

QTG Postgres Migration



































Presented By
Bala D



Migration Checklist

- Understand and follow the data migration best practices.
- Analyze the QTG tables moved over to PostgreSQL .
- Analyze legacy tables associated with QTG tables (like Joins)
- Analyze the impacted applications.
- Compare DB2 vs PostgreSQL performance
- Analyze the db2 vs PostgreSQL compatibilities. (like blob would not be supported in PostgreSQL).
- Analyze PostgreSQL performance tuning checklist .
- Analyze data migration tool to move data from DB2 to PostgreSQL .
 - [Informatica, Aqua Data Studio, Etc]
- Analyze QTG codebase to isolate the legacy tables access logic.
- Analyze the content/document management tool to manage pdf docs.
- How to verify/validate the data migrated from DB2 to PostgreSQL?

DB2Vs Postgres Comparision

Name	DB2 	PostgreSQL 
Description	Common in IBM host environments, 2 different versions for host and Windows/Linux	Based on the object relational DBMS Postgres 
Database model	Relational DBMS 	Relational DBMS 
DB-Engines Ranking  Trend Chart 	Score 206.81 Rank #6 Overall #5 Relational DBMS	Score 282.13 Rank #5 Overall #4 Relational DBMS
Website	ibm.com/software/data/db2	www.postgresql.org
Technical documentation	www.ibm.com/support/entry/portal/documentation/software/information_management	www.postgresql.org/docs/manuals
Developer	IBM	PostgreSQL Global Development Group 
Initial release	1983 	1989 
Current release	DB2 Data Server (10.5), April 2013	9.4.5, October 2015
License	commercial 	Open Source 
Database as a Service (DBaaS) 	no	no
Implementation language	C and C++	C
Server operating systems	Linux Unix Windows z/OS	FreeBSD HP-UX Linux NetBSD OpenBSD OS X Solaris Unix Windows
Data scheme	yes	yes
Typing 	yes	yes
Secondary indexes	yes	yes
SQL	yes	yes 
APIs and other access methods	JSON style queries  XQuery ADO.NET JDBC ODBC	native C library streaming API for large objects ADO.NET JDBC ODBC
Supported programming languages	C C# C++ Cobol Fortran Java Perl PHP Python Ruby Visual Basic	.Net C C++ Java  Perl Python Tcl
Server-side scripts 	yes	user defined functions 
Triggers	yes	yes
Partitioning methods 	Sharding 	no, but can be realized using table inheritance 
Replication methods 	yes 	Master-slave replication 
MapReduce	no	no
Consistency concepts 		Immediate Consistency
Foreign keys 	yes	yes
Transaction concepts 	ACID	ACID
Concurrency 	yes	yes
Durability 	yes	yes
In-memory capabilities 		no
User concepts 	fine grained access rights according to SQL-standard	fine grained access rights according to SQL-standard

For more details refer - <http://vschart.com/compare/ibm-db2/vs/postgresql>



High Level Migration Strategy

- Isolate the QTG codes which access the legacy data (Use multiple datasource now. In future , legacy codes will be moved to legacy service.)
- Replicate legacy data in PostgreSQL which are tightly referenced with QTG tables using event architecture.
- Copy DB2 QTG Dev Data to PostgreSQL for functional testing in DEV environment.
- Copy DB2 QTG stage data to PostgreSQL for performance testing in STAGE environment.
- Compare performance with DB2 baseline. (loop through performance tuning cycle until get expected results).
- Go-Live : Copy the DB2 production data to PostgreSQL during the implementation window (~4hrs for data copy).
- Back Out Plan :
 - Get the script ready to copy the data from Postgres to DB2.
 - Revert the QTG application back to DB2 version.



