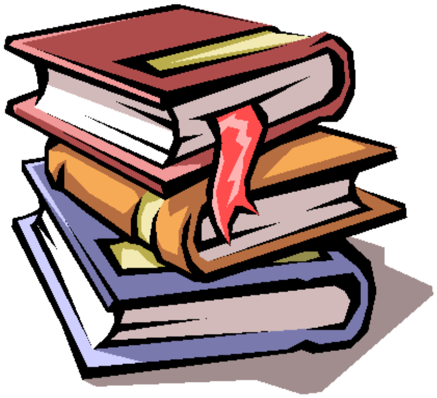
optymyze

Optymyze Support



Reference Book

(Version 1)

Documented By:

Rahul Arora

Table of Contents

Quick Links

This section contains all the links for easy navigation so that you can directly look into particular Topic:

Support Action Items	Optymyze Architecture
Farm Concept	Multi-byte vs single Byte DB
DB Schemas	Optymyze Back up
Optymyze Services	Service Desk
How to start/stop service desk	Feedback Tool details
Gaming Tool Details	Reporting Tool
Linux Commands	How to set Linux File permissions
start/stop Linux services	Oracle Select Commands
Oracle Joins	Oracle Fragmentation
DB Indexing	

Course Overview

The Objective of this book is to assist Optymyze support team members in improving the quality of services in terms of:

a) Understanding the issue with better approach
b) Helping to figure out the cause quicker
c) Giving fast workaround or solution to our Clients
d) Bringing more visibility among teammates in transitioning the knowledge

Use this book as for reference purpose and this will assist you along with your valuable understanding to explore more ways for investigating the ongoing problem.

The Course tries to cover all the aspects related to the Product and technical Operational background so that we can lay the foundation to align individual member of the team as united group. It also helps you to go through basics of the technologies used in the Optymyze platform on the backend to put everyone on right direction to handle the product wisely.

Estimated time to go through this book would be around 3-4 weeks depending upon your capabilities. Some of the Chapters are filled with assessment to test your knowledge and evaluate you as productive member of the team.

You need access to the Service Desk, Support Optymyze Portal, Optymyze Cloud Manager, Support Application Server and Database as recommended tools for the explorations & practices

Most of other additional tools are present on https://rdc.synygy.com and can be used directly from this location if required across the coursework. However, any free tool can also be downloaded from the internet as well.

Apart from this, AWS console is accessible using Multi factor authentication and the activation process for 2FA is listed in the Article 1.

Community Portal & Confluence are two other knowledge base systems that you can use for your learning purpose.

In case of anymore concerns, you can reach out to the senior member of the support team and for queries related to the exercises, you can contact **Rahul Arora**

*****Start of Module 1****

Chapter 1 – Product Optymyze

Domain Introduction:

Optymyze, as product helps in improving sales performance across the entire organization—from sales and sales operations to information technology, talent management, and finance. It acts as a calculating engine on the data derived from hundreds of sources, combined with artificial intelligence, to constantly learn about the drivers and predictors of performance.

***Technological aspect*:**

Optymyze is scalable platform comprises of numerous different applications owned and run by R&D department and target all the domains in the market starting from pharmaceutical companies, financial market, banking or retail sector, Airlines etc. and helps to transform their data into actionable insights that increase revenue and lower cost of sales for our clients. This is *Optymyze SPM Cloud*.

Other than this, we also have our own *Cloud Deployment services* to continuously monitor performance, dynamically tune computing resources, and regularly update the applications for all customers. This includes deployment on public clouds (AWS), hybrid clouds and customer's own data center

More information about the product, our clients, and Customer success stories can be read through website: www.optymyze.com

Support Action Items:

Support team works in diversify roles handling product throughout from the feature assistance to the technical nature of the issues raised from Client or Professional services.

We act as front gate for technical department of an Optymyze product serving 24/7 customer support in terms of dealing any kind of issues in our platform.

A Major Key area of our goal lies with L2 investigation process which comprises of:

a) Understanding the nature of problem and reproduce the scenario
b) Assist in resolving underlying infrastructure issues
c) Troubleshoot the performance issues with Optymyze
d) Handling Data/ configuration issues (This requires knowledge of product feature)
e) Triage third Party tool problems which include Logi, Excel, web services tools etc.
f) Batch Processing related issues

Apart from L2 support, if the issue is found to be defect in the Optymyze, Support team raise this issue to Development team and engage ourselves in providing continuous updates to the Client with resolution/workaround.

Considering that support has to work in the shifts, there is process established for follow-up so that we can maintain the decorum for our continuous service to our customers. Everyone in the team ensures to provide proper hands-off at the end of his/her shift.

Support ensures that a correct, complete & easily understandable root cause analysis is shared with Clients to gain their confidence in your product capabilities.

Few Basic rules to follow:

- It is mandatory that any reported issue should be tracked through service desk for tracking and re-usable purpose.
- Always strive to resolve the issue at your level and if needs to be escalated, we should follow proper path
- Priority of the issue is always mentioned by Customer however severity could be adjusted based on business impact. Try to evaluate while giving preferences in your queue.
- Follow to document the steps of reproducing the issue along with all relevant information in the ticket including screenshots
- Educate your client to keep every different issue logged in separate ticket assuring the correct resolution of the problem

For all the above investigation process, Support should have access to the required tools. Most of them are available on your remote desktop connected to Chester; however few of them are configured with Multi-factor authentication. One major example is AWS console. And here is the process to configure token system for your ID.

Console Access:

a) Browse to self service console: https://rsa.optymyze.com
b) Use your company username and select authentication method as password to login through your company credentials
c) Once you log in for the first time you should be displayed with a warning that you have not created security questions. Set them up and once you are done with security question, request for token as of type: Software
d) Pick the applicable device that you will use for the token and create the PIN that you will use to retrieve your pass code. Recommended to use Smartphone for the token
e) Once the request is approved, you need to activate your token from self service console
f) You will get a QR code that you can scan to import your token. To scan this QR code you must have installed RSA app on your phone. Once you scan it, token will import and you can use it for MFA applications.

Chapter 2 – Infrastructure

Datacenter

The Optymyze platform is deployed in several datacenters and it supports several Deployment models.

Currently, initiative is to bring everything in AWS VPC which is distributed in 4 different regions.

a) Virginia
b) Ireland
c) Frankfurt
d) Canada

We can either do minimal deployment for on-premises clients OR entire stack deployment in our own VPC. Methodologies are categorized due to offerings in different categories for Optymyze application to run.

In AWS, we use following key services to manage our Systems:

- **EC2**: Compute Services
- **IAM**: Identity services, federation using SAML2, roles for servers
- **EBS**: provides block storage for VM's.
- **S3**: provides object storage for backups and other files which require H/A.
- **VPC**: AWS network isolation for creating a logical bubble that isolates us from other Amazon clients.
- **KMS**: AWS encryption service which provides encryption keys used to encrypt volumes as well as S3 buckets
- **ELB**: Elastic Load Balancer which provides a load balancer that exposes all Optymyze projects to the internet
- **CloudWatch**: AWS Monitoring services which provides metrics for servers and other AWS services that we use
- **AWS Route53**: Provides DNS services for the AWS infrastructure
- **SES**: AWS Simple email service which provides email services required by Optymyze. All emails send from a datacenter go though SES while sending notifications to the client.
- **CloudTrail**: Audit logging services, that records all actions performed in AWS.
- **AWS Lambda**: AWS Service which allows you to run arbitrary code based on events. Events are written to S3 bucket, then it writes to a SNS queue and so on.
- **SQS** : A topic based Queue service which is used to send events between different Infrastructure components

This is not the entire list of services we use; however it provides the most important ones currently in use.

Optymyze Architecture

Farm

A unit of deployment in an Optymyze cloud. Multiple Farms can be deployed in one Data center and all of them are managed by OCM.

All the servers are organized into farms to allow isolation in form of security, availability and maintainability where we can attach different environments (the one which contain client Specific information) in different farms to protect their content.

Security: clients can be isolated in individual farms if required by contract
Availability: farms fail independently since they do not share any resource
Maintainability: farms can be upgraded independently of each other

Farm level components:

Web tier

Optymyze stack consists of two web applications: a Login Server and the main Optymyze app. For small clients the web application components consists of a single instance, but for large deployments it can grow to accommodate more users and can have multiple instances as part of single farm

Job tier

Optymyze processes data in batches (called processes or jobs) using a cluster of workers managed by a dispatcher. The dispatcher's role is to receive run requests and to dispatch them to free workers. It also manages run scheduling and monitors all workers

Reporting

Optymyze uses Logi Analytic's applications to generate reports and offer analytics support.

Storage/Data Tier

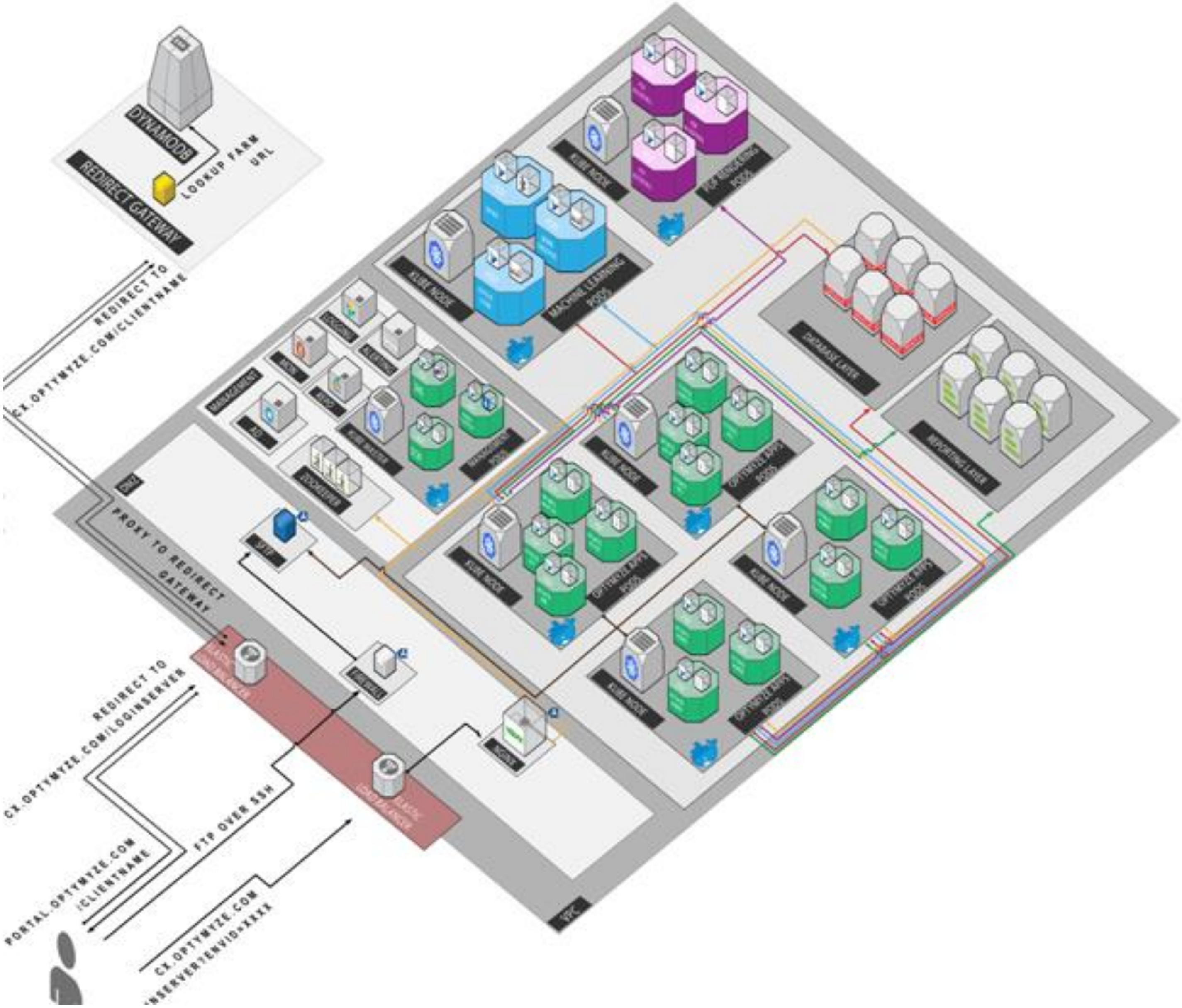
Optymyze applications use two kinds of storage: files on File-system and Relational databases.

Each client environment has its own file-system directory called "project root" and several Oracle Database schemas. Documents like static reports and images are stored in the project root. Everything else is saved in the database.

Apart from the above, we also have **Datacenter level components**:

- OCM
- Zookeeper cluster

- PDF Rendering Cluster
- Kafka Broker & Topics
- Nginx
- Repositories (Docker images, RPMS and Optymyze zipped artefacts)
- Puppet
- SFTP Server
- Alerting and Metrics
- Kubernetes Cluster (where the Optymyze applications are deployed)



Farm Architecture

As we mentioned earlier about the farm components, it is mainly made of 4 applications:

- Worker Application
- Dispatcher Application
- Optymyze Web Application
- Login Server Application

Inside the farm we have client specific metadata, another logical component known as Environment and can be classified in below components

- Environment Root - a folder on the File-system which stores environment files of an application
- Environment Temp – a folder on the File-system where the applications store temporary Files)
- Environment Oracle Schema's - (Database schema's that store the environment tables, Views and other database objects belonging to the environment)

Optymyze farm components are packaged as Docker containers and are deployed on Kubernetes as pods.

Some important Conceptual notes about Farm:

1) Optymyze applications use framework "**Spring Boot**" to create self contained java application with embedded tomcat servlet container to provide web server functionality.
2) We have one Dispatcher ,Login server application per Farm but can have multiple instance of other applications
3) We have pod for Optymyze, Worker and Dispatcher application containers including fluentd as logging container as well as another separate pod for login server application placed in Optymyze suite.
4) Few helper tools are packaged together like farmount and promotheus java agent with our applications
5) Optymyze applications running inside containers in the pods talk with zookeeper node to discover their farm and environments on that farm.
6) Once farm is discovered, they publish the data in zookeeper to let OCM knows for their status

Optymyze Relational Database – Oracle

The database server is a core component of the Optymyze application and we run Oracle 12c Database over oracle Linux 7.x.

Optymyze uses a set of schemas to provide separation at the application level. We support two different Categories of databases.

