College of Engineering, Technology, and Computer Science

Design and Implementation of Cloud-based Data Warehousing

In partial fulfillment of the requirements for the Degree of Master of Science in Technology (Information Technology in Advanced Computer Applications)

A Directed Design Project

By

Samson Amede

May 2014

Overview

- Executive Summary
- Multi-Campus Transaction System
- Statement of Problem
- Significance of Problem
- Proposed Solution – Data Warehousing
- Data Warehouse Design
- Hardware and Software Tools
- Procedures Employed
- Reporting
- Conclusion

Executive Summary

Multi-campus transaction systems are quickly becoming popular in the University environment because of the cost saving offered by sharing one system. Typically, the transaction system will be installed on one campus which will host the system to be shared between branch campuses. The challenge encountered with hosted system is the accessibility and interdependence of multiple clients on one centralized hosted system. Constraints in accessibility further hinder providing critical reports in a timely manner. Data warehouse has been the leading solution that most business entities have turned to with a heavy reliance on IT to implement it. In order to overcome these obstacles local data warehouse implementation on an Oracle 11g Database Enterprise system as well as cloud-based data warehouse on Amazon Redshift system will be studied for implementation.

3

Multi-Campus Transaction System

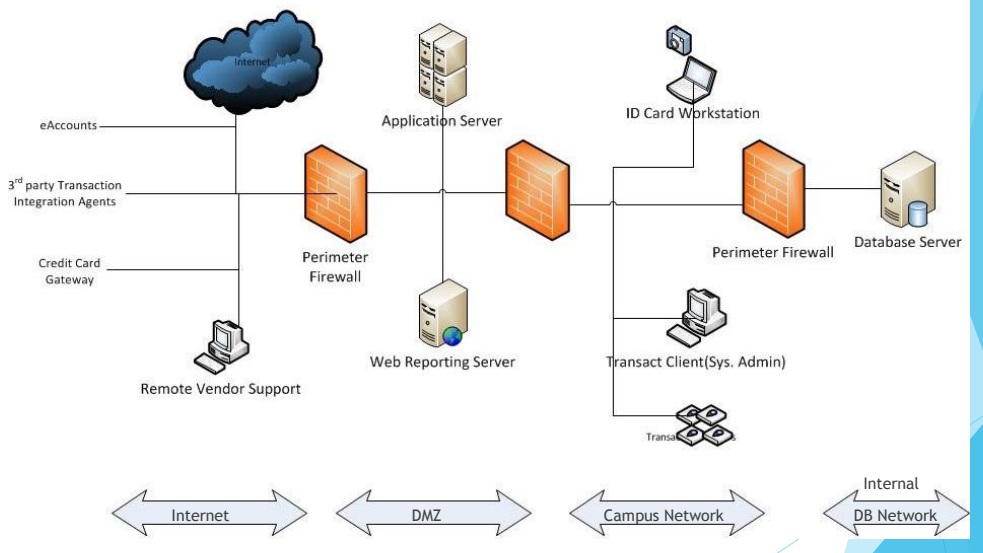
- Integrate various daily business activities to interface with 3rd party vendors.
  - Interface with IPFW Student Information System SIS – Banner.
  - Campus card system
  - Copier system
  - Door Access
  - Vending
  - On-Campus Dining
  - Off-Campus Dining
- Inter-Campus functionality

4

Multi-Campus Transaction System

- Hosted at Purdue Main Campus
- Dedicated Application Server for Transact System
  - Virtualized Server
  - Windows Server 2008 R2
- Dedicated Database Server
  - Linux Platform
  - Oracle Database 11g R2 (Enterprise Edition)
- Shared Web Server for Reporting

5



IT Infrastructure

6

Problem Statement

- Limited accessibility to Off-site Hosted Transact System
  - Firewall Policy
  - Limits IPFW users
- Limited capability of existing Reporting System
  - Only simple query
  - Not customizable
  - Limited data output capability – Customer defined fields