Why Migrate to PostgreSQL

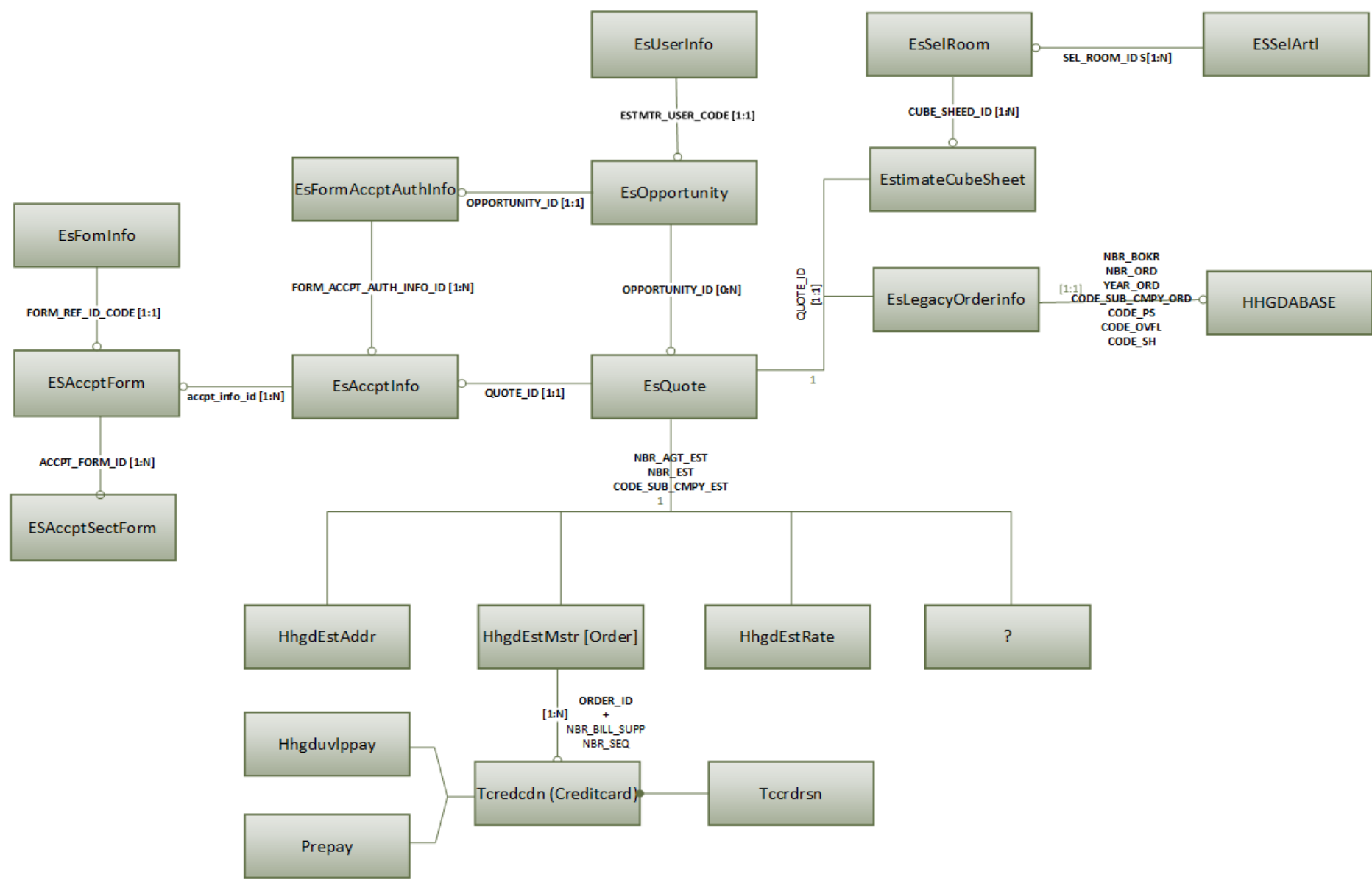
- Reduce Operating Costs
- Inline with enterprise architecture changes.