- Single-Byte Database
- Multi-Byte Database

Selection of the Database is done by the client while they sign the contract based on their requirements

For Clients that have Latin characters and few European languages which use only 1 byte (8 bit of characters) to store the data in Database are recommended to choose Single-Byte Database.

If the client has data coming from international markets and have Mandarin/Arabian character to store in Optymyze side, in that case they need to support the additional bytes required for Unicode which can be done through multi-byte database deployment.

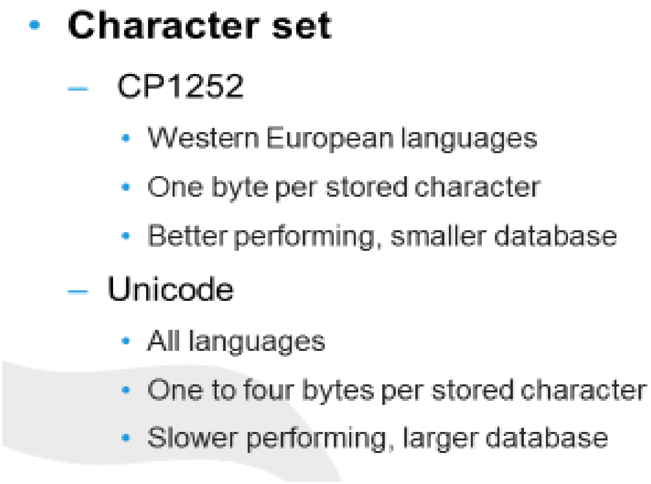
Limitations of Single byte: Any Arab/Chinese/other Non-Latin characters will not be interpreted by the database and it will replace them with "?" which will corrupt the data

Downside of Multi byte: Performance will be the cost to translate larger blocks of the character.

So, when we create database, correct character set is required to be chosen.

We can identify the character set of the database in OCM under SPM Properties tab and try filtering "DB_CHARSET"

AL32UTF8: Multi-bytes, WE8MSWIN1252: Single byte



- **Character set**
  - CP1252
    - Western European languages
    - One byte per stored character
    - Better performing, smaller database
  - Unicode
    - All languages
    - One to four bytes per stored character
    - Slower performing, larger database

All the Database servers from AWS using automatic storage Management Disks so that we don’t need to directly manage those database files. ASM groups the disks in your storage into one or more disk groups and we just need to manage those disk group to place Database files.

ASM Disk groups: u01 (Contain Oracle Binaries), DATA (Oracle data stores here), TEMP (Oracle temporary files), TECH (Oracle undo & System files)

Schemas: It is a collection of logical structures of data, or schema objects. A schema is owned by a database user and has the same name as that user. Each user owns a single schema.

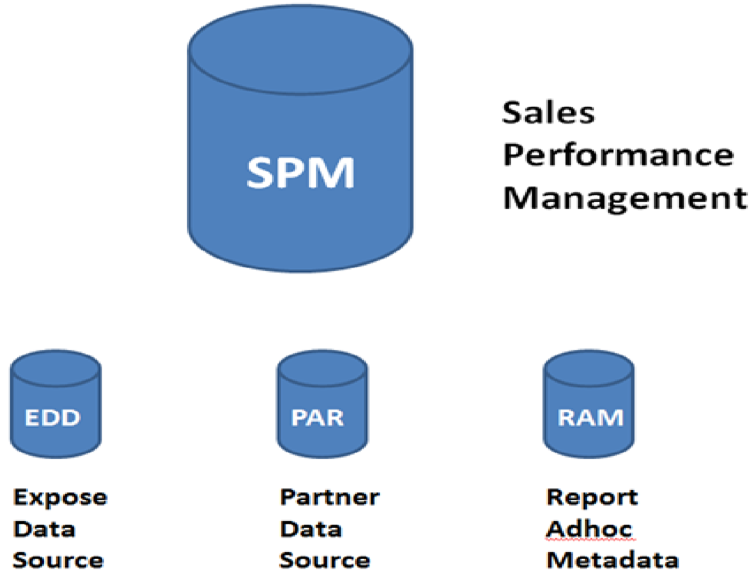
Optymyze uses an array of schemas that provide different functionality and to attain different goals

Any Optymyze project is based on six Oracle schemas:

- Sales Performance Compensation (SPM) schema containing most of the code and objects
- Expose Data Source (EDD) schema containing a subset of data and used for exposing that data to some external tools
- Partner Data Source (PAR) schema containing a subset of data and used for testing reports
- Reports Data Source (RDS) schema containing a subset of data and used for standard reports
- Report Ad-hoc Metadata (RAM) schema containing a subset of data and used for ad-hoc reports
- **Sales Planning (PLA) schema** containing sales planning application as part of Optymyze Suites

First five schemas must be installed on the same database server however; **PLA is exception** which is deployed on separate server.

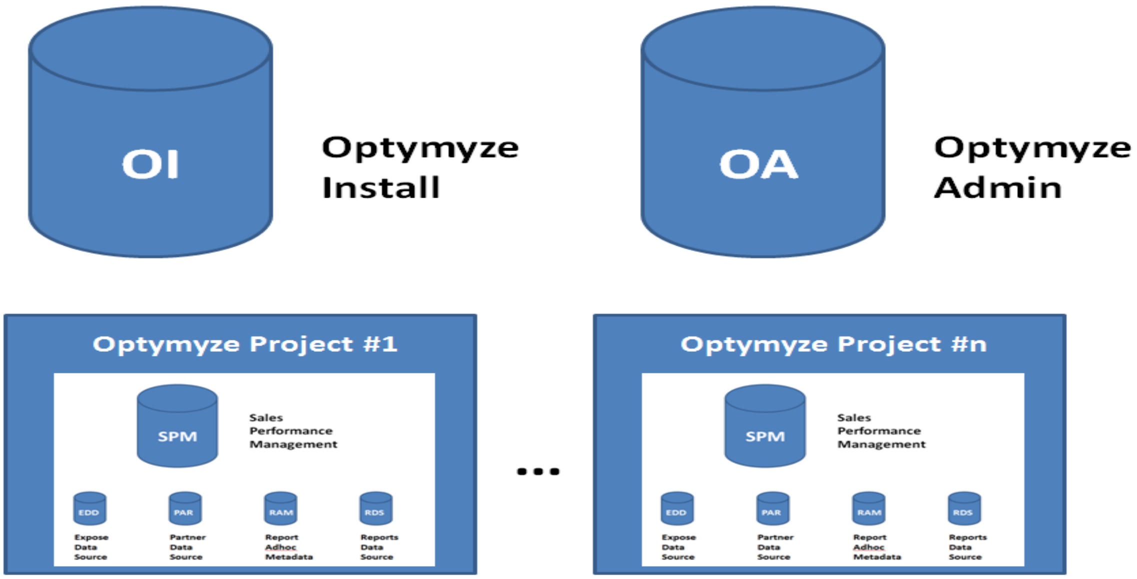
Optymyze DB server can be used for multiple projects and we can see few examples on sales projects



Regardless of the number of Optymyze projects, two more schemas must be configured:

Optymyze Install (OI) schema, a tailor made schema with the needed privileges to install and configure the 5 schemas needed for an Optymyze project