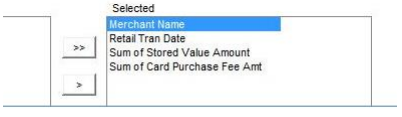
Significance of Problem

- **SELECT last name, firstname, gymexpdate, printdate, barcode**

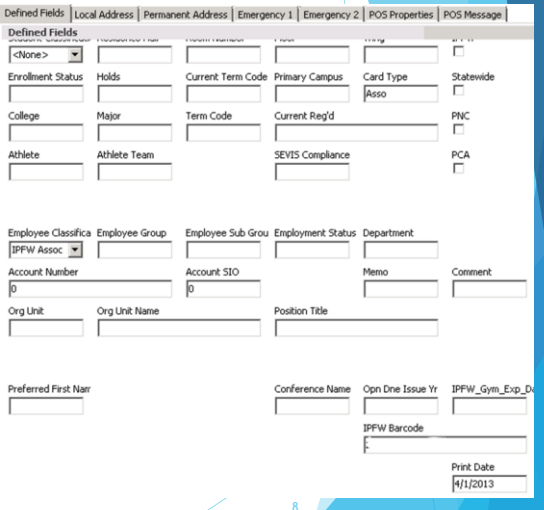
  FROM CDF_table a, IDW b

  WHERE a.custno = b.custno AND

  cardnum like '000000000009%'
- Most Canned Report do not have options to select customer number



GUI for SQL Query ("SELECT" Clause)



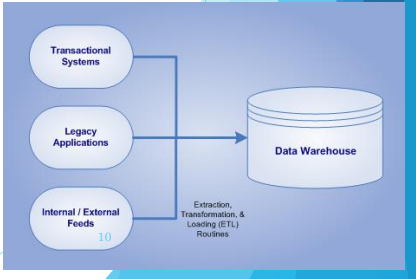
Significance of Problem

- Large volume of Department Copy Card Transactions
  - Approximately 90 on campus departments
  - Multiple Copy Cards per departments
- Limited access for IPFW personnel to deliver reports
  - Monthly Departmental Statement Reports
  - Monthly Internal Billing Reports
  - Monthly Account Deposit Report

9

Proposed Solution – Data Warehousing

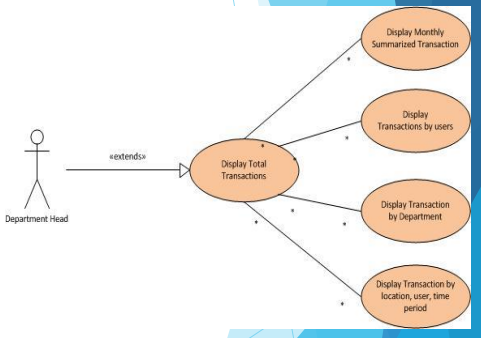
- A database that is accessible across the enterprise that contains current as well as historical data that are important for various business entities for reporting as well as analytical data.
- A copy of transaction data specifically structured for query and analysis.
- Why Data Warehousing?
  - Consolidates, standardizes, and organizes data in order to support business decisions that are made through analysis and reporting.
  - Capture only specific data pertaining to our business model.
  - Future analysis of business process as campus needs expand.



