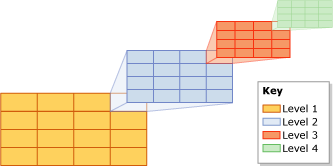
**Про просторових індексах**

**Декомпозиція індексованого простору в сітковий ієрархію**

У SQL Serverпространственние індекси будуються за допомогою збалансованих дерев, що означає, що індекси повинні представляти двовимірні просторові дані в лінійному порядку збалансованих дерев. Тому перед зчитуванням даних в просторовий індекс SQL Server проводить ієрархічну декомпозицію простору. У процесі створення індексу відбувається декомпозиція простору в чотирьохрівневу сітковий ієрархію. Ці рівні називають Рівень 1 (верхній), Рівень 2, Рівень 3 і Рівень 4.

Кожен наступний рівень містить подальшу декомпозицію рівня вище, так що кожна клітинка рівня вище містить повну сітку наступного рівня. На заданому рівні все сітки мають однакове число осередків на обох осях (наприклад, 4x4 або 8x8), і все осередки мають однаковий розмір.

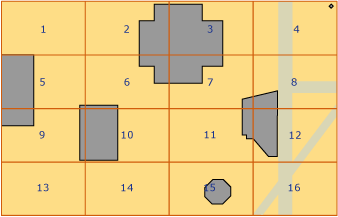
На наступному малюнку показана декомпозиція верхній правій осередки на кожному рівні гратчастої ієрархії в сітку 4x4. Фактично, таким чином декомпозиція виконується для всіх осередків. Наприклад, декомпозиція простору в чотири рівні сіток 4x4 фактично призводить до створення 65 536 осередків четвертого рівня.



 Примітка

Декомпозиція простору в просторовий індекс не залежить від одиниць виміру в даних програми.

Осередки в гратчастої ієрархії нумеруються в лінійному порядку з використанням варіанту заповнення простору кривої Гільберта. Однак на даному малюнку використовується проста построковая нумерація замість нумерації, яка фактично створюється кривої Гільберта. На наступному малюнку кілька багатокутників, що представляють будівлі, і ліній, що представляють вулиці, поміщені в сітку 4x4 рівня 1. Осередки першого рівня нумеруються від 1 до 16, починаючи з верхньої лівої.



**щільність сітки**

Число осередків по осях сітки визначає її щільність: чим більше число, тим щільніше сітка. Наприклад, сітка 8x8 (яка породжує 64 осередки) щільніше сітки 4x4 (яка породжує 16 осередків). Щільність сітки визначається за рівнями.

Інструкція CREATE SPATIAL INDEXTransact-SQL підтримує використання пропозиції GRIDS, яке дозволяє вказувати різні щільності сітки на різних рівнях. Щільність сітки для даного рівня задається за допомогою одного з наступних ключових слів.

|  |  |  |
| --- | --- | --- |
| Ключове слово | Конфігурація сітки | Ключове слово |
| LOW | 4X4 | 16 |
| MEDIUM | 8X8 | 64 |
| HIGH. | 16X16 | 256 |

У SQL Server, якщо рівень сумісності бази даних має значення 100 або нижче, за замовчуванням використовується значення MEDIUM на всіх рівнях. Якщо рівень сумісності бази даних має значення 110 або вище, то за замовчуванням використовується автоматична схема сітки. (Автоматична сітка вказує 8 рівень конфігурації HLLLLLLL.) Замість зміни щільності сітки індексу можна змінювати число осередків на об'єкт і число осередків вікна запиту на об'єкт за допомогою вказівки.

Управління процесом декомпозиції проводиться завданням щільності сітки, відмінною від значення за замовчуванням. Наприклад, може виявитися корисним використання різної щільності сіток на різних рівнях для точного налаштування індексу, виходячи з розміру індексованого простору і об'єктів в просторовому стовпці.

Примітка

Значення щільності сіток в просторовому індексі можна переглянути в стовпцях level\_1\_grid, level\_2\_grid, level\_3\_grid і level\_4\_grid подання каталогу sys.spatial\_index\_tessellations, якщо рівень сумісності бази даних має значення 100 або нижче. Параметри схеми тесселяции GEOMETRY\_AUTO\_GRID / GEOGRAPHY\_AUTO\_GRID не заповнюють ці стовпці. Представлення каталогу sys.spatial\_index\_tessellations містить значення NULL в цих стовпцях, якщо будуть використовуватися установки автосеткі.

