Ανάπτυξη Λογισμικού Πληροφορικής
Χειμερινό Εξάμηνο 2015-2016

“Σύστημα ανίχνευσης συγκρούσεων σε επερωτήσεις βάσεων δεδομένων”
Επίπεδο 2

Πίνακας Περιεχομένων

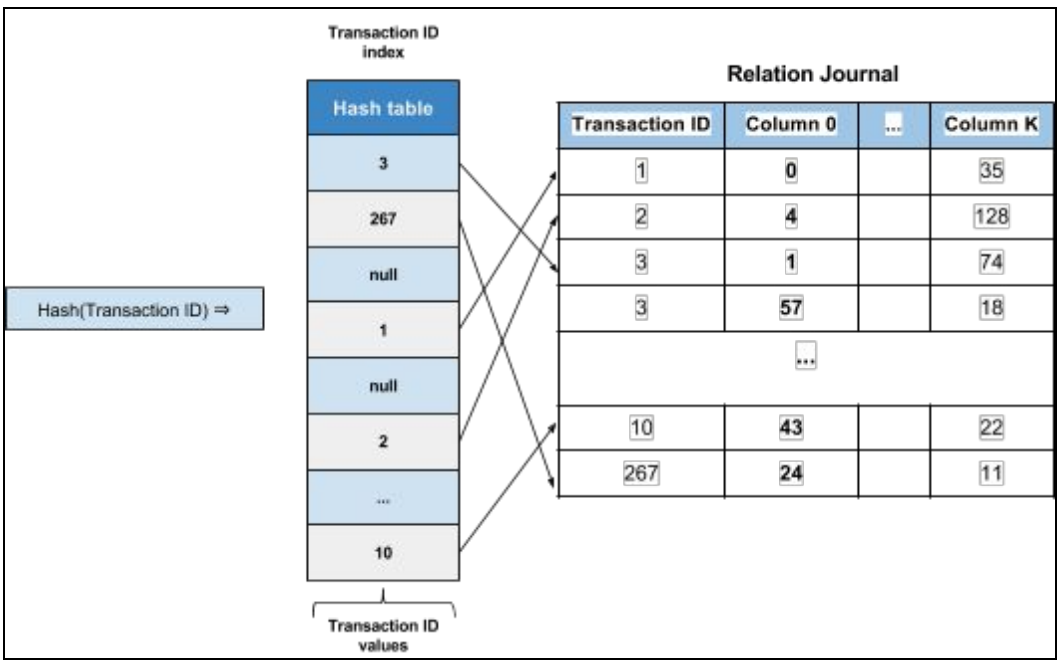
Γενική περιγραφή

Σε αυτό το επίπεδο θα αναπτύξετε επιπλέον ευρετήρια στην εφαρμογή σας, με σκοπό (1) να γίνεται άμεση αναζήτηση εγγραφών στα ημερολόγια των σχέσεων με βάση το πεδίο transaction id, (2) να βελτιωθεί η ταχύτητα στους υπολογισμούς των validations, στην περίπτωση όπου υπάρχουν επικαλύψεις στα predicates αυτών.

Περιγραφή παραδοτέων δεύτερου επιπέδου

1. Ευρετήριο στα Transaction IDs στο Ημερολόγιο μιας Σχέσης

Στο 1ο επίπεδο, όταν θέλαμε να εντοπίσουμε την 1η εγγραφή στο ημερολόγιο μιας σχέσης, με βάση το transaction id πραγματοποιούσαμε δυαδική αναζήτηση, εκμεταλλευόμενοι το γεγονός ότι τα ids αυτά είναι ταξινομημένα σε αύξουσα σειρά. Στο 2ο επίπεδο, θα χρησιμοποιήσετε ένα ευρετήριο επεκτατού κατακερματισμού, με στόχο να αποφευχθεί η δυαδική αναζήτηση, όταν αυτή απαιτείται. Το ευρετήριο αυτό θα δημιουργείται ανά σχέση, θα έχεις ως κλειδί το *transaction_id* και τιμή ένα *δείκτη (ή offset από την αρχή του ημερολογίου)* στην πρώτη εγγραφή του ημερολόγιο με το συγκεκριμένο transaction_id.



Λειτουργίες TransactionIndex

TransactionIndex createTransactionIndex();*

OK_SUCCESS insertTransactionIndexRecord(TransactionIndex, Key, JournalRecordOffset, …);*

JournalRecordOffset getTransactionIndexRecord(TransactionIndex, Key, …);*

OK_SUCCESS destroyTransactionIndex(TransactionIndex);*

2. Ευρετήριο στους όρους (predicates) των Validations

Το ευρετήριο αυτό αποθηκεύει τα predicates των validations που ανήκουν σε συγκεκριμένη σχέση. Συνεπώς, κάθε σχέση θα έχει το δικό της ευρετήριο, στο οποίο θα εισάγονται τα νέα predicates, που εμφανίζονται στα validations. Τα validations είναι αποθηκευμένα σε μια συνδεδεμένη λίστα, ενώ κάθε μεμονωμένο validation διατηρεί δείκτες προς τα predicates στα ευρετήρια των validations που μετέχουν αυτά.