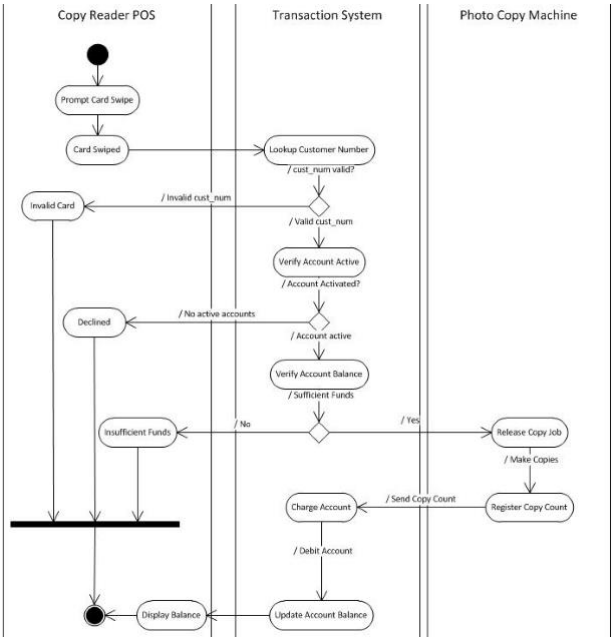
10

Data Warehouse Design

- Requirements Analysis
  - On-demand availability of copier-related data or reports from the data warehouse
  - View joint reports of transactions and any comparative data of summarized transactions between different locations and time periods
  - Long term goal - data warehouse should have the ability to accommodate future changes in the business process where there would be additional merchants and POS coming online as transactions



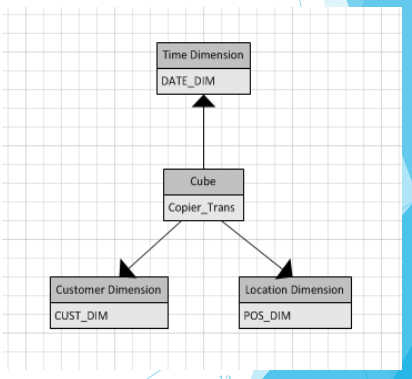
UML Activity Diagram



- Defining Source Data Using Activity Diagram
  - POS Reader Information
    - POS Location
  - Customer Information
    - Customer number
  - Transaction Information
    - Amount of transaction
    - Time of transaction
- Data dictionary
  - Transaction table
  - Customer table
  - POS table
  - ProfitCenter table
  - Account table

Data Warehouse Design

- Dimensional Modeling
  - Unlike Relational model – Denormalized data
  - Brings together data from several tables – Redundancy
  - Central fact table
    - Numeric Measures ex: Sales, # of transactions, Quantity,etc
  - Distributed dimension tables
    - Attributes explaining the fact
  - Star Schema



13

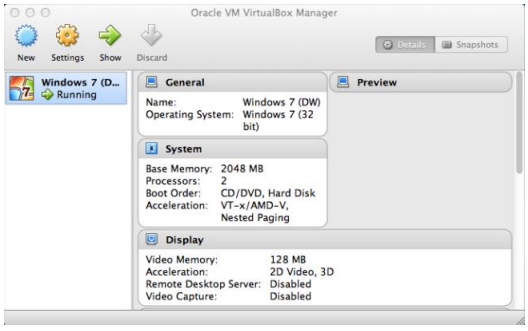
Data Warehouse Design

- Advantages of Dimensional Modeling
  - Queries are consistent, fast and understandable to make business data available to more users because query structure is less of a mystery.
  - The dimensional model applies business rules so the same fact or dimensional attribute always has the same definition.
  - Scalable - Adding new data sources to and adapting to accommodate future changes in the business process are handled in a consistent, reproducible manner.
  - Understandable - data relationships are consistent and typically no more than one level deep. This makes the data structure more understandable for experts and casual users alike. It also facilitates documentation and meta-data set up.

14

Hardware and Software Tools

- Hardware
  - Apple Macbook Air 1.33GHz i-5 processor 4GB RAM 128GB SSD
- Operating System
  - Windows 7 Ultimate 32-bit – VirtualBox 4.2
- Virtual Environment Configuration



