



SQL-SERVER Ευρετήρια (Indexes) & Βελτιστοποίηση Επερωτήσεων (Query Optimization)

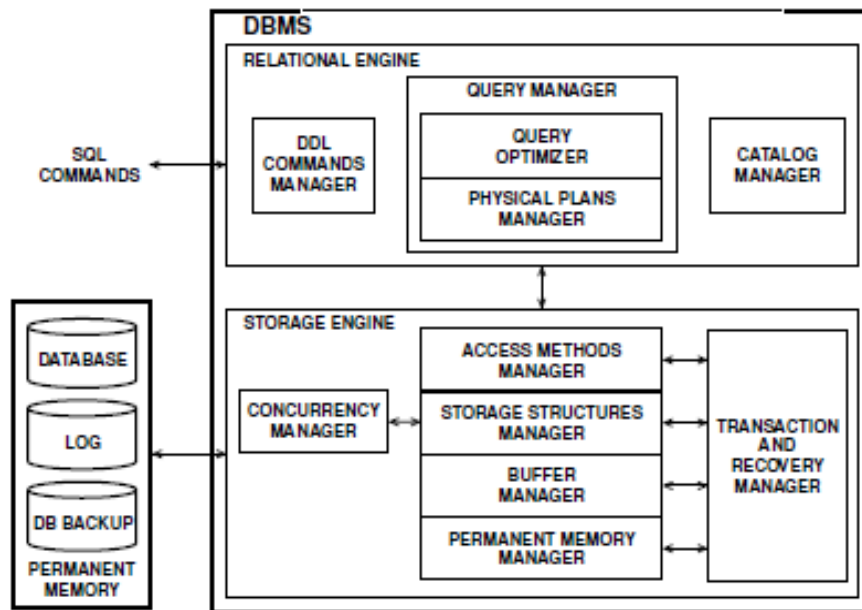
Χρυσόστομος Καπέτης



Αρχιτεκτονική Συστημάτων Διαχείρισης Βάσεων Δεδομένων (DBMS)

Query Optimization

- Ένα απλοποιημένο μοντέλο αρχιτεκτονικής ενός DBMS περιλαμβάνει τα ακόλουθα βασικά συστατικά (components).



Στην πραγματικότητα η λειτουργικότητα όλων αυτών των επιμέρους συστατικών δεν είναι τόσο ξεκάθαρη όσο φαίνεται στο διπλανό σχήμα.

Source: A. Albano, D. Colazzo, G. Ghelli, R. Orsini, DBMS Internal, 2015.

- Relational Engine:** DBMS “Brain”
- Storage Engine** : DBMS “Muscle”



Relational Engine

- **Data Definition Language (DDL) Manager:** Επεξεργάζεται το σχήμα της βάσης.
- **Query Manager:** επεξεργάζεται το ερώτημα ενός χρήστη και το μετατρέπει σε μια ισοδύναμη αλλά πιο αποτελεσματική μορφή βρίσκοντας έτσι μια καλή στρατηγική για την εκτέλεσή του.
- **Catalogue Manager:** διαχειρίζεται τα μεταδεδομένα της βάσης.



Storage Engine

Query Optimization

- **Permanent Memory Manager:** διαχειρίζεται τις σελίδες στον δίσκο.
- **Buffer Manager:** διαχειρίζεται την μεταφορά των σελίδων από τον δίσκο στην μνήμη και αντίστροφα.
- **Storage Structure Manager:** διαχειρίζεται τις δομές δεδομένων με στόχο την αποθήκευση και αποτελεσματική ανάκτηση των δεδομένων.
- **Access Methods Manager:** παρέχει την απαιτούμενη λειτουργικότητα για την δημιουργία και την διαγραφή βάσεων, αρχείων και δεικτών καθώς επίσης και τις μεθόδους πρόσβασης για σάρωση πινάκων και δεικτών.
- **Transaction and Recovery Manager:** διασφαλίζει την ακεραιότητα της βάσης δεδομένων σε περιπτώσεις αποτυχημένων συναλλαγών και αστοχιών του συστήματος
- **Concurrency Manager:** φροντίζει για την αποφυγή συγκρούσεων σε περιπτώσεις ταυτόχρονης πρόσβασης.



