

МОДЕЛИРАНЕ И АНАЛИЗ НА СОФТУЕР

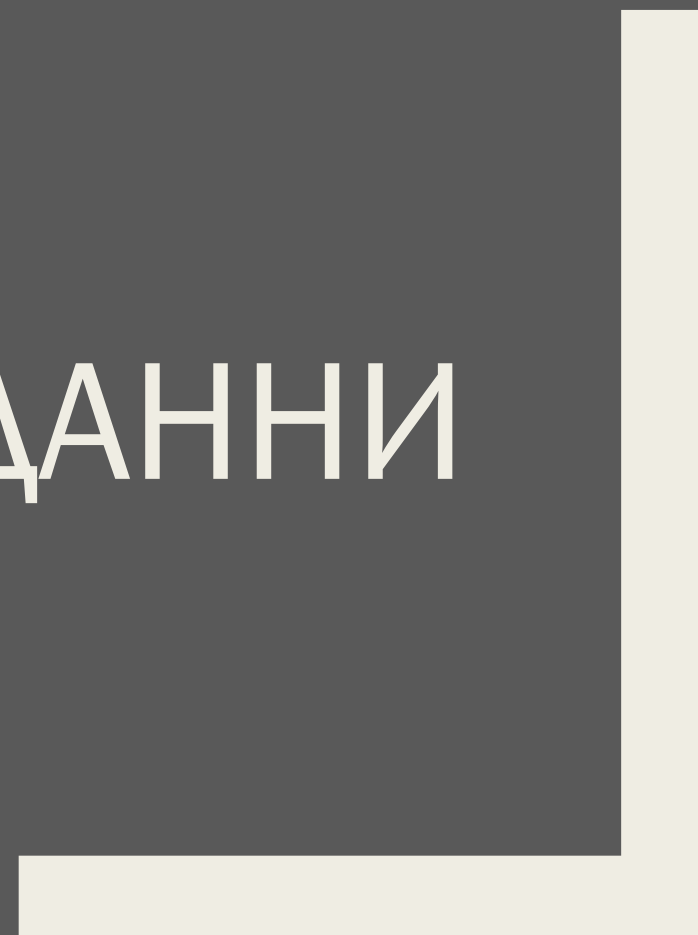
Павел Кюркчиев

Ас. към ПУ „Паисий Хилендарски“

<https://github.com/pkyurkchiev>

@pkyurkchiev

ТАБЛИЧНА БАЗА ДАННИ



Таблична база данни (Tabular database)

- Табличната база данни, както подсказва името, е база данни, която е структурирана в таблична форма. Той организира елементи от данни във вертикални колони и хоризонтални редове. Всяка клетка се формира от пресечната точка на колона и ред. Всеки ред и колона е уникално номериран, за да бъде правилен и ефективен. Този тип база данни има почти безкраен обхват за масово съхраняване на данни.

Режими на изпълнение

- In-Memory
- DirectQuery
- In-Memory with DirectQuery
- DirectQuery with In-Memory

In-Memory

- In-Memory режима на работа е по подразбиране. Данните в табличния модел се обработват и компресират, като се използва метода xVelocity in-memory analytics (също наричан VertiPaq). Този in-memory режим е оптимизиран за работа с висококачествен анализ и изследване на данните. Той осигурява бързо време за извличане на заявки изградени от агрегиращи операции.

Недостатъци на In-Memory

- Данните не се актуализират, когато източника данни бъде актуализиран, така че моделът трябва да бъде обработен отново.
- Когато изключите компютъра, на който се намира модела, кеша се запазва на диска. При зареждане на компютъра модела също трябва да бъде отворен и зареден. Запазването и зараждането могат да отнемат доста време.
- Сървъра се нуждае от много памет, ако искаме да пазим много фактически данни.

DirectQuery

- Този режим използва релационни данни, които се съхраняват в SQL Server база данни (подобно на режима ROLAP в многоизмерния модел). Тя позволява на потребителите да извличат данни директно от източник на данни на SQL Server в реално време. Всички DAX заявки за данните се превеждат от Analysis Services в еквивалентни SQL изрази срещу определения източник на релационни данни.

Ползи от DirectQuery

- Възможно е да има модел, който е твърде голям, за да се побере в паметта на сървъра на аналитичните услуги. Ако данните от източника на данни не могат да бъдат компресирани в 1 терабайт или по-малка база данни за аналитични услуги, трябва да се помисли за DirectQuery. Ако база данни за анализ на услуги не се побере в половината от паметта на машината, която е домакин на аналитичните услуги, трябва да се помисли за DirectQuery.

- Гарантира се актуалността на данните без нужда от допълнителни заявки.
- Има предимството да използва ускорение на заявките от страна на доставчика, като индекси на колони за SQL Server 2012 и нагоре.
- Анализните услуги могат да извършват оптимизация, за да гарантират, че планът на заявките към базата данни ще бъде колкото е възможно по-ефективен.
- Намалено използване на памет и процесорно време.

In-Memory with DirectQuery

- Това е хибриден режим. По подразбиране заявките трябва да се отговарят с In-memory режим, но режима може да бъде сменен от клиентите чрез промяна в connectionString в режим DirectQuery.

DirectQuery with In-Memory

- Това е хибриден режим. По подразбиране заявките трябва да се отговарят с помощта на режима DirectQuery, но режима може да бъде сменен от клиентите чрез промяна в connectionString в режим In-Memory.

Data Analysis Expressions (DAX)

- DAX е език на формулата, използван за създаване на персонализирани изчисления в услугите за анализ като Power BI Desktop и Power Pivot в Excel и други. DAX формулите включват функции, оператори и стойности за извършване на усъвършенствани изчисления на данни в таблици и колони. DAX формулите се използват в изчислените колони, мерки и филтри на редове.

DAX формули (DAX formulas)

- Изчислени клони (Calculated columns) – представляват изчислими колони.

Пример: `=[SalesAmount]-[TotalCost]-[ReturnAmount]`

- Мерки (Measures) – представляват изчислими полета.

Пример: `=SUM('FactResellerSales'[SalesAmount])*1.06`

- Филтри по редове (Row filters)

DAX функции (DAX functions)

- Функциите се използват за по – лесно калкулиране на изчислими колони и мерки.
- Основните типове:
 - *Date and time functions*
DAY(<date>)
 - *Filter functions*
FILTER(<table>,<filter>)
 - *Logical functions*
AND(<logical1>,<logical2>)

- *Mathematical and trigonometric functions*

ACOS(number)

- *Statistical functions*

AVERAGE(<column>)

- *Text functions*

FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])

ВЪПРОСИ ?

