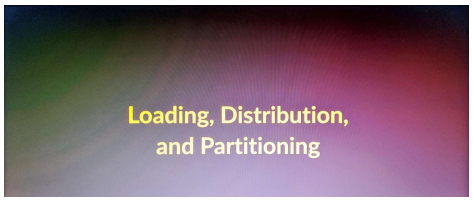
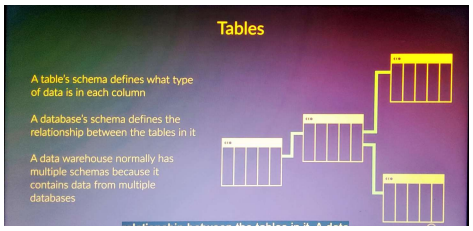
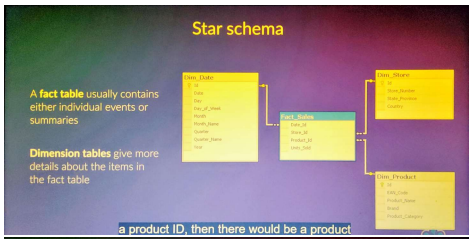
Optimizing dedicated sql pools in A Synapse Analytics

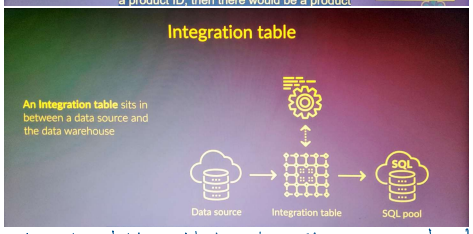
Wednesday, January 4, 2023 7:53 PM

CourseOptimizing dedicaredsql poolsin A sYNAPSE Analytics



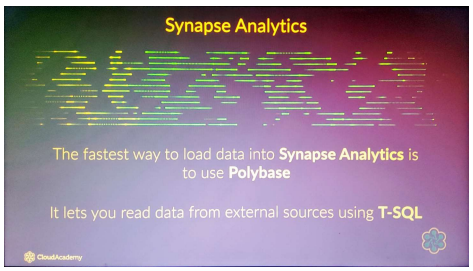




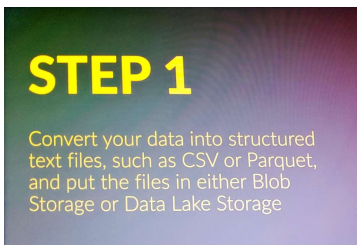


Integration table → it is not part of a start schema.

→ 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.



↳ Steps:



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

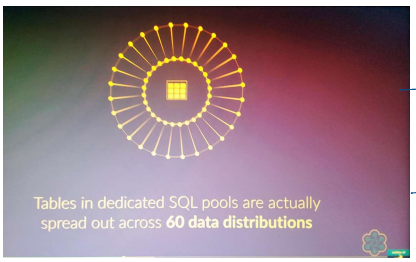
STEP 3

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

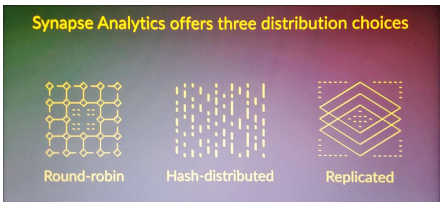
Insert the data into production tables

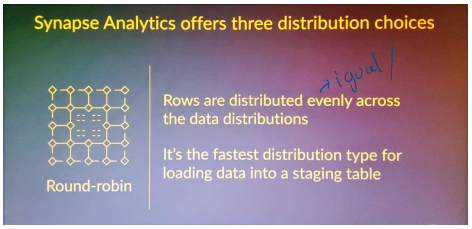


→ this is why queries are so fast on this service, they're massively parallelized.
(esta es la razón)

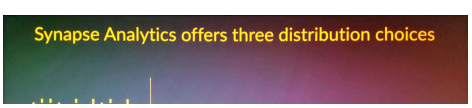
→ when you run a query, it spawns 60 queries that each run on one data distribution. (engendra)

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

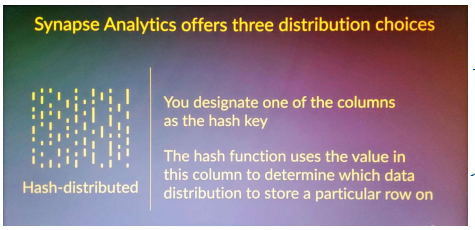




→ (igual)
→ the simplest
No optimize.
It doesn't perform any optimization.

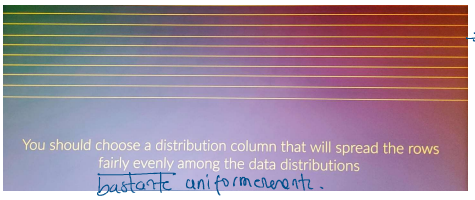


→ A bit more complicated.
→ As long as you choose a hash

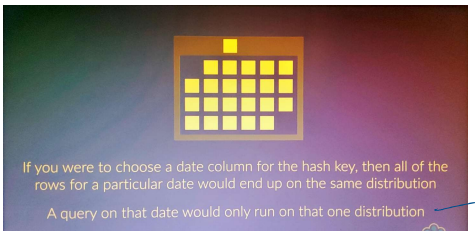


→ A bit more complicated.

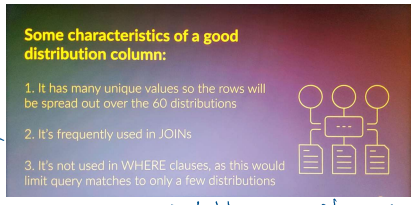
→ As long as you choose a hash key that is appropiate for the most commonly run queries on this table, then query performance will be much better than it would be with a round-robin table.



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

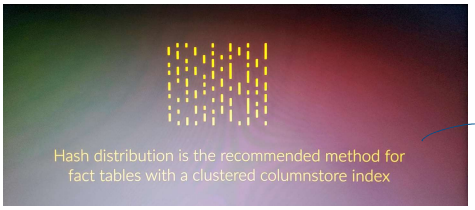


→ which would make the query take much longer than if it were to run across all 60 distributions in pacallel.

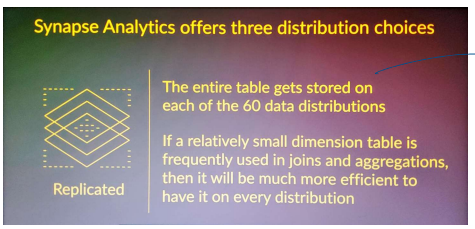


→ If two fact tables are often joined together, then distribute both of the tables on the same join column. that way, rows from the 2 tables that have the same value in the join column will be stored on the same distribution, so they can be joined together easily.

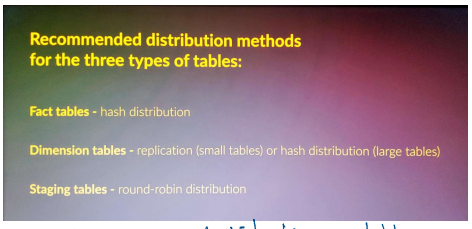
→ If you don't have frequent joins, then choose a column that's often in group by clauses.



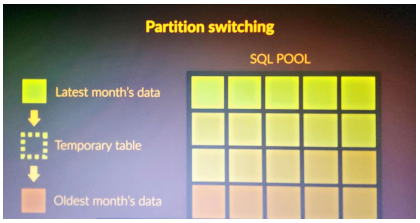
→ this is the default.



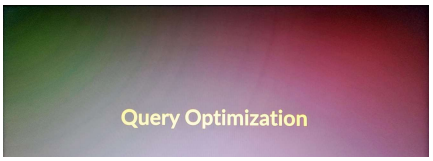
→ It's simpler.



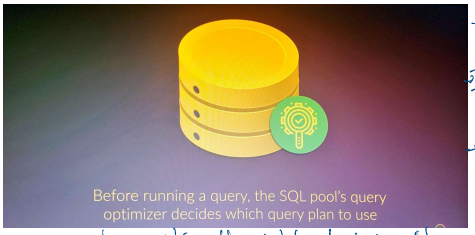
→ In addition to distributing a table, you can also partition it by date range. (each month in separated partition), the benefit of doing this is that you could use something called partition switching



→ 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.