Εγγραφές (Records)

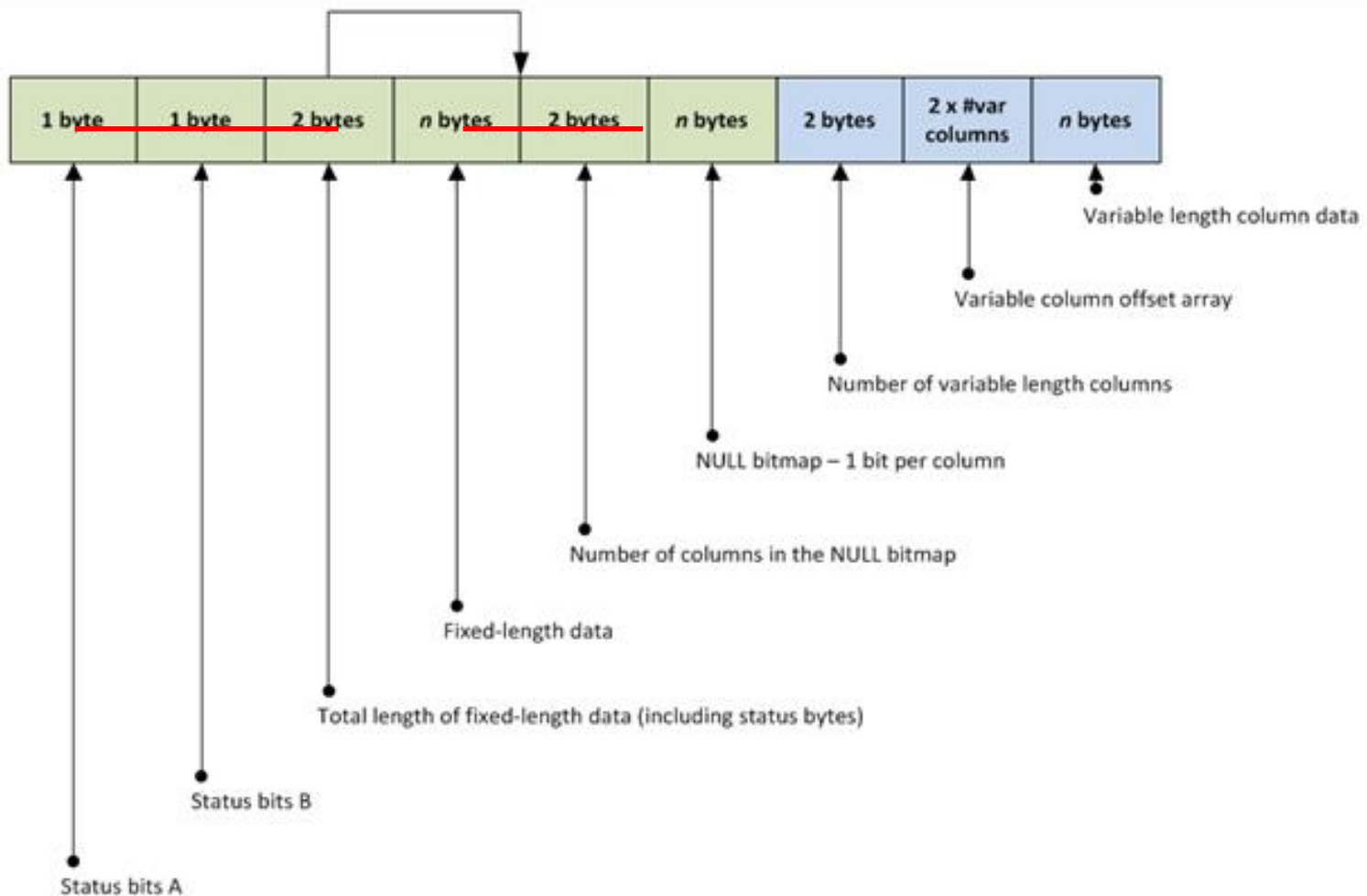
- Μία εγγραφή είναι η μικρότερη αποθηκευτική δομή ενός αρχείου στον SQL Server.
- Κάθε γραμμή ενός πίνακα αποθηκεύεται στον δίσκο με την μορφή μιας εγγραφής.
- Υπό την μορφή εγγραφών δεν αποθηκεύονται μόνο τα δεδομένα των πινάκων.
- Υπό την μορφή εγγραφών αποθηκεύονται επίσης οι δείκτες και τα μεταδεδομένα.



