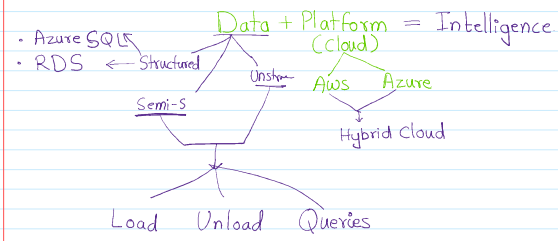
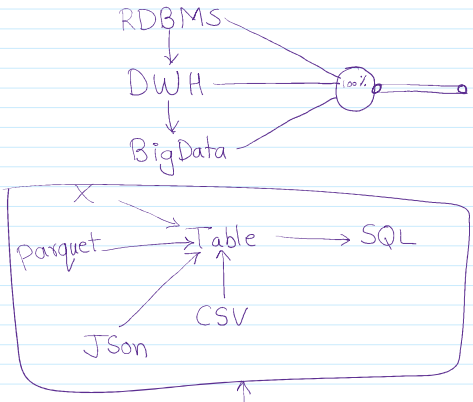
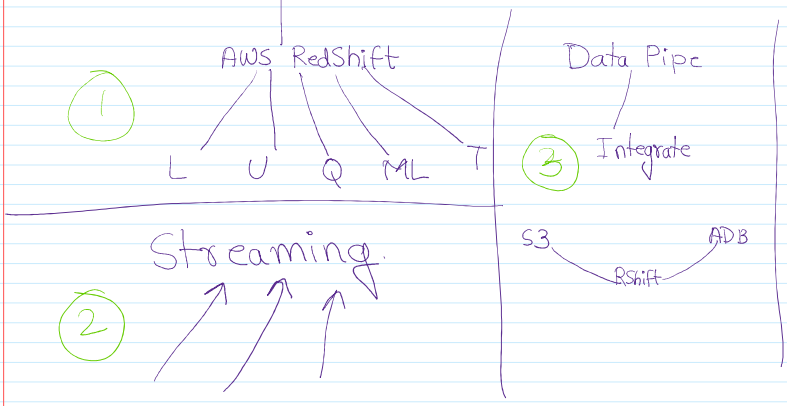
10.03Aug. Enterprise Data