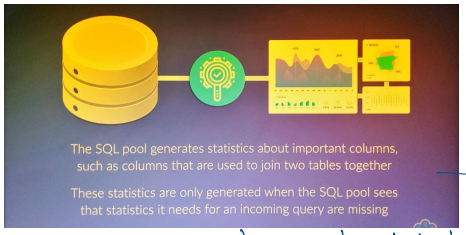


→ to reduce both time queries take and the resources they consume.

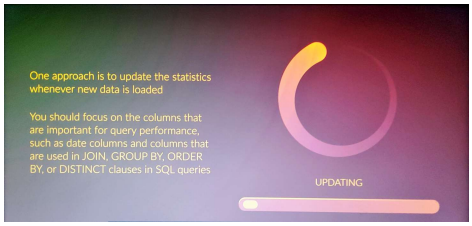


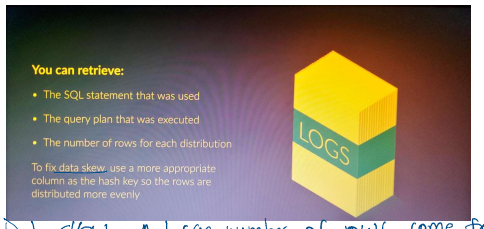
→ for ex, it estimates the number of rows to return.
→ If it is a small amount, only a plan is used.
→ If it is larger, it uses a different query plan.

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.

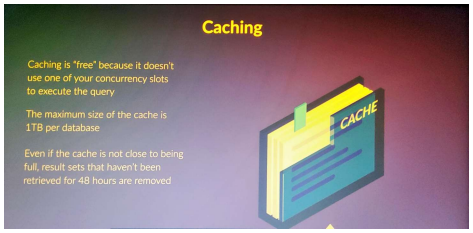


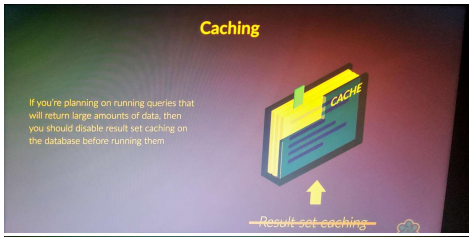


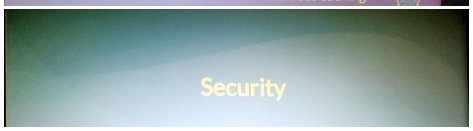
~~Data skew~~ = A large number of rows come from one distribution (hotspot)
↳ Sesgo de datos.

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 from executing the query.







Synapse offers a wide variety options for security for dedicated SQL pools including:
i) → Data Discovery & Classification.
ii) → Dynamic data Masking
iii) → Vulnerability Assesment
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

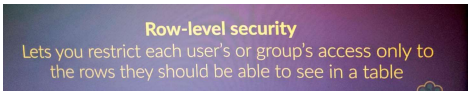
→ After labelling, you can use the [illegible] access to the sensitive data.

ii) → will obscure some of the info in a particular column when it is retrieved in a query.

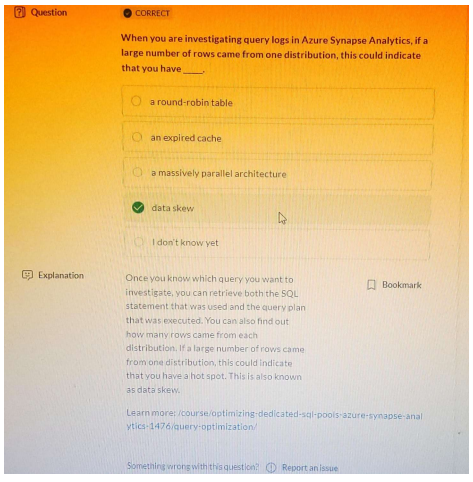
iii) → scans your database looking for potential security issues, such as loose permissions and dangerous firewall settings. Then you can go through the list of issues and decide whether or not they truly are issues that need to be addressed.
→ If you want to address an issue, there will be a recommendation to remediate, in many cases there will be a remediation script you can run.