SQL SERVER: Δομή Εγγραφής

Query Optimization

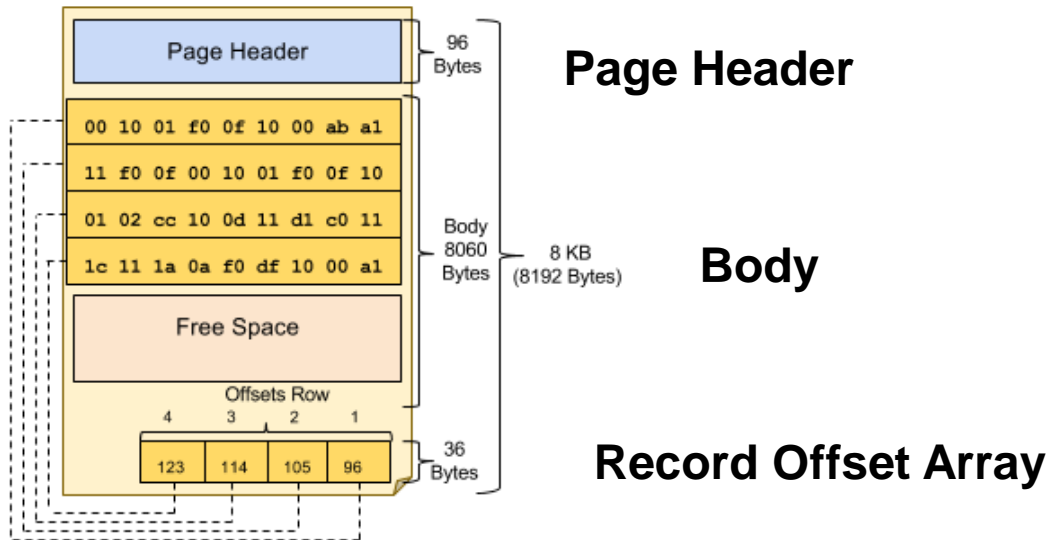
- Στον SQL Server μια εγγραφή αποθηκεύεται σε μορφή **fixedvar**.





SQL SERVER: Δομή Σελίδας

- Οι εγγραφές αποθηκεύονται σε σελίδες.
- Τα δεδομένα μεταφέρονται από τον δίσκο στην μνήμη και αντίστροφα υπό την μορφή σελίδων, τις οποίες διαχειρίζεται ο **Buffer Manager**.
- Η δομή μιας σελίδας στον SQL server είναι η ακόλουθη.





Στατιστικά Στοιχεία (TIME)

SET STATISTICS TIME ON

Εμφανίζει στατιστικά στοιχεία σχετικά με τον χρόνο εκτέλεσης μιας συναλλαγής.

SET STATISTICS TIME OFF

Απενεργοποιεί την εμφάνιση στατιστικών στοιχείων



SET STATISTICS IO ON

Εμφανίζει στατιστικά στοιχεία σχετικά με τις απαιτήσεις IO μιας συναλλαγής.

SET STATISTICS IO OFF

Απενεργοποιεί την εμφάνιση των στατιστικών στοιχείων .