Review QTG and Legacy Tables.

- Refer spread sheet.
- Total Tables : 183 , QTG Tables : 108 , Legacy Tables : 75
- Opportunity total count -> ~3 million as on 7/15/2015
- Quote total count -> ~2.6 million as on 7/15/2015

QTG Data Model





Impacted Applications

- Estimates-UI
- Estimates-Service (Internal Soap Service)
- ExternalEstimatesWebservice
- EstimatesSurveyNativeWebservice
- QTG-Batch (Pushmail & Qtg Jobs)
- QTG-Service (Mysites , Pushmail & OIS)
- ETL Batch Job. (Informatica data replication job)
- PRIC – Mainframe program.
- Any other applications???



High Level Migration Tasks

- DBA
 - Create DB script to copy qtg schema from DB2 to PostgreSQL.
 - Create DB script to copy data from PostgreSQL to DB2. (Back out plan)
 - Update Informatica job to copy the QTG tables from PostgreSQL and Legacy tables from DB2 to ETL database.
- QTG Team
 - Isolate the Legacy data access code from existing code base.
 - Support multi data source to connect to both DB2 & PostgreSQL.
 - Handle Postgres compatibility code changes.
 - Update impacted db2 applications to connect to PostgreSQL.
 - Take care of functional and performance testing.

Continued....



High Level Migration Tasks

- Handle the implementation on different environments.
- Develop legacy event listeners to replicate legacy tables in PostgreSQL.
- Develop utility to convert the blob data to byte array.
- Generate legacy event message on save/update from QTG.
- Revisit ETL Data mart aggregation batch job ???
- Mainframe
 - Generate Event Message on save/update for legacy tables Master, Address & Rate to replicate the data between DB2 and Postgres to take care of joins functionality.
- Tech Service
 - Configure required Database servers for different environments.
 - Handle the load balance and fail overs.
 - Configure required monitoring system in place.