15

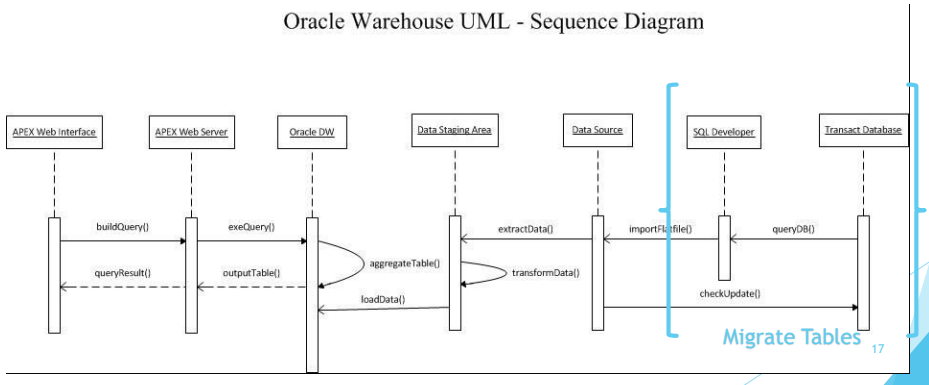
Hardware and Software Tools

- Database Tools
  - Oracle (Thick) 11g R2 Enterprise Ed.
  - Oracle Warehouse Builder
  - Oracle SQL Developer
  - SQLWorkbench/J
  - Postgresql
- Cloud Services
  - Amazon Redshift
  - Amazon S3
- Reporting Tool
  - Pentaho Business Analytics

16

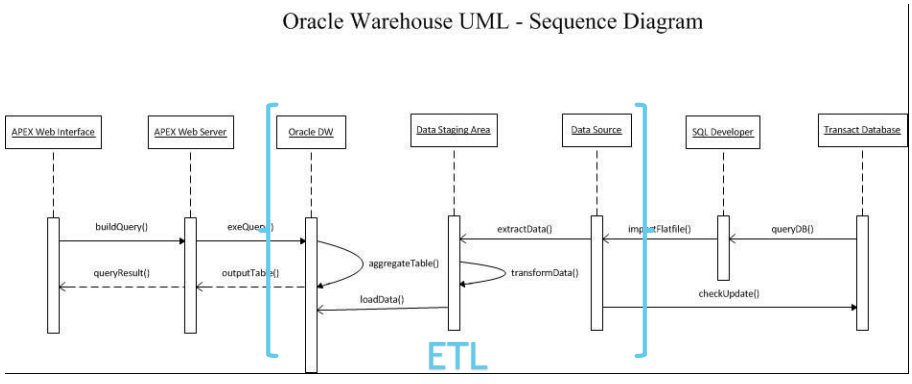
Procedures Employed – Migrate

- Migrate specific tables from Hosted system using SQL Developer
- Instantiate a new Database to load tables on local Oracle Database



17

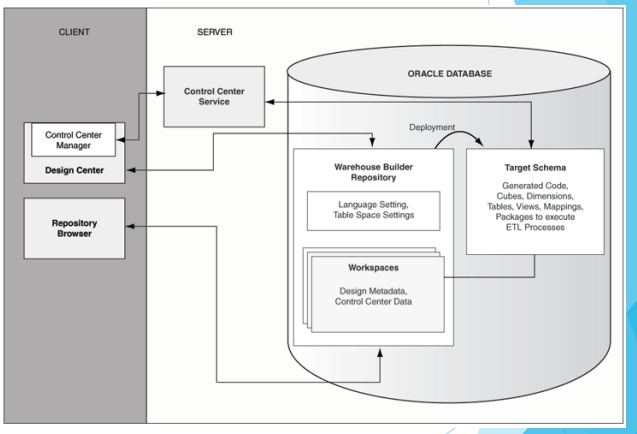
Procedures Employed – Oracle Warehouse



18

Procedures Employed – Oracle Warehouse

- Design Component
  - Design Center GUI
  - Repository Browser
- Runtime Component
  - Workspace
  - Validate, Generate, Deploy
  - Execute



