dès années 60-70,

- Base de données **hiérarchique**
- Base de données **réseau**

dès années 70-80,

- Base de données **relationnelle**

dès années 90,

- Base de données **objet**
- Base de données **semi-structurée**



- Basé sur la théorie des ensembles
- Enregistrements stockés dans des tables
- Chaque ligne correspond à un enregistrement
- Chaque colonne à un attribut

Introduction aux systèmes d'information géographique

Nous passons donc maintenant à la typologie historique des bases de données. Dès les années 50 et 60 s'est développé le stockage sous forme de fichiers, dont nous avons vu quelques exemples lors de la leçon précédente, puis à partir des années 60-70 se sont développées dans un premier temps les bases de données hiérarchiques. Les enregistrements sont associés par des relations selon une arborescence descendante. Chaque élément a un et un seul parent. Dans les domaines d'application, les structures d'organisation, les systèmes de fichier, les systèmes taxonomiques, etc. Par la suite les bases de données réseaux qui sont une variante des bases de données hiérarchiques avec simplement une multiplicité des parents possibles dont l'arborescence n'est plus strictement descendante. Il peut y avoir des structures cycliques. A partir des années 70-80 les bases de données relationnelles qui proposent sur le principe d'enregistrement hébergé dans des tables à deux dimensions. Une relation étant une dalle, un attribut se trouvant dans une colonne ou un champ et l'objet dans les lignes.

Notes

Summary



Typologie historique des bases de données

dès années 1950-60,

- Stockage en **fichiers**

dès années 60-70,

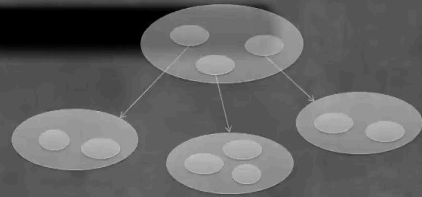
- Base de données **hiérarchique**
- Base de données **réseau**

dès années 70-80,

- Base de données **relationnelle**

dès années 90,

- Base de données **objet**
- Base de données **semi-structurée**



- Données stockées sous forme d'objets, c'est-à-dire de structures (classes) présentant des membres (instances).
- Divers types: clé-valeur, colonnes, documents, graphes
- BD semi-structurées ou XML

Introduction aux systèmes d'information géographique

Et à partir des années 90 les bases de données objets et les bases de données semi-structurées dans lesquelles les données sont stockées sous forme d'objets qui peuvent avoir des structures spécifiques et variables et de différents types. Nous parlerons dans cette leçon plus précisément des bases de données relationnelles et le thème des bases de données objets sera quant à lui abordé plus spécifiquement dans un cours ultérieur.

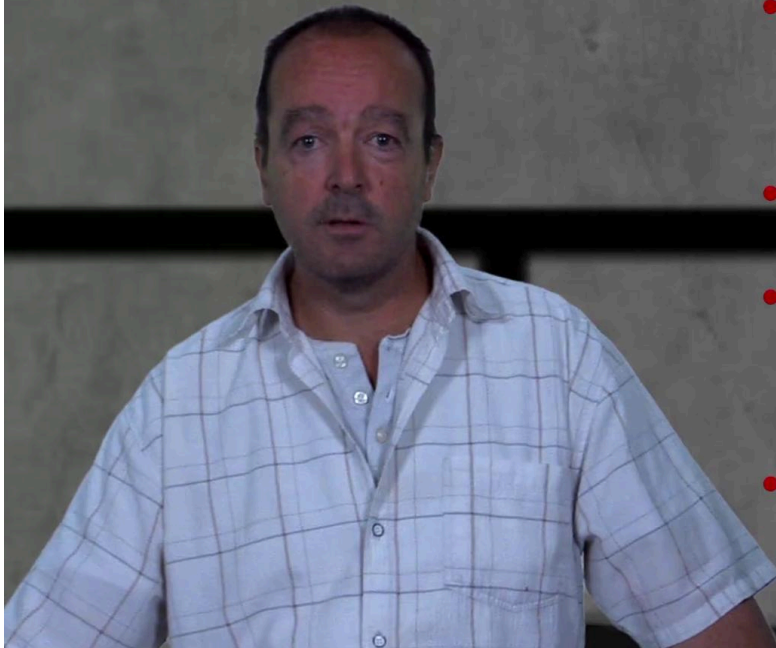
Notes

Summary



8m 29s

Le modèle relationnel – éléments généraux



- Objectifs:
 - indépendance du mode de stockage
 - cohérence des données
 - éviter les redondances
- Repose sur l'algèbre relationnelle, dérivé de la théorie des ensembles
- Modèle de loin le plus largement utilisé, notamment par toute une série de logiciels SGBD (Oracle, DB2, PostgreSQL, SQLite, etc.) et SIG (ArcGIS, Manifold, MapInfo)
- Développé dans les années 70 par Ted Codd chez IBM

Introduction aux systèmes d'information géographique

Le modèle relationnel a donc été développé essentiellement pour répondre aux principes ACID dans les transactions qui permettent de modifier une base de données, de s'assurer d'avoir une bonne cohérence de l'information, d'éviter la redondance des informations. C'est un modèle qui est fondé sur des bases théoriques solides puisque cela s'appuie sur la théorie des ensembles dont est tiré l'algèbre relationnel. C'est un modèle qui a été très largement utilisé par tous les grands acteurs du domaine des bases de données. Aujourd'hui, on peut dire qu'environ 80 à 90% des bases de données sont construites sur le modèle relationnel. C'est quelque chose qui change un petit peu avec le Big Data qui fait appel à d'autres types de modèles. Mais le modèle relationnel reste largement le plus important. C'est un modèle finalement qui a été développé à partir des années 70 chez IBM par un ingénieur du nom de Ted Codd.

Notes

Summary



9m 05s

Le modèle relationnel – Concept de base

- Les données sont organisées sous formes de tables, appelées relations.
- Une relation est constituée de colonnes ou attributs caractérisés par un nom et un domaine.
- Chaque ligne correspond à un enregistrement ou tuple.
- Un attribut peut ne pas être valué pour un tuple. Il a alors une valeur nulle marquée NULL.