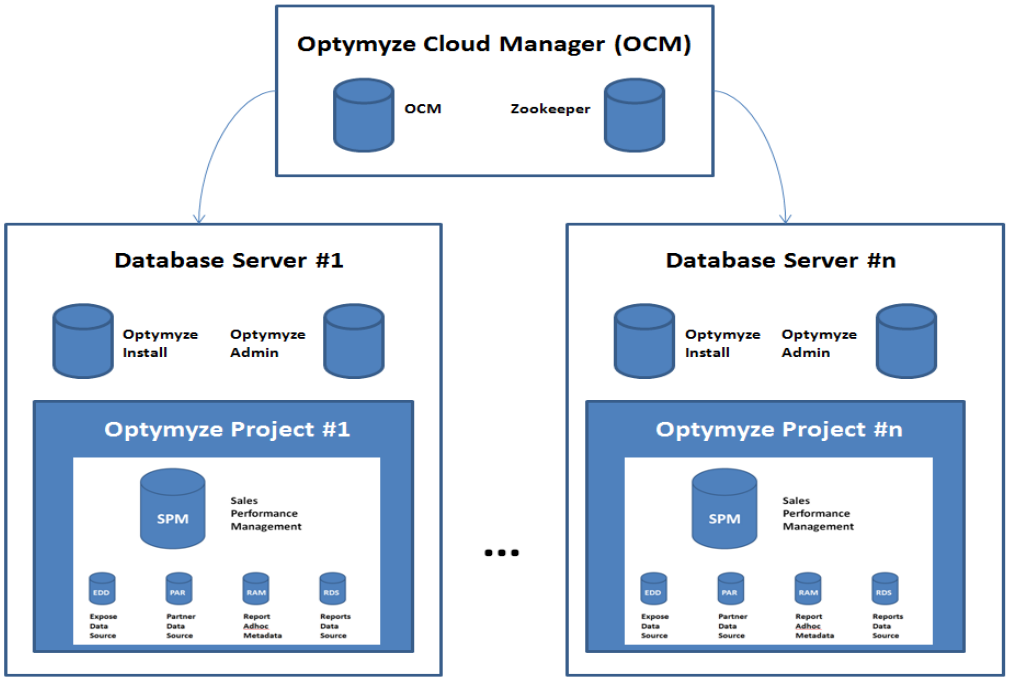
Optymyze Admin (OA) schema, it is special administrator level schema which is used for monitoring all the Optymyze projects on particular DB server. For security reasons, this schema cannot be accessed and its password is unknown. It basically allows performing security sensitive tasks without granting explicit rights to the actual schema.



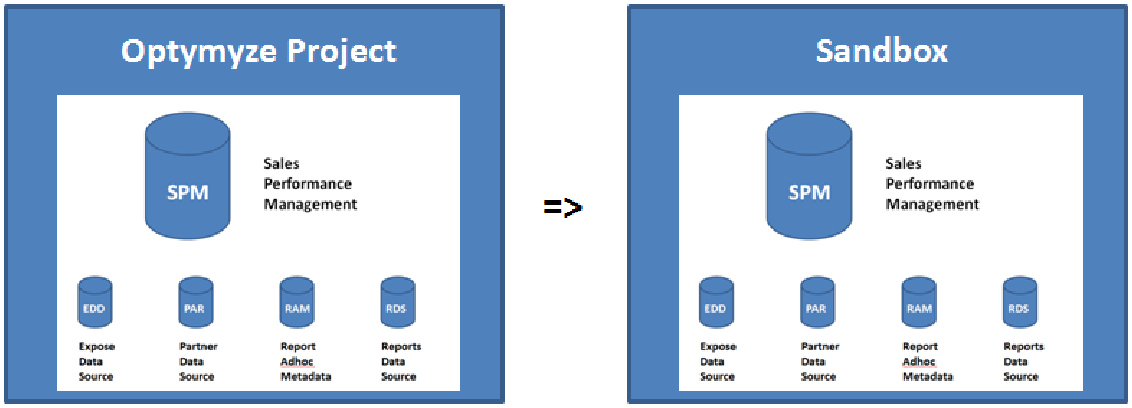
Further, Optymyze projects located on one or more database servers are logically co-ordinate by a structure called Optymyze Cloud Manager (OCM).

One instance of OCM contains two databases:

- OCM database
- Zookeeper database



Additionally, we have capability to migrate the objects from one database to another Database for testing purposes or at the request of a project team. This copy is called Sandbox and can be used to test defects and fixes.



Back up process for Database: Optymyze support the migration and restore of individual schemas or group of schemas.

a) RMAN is the supported oracle backup method and includes a backup of the entire database
b) DATAPUMP is a schema level backup method. It's more efficient than RMAN however it has some limitations that it needs non-processing window otherwise if some table structure gets changed during back-up, it will fail.

Optymyze Services

Major Apps or services at operation level that we should be aware of in simple terms are listed below:

Optymyze Cloud Manager: It is the central app used to manage the Optymyze deployments in a datacenter. It has a broad feature set used to configure Optymyze farms, login settings, environments and servers. It also provides support on directly accessing an Optymyze instance, sign Optymyze licenses and many other things. We can explore it using one of the OCM URL's . Access to the URL is through your windows credential.

Dependencies: Oracle Linux 7 and Zookeeper Cluster

Zookeeper: Zookeeper is a distributed, open-source coordination service for distributed applications. It implements higher level services for Synchronization and configuration maintenance.
It is simple, replicated, ordered and fast in nature. It works on Data model with hierarchal tree structure. It is deployed in the cluster form

Nginx: Web server which balances the Optymyze web apps and configured through zookeeper template to set up URI for clients

Kafka: It is distributed streaming platform or message Broker used between OCM and other tools like storagem or farmount. Idea is to write events in Kafka topics from OCM to react immediately for the changes on these applications.

Kubernetes: Framework that helps to orchestrate the containers of Optymyze Application. The smallest unit of deployment is known as POD in its cluster

Optymyze applications: This includes login server, Dispatcher, worker and Optymyze app which are main components installed in the containers

- The login server application was designed to support a multitenant application and work with multiple PAMs in the same deployment. We support numerous authentication

Modules. Here is the list : LDAP, Windows IWA, SAML 2.0, Salesforce PAM, V11 PAM, OAuth/Open ID, Multi-tenant PAM

- Optymyze Web application itself is build by a set of components which provide business functionality. Top level components include System Management, Data Repository, Repository, Data Automation, Data Transformation, Visualization, Sales Studio, Sales Compensation, Sales Force Performance, Sales planning……etc.

- Optymyze Processing: Application use to process/transform the client data for necessary outcome is mainly handled by processing framework which needs high compute resources and doesn't happen on web server. Dispatcher is the first component which assists in scheduling, task management, status updating for the tasks executed by other component known as Worker. Both of them are placed in separate containers.

Storage Manager (Storagem): A special purpose tool installed on the Database server which uses a dedicated Linux volume group (vg_opt) and is managed by the storagem application to create logical volumes for each environment and a shared logical volume for all the temp files used by environments hosted on the same database server. The size of these logical volumes is controlled though OCM and the storagem application ensure logical volumes with the requested sizes are created. The application also takes care of sharing the environment root and temp directory though samba and publishing the share details in zookeeper.

Farmount: It is an application that mounts/un-mounts environment directories (Project Root, Temp or Logi-XML) based on which environments are defined for a certain farm/app server. This application is installed on each container that has an Optymyze app, a dispatcher or a worker.

As of now, we have farmount service installed on Application servers which ensures that this application is running but with containerization, it will not required further

FileWatch: FileWatch is a tool designed to do transformations on files appearing in the environment root directories. This tool is used to fill some of the limitation of Optymyze in dealing with files.

