ETL Review Document

24/6/2014

Document revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Comments | Author |
| 1 | 26/6/2014 | Initial document | Venu |
| 2 | 27/6/2014 | Revision | venu |
|  |  |  |  |
|  |  |  |  |

TABLE OF CONTENTS

[1 Introduction………………………………………………………………….4](#_Toc368045125)

1.1 Purpose of this Document………………………………………………………………………4

1.2 Socpe of this documents……………………………………………………………………….4

1.3 Intended Audience…………………………………………………………………………………4

[2 Project rEview……………………………………………………………….5](#_Toc368045129)

2.1 ETL Architechture Diagram……………………………………………………………………4

2.2 ETL DB Schema………………………………………………………………………………………5

2.3 Facts and Dimensions…………………………………………………………………………….5

2.4 SCD's……………………………………………………………………………………………………….6

2.5 OLTP-OLAP sync……………………………………………………………………………………6

2.6 CSV Imports and data types……………………………………………………………….6

2.7 Batch Jobs…………………………………………………………………………………………….7

2.8 Naming conventions…………………………………………………………………………….8

2.8.1 Talend Naming conventions………………………………………………………………8

2.8.2 SQL Naming conventions………………………………………………………………….9

2.9 SQL coding standards………………………………………………………………………….9

2.10 Social networks job…………………………………………………………………………….10

2.11 Semantria Job…………………………………………………………………………………….10

**3 ISSUES AND CORRECTIONS…………………………………………………11**

1. **INTRODUCTION**
   1. Purpose of this document

The purpose of this document is to review the spotcrowd analytics design specifications and development process.

* 1. Scope of this document

The document provides details about how the ETL process is used to move data from source to target.

The ETL process is build over two different data sources. The first source is web scraping data using specific data tools and dumping the data into staging. The data is then moved to core database.

The second source of data is oltp application which uses webforms to capture data. This data is moved from oltp database to core databse in DWH.