Colonnes ou attributs

Table ou relation

ID	Name	Pop 2006	Veh1000	Start Date	End Date	Shape
3407	Aclens	389	871.47	1/1/2006	12/31/2006	<binary>
3518	Agiez	235	600.00	1/1/2006	12/31/2006	<binary>
3664	Aigle	8154	492.27	1/1/2006	12/31/2006	<binary>
3408	Allaman	401	648.38	1/1/2006	12/31/2006	<binary>
3360	Apples	1177	590.48	1/1/2006	12/31/2006	<binary>
3365	Arnex-sur-Nyon	108	648.15	1/1/2006	12/31/2006	<binary>
3501	Arnex-sur-Orbe	552	525.36	1/1/2006	12/31/2006	<binary>
3585	Arissoules	122	577.46	1/1/2006	12/31/2006	<binary>
3392	Arzier	2062	575.17	1/1/2006	12/31/2006	<binary>
3459	Assens	872	657.11	1/1/2006	12/31/2006	<binary>
3331	Aubonne	2668	636.06	1/1/2006	12/31/2006	<binary>
3683	Avenches	2753	582.27	1/1/2006	12/31/2006	<binary>
3394	Ballaigues	892	563.90	1/1/2006	12/31/2006	<binary>
3324	Ballens	414	570.05	1/1/2006	12/31/2006	<binary>
3419	Bassins	1047	571.16	1/1/2006	12/31/2006	<binary>
3441	Baulmes	959	523.46	1/1/2006	12/31/2006	<binary>
3470	Bavois	695	595.68	1/1/2006	12/31/2006	<binary>

Introduction aux systèmes d'information géographique

Dans le modèle relationnel les données sont donc organisées sous forme de tables également appelées "relation".

Notes

Summary



Le modèle relationnel – Concept de base

- Les données sont organisées sous formes de tables, appelées relations.
- Une relation est constituée de colonnes ou attributs caractérisés par un nom et un domaine.
- Chaque ligne correspond à un enregistrement ou tuple.
- Un attribut peut ne pas être valué pour un tuple. Il a alors une valeur nulle marquée NULL.



ID	Name	Pop 2006	Veh1000	Start Date	End Date	Shape
3407	Aclens	389	871.47	1/1/2006	12/31/2006	<binary>
3518	Agiez	235	600.00	1/1/2006	12/31/2006	<binary>
3664	Aigle	8154	492.27	1/1/2006	12/31/2006	<binary>
3408	Allaman	401	648.38	1/1/2006	12/31/2006	<binary>
3360	Apples	1177	590.48	1/1/2006	12/31/2006	<binary>
3365	Arnex-sur-Nyon	108	648.15	1/1/2006	12/31/2006	<binary>
3501	Arnex-sur-Orbe	552	525.36	1/1/2006	12/31/2006	<binary>
3585	Arrossoules	122	577.46	1/1/2006	12/31/2006	<binary>
3392	Arzier	2062	575.17	1/1/2006	12/31/2006	<binary>
3459	Assens	872	657.11	1/1/2006	12/31/2006	<binary>
3331	Aubonne	2668	636.06	1/1/2006	12/31/2006	<binary>
3683	Avenches	2753	582.27	1/1/2006	12/31/2006	<binary>
3394	Ballaigues	892	563.90	1/1/2006	12/31/2006	<binary>
3324	Ballens	414	570.05	1/1/2006	12/31/2006	<binary>
3419	Bassins	1047	571.16	1/1/2006	12/31/2006	<binary>
3441	Baulmes	959	523.46	1/1/2006	12/31/2006	<binary>
3470	Bavois	695	595.68	1/1/2006	12/31/2006	<binary>

Introduction aux systèmes d'information géographique

Une relation est constituée de colonnes ou d'attributs caractérisés par un nom et un domaine, un domaine étant un ensemble de valeurs, par exemple le domaine des valeurs entières, les domaines des booléens, le domaine des disciplines sportives, etc. Dans une table, chaque ligne correspond à un enregistrement également appelé "tuple".

Notes

Summary



Le modèle relationnel – Concept de base

- Les données sont organisées sous formes de tables, appelées relations.
- Une relation est constituée de colonnes ou attributs caractérisés par un nom et un domaine.
- Chaque ligne correspond à un enregistrement ou tuple.
- Un attribut peut ne pas être valué pour un tuple. Il a alors une valeur nulle marquée NULL.

Table attributaire - districts :: Total des entités : 25, filtrées : 25, sélectionnées : ...

	ID	NAME	PERIM_KM	AREA_KM2	SOURCE	COMMENT
0	46226	Baie Ste Anne	43.686354	22.479533	Praslin: District 2...	NULL
1	46227	Grand Anse Praslin	28.876896	15.087715	Praslin: District 2...	NULL
2	46203	La Digue	19.873296	9.808240	La Digue: Base.shp	NULL
3	46204	Takamaka	29.882000	14.217000	Mahe : District 2...	NULL
4	46205	St Louis	7.931000	1.383000	Mahe : District 2...	NULL
5	46206	Roche Caiman	14.599000	1.159000	Mahe : District 2...	NULL
6	46207	Port Glaud	45.310000	25.249000	Mahe : District 2...	NULL
7	46208	Pointe Larue	12.010000	3.340000	Mahe : District 2...	NULL
8	46209	Plaisance	10.454000	3.372000	Mahe : District 2...	NULL
9	46210	Mont Fleuri	9.429000	1.838000	Mahe : District 2...	NULL
10	46211	Mont Buxton	5.369000	1.165000	Mahe : District 2...	NULL
11	46212	Les Mamelles	6.827000	1.659000	Mahe : District 2...	NULL
12	46213	Grand Anse Mahe	23.802000	15.632000	Mahe : District 2...	NULL
13	46214	Glacis	17.488000	6.879000	Mahe : District 2...	NULL
14	46215	English River	7.612000	1.380000	Mahe : District 2...	NULL
15	46216	Cascade	24.513000	10.246000	Mahe : District 2...	NULL
16	46217	Bel Ombre	18.058000	9.409000	Mahe : District 2...	NULL

Montrer toutes les entités

Introduction aux systèmes d'information géographique

Pour un tuple il peut arriver qu'un attribut ne soit pas valué, donc qu'il n'ait pas de valeur et dans ce cas-là, on indique ce fait par la valeur NULL.

Notes

Summary



Identifiant = clé primaire

Retrouver n'importe quelle donnée à partir de 3 éléments: nom de table, no de colonne et identifiant.

