2022-10-21 Meeting Agenda

Date

21 Oct 2022

Attendees

- Nathan Southern
- Jeff Braswell
- peter antley
- Jenny Tornquist
- Bourjali Hi
- Dale Harris
- Crystal Lewis
- Brian Hoffman
- Libby Crews
- Mike Nurse
- Ron Camarota
- Susan Chudwick
- Reggie Scarpa

Antitrust Policy Notice

Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

openIDL

This is a weekly series for The Regulatory Reporting Data Model Working Group. The RRDMWG is a collaborative group of insurers, regulators and other insurance industry innovators dedicated to the development of data models that will support regulatory reporting through an openIDL node. The data models to be developed will reflect a greater synchronization of data for insurer statistical and financial data and a consistent methodology that insurers and regulators can leverage to modernize the data reporting environment. The models developed will be reported to the Regulatory Reporting Steering Committee for approval for publication as an open-source data model.

openIDL Community is inviting you to a scheduled Zoom meeting.

Join Zoom Meeting
https://zoom.us/j/98908804279?pwd=Q1FGcFhUQk5RMEpkaVlFTWtXb09jQT09

Meeting ID: 989 0880 4279
Passcode: 740215

One tap mobile

Minutes:

I. Introductory Comments by Mr. Antley

A. Reading of LF Antitrust Statement

B. Welcome to All Attendees.

II. Agenda

A. Updates - for AWG and RRDMWG (high level)

1. In last week, switch to PostgreSQL. Data engineer has been onboarded - he's working on loss records, Mr. Antley working on Premium and ETL to load Premium records. Table set up. Function distinguishing between dates (SQL server function).
2. Two queries for calculating EP based on work over the summer - laid foundation for a very quick transition
3. Observations
   a. We originally talked about utilizing year/month/day. The team didn't want to put a timestamp, but to keep things simple. However on this database engine, that creates a problem. Mr. Antley worked with this and couldn't get it to drop timestamp aspect. Mr. Antley asked if having extra precision on dates will cause issue. (Unsure about the exact source of this issue, i.e., auto add of timestamp). No concerns presented. Mr. Antley will discuss in AWG Monday. Mr. Harris noted the variable we get today is 3 digits - month month year. Asked how this will be transformed into date. Mr. Antley: AAIS handles this by assuming the 15th (mid-month accounting), and effectively adds a level of precision. For now everything will stay the same.
   b. 2b - numeric for data type. Car years - exposure x months covered / 12. Calculation for car years requires some division, auto generates four decimal places. Mr. Antley asked if team wants to do fixed decimal places. Are 4 decimal places enough? Should we cap everything off to round dollars/round #s. Mr. Braswell pointed out that this may be more of a formatting issue, or a question of how it is represented; asked if it is just a floating point #. Fine as long as numeric translates to a double. Mr. Madison: per PostgrSQL manual it is not a floating point, but a user-specified precision. Mr. Braswell: level of precision depends on how/for what fraction will be used, in terms of how much it impacts the end line #s.
   c. Mr. Madison: if we agree that we maintain the highest level of precision as long as possible into the processing, this is a solid guideline. Only downshot w/higher precision is cost. Mr. Madison advocated taking the performance and storage drawbacks to store it more exactly and avoid major end-line aberrations.
   d. Mr. Madison: it's a business question to decide what level of precision we want to use (4th place, 10th place). 4 places feels comfortable. Simply rounding to penny is probably inadequate.
   e. **Group decided to run decimals to six places as a fault as a gen. rule. JM: if we start getting into more complex models than flat and wide we will have a conversation about real calculations. (E.g., extensions of exposures across 5 years). Our base trans. fine grain data is numeric, we favor this because it's human-consumable. If we are in a calculation situation that is difficult, we can favor double-precision. Numeric 6 should be our default, because reel only handles 6 anyway. Complex math and double precision for special circumstances (e.g., 15 digits)**
4. In Mongo: we're working with JavaScript. Mongo & Javascript do Pascal and Camelcase to make it easier to pass keys & have similar keys. Version of postgreSQL... When basic database is stood up through AWS, function names and table names are not case sensitive (standard in SQL). Going w/best practices: using underscores between the words, having to translate from underscores in the data layer to camel and pascal case in the JS layer is what we want to do. This is primarily an AWG question. Our column names moving twd underscore notation. (JM: Yes, they have to)
5. Test Data Set
   a. Review of last week: we got mongo vs. excel outstanding to about a 1.4% variation in outstanding loss.
   b. With PostgreSQL, we have not finished the loss record load as of now. PA: plan before trying to seal 1.4% variation on excel vs. mongo is to calculate what are differences between example set Mr. Harris produced and what's in PostgreSQL already.
   c. PA: data set given to Dale not as complete as it could've been. For premium records: minimum record starts on day 1 of yr. 2000. Oldest premium record: from March 2000. In Lucid, the visual chart we made this summer, we see that calculating EP is 4 separate aggregations. 4 records in extraction period handled one way. Records before extraction period handled a different way. We're not earning premium before this date with Group 2. All 4 groups needed to have a valid test that EP is calculated correctly.
   d. We should try to enhance our premium record set by generating more records from year before and year after this. We may also want to generate records for the end of the year. PA: duplicating and slightly modifying records he already has, for testing out EP.
   e. Mr. Harris: bigger issue from a stat reporting perspective is that one doesn't have to calculate EP - because premium rec'd 2 years later will be equal to earned premium. Last year's, current year and year before are necessary to calculate EP.
   f. Mr. Harris: cancellation won't be received years later - will be within reporting period that we have.
   g. PA: initial focus will be on getting Group 1 to work, for car years, for earned premium, and for the loss in the claim count. Car years and incurred loss will require calculations.
   h. Lower priority to expand said records for earned premium.

III. Looking Ahead

A. Next Friday: PA will be presenting on car years and earned premium in PostgreSQL vs. expected values as provided by Mr. Harris. PA is onboarding Tsai who is working on loss records

B. Soon we will be getting into homeowners

GMT20221021-1...1920x1080.mp4

Goals

Discussion items

Time	Item	Who	Notes

Action items

Wrote

Cre

In

I

☐