**тесселяція**

Після декомпозиції індексованого простору в сітковий ієрархію просторовий індекс підрядник зчитує дані з просторового стовпчика. Після зчитування даних для просторового об'єкта (або примірника) просторовий індекс виконує процес тесселяції для цього об'єкта. Процес тесселяції поміщає об'єкт в cеточную ієрархію, встановлюючи зв'язок між об'єктом і набором сіткових осередків, з якими він стикається (контактні осередки). Починаючи з рівня 1 гратчастої ієрархії, процес тесселяції спочатку обробляє рівень в ширину. В принципі процес може продовжитися для всіх чотирьох рівнів, по одному рівню за раз.

Результатом процесу тесселяции є набір контактних осередків, які записуються в просторовий індекс об'єкта. Посилаючись на ці записані осередки, просторовий індекс може знайти об'єкт в просторі по його розташуванню щодо інших об'єктів в просторовому стовпці, які також зберігаються в індексі.

Правила тесселяции

Щоб обмежити число контактних осередків, що записуються для об'єкта, при проведенні тесселяции застосовується ряд правил тесселяции. Ці правила визначають глибину процесу тесселяции і контактні осередки, які записуються в індекс.

Нижче наведені ці правила.

* правило покриття

Якщо об'єкт повністю покриває осередок, кажуть, що цей осередок накрита об'єктом. Накрита осередок вважається і не піддається тесселяции. Це правило застосовується на всіх рівнях гратчастої ієрархії. Це правило спрощує процес тесселяції і скорочує обсяг даних, що записуються в просторовий індекс.

* Правило осередків на об'єкт

Це правило перевіряє обмеження осередків на об'єкт, що визначає максимальне число осередків, які можуть бути підраховані для кожного об'єкта, за винятком першого рівня. На більш низьких рівнях правило осередків на об'єкт визначає обсяг даних, які можуть бути записані про об'єкт.

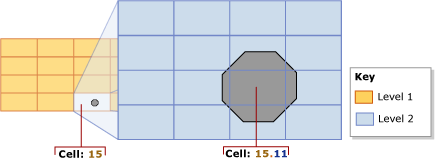
* Правило найглибшої осередки

Правило найглибшої осередку створює найкраще наближення для об'єкта, записуючи тільки самі нижні осередки, створені при тесселяции об'єкта. Батьківські осередки не включаються в лічильник осередків на об'єкт і не записуються в індекс.

Ці правила тесселяции застосовуються рекурсивно на кожному рівні сітки. У решти цього розділу правила тесселяции описуються більш детально.

**правило покриття**

Якщо об'єкт повністю покриває осередок, кажуть, що цей осередок накрита об'єктом. Наприклад, на наступному малюнку один з осередків другого рівня (15.11) повністю накрита середньою частиною восьмикутника.



Накрита осередок вважається і записується в індекс, і подальша тесселяция для осередку не проводиться.

**Правило осередків на об'єкт**

Екстент тесселяции для кожного об'єкта в основному залежить від обмеження осередків на об'єкт в просторовому індексі. Це обмеження визначає максимальне число осередків, які можуть вважатися в процесі тесселяции об'єкта. Однак майте на увазі, що правило осередків на об'єкт не діє для рівня 1, тому це обмеження може бути перевищено. Якщо рахунок на рівні 1 досягає або перевищує обмеження осередків на об'єкт, на нижніх рівнях подальша тесселяция не проводиться.

Поки рахунок менше обмеження осередків на об'єкт, процес тесселяції триває. Починаючи з контактної осередки з мінімальним номером (наприклад, осередок 15.6 на попередньому малюнку), процес перевіряє кожну клітинку і приймає рішення, вважати її або проводити тесселяцию. Якщо при тесселяции осередку буде перевищено обмеження осередків на об'єкт, то осередок буде порахована, але тесселяция для неї проведена не буде. В іншому випадку проводиться тесселяция осередки, а осередки нижнього рівня, що стикаються з об'єктом, вважаються. Процес тесселяції триває, таким чином, в ширину по всьому рівню. Цей процес повторюється рекурсивно для сіток нижнього рівня тесселірованних осередків, поки не буде досягнуто обмеження або все вічка не будуть підраховані.