Règles

- Un relation peut avoir plusieurs identifiants
- L'identifiant d'une relation est une collection d'attributs telle qu'il n'existe jamais 2 tuples ayant mêmes valeurs pour tous ces attributs
- Les attributs participant à un identifiant n'admettent pas de valeurs nulles

No Etud

Etudiants						
No Etud	Nom	Prénom	Rue	No	NPA	Localité
156	Dubois	André	Chemin des Mésanges	17	1260	Nyon
122	Balet	Lise	Rue de la Tine	2	1969	Suen
243	Gachet	John	Avenue de la gare	10	1400	Yverdon

Nom + Prénom

Etudiants						
No Etud	Nom	Prénom	Rue	No	NPA	Localité
156	Dubois	André	Chemin des Mésanges	17	1260	Nyon
122	Balet	Lise	Rue de la Tine	2	1969	Suen
243	Gachet	John	Avenue de la gare	10	1400	Yverdon

Nom + Prénom + Rue + No + NPA + Localité

Etudiants						
No Etud	Nom	Prénom	Rue	No	NPA	Localité
156	Dubois	André	Chemin des Mésanges	17	1260	Nyon
122	Balet	Lise	Rue de la Tine	2	1969	Suen
243	Gachet	John	Avenue de la gare	10	1400	Yverdon

Introduction aux systèmes d'information géographique

Un ensemble ne peut contenir deux fois le même objet et c'est pour cette raison que dans le modèle relationnel on doit absolument s'assurer que chaque objet est unique. La meilleure façon de s'assurer que cet objet est unique c'est de définir un identifiant spécifiquement pour les besoins comme dans le cas de la première table que l'on voit ici où on a créé un numéro d'étudiant qui permet d'identifier sans ambiguïté chaque étudiant. Cet identifiant, également parfois appelé "clé primaire", peut être fabriqué pour les besoins de la cause mais peut aussi être construit à partir des attributs existants par exemple en associant le nom et le prénom des étudiants. On voit bien que si on se contente du nom et du prénom on peut être confronté au problème des homonymes et avoir plusieurs personnes qui ne pourraient pas être distinguées par cet identifiant. D'où la nécessité d'étendre le concept et d'intégrer l'adresse, le nom de la rue, le numéro postal, etc. En fait, l'identifiant pourrait être constitué de l'ensemble des champs de la table avec évidemment, on imagine bien, un certain nombre d'inconvénients lorsqu'il s'agit d'indexer les objets pour pouvoir les retrouver plus vite lors d'une recherche ou bien pouvoir faire des tris et des choses comme ça. C'est pour cette raison que l'utilisation d'un identifiant spécifique est le cas de figure largement le plus souvent rencontré.

Notes

Summary



10m 44s

Identifiant externe = clé étrangère

Les identifiants externes décrivent les liens entre les relations

Exemple: suivi des cours par les étudiants

- Etudiants, Cours, Professeurs
- Un cours est donné par un professeur
- Dans la relation Suivi de cours, No Etud référence un étudiant (Etudiants.No Etud) et TitreC un cours (Cours.TitreC)

Règles

- Référence un tuple unique
- Peut éventuellement être NULL si l'attribut est facultatif, p.ex une salle de cours non définie

Professeurs			
ID	Nom	Prénom	Adresse
1	Simon	Jules	Place Neuve 5, 1350 Cossonay
2	Musy	André	Impasse du souvenir, 2012 Domdidier

Etudiants	
No Etud	Nom
156	Dubois
122	Balet
243	Gachet

Cours			
TitreC	Horaire	Professeur	[Salle]
Hydrologie	Ma 10-12h	Musy	GR A32
Matériaux	Me 13-15h	Simon	NULL
Irrigation	Ve 8-10h	Musy	NULL

Suivi	
No Etud	TitreC
156	Hydrologie
122	Hydrologie
243	Matériaux

Salles	
Nom	Responsable
GR C01	Monnet
CM 112	Mermet
GR A32	Monnet

Introduction aux systèmes d'information géographique

Les identifiants externes, également appelés "clés étrangères", décrivent des liens entre différentes relations, par exemple, dans le cas d'un cours suivi par des étudiants on a une table d'étudiants, une table de cours, une table de professeurs et on voit qu'un cours est donné par un professeur et que ce professeur en fait doit exister dans la table des professeurs pour que la base de données soit intègre. Dans la relation suivi de cours, qui associe un numéro d'étudiant avec un titre de cours, donc la liste des cours suivis par les étudiants ou la liste des étudiants qui suivent un cours, on voit que le numéro d'étudiant référence un étudiant de la relation étudiant et "titre cours" référence un cours dans la table des cours. La référence doit pointer sur un objet unique évidemment et il peut arriver que cet objet soit NULL si l'attribut est facultatif, par exemple dans le cas de la salle de cours qui pourrait ne pas avoir encore été attribuée au moment où on documente la table.

Notes

Summary



12m 26s

Vérification de l'intégrité référentielle

- Effectuée automatiquement, par le SGBD
- Si un utilisateur veut insérer un tuple dans Suivi avec un TitreC qui n'existe pas dans Cours

Refus

- Si un utilisateur veut modifier le nom du cours d'un tuple dans Suivi avec un TitreC qui n'existe pas dans Cours

Refus

Cours			
TitreC	Horaire	Professeur	[Salle]
Hydrologie	Ma 10-12h	Musy	GR A32
Matériaux	Me 13-15h	Simon	NULL
Irrigation	Ve 8-10h	Musy	NULL

Suivi	
No Etud	TitreC
156	Hydrologie
122	Hydrologie 2
243	Matériaux

Introduction aux systèmes d'information géographique

L'intégrité référentielle de la base de données est en principe vérifiée automatiquement par le système de gestion de bases de données. Si par exemple un utilisateur veut assurer un nouveau cours de géologie dans la table des cours suivis par les étudiants et que ce cours n'existe pas dans la table cours, la transaction sera refusée. De même, si un utilisateur veut changer le nom d'un cours et que ce nom n'existe pas dans la table de référence, la transaction sera également refusée.

Notes

Summary



13m 34s

Contraintes – Attribut complexe ou multivalué

Les notions d'attribut multivalué ou complexe n'existent pas dans le modèle relationnel. Il faut donc les modéliser autrement.

