Ad-hoc Analytics

A SaaS-in-a-Box Tutorial

# Purpose of this tutorial

This tutorial uses Elastic Query in conjunction with an analytics database to run ad-hoc analytics queries that are distributed across all the tenants in the Wingtip catalog. These queries can extract insights buried in the day-to-day operational data of the WTP app. In this tutorial, you'll create an ad-hoc analytics database, with external tables mirroring tables in the tenant database schema, and which use the catalog to define the list of tenants to be queried. You’ll then run several queries across the tenants to uncover hidden insights.

# Ad-hoc Analytics pattern

One of the great opportunities with SaaS applications is to leverage the vast amount of tenant data you have stored centrally in the cloud, to gain insights into the operation and usage of your applications, your tenants and their users, and their preferences and behavior. This can guide feature development, usability improvements and other investments in your apps and services.

Accessing this data when it’s in a single multi-tenant database is easy, but not so easy when distributed at scale across potentially thousands of databases. One approach is to use Elastic Query, which enables ad-hoc query across a distributed set of databases with common schema. Elastic Query uses a single ‘head’ database in which external tables are defined that mirror tables in the distributed (tenant) databases. Queries submitted to this head database are compiled to produce a distributed query plan, with portions of the query pushed down to the tenant databases as needed. Elastic Query uses the shard map in the catalog database to provide the location of the tenant databases. Setup and query is straightforward using normal T-SQL, and supports ad hoc query from tools like Power BI and Excel.



# Setup

Download and extract **WTPLearningModules.zip** to a convenient folder.

Deploy the **WTP Application.** Ensurethe catalog is initialized using the Demo Assistant app. See the Introduction to the WTP SaaS Application tutorial for deployment instructions.

**SSMS** can be used to explore database schema and execute SQL queries directly.

**PowerShell ISE** is recommended to execute scripts and follow their execution in debug mode.

**PowerShell Tips**

* Open and configure demo- scripts in the PowerShell ISE.
* Use F5 to run the script (using F8 is not advised as the $PSScriptRoot variable is not evaluated when running snippets of a script).
* Use F9 to set a breakpoint to let you trace the script in debug mode to see how it works
* Use F10 to step through the script, F11 to step into a function, and Shift-F11 to step out.

## Getting Started

1. Update the user configuration file used by all tutorial scripts. Update again if you redeploy the app.
   1. Open ...\Learning Modules\UserConfig.psm1 in **PowerShell ISE**
   2. Modify **$userConfig.ResourceGroupName** to the resource group used for the deployed app.
   3. Modify **$userConfig.Name** to the User name used for the deployed app.
2. Open the following scripts in **PowerShell ISE**
   1. ...\Learning Modules\Operational Analytics\Adhoc Analytics\Demo-AdhocAnalytics.ps1
   2. …\Learning Modules\Operational Analytics\Adhoc Analytics\Deploy-AdhocAnalyticsDB.ps1
3. Open the following file in **SSMS**
   1. ...\Learning Modules\Operational Analytics\Adhoc Analytics\Demo-AdhocAnalyticsQueries.sql

