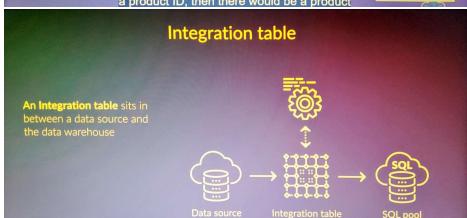
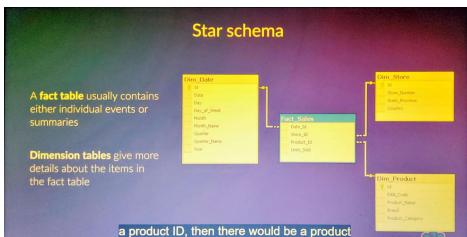
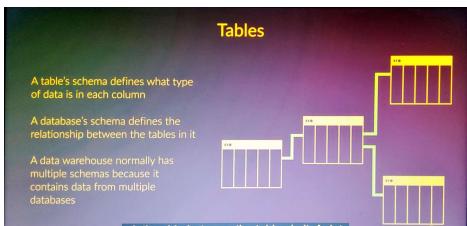
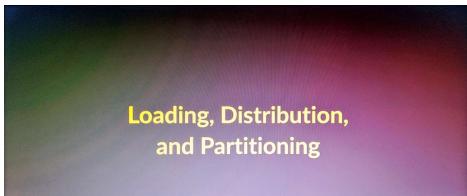


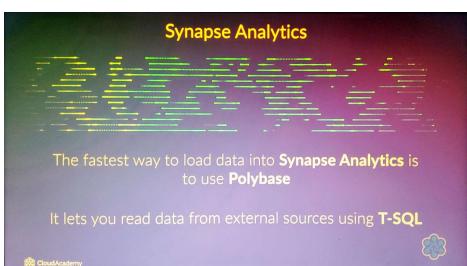
## Optimizing dedicated sql pools in A Synapse Analytics

Wednesday, January 4, 2023 7:53 PM

Course Optimizing dedicated sql pools in A sYNAPSE Analytics



Integration table → it is not part of a star schema.  
→ A common practice when loading data into a dedicated SQL pool is to first load the data into a staging table, perform some transformations on that data, and then load it into the SQL pool.



## STEP 2

Create external tables by using these three T-SQL commands in this order:

CREATE EXTERNAL DATA SOURCE  
CREATE EXTERNAL FILE FORMAT  
CREATE EXTERNAL TABLE

## STEP 3

Load the data into a staging table in Synapse Analytics

↳ This is a best practice, so you can deal with data loading issues without affecting production tables.

## STEP 4

Insert the data into production tables



When you're loading data into staging tables, you should use a round-robin distribution method



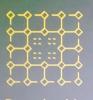
Tables in dedicated SQL pools are actually spread out across 60 data distributions

→ Esta es la  
forma de  
distribución  
en este servicio.  
→ this is why queries are so fast  
on this service, they're massively  
parallelized.

→ When you run a query, it spawns  
60 queries that each run on one  
data distribution.

→ To make this work efficiently, you have to decide how the data  
will be distributed. (Known as sharding), 3 options

Synapse Analytics offers three distribution choices



Round-robin      Hash-distributed      Replicated

Synapse Analytics offers three distribution choices



Rows are distributed evenly across the data distributions  
It's the fastest distribution type for loading data into a staging table

→ The simplest  
No optimization.  
It doesn't perform any optimization.

Synapse Analytics offers three distribution choices

→ A bit more complicated.  
→ As long as you choose a hash

**Synapse Analytics offers three distribution choices**



**Hash-distributed**

You designate one of the columns as the hash key

The hash function uses the value in this column to determine which data distribution to store a particular row on

→ A bit more complicated.  
 → As long as you choose a hash key that's appropriate for the most commonly run queries on this table, then query performance will be much better than it would be with a round-robin table.



You should choose a distribution column that will spread the rows fairly evenly among the data distributions  
 (but still uniform across).

→ If too many of the rows are on the same data distribution, then it will be a hot spot that reduces the advantages of Synapse Analytics' massively parallel architecture.



If you were to choose a date column for the hash key, then all of the rows for a particular date would end up on the same distribution

A query on that date would only run on that one distribution

→ Which would make the query take much longer than if it were to run across all 60 distributions in parallel.