PostgreSQL Configuration Requirements

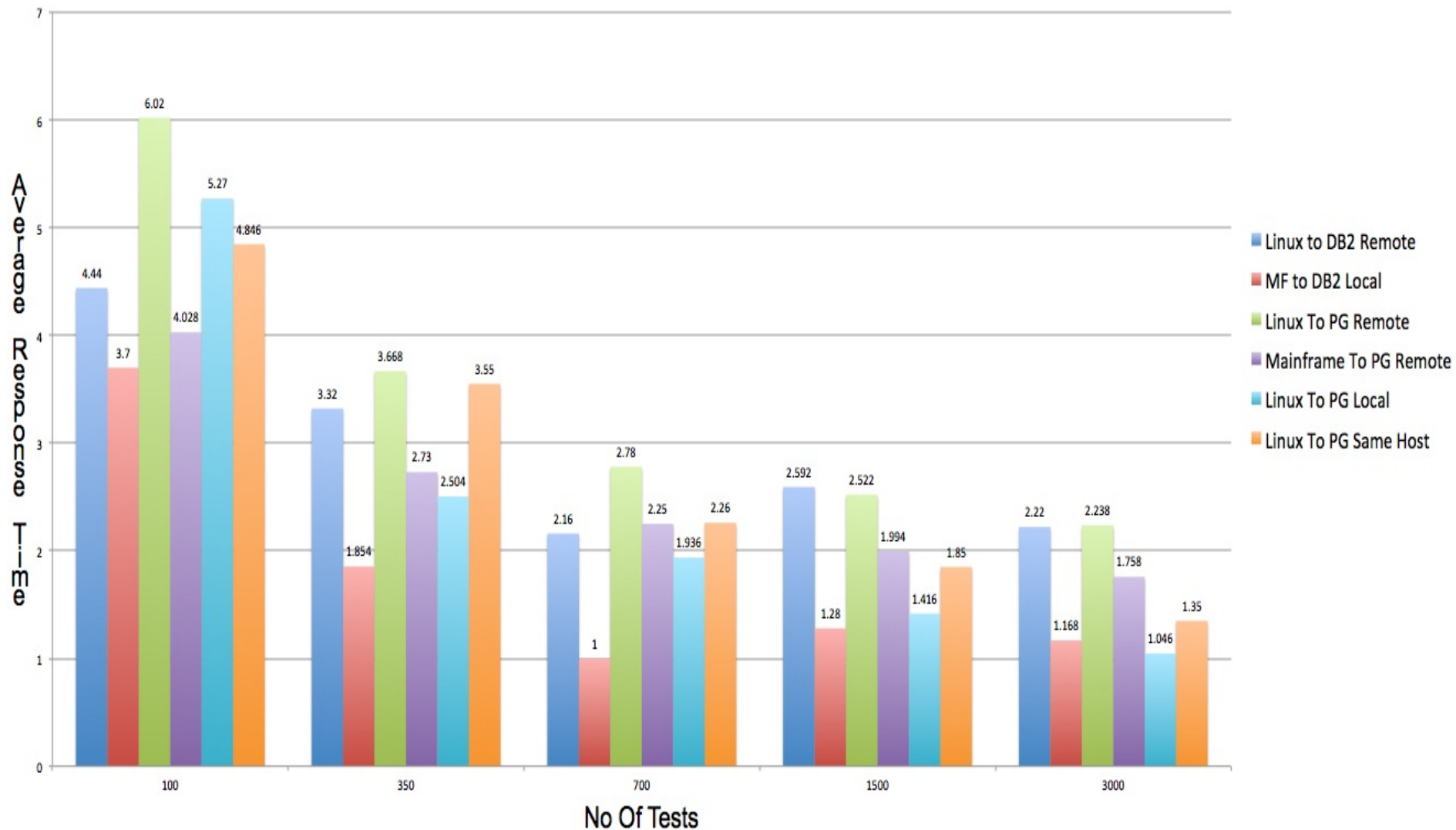
- Linux Server Configuration – no of cpus, memory, etc??
- Disk Storage?
- Data redundancy policy ?
- Any table partitions required?
- Database load balance and fail over strategy?
- Data backups.
- Revisits table indexes?



DB2Vs Postgres Load Test Comparision

- Will come soon..... ☺

JDBC Test
Mainframe/Linux Vs Db2/Posgres Performance Analysis
Linux : deva542a, Postgres : devd542a
Mainframe : UG04, DB2: UG04:DBT