19

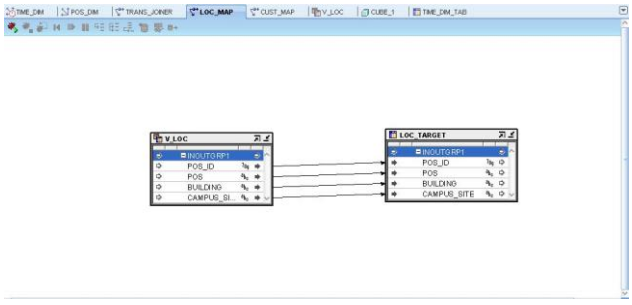
Procedures Employed - OWB

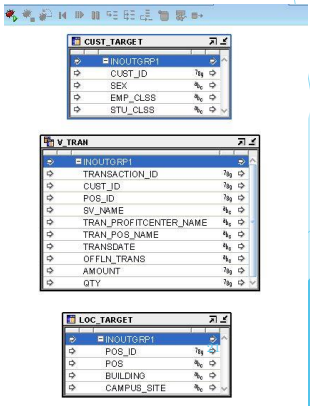
- Oracle Warehouse Builder
  - Create workspace in OWB to load source data and target data
  - Connect OWB with local Oracle DB to load source data
  - Extract, Transform, Load – ETL process
    - Extract and Transform source data in the staging area
    - Configure Target data structure - Dimension Table and Fact Table
    - Load data into Data Warehouse
  - Deploy Oracle Data Warehouse

20

Procedures Employed- Extract

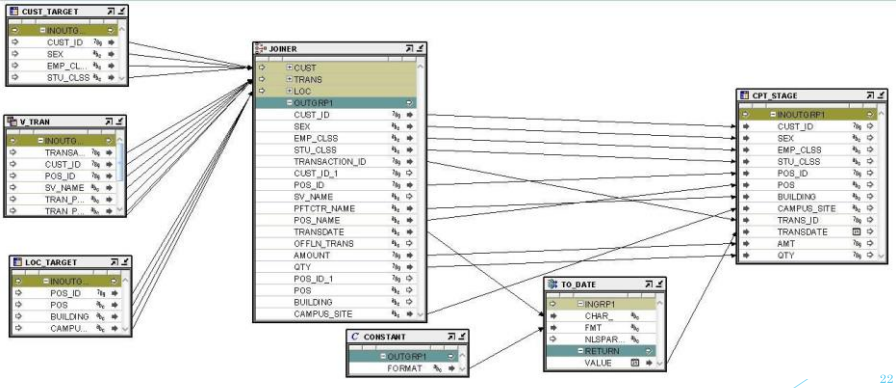
- Extraction
  - Extract from source data table
  - Map to staging table
- Tables to Extract
  - Transaction table (Fact Table)
  - POS table (Dimension Table)
  - Customer table (Dimension Table)





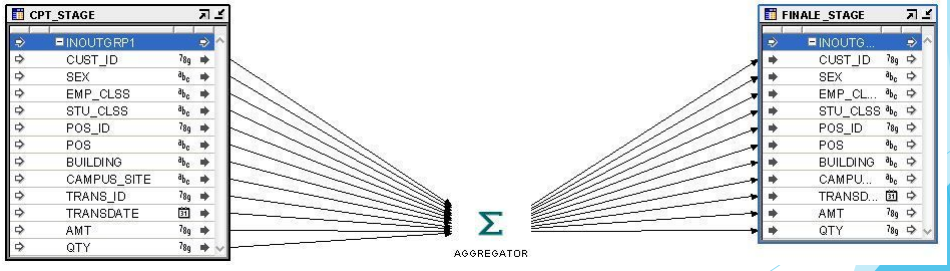
Procedures Employed - Transform

- Transformation
  - Joiner Operator
  - Conversion Operator - Char ToDate()
- Joiner Operator
  - Join tables
  - Load to next staging table



Tools and Procedures Employed - Transform and Load

- Transformation
  - Aggregate Operator
  - Select group by attributes
  - Sum up measures: transaction amount & Quantity
- Load
  - Final staging table to data warehouse (Fact and Dimension Tables)
  - Validate, generate, deploy
  - Execute mapping to deploy data warehouse