Filewatch service is installed on the Database server.

Logging Services: Application and systems logs can help you understand what is happening inside your cluster. The logs are particularly useful for debugging problems and monitoring cluster activity. Our logging architecture uses fluent which runs along the Optymyze application into its own container as part of the Optymyze pods. It looks for the logs generated by Optymyze and parses them into fields. It then sends them to an AWS Elastic Search cluster which is configured though OCM and is available as a service in the cluster. Later, we can use Kibana to extract the meaningful information from the logs.

Chapter 3 – Community & Service Desk

Community is the platform where Optymyze customers, partners, prospects, sales and marketing teams can come together to gain, share and disseminate knowledge and information about Optymyze products and services. The eventual goal of the community site is to promote Optymyze brand, share our expertise and knowledge through training & Documentations, attract prospects and customers as well as get quick feedback on new products and services.

We have expert in support team for community administration. On technology side, Community is PHP based application on the Drupal platform using Apache Web server on the backend with MySQL database.

As usual tasks, we handle user account management and configuring the modules on this portal as per requirement plus handle the maintenance/upgrade for this system. Community is also integrated with support site which manages release notes, enhancement requests and software issue information.

Service Desk is JIRA based application used for support site which is main tool for support team to receive software issues from the client or professional services. All the implementation is configured by few members of the support team only and team is responsible for administrating/upgrading or backup process for this application.

We have following below listed projects present in the application:

Customer facing projects

a) ***Enhancement Request*** – It receives all the requests opened by client for improvisation, adding new feature/functionality in our Product Optymyze. The owner of this project is Product Managers
b) ***Optymyze Customer Support*** - This project is mainly handled by Cloud Operation teams which includes support, Performance Management, application management and DBA's to resolve the issues or making changes in the backend configuration of Optymyze Portal

Non-Customer Facing Projects

c) ***Optymyze Security Management***- This project is reviewed by security team to track all the compliance related information
d) ***Optymyze Change Control*** - This is purely project handled by Migration teams and supervised by security when there is request for Server Decommission or addition.
e) ***Optymyze Customers***: Project allocated to the installation team and administrators for tracking the records of the fresh install of any customer while provisioning.

Apart from all the projects, we have our customized Plugin installed on this system to isolate the client requests from each other with security set up and few other improvisations are made to escalate the issue in another Development JIRA system through API calls for Development teams.

Account Management

User-base for Service Desk & Community portal is managed through Atlassian SSO capable tool Crowd which is integrated with both applications and manages their groups along with user Data in itself.

The above statement is applicable if user is member of both Service Desk and community Portal. Service is user license based application where only 5 members from the client side are given access and rest of them can only login into the community.

User only present in community portal is managed through its own user directory and support is responsible to create accounts for the requests raised by client or PS through an email triggered from Community login page to CommunityAdmins.

Members who have permissions to create users: *Rahul Arora, Asmita Mer, Devanshi Kachrani, Anil Tatipaka*

For Crowd based access, we have two major directories where few teams are incorporated to use their windows credentials through integrated Active directory and all other teams (PS/Clients) are managed through Public sign-up directory.

Crowd group has important role to isolate the users from each other by differentiating them through separate group allocations. For example:

- If the user is partner, we need to assign the group name of his company (Like for Optymyze, it is Optymyze group)
- If it is direct client, always assign the group "customers".

This is first mandatory group and other than this we can assign single group to the customer associated to their company name and one or multiple groups to the partner corresponding to their assigned projects.

For community users, we just create user in the Drupal directory with role: Employee/Partner or Customer based on the request we received.

One typical account sync scenario:

Community user may need support site access in future and in such cases, the password sync is required on the very first login to the service desk for SSO capability between both applications which can only trigger if that user reset his community credentials and re-login with new password to access support portal.

Daily Usage and Feedback

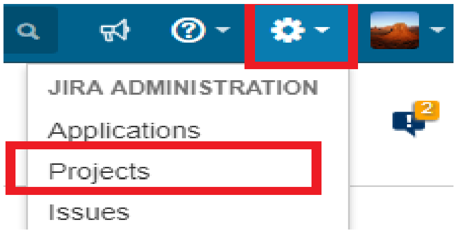
As a support team member, we are solely responsible to handle support issues opened in service desk everyday meeting the SLA's configured in this system itself. You can find SLA related details here

All the support issues are handled hierarchal under OCS project and Software Support Issue Type and we have dashboard created to monitor the issues. All newly created issues also trigger an email to all the members of the team.

Last year, we integrated ***Feedback tool*** in our service desk system where all the closed tickets send notification to the Report for survey and the data metrics are collected from them on basis of three aspects: Communication, Resolution provided and Overall satisfaction.

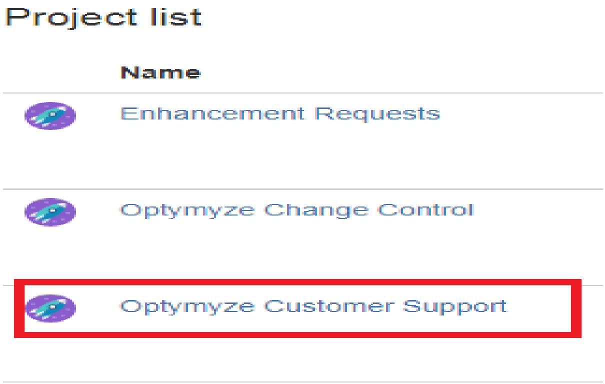
How to check feedback report:

- Click on right hand side wheel button and select Projects



JIRA ADMINISTRATION
Applications
Projects
Issues

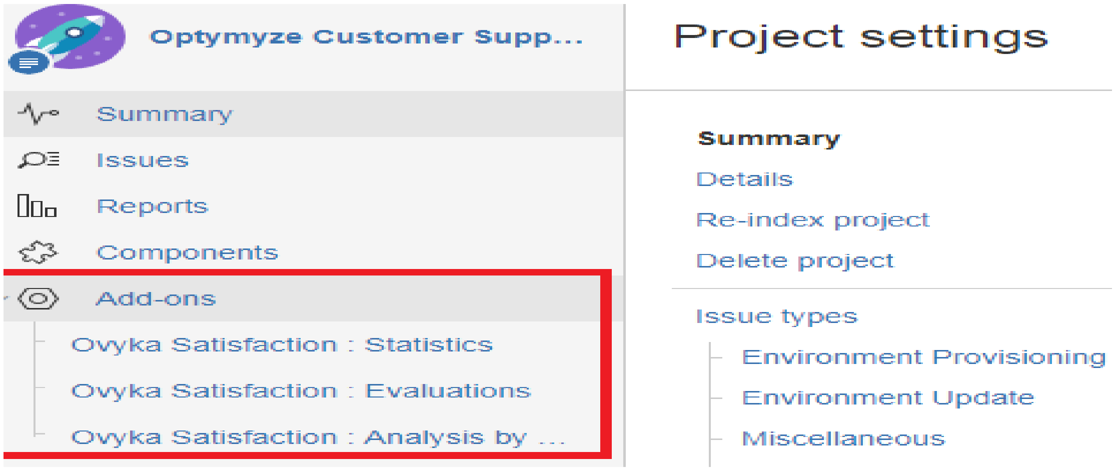
- Go to Optymyze Customer Support



Project list

Name
Enhancement Requests
Optymyze Change Control
Optymyze Customer Support

- Click Add-ons and select different satisfaction metrics to verify



Optymyze Customer Supp...