JDBC Java Test

Linux Vs Mainframe & DB2 Vs Postgres

Linux_To_Db2_Remote

Vs

MF_To_Db2_Local

Linux 63% Slower

Linux_To_Pg_Remote

Vs

MF_To_Pg_Remote

Linux 35% Slower

Linux_To_Pg_Local

Vs

MF_To_Db2_Local

Linux 35% Slower

Linux_To_Pg_Remote

Vs

Linux_To_Pg_Local

Remote 43% Slower

Linux_To_Pg_Remote

Vs

Linux_To_Pg_Remote_Same Host

Same host 24% Faster

Migration Code Changes

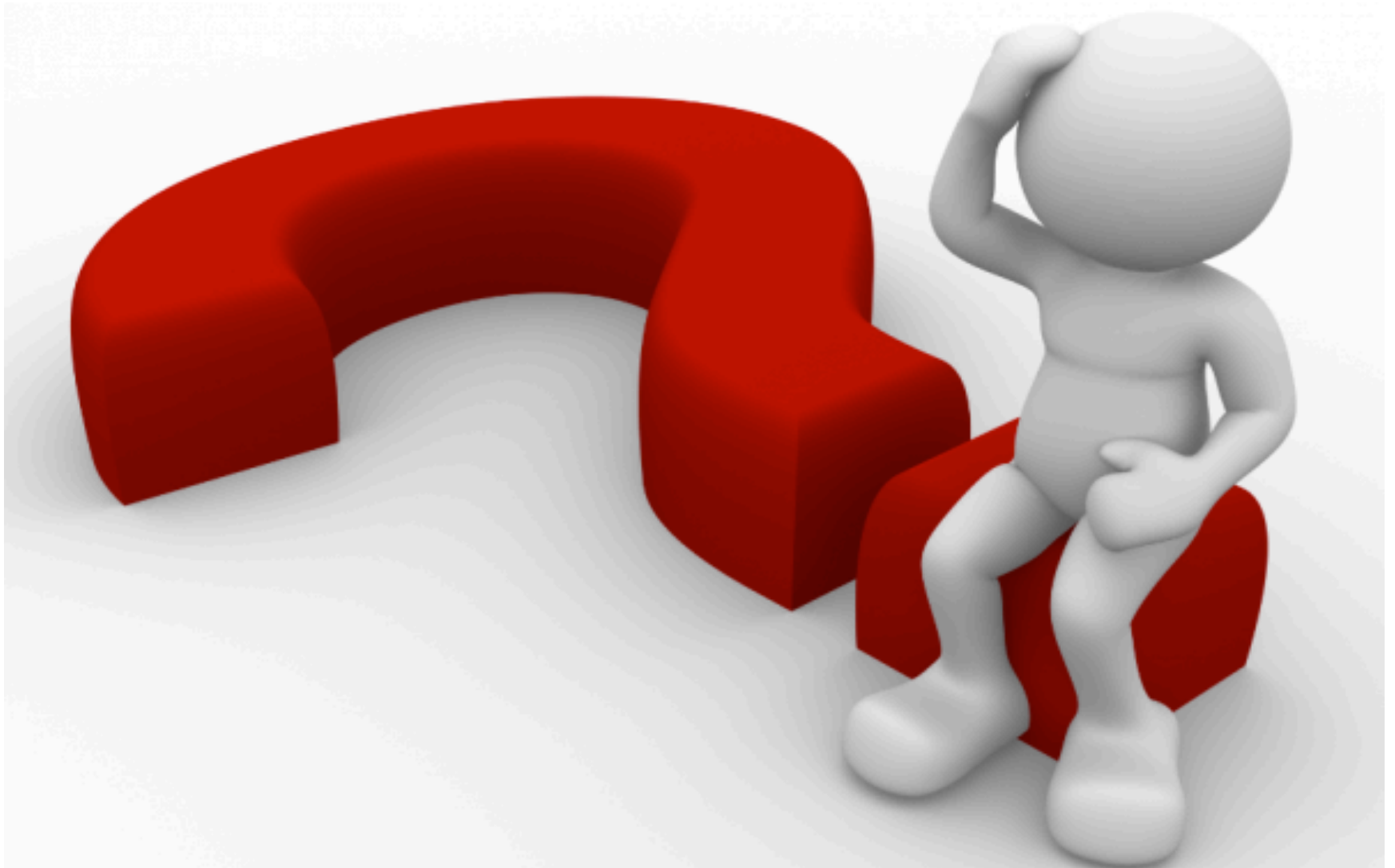
- Refactor QTG code base to support multiple data sources. Isolate legacy data access code. (change in spring configuration, hbm files and java files).
- Count Data Type - > returns BigInt instead of Integer
 - OpportunityManager.java, TaskManager.java, LeadsManager.java, EstimatesManager : use getCountAsInteger()
- Keyword 'Current Date' to be changed to current_date
 - EstimatesManager.java
- 'With ur' would not work in Postgres
 - EstimatesManager.java, LeadsManager.java, VoliManager.java
- Table Alias name , Keyword 'USER' -> Would not be supported in Postgres as a keyword. TaskManager.java
- With clause would not work.
 - VoliManger: retrieveVoliFormData();

Continued...



Migration Code Changes

Questions & Answers





Thanks!

