Presto by Group LX

SQL Query Engine

Τι είναι;

- Query Engine που μπορεί να εκτελεί και να συνδυάζει queries από πολλαπλά data sources. Υποστηρίζει:
  - Hbase
  - Hive
  - Σχεσιακές βάσεις δεδομένων
- Αναπτύχθηκε από το Facebook

Τεχνικά χαρακτηριστικά

- Το Presto είναι γραμμένο σε Java και γίνεται build με Maven
- Γίνεται checkstyle check σε κάθε build
- Τρέχει automated test suite σε κάθε build

Συνεισφορές

- Προσθήκη δυνατότητας auto-complete εντολών στο presto-cli (πχ. SELECT, SHOW, EXPLAIN, USE κτλ.)
- Προσθήκη auto-suggestions για τυπικά misspellings στα error messages πχ. Για την εντολή “SELE FROM foo” εμφανίζει: Did you mean “SELECT FROM foo”?

Αρχιτεκτονική presto-cli

- Η κονσόλα υλοποιείται χρησιμοποιώντας τη βιβλιοθήκη Jline
- Το Jline παρέχει tab-autocomplete μέσα από κλάσεις που υλοποιούν το interface jline.console.completer.Completer
- Η γραμματική της γλώσσας λειτουργεί με Antlr. Το κύριο αρχείο γραμματκής είναι το Statement.g

Υλοποίηση autocomplete

- Προστέθηκε η κλάση CommandCompleter που υλοποιεί το Jline ConsoleReader
  - Τελικά μετά από feedback της ομάδας του Presto η νέα κλάση αφαιρέθηκε και χρησιμοποιήθηκε η κλάση StringsCompleter
- Αλλάχθηκε ο constructor της κλάσης com.facebook.presto.cli.LineReader ώστε να δέχεται πολλαπλά Completers. Έγινε με varargs:
  - LineReader(History history, Completer... completers)

Auto-suggestions on failure

- Ροή:
  - Το query γίνεται tokenize
  - Δημιουργούμε πίνακα με όλα τα σωστά tokens ο οποίος εξάγεται από το αρχείο Statement.g
  - Υπολογίζουμε το Levenshtein distance από κάθε query token με κάθε σωστό token
  - Αν η αποστάση είναι <2 τότε κρατάμε το suggestion σε έναν πίνακα
  - Ταξινομούμε τον πίνακα κατά αύξουσα σειρά βάσει της απόστασης
  - Αντικαθιστούμε το query token με το πρώτο στοιχείο του πίνακα

Τρόπος εργασίας

- Δημιουργήσαμε ένα fork του presto
  - https://github.com/eantonopoulos/presto
- Το πρώτο contribution υλοποιήθηκε στο master branch του fork
- Το δεύτερο contribution υλοποιήθηκε στο autosuggest branch του fork

Έλεγχος

- Για την πρώτη συνεισφορά δεν απαιτήθηκαν προσθήκες στα tests καθώς όλες οι συμβαλλόμενες κλάσεις περιλαμβάνονται στα υπάρχοντα tests

- Για τη δεύτερη συνεισφορά προσθέσαμε unit tests για την κλάση QueryAutoSuggester. Τα tests περιλαμβάνουν ελέγχους για:
  - Σωστή εντολή
  - Incomplete εντολή (selec * FROM foo)
  - Κενή εντολή

Επικοινωνία με ομάδα

- Έγιναν δύο pull requests
- Για να μπορούν να γίνουν δεκτά χρειάστηκε να υπογράψουμε το Contributor License Agreement του Facebook
- Για την πρώτη συνεισφορά πήραμε feedback για επαναχρησιμοποίηση StringsCompleter και για χρήση varargs στον constructor της LineReader
- Για τη δεύτερη συνεισφορά περιμένουμε feedback

Ευχαριστώ πολύ και Καλό Καλοκαίρι