- Pour un attribut monovalué complexe, il faut choisir entre le composé ou les composants
- Pour un attribut multivalué, il faut créer une autre relation

Adresse : Rue , No, NPA, Localité

- un attribut par composant

Etudiants						
No Etud	Nom	Prénom	Rue	No	NPA	Localité
156	Dubois	André	Chemin des Mésanges	17	1260	Nyon
122	Balet	Lise	Rue de la Tine	2	1969	Suen
243	Gachet	John	Avenue de la gare	10	1400	Yverdon

il est possible de définir une vue restituant la notion globale d'adresse

- Un attribut Adresse, de domaine «chaîne de caractères»

Etudiants			
No Etud	Nom	Prénom	Adresse
156	Dubois	André	Chemin des Mésanges 17, 1260 Nyon
122	Balet	Lise	Rue de la Tine 2, 1969 Suen
243	Gachet	John	Avenue de la gare 10, 1400 Yverdon

le système ignore Rue, No, NPA et Localité

No Etud est un identifiant dans la relation Etudiants

Introduction aux systèmes d'information géographique

De même, si un utilisateur veut supprimer un cours qui est référencé par d'autres relations, l'opération peut être refusée. On peut supprimer les références dans l'autre relation ou alternativement annuler ces références. Pareillement, si un utilisateur veut modifier le nom d'un cours dans une table et que ce nom est référencé ailleurs on peut refuser l'opération ou mettre à jour l'autre table également. Le modèle relationnel ne se prête pas à un enregistrement d'attributs multivalués, donc qui ont plusieurs valeurs ou d'attributs complexes, donc composés de plusieurs éléments. Il est de ce fait nécessaire de les modéliser autrement. Dans le cas d'un attribut monovalué complexe, comme par exemple une adresse qui est composée d'une rue, d'un numéro dans la rue, d'un numéro postal, d'une localité, une possibilité consiste à définir un attribut, un champ par composant, comme illustré par la table qui se trouve ici à droite, et par la suite de construire dans la base de données une vue, qui est une table virtuelle contenant en fait l'agrégation de ces différents champs pour restituer l'idée d'adresse.

Notes

Summary

14m 04s



Contraintes – Attribut complexe ou multivalué

Les notions d'attribut multivalué ou complexe n'existent pas dans le modèle relationnel. Il faut donc les modéliser autrement.

- Pour un attribut monovalué complexe, il faut choisir entre le composé ou les composants
- Pour un attribut multivalué, il faut créer une autre relation

Prénoms multiples

- Plusieurs attributs >> mauvais choix

Etudiants				
No Etud	Nom	Prénom 1	Prénom 2	Prénom ...
156	Dubois	André	Louis	NULL
122	Balet	Lise	NULL	NULL
243	Gachet	John	André	NULL

- Relation / table supplémentaire

Etudiants		EtudPrénoms	
No Etud	Nom	No Etud	Prénom
156	Dubois	156	André
122	Balet	156	Louis
243	Gachet	122	Lise
		243	John
		243	André

No Etud est un identifiant pour Etudiants et un identifiant externe pour EtudPrénoms

Introduction aux systèmes d'information géographique

L'autre possibilité consiste à créer un attribut "adresse globale" qui serait de type chaîne de caractère dans lequel on va enregistrer l'ensemble de l'adresse, chemin, numéro, numéro postal, localité. On voit que dans ces relations le numéro d'étudiant est un identifiant de la relation étudiant. Pour un attribut multivalué, comme par exemple le cas de prénoms multiples, une possibilité serait de définir plusieurs attributs pour chaque prénom, prénom 1, prénom 2, etc. C'est un mauvais choix parce qu'en fait on ne sait pas combien de prénoms il faut prévoir, si c'est pas des prénoms ça peut être un autre type d'attribut qui pourrait avoir une succession innombrable d'éléments donc on voit bien que ça ne fonctionne pas très bien. D'autant plus que cela conduit à définir beaucoup de champs qui seront remplis de valeurs nulles. C'est pas quelque chose qui est très optimal du point de vue base de données. Une solution alternative consiste à créer une table supplémentaire dans laquelle vont être hébergés les prénoms en relation avec l'identifiant des étudiants auxquels ces prénoms se rattachent et donc on voit que le champ "numéro étudiant", qui est une clé primaire de la table étudiant, devient une clé étrangère de la table "étude prénom".

Notes

Summary



15m 27s

Contraintes – Attribut complexe ou multivalué

Les notions d'attribut multivalué ou complexe n'existent pas dans le modèle relationnel. Il faut donc les modéliser autrement.

- Pour un attribut monovalué complexe, il faut choisir entre le composé ou les composants
- Pour un attribut multivalué, il faut créer une autre relation

Introduction aux systèmes d'information géographique

Une autre possibilité un peu plus subtile consiste à intégrer dans la table "étude prénom" le numéro du prénom de sorte à pouvoir restituer l'ordre dans lequel les prénoms apparaissent, premier prénom, second prénom, etc. Dans ce cas également, le numéro d'étudiant est une clé primaire pour la table "étudiant" et une clé étrangère pour la table "étude prénom" qui elle n'a pas de clé primaire.

Notes

Summary

16m 58s



Algèbre relationnelle

Notions de base

- domaine, attribut, relation, tuple, identifiant, identifiant externe

Opérateurs

- Union
- Différence
- Produit croisé
- Sélection
- Projection
- Jointure

Ville	Auteur	Année
Aden	John	2001
Dakar	Pierre	2008
Oran	Roger	2005

\

Ville	Auteur	Année
Dakar	Pierre	2008
Bobo	Henri	2001

=

Ville	Auteur	Année
Aden	John	2001
Oran	Roger	2005

Ville	Auteur	Année
Dakar	Pierre	2008
Bobo	Henri	2001

\

Ville	Auteur	Année
Aden	John	2001
Dakar	Pierre	2008
Oran	Roger	2005

=

Ville	Auteur	Année
Bobo	Henri	2001

Introduction aux systèmes d'information géographique

Nous avons donc passé en revue les principales notions de base du modèle relationnel à savoir le domaine de valeurs l'attribut, la relation, le tuple l'identifiant, l'identifiant externe. Il nous reste encore à évoquer les différents opérateurs de l'algèbre relationnel qui fondent les opérations que l'on va pouvoir par la suite faire sur ces tables, opérations que nous verrons de manière assez détaillée dans la deuxième semaine de ce module avec tout ce qui est requête, langage SQL, etc. Ces opérateurs sont en premier lieu l'opérateur d'unions, donc on est dans une logique ensembliste. Pour l'exemple, on prend ici une série de photographies qui ont été réalisées par un auteur, une certaine année, dans une certaine ville, et on voit que l'union de deux tables consiste en fait à agréger ces deux tables en une seule. La différence entre deux tables est la soustraction de la seconde à la première donc l'élimination des objets redondants de la première. Cette opération n'est pas commutative, si on la fait dans l'autre sens le résultat va être différent.