* 1. Intended Audience

1. Developers.
2. Production Support.
3. **Project Review**
   1. ETL Architecture diagram

**ETL (Talend)**

Extraction,

Transformation,

Loading

CSV

Core

Access

Reporting

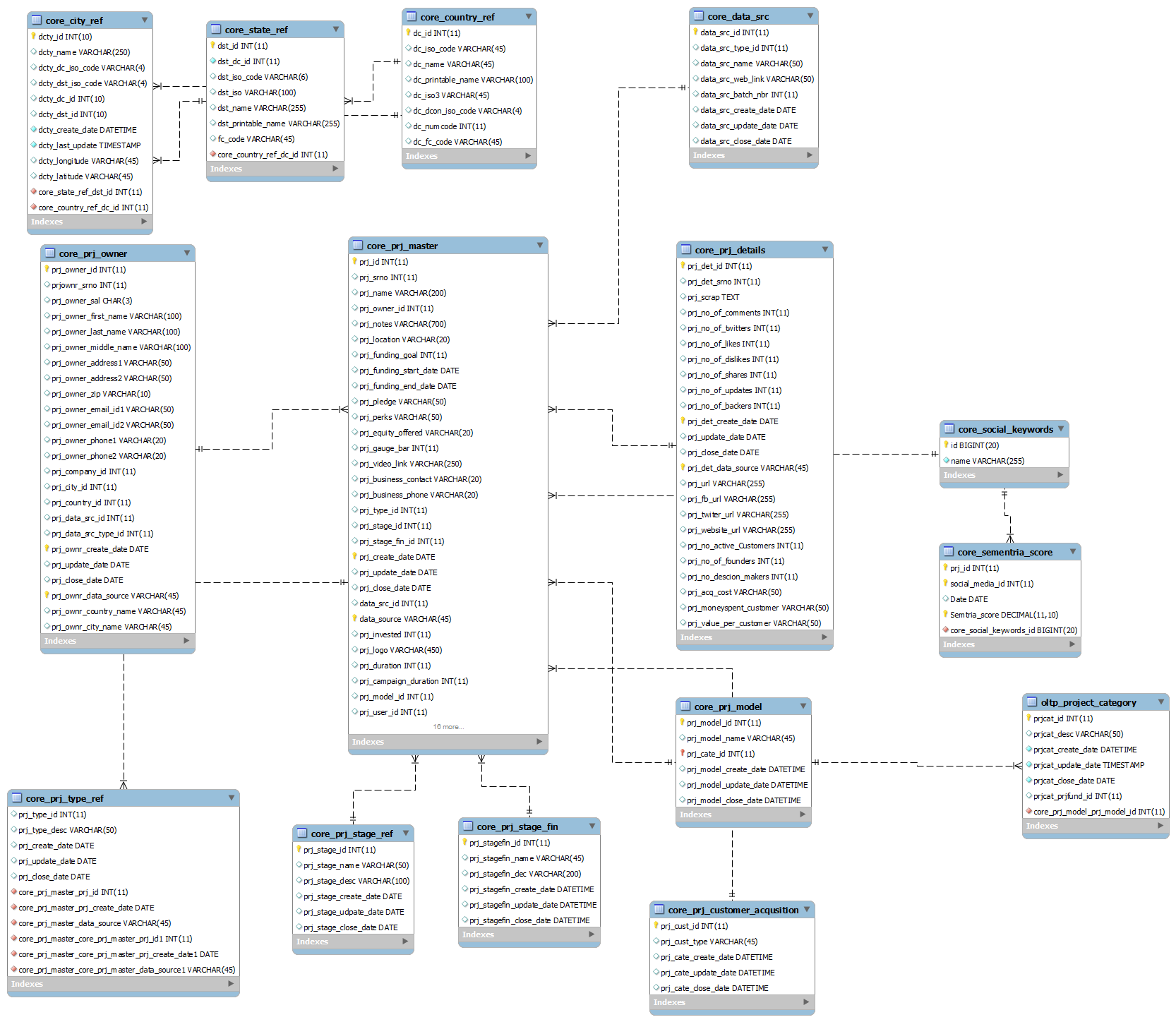
OLTP

Stg

The above architectural diagram defines the data flow in spotcrowd. CSV files are processed using talend and loaded into the sp\_core database in the data warehouse. Secondly, data captured using webforms is loaded into OLTP system. The data from OLTP is then processed using talend and loaded into sp\_core database in data warehouse.

* 1. ETL DB Sechema

Below is the database schema used for the spotcrowd project. The schema is in sp\_core database and is a OLAP system of spotcrowd.



* 1. Facts and Dimensions

Below are the list of facts and dimensions in the OLAP database of spotcrowd.

Facts:

1. Core \_prj\_master

Dimensions:

1. Core \_prj\_owner
2. Core\_prj\_details
3. Core\_social\_keywords
4. Core\_semantria\_social
5. Core\_prj\_model
6. Oltp\_project\_category
7. Core\_prj\_type\_ref
8. Core\_prj\_stage\_ref
9. Core\_prj\_stage\_fin
10. Core\_prj\_customer\_acqusition
11. Core\_data\_src
12. Core\_country\_ref
13. Core\_state\_ref
14. Core\_city\_ref
    1. SCD

Two SCD’s have been identified in the present OLAP system

1. Semantria score - This is maintained as a type 3 SCD
2. Project stage name – This is also maintained as a type 3 SCD
   1. Database sync

Database sync is done in two ways OLAP-OLTP and OLTP-OLAP.

Below are tables which are synchronized from OLAP-OLTP

1. Project master (core\_prj\_master - project\_master)
2. Project owner (core\_prj\_owner – project\_owner)
3. Project details (core\_prj\_details – project\_details)

Below are tables which are synchronized from OLTP-OLAP

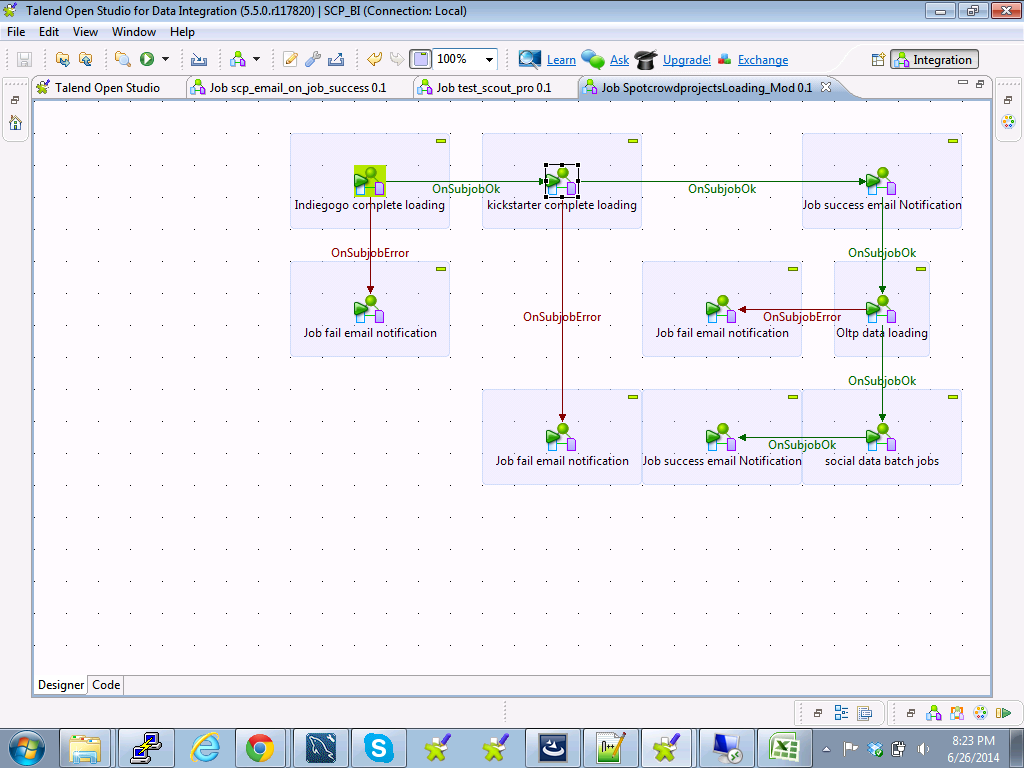
1. User table (scf\_user – core\_scf\_user)
2. User activity table ( scf\_user\_activity – core\_scf\_user\_activity)
3. Semantria score table (tbl\_sementria\_score – core\_sementria\_score)
4. Business plan denormalization (bp\_den – bp\_den)
5. User denormalization (scf\_den- scf\_den)
   1. CSV imports

