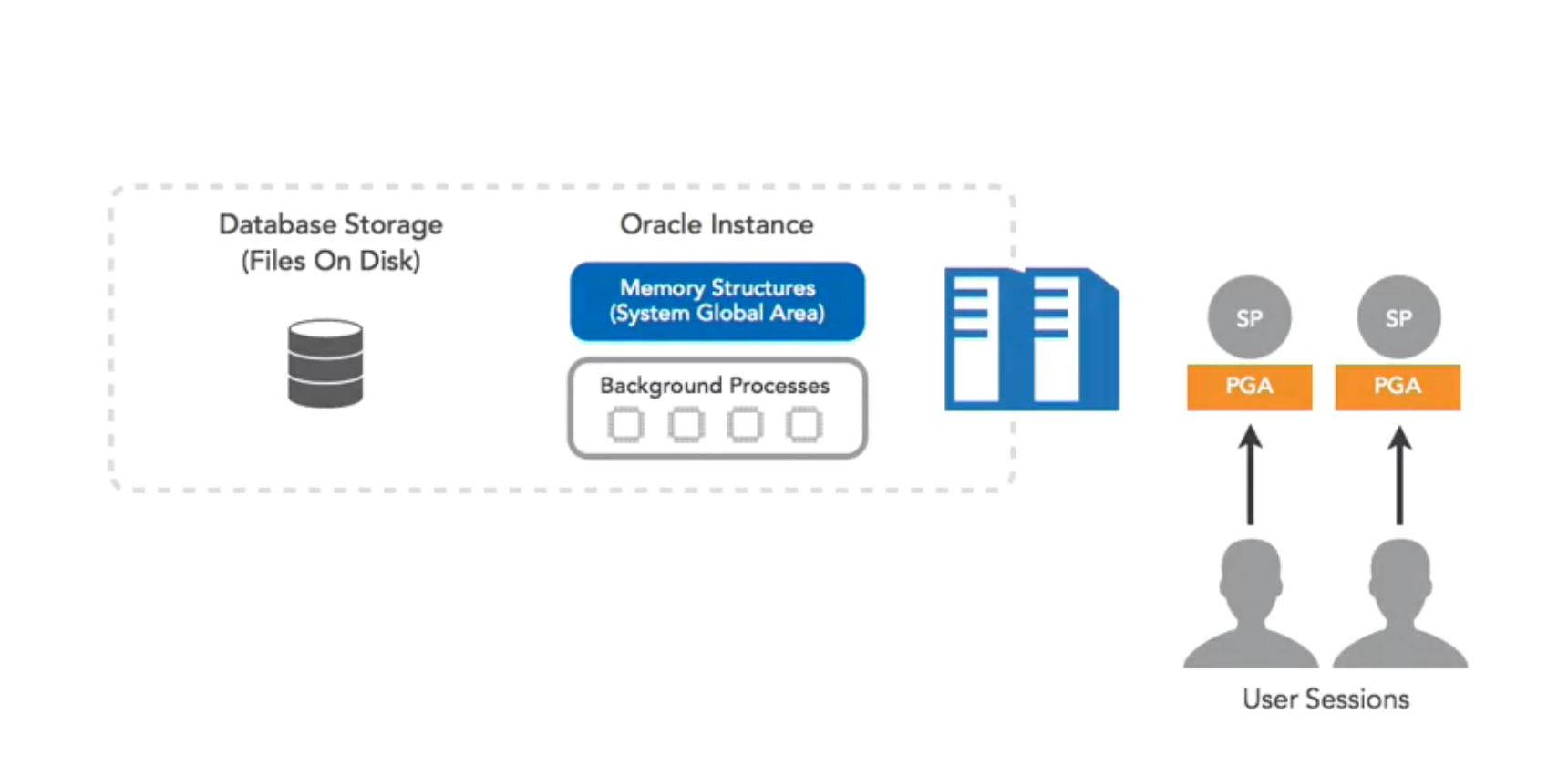
ORACLE DB ARCHITECTURE



DB INSTANCE

The Oracle database architecture contains a large number of individual components working in tandem to allow an Oracle database to be a database, allowing you to read and write data from tables in a way that is safe, speedy, scalable, and secure.

The Oracle database architecture can be divided into three major components.

1. The Oracle instance.
2. The Oracle database storage.
3. The Oracle server processes (SPs).

The Oracle instance, which is basically the Oracle program, or binary, loaded into the server RAM. It exists only in memory, and created by Oracle every time you start up your database (it is non-persistent and disappears each time the database re-starts).

It contains special areas used to cache really important data as well as metadata into memory.

The Oracle instance caches frequently accessed data and metadata into memory. So why do we need to cache data in memory in the first place? Because reading from memory is a lot faster compared to reading from disk. It makes sense to store frequently accessed data in memory so that after the first database session reads the data from disk, which will be slow, it will be cached so that subsequent sessions requesting the same data or metadata will get a more responsive read performance.

The Oracle Instance also includes a set of processes running in the background named the background processes. These background processes work in tandem with the Oracle instance in memory caches to perform various routine database operations. These operations include writing data to disk, ensuring database consistency, and every other duty that an Oracle database has to do on a regular basis to ensure it is consistent, well tuned, operational, and secure.

