
ESSENSYS™
SOFTWARE SOLUTIONS
Center for Career Development
by LINKgroup
Microsoft Partner
Gold Application Development
Silver Application Integration
Silver Data Platform
Silver Collaboration and Content
Quality software. On time.
Every time.

ROADMAP

- Limite în SQL. Câte rânduri, câte coloane?
- De ce ne trebuie indecși
- Am o problemă de performanță. Unde mă uit?
- Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?
- Pierderi de date
- Cum să îmi fac viața de dezvoltator SQL mai ușoară
- Întrebări și răspunsuri
- Tips & tricks
- Concluzii

Limite în SQL. Câte rânduri, câte coloane?

Extras din Maximum Capacity Specifications for SQL Server

SQL Server Database Engine object	Maximum sizes/numbers SQL Server (64 -bit)
Rows per table	Limited by available storage
Columns per nonwide table	1,024
Columns per wide table	30,000
Bytes per row	8,060

Limite în SQL. Câte rânduri, câte coloane?

Cerința:

aducerea datelor în tabele SQL așa cum sunt la sursă (structură, denumiri câmpuri)

Problema 1:

unele obiecte au foarte multe câmpuri (peste 300 - până la 800)

Problema 2:

mărimea maximă a unui rând ce poate fi inserat într-o tabela SQL este de 8060 bytes.

Demo

Limite în SQL. Câte rânduri, câte coloane?

Soluția adoptată: partiționarea verticală a tabelelor

Ce înseamnă:

un set de tabele parțiale, independente, care au aceleași înregistrări pe coloana PK și fiecare alt set de coloane.

Cum se face:

cu comenzi de

SELECT <coloane> INTO <tabela parțială n> FROM <tabela originală>

se pot partiționa tabele care conțin date

se pot partiționa inclusiv tabele deja partiționate

Limite în SQL. Câte rânduri, câte coloane?

Soluția adoptată: partiționarea verticală a tabelelor

Vizualizare date:

peste setul de tabele parțiale se definește un view, cu aceeași denumire ca tabela originală
view-ul referă toate coloanele din toate tabelele parțiale. Coloana PK este referită o singură dată în view.
Tabelele parțiale sunt legate de prima tabelă parțială prin clauze de LEFT JOIN după valoarea coloanelor PK.

Limite în SQL. Câte rânduri, câte coloane?

Soluția adoptată: partiționarea verticală a tabelelor

Adăugare de coloane:

coloanele sunt repartizate în tabelele cu cele mai puține coloane, pentru echilibrarea numărului coloanelor între tabelele parțiale

view-ul este recreat după fiecare adăugