ΒΆΣΕΙΣ ΔΕΔΟΜΈΝΩΝ

ΑΝΑΦΟΡΆ ΕΞΑΜΗΝΙΑΊΑΣ ΕΡΓΑΣΊΑΣ

Ομάδα: 44

Μέλη:

- Ιωάννης Καντής Τσαμπίκος 03121721
- Σπυρίδων Αγάθος 03120858
- Παναγιώτης-Νεκτάριος Κατσαλίφης 03118939

Για την εκπόνηση της εργασίας ακολουθήσαμε την εξής μεθοδολογία:

Μέρος Α: Σχεδιασμός και υλοποίηση Βάσης Δεδομένων.

Ξεκινήσαμε μελετώντας την εκφώνηση για να εντοπίσουμε τις οντότητες και τις σχέσεις τους και να φτιάξουμε το ER διάγραμμα .

Στην συνέχεια αναπτύξαμε την βάση μας λαμβάνοντας υπόψη όλους τους περιορισμούς. Συγκεκριμένα :

1. Περιορισμοί ακεραιότητας:

   Σύμφωνα με τις απαιτήσεις της εφαρμογής θέσαμε στις στήλες των πινάκων του κατάλληλους τύπους και επιβάλλαμε όπου χρειάστηκε UNIQUE και NOT NULL.

2. Κλειδιά:

   Κάθε οντότητα έχει ως primary key ένα id το οποίο αυξάνεται αυτόματα με κάθε εισαγωγή.

3. Αναφορική ακεραιότητα:

Για κάθε σχέση οντοτήτων στην βάση μας ορίσαμε κατάλληλα foreign keys ώστε να υπάρχει σχέση μεταξύ των γραμμών του πίνακα της μιας οντότητας με τις γραμμές του πίνακα της άλλης . Για αυτά τα κλειδιά ορίσαμε περιορισμούς έτσι ώστε να ενημερώνονται κατάλληλα όταν αλλάζει κάτι στον έναν πίνακα ή να μην επιτρέπεται η διαγραφή κάποιας γραμμής σε έναν πίνακα (on delete restrict, on update cascade).

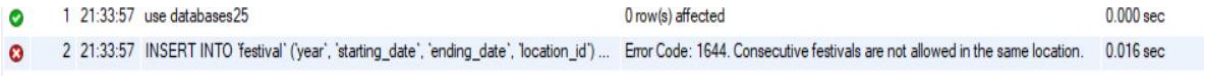
4. Ακεραιότητα πεδίου τιμών:

   Εξασφαλίσαμε όπου χρειάστηκε με χρήση check ότι οι τιμές μιας στήλης είναι στο σωστό εύρος τιμών (π.χ οι εμφανίσεις να έχουν διάρκεια το πολύ 3 ώρες και τα διαλείμματα μεταξύ εμφανίσεων να είναι από 5 έως 30 λεπτά).

5. Περιορισμοί οριζόμενοι από τον χρήστη :

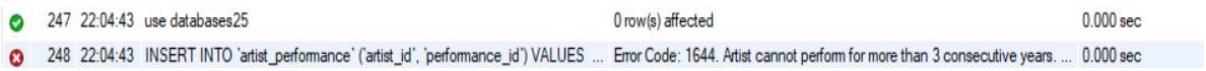
   Με χρήση triggers εξασφαλίσαμε ότι:

- Το φεστιβάλ διεξάγεται σε διαφορετική τοποθεσία ανά έτος.



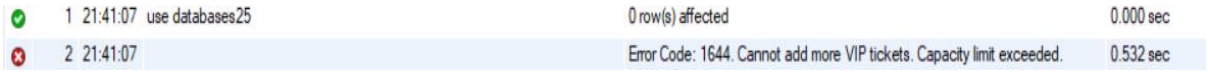
- Κάθε σκηνή μπορεί να φιλοξενεί μόνο μία παράσταση την ίδια στιγμή.

- Ο ίδιος καλλιτέχνης (συγκρότημα) δεν συμμετάσχει για περισσότερα από 3 συνεχή έτη.



- Το προσωπικό ασφαλείας καλύπτει τουλάχιστον το 5% του συνολικού αριθμού θεατών σε κάθε σκηνή και το βοηθητικό προσωπικό το 2%.
- Η χωρητικότητα της σκηνής δεν μπορεί να ξεπεραστεί κατά την πώληση εισιτηρίων.

- Τα VIP εισιτήρια περιορίζονται στο 10% της χωρητικότητας κάθε σκηνής.



- Σε περίπτωση εξάντλησης των εισιτήριων, για μια παράσταση του φεστιβάλ, ενεργοποιείται αυτόματα η ουρά μεταπώλησης.

Επίσης ορίσαμε στον πίνακα των εισιτηρίων τον επιπλέον περιορισμό

CONSTRAINT uq_one_ticket_per_event UNIQUE (visitor_id,event_id)

έτσι ώστε να μην υπάρχουν περισσότερο από ένα εισιτήρια του ίδιου επισκέπτη για την ίδια ημέρα και παράσταση.

Όσον αφορά τα ευρετήρια (indexes) , επειδή έχουμε τα primary και foreign keys τα οποία λειτουργούν σαν indexes, στις περισσότερες περιπτώσεις δεν θα επωφελούμασταν ιδιαίτερα από τον ορισμό περαιτέρω ευρετηρίων.

Μέρος Β: Υλοποίηση και εκτέλεση ερωτημάτων.