Наприклад, розглянемо попередній малюнок, де восьмикутник повністю поміщається в комірку 15 сітки рівня 1. На цьому малюнку для осередку 15 була проведена тесселяция, яка розбила восьмикутник на дев'ять осередків рівня 2. На малюнку передбачається, що обмеження осередків на об'єкт дорівнює 9 або більше. Однак, якщо б обмеження осередків на об'єкт дорівнювало 8 або менше, то для осередку 15 тесселяция б не проводилася і для об'єкта вважалася б тільки осередок 15.

За замовчуванням обмеження осередків на об'єкт дорівнює 16, що забезпечує розумний компроміс між охопленням і точністю для більшості просторових індексів. Однак інструкція CREATE SPATIAL INDEXTransact-SQL підтримує пропозицію CELLS\_PER\_OBJECT = n, яке дозволяє обмежити кількість осередків на об'єкт (від 1 до 8192 включно).

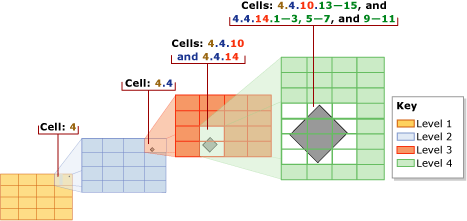
Примітка

Параметр cells\_per\_object просторового індексу можна знайти в поданні каталогу sys.spatial\_index\_tessellations.

**Правило найглибшої осередки**

Правило найглибшої осередку враховує той факт, що кожна клітинка нижнього рівня належить осередку над нею. Осередок рівня 4 належить осередку рівня 3, осередок рівня 3 належить осередку рівня 2, а осередок рівня 2 належить осередку рівня 1. Наприклад, об'єкт, який відноситься до осередку 1.1.1.1, також належить осередку 1.1.1, осередку 1.1 і осередку 1. відомості про таких ієрархічних зв'язках осередків вбудовані в обробник запитів. Тому в індекс необхідно записувати тільки осередки самого нижнього рівня, мінімізуючи обсяг даних, що зберігаються в індексі.

На наступному малюнку проводиться тесселяция відносно невеликого ромбовидного багатокутника. В індексі використовується обмеження осередків на об'єкт, за замовчуванням рівне 16, яке для цього невеликого об'єкта не досягається. Таким чином, тесселяция триває до рівня 4. Багатокутник розташовується в наступних комірках на рівнях з 1 по 3: 4, 4.4, 4.4.10 і 4.4.14. Однак, згідно з правилом найглибшої осередки, тесселяция враховує тільки дванадцять осередків на рівні 4: 4.4.10.13-15 і 4.4.14.1-3, 4.4.14.5-7 і 4.4.14.9-11.



**схеми тесселяции**

Поведінка просторового індексу частково залежить від використовуваної схеми тесселяции. Схема тесселяции залежить від типу даних. У SQL Serverпространственние індекси підтримують дві схеми тесселяции.

* Тесселяція геометричній сітки ця схема призначена для типу даних geometry).
* Тесселяція сітки географічних об'єктів, яка застосовується до стовпців типу даних geography.

Примітка

Параметр tessellation\_scheme просторового індексу можна знайти в поданні каталогу sys.spatial\_index\_tessellations.

**Схема тесселяции сітки геометричних об'єктів**

Тесселяція GEOMETRY\_AUTO\_GRID використовується за умовчанням для типу даних geometry в SQL Server 2012 (11.x) і більш пізніх версіях. Схема тесселяции GEOMETRY\_GRID є єдиною доступною для геометричних типів даних в SQL Server 2008. У цьому розділі розглядаються аспекти тесселяции сітки геометричних об'єктів, релевантні для просторових індексів: підтримувані методи і обмежують прямокутники.

Примітка

Цю схему тесселяции можна задати явно за допомогою пропозиції USING (GEOMETRY\_AUTO\_GRID / GEOMETRY\_GRID) інструкції CREATE SPATIAL INDEX Transact-SQL.

**обмежує прямокутник**

Геометричні дані займають площину, яка може бути нескінченною. Однак в SQL Serverдля просторового індексу потрібно кінцеве простір. Для визначення кінцевого простору для декомпозиції схемою тесселяции сітки геометричних об'єктів потрібно обмежує прямокутник. Обмежує прямокутник визначається чотирма координатами (x-min, y-min) і (x-max, y-max), які зберігаються у вигляді властивостей просторового індексу. Ці координати представляють наступне.

* x-min - це координата лівого нижнього кута прямокутника по осі X.
* y-min - це координата лівого нижнього кута по осі Y.
* x-max - це координата верхнього правого кута по осі X.
* y-max - це координата верхнього правого кута по осі Y.