3 August 2023 08:38 PM





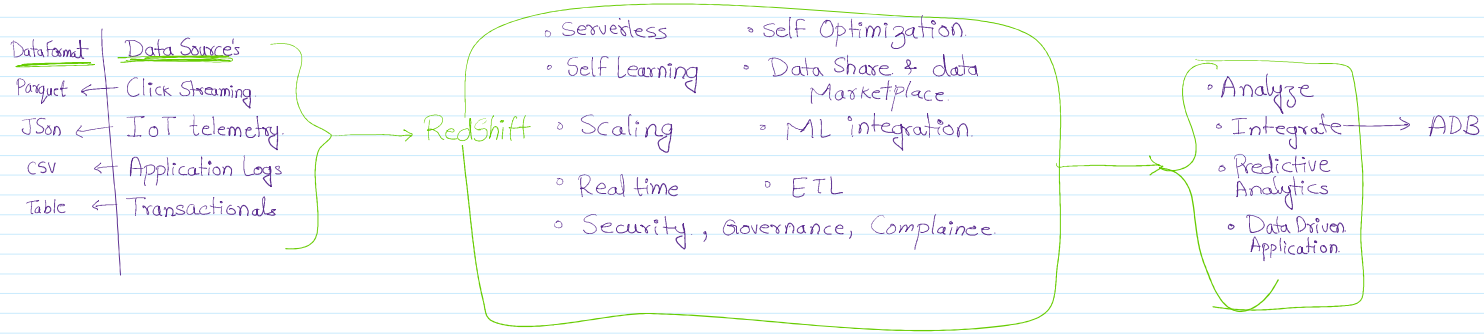


1] AWS RedShift

- Cloud Data Warehouse
- Best in
  - price
  - Performance
  - Reliable

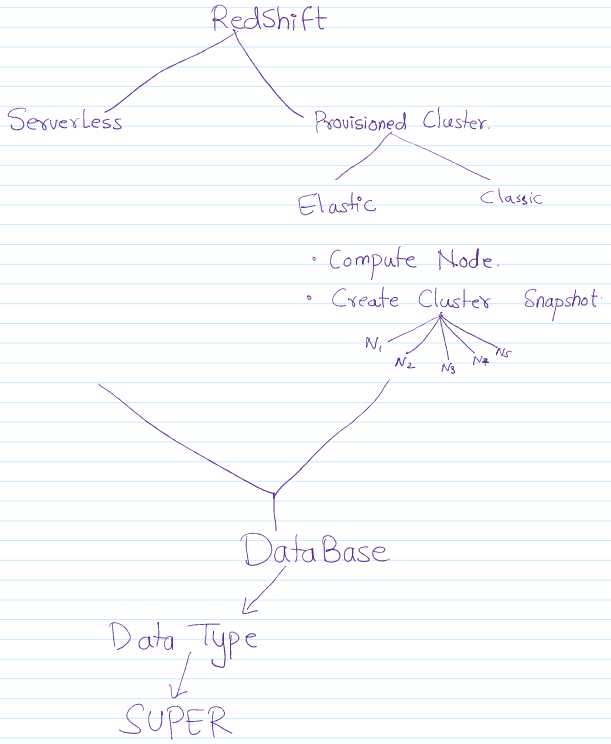
SX

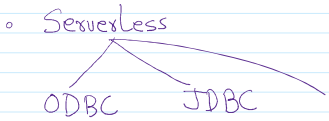
- ServerBase or Serverless



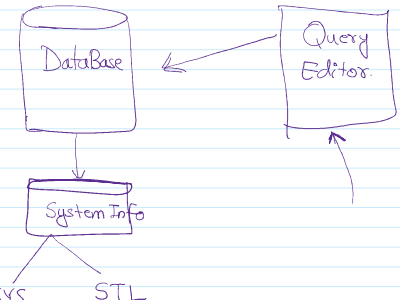
- Using RedShift you can Manage

- Using RedShift you can Manage Exabytes of data.
- Run Complex Analytics Queries.
- Run & Scale in Sec.





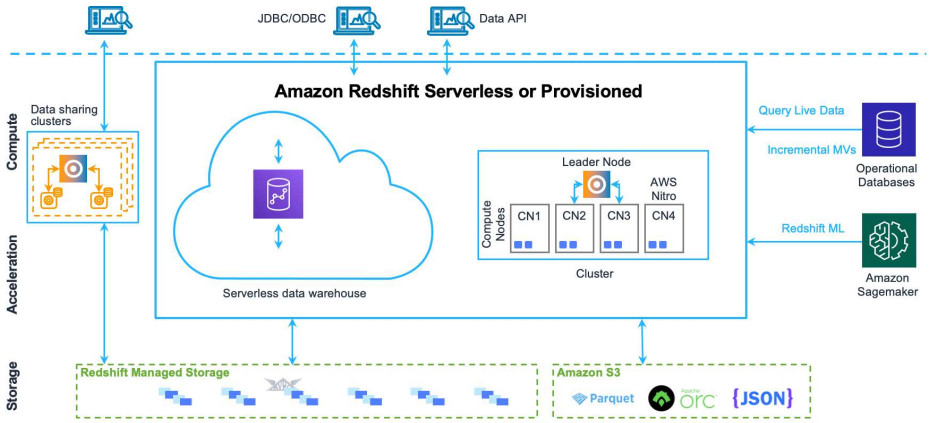
1. Namespace = 
2. Workspace 
3. RPU ⟶ Capacity Unit



System Info

SYS — STL

- SYS_LOAD_HISTORY
- STL_LOAD_ERRORS ⟶ Copy Command Error's



Analyze Redshift Information

```
select  default_iam_role();

select current_user;

select u.* from pg_user u;

select current_schema();  -- public

select * from pg_tables


UserName, SchemaName, UserHasCREATEPermission,UserHasUSAGEPermission


SELECT  * FROM pg_user u


query, starttime, endtime, total_exec_time
```

STL_WLM_Query = Information About Queries executed on the Cluster.

STL_Query = Detailed Query-Level information, include execution times.

