Diamond Enhancement Requests

Release 19.11

Jim Schmidt

Dec 13, 2019

CONTENTS:

CHAPTER

ONE

ITEM STATISTICS

1.1 Overview

During a team phone call on December 15 ABC code requirement was identified.

This demonstrates computing a number of potentially useful statistics

1.2 Approach

- Create a table to hold statistics
- Create a script to populate statistics by item
- Create a service to obtain the data model for the web pages
- Modify the filter screen to allow query filters on the statistics
- Modify the web pages to show the statistics information

1.3 Code

1.3.1 Create the table

drop table ic_item_stat;

create table ic_item_stat (item_nbr integer primary key references ic_item_mast, abc_cd varchar(1), distinct_open_ord_cust_count integer, distinct_org_cust_qte integer, distinct_cust_open_order_count integer

);

alter table ic_item_stat add(check (abc_cd in ('A','B','C')

);

1.3.2 SQlrunner file

This is a yaml file https://en.wikipedia.org/wiki/YAML .. code:: sql

truncate_ic_item_stats: sql: > truncate table ic_item_stat description: delete all the rows from ic_item_stat

distinct_cust_open_order_count:

sql: > update ic_item_stat set distinct_cust_open_order_count = (select count(distinct org_nbr_cst

from oe_ord_hdr_dtl_vw

)

description: distinct cust open order count

number_of_open_orders sql: > description: number of open orders

number_of_distinct_customer_quotes_last_three_months:

sql: > update ic_item_stats (select count(distinct(org_nbr_cst) from sq_qte_vw

distinct_customer_quotes_prev_3_months description: Count distinct customer quotes last three months

max_sales_price_last_three_months sql: > description: Max sales price last three months

quote_conversion_ratio: sql: > description: quote conversion ratio

dollars_open_orders: sql: > description: Dollar open orders

number_of_customers_open_orders:

sql: > update ic_item_stats (select count(distinct(org_nbr_cst) from oe_ord_hdr_dtl_vw where ord_stat_id = 'O'

description: number of distinct open order customers

abc_code: sql> description: Pareto code narrative: >

A,B,C classification compute aggregate sum of dollars Basing on shipments does not reflect new items Basing on open orders the dollar totals will be low for new items

unallocated_order_qty: description: Unalloc ord qty sql: >

percent_of_open_orders_base_curr sql: > description: Percent of open orders for base currency

1.3.3 Create a service

package com.pacificdataservices.diamond.apsweb;

import java.io.IOException; import java.sql.Connection; import java.sql.SQLException;

import javax.sql.DataSource;

import org.javautil.core.json.JsonSerializer; import org.javautil.core.json.JsonSerializerGson; import org.javautil.core.sql.Binds; import org.javautil.core.sql.SqlStatement; import org.javautil.util.NameValue; import org.slf4j.Logger; import org.slf4j.LoggerFactory; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RequestParam; import org.springframework.web.bind.annotation.RestController;

@RestController public class IcItemStatController {

private Logger logger = LoggerFactory.getLogger(getClass()); @Autowired private DataSource datasource;

@RequestMapping("/icItemStat") public String planData(

@RequestParam(value="itemNbr") String itemNbr) throws SQLException, IOException {

logger.info("invoked with itemNumber {}",itemNbr); Connection conn = datasource.getConnection(); SqlStatement ss = new SqlStatement(conn, "select * from ic_item_stat where item_nbr = :item_nbr"); Binds binds = new Binds(); binds.put("item_nbr", itemNbr); NameValue nameValue = ss.getNameValue(binds,true); JsonSerializer serializer = new JsonSerializerGson(); String json = serializer.toJsonPretty(nameValue); return json;

```
    }
}
```

1.3.4 Create a node service

```
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
...
getIcItemStat(): Observable<IcItemStat[]> {
     return this.http.get<IcItemStat[]>(HOST + '/api/v1/icitemstat/' + itemnbr);
 }
```

1.4 Modify web page template

CHAPTER

TWO

OPTIMAL REPLENISHMENT QUANTITY

2.1 Overview

Optimal Replenishment Quantity has two contradictory goals

- Larger replenishment quantities may reduce cost per unit if the item is manufactured on

demand due to the setup cost being apportioned over a smaller number of units. However, popular items may be continually produced by a manufacturer or produced in sufficient quantities to TODO ameliorate or minimize this.

Larger replenishment quantities reduce safety stock requirements as variations in demand may be satisfied by working inventory.

Larger replenishment quantities reduce inventory turns. Inventory turns are defined as average months supply on hand / (average monthly consumption * 12).

Higher replenishment quantities may negatively effect cash flow. If inventory can be acquired on consignment or larger purchase quantities can be scheduled over several months with minimal additional per unit cost this effect can be mitigated. Vendor negotiations are required to achieve either objective, vendor incentive may be a commitment to buy a larger number of parts.

Not to be dismissed lightly is the strange pricing as a part is quoted as a different part number and a vendor willingness to issue a certificate of compliance for the equivalent part or customer willingness to accept an alternate part number.

Ignorance of equivalent part numbers is common in the industry.