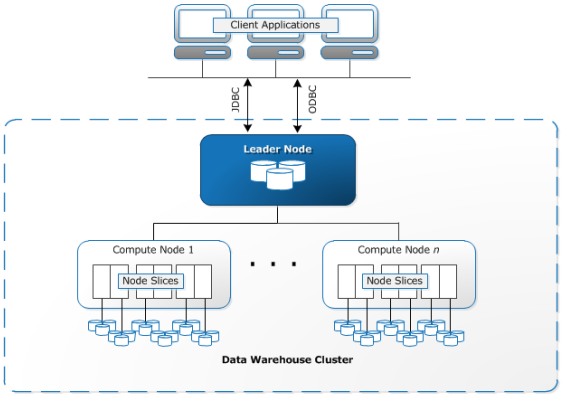


Amazon Redshift Architecture

Client Applications: On the data source loading side Amazon Redshift integrates with various vendor solutions to perform data loading and ETL procedures to load to the Redshift Data Warehouse. On the data extraction and querying side Redshift also has the ability to integrate with various Business Intelligence reporting, analysis, and mining tools provided by various business partners.

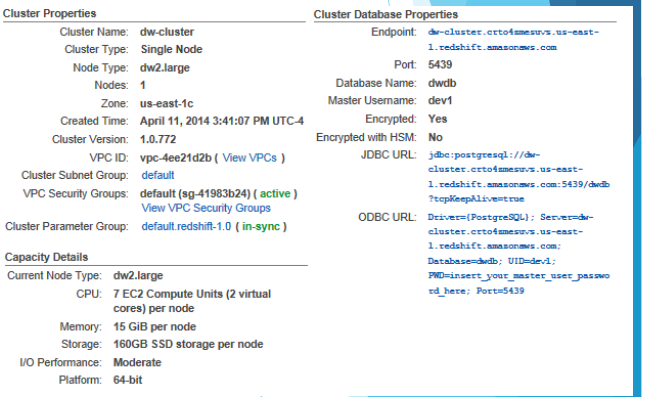
Connections: In order to load and extract data to and from the data ware house, Redshift communicates using industry-standard PostgreSQL JDBC and ODBC drivers

Clusters: A cluster is the core infrastructure component of Redshift data warehouse. Depending on the data warehouse that needs to be provisioned Redshift can be configured to accommodate the needs of any size data warehouse. If we provision more two compute nodes Redshift will put an additional leader node to coordinate the compute nodes. The client applications will only interact to with the leader node



Procedures Employed – Amazon Redshift Cluster

- Amazon AWS
  - Provision cluster in Amazon Redshift and
  - Create buckets in Amazon S3 to load OWB Data Warehouse
  - Load Amazon Redshift from Amazon S3 using SQLWorkbench/J
  - Launch Amazon Redshift Data Warehouse



Cluster Properties

Cluster Name:	dw-cluster
Cluster Type:	Single Node
Node Type:	dw2.large
Nodes:	1
Zone:	us-east-1c
Created Time:	April 11, 2014 3:41:07 PM UTC-4
Cluster Version:	1.0.772
VPC ID:	vpc-4ee21d2b (View VPCs)
Cluster Subnet Group:	default
VPC Security Groups:	default (sg-41983b24) (active) View VPC Security Groups
Cluster Parameter Group:	default.redshift-1.0 (in-sync)

Capacity Details

Current Node Type:	dw2.large
CPU:	7 EC2 Compute Units (2 virtual cores) per node
Memory:	15 GiB per node
Storage:	160GB SSD storage per node
I/O Performance:	Moderate
Platform:	64-bit