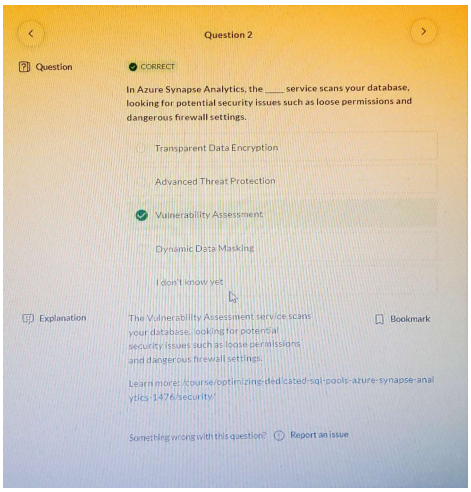
iv) → look for unusual attempts to access or exploit databases.

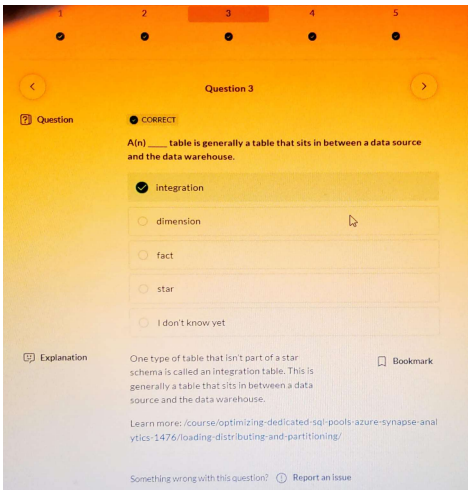
v) → is used to encrypt the entire database, including the log files, and the backups. If a hacker gets a copy of the data, they don't be able to read any of the data in it

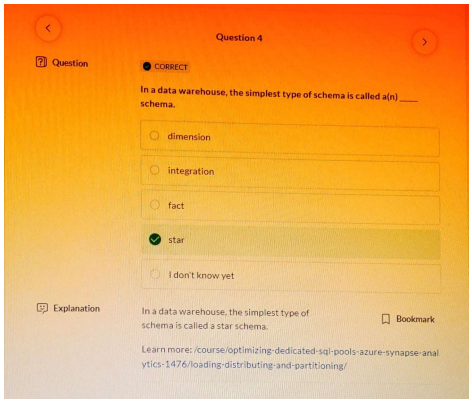


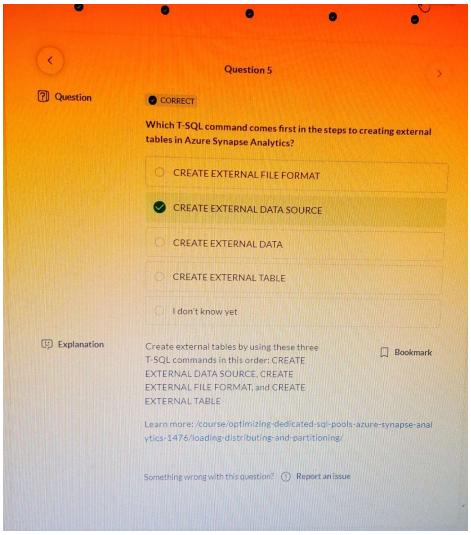
→ If you need to restore your data warehouse to a previous state, you can simply choose which of the automatic restore points you want to go back to.
or you can create an user defined restore point, in addition, to the automatic restore points.
→ Both the automatic and user-defined restore points are 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 4
Question
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.
Bookmark
Learn more: /course/optimizing-dedicated-sql-pools-azure-synapse-analytics-1476/loading-distributing-and-partitioning/


Question 5
Question
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
Bookmark
Learn more: /course/optimizing-dedicated-sql-pools-azure-synapse-analytics-1476/loading-distributing-and-partitioning/
Something wrong with this question?
Report an issue