Although part numbers are specified in engineering drawings vendors and customers may give different names, including varying use of dashes. This is addressed in Diamond and is called *nomenclature*. Prior deployments have had in excess of one million alternate names for various parts. Acquiring this data may be labor intensive with benefits for Pareto part class ("ABC_CODE") of class "A", the top 20 percent of items based on annual sales currency.

Let us assume in the interest of simplification a scenario that ignores Approved Manufacturer constraints, lead times and cash-flow mitigation strategies, such as

2.2 Optimal Cash Flow

Approaches

Consignment inventory Scheduled Delivery

2.3 Stock outs

2.4 Lowest cost

2.5 Compute Optimal Replenishment

UC = Unit Cost : Setup Cost / unit qty + incremental cost

Replenishment Cost = Unit Cost * Unit Qty AMC Average Monthly Consumption AMOH : Average Monthly Onhand Replen Qty - sum AMC for i (1 : n) < Replen Qt Replen Qty Carrying cost = AMOH * COF COF : Cost of Funds Receiving Cost: Cost to place a purchase order line, receive, putaway and pay an AP line Risk of Obsolescence : Fewer customers increases risk amonth other factors, varies over time

Compusting optimal replenishent quantity is straight forward math at this point.

Throw in a maximum outstanding value of new inventory accross ABC Codes and there are judgment calls required.

CHAPTER

THREE

DELIVERY LOCATION

U.S.A. vs France

CHAPTER

FOUR

LEAD TIME BY VENDOR

4.1 Overview

Vendors may have drastically different lead times.

4.2 Acquiring

Data for lead times should be derived from the latest vendor quotes that is:

Note that item_nbr, the item surrogate key is not used, allowing you to get vendor quotes for items that have not been set up;

create view ic_item_vnd_lead_time as select item_cd_qte,

vq_qte_dt, vq_qte_eff_dt, max(vq_qte_exp_dt) max_qte_exp_dt, (max(vq_qte_exp_dt) - vq_qte_eff_dt) / 7 lead_tm_wks

from vq_qte_vw group by org_nbr_vnd,

item_cd_qte, vq_qte_eff_dt, vq_qte_dt;

4.3 YAML

ic_item_vnd_lead_tm:

sql: >

create view ic_item_vnd_lead_time as select item_cd_qte,

vq_qte_dt, vq_qte_eff_dt, max(vq_qte_exp_dt) max_qte_exp_dt, (max(vq_qte_exp_dt) - vq_qte_eff_dt) / 7 lead_tm_wks

from vq_qte_vw group by org_nbr_vnd,

item_cd_qte, vq_qte_eff_dt, vq_qte_dt;

description: Create ic_item_vnd_lead_tm narrative: >

TODO describe consequence of using the quote expiration date, vq_qte_exp_dt rather than the quote of quotation or the effectivedate of quotation.

4.4 Assumptions

The most recent vendor quote for lead times will be used.

If multiple vendor quotes exist for the same quote date, the maximum lead time will be used.

Full historical vendor quotes are useful to see trends in cost and lead time.

4.5 Issues

Vendor on-time performance.

TODO flesh out

CHAPTER

FIVE

SIMULATION

Planning simulation

CHAPTER

SIX

QUESTIONS

Vendor Quotes

Customer Quotes

Base Currency

Supply pools implementation in US and French operations.

How many distinct part numbers?

How many Customer and vendor quotes?

Sales volumes for US and France, number of sales and dollars per operation per year.

Describe your current kitting for both operations.

Describe supply eligibility features and deficiencies for both operations.

Describe project approval functional, technical, financial.

Post implementaton functional specifications consulting is time and materials, coding desing,coding and documetation is fixed price.

CHAPTER

SEVEN

DATA LOADING

Base currency

Acceptance criteria

Funding

Accept base functionality

Accept willingness to extend existing functionality

Accept ability to contribute to functional requirements for extensions

Define minimal viable product.

Accept that there is no commercial “off the shelf” product that will satisfy Align needs and desires.

Accept that in-house development cost and risk is far higher than licensing Diamond APS, which has near 25 years of incorporation of distributor requirements.

Accept that there is low risk associated with application support.

Address current promise date for open orders is before the planning date, that is, the purchase order is past due.

Acknowledge that we provide valuable assistance in defining user requirements, that this experience vastly exceeds consulting experience from other consulting companies.

Reject other consultants that state *SMOP* Simply a matter of Coding without demonstrated ability to actually deliver a high performance, *non-kludgy*, supportable solution.

Accept that an optimal solution involves a general issue and that we provide very valuable service in defining detailed functional requirements and a clear technical implementation plan.

CHAPTER

EIGHT

INTEGRATION

US operations and Frenc operations should continue with no system modifications.

The ability to used Diamond facility transfers will be supported by intra-company purchases.

CHAPTER

NINE

TECHNOLOGY

9.1 Java vs PHP

9.2 Postgres vs mysql

9.3 Spring MVC vs Angular

9.4 Stack

java open_jdk1.8 postgres 10 order later maven git AWSEC2 Linux Redhat 8 Tomcat Node.js and Angular 8

CHAPTER

TEN

INDICES AND TABLES

- genindex
- modindex
- search