Σκοπός του συγκεκριμένου Query είναι να προτείνει σε κάποιον επισκέπτη (εδώ αυτόν με Visitor\_ID = 276) άλλα events με σειρά προτεραιότητας, βάσει:

* Των γεγονότων που έχει ήδη αξιολογήσει θετικά.
* Κοινών καλλιτεχνών (solo ή μπάντων) σε αυτά.
* Κοινών υπο-ειδών (subgenres) των καλλιτεχνών.

Συγκεκριμένα, εξάγει όλα τα Review του επισκέπτη, επιστρέφοντας το review\_id, την συνολική βαθμολογία (score) και το event\_id. Στην συνέχεια, με χρήση ROW\_NUMBER() κατατάσσει τις ανασκοπήσεις κατά φθίνουσα σειρά βαθμολογίας και κρατά μόνο τις τρεις (3) κορυφαίες (rn ≤ 3). Εντοπίζει τους καλλιτέχνες (Performer\_ID) που συμμετείχαν σε αυτά τα τρία κορυφαία γεγονότα και, εφόσον υπάρχουν solo καλλιτέχνες στα top events, βρίσκει τις μπάντες (Band\_ID) στις οποίες ανήκουν—επεκτείνοντας έτσι το seed set.

Η ένωση (UNION) αυτών των καλλιτεχνών και των μπαντών αποθηκεύεται σε ένα νέο ενιαίο σύνολο, το seed\_performers. Αντίστοιχα, γίνεται και για το seed\_subgenres. Εξάγονται όλα τα subgenre\_ID που σχετίζονται με αυτούς τους καλλιτέχνες, δηλαδή τα μουσικά υπο-είδη που προτιμά ουσιαστικά ο επισκέπτης. Το all\_similar σύνολο θα περιέχει όλους τους καλλιτέχνες που μοιράζονται αυτά τα υπο-είδη.

Ο Υπολογισμός Βαθμολογίας Γεγονότων γίνεται μέσω του event\_scores ως εξής:

Για κάθε γεγονός που περιλαμβάνει κάποιον από τους καλλιτέχνες του all\_similar και δεν έχει ήδη παρακολουθήσει ο επισκέπτης (εξαιρεί τα event\_id του visitor\_reviews), υπολογίζει δύο μετρικές:

* perf\_matches: πλήθος μοναδικών καλλιτεχνών/μπάντων seed που εμφανίζονται στο γεγονός
* subgenre\_matches: πλήθος μοναδικών υπο-ειδών seed στο σύνολο όλων των καλλιτεχνών του γεγονότος

Τέλος για την τελική επιλογή και ταξινόμηση:

το κύριο SELECT προσθέτει μία στήλη total\_score = perf\_matches + subgenre\_matches και ταξινομεί με βάση το total\_score.