Notes

Summary



17m 28s

Algèbre relationnelle

Notions de base

- domaine, attribut, relation, tuple, identifiant, identifiant externe

Opérateurs

- Union
- Différence
- Produit croisé
- Sélection
- Projection
- Jointure

Ville	Auteur	Année	Version
Aden	John	2001	Couleur
Dakar	Pierre	2008	Couleur
Oran	Roger	2005	Couleur
Aden	John	2001	N/B
Dakar	Pierre	2008	N/B
Oran	Roger	2005	N/B

=

Ville	Auteur	Année
Aden	John	2001
Dakar	Pierre	2008
Oran	Roger	2005
Aden	John	2001
Dakar	Pierre	2008
Oran	Roger	2005

Introduction aux systèmes d'information géographique

Nous avons ensuite le produit croisé qui consiste à associer chaque élément chaque tuple de la première table à chaque tuple de la seconde. Donc ici, les version couleurs et noir / blanc des photos qui... nous avons donc 2 fois 3, 6 photos. Et puis les opérations de sélection sur une table, sélection qui consiste à identifier, à extraire un certain nombre d'éléments. Des opérations de projection qui consistent à extraire un certain nombre d'attributs pour l'ensemble de la table. Ce qui dans le cas précis débouche sur quelque chose qui n'a pas grand sens, en tout cas qui est contraire aux principes de la théorie des ensembles puisqu'on a des objets identiques. Donc c'est une opération qui ne serait pas valide.

Notes

Summary



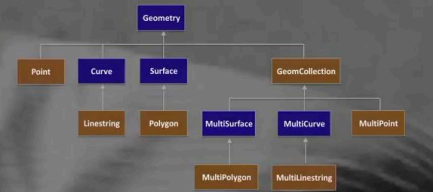
18m 59s

Bases de données spatiales

Base de données optimisée afin d'héberger et de traiter des géodonnées: stockage, indexation, requêtes

- Type de données (domaines)
- Indexation
- Fonctions

BD Standard	BD Spatiale
varchar	Point
integer	Linestring
real	Polygon
date	Multipoint
	Multilinestring
	Multipolygon



Type	Exemples
Point	POINT (30 10)
LineString	LINESTRING (30 10, 10 30, 40 40)
Polygon	POLYGON ((30 10, 10 20, 20 30, 40 30, 30 10)) POLYGON ((35 10, 10 20, 15 40, 40 40, 35 10), (20 30, 35 35, 30 20, 20 30))
Multipoint	MULTIPOINT ((10 40), (40 30), (20 20), (30 10)) MULTIPOINT (10 40, 40 30, 20 20, 30 10)
MultiLineString	MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))
MultiPolygon	MULTIPOLYGON (((30 20, 10 40, 10 40, 45 40, 30 20), (15 5, 40 10, 10 20, 3 10, 15 5)), ((20 35, 45 20, 30 5, 10 10, 10 30, 20 35), (30 20, 20 25, 20 15, 30 20)))

Introduction aux systèmes d'information géographique

Et finalement les opérations de jointure qui consistent à associer deux tables par l'intermédiaire d'un champ qui leur serait commun. Ici, la table des photographies couleurs avec une table qui associerait villes et pays. Le champ commun étant évidemment la ville. Ces opérations de jointure peuvent être assorties d'une condition, donc on pourrait ne garder que les éléments jointifs qui ont pour pays le Sénégal et donc une seule photographie sur les trois qui étaient comprises dans la jointure de départ. Nous en venons maintenant aux spécificités des bases de données qui possèdent une composante spatiale spécificités qui sont au nombre de trois principalement. Tout d'abord, le type de données, donc les domaines concernant ces types de données. On voit que dans la table en haut à gauche, dans une base de données traditionnelle on a un certain nombre de type de données bien défini, varchar pour le texte, les entiers pour le numérique, le réel également pour les nombres et les dates. Dans la base de données spatiale on ajoute de nouveaux types de données, de type point, ligne, polygone, etc. Enfin, les géométries de base qui sont gérées par les systèmes d'information géographique.

Notes

Summary



19m 53s

Bases de données spatiales - Indexation

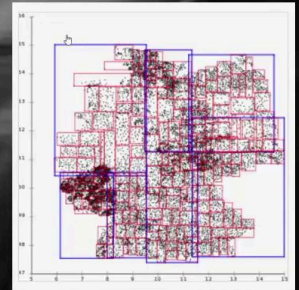
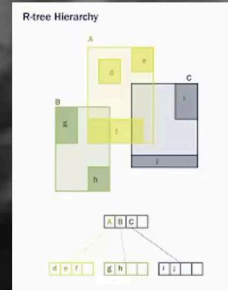
BD non spatiale

- B-tree: arborescence hiérarchique
- Hash table: dictionnaire clé-valeur

BD spatiale

- R-tree
Transposition au objets spatiaux de l'idée d'arborescence hiérarchique

- Les objets proches sont regroupés et représentés au niveau supérieur par leur enveloppe minimale (minimum bounding rectangle)
Si une requête spatiale n'intersecte pas une enveloppe minimale, elle n'intersectera pas non plus les objets qu'elle représente, ce qui permet de décider qu'une requête ne doit pas être poursuivie dans cet embranchement



Introduction aux systèmes d'information géographique

Le second domaine important est celui de l'indexation, l'indexation des objets spatiaux pour pouvoir par la suite effectuer des requêtes et retrouver rapidement les objets et finalement un certain nombre de fonctions qui permettent d'effectuer des opérations spécifiques sur les objets géométriques. Il existe plusieurs formes d'indexation des données. Dans le domaine des bases de données non-spatiales on utilise souvent une structure appelée B-tree, donc une arborescence hiérarchique qui permet en fait de retrouver facilement les données. On a dans cette illustration un exemple d'un arbre d'ordre 5 dans lequel un noeud peut avoir au plus 4 clés et 5 enfants, chaque noeud ayant au moins 2 clés et 3 enfants. Cette idée d'arborescence hiérarchique se retrouve dans le domaine des bases de données spatiales sous la forme de R-tree dans lequel les objets les plus proches sont regroupés et représentés à l'échelle supérieure par une enveloppe minimale qui est le minimum bounding rectangle. Et lorsque l'on fait ensuite une requête pour retrouver des objets en fait si la requête n'intersecte pas une enveloppe minimale on peut écarter tous les objets de cette enveloppe et se concentrer sur ceux qui restent en jeu.