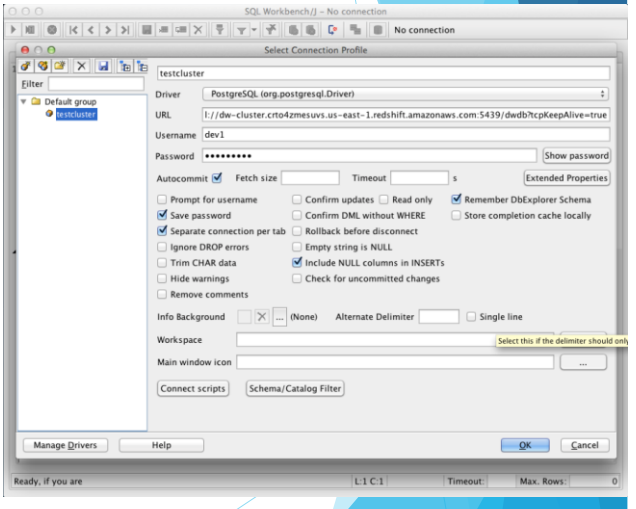
Cluster Database Properties

Endpoint:	dw-cluster.crto4zmesuvs.us-east-1.redshift.amazonaws.com
Port:	5439
Database Name:	dwdb
Master Username:	dev1
Encrypted:	Yes
Encrypted with HSM:	No
JDBC URL:	jdbc:postgresql://dw-cluster.crto4zmesuvs.us-east-1.redshift.amazonaws.com:5439/dwdb?tcpKeepAlive=true
ODBC URL:	Driver={PostgreSQL}; Server=dw-cluster.crto4zmesuvs.us-east-1.redshift.amazonaws.com; Database=dwdb; UID=dev1; PWD=insert_your_master_user_password_here; Port=5439

25

Transfer Data from Amazon S3 to Redshift

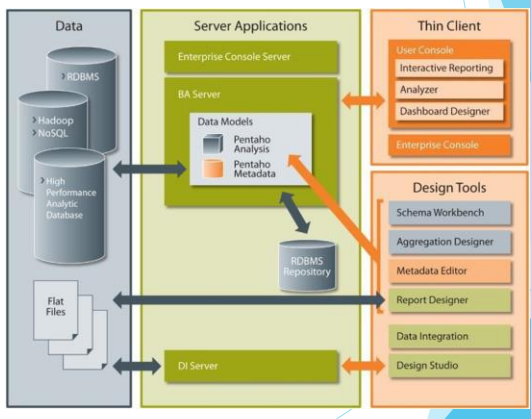
- SQLWorkbench/J
  - Create JDBC connection to Amazon Redshift
  - Using special Postgresql jdbc driver for Redshift
  - Create empty table in Amazon Redshift – SQL code
  - Load table data from Amazon S3 on to Amazon Redshift – SQL code
  - Launch Amazon Redshift data warehouse



26

Reporting – Pentaho Business Analytics

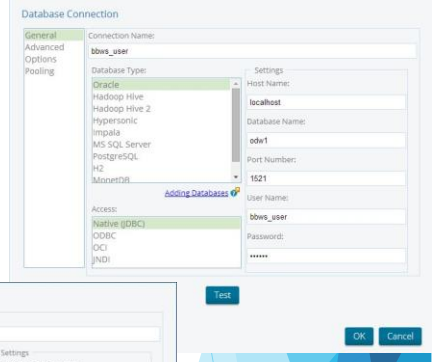
- Pentaho is a certified business analytics and data integration tool that works with Amazon Redshift
- Commercial open source
- Data Integration (ETL)
- Reporting
- Analysis
- Dashboards
- Data Mining
- Business Intelligence Platform

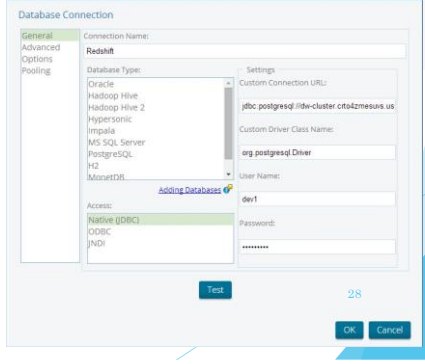


27

Reporting – Pentaho Business Analytics

- Pentaho Business Analytics
  - Create JDBC connection to OWB workspace
  - Create JDBC connection to Amazon Redshift Cluster
  - Configure Interactive Reports for both Data warehouses