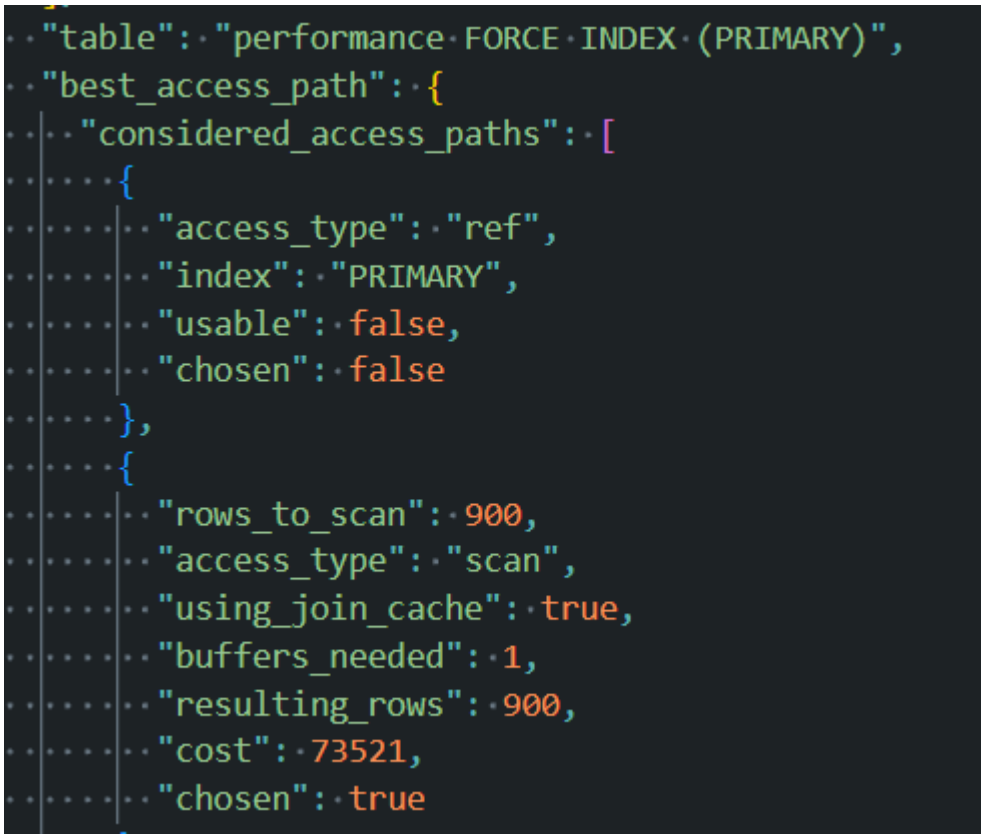
Για τα ερωτήματα 4 και 6 συγκρίναμε τις μεθόδους Nested Loop Join με forced index και Hash-Join.

Ανάλυση αποτελεσμάτων:

Ερώτημα 4)

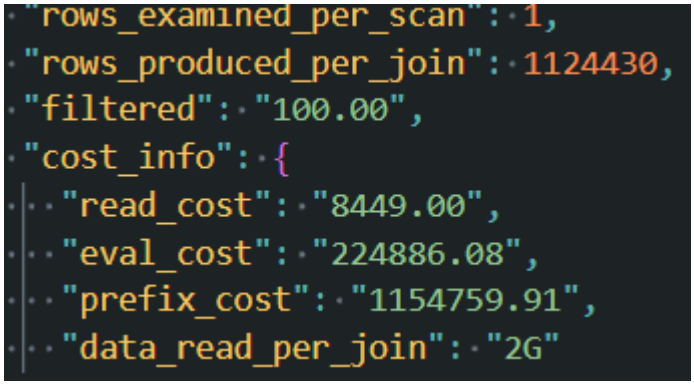
Με την πρώτη μέθοδο βλέπουμε ότι το query χωρίζεται σε δύο στάδια. Join preparation και join optimization. Στο πρώτο στάδιο μετατρέπονται τα join σε where και απαλείφονται παρενθέσεις. Στο δεύτερο στάδιο δημιουργούνται τα table dependencies και επιβάλλεται η σειρά με την οποία θα εκτελεστεί το query και στη συνέχεια γίνεται μια εκτίμηση των γραμμών των πινάκων. Τέλος βλέπουμε τα εναλλακτικά query plans που

δείχνουν τις διαφορετικές επιλογές που είχε να κάνει ο optimizer και ποια διάλεξε τελικά. Παρόλο που έχουμε χρησιμοποιήσει forced indexes βλέπουμε ότι λόγω της σειράς που έχουν γραφτεί τα join στο query στην πραγματικότητα δεν μπορεί πάντα να τα αξιοποιήσει και κάποιες φορές εκτελεί full scan.



Το τελικό κόστος όπως βλέπουμε είναι περίπου 115M το οποίο είναι εξαιρετικά μεγάλο. Αυτό συμβαίνει γιατί τα κόστη με αυτόν τον τρόπο λειτουργούν πολλαπλασιαστικά σε κάθε join.

Με την δεύτερη μέθοδο δημιουργούνται 2 hash tables , ένα για το review και ένα για το performances οπότε το κόστος μειώνεται σημαντικά σε συνολικά 1.15M δηλαδή μειωμένο κατά 100 φορές σε σχέση με την πρώτη μέθοδο.



Ερώτημα 6)

Με την πρώτη μέθοδο βλέπουμε ότι αυτή την φορά τα forced indexes λειτούργησαν καλύτερα και πήραμε συνολικό κόστος 1.600 ενώ όπως βλέπουμε ένα full scan θα είχε σαν αποτέλεσμα 287000.

Με την δεύτερη μέθοδο σε αυτήν την περίπτωση παίρνουμε σχεδόν ίδιο κόστος δηλαδή 1600. Αυτό συμβαίνει γιατί επιλέγουμε μόνο έναν επισκέπτη δηλαδή μία γραμμή και άρα η αναζήτηση γίνεται ήδη σε σταθερό χρόνο.