Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут”

Кафедра АСОІУ

**ЗВІТ**

про виконання комп’ютерного практикуму № 4

з дисципліни

“ Програмні додатки з використанням баз даних”

по темі

«Запити з використанням PIVOT та UNPIVOT»

Варіант 1

|  |  |  |
| --- | --- | --- |
| Прийняв: |  | Виконав: |
| Cт.вик. Клименко О. М. |  | студент 3-го курсу  гр. ІП-51 ФІОТ  Булатов Дмитро Єгорович |

Київ – 2017

**ЗМІСТ:**

[1 Завдання 3](#_Toc495230724)

[2 Виконання Завдання 4](#_Toc495230725)

# Завдання

Завдання 1:

Згідно предметної області визначенної варіантом придумати та записати наступні запити:

1) Запит з PIVOT з використанням однієї агрегатної функції та 3-5

стовпчиками.

2) Запит з PIVOT з використанням двох агрегатних функцій та 3-5

стовпчиками.

3) Запит з UNPIVOT.

Завдання 2:

Вибрати та описати будь-яку OLAP, DWH (крім Teradata та Hadoop), ETL, бажано, не більше ніж 2 сторінки.

**Преметна область**: Відеопрокат

# Виконання Завдання

1. Завдання по написанню запитів
   1. Запит з PIVOT з використанням однієї агрегатної функції та 3-5 стовпчиками.  
      *Призначення*: Виводить кількість касет кожного режисера.

*Текст запиту*:

CREATE TABLE VHS(

ID int NOT NULL PRIMARY KEY IDENTITY(1,1),

Director varchar(40),

Name varchar(40)

)

INSERT INTO VHS VALUES

('Martin Scorsese','The Wolf of Wall Street'),

('Martin Scorsese','Goodfellas'),

('Martin Scorsese','Shutter Island'),

('Martin Scorsese','The Departed'),

('Alfred Hitchcock','Psycho'),

('Alfred Hitchcock','Vertigo'),

('Alfred Hitchcock','Rear Window'),

('Steven Spielberg','Saving Private Ryan'),

('Steven Spielberg','Schindlers List'),

('Roman Polanski','The Pianist'),

('Roman Polanski','Rosemarys Baby'),

('Roman Polanski','Chinatown')

SELECT \* FROM

(

SELECT Director, ID

FROM VHS

) AS Original

PIVOT

(

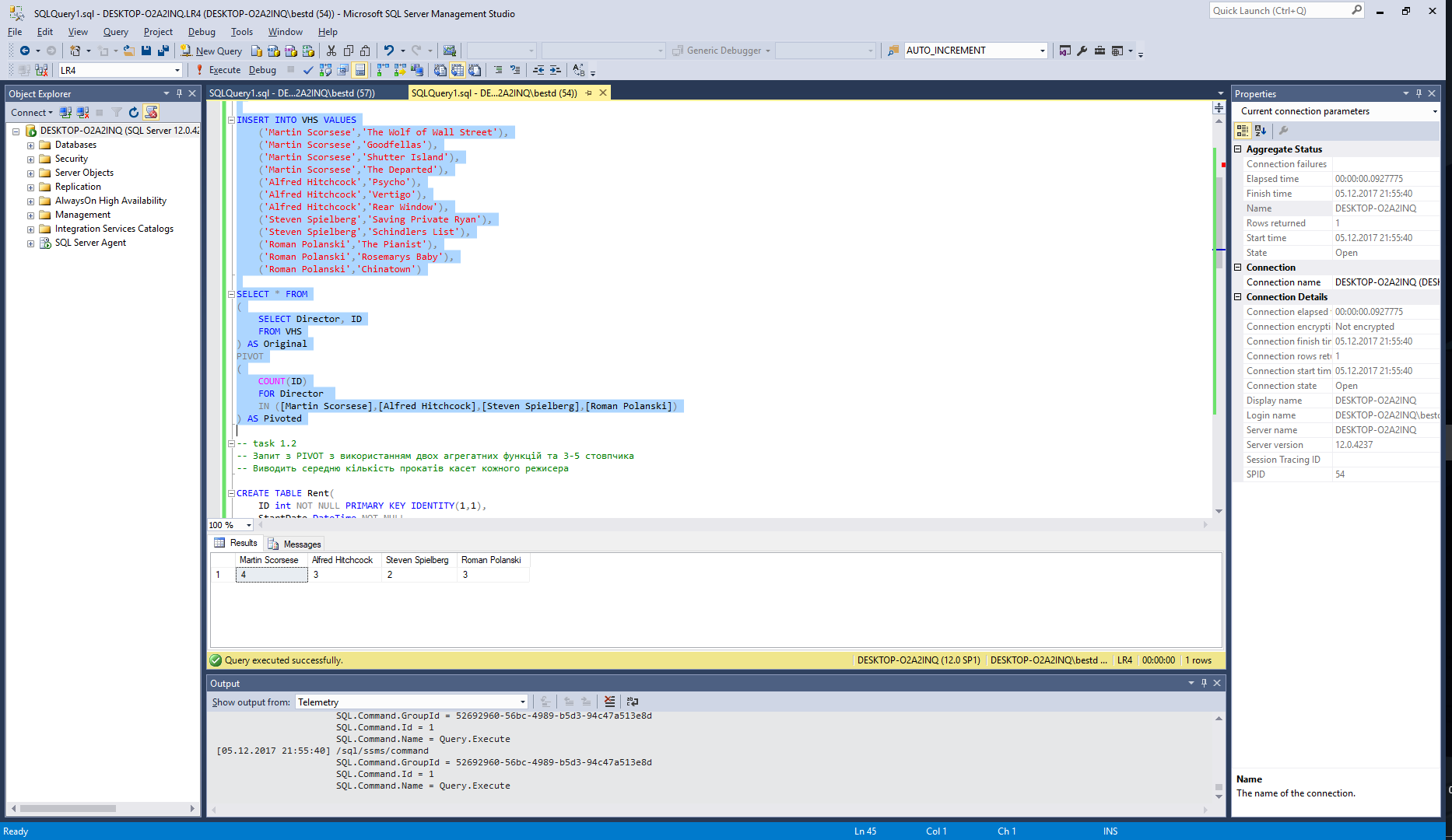
COUNT(ID)