Ευρετήρια (Indexes)

- Τα ευρετήρια είναι δομές δεδομένων που βοηθούν στην ταχύτερη ανάκτηση των δεδομένων της βάσης.
- Ένα ευρετήριο δημιουργείται σε μία ή περισσότερες στήλες ενός πίνακα.
- Υπάρχουν δύο βασικοί τύποι ευρετηρίων στον SQL Server
 - **Clustered Index** (Ευρετήριο συστάδων)
 - **Non-Clustered Index** (Απλό ευρετήριο)
- Τα ευρετήρια στον SQL server οργανώνονται σε μορφή B+δέντρου (B+tree).



Ευρετήριο Συστάδων B+tree (B+ tree Clustered Index)

- Ένα ευρετήριο συστάδων ορίζει την φυσική διάταξη (ταξινόμησης και αποθήκευση) των εγγραφών ενός πίνακα.
- Όταν δημιουργούμε ένα ευρετήριο συστάδων σε μία στήλη ενός πίνακα οι εγγραφές του πίνακα θα ταξινομηθούν με βάση τις τιμές της συγκεκριμένης στήλης και θα αποθηκευτούν στον δίσκο με αυτήν την διάταξη.
- Σε κάθε πίνακα μπορούμε να δημιουργήσουμε μόνο ένα ευρετήριο συστάδων.
- Ο SQL server δημιουργεί αυτόματα ένα μοναδικό ευρετήριο συστάδων στην στήλη (ή στήλες) που απαρτίζουν το πρωτεύον κλειδί.
- Ένα ευρετήριο συστάδων μπορούμε να το δημιουργήσουμε με την παρακάτω εντολή:

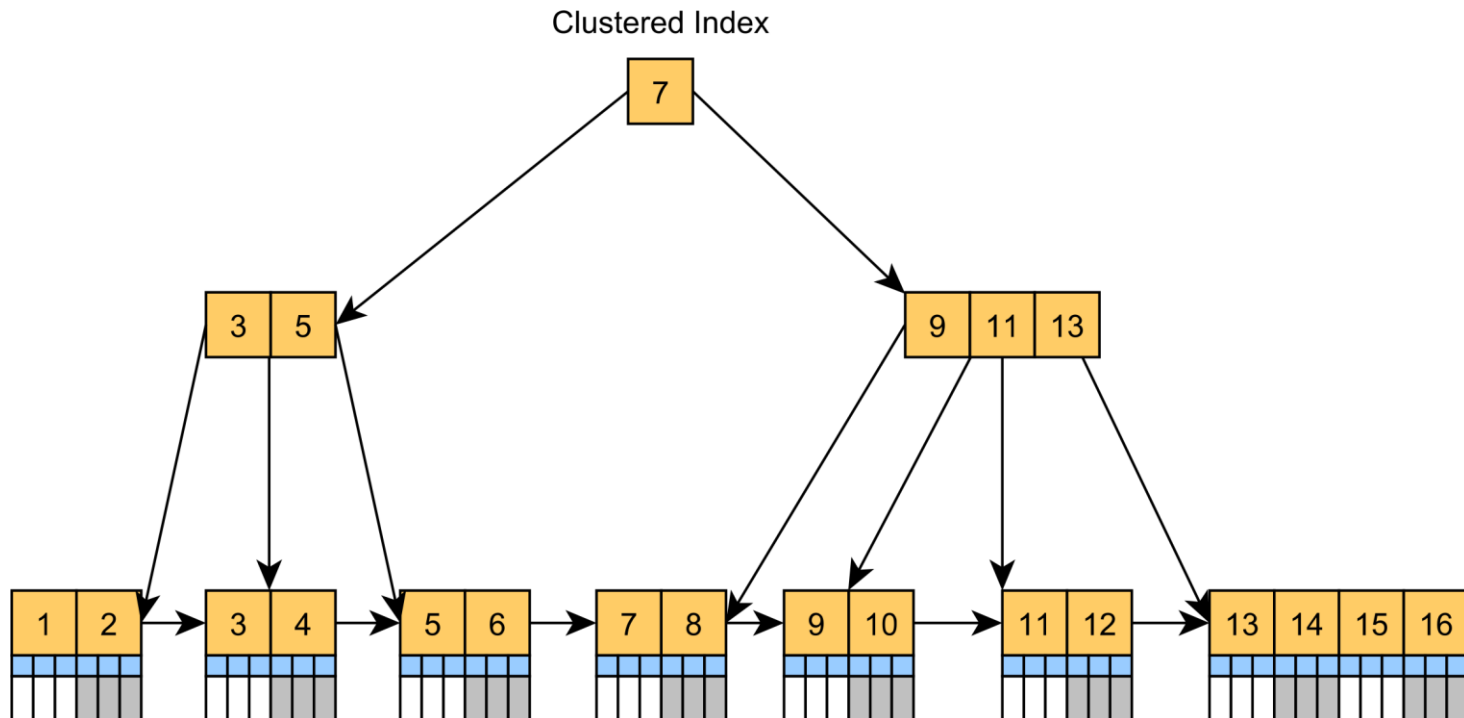
```
CREATE CLUSTERED INDEX <index_name>  
ON <table_name> (column_name);
```