Δημιουργία Ευρετηρίου

Καθώς έρχονται τα validations πρέπει να φορτωθούν τα predicates τους στα αντίστοιχα ευρετήρια των σχέσεων. Η συνάρτηση κατακερματισμού που αποφασίζει το κάθε predicate σε ποιο bucket του επεκτατού κατακερματισμού ανήκει (δηλαδή το κλειδί), πρέπει να λαμβάνει ως είσοδο τους όρους **[range start, range end, condition]**, τα οποία ορίζουν τη μοναδικότητα ενός predicate σε μια σχέση. Ως τιμή για κάθε εγγραφή στη συγκεκριμένη δομή κατακερματισμού, θα αποθηκεύεται μια δομή/κλάση ValidationPredicate η οποία θα αποθηκεύει πληροφορίες σχετικές με τη συνθήκη που εξετάζει, ένα πεδίο Conflict που αποθηκεύει το αποτέλεσμα (δηλαδή αν δημιουργεί σύγκρουση ή όχι) καθώς και μια λίστα που κρατάει δείκτες προς τα validations που ανήκει το predicate αυτό.

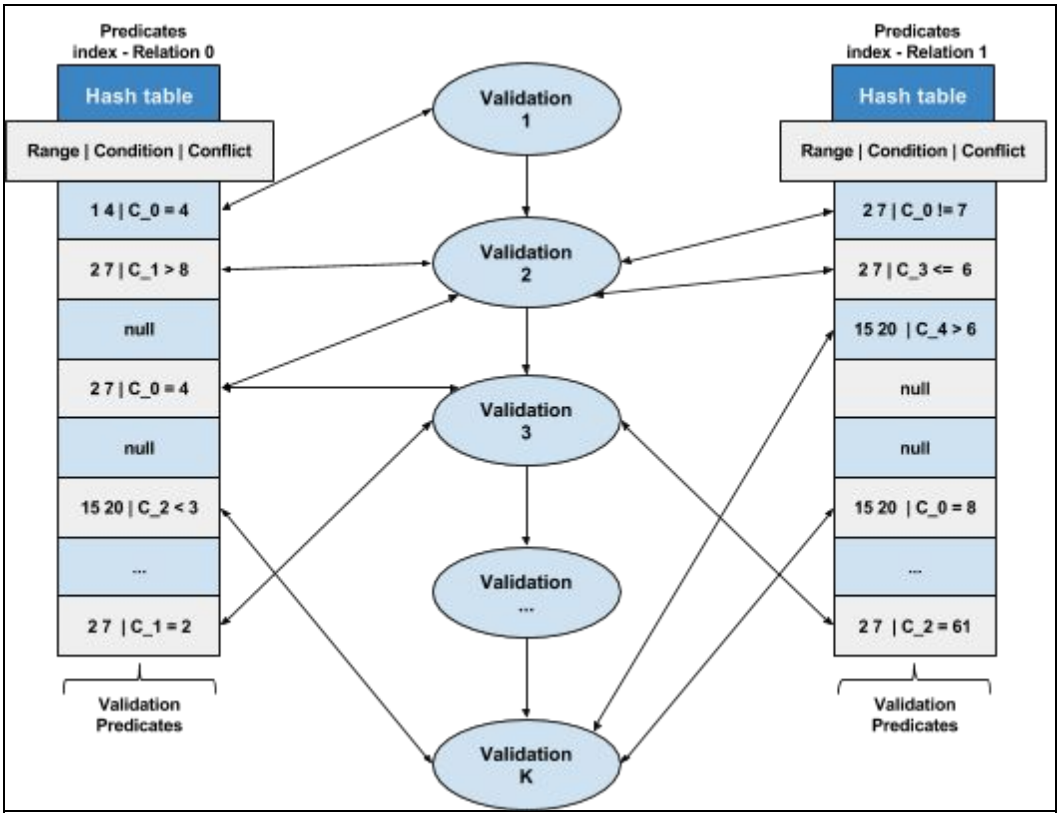
Παράδειγμα Εισαγωγής δεδομένων στο ευρετήριο

Έστω ότι υπάρχουν 2 σχέσεις και σε μια ριπή έρχονται τα ακόλουθα validations:

.

```
validation 1   1  4  [0 c0=4]
validation 2   2  7  [0 c1>8 c0=4] [1 c0!=7 c3<=6]
validation 3   2  7  [0 c0=4 c1=2] [1 c2=61]
                ...
validation k 15 20 [0 c2<3] [1 c4>6 c0=8]
```

Η κατάσταση των ευρετηρίων στα validations μετά την εισαγωγή τους παρουσιάζεται στο ακόλουθο σχήμα.