The Oracle instance alone is not a fully functional database. The non-persistent nature of the Oracle instance would mean that the data cached in the Oracle instance will get vaporized each time we re-start our database therefore it is essential that the data in the database is safe and secure – disk storage.

DB STORAGE

The need to make data consistent and safe is where the second part of the Oracle database architecture comes to prevalence, the Oracle database storage - the storage of database data on-disk.

The Oracle database storage is a special collection of Oracle files used to persist the data from the Oracle instance (which is only in memory) to a physical disk. In order for Oracle to ensure that all of the data contained in the database is safe and secure, Oracle has to store that data on a disk somewhere as a set of files. These disk files are what actually make up the physical Oracle database.

The Oracle instance is only a copy of the Oracle program loaded into memory, not the actual data. A good analogy is Microsoft Word and Microsoft Word documents. Similarly to the way Microsoft Word is a program you run on a computer but the actual data is stored in Word documents on disk, the Oracle instance is a program and the database storage is our data. However, unlike the Microsoft Word analogy, the Oracle database storage is usually placed on a SAN- or NAS-type enterprise-grade, highly available, and highly resilient storage system.

SAN stands for storage area network which is the type of storage accessed by fiber channel infrastructure whereas NAS stands for network attached storage which is the type of storage accessed via a network, usually ethernet. Both types of storage are natively supported by Oracle. Oracle database storage can also be placed on a local disk such as the individual disk inside your computer. Instead of connecting your Oracle server to an enterprise-grade storage system, you can actually tell Oracle to store the database on a local drive that is physically inside the Oracle server.

Traditionally it is not usually up to the DBA or database administrator to handle the actual configuration of the storage system used for the Oracle database. It is the role of storage experts to configure the storage, however, it is up to the DBA to request, or more accurately, demand that the storage experts will allocate sufficiently powerful and highly available storage for the database.

It is imperative that the storage system used is fault tolerant - if the database storage is lost, the data is lost, and the database is lost. Losing the Oracle instance because the database server crashed is not an issue, as, on restart, Oracle will reload it into memory.

SERVER PROCESSES

The last of the three major components, making up the Oracle database architecture are the Oracle SPs, server processes. Their purpose is simple. Oracle starts up server processes to handle the requests of client processes connected to the Oracle database.

A client process, also called a user session that is connected to the database, always communicates through a server process. Server processes are created on behalf of a session that initiated the connection to the Oracle database. These server processes performs one or more of the following tasks.

* The first role of the server processes is to pass and execute SQL statements issued by the connecting session. So when a user issues a select statement, the server process is responsible for making sure the select statement is syntactually correct and find the best way to execute it.
* The second role for the server process is to act on behalf of a client session that needs to read data from disk. The server process is responsible for reading database data from the database storage and loading that data into the Oracle instance buffer cache (more detail later).

The buffer cache region is inside the Oracle instance and is used for caching recently accessed data. Remember, Oracle aims to be a highly responsive database, and one of the best ways to achieve that is to cache frequently accessed data into memory so that we minimize disk rates.

* The third role for the server process is to return the results of an executed SQL statement back to the client.

So to summarize, the major rules and responsibilities of the server processes include one, they act on behalf of client sessions to pass SQL statements, two, they read data from disk and cache it into memory, three, they return the results back to the user.

More users equals more dedicated server processes. In addition to the roles of the server processes we just reviewed, also note that each server process has its own cache dedicated for each connecting client. This cache is also known as the PGA, program global area. The PGA used by each server process is a non-shared memory region reserved only for the specific user session connected to that specific server process.

The PGA contains data and control information used by the server processes when sorting data, joining large tables together as part of a SQL statement, and so on. Essentially, data that should not persist after a user session has finished doing whatever it needed to do. Temporary data. This is in direct contrast to the various other caches used by the Oracle instance itself, which are shared across all connecting sessions.

In most Oracle deployments, you will usually find a one-to-one relation between connecting user sessions and server processes. So for each user that is connected to the database, you will also find a single server process. This is also known as the Oracle dedicated server process model. Can you think of a problem with this approach? Well this approach might work if you have two, 10, or maybe 100 users connected to a database.

But what happens if you have 100,000 users, a million users that need database access? Having the Oracle database start one million server processes for one million user sessions isn't very efficient and can be very resource-demanding from the Oracle server. The solution? What most database professional choose to do is introduce a connection-pooling middleware somewhere between the actual users and the database itself.

An example for such a middleware is having an application server between the actual user sessions and the database server so that our million users will connect to one or more application servers and these application servers will open a fixed set of database connections, thus translating into a fixed set of database server processes. So in our example here, we have a lot of end users connecting to an application server, but that application server is only opening two database connections, and thus the Oracle database only needs to start up two server processes.

This is very efficient as he application server here is basically acting as a connection pool for the users. It's not super-important for you to fully understand how connection-pooling works at this stage, only that you will know that because of the one-to-one user session to Oracle's server process relation, connection-pooling is definitely something you should explore in case the need arises.