

This page describes how to do reads in Cloud Spanner outside the context of read-only and read-write transactions. That means if either of the following applies, you should read the [Transactions \(/spanner/docs/transactions\)](/spanner/docs/transactions) page instead:

- **If you need to write, depending on the value of one or more reads**, you should execute the read as part of a read-write transaction. Read more about [read-write transactions \(/spanner/docs/transactions#read-write\\_transactions\)](/spanner/docs/transactions#read-write_transactions).
- **If you are making multiple read calls that require a consistent view of your data**, you should execute the reads as part of a read-only transaction. Read more about [read-only transactions \(/spanner/docs/transactions#read-only\\_transactions\)](/spanner/docs/transactions#read-only_transactions).

If you need to make a single read call or read data in parallel and you don't also need to write, read on.

Cloud Spanner allows you to determine how current the data should be when you read data by offering two types of reads:

- A *strong read* is a read at a current timestamp and is guaranteed to see all data that has been committed up until the start of this read. Cloud Spanner defaults to using strong reads to serve read requests.
- A *stale read* is read at a timestamp in the past. If your application is latency sensitive but tolerant of stale data, then stale reads can provide performance benefits.

To choose which type of read you want, set a [timestamp bound \(/spanner/docs/timestamp-bounds\)](/spanner/docs/timestamp-bounds) on the read request. Use the following best practices when choosing a timestamp bound:

- **Choose strong reads whenever possible.** These are the default timestamp bound for Cloud Spanner reads, including read-only transactions. Strong reads are guaranteed to observe the effects of all transactions that committed before the start of the operation, independent of which replica receives the read. Because of this, strong reads make application code simpler

and applications more trustworthy. Read more about Cloud Spanner's consistency properties in [TrueTime and External Consistency](/spanner/docs/true-time-external-consistency) (/spanner/docs/true-time-external-consistency).

- **If latency makes strong reads infeasible in some situations, then use stale reads** (bounded-staleness or exact-staleness) to improve performance in places where you do not need reads to be as recent as possible. As described in [Cloud Spanner Replication](/spanner/docs/replication#read-only) (/spanner/docs/replication#read-only), 15 seconds is a reasonable staleness value to use for good performance.

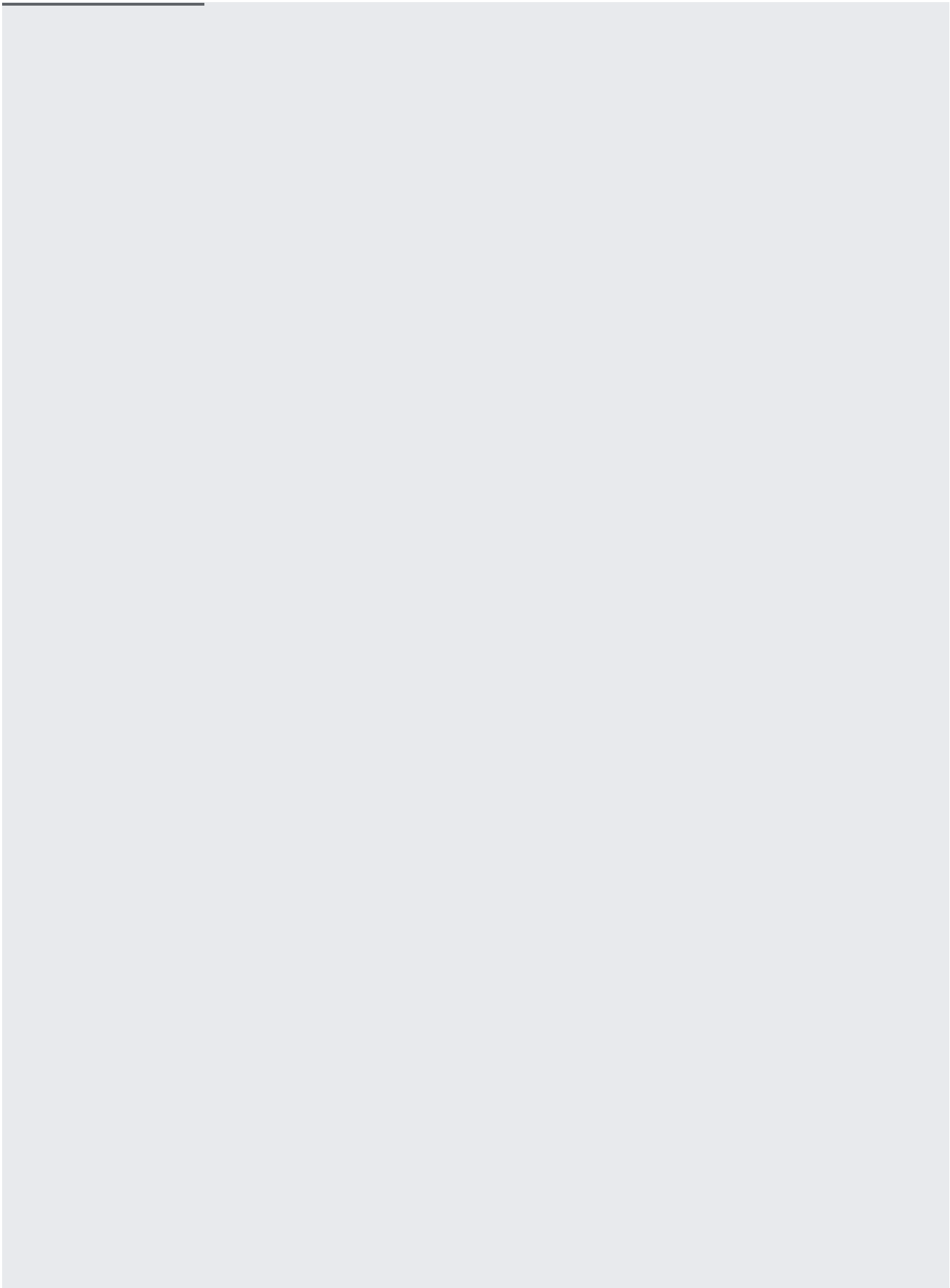
Stale reads do not provide any latency benefits in regional configurations, so you should almost always use strong reads if your instance does not have a multi-region configuration.

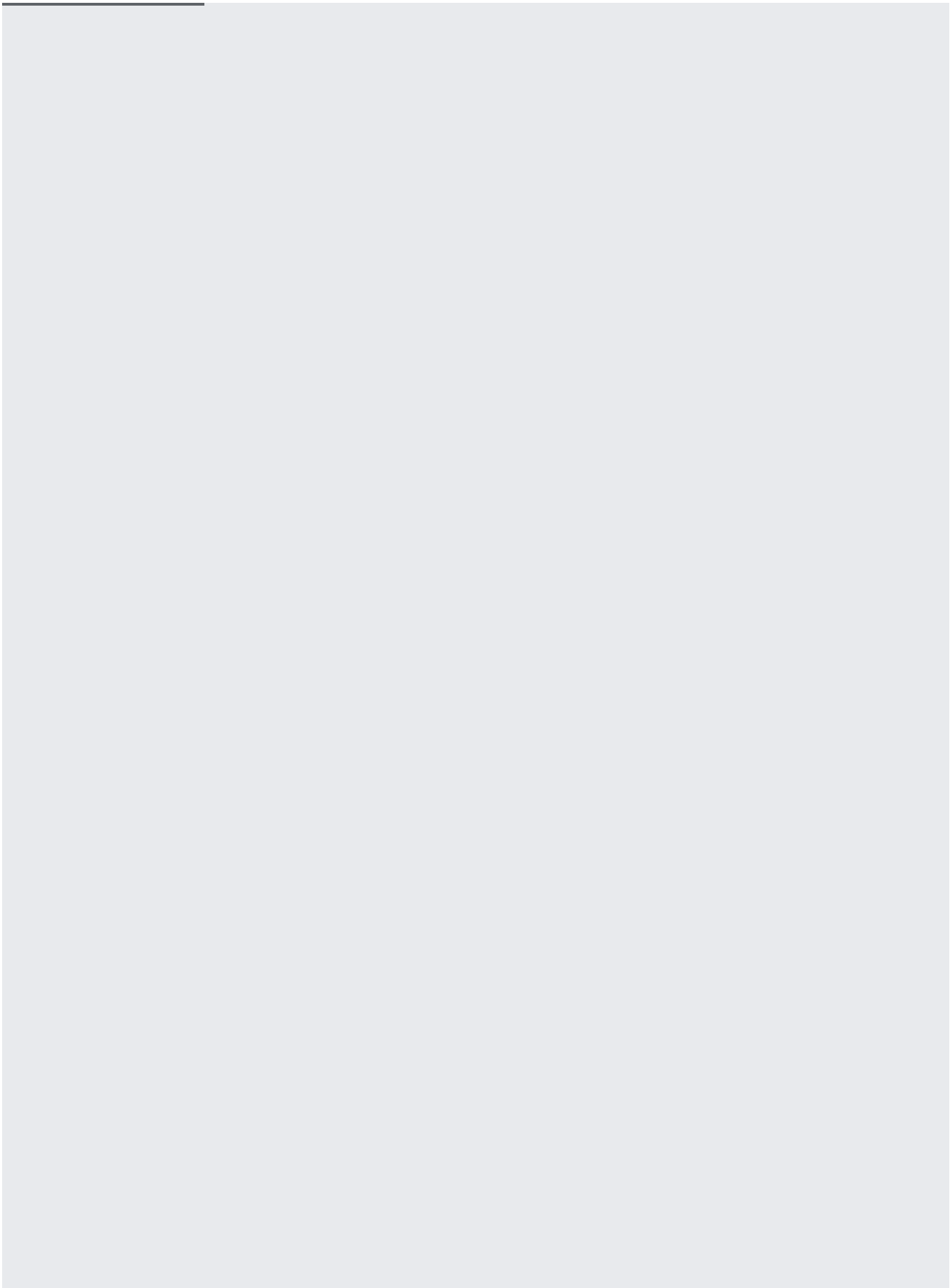
Cloud Spanner supports single read methods (that is, a read outside the context of a transaction) on a database for:

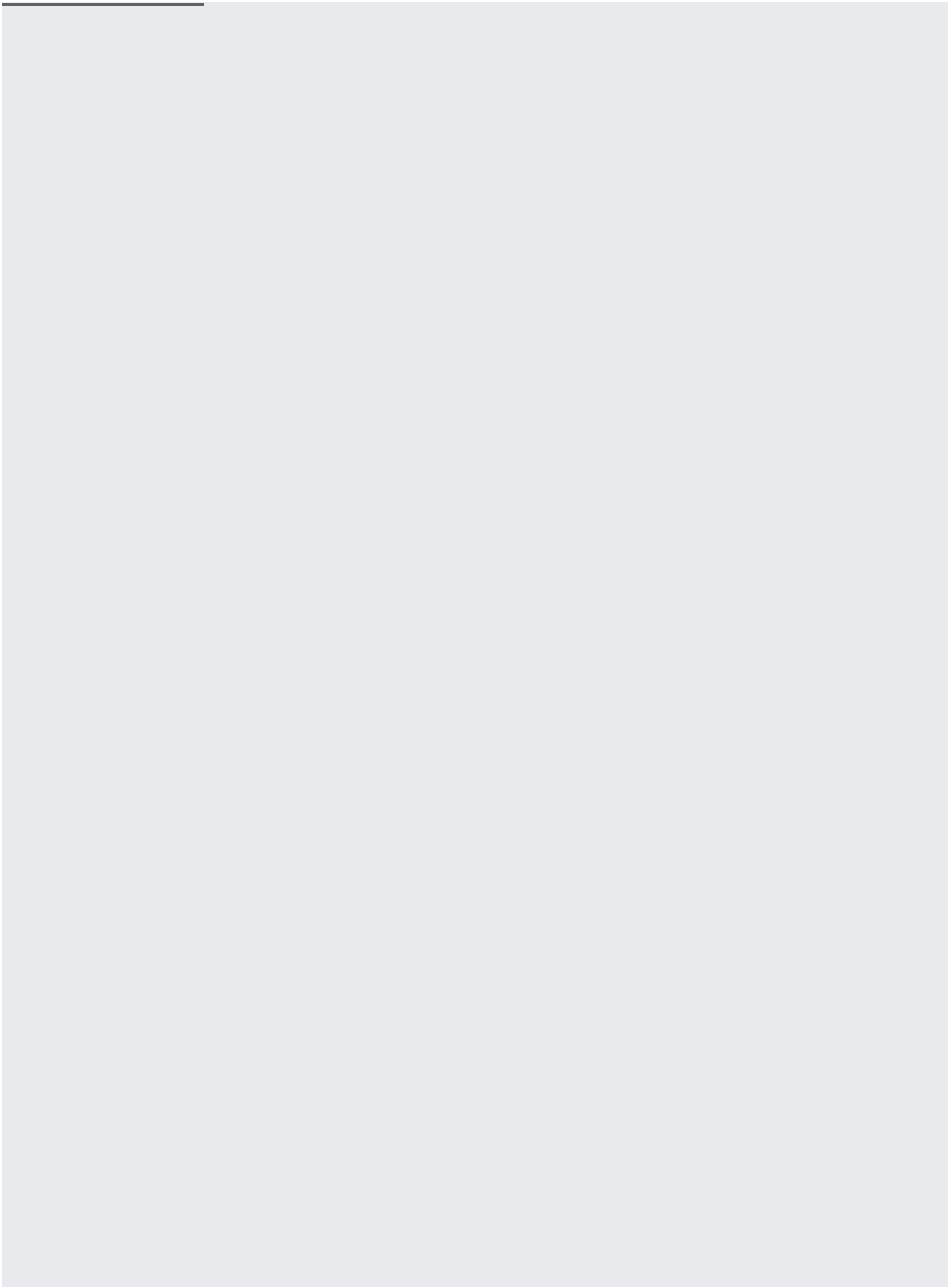
- Executing the read as a SQL query statement or using Cloud Spanner's read API.
- Performing a strong read from a single row or multiple rows in a table.
- Performing a stale read from a single row or multiple rows in a table.
- Reading from a single row or multiple rows in a secondary index.

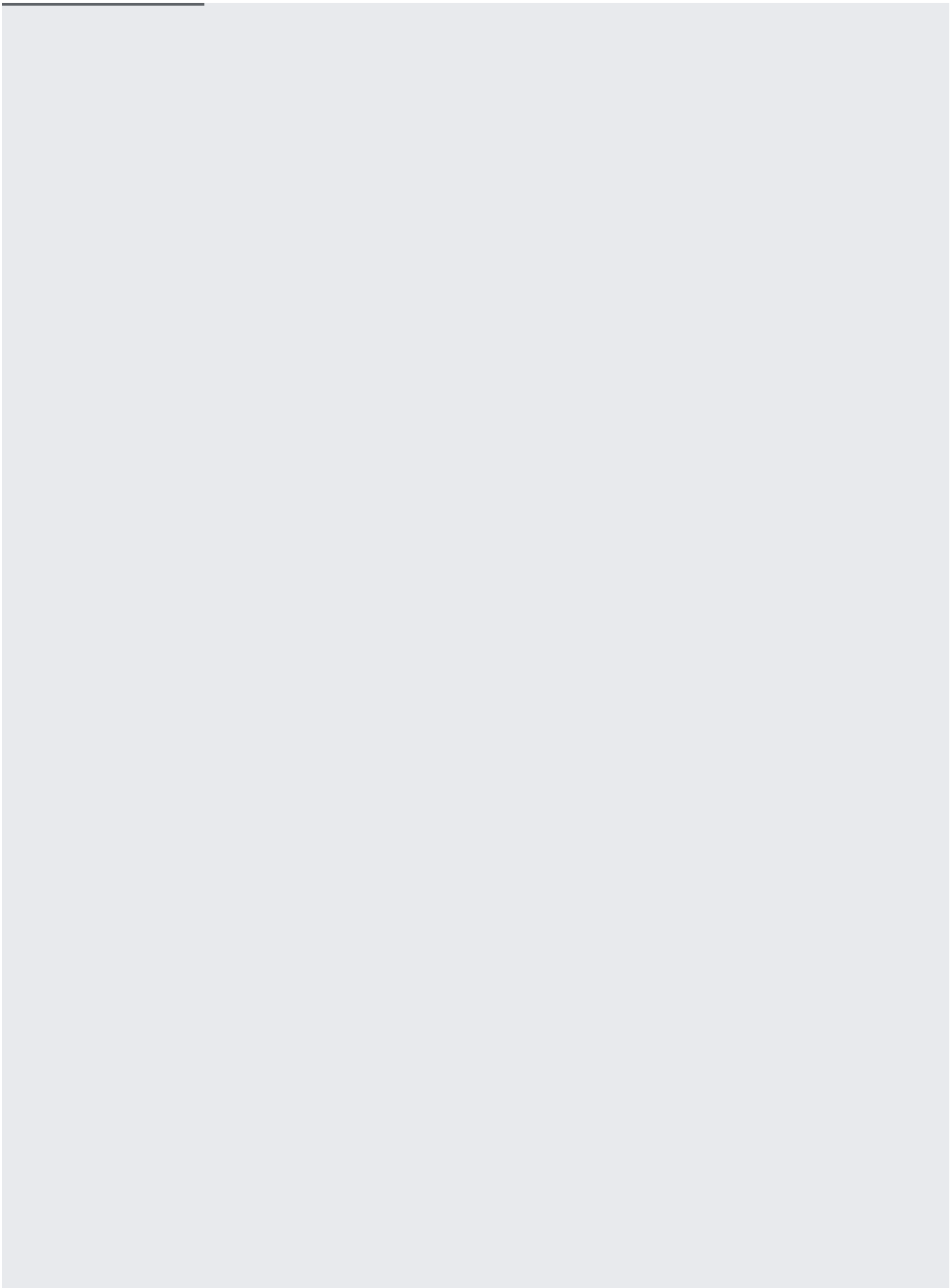
The sections below describe how to use read methods using the Cloud Client Libraries for the Cloud Spanner API.

The following shows how to execute a SQL query statement against a database.