Notes

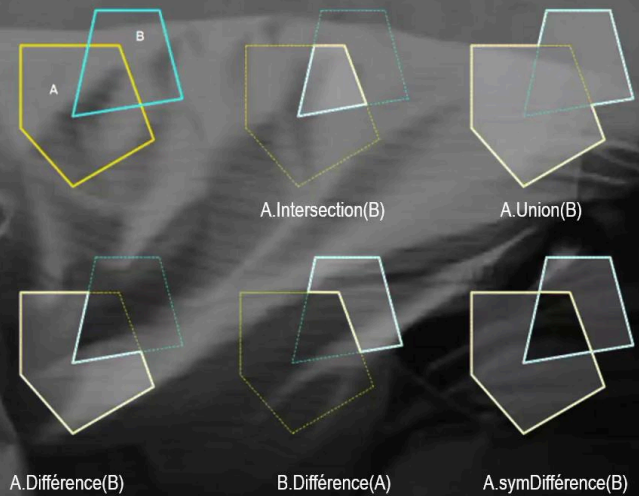
Summary



21m 35s

Bases de données spatiales – Fonctions

- Mesures spatiales
Longueur, Surface, Distance, etc.
- Fonctions spatiales (créent des entités)
Zone tampon, Intersection, Union, Différence, etc.
- Opérateurs topologiques (testent des relations topologique et retournent un booléen)
Disjoint, Touche, Contient
- Etc.



Introduction aux systèmes d'information géographique

Dans l'illustration qui est ici on voit à gauche le lien entre une approche hiérarchique non-spatiale et une approche hiérarchique spatiale et puis à droite un exemple de structuration d'index spatial qui concerne les bureaux de poste en Allemagne. Le Quadtree est une autre méthode d'indexation spatiale qui est pas mal utilisée dans les systèmes de tuilage de cartes géographiques type Google, etc. Il s'agit d'une structure arborescente dans laquelle chaque noeud possède exactement 4 enfants donc chaque zone géographique est divisée en 4 et à nouveau en 4 chaque fois que l'on descend d'un niveau de zoom. Le mode d'indexation en grille est similaire dans le sens qu'il s'agit aussi d'une tessellation régulière simplement que la subdivision de chaque entité ne se fait pas nécessairement en 4 mais peut se faire en 9, en 16, etc. Parmi les fonctions spécifiques aux bases de données spatiales on peut distinguer trois familles principales. Tout d'abord la famille des mesures spatiales qui donne des indications de longueur de surface, de distance à propos des objets géométriques. Des fonctions spatiales qui créent de nouvelles entités, par exemple on peut imaginer la création d'une zone tampon autour d'un objet géométrique.

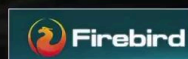
Notes

Summary



23m 05s

Logiciels SGBD – Bases de données relationnelles



Introduction aux systèmes d'information géographique

L'intersection de deux objets qui crée un nouvel objet, l'union de deux objets, etc. Et des opérateurs topologiques qui testent la véracité de relations de voisinage. Est-ce que deux géométries se chevauchent ? Est-ce qu'elles se touchent ? Est-ce qu'elles sont contenues l'une dans l'autre ? Ou vice versa, etc. Tous ces différents types de fonctions sont de plus en plus souvent implémentés dans les systèmes de gestion de bases de données spatiales. Comme je l'ai dit au début de ce cours, il existe un très grand nombre de systèmes de gestion de bases de données consacrés au modèle relationnel non seulement il existe un très grand nombre de bases de données mais ces bases de données sont accessibles par l'intermédiaire de toute une série de clients qui sont soit des logiciels commerciaux soit des logiciels libres.

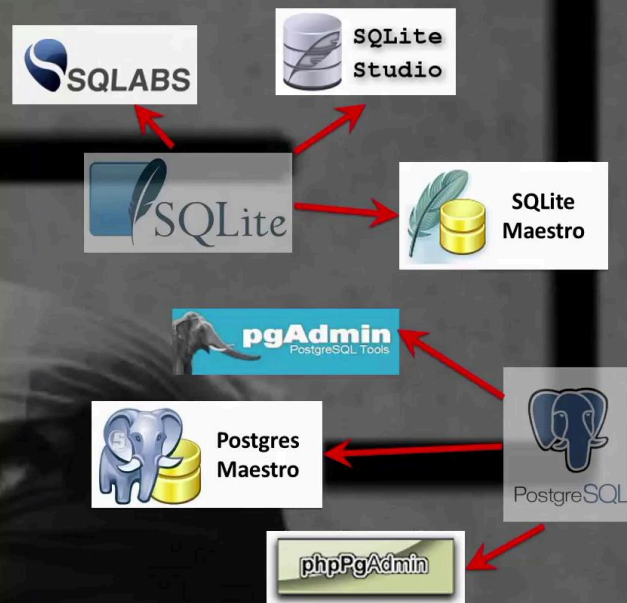
Notes

Summary



24m 50s

Logiciels SGBD – Bases de données relationnelles



Introduction aux systèmes d'information géographique

Tous ces clients offrent en fait un interface utilisateur et permettent d'entrer en contact avec ces bases de données et de les manipuler.

Notes

Summary



25m 58s

Logiciels SGBD – Bases de données relationnelles

Fonctionnalités essentielles

- Gestion de la structure des données
- Consultation et gestion des données
- Construction et exécution de requêtes