Ευρετήριο Συστάδων B+tree (B+ tree Clustered Index)

- Παράδειγμα:

CREATE CLUSTERED INDEX cid_idx ON customer(cid)





Απλά ευρετήρια B+tree (B+ tree non-Clustered Index)

- Ένα απλό ευρετήριο δεν καθορίζει την φυσική διάταξη (ταξινόμηση και αποθήκευση) των εγγραφών ενός πίνακα.
- Δημιουργεί μια ξεχωριστή δομή **κλειδιού-τιμής** από τα δεδομένα του πίνακα. Το κλειδί περιέχει τις τιμές των στηλών που αποτελούν το ευρετήριο και κάθε τιμή περιέχει έναν δείκτη (pointer) προς την εγγραφή του πίνακα που περιέχει την πραγματική τιμή (δηλαδή την τιμή του κλειδιού).
- Σε κάθε πίνακα μπορούμε να δημιουργήσουμε μέχρι και 999 απλά ευρετήρια.
- Ένα απλό ευρετήριο μπορούμε να το δημιουργήσουμε με την παρακάτω εντολή

```
CREATE NONCLUSTERED INDEX <index_name>  
ON <table_name> (column_name);
```

- Με τον προσδιοριστή UNIQUE μπορούμε να ορίσουμε ότι οι όλες οι τιμές του ευρετηρίου είναι μοναδικές.

```
CREATE UNIQUE NONCLUSTERED INDEX <index_name>  
ON <table_name> (column_name);
```



Απλά ευρετήρια B+tree (B+ tree non-Clustered Index)

- Ένα απλό ευρετήριο μπορούμε να το δημιουργήσουμε με την παρακάτω εντολή

```
CREATE NONCLUSTERED INDEX <index_name>  
ON <table_name> (column_name);
```

- Αν παραλείψουμε τον προσδιοριστή **NONCLUSTERED** το αποτέλεσμα θα είναι το ίδιο. Με άλλα λόγια η παραπάνω εντολή είναι ισοδύναμη με την ακόλουθη:

```
CREATE INDEX <index_name>  
ON <table_name> (column_name);
```

- Σε ένα απλό ευρετήριο μπορούμε να συμπεριλάβουμε και άλλα πεδία εκτός από το κλειδί αναζήτησης με τον προσδιοριστή **INCLUDE**. Π.χ.

```
CREATE INDEX ON customer (lastname) INCLUDE (firstname)
```

- Για να διαγράψουμε ένα ευρετήριο χρησιμοποιούμε την εντολή **DROP INDEX**:

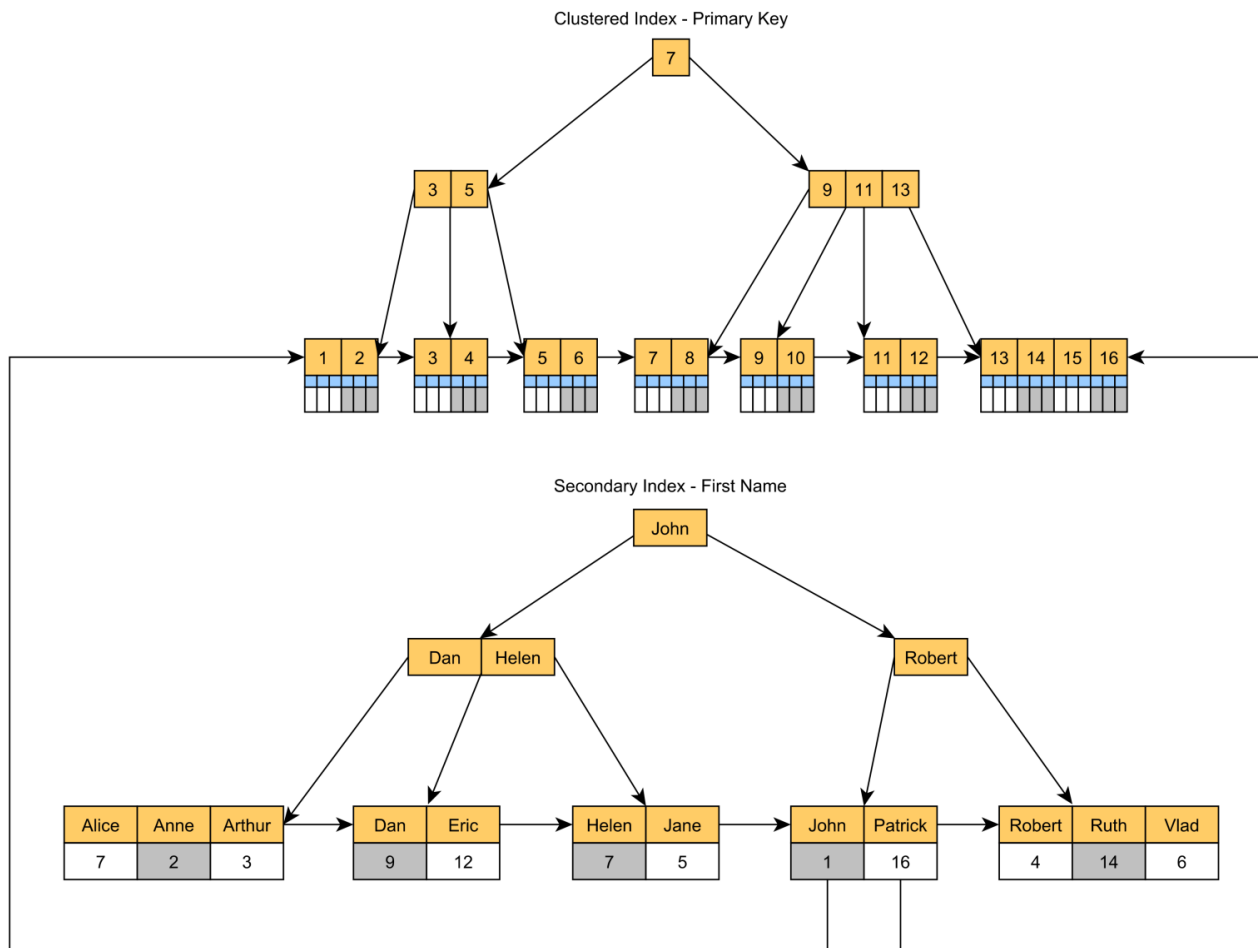
```
DROP INDEX <index_name>  
ON <table_name>;
```