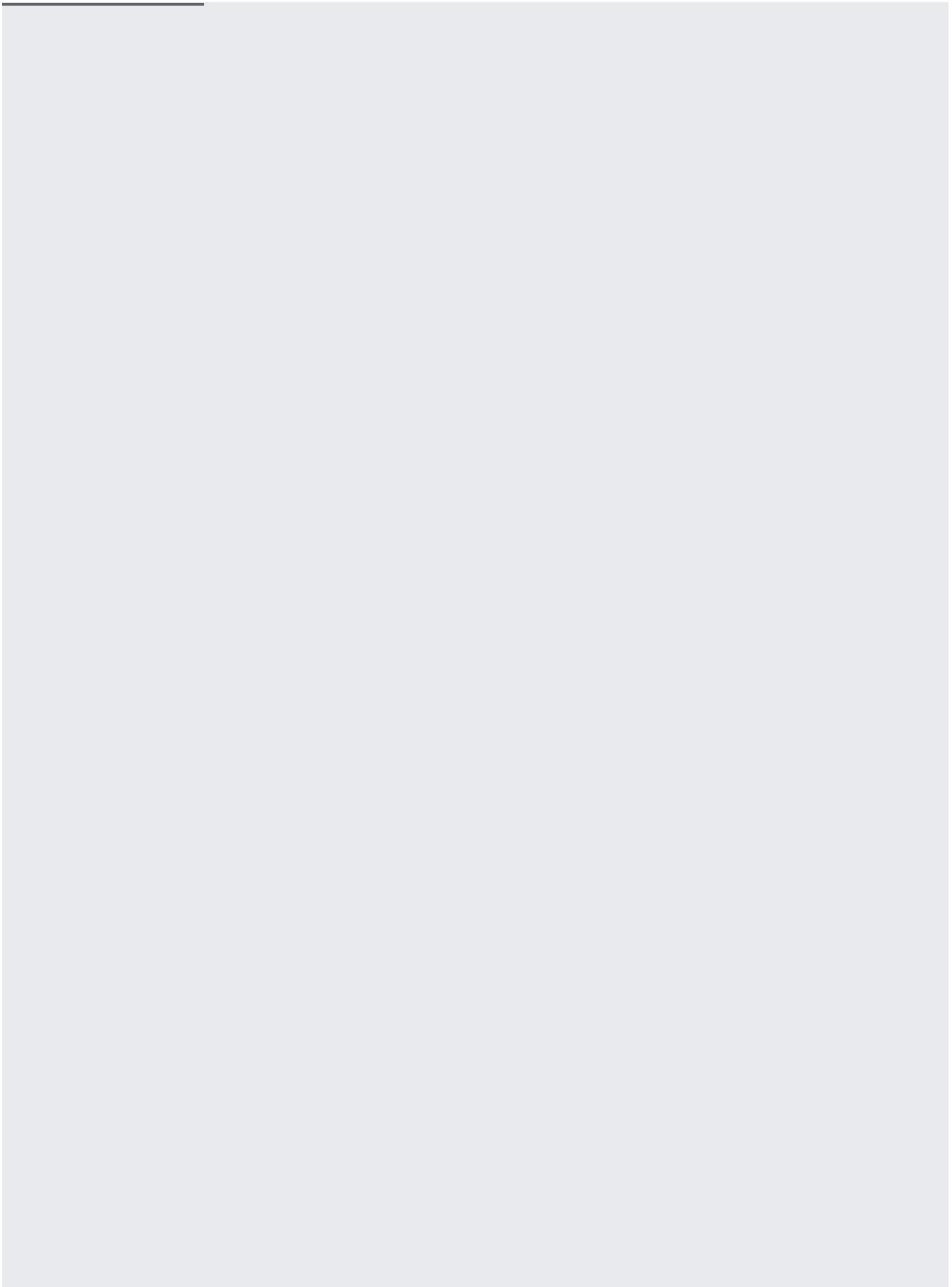




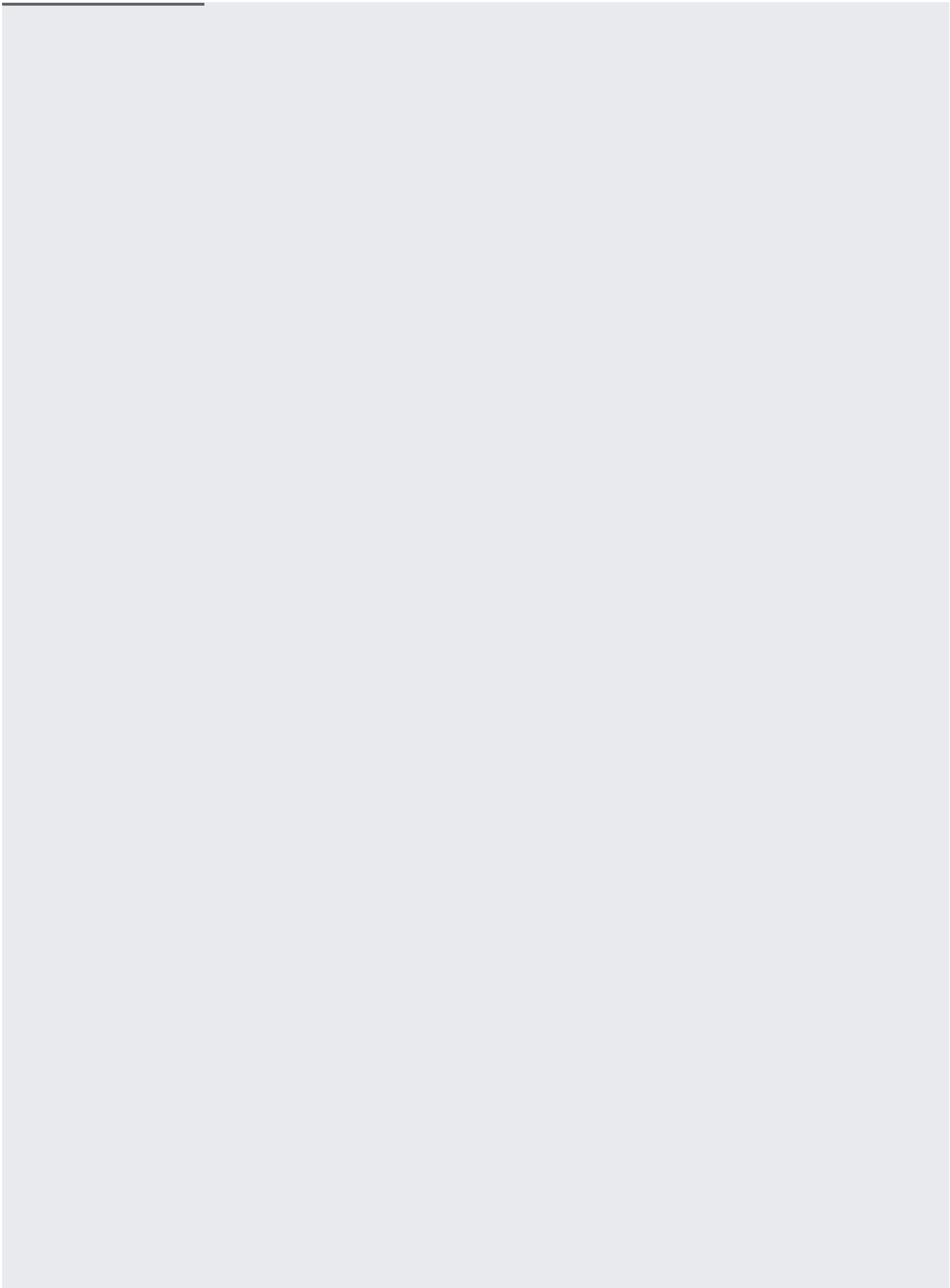


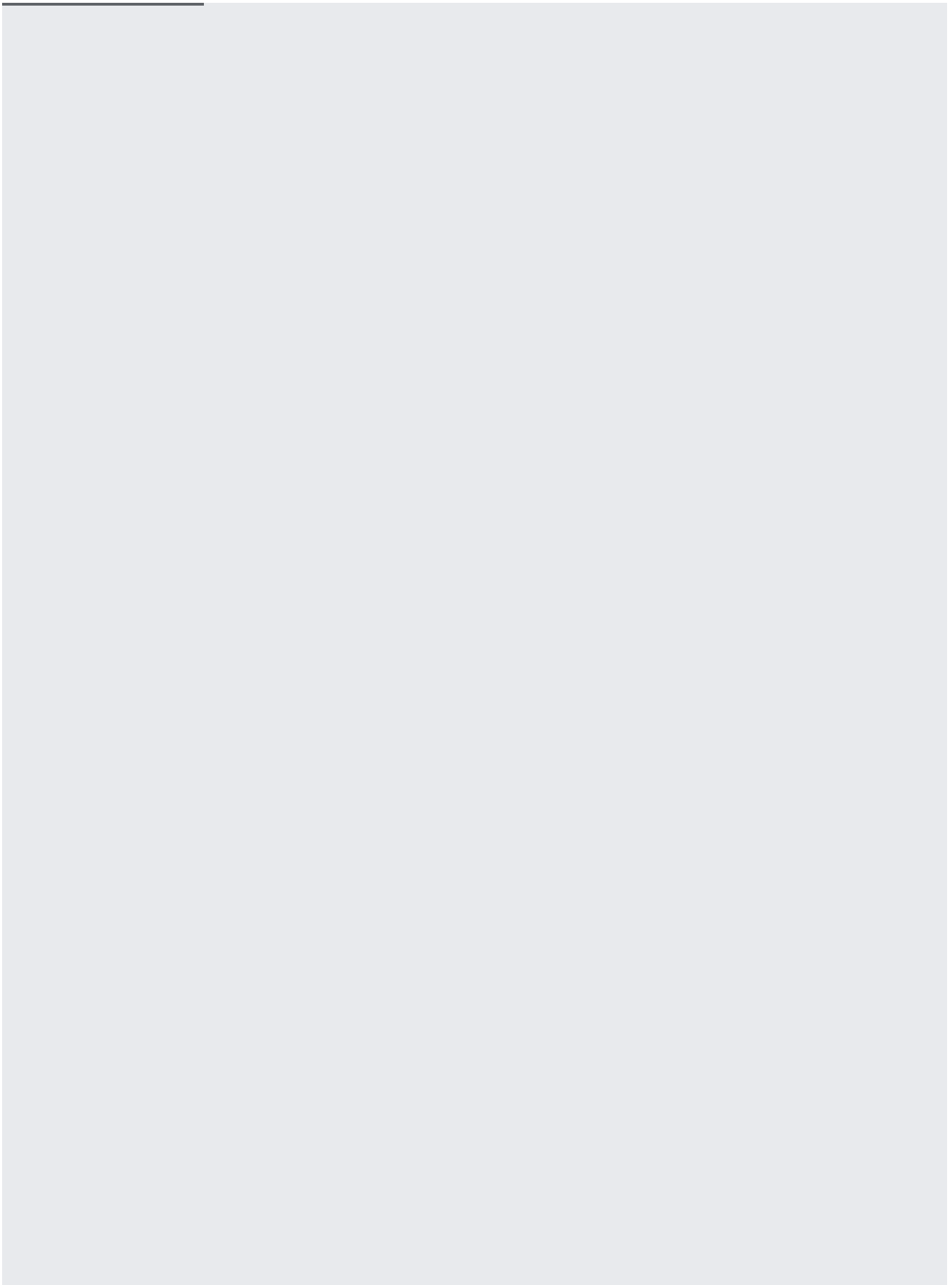
Consult the SQL [Query Syntax](/spanner/docs/query-syntax) (/spanner/docs/query-syntax) and [Functions and Operators](/spanner/docs/functions-and-operators) (/spanner/docs/functions-and-operators) references when constructing a SQL statement.