Introduction aux systèmes d'information géographique

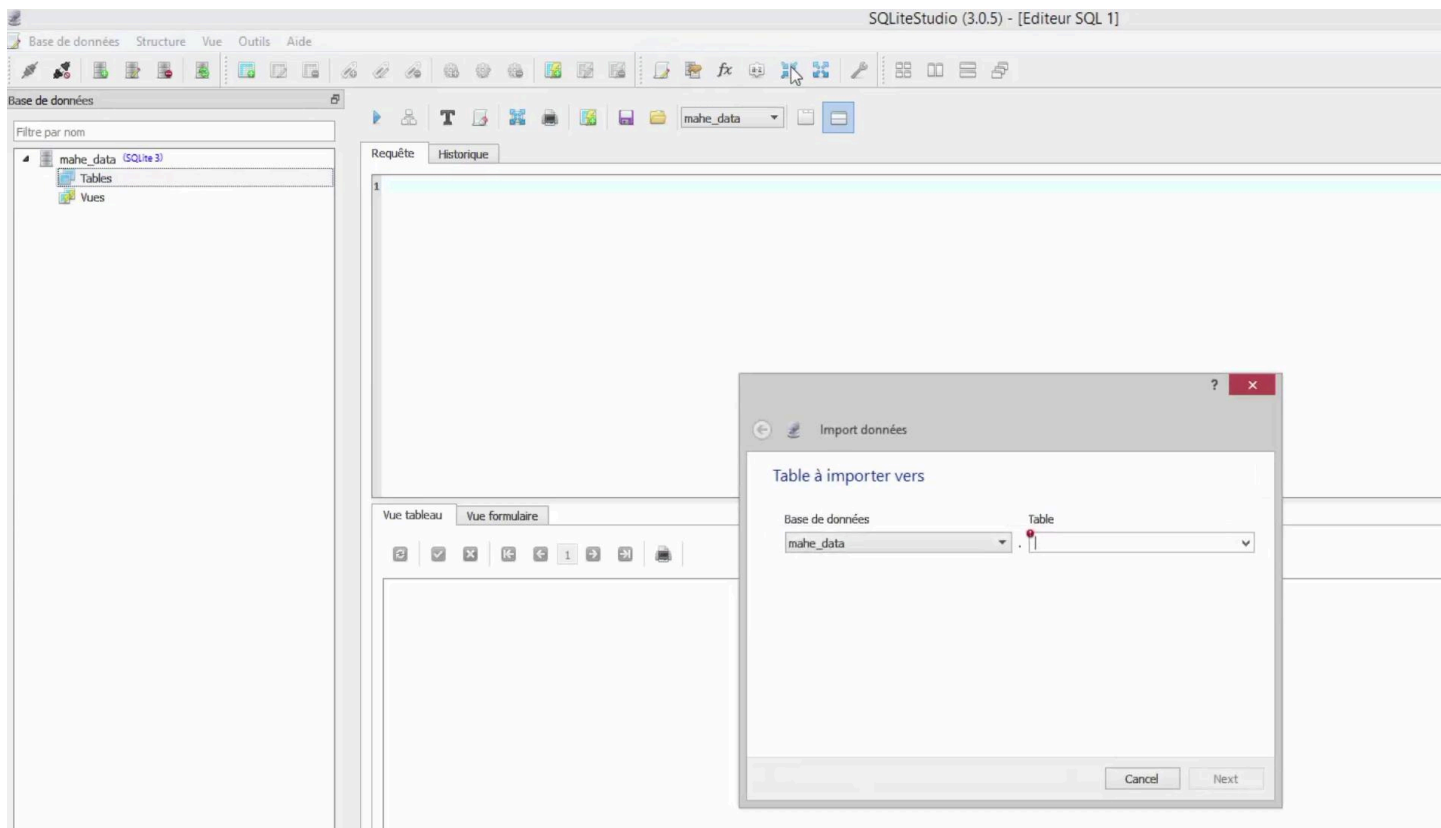
Parmi les fonctionnalités que l'on souhaite trouver dans les systèmes de gestion de bases de données, il en est en fait de trois types, tout d'abord celles qui permettent de gérer la structure des données, donc de créer de nouveaux champs, de nouveaux attributs avec leur domaine de valeur, de créer de nouveaux index, clés primaires, clés externes, etc., de consulter les données enregistrées dans la base de données, de consulter les tables de valeur éventuellement de pouvoir importer des données depuis l'extérieur pour peupler ces différentes tables et finalement toutes les fonctionnalités qui sont liées à la construction et à l'exécution de requêtes pour aller rechercher des objets dans une base de données sur la base d'un certain nombre de critères. C'est une chose que nous allons voir plus en détail en particulier dans les leçons de la deuxième semaine de ce module.

Notes

Summary



26m 08s



Dans l'immédiat, nous allons prendre un petit exemple qui est celui des districts des Seychelles que l'on peut exporter dans un format base de données, en l'occurrence une base de données SpatiaLite. Donc on va appeler ce fichier "mahe.sqlite" et l'enregistrer. On peut préciser quelques paramètres, la couche enregistrée va être ajoutée en fait... ah, il demande de préciser le système de projection utilisé, UTM sud 40. On peut masquer cette nouvelle couche. Et dans un deuxième temps on va enregistrer simplement les attributs de la couche, donc sans la géométrie, sous la forme d'un fichier ASCII, un fichier texte.csv, donc un fichier texte avec des valeurs séparées par des virgules. Pareillement, la table va être ajoutée dans la liste des objets disponibles. Si on passe maintenant à un logiciel de gestion de bases de données. Il s'agit de SQLite Studio. On peut importer. On va commencer par créer une base de données en fait dans laquelle on va importer le fichier texte. Donc on va appeler cette base de données "mahe_data". Cette base de données une fois créée on va commencer par se connecter à cette base de données et on voit qu'elle peut contenir des tables et des vues qui sont en fait des résultats de requêtes rendues un peu permanent.

Notes

Summary



AREA_KM2

COMMENT

Index

Déclencheurs

Vues

mahe (SQLite 3)

Tables (22)

SpatialIndex (virtual)

geometry_columns

geometry_columns_auth

geometry_columns_field_infos

geometry_columns_statistics

geometry_columns_time

idx_mahe_GEOMETRY (virtual)

idx_mahe_GEOMETRY_node

idx_mahe_GEOMETRY_parent

idx_mahe_GEOMETRY_rowid

mahe

Colonnes (7)