FOR Director

IN ([Martin Scorsese],[Alfred Hitchcock],[Steven Spielberg],[Roman Polanski])

) AS Pivoted

*Результат виконання*:



* 1. Запит з PIVOT з використанням двох агрегатних функцій та 3-5 стовпчиками:

*Призначення*: Виводить середню кількість прокатів касет кожного режисера.

*Текст запиту*:

CREATE TABLE Rent(

ID int NOT NULL PRIMARY KEY IDENTITY(1,1),

StartDate DateTime NOT NULL,

ReturnDate DateTime DEFAULT(NULL),

VHS\_ID int NOT NULL FOREIGN KEY REFERENCES VHS(ID)

)

INSERT INTO Rent VALUES

('2017-09-15 00:00:00.000', '2017-09-20 00:00:00.000', 8 ),

('2016-10-17 00:00:00.000', '2016-10-27 00:00:00.000', 1 ),

('2017-10-12 00:00:00.000', '2017-10-21 00:00:00.000', 4 ),

('2017-10-22 00:00:00.000', NULL, 6 ),

('2017-10-20 00:00:00.000', NULL, 4 ),

('2016-10-11 00:00:00.000', '2017-10-11 00:00:00.000', 3 ),

('2017-07-12 00:00:00.000', '2017-08-02 00:00:00.000', 12),

('2017-10-13 00:00:00.000', NULL, 1 ),

('2016-10-14 00:00:00.000', NULL, 6 ),

('2017-10-15 00:00:00.000', '2017-10-22 00:00:00.000', 3 ),

('2016-10-16 00:00:00.000', '2016-10-17 00:00:00.000', 10)

SELECT \* FROM

(

SELECT Director, 1.0\*COUNT(Rent.ID) AS [VHS Rents]