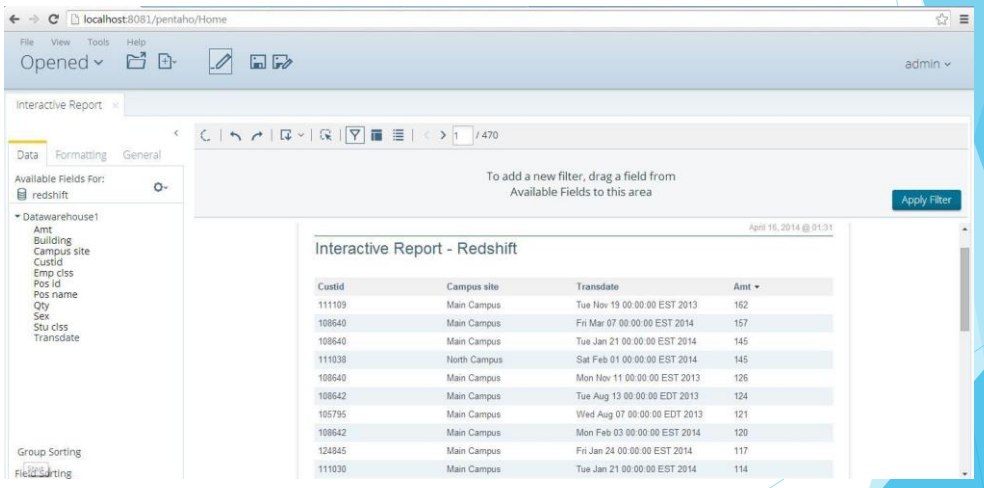




28

Reporting - Pentaho Business Analytics

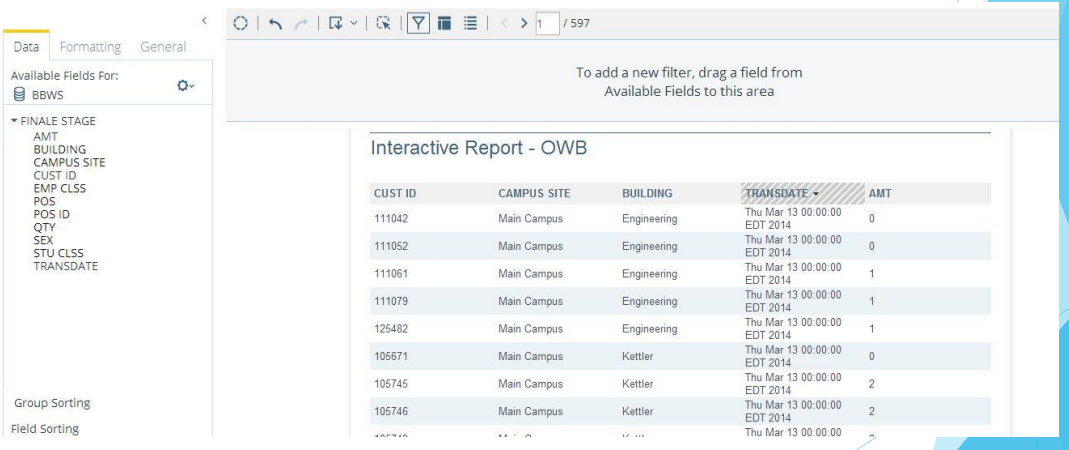
- Interactive Report Output – Amazon Redshift



29

Reporting - Pentaho Business Analytics

- Interactive Report Output – OWB



30

Conclusion

- Amazon Redshift and Oracle Warehouse both address our accessibility issue to reporting since these data warehouses are hosted either in our private cloud or on IPFW premise.
- Pentaho Business Analytics has the option to create profiles for reporting users, power users and business analysts to further expand the access to reporting.
- Pentaho Business Analytics will provide the reporting tools necessary to design and schedule reports such as interactive reports or analyzer reports.
- Oracle Warehouse Builder vs. Amazon Redshift.
- Edge to Oracle Warehouse Builder.