Απλά ευρετηρια B+ tree (Non Clustered Index B+ tree)

- Παράδειγμα:

CREATE NONCLUSTERED INDEX fname_idx **ON** customer(fname)





Βελτιστοποίηση Επερωτήσεων

Query Optimization

- SQL Performance Tuning: Διαδικασία βελτιστοποίησης της απόδοσης των επερωτήσεων.
 - Ελαχιστοποίηση του χρόνου που απαιτείται για να λάβει ο χρήστης τα αποτελέσματα μετά την υποβολή ενός επερωτήματος.
 - Ελαχιστοποίηση των πόρων που χρειάζονται για την επεξεργασία του επερωτήματος.
- Τα περισσότερα σχεσιακά συστήματα DBMS διενεργούν αυτοματα βελτιστοποίηση της απόδοσης των επερωτήσεων στην πλευρά του διακομιστή.
- Οι περισσότερες τεχνικές βελτιστοποίησης της απόδοσης των επερωτήσεων SQL εξαρτώνται από το εκάστοτε DBMS.
- Η πλειονότητα των προβλημάτων απόδοσης οφείλεται σε κακογραμμένο κώδικα SQL.



Επιλεξιμότητα Ευρετηρίων (Index Selectivity)

- Τα ευρετήρια χρησιμοποιούνται όταν:
 - Η ευρετηριασμένη στήλη εμφανίζεται στα κριτήρια αναζήτησης (προσδιοριστές WHERE/HAVING)
 - Η ευρετηριασμένη στήλη ακολουθεί τους προσδιοριστές GROUP BY ή ORDER BY.
 - Οι συναρτήσεις MAX ή MIN εφαρμόζουν στην ευρετηριασμένη στήλη.
- Η επιλεξιμότητα ενός ευρετηρίου (index selectivity) είναι ένα μέτρο της πιθανότητας χρήσης του ευρετηρίου στην επεξεργασία ερωτήσεων.



Επιλεξιμότητα Ευρετηρίων (2)

- Γενικές οδηγίες για την χρήση ευρετηρίων:
 - Εξετάστε την δημιουργία ευρετηρίων για τα γνωρίσματα που ακολουθούν τους προσδιοριστές WHERE, HAVING, ORDER BY, ή GROUP BY.
 - Απογύγετε την δημιουργία ευρετηρίων σε μικρούς πίνακες (πίνακες με λίγα δεδομένα).
 - Ορίστε τα πρωτεύοντα και ξένα κλειδιά Declare έτσι ώστε ο βελτιστοποιητής να χρησιμοποιεί ευρετήρια στην λειτουργία της ισοσύνδεσης των πινάκων.
 - Δημιουργήστε ευρετήρια σε στήλες βάσει των οποίων γίνεται η ισοσύνδεση των πινάκων εφόσον αυτές δεν είναι πρωτεύοντα ή ξένα κλειδιά.



Κριτήρια Αναζήτησης

TABLE
11.6

Conditional Criteria

OPERAND1	CONDITIONAL OPERATOR	OPERAND2
P_PRICE	>	10.00
V_STATE	=	"FL"
V_CONTACT	LIKE	"Smith%"
P_QOH	>	P_MIN * 1.10



Κριτήρια Αναζήτησης (1)

- Χρησιμοποιείτε απλές στήλες ή λεκτικά
 - Αποφύγετε τις συναρτήσεις
- Η σύγκριση αριθμητικών πεδίων είναι ταχύτερη σε σχέση με αλφαριθμητικά πεδία και πεδία ημερομηνιών.
 - than character, date, and NULL comparisons
- Η ισότητα εκτελείται ταχύτερα από την ανισότητα.
 - Ο τελεστής LIKE είναι πιο αργός
- Γράψτε πρώτα τους όρους ισότητας.
- AND: Χρησιμοποιείτε πρώτα την συνθήκη που πιθανότητα είναι ψευδής.
- OR: Χρησιμοποιείτε πρώτα την συνθήκη που είναι πιθανότερο να ισχύει.
- Να αποφεύγετε τον τελεστή NOT