FROM VHS

LEFT JOIN Rent ON VHS.ID = Rent.VHS\_ID

GROUP BY VHS.ID, Director

) AS Original

PIVOT

(

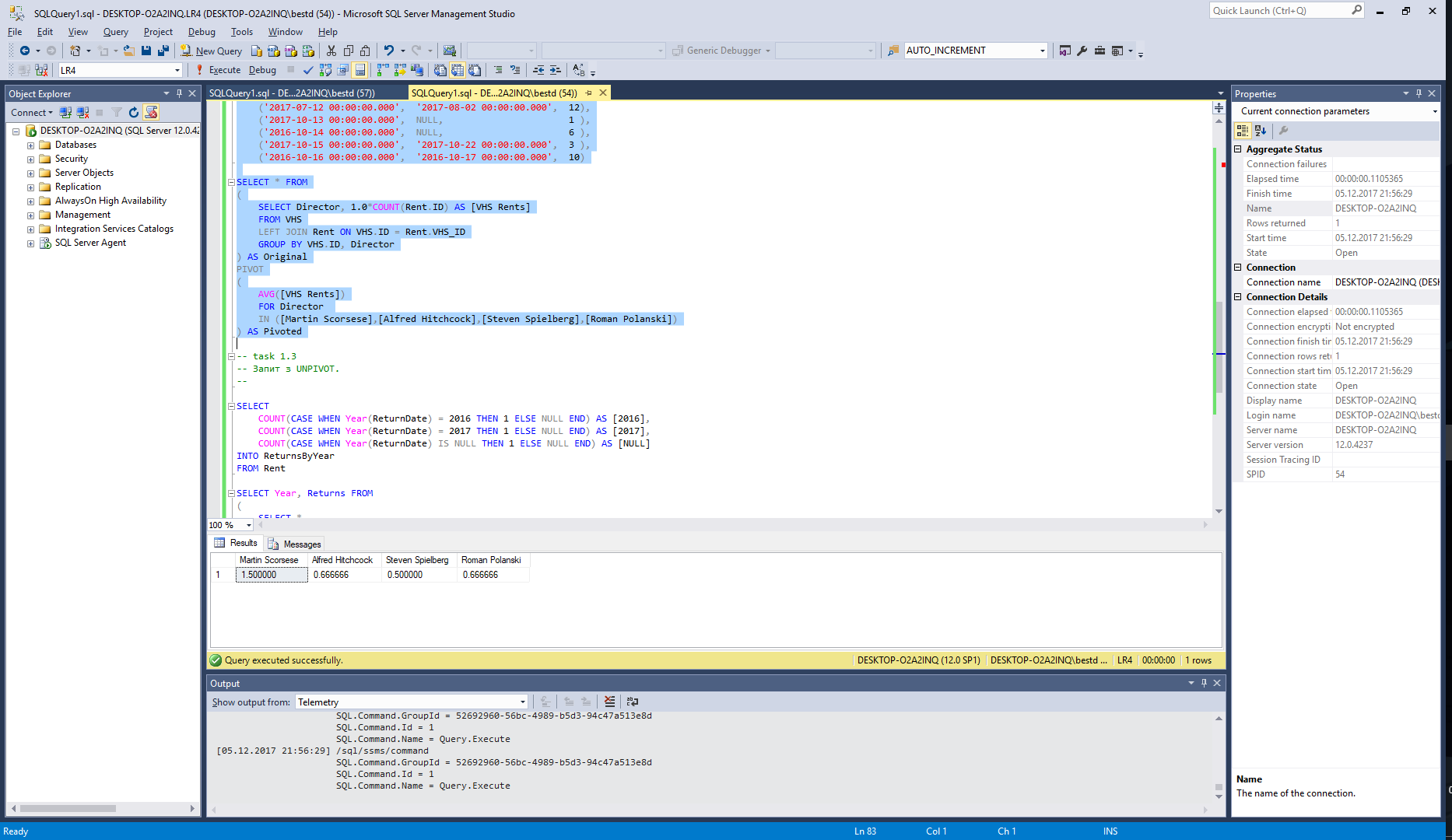
AVG([VHS Rents])

FOR Director

IN ([Martin Scorsese],[Alfred Hitchcock],[Steven Spielberg],[Roman Polanski])

) AS Pivoted

*Результат виконання*:



* 1. Запит з UNPIVOT.

*Призначення*: Розбиття рядку з данними на декілька.

*Текст запиту*:

SELECT

COUNT(CASE WHEN Year(ReturnDate) = 2016 THEN 1 ELSE NULL END) AS [2016],

COUNT(CASE WHEN Year(ReturnDate) = 2017 THEN 1 ELSE NULL END) AS [2017],

COUNT(CASE WHEN Year(ReturnDate) IS NULL THEN 1 ELSE NULL END) AS [NULL]