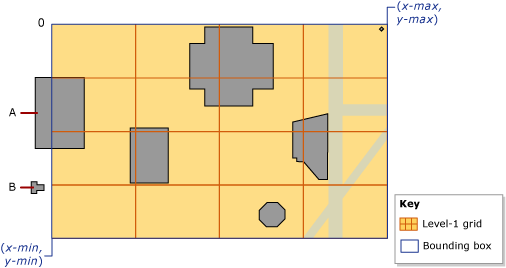
Примітка

Ці координати задаються пропозицією BOUNDING\_BOX в інструкції CREATE SPATIAL INDEXTransact-SQL.

Координати (x-min, y-min) і (x-max, y-max) визначають розташування і розміри прямокутника. Простір за межами прямокутника вважається одним осередком з номером 0.

У просторовому індексі проводиться декомпозиція простору всередині прямокутника. Сітка рівня 1 гратчастої ієрархії заповнює обмежує прямокутник. Щоб помістити геометричний об'єкт в сітковий ієрархію, просторовий індекс порівнює координати об'єкта з координатами прямокутника.

На наведеному нижче малюнку показані точки, певні координатами (x-min, y-min) і (x-max, y-max) прямокутника. Верхній рівень гратчастої ієрархії показаний як решітка 4x4. На даному малюнку нижні рівні опущені. Простір за межами прямокутника позначається нулем (0). Зверніть увагу, що об'єкт A частково виходить за межі прямокутника, а об'єкт B знаходиться повністю поза прямокутника в осередку 0.



Обмежує прямокутник відноситься до деякої частини просторових даних програми. Додаток саме визначає, чи буде обмежує прямокутник індексу повністю включати дані з просторового шпальти чи тільки їх частина. Просторовий індекс має сенс використовувати тільки для операцій з об'єктами, повністю знаходяться всередині прямокутника. Тому для максимально ефективного використання просторового індексу по стовпцю geometry необхідно вказати обмежує прямокутник, що містить всі об'єкти або більшість з них.

Примітка

Щільності сіток в просторовому індексі можна переглянути в стовпцях bounding\_box\_xmin, bounding\_box\_ymin, bounding\_box\_xmax і bounding\_box\_ymax подання каталогу sys.spatial\_index\_tessellations.

**Схема тесселяции сітки географічних об'єктів**

Ця схема тесселяции застосовується тільки до колонку geography. У цьому розділі розглядаються методи, підтримувані тесселяції географічних об'єктів, і описується, як геодезичне простір проектується на площину, яка потім піддається декомпозиції в сітковий ієрархію.

Примітка

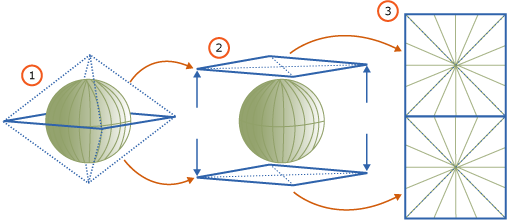
Цю схему тесселяции можна задати явно за допомогою пропозиції USING (GEOGRAPHY\_AUTO\_GRID / GEOGRAPHY\_GRID) в інструкції CREATE SPATIAL INDEXTransact-SQL.

**Проекція геодезичного простору на площину**

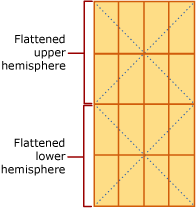
При обчисленнях з екземплярами geography (об'єктами) простір, що містить об'єкти, вважається геодезичним еліпсоїдом. Для декомпозиції цього простору схема тесселяции сітки географічних об'єктів розділяє еліпсоїд на верхню і нижню півкулі, а потім виконує наступні кроки.

1. Проектує кожну півкулю на межі чотиристоронньої піраміди.
2. Робить обидві піраміди плоскими.
3. Поєднує плоскі піраміди для створення неевклідової площині.

На наступному малюнку показано схематичне уявлення трьохетапного процесу декомпозиції. У пірамідах пунктирні лінії представляють кордону чотирьох граней кожної піраміди. На кроках 1 і 2 показаний геодезичний еліпсоїд: зелена горизонтальна лінія являє екваторіальну широту, а ряд зелених вертикальних ліній представляють кілька довгот. На кроці 1 показано проектування двох півкуль на піраміди. На кроці 2 показано уплощение пірамід. На кроці 3 показані плоскі піраміди після їх об'єднання в площину і число спроектованих ліній довготи. Зверніть увагу, що ці спроектовані лінії випрямлені і розрізняються по довжині в залежності від місця проектування на піраміду.