**Some characteristics of a good distribution column:**

1. It has many unique values so the rows will be spread out over the 60 distributions
2. It's frequently used in JOINs
3. It's not used in WHERE clauses, as this would limit query matches to only a few distributions



→ If two fact tables are often joined together, then distribute both of the tables on the same join column. That way, rows from the 2 tables that have the same value in the join column will be stored on the same distribution, so they can be joined together easily.  
 → If you don't have frequent joins, then choose a column that's often in group by clauses.



Hash distribution is the recommended method for fact tables with a clustered columnstore index

→ This is the default.

**Synapse Analytics offers three distribution choices**

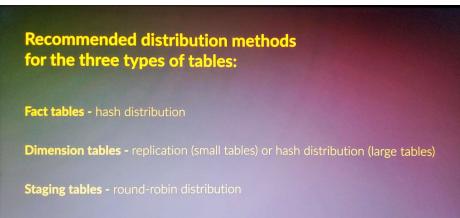


**Replicated**

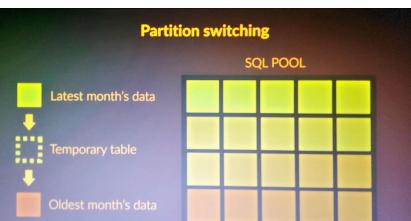
The entire table gets stored on each of the 60 data distributions

If a relatively small dimension table is frequently used in joins and aggregations, then it will be much more efficient to have it on every distribution

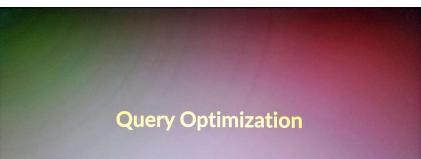
→ It's simpler.



→ In addition to distributing a table, you can also partition it by date range. (each month in separated partition), the benefit of doing this is that you could use something called partition switching



→ You can load the latest month's data into a temporary table, and then in the production table, replace the old partition with the new one.



→ to reduce both time queries take and the resources they consume.



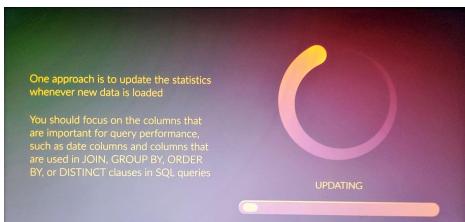
→ for ex, it estimates the number of rows to return.  
If it is a small amount, only a plan is used.  
If it is larger, it uses a different query plan.

In order to make the right decision, it needs to know your data pretty well.  
To do that, SQL generates statistics automatically, by default.



→ If the statistics aren't already there, the query will be slower the

first time, it's run because the statistics have to be generated.  
→ this is why it's important to generate statistics ahead of time, if possible.  
→ also, even if statistics have already been created they will become out-of-date as new data gets added.



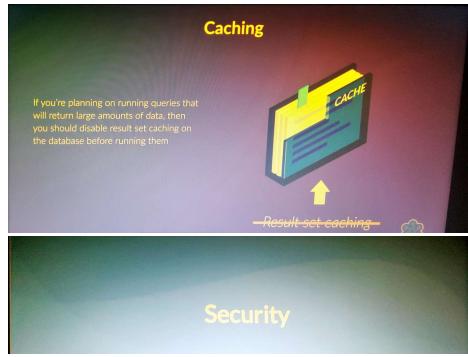
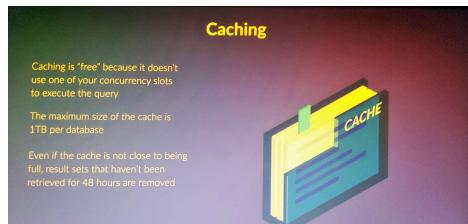
One approach is to update the statistics whenever new data is loaded

You should focus on the columns that are important for query performance, such as date columns and columns that are used in JOIN, GROUP BY, ORDER BY, or DISTINCT clauses in SQL queries



Data skew → A large number of rows come from one distribution (hotspot)  
 ↳ See go de datos.

Caching  
 One feature that dramatically speeds up queries is caching.  
 → If you enable result set caching, then the results from all queries (with a few exceptions) get cached.  
 → If you run again the same query, and the data hasn't changed in the meantime, then the results are retrieved from the cache instead of executing the query.