The CSV imports are captured by running perl scripts maualy as of now. These jobs need to be scheduled as batch file in future. Below is the list of scripts run to get CSV files

|  |  |  |
| --- | --- | --- |
| Location | Script name | frequency |
| C://perlproject (63.134.216.168) | Kicks.pl | NA(run manually after each cycle) |
| C://perlproject (63.134.216.168) | Indigo.pl | NA(run manually after each cycle) |

* 1. Batch Jobs

Below is the screenshot of the talend main job for spotcrowd project. The job has all the sub jobs which are initiated in a sequence.



2.7.1 Sequence

Below is the list of jobs in sequence which are executed in talend to extract, transform and load data into database.

1. FTP\_loading
2. SourceToStageLoading\_KickStarter
3. SourceToStageLoading\_Indiegogo
4. TempProjectLoading
5. scp\_stg\_core\_loading\_0.1
6. scp\_stg\_core\_loading\_0.2
7. scp\_stg\_core\_loading\_0.3
8. scp\_oltp\_prj\_loading\_0.1
9. scp\_oltp\_prj\_loading\_0.2
10. scp\_oltp\_prj\_loading\_0.3
11. scp\_oltp\_keywords\_loading
12. scp\_social\_data\_batch\_jobs
13. scp\_email\_on\_job\_success
14. scp\_email\_on\_job\_fail

2.7.2 Scheduling

The batch job ‘SpotCrowdProjectsLoading’ is scheduled to run every three days. The three days time frame is set due to the scripts which are used to screen scrap data usually take three days to return a data file.

Perl scripts used to screen scrap data are run manually and presently takes three days to return an input CSV file.

2.7.3 Logs

The logs for talend job execution is stored in ‘sp\_stage’ in logscatcher table on 63.134.216.168 server.

2.7.4 Exception Handling

Exceptions in talend are handled by the java script generated with components. In the present situation exception raised are captured in a logcatcher table, on sub-job error. A mail is automatically generated along with termination of job.

* 1. Naming Conventions
     1. Talend naming conventions

Guidelines for developers for ETL naming conventions. We will be using PascalCasing or camelCasing for as naming conventions in Talend.

1. Package Name Conventions

It is suggested to have naming conventions for ETL packages as the number of packages grows, to manage the packages easily. The following naming convention should be applied to package names

**[PackageName].[DestionDatabaseName].[SubjectName].[TableName]**

All the names use PascalCasing.

1. Tasks/Components naming conventions

Usually the best practice is to start the name with acronym. This approach will help to easily identify the tasks/components at the debug-time and run-time. The acronym should be used at the start of the name

1. Variable Names

Name convention for variables should be PascalCasing.

Eg:

SourceConnectionString

TargetConnectionString

LogConnectionString

1. Scripts

Talend supports java scripts. The script is used across talend using a script component. The following table summarizes the capitalization rules for identifiers and provides examples for the different types of identifiers.

|  |  |  |
| --- | --- | --- |
| Identifier | Case | Example |
| Class | Pascal | AppDomain |
| Interface | Pascal | IDisposable |
| Namespace | Pascal | System.Drawing |
| Method | Pascal | ToString |
| Parameter | Camel | typeName |
| Property | Pascal | BlackColor |
| Exceptional class | Pascal | WebException |

* + 1. Sql Naming Conventions

All Sql keywords must be upper case. All declared variables must be Camel case while all stored procedure names,function names,trigger names, Table names and column names in query must be in Pascal case.