OGC_FID

GEOMETRY

id

name

perim_km

area_km2

comment

Index

Déclencheurs (5)

ggi_mahe_GEOMETRY

ggv_mahe_GEOMETRY

gid_mahe_GEOMETRY

gi_mahe_GEOMETRY

giu_mahe_GEOMETRY

spatial_ref_sys

spatialite_history

sql_statements_log

views_geometry_columns

4	4	46204.0	Takamaka	29.882	14.217	NULL
5	5	46205.0	St Louis	7.931	1.383	NULL
6	6	46206.0	Roche Caiman	14.599	1.159	NULL
7	7	46207.0	Port Glaud	45.31	25.249	NULL
8	8	46208.0	Pointe Larue	12.01	3.34	NULL
9	9	46209.0	Plaisance	10.454		
10	10	46210.0	Mont Fleuri	9.429		
11	11	46211.0	Mont Buxton	5.369		
12	12	46212.0	Les Mamelles	6.827		
13	13	46213.0	Grand Anse Mahe	23.802		
14	14	46214.0	Glacis	17.488		
15	15	46215.0	English River	7.612		
16	16	46216.0	Cascade	24.513		
17	17	46217.0	Bel Ombre	18.058		
18	18	46218.0	Bel Air	16.288		
19	19	46219.0	Beau Vallon	13.218		
20	20	46220.0	Baie Lazare	23.982		
21	21	46221.0	Au Cap	16.406		
22	22	46222.0	Anse Royale	15.695		
23	23	46223.0	Anse Etoile	14.138		
24	24	46224.0	Anse Boileau	18.985		
25	25	46225.0	Anse Aux Pins	7.946		

Déclencheur

DDL

Nom du déclencheur:

ggi_mahe_GEOMETRY

Quand:

BEFORE

Action:

INSERT

Sur table:

mahe

Portée:

FOR EACH ROW

Précondition:

1

Code:

```

1 SELECT RAISE(ROLLBACK, 'mahe.GEOMETRY violates Geometry
constraint [geom.type or SRID not allowed]')
2 WHERE (SELECT geometry.type FROM geometry.columns
3 WHERE Lower(f.table name) = Lower('mahe') AND Lower(f.geometry.column)
= Lower('GEOMETRY'))
4 AND GeometryConstraints(NEW."GEOMETRY".geometry.type, srid) = 1) IS
NULL;
```

OK

Cancel

Dans le tables, on va importer des données dans une nouvelle table, on va créer cette nouvelle table qu'on va appeler "district" et importer le fichier texte qui contient les attributs de ces districts qu'on a sauvegardé précédemment. On dit que la première ligne représente les en-têtes des colonnes et on voit que cette table apparaît maintenant dans l'arborescence, cette table district qui contient un certain nombre de colonnes qui sont les différents champs que nous avons dans la couche district de QGIS. On peut consulter également les données elles-mêmes et modifier la structure de données si nécessaire. Dans un deuxième temps, on va ajouter une autre base de données qui est en fait la base de données sauvegardée de QGIS avec les éléments de géométrie et on voit que si on se connecte à cette base de données on a un nombre d'objets beaucoup plus considérable puisqu'en fait, pour gérer la dimension géométrique la base de donnée SQLite génère tout un tas d'objets qui sont nécessaires et qui rendent la chose quand on l'aborde comme ça de loin un peu plus complexe. Mais on retrouve aussi la table de données des districts de Mahé avec les mêmes colonnes, les mêmes valeurs que l'on peut consulter sous forme de table.

Notes

Summary

28m 48s

SQLiteStudio (3.0.5) - [mahe (mahe)]

Base de données

mahe_data (SQLite3)

- Tables (1)
- districts
 - Colonnes (5)
 - ID
 - NAME
 - PERIM_KM
 - AREA_KM2
 - COMMENT
 - Index
 - Déclencheurs

Vues (SQLite3)

- mahe (mahe)
 - Tables (22)
 - SpatialIndex (virtuel)
 - geometry_columns
 - geometry_columns_auth
 - geometry_columns_field_info
 - geometry_columns_statistics
 - geometry_columns_time
 - idx_mahe_GEOMETRY (virtuel)
 - idx_mahe_GEOMETRY_node
 - idx_mahe_GEOMETRY_parent
 - idx_mahe_GEOMETRY_rowid
 - mahe
 - Colonnes (7)
 - OGC_FID
 - GEOMETRY
 - id
 - name
 - perim_km
 - area_km2
 - comment
 - Index
 - Déclencheurs (5)
 - ggp_mahe_GEOMETRY
 - ggp_mahe_GEOMETRY
 - gid_mahe_GEOMETRY
 - gi_mahe_GEOMETRY
 - gu_mahe_GEOMETRY
 - spatial_ref_sys
 - spatialite_history
 - sql_statements_log
 - views_geometry_columns

Vue tableau

OGC_FID GEOMETRY id name perim_km area_km2 comment

1	1	46226.0	Baie Ste Anne	43.686354	22.479533	NULL
2	2	46227.0	Grand Anse Praslin	28.876896	15.087715	NULL
3	3	46203.0	La Digue	19.873296	9.808824	NULL
4	4	46204.0	Takamaka	29.882	14.217	NULL
5	5	46205.0	St Louis	7.931	1.383	NULL
6	6	46206.0	Roche Caiman	14.599	1.159	NULL
7	7	46207.0	Port Glaupe	45.31	25.249	NULL
8	8	46208.0	Pointe Larue	12.01	3.34	NULL
9	9	46209.0	Plaisance	10.454	3.372	NULL
10	10	46210.0	Mont Fleuri	9.429	1.838	NULL
11	11	46211.0	Mont Bueton	5.369	1.165	NULL
12	12	46212.0	Les Mamelles	6.827	1.659	NULL
13	13	46213.0	Grand Anse Mahe	23.802	15.632	NULL
14	14	46214.0	Glacis	17.488	6.879	NULL
15	15	46215.0	English River	7.612	1.38	NULL
16	16	46216.0	Cascade	24.513	10.246	NULL
17	17	46217.0	Bel Ombre	18.058	9.409	NULL
18	18	46218.0	Bel Air	16.288	4.383	NULL
19	19	46219.0	Beau Vallon	13.218	4.451	NULL
20	20	46220.0	Baie Lazare	23.982	12.058	NULL
21	21	46221.0	Au Cap	16.406	8.27	NULL
22	22	46222.0	Anse Royale	15.895	7.164	NULL
23	23	46223.0	Anse Etoile	14.138	5.961	NULL
24	24	46224.0	Anse Boileau	18.985	12.021	NULL
25	25	46225.0	Anse Aux Pins	7.946	2.459	NULL

Status

[08:27:19] Importation des données de la table 'districts' réussie.

Editeur SQL 1 districts (mahe_data) mahe (mahe)

On voit qu'il n'y en plus pas encore d'index mais des déclencheurs, triggers en anglais, qui sont en fait les règles que le système de gestion de bases de données doit vérifier lors de chaque transaction. Donc on peut définir les règles de validation au cas par cas. Voilà pour cette introduction aux logiciels système de gestion de bases de données que nous reverrons plus en détails dans d'autres leçons du cours.

Notes

Summary

30m 15s