Διαμόρφωση Επερωτήσεων

- Προσδιορίστε τις στήλες και τους υπολογισμούς που απαιτούνται
 - Λογικές εκφράσεις
 - Συγκεντρωτικές συναρτήσεις
- Προσδιορίστε τους κατάλληλους πίνακες
- Προσδιορίστε τον τρόπο ισοσύνδεσης των πινάκων.
- Προσδιορίστε τα κριτήρια αναζήτησης.
- Προσδιορίστε την σειρά εμφάνισης των αποτελεσμάτων.



Πλάνα εκτέλεσης ερωτήσεων (Query Execution Plans)

- Το πλάνο εκτέλεσης μιας ερωτήσεως παρέχει πληροφορίες σχετικά με τον τρόπο εκτέλεσης της ερωτήσεως, όπως:
 - Την σειρά προσπέλασης των πινάκων
 - Τις μεθόδους πρόσβασης και ανάκτησης δεδομένων από τους πίνακες της βάσης
 - Τις μεθόδους που χρησιμοποιούνται για την σύζευξη των πινάκων, το φιλτράρισμα και την ταξινόμηση των δεδομένων.
- Υπάρχουν δύο τύποι πλάνων εκτέλεσης:
 - **Estimated execution plan:** παράγεται από τον optimizer με βάση τις εκτιμήσεις δίχως να εκτελεστεί το ερώτημα.
 - **Actual execution plan:** το πραγματικό πλάνο εκτέλεσης το οποίο είναι διαθέσιμο κατόπιν της εκτέλεσης της ερωτήσεως.



Πλάνα εκτέλεσης ερωτήσεων (Query Execution Plans)

- Μέθοδοι πρόσβασης και ανάκτησης δεδομένων από τους πίνακες της βάσης:
 - **Table scan:** Ο sql-server διαβάζει όλες τις εγγραφές του πίνακα προκειμένου να επιστρέψει το επιθυμητό αποτέλεσμα
 - **Index scan:** Ο sql-server επιλέγει να διαβάσει τα δεδομένα του ευρετηρίου. Σε περιπτώσεις που τα επιθυμητά δεδομένα υπάρχουν στο ευρετήριο, προτιμάται αυτή η μέθοδος διότι απαιτείται η ανάγνωση λιγότερων σελίδων.
 - **Index seek:** Ο sql-server χρησιμοποιεί το ευρετήριο με αποδοτικό τρόπο για να ανακτήσει τα επιθυμητά δεδομένα από τον πίνακα. Το ευρετήριο αντιστοιχεί πλήρως στο κριτήριο αναζήτησης της ερώτησης.



Πλάνα εκτέλεσης ερωτήσεων (Query Execution Plans)

- Μέθοδοι σύζευξης πινάκων:
 - **Nested Loop Join:** η σύζευξη δύο πινάκων επιτυγχάνεται με την χρήση δύο εμφωλευμένων βρόχων
 - **Sort Merger Join:** Αυτή η μέθοδος ταξινομεί τις εγγραφές των δύο πινάκων με βάση το κλειδί της σύζευξης και στην συνέχεια συγχωνεύει τις ταξινομημένες λίστες και εντοπίζει τις εγγραφές που πληρούν το κριτήριο της σύζευξης.
 - **Hash Join:** Ο sql-server δημιουργεί έναν πίνακα κατακερματισμού στην μνήμη εφαρμόζοντας μια συνάρτηση κατακερματισμού στο πεδίο της σύζευξης του εσωτερικού πίνακα (build table). Στην συνέχεια η ίδια συνάρτηση κατακερματισμού εφαρμόζει στο πεδίο σύζευξης του εξωτερικού πίνακα (probe table). Τα κατακερματισμένα κλειδιά συγκρίνονται και εντοπίζονται αυτά που ταιριάζουν.