Cloudera Runtime 7.1.8

# Phoenix Frequently Asked Questions

**Date published: 2020-02-29**
**Date modified: 2021-09-09**

## CLOUDERA

# Legal Notice

# Contents

# Frequently asked questions

Find answers to frequently asked questions around Apache Phoenix and its deployment.

### Can Phoenix be used for ETL use cases?

Yes. Apache Phoenix is used for OLTP (Online Transactional Processing) use cases and not OLAP (Online Analytical Processing) use cases. Although, you can use Phoenix for real-time data ingest as a primary use case.

### What is the typical architecture for a Phoenix deployment?

A typical Phoenix deployment has the following:

- Application
- Phoenix Client/JDBC driver
- HBase client

A Phoenix client/JDBC driver is essentially a Java library that you should include in your Java code. Phoenix uses HBase as storage similar to how HBase uses HDFS as storage. However, the abstraction for Phoenix is not yet complete, for example, for implementing access controls, you need to set ACLs on the underlying HBase tables that contain the Phoenix data.
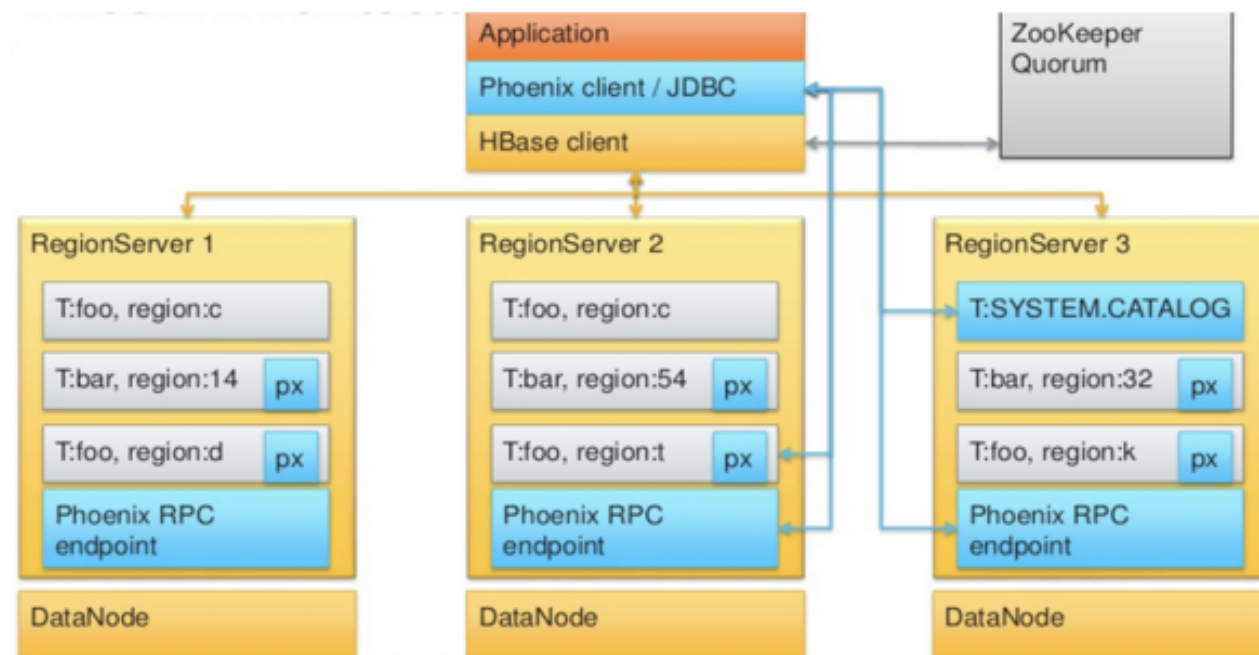


### Are there sizing guidelines for Phoenix JDBC servers?

For Phoenix applications, you must follow the same sizing guidelines that you follow for HBase. For more information about Phoenix performance tuning, https://phoenix.apache.org/tuning_guide.html.

### Can I govern access to the Phoenix servers?

Yes, you can use Kerberos for authentication. You can configure authorization using HBase authorization.

## Can I see the individual cell timestamps in Phoenix tables? Is this something that's commonly used?

You can map HBase's native row timestamp to a Phoenix column. By doing this, you can take advantage of the various optimizations that HBase provides for time ranges on the store files as well as various query optimization capabilities built within Phoenix.

For more information, see https://phoenix.apache.org/rowtimestamp.html

## What if the Phoenix index is being built asynchronously and data is added to a table during indexing?

Phoenix does local Indexing for deadlock prevention during global index maintenance.: Phoenix also atomically rebuild index partially when index update fails (PHOENIX-1112).

## How do sequences work in Phoenix?

Sequences are a standard SQL feature that allows for generating monotonically increasing numbers typically used to form an ID.

For more information, see https://phoenix.apache.org/sequences.html.

## What happens to a Phoenix write when a RegionServer fails?

Writes are durable and durability is defined by a WRITE that is committed to disk (in the Write Ahead Log). So in case of a RegionServer failure, the write is recoverable by replaying the WAL. A "complete" write is one that has been flushed from the WAL to an HFile. Any failures will be represented as exceptions.

## Can I do bulk data loads in Phoenix?

Yes, you can do bulk inserts in Phoenix. For more information see https://phoenix.apache.org/bulk_dataload.html.

## Can I access a table created by Phoenix using the standard HBase API?

Yes, but is it is not recommended or supported. Data is encoded by Phoenix, so you have to decode the data for reading. Writing to the HBase tables directly would result in corruption in Phoenix.

## Can I map a Phoenix table over an existing HBase table?

Yes, as long as Phoenix data types are used. You have to use asynchronous indexes and manually update them since Phoenix won't be aware of any updates.

## What are guideposts?

For information about guideposts, see https://phoenix.apache.org/update_statistics.html.