STL_alert_event_log = Logs of system alerts & Events.

STL_SCan = Records of tables Scans performed by queries.

STL_Connection_Log = Log of User Conn & disconnections.

Task 1:
Access AWS Redshift Cluster Ensure you have access to an AWS Redshift cluster and necessary permissions to query system tables, or Switch to user where we have full permission.

Task 2:
Using pg_catalog Schema --> Tables

- To get Information about the Clusters Configuration
- Usages
- Executions
- Performance

Task 3:
To retrieve information about the top 10 queries executed on your cluster

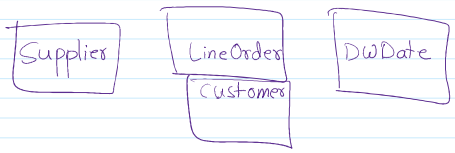
- Query Executed
- Start Time
- End Time
- Total_Exec_time

From STL_Query

Task 4: Analyze Query Performance
Analyze query performance like Identify queries with high execution times

From STL_Query.

Task 5: Monitoring Workload Management(use stl_wlm_query table)



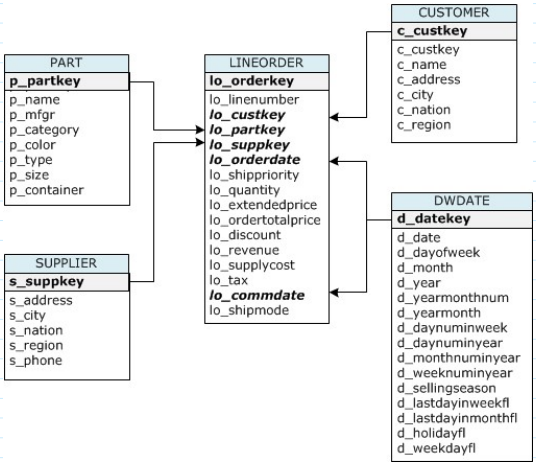

```
drop table part;
drop table supplier;
drop table customer;
drop table dwdate;
drop table lineorder;

CREATE TABLE part
(
 p_partkey     INTEGER NOT NULL,
 p_name        VARCHAR(22) NOT NULL,
 p_mfgr        VARCHAR(6),
 p_category    VARCHAR(7) NOT NULL,
 p_brand1      VARCHAR(9) NOT NULL,
 p_color       VARCHAR(11) NOT NULL,
 p_type        VARCHAR(25) NOT NULL,
 p_size        INTEGER NOT NULL,
 p_container   VARCHAR(10) NOT NULL
);

CREATE TABLE supplier
(
 s_suppkey   INTEGER NOT NULL,
 s_name      VARCHAR(25) NOT NULL,
 s_address   VARCHAR(25) NOT NULL,
 s_city      VARCHAR(10) NOT NULL,
 s_nation    VARCHAR(15) NOT NULL,
 s_region    VARCHAR(12) NOT NULL,
 s_phone     VARCHAR(15) NOT NULL
);

CREATE TABLE customer
(
 c_custkey     INTEGER NOT NULL,
 c_name        VARCHAR(25) NOT NULL,
 c_address     VARCHAR(25) NOT NULL,
 c_city        VARCHAR(10) NOT NULL,
 c_nation      VARCHAR(15) NOT NULL,
 c_region      VARCHAR(12) NOT NULL,
 c_phone       VARCHAR(15) NOT NULL,
 c_mktsegment  VARCHAR(10) NOT NULL
);

CREATE TABLE dwdate
(
 d_datekey          INTEGER NOT NULL,
 d_date             VARCHAR(19) NOT NULL,
 d_dayofweek        VARCHAR(10) NOT NULL,
 d_month            VARCHAR(10) NOT NULL,
 d_year             INTEGER NOT NULL,
 d_yearmonthnum     INTEGER NOT NULL,
 d_yearmonth        VARCHAR(8) NOT NULL,
 d_daynuminweek     INTEGER NOT NULL,
 d_daynuminmonth    INTEGER NOT NULL,
 d_daynuminyear     INTEGER NOT NULL,
 d_monthnuminyear   INTEGER NOT NULL,
 d_weeknuminyear    INTEGER NOT NULL,
 d_sellingseason    VARCHAR(13) NOT NULL,
 d_lastdayinweekfl  VARCHAR(1) NOT NULL,
 d_lastdayinmonthfl VARCHAR(1) NOT NULL,
 d_holidayfl        VARCHAR(1) NOT NULL,
 d_weekdayfl        VARCHAR(1) NOT NULL
);
CREATE TABLE lineorder
(
 lo_orderkey        INTEGER NOT NULL,
 lo_linenumber      INTEGER NOT NULL,
 lo_custkey         INTEGER NOT NULL,
 lo_partkey         INTEGER NOT NULL,
 lo_suppkey         INTEGER NOT NULL,
 lo_orderdate       INTEGER NOT NULL,
 lo_orderpriority   VARCHAR(15) NOT NULL,
 lo_shippriority    VARCHAR(1) NOT NULL,
 lo_quantity        INTEGER NOT NULL,
 lo_extendedprice   INTEGER NOT NULL,
 lo_ordertotalprice INTEGER NOT NULL,
 lo_discount        INTEGER NOT NULL,
```



```
    lo_revenue         INTEGER NOT NULL,
    lo_supplycost      INTEGER NOT NULL,
    lo_tax             INTEGER NOT NULL,
    lo_commitdate      INTEGER NOT NULL,
    lo_shipmode        VARCHAR(10) NOT NULL
   );
```