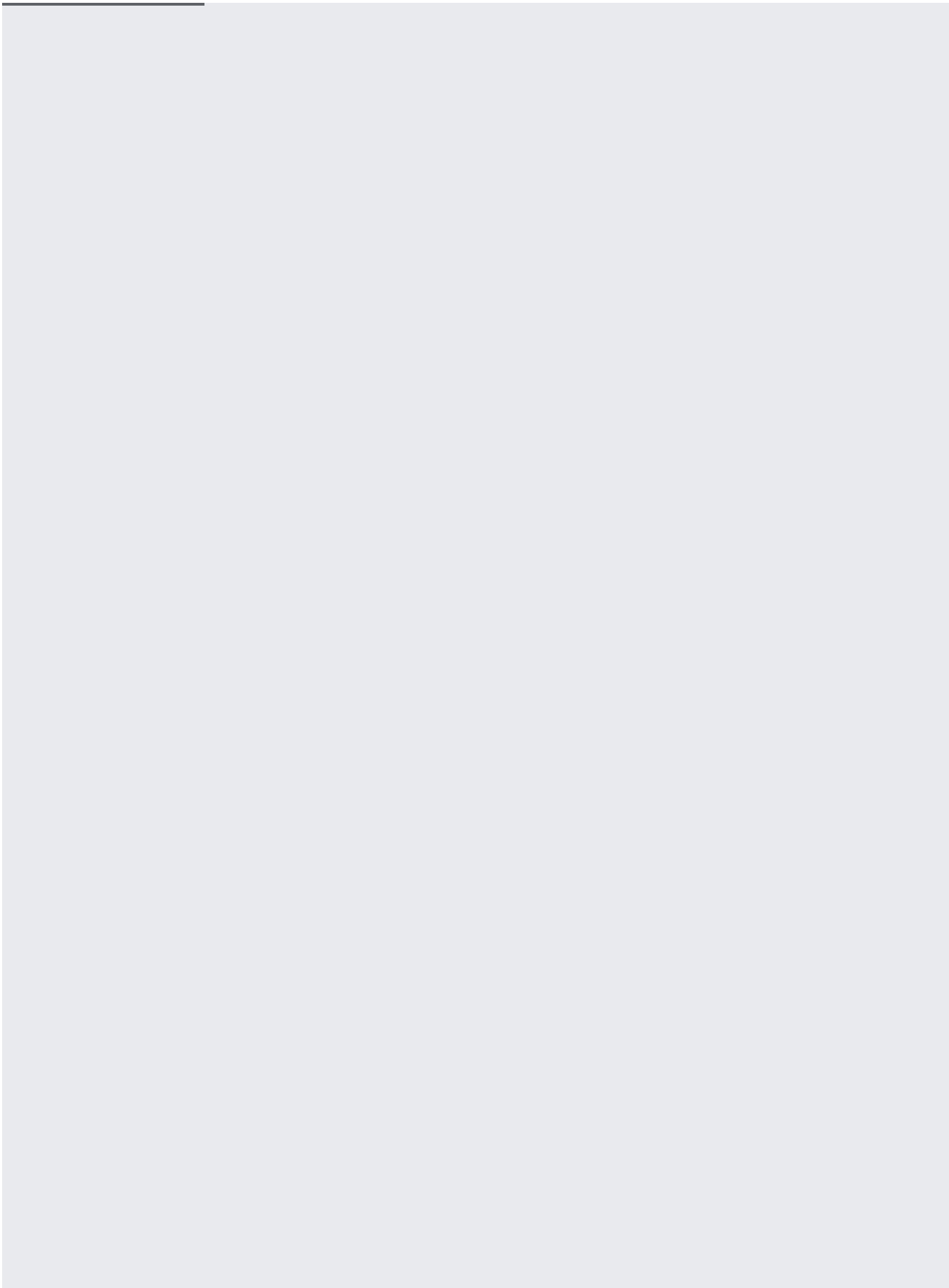
The following shows how to perform a strong read of zero or more rows from a database.