All view names must start with letter ‘v’ followed by the name of the view in pascal case.. All table names must be singulars.

If the table belongs to specific schema/module, it should be appended with a acronym of the module/schema

* 1. Sql coding standards

1. Create indexed only on tables that have high query pressure for select statements. But indexed should not be created on tables where CURD operations are used frequently. An index speeds up the selection if index is used in the select statement in where clause. The same index will sow down the insert statement if used in the statement or not as the indexes readjust everytime the structure of the table is altered.
2. Primary Key is a field or a set of fields in the database table that uniquely identify records in the database table. A table can have only one primary key. The naming conventions for primary key constraints should have a "PK\_" prefix, followed by the table name. The syntax should be "PK\_<TableName>".
3. Foreign Key is a field in the database table that is primary key in other table. The naming conventions for a foreign key constraint should have a "FK\_" prefix, followed by the target table name, followed by the source table name. The syntax should be "FK\_<TargetTable>\_<SourceTable>".
4. Unique key is a set of one or more fields/columns of a table that uniquely identify a record in database table. It is like Primary key but it can accept only one null value. The naming conventions for unique key constraints should have a "UQ\_" prefix, followed by the table name, followed by the column name. The syntax for a unique constraint should be "UQ\_<TableName>\_<ColumnName(s)>".
5. Try to avoid wildcard characters at the beginning of a word while searching using the LIKE keyword, as that result in an index scan, which defeats the purpose of an index. The following statement results in an index scan, while the second statement results in an index seek
6. Avoid searching using not equals operators (<> and NOT) as they result in table and index scans.
7. Use derived tables whenever possible as they have high query performance.
8. Use the more readable ANSI-Standard Join clauses instead of the old style joins. With ANSI joins, the WHERE clause is used only for filtering data. Whereas with older style joins, the WHERE clause handles both the join condition and filtering data.
9. Try to avoid server side cursors as much as possible. Always stick to a ‘set-based approach’ instead of a ‘procedural approach’ for accessing and manipulating data. Cursors can often be avoided by using SELECT statements instead.
10. Do not prefix your stored procedure names with “sp\_”. The prefix sp\_ is reserved for system stored procedure that ship with SQL Server. Whenever SQL Server encounters a procedure name starting with sp\_, it first tries to locate the procedure in the master database, then it looks for any qualifiers (database, owner) provided, then it tries dbo as the owner.
11. Incorporate your frequently required, complicated joins and calculations into a view so that you don’t have to repeat those joins/calculations in all your queries. Instead, just select from the view. Views are also used to restrict access to the base tables by granting permission only on views.
12. Always access tables in the same order in all your stored procedures and triggers consistently. This helps in avoiding deadlocks. Never, ever wait for user input in the middle of a transaction.
13. Offload tasks, like string manipulations, concatenations, row numbering, case conversions, type conversions etc., to the front-end applications if these operations are going to consume more CPU cycles on the database server. Also try to do basic validations in the front-end itself during data entry. This saves unnecessary network roundtrips.
    1. Social networking job

The social networking job is a batch job executed using talend to obtain keywords for projects. The keywords are dumped into OLTP and copied to OLAP system.

|  |  |  |
| --- | --- | --- |
| Talend Job name | frequency | Server |
| scp\_social\_data\_batch\_jobs (batch job) | Daily | 63.134.216.168 |

* 1. Semantria Job and Data Updates

The semantria job is a batch job executed using talend to obtain semantria score for the projects. The score is dumped into OLTP and copied to OLAP system.

|  |  |  |
| --- | --- | --- |
| Talend Job name | frequency | Server |
| scp\_social\_data\_batch\_jobs (batch job) | Daily | 63.134.216.168 |

1. **Issues**
2. The pearl scripts for screen scraping are taking three days to return a source file. This may increase in future, if the number of projects in the data source increases.
3. There are no proper relationships between tables in database. Missing foreign keys in tables. Need to correct soon as this will cause error while inserting and updating data. Below is the list of missing foreign keys

|  |  |
| --- | --- |
| Table name | Missing Keys |
| sp\_core.core\_prj\_owner | prj\_company\_id |
|  | prj\_city\_id |
|  | prj\_country\_id |
|  | prj\_data\_src\_id |
|  | prj\_data\_src\_type\_id |
| sp\_core.core\_prj\_master | data\_src\_id |
|  | prj\_type\_id |
|  | prj\_stage\_id |
|  | prj\_stage\_fin\_id |
|  | data\_src\_id |
| sp\_core .core\_city\_ref | dcty\_dc\_id |
|  | dcty\_dst\_id |
| sp\_core .core\_state\_ref | dst\_dc\_id |