INTO ReturnsByYear

FROM Rent

SELECT Year, Returns FROM

(

SELECT \*

FROM ReturnsByYear

) AS Original

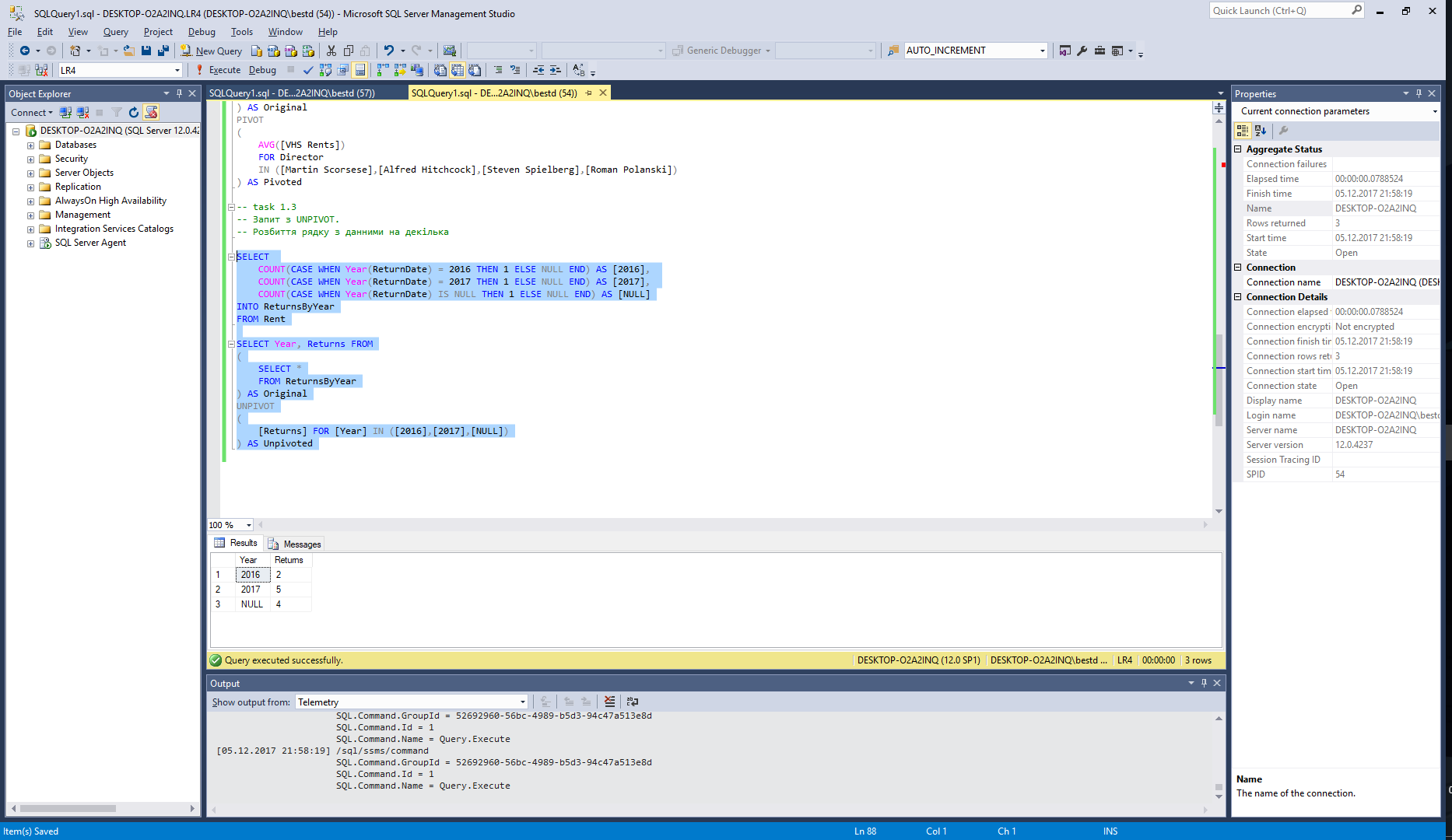
UNPIVOT

(

[Returns] FOR [Year] IN ([2016],[2017],[NULL])

) AS Unpivoted

*Результат виконання*:



1. Завдання по опису OLAP/ DWH/ ETL:

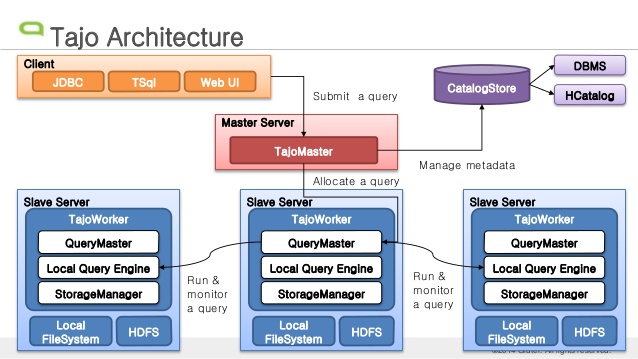
**Apache Tajo**

Apache Tajo – це реляційна розподілена DWH основана на Apache Hadoop. Вона створена для швидких запитів, агрегування та ETL на великих наборах даних що зберігаються в HDFS (Hadoop Distributed File System) або інших джерелах данних. На відміну від Hadoop, Tajo повністю підтримує SQL і підтримує прямий контроль розподілених запитів на базах даних що дозволяє оптимізувати запити.

Особливості Apache Tajo:

* Повністю розподілений процес обробки запитів SQL
* Розширена оптимізація запитів
* Інтерактивний аналіз на основі розумних даних
* Допустимість помилок та динамічне планування для довгих запитів
* Відповідність стандарту ANSI / ISO SQL
* Підтримка Hive MetaStore
* Підтримка драйвера JDBC
* Підтримуються різні формати файлів, такі як CSV, JSON, RCFile, SequenceFile, ORC та Parquet
* Визначені користувачем функції
* Інтерактивна оболонка
* Зручна утиліта резервного копіювання / відновлення
* Асинхронний / синхронний Java API

Схема архітектури Apache Tajo:



Наступні діаграми порівнюють швидкість виконання запитів на Tajo та інших інструментах що дозволяють використовувати SQL на Apache Hadoop (Данні для тестування мали розмір 1.7 TB а результуючі данні були менші за 8GB)