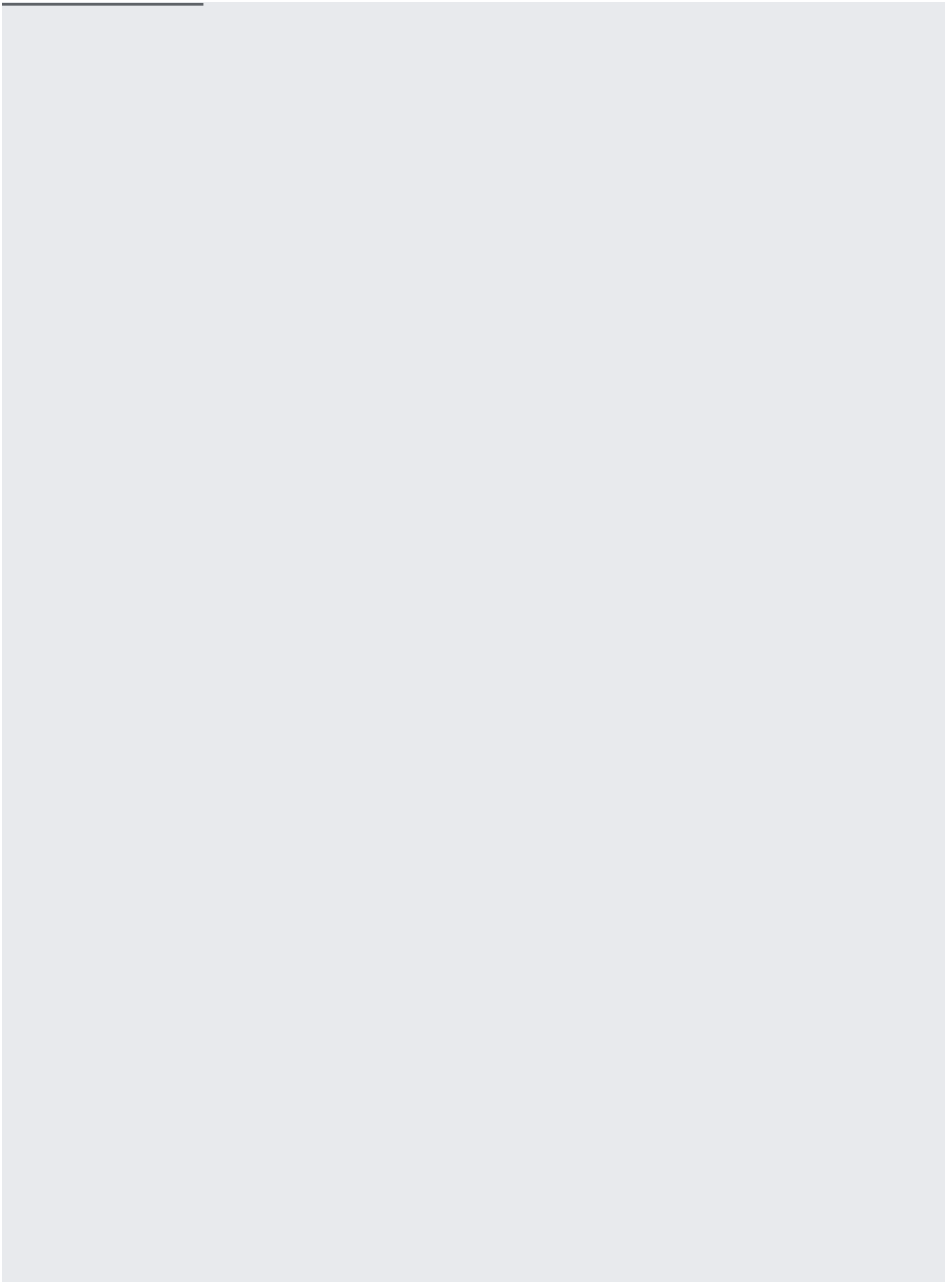




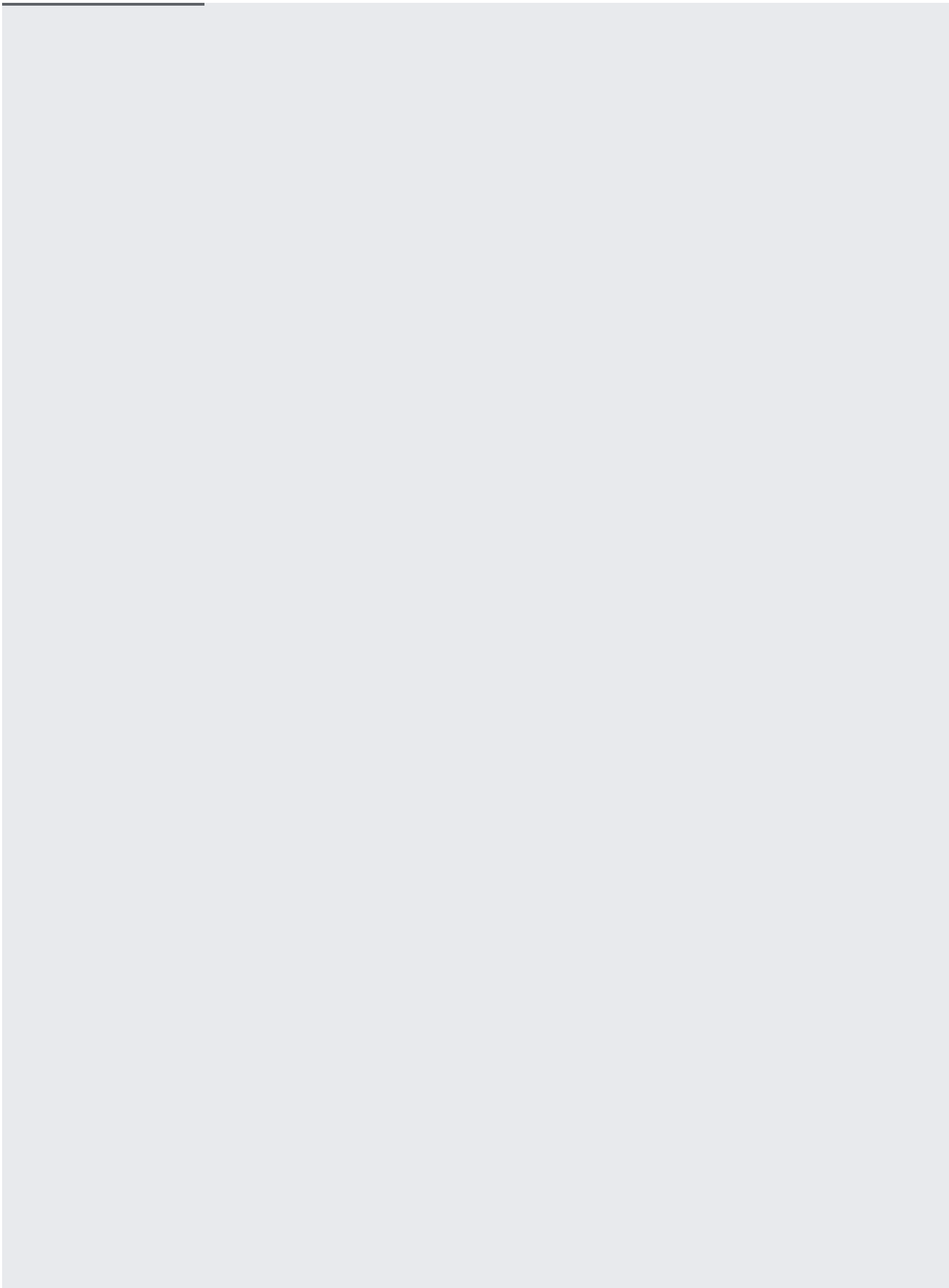


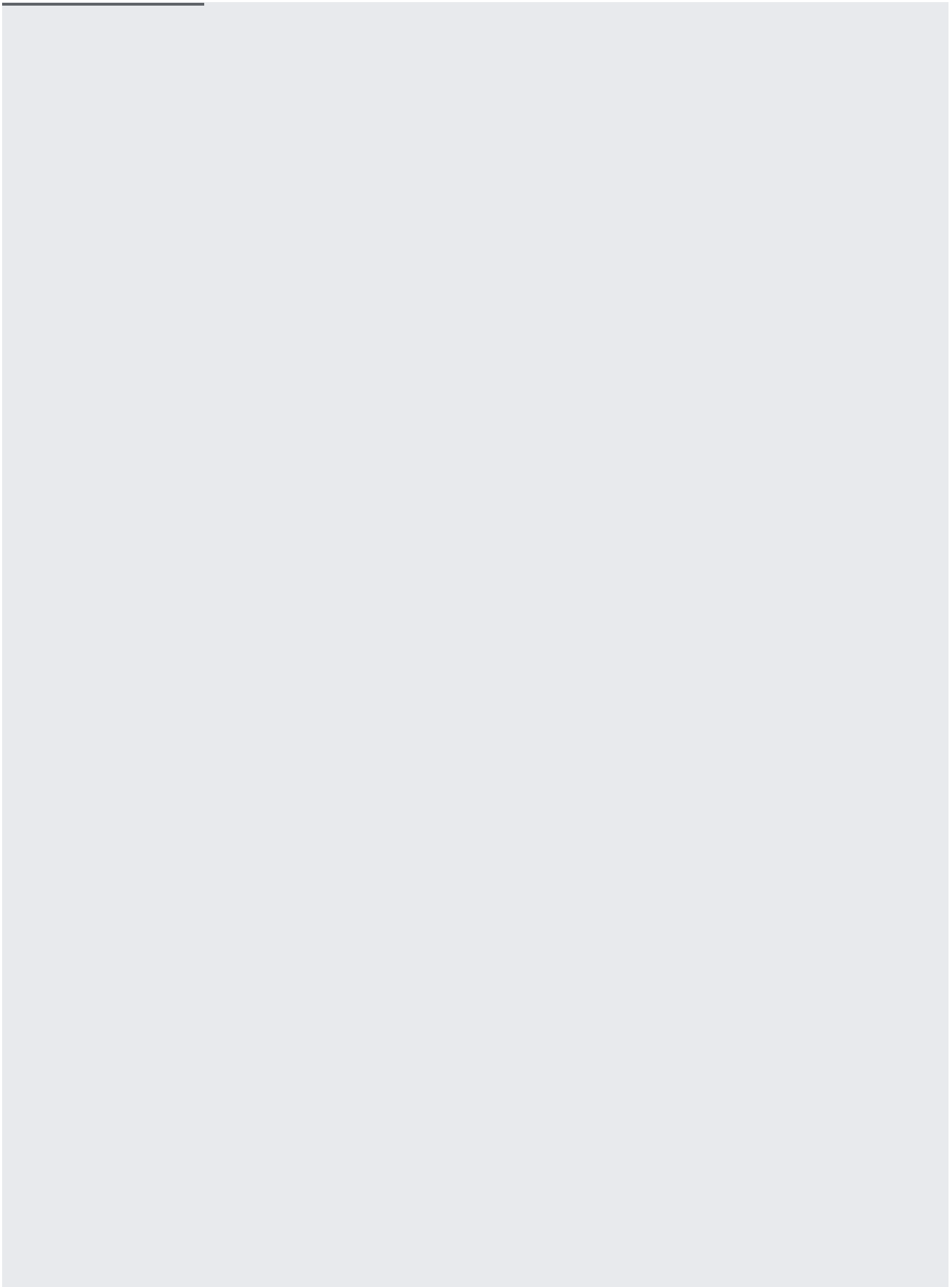


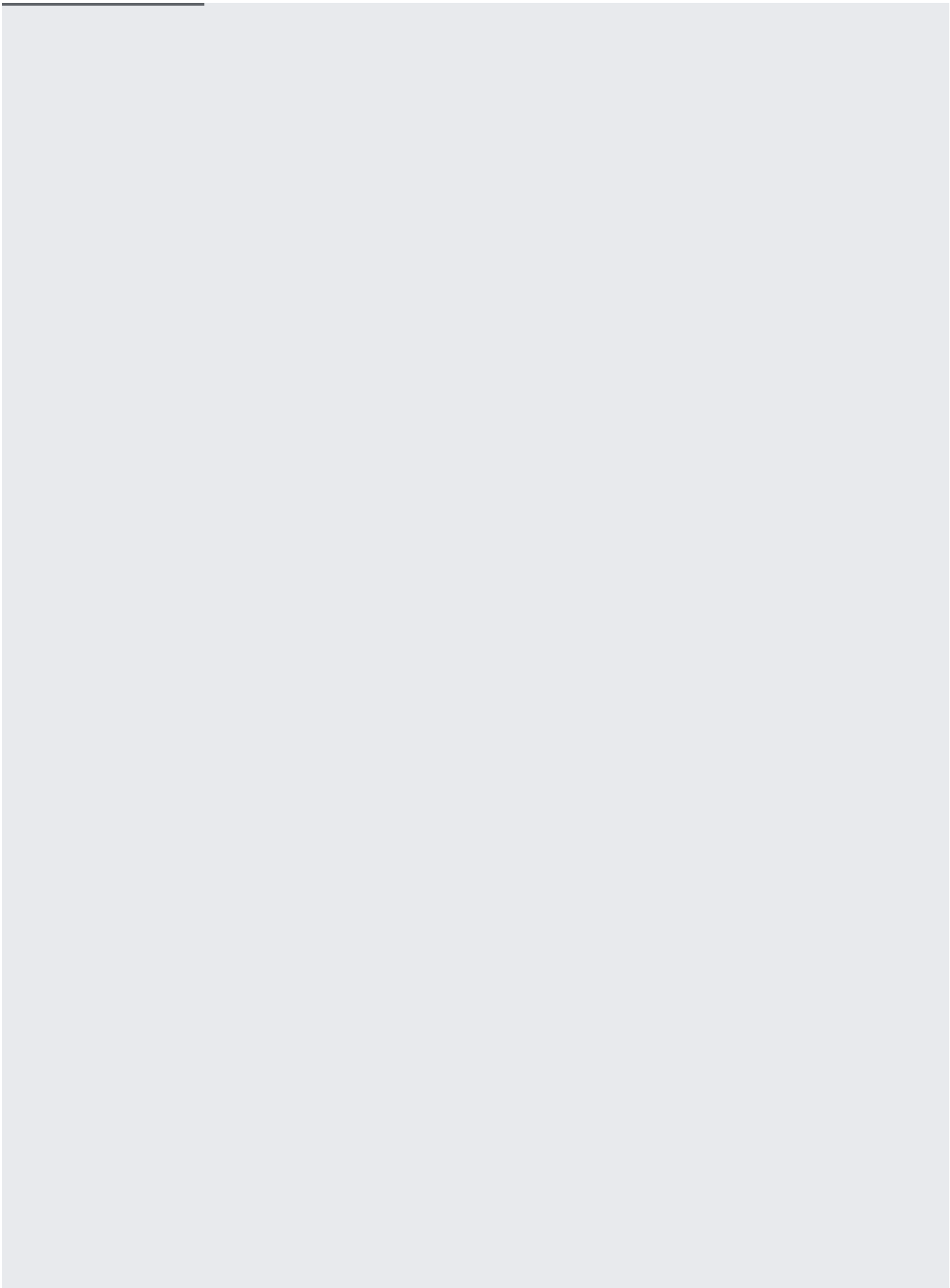




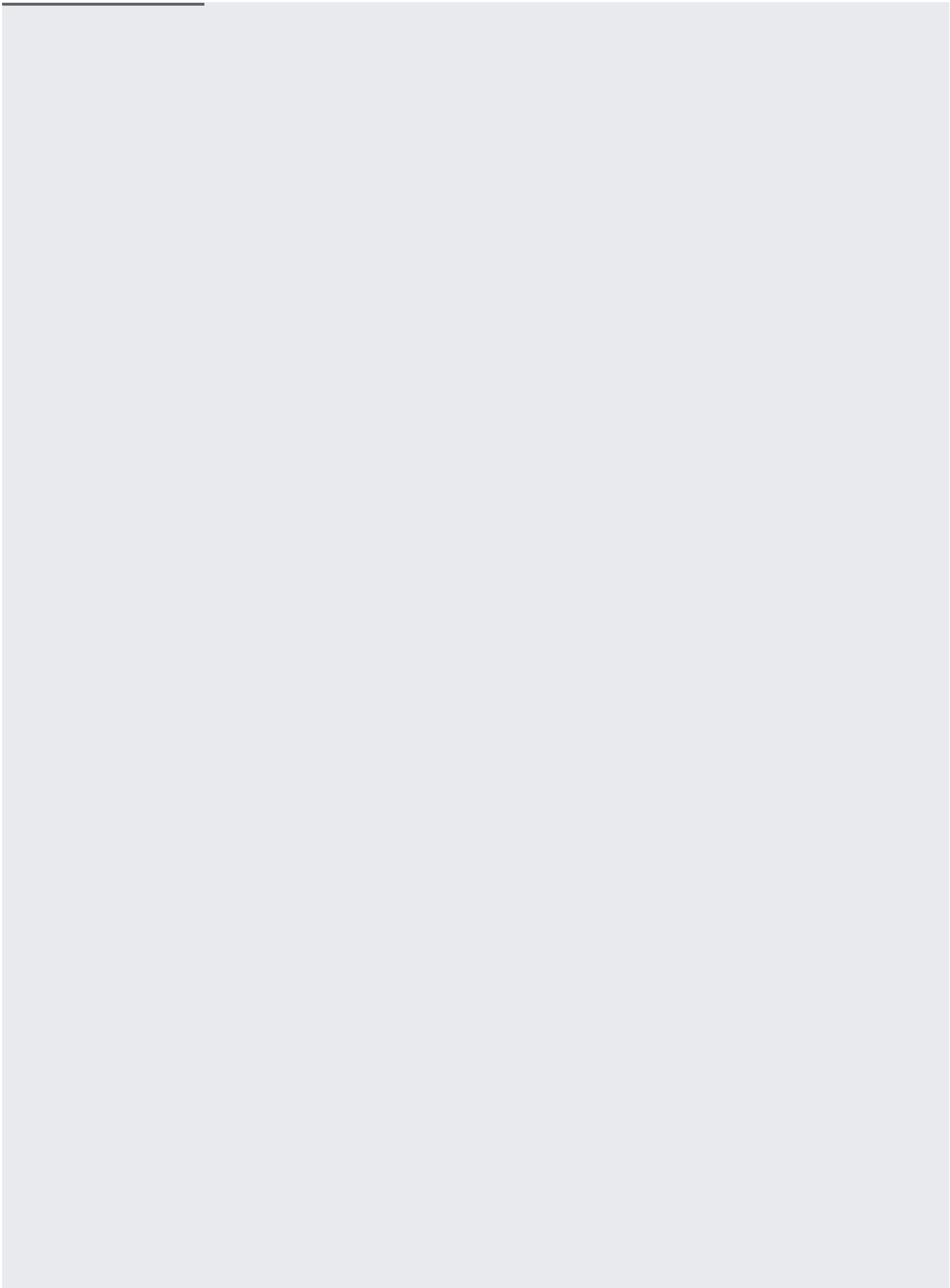
The following sample code shows how to perform a stale read of zero or more rows from a database using an **exact-staleness** timestamp bound. For instructions on how to perform a stale read using a **bounded-staleness** timestamp bound, see the note after the sample code. See [Timestamp bounds](/spanner/docs/timestamp-bounds) (/spanner/docs/timestamp-bounds) for more information on the different types of timestamp bounds that are available.