Synapse offers a wide variety of security options for dedicated SQL pools including:

- i) → Data Discovery & Classification.
  - ii) → Dynamic Data Masking
  - iii) → Vulnerability Assessment
  - iv) → Advanced Threat Protection
  - v) → transparent Data Encryption.
- i) → scans your database looking for sensitive data, such as names, addresses, and credit card numbers. It then gives you a list of recommendations for how these columns should be classified, such as "Confidential" or "Highly Confidential".  
 → If you accept the recommendations, then those columns will be labeled with those classifications.
- You can label them manually.
- After labelling, you can use the database auditing feature to monitor access to the sensitive data.
- ii) → will obscure some of the info in a particular column when

→ After labelling, you can use fine-grained access to the sensitive data.  
ii) → will obscure some of the info in a particular column when it is retrieved in a query.

iii) → scans your database looking for potential security issues, such as loose permissions and dangerous firewall settings, then you can go through the list of issues and decide whether or not they truly are issues that need to be addressed.

→ If you want to address an issue, there will be a recommendation to remediate, in many cases there will be a remediation script you can run.

iv) → look for unusual attempts to access or exploit databases.

v) → is used to encrypt the entire database, including the log files, and the backups, if a hacker gets a copy of the data, they don't be able to read any of the data in it



Azure takes snapshots of your Synapse Analytics data warehouse throughout the day

- If you need to restore your data warehouse to a previous state, you can simply choose which of the automatic restore points you want to go back to, or you can create an user-defined restore point, in addition to the automatic restore points.
- Both the automatic and user-defined restore points are stored in the primary region where your data warehouse runs, if it is down, you won't be able to access any of those restore points.
- To ensure that you will still be able to bring up your data warehouse in another region, Azure makes a geo-redundant backup once a day to a paired data center.
- If a disaster strikes, you can use that backup to restore a copy of your data warehouse to any region that supports Synapse Analytics.

Question ● CORRECT

When you are investigating query logs in Azure Synapse Analytics, if a large number of rows came from one distribution, this could indicate that you have \_\_\_\_\_

a round-robin table  
 an expired cache  
 a massively parallel architecture  
 data skew  
 I don't know yet

Explanation Once you know which query you want to investigate, you can retrieve both the SQL statement that was used and the query plan that was executed. You can also find out how many rows came from each distribution. If a large number of rows came from one distribution, this could indicate that you have a hot spot. This is also known as data skew.

Learn more: /course/optimizing-dedicated-sql-pools-azure-synapse-analytics-1476/query-optimization/

Something wrong with this question? Report an issue

Question 2 ● CORRECT

In Azure Synapse Analytics, the \_\_\_\_\_ service scans your database, looking for potential security issues such as loose permissions and dangerous firewall settings.

Transparent Data Encryption  
 Advanced Threat Protection  
 Vulnerability Assessment  
 Dynamic Data Masking

I don't know yet

Explanation The Vulnerability Assessment service scans your database, looking for potential security issues such as loose permissions and dangerous firewall settings.

Learn more: /course/optimizing-dedicated-sql-pools-azure-synapse-analytics-1476/security/

Something wrong with this question? Report an issue

Question 3 ● CORRECT

A(n) \_\_\_\_\_ table is generally a table that sits between a data source and the data warehouse.

integration  
 dimension  
 fact  
 star  
 I don't know yet

Explanation One type of table that isn't part of a star schema is called an integration table. This is generally a table that sits in between a data source and the data warehouse.

Learn more: /course/optimizing-dedicated-sql-pools-azure-synapse-analytics-1476/loading-distributing-and-partitioning/

Something wrong with this question? Report an issue

Question 4

CORRECT

In a data warehouse, the simplest type of schema is called a(n) \_\_\_\_\_ schema.

dimension  
 integration  
 fact  
 star  
 I don't know yet

Explanation

In a data warehouse, the simplest type of schema is called a star schema.

Learn more: /course/optimizing-dedicated-sql-pools-azure-synapse-analytics-1476/loading-distributing-and-partitioning/

Bookmark

Question 5

CORRECT

Which T-SQL command comes first in the steps to creating external tables in Azure Synapse Analytics?

CREATE EXTERNAL FILE FORMAT  
 CREATE EXTERNAL DATA SOURCE  
 CREATE EXTERNAL DATA  
 CREATE EXTERNAL TABLE  
 I don't know yet

Explanation

Create external tables by using these three T-SQL commands in this order: CREATE EXTERNAL DATA SOURCE, CREATE EXTERNAL FILE FORMAT, and CREATE EXTERNAL TABLE.

Learn more: /course/optimizing-dedicated-sql-pools-azure-synapse-analytics-1476/loading-distributing-and-partitioning/

Something wrong with this question? Report an issue

Bookmark