Після проектування простору на площину проводиться її декомпозиція в чотирьохрівневу сітковий ієрархію. На різних рівнях можуть використовуватися різні щільності сітки. На наступному малюнку показана площина після її декомпозиції в сітку 4x4 рівня 1. На даному малюнку нижні рівні гратчастої ієрархії опущені. Насправді площину піддається повній декомпозиції в чотирьохрівневу сітковий ієрархію. Після закінчення декомпозиції географічні дані з шпальти geography зчитуються по рядках, і для кожного об'єкта виконується процедура тесселяции.



**Підтримувані методи для просторових індексів**

**Геометричні методи, підтримувані просторовими індексами**

При певних умовах просторові індекси підтримують наступні геометричні методи на основі наборів: STContains (), STDistance (), STEquals (), STIntersects (), STOverlaps (), STTouches () і STWithin (). Щоб просторовий індекс підтримував ці методи, в запиті їх необхідно використовувати в межах пропозиції WHERE або JOIN ON, включивши до складу предиката наступного загального вигляду:

geometry1.імя\_метода (geometry2) оператор\_сравненія \*\* допустімое\_чісло

Щоб отримати ненульовий результат, аргументи geometry1 і geometry2 повинні мати однаковий ідентифікатор просторової посилання (SRID). В іншому випадку метод повертає значення NULL.

Просторові індекси підтримують предикати наступних форм:

* geometry1.STContains (geometry2) = 1
* geometry1.STDistance (geometry2) <номер
* geometry1.STDistance (geometry2) <= номер
* geometry1.STEquals (geometry2) = 1
* geometry1.STIntersects (geometry2) = 1
* geometry1. STOverlaps (geometry2) = 1
* geometry1.STTouches (geometry2) = 1
* geometry1.STWithin (geometry2) = 1

Географічні методи, підтримувані просторовими індексами

При певних умовах просторові індекси підтримують такі географічні методи для роботи з наборами: STIntersects (), STEquals (), and STDistance (). Щоб просторовий індекс підтримував ці методи, їх необхідно використовувати в реченні WHERE запиту, включивши до складу предиката наступного загального вигляду:

geography1.імя\_метода (geography2) оператор\_сравненія \*\* допустімое\_чісло

Щоб отримати ненульовий результат, аргументи geography1 і geography2 повинні мати однаковий ідентифікатор просторової посилання (SRID). В іншому випадку метод повертає значення NULL.

Просторові індекси підтримують предикати наступних форм:

* geography1.STIntersects (geography2) = 1
* geography1.STEquals (geography2) = 1
* geography1.STDistance (geography2) <номер
* geography1.STDistance (geography2) <= номер

Запити, що використовують просторові індекси

Просторові індекси підтримуються тільки в запитах, в реченні WHERE яких є індексований просторовий оператор. Приклад синтаксису.

**[spatial object].SpatialMethod([reference spatial object]) [ = | < ] [const literal or variable]**

Оптимізатор запитів враховує коммутативность просторових операцій (що @ a.STIntersects (@b) = @ b.STInterestcs (@a)). Однак просторовий індекс не використовуватиметься, якщо на початку порівняння немає просторового оператора (наприклад, WHERE 1 = spatial op не використовуватиме просторовий індекс). Для використання просторового індексу перепишіть порівняння (наприклад, WHERE spatial op = 1).

Як і при використанні будь-якого іншого індексу, якщо просторовий індекс підтримується, то рішення його використовувати приймається, виходячи з витрат, тому оптимізатор запитів може не прийняти рішення про використання, навіть якщо для цього дотримані всі вимоги. Використовуйте інструкції showplan, щоб визначити, чи використовувався просторовий індекс і чи потрібно надавати вказівки для примусового використання плану запиту.

Найближчий сусідній тип запиту також підтримує просторові індекси, але тільки якщо використовується певний синтаксис запиту. Правильний синтаксис:

**SELECT TOP(K) [WITH TIES] \***

**FROM <Table> AS T [WITH(INDEX(<SpatialIndex>))]**

**WHERE <SpatialColumn>.STDistance(@reference\_object) IS NOT NULL**

**ORDER BY <SpatialColumn>.STDistance(@reference\_object) [;]**