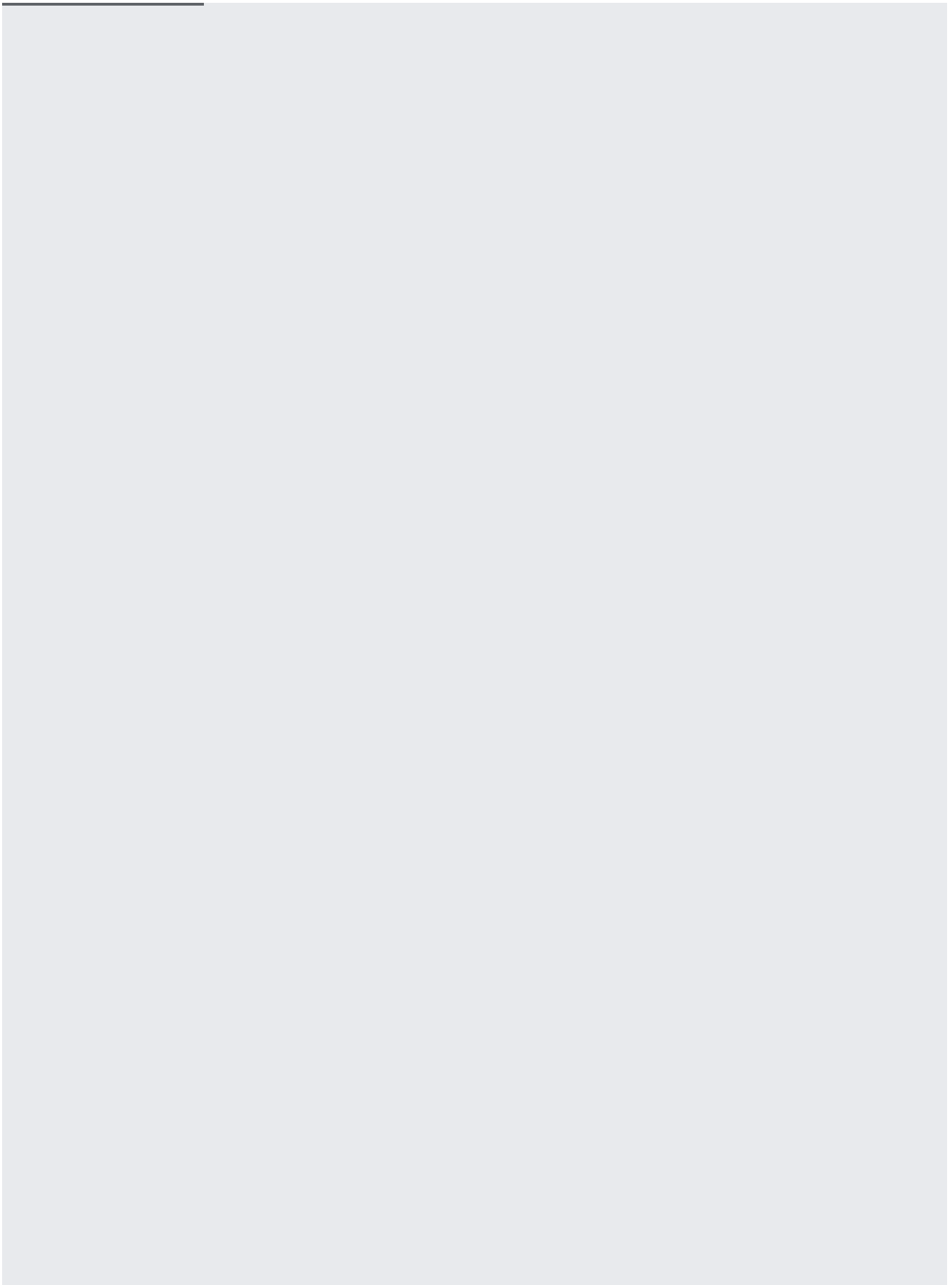


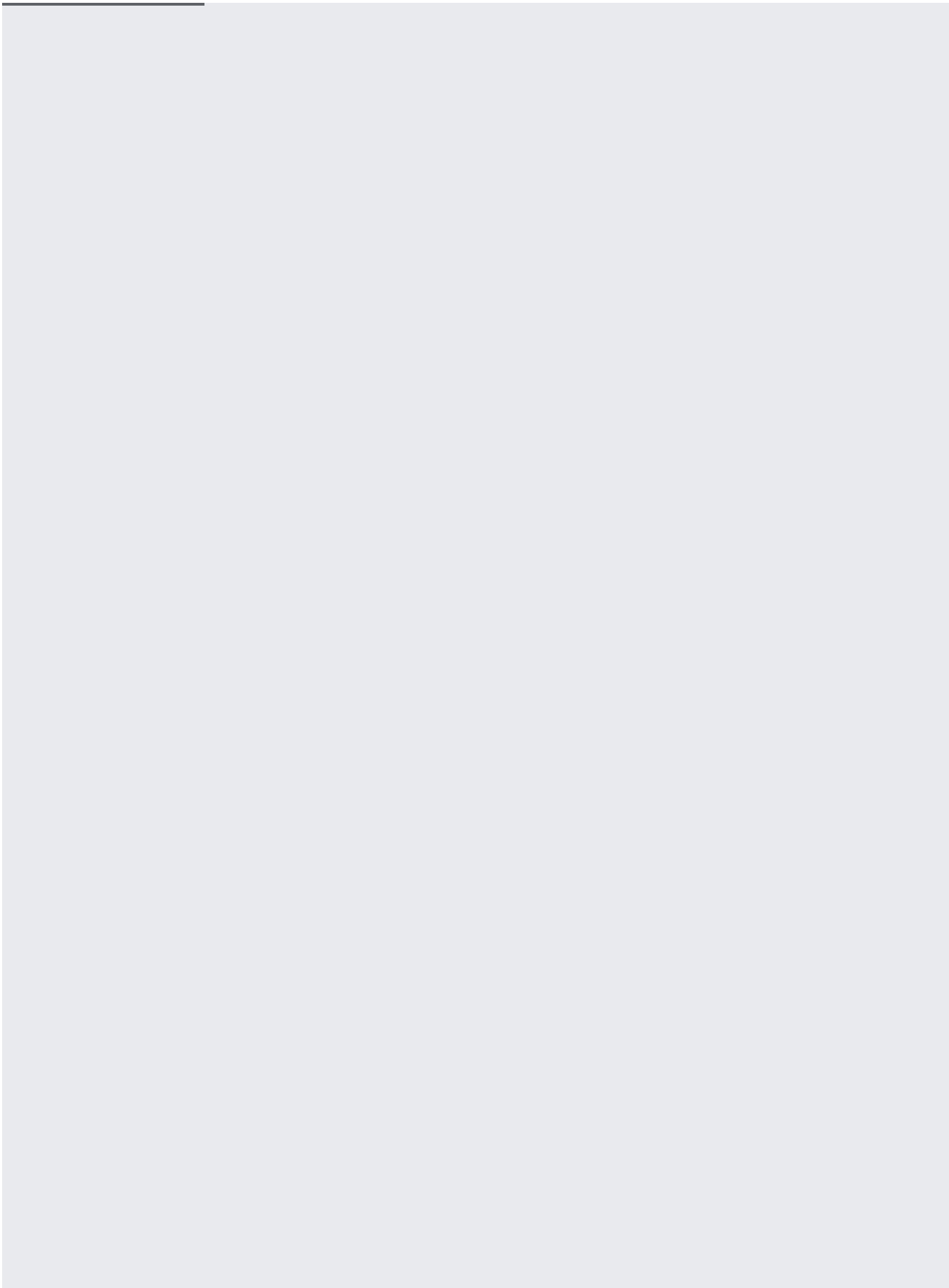




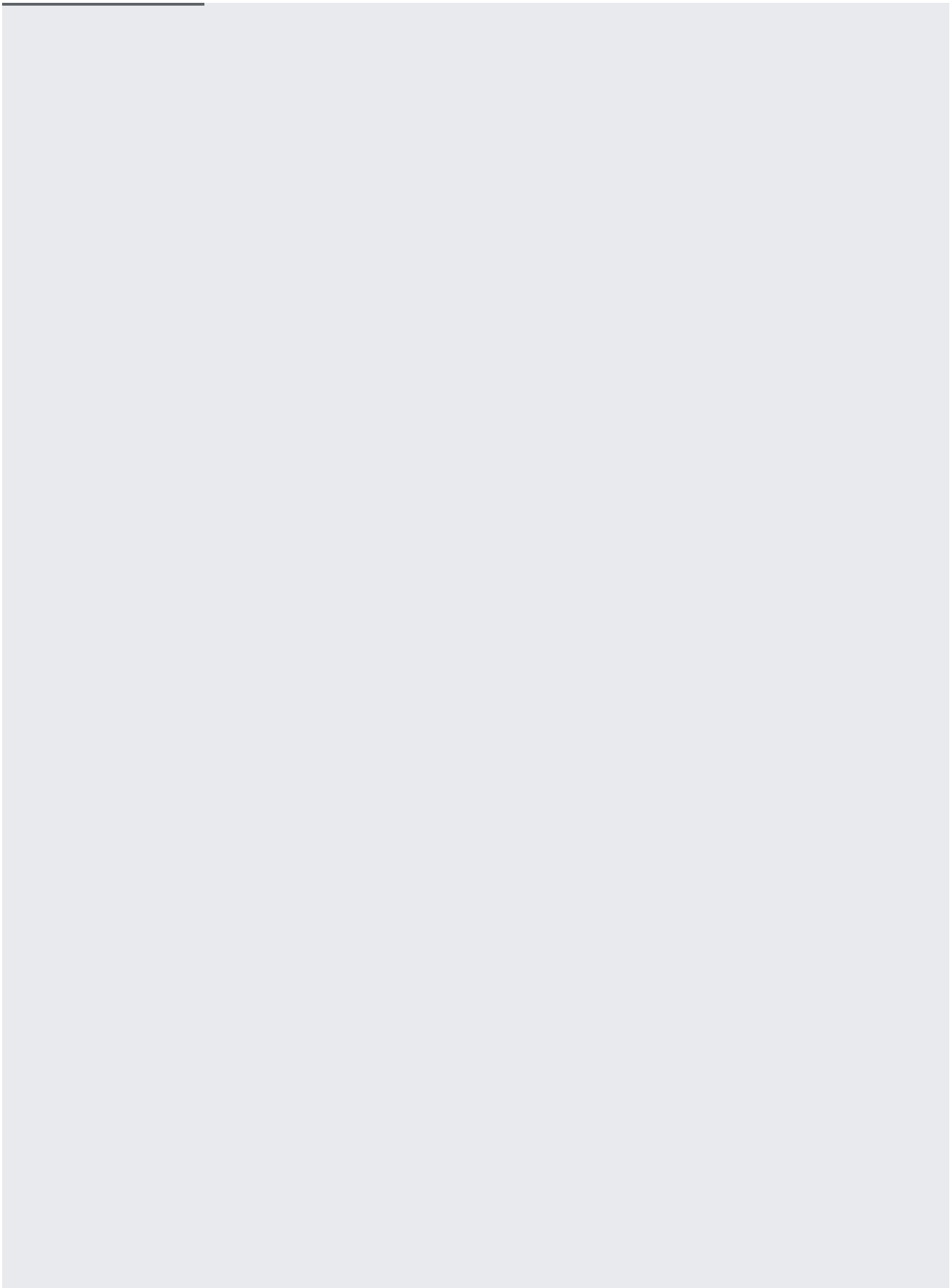


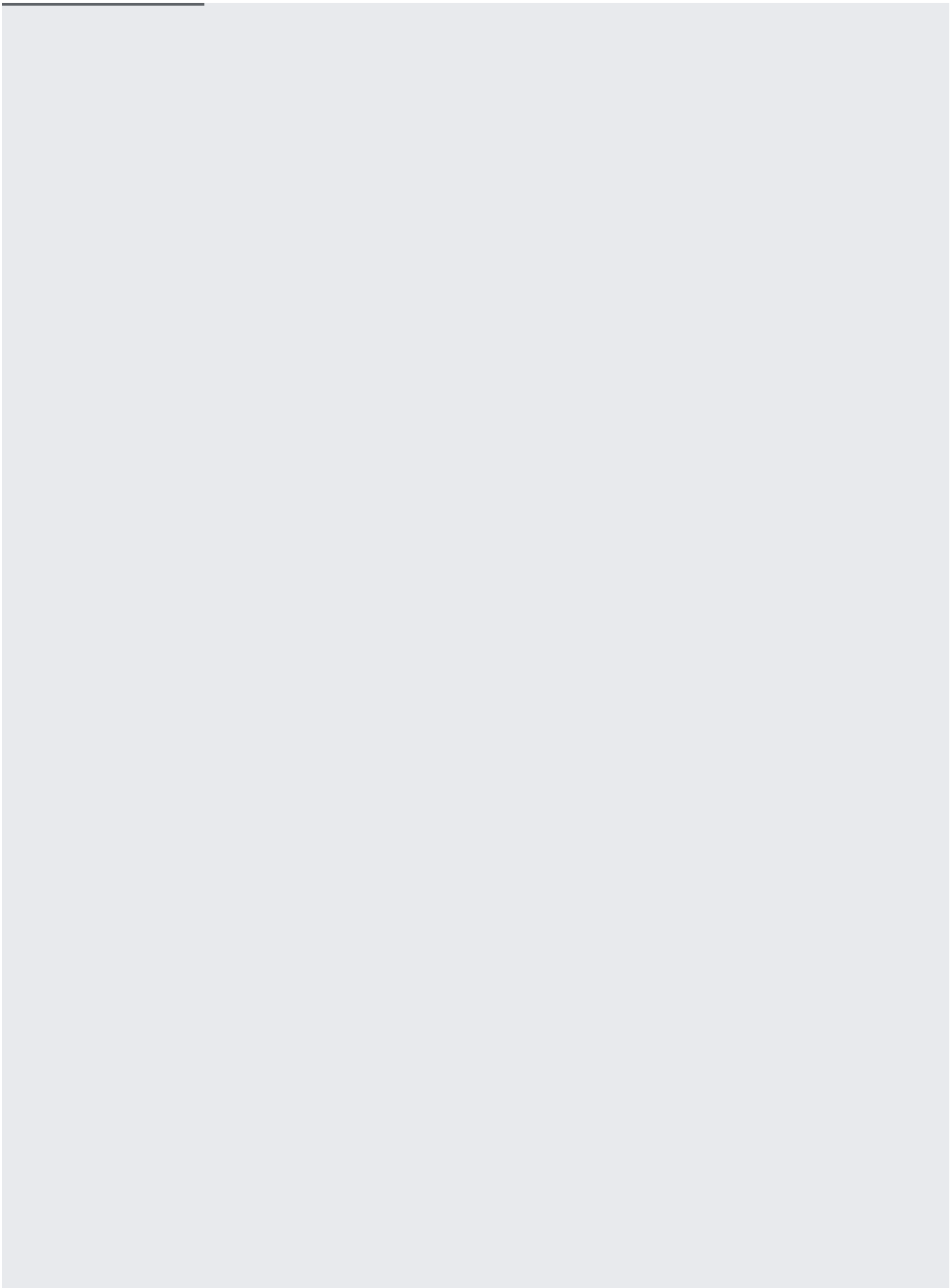


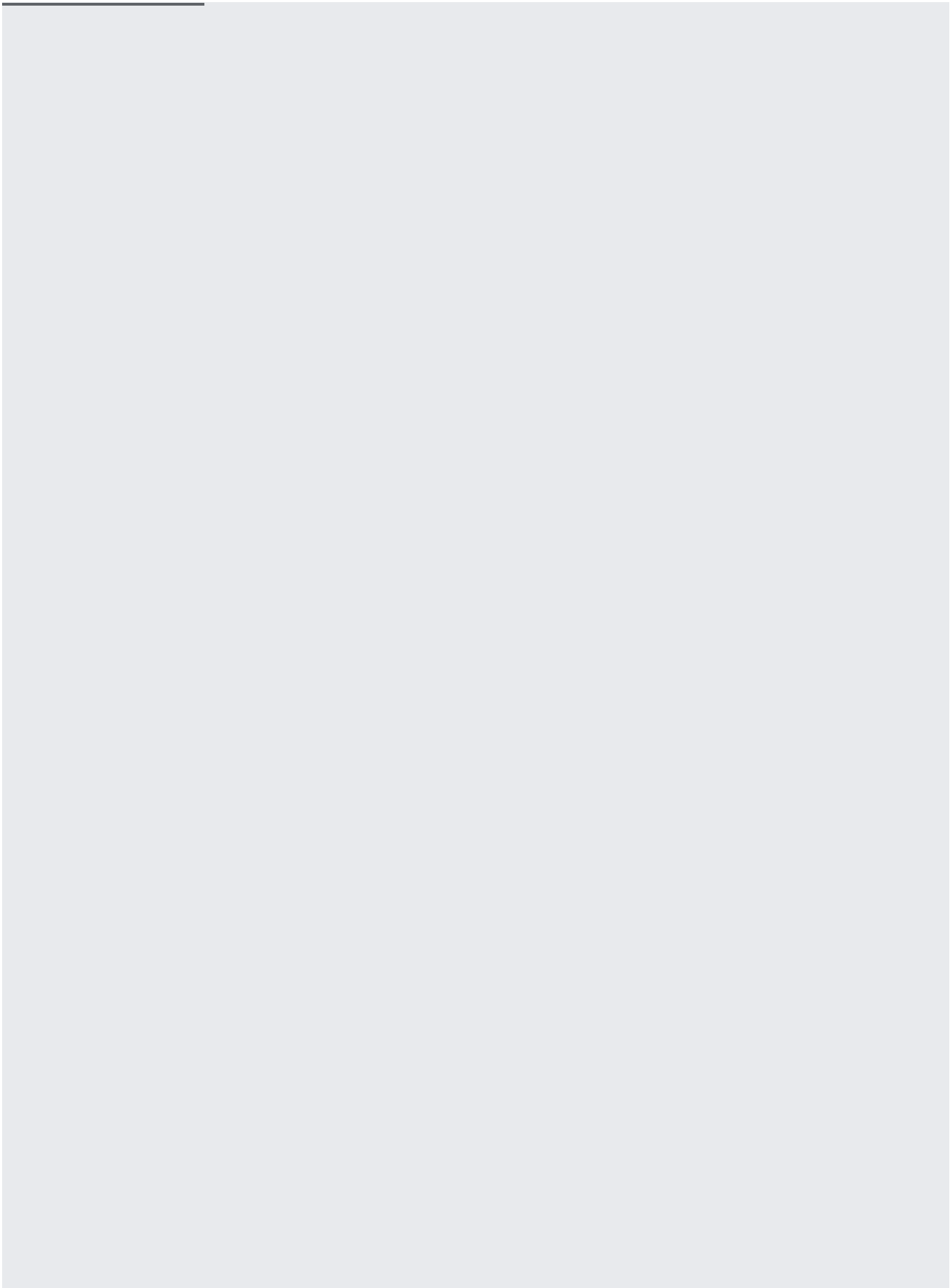


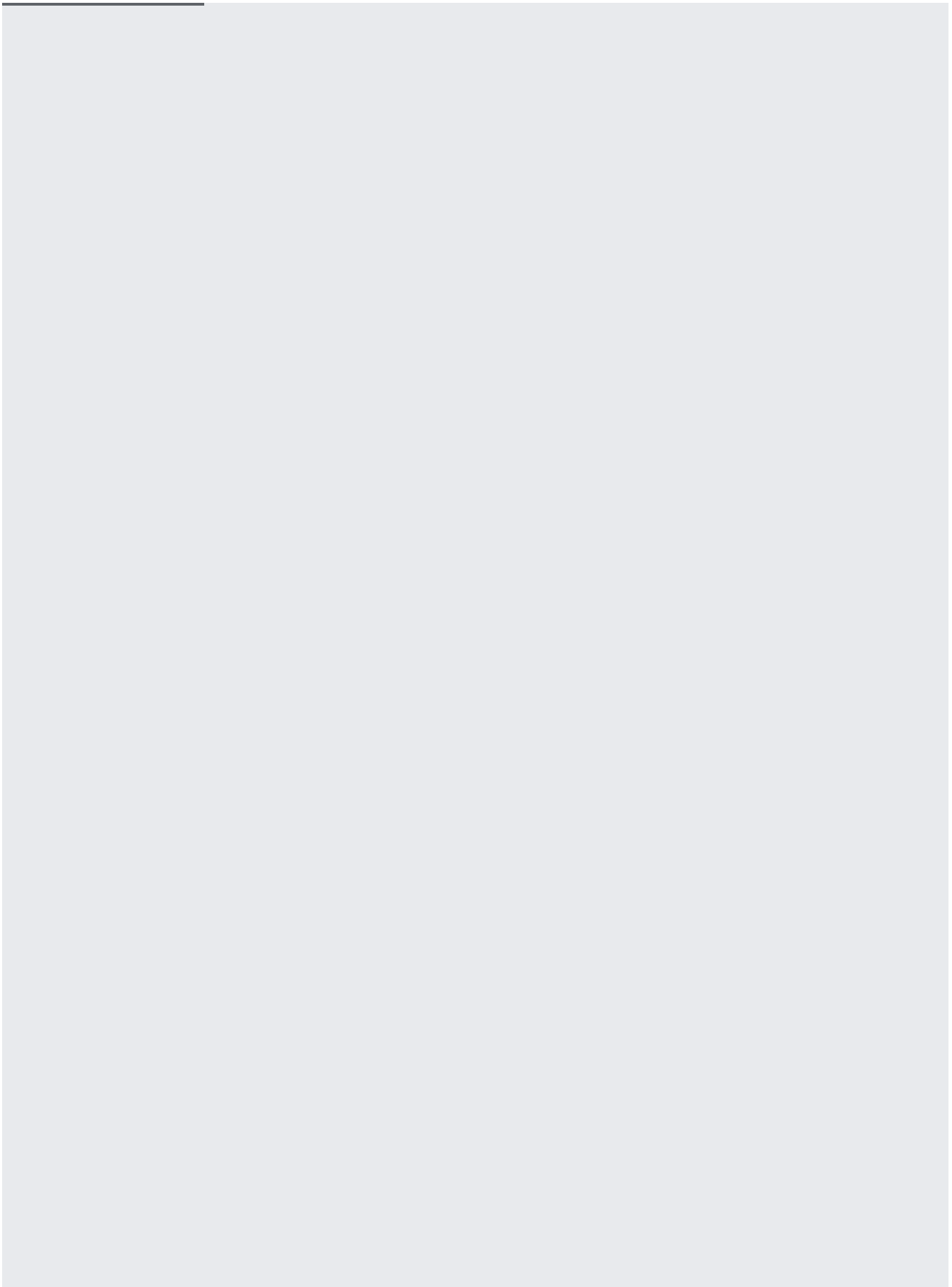


The following shows how to read zero or more rows from a database using an index (/spanner/docs/secondary-indexes).











When reading or querying large amounts of data from Cloud Spanner, it can be useful to divide the query into smaller pieces, or *partitions*, and use multiple machines to fetch the partitions in parallel.

You can perform any read API operation in parallel using the Cloud Spanner client libraries. However, you can only partition SQL queries where the first operator in the query execution plan is **Distributed Union** ([/spanner/docs/query-execution-operators#distributed\\_union](/spanner/docs/query-execution-operators#distributed_union)). To view the query execution plan for a particular SQL query, follow the instructions from [SQL best practices](/spanner/docs/sql-best-practices#how-execute-queries) (</spanner/docs/sql-best-practices#how-execute-queries>).

After you have the query execution plan, make sure that the first operator it contains is **Distributed Union**.

The following samples work with the DDL table definition found in the [Quickstart](#) (https://cloud.google.com/spanner/docs/quickstart-console#short-schema).