# Walkthrough

## Exercise 1: Deploy the database that will be used for ad-hoc analytics queries

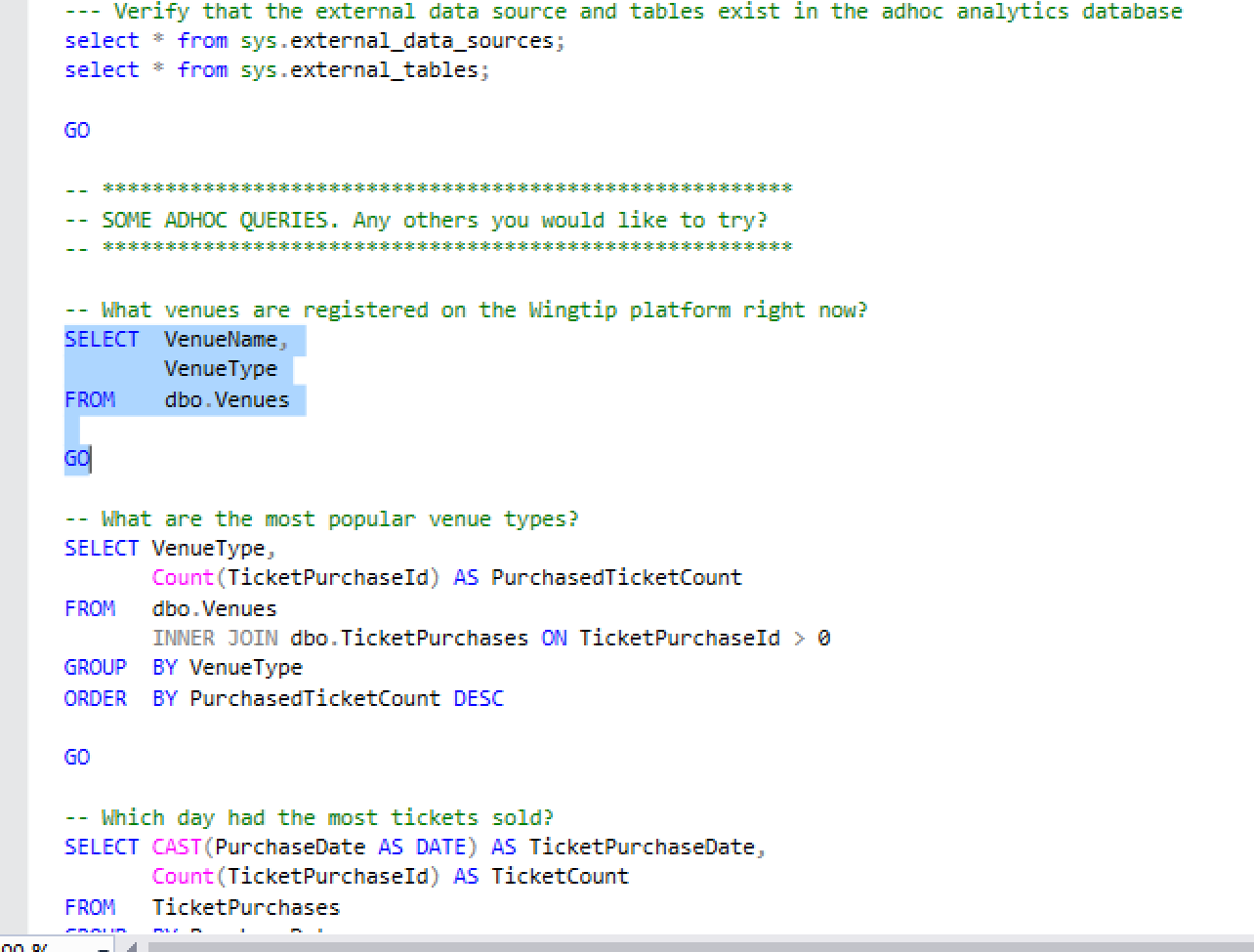
This exercise deploys the Ad-hoc Analytics database that contains the schema used for issuing ad-hoc queries that span all the tenant databases.

1. Complete the steps in the ‘Getting Started’ section first.
2. Navigate to …\Deploy-AdhocAnalyticsDB.ps1 in PowerShell ISE
3. Scroll down and review the SQL script that gets submitted to define the ad hoc analytics database. Note:
   1. The external data source which refers to the shard map in the catalog. This ensures the queries are distributed to all registered tenants.
   2. The set of external tables that are included: Events, Tickets, TicketPurchases, Venues. These are a subset of the tables in the tenant database schema.
   3. The definition of the local reference data table, VenueTypes, with its initial data. Having this locally avoids having to go to the remote table to fetch this data, which is the same in every tenant database.
4. Navigateto …\Demo-AdhocAnalytics.ps1 in PowerShell ISE
5. Run the script using F5.
   1. This script creates a new SQL database that will be used for ad-hoc analytics and adds the WTP catalog, TicketPurchases, Tickets, and Venues tenant tables as external tables that can be queried for data. If the database already exists this step is skipped.

You have now created the ad-hoc analytics database and initialized it with tables and it can be used to submit queries.

## Exercise 2: Run Ad-hoc analytics queries

This exercise will run ad-hoc analytics queries to uncover tenant insights from the WTP application.

1. Complete the steps in the ‘Getting Started’ section and Exercise 1 before this exercise.
2. Open Demo-AdhocAnalyticsQueries.sql in SSMS
3. Selectively execute the included demo queries
   * Select the query you want to execute and use F5 to execute (see a sample selection in the screenshot below).  
     

Congratulations, you have now created an ad-hoc analytics database and run some sample distributed queries across using Elastic Query across all the tenant databases in the WTP app.

## Next Steps

Also try the Tenant Analytics tutorial which explores another way to run analytics over your tenants by extracting tenant data into a separate database and querying it there.

Learn more about Elastic Query:

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-query-overview>