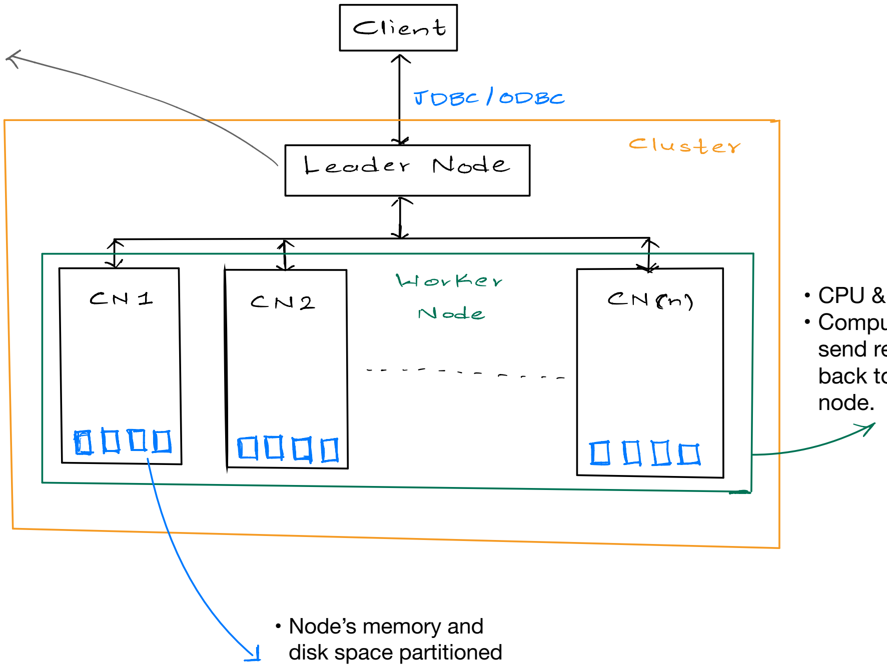
## 1. Build redshift architecture diagram.



- Handles client requests.
- Query optimization.
- Develop execution plan.
- Send compiled code and allocates data to each worker node.

- CPU & Memory.
- Computes and send results back to leader node.

- Node's memory and disk space partitioned as slices.
- Works in parallel to complete operation.
- Number of slice is determined by the node size of the cluster.

## 2. What is columnar data format? What are the advantages?

In a columnar data format the values for the entire column is stored in one or more blocks as opposed to row based data format where each row with multiple column is stored in each block. The advantage of the columnar data format is that it is more efficient in performing large data queries as the data is stored in single column within same block the database does not have to read multiple blocks thus improving I/O performance. Since the entire column is of same data type, columnar data format takes advantage of columnar compression which reduces the amount of data that needs to be read from the disk and save a disk space.

3. **What is encoding? Different type of encoding? A sample create table statement with encoding.**

Encoding is a technique which compresses the data and reduces the amount of data that needs to be read from a disk.

The different types of encoding available in redshift are:

| Encoding type | Keyword | Data types |
|---|---|---|
| Raw - no compression | RAW | All |
| AZ64 | AZ64 | SMALLINT, INTEGER, BIGINT, DECIMAL, DATE, TIMESTAMP, TIMESTAMPTZ |
| Byte dictionary | BYTEDICT | SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ |
| Delta | DELTA<br>DELTA32K | SMALLINT, INT, BIGINT, DATE, TIMESTAMP, DECIMAL<br>INT, BIGINT, DATE, TIMESTAMP, DECIMAL |
| LZO | LZO | SMALLINT, INTEGER, BIGINT, DECIMAL, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ, SUPER |
| Mostlyn | MOSTLY8<br>MOSTLY16<br>MOSTLY32 | SMALLINT, INT, BIGINT, DECIMAL<br>INT, BIGINT, DECIMAL<br>BIGINT, DECIMAL |
| Run-length | RUNLENGTH | SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ |
| Text | TEXT255<br>TEXT32K | VARCHAR only<br>VARCHAR only |
| Zstandard | ZSTD | SMALLINT, INTEGER, BIGINT, DECIMAL, REAL, DOUBLE PRECISION, BOOLEAN, CHAR, VARCHAR, DATE, TIMESTAMP, TIMESTAMPTZ, SUPER |

***Example:***

```
CREATE TABLE customers(
        age int encode delta,
        city varchar(10) encode ZSTD
);
```

**4. What is distribution key? How its is helpful in query performance.**

In redshift cluster the data is stored across multiple compute nodes, the distribution key allows us to define how the data is distributed across the nodes which improves the query performance by reducing the in network data transfer. The distribution can be any column of a data table.