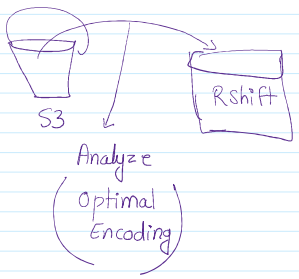
```
aws s3 ls s3://awssampledbuswest2/ssbgz/supplier.tbl

COPY dev.public.supplier_mujahed FROM 's3://awssampledbuswest2/ssbgz/supplier.tbl'
CREDENTIALS 'aws_iam_role=arn:aws:iam::475184346033:role/service-role/AmazonRedshift-CommandsAccessRole-20230802T180608'
DELIMITER '|'
gzip
REGION 'us-west-2';

CREATE TABLE customer
(
  c_custkey     INTEGER NOT NULL,
  c_name        VARCHAR(25) NOT NULL,
  c_address     VARCHAR(25) NOT NULL,
  c_city        VARCHAR(10) NOT NULL,
  c_nation      VARCHAR(15) NOT NULL,
  c_region      VARCHAR(12) NOT NULL,
  c_phone       VARCHAR(15) NOT NULL,
  c_mktsegment  VARCHAR(10) NOT NULL
);


COPY dev.public.customer_mujahed FROM 's3://bkt-03aug-mujahed/load/customer-fw-manifest'
CREDENTIALS 'aws_iam_role=arn:aws:iam::475184346033:role/service-role/AmazonRedshift-CommandsAccessRole-20230802T180608'
fixedwidth 'c_custkey:10,c_name:25,c_address:25,c_city:10,c_nation:15,c_region:12,c_phone:15,c_mktsegment:10'
maxerror 10
acceptinvchars as '^'
manifest;
```

CompUpdate OFF



- To Clean Up
- Ready to Use

VACCUM = Recover the space from deleted rows & restores the sort order.

Analyze = update statistics of metadata
- Enable the Query Optimizer
- Generate More accurate Query Plans.

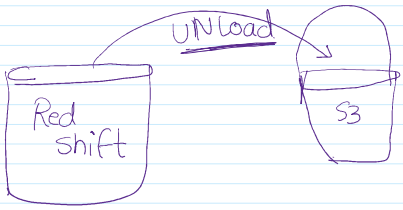
Data Loading
- CSV
- Character-Delimited
- Fixed-width

- STL_LOAD_ERRORS
- Region • MaxError • AcceptInv.Chars
- DateFormat

→ Best Practice

- Date format.

→ Best Practice.
- COPY → Loading data.
- Loading data files
- Single → Multiple
  ↓
  Manifest
- Compressing.
- Verify & Vacuum & Analyze.



unload (DataSet)
to 'S3_URL'
iam_role