- Summary
- Issues
- Reports
- Components
- Add-ons
  - Ovyka Satisfaction : Statistics
  - Ovyka Satisfaction : Evaluations
  - Ovyka Satisfaction : Analysis by ...

Project settings

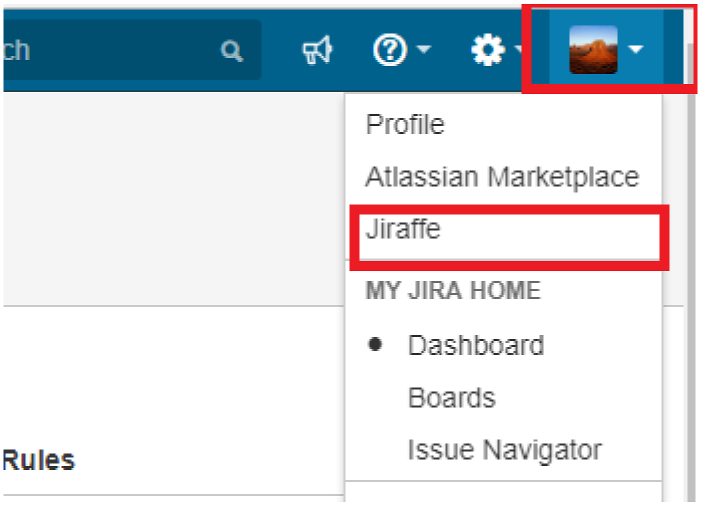
Summary

- Details
- Re-index project
- Delete project

Issue types

- Environment Provisioning
- Environment Update
- Miscellaneous

Apart from this tool, we also have capability to assign points for every support member to see how they are involved in handling support issues. It measures your ***daily usage*** on the system and is managed through Jiraffe. It is accessible to everyone under their service Desk Profile on Right hand side in menu bar.



- Profile
- Atlassian Marketplace
- Jiraffe

MY JIRA HOME

- ● Dashboard
- Boards
- Issue Navigator

We also have additional report capability tool (**EazyB**i) which helps to create different kind of metric reports for leadership to analyze the client problems on every release basis. This is managed by Service Desk Admins (*lead by Asmita Mer*).

These reports also integrated with Service Desk Dashboards for client exposure.

Community & Service Desk Administration

Support team take care of all the maintenance/ upgrades and back up activity for Service Desk and community.

Backups are configured through cron job to run daily on both of these applications and later they are migrated on AWS S3 buckets thrice in week.

Weekly service Desk maintenance is scheduled to run automatically, no more human efforts required.

Process to start & Stop service Desk

- Login to the service desk server and locate to path: */opt/atlassian/newjira/bin*
- Run script as it is: **sh stop-jira.sh**
- Check Process ID of JIRA (command: **ps -ef |grep –v jira**) and it should give empty results
- Check portal if it is down
- Wait for 2-3 minutes and Start JIRA using command: **sh start-jira.sh** at same location and make sure URL is up and running.

*****End of Module 1****

We have small assessment test to check your knowledge before moving forward.

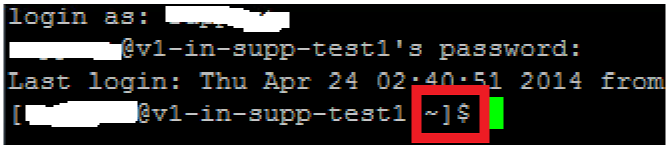
*****Start of Module 2****

Chapter 4 – Linux Fundamentals

As we are using Linux operating system in our Optymyze for all servers, so we should have basic knowledge of Linux command so that we know how to operate and make changes in the system.

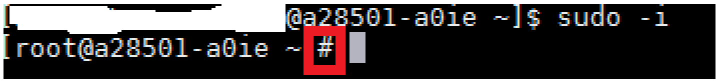
Few important things to be noted before we go into further details:

1) Linux commands are case-sensitive
2) When you login to the system through Putty, you get shell prompt ($)



```
login as: [redacted]
[redacted]@v1-in-supp-test1's password:
Last login: Thu Apr 24 02:40:51 2014 from
[[redacted]@v1-in-supp-test1 ~]$
```

3) Support has rights for root access which is available through command : **sudo -i**
4) Once you access the system through root user, shell prompt is changed to (#)



```
[[redacted]@a28501-a0ie ~]$ sudo -i
[root@a28501-a0ie ~]#
```

5) Root access shouldn't be used for normal activities on the system. Use it only if the task requires root permission

Commands: (in Italics)

- Display Linux system information
  uname -a

- Display kernel release information
  uname -r

- Show how long the system has been running + load
  uptime

- Show system host name
  hostname

- Show the current date and time
  date

- Display free and used memory (-h for human readable, -m for MB, -g for GB.)
  free –h

- Display and manage the top processes along with CPU consumption

  ```
  top -c
  ```

- Interactive process viewer (top alternative)

  ```
  htop
  ```

- List all open files on the system

  ```
  lsof
  ```

- List files opened by user

  ```
  lsof -u user
  ```

- List the file-system with free disk space and file-system mounts

  ```
  df –kh
  ```

- List all files in a long listing format including hidden files

  ```
  ls -al
  ```

- Display the present working directory (current path)

  ```
  pwd
  ```

- Copy file1 to file2

  ```
  cp file1 file2
  ```

- Rename file1 to file2

  ```
  mv file1 file2
  ```

- Move file1 from one directory to other directory

  ```
  mv (source path of file) (Destination path of file)
  ```

For ex. Moving file "temporary.txt" from /opt/Mytest/ to /var/Mytest directory

```
mv  /opt/Mytest/temporary.txt  /var/Mytest
```

- Force removal of file without prompting for confirmation

  ```
  rm -f file
  ```

- View the contents of file

  ```
  cat file
  ```

- Display the last 10 lines of file and follow the file as it grows (Good to use for checking live logs)

  ```
  tail -f file
  ```

- Display process information for processname

  ps -ef | grep processname

- Kill process with process ID (pid can be fetched for the process from above command)

  kill pid

- Display listening tcp and udp ports and corresponding programs

  netstat –nutlp

- Display description and summary information about package

  yum info package

- Search for pattern in file (Good to search exception in the log file)

  grep pattern file

- Directory navigation

  cd

For ex. changing to web log directory
cd /opt/glassfish4/glassfish/nodes/localhost-domain/(WEBCLUSTER_NAME)/logs
WEBCLUSTER NAME is different for all projects so you can navigate directory by directory as well

- To go up one level of the directory tree

  cd..