Υπολογισμός Αποτελεσμάτων

Όταν έρχεται η εντολή *flush*, τότε διατρέχουμε τη λίστα των validations μέχρι το validation id που απαιτείται να υπολογιστεί με τον εξής τρόπο. Για κάθε validation, προσπελαύνουμε τα predicates στα ευρετήρια των validations. Αν το predicate αυτό δεν έχει ακόμα υπολογιστεί, τότε το υπολογίζουμε και αποθηκεύουμε στο ευρετήριο μια τιμή True / False στο πεδίο Conflict, ανάλογα με το αν δημιουργεί σύγκρουση ή όχι. Σε αντίθετη περίπτωση απλά διαβάζουμε τη τιμή του πεδίου Conflict. Αφότου ένα validation λάβει τη τιμή ενός predicate, τότε πρέπει επιπλέον να διαγράψει το δείκτη προς αυτό από τη λίστα του ValidationPredicate. Κατά την προσπέλαση των predicates ενός validation, αν διαπιστωθεί ότι αυτό το validation δημιουργεί σύγκρουση, τότε προχωράμε στο επόμενο. Τέλος, τη στιγμή που ένα validation πάρει τη τιμή του, τότε πρέπει επίσης να διαγραφεί από τη συνδεδεμένη λίστα, αφού έχει υπολογιστεί.

Εντολή Forget

Η εντολή forget, δηλώνει ότι σε ένα διάστημα δοσοληψιών, δε θα υπάρξουν στο μέλλον validations, τα οποία κάνουν επερωτήσεις στο διάστημα αυτό. Με βάση τα παραπάνω, είναι πλέον δυνατό να μειωθεί δεσμευμένος χώρος από τα ευρετήρια στα validations των σχέσεων, διαγράφοντας predicates που δε χρειάζεται η τιμή τους.

Η διαγραφή ενός κλειδιού - predicate από ένα validation index μιας σχέσης και τελικά η αποδέσμευση μνήμης θα πραγματοποιείται κατά τον υπολογισμό ενός predicate του ευρετηρίου εφόσον ισχύουν οι εξής δύο συνθήκες:

1) Το στοιχείο που θα διαγραφεί (δηλαδή το validation id) από τη λίστα του ValidationPredicate να είναι το τελευταίο.
2) Το κάτω άκρο του transaction range (range start) του ValidationPredicate, να ανήκει σε διάστημα που έχει έρθει εντολή forget.

Όταν ισχύουν οι παραπάνω συνθήκες είναι ασφαλές να διαγραφεί από το ValidationIndex το κλειδί αυτό. Επιπλέον, σε περίπτωση που το κλειδί είναι το τελευταίο στο bucket, τότε είναι πιθανό να προκληθεί υποδιπλασιασμός στο ευρετήριο.

Προσοχή: Κατά τη διαγραφή κλειδιών αλλά και γενικότερα όταν πραγματοποιείται σμίκρυνση/μεγέθυνση του ευρετηρίου πρέπει να ενημερώνονται οι αλλαγές στους δείκτες που θα προκληθούν από τα validations προς τα ValidationPredicate.

Λειτουργίες ValidationIndex

ValidationIndex createValidationIndex();*

OK_SUCCESS insertValidationIndexRecord(ValidationIndex, Key, ValidationPredicate, …);*

ValidationPredicate getValidationIndexRecord(ValidationIndex*, Key, …);*

OK_SUCCESS deleteValidationIndexRecord(ValidationIndex, Key, …);*

OK_SUCCESS destroyValidationIndex(ValidationIndex);*

Οδηγία

Είναι σημαντικό η λειτουργικότητα της εφαρμογής σας να σχεδιαστεί με τέτοιο τρόπο, ώστε να υπάρχει δυνατότητα να ενεργοποιείτε / απενεργοποιείτε τα ευρετήρια που έχετε δημιουργήσει. Δηλαδή, η εφαρμογή σας θα πρέπει να εξακολουθεί να λειτουργεί για παράδειγμα με τις δομές του 1ου επίπεδου κλπ. Με αυτό το τρόπο, θα μπορείτε να διαπιστώνετε κατά πόσο βελτιώνει το χρόνο εκτέλεσης κάθε ευρετήριο ή λειτουργικότητα που προσέθετε και γενικότερα να παρακολουθείτε ευκολότερα τη συμπεριφορά της εφαρμογής σας.