- Go to root directory

  cd /

- Go to your Home directory

  cd $HOME

FILE PERMISSIONS

4 – Read
2 – Write
1 – Execute
If you want to give all permissions to user, group or others, it will sum of all
4+2+1 (User)
4+2+1 (Group)
4+2+1 (Others)
i.e. 777

Command to change permissions: ***chmod (permission) (file_name)***

Let's consider an example:
If I need to provide permissions to file "**bravo.txt**" as following:

- User with read, write and execute
- Group with read, write
- Others with read-only

Command would be: **chmod 764 bravo.txt (7 for user, 6 for group, 4 for others)**

In case, we need to provide permissions to all the files in directory, we can use recursive parameter.

chmod –R (permissions) (directory_name)

<u>Another way to provide permissions using "**ugo**" parameter</u>

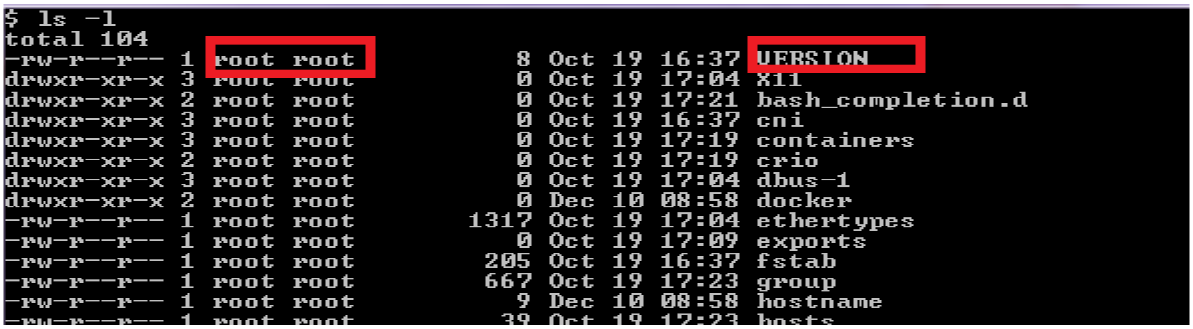
chmod u+rwx,g+rw,o+r bravo.txt

This will provide similar permissions as asked in question
(Parameters: r- read, w-write,x-execute) (u- User, g-group,o-others)

Now, here is Question: what is user, group and others in the file permissions?

User: It is the owner of particular file or directory
Group: Group to which file are accessible (sometime, we need to provide access to number of users
Others: Users who neither own the file nor present in the group attached to the file but sometimes need permissions for particular tasks.

Using (**ls –l**), we can list the information of the files which shows user and group of the file as well. Below screenshot shows VERSION file with root user and root group



```
$ ls -l
total 104
-rw-r--r-- 1 root root         8 Oct 19 16:37 VERSION
drwxr-xr-x 3 root root         0 Oct 19 17:04 X11
drwxr-xr-x 2 root root         0 Oct 19 17:21 bash_completion.d
drwxr-xr-x 3 root root         0 Oct 19 16:37 cni
drwxr-xr-x 3 root root         0 Oct 19 17:19 containers
drwxr-xr-x 2 root root         0 Oct 19 17:19 crio
drwxr-xr-x 3 root root         0 Oct 19 17:04 dbus-1
drwxr-xr-x 2 root root         0 Dec 10 08:58 docker
-rw-r--r-- 1 root root      1317 Oct 19 17:04 ethertypes
-rw-r--r-- 1 root root         0 Oct 19 17:09 exports
-rw-r--r-- 1 root root       205 Oct 19 16:37 fstab
-rw-r--r-- 1 root root       667 Oct 19 17:23 group
-rw-r--r-- 1 root root         9 Dec 10 08:58 hostname
-rw-r--r-- 1 root root        39 Oct 19 17:23 hosts
```

Command to change the owner of file: ***chown (new_owner_name) (file_name)***

Command to change the group of file**: chgrp** ***(new_group_name) (file_name)***

We can do recursive owner change in the directory using parameter "–R"

chown –R (new_owner_name) (directory_name)

As an example:

Assume **bravo.txt** file has owner "apache" and group "apachesvc" and we would like to change the group to "apachegrp"

Command to use: **chgrp apachegrp bravo.txt**

Services related commands:

LINUX6:

- Check the status of service
  service (service_name) status
  ex. service farmount status
- Start the service
  service (service_name) start
- Stop the service
  service (service_name) stop

LINUX7:

- Check the status of service
  systemctl status (service_name)
  ex. systemctl status filewatch
- Start the service
  systemctl start (service_name *)*
- Stop the service
  systemctl stop (service_name *)*

Chapter 5 – ORACLE Fundamentals

Since we work on troubleshooting Database related issues, we have access to create Read-only and investigation user for triaging the problem at some level. In order to troubleshoot such issues, it is necessary that support team should able to run SQL queries and gets output as required to understand the scenario or cause of an issue. This section is particularly give us brief introduction about the oracle important concepts which we should aware of but not related to Optymyze. It is meant for everyone to go through this article to have conceptual knowledge for troubleshooting issues with Database performance and running queries to retrieve details.

Troubleshooting related to Optymyze would be something discussed later in this book. So, let's start with basics at this moment.

The commands in SQL are called queries and there are two main types:

- Data definition queries (DDL) – the statements that define the structure of your database, create tables, specify their keys, indexes, and so on.

- Data manipulation queries (DML) –Statements used to retrieve the data from the database, and create or update the records in it. For e.g. select, update, insert operations etc.

We mainly deal with DML queries for data in the tables or views and never work to modify the structure at our level of support. Though we get request to create or remove indexes but that is something which can be enabled from UI as well and recommended to do from there itself. However; in the table having huge records, structure changes can create unwanted scenario which required technical support from the backend sometimes.

I am not going in-depth of DDL queries in this chapter but will discuss in details under troubleshooting process so that we know at what level we should escalate the issue to Development team

DML SELECT queries:

a) Query data in columns c1, c2 from a table t
SELECT c1, c2 FROM t; (use * instead of c1, c2 if want to return all columns)

b) Query data and filter rows with a condition
SELECT c1, c2 FROM t WHERE condition; (c1,c2=SOME VALUE as condition)

c) Sort the result set in ascending or descending order
SELECT c1, c2 FROM t ORDER BY c1 ASC [DESC];

d) Count the rows for column c1 from the table t
SELECT COUNT(c1) FROM t

e) Query data and filter rows on multiple value in the condition
SELECT c1, c2 FROM t WHERE c1 IN (value1, value2….)

f) Query data and filter rows on expression based condition (This is valid for character values)
SELECT c1, c2 FROM t WHERE c1 LIKE %val% (c1 COLUMN is of type character)

What is JOIN? (This is something used a lot during merge field Conditions)

A JOIN clause is a part of a SELECT statement; it allows you to specify multiple tables for data retrieval.

Syntax:

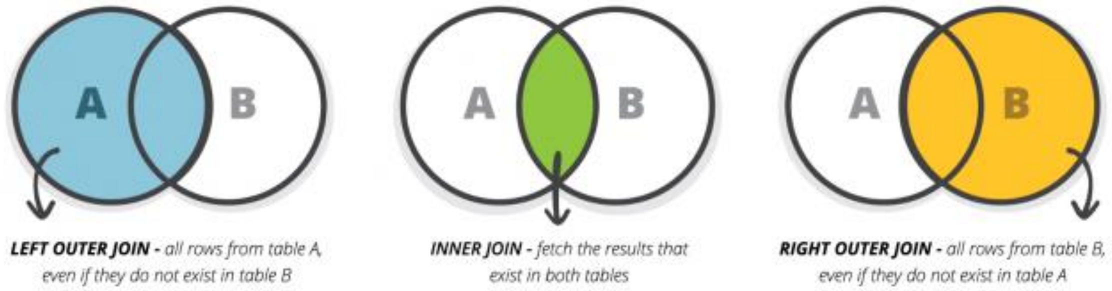
SELECT … from table1 JOIN table2 ON table1.id = table2.t1_id

We have joined two tables in the above syntax using id of table1 with t1_id of table2. This is possible if we keep reference of unique key of first table as foreign key in other table.

ID Column is unique in table1 and t1_id has the value of table1 ID's present as column in table 2 acts as foreign key.

Types of Join:

- INNER JOIN – Fetch the rows that exist in both tables
- LEFT OUTER – Fetch all rows from the Left Table, even if they do not exist in right table so the result set shows empty value for right table corresponding to those rows.
- RIGHT OUTER – it is vice-versa of LEFT JOIN: fetch all rows from Right table, even when the corresponding data in Left table is absent.


A
B
LEFT OUTER JOIN - all rows from table A, even if they do not exist in table B
A
B
INNER JOIN - fetch the results that exist in both tables
A
B
RIGHT OUTER JOIN - all rows from table B, even if they do not exist in table A

<u>INDEX</u> (Good for filtering/sorting views**)**

An **index** is a schema object that contains an entry for each value that appears in the **indexed** column(s) of the table or cluster and provides direct, fast access to rows. They are **used in Oracle** to provide quick access to rows in a table and in general to speed up the query.

Creating indexes without any usage is unnecessary overhead because if you change the structure or data in the table, it will need to update the indexed columns as well.

<u>Table/Column/Index Statistics</u> (Good for updating/deleting records for table**)**

***Brief Introduction*:**

Oracle has in-built program called Optimizer which decides what to do with SQL statements and how can we do in quickest possible way.

When a SQL statement is executed, the database must convert the query into an execution plan and choose the best way to retrieve the data. For Oracle, every SQL query has many choices for execution plan like which index to use to retrieve table row, what order in which to join multiple tables together, and which internal join methods to use. These are computed by Oracle Optimizer.

The choice of executions plans made by the Oracle SQL optimizer is only as good as the Oracle statistics which means best execution plan is always chosen based on information about the tables or indexes present in the query.

For better understanding, I have given example here:

Let's says you have table called Support Members which has around 14 rows containing the details of these members. If we write query to get the records of this table in descending order, there will be execution plan created at the backend where it would be having sequence like:

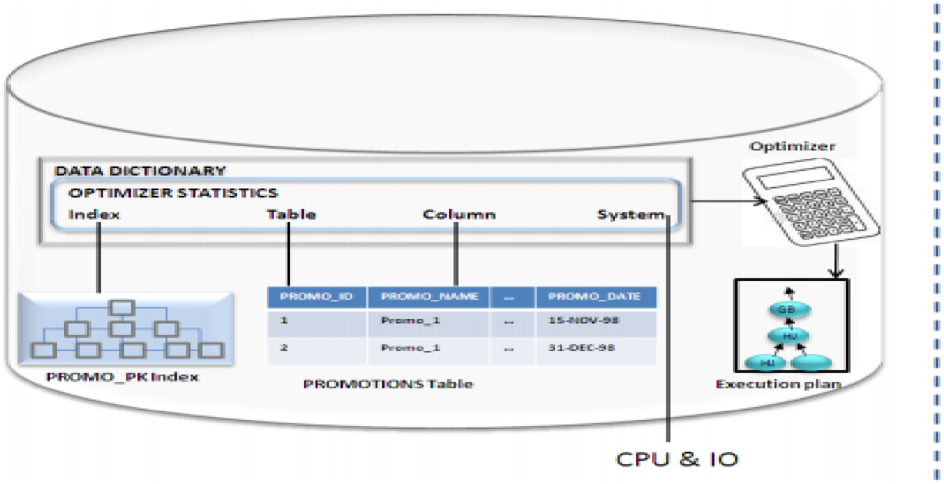
Optimizer tells Oracle to first go get all the rows of the Support Member table, and then sort those rows as required. (So we have table full scan as first step in the plan sequence).

In this whole process, we will have column called ROWS in backend plan which will say that we are going to process 14 rows. Isn't it smart that oracle find it? Well Oracle isn't smart, it knew because we generated statistics on Support member table after we created it.

In general, statistics is nothing but metadata for all the tables, indexes, Columns, schemas that we create in oracle Database and the optimizer program use these statistics to get execution plan for every statement which needs to be executed on the database.

Important Note: *dbms_stats* is the default package used by Oracle to generate or gather statistics

For information, Statistics are stored in the data dictionary, and can be accessed using data dictionary views



PROMO_ID	PROMO_NAME	...	PROMO_DATE
1	Promo_1	...	15-NOV-98
2	Promo_1	...	31-DEC-98

As we discussed about the Optimizer, statistics and Execution plan, there is an important component which helps in improvising the oracle query performance. It is none other than **HINTS**

A hint is an instruction to the optimizer. We can alter the SQL execution plans using these instructions and help to use specific approach for particular coming query. These can be specified on particular single table, multiple tables and specific query block or to the entire SQL statement.

One of the examples for hints: We can set instructions to do index scan for the data views if indexes are present and the query is still choosing full table scan while creating executing plan and resulting in retrieving data slowly on the portal.

FRAGMENTATION

Oracle is high performance engine where data is stored in the blocks under table-spaces which are nothing but logical components of your data files. Since, Oracle is capable of doing thousand of concurrent tasks at same time to run DML queries (like inserting the data), there is possibility that data is scattered among discontiguous data blocks (i.e. disconnected data blocks in different clusters).

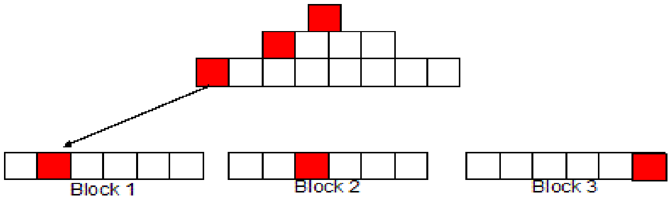
To understand table fragmentation, consider a table in original state where all the table rows are tightly packed onto the table blocks and as soon you perform DML activity over those table rows, there is possibility that data is now getting inserted into different, non-adjacent blocks which makes table fragmented and similarly, if you delete the rows, there will be unused spaces scattered over all in the table space.

It poses two main problems:

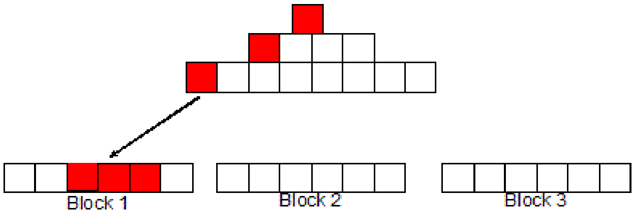
a) Slow scan activity for the tables or Indexes since data is scattered across the different blocks
b) Wastage of Disk due to unused spaces

So, there is need of re-organizing the tables to remove fragmentation and to avoid slowness across your system.

- Data is scattered in different blocks, so scanning the full table rows will be slower here:



- Here we have clustered the data in one block which will definitely help to reduce the slowness



There are many other different ways to remove defragmentation and one of the ways we use to shrink the table for disk reclaim as well. (DBA can be involved in such activities)

Even Data Pump import/export activity helps in reducing the table fragmentation. Usually DBA's have maintenance schedule with all the projects to perform this activity and we are already using ASM disks which are capable of handling some of its part automatically.

*****End of Module 2****

Assessment Test is present for the above section as well. It is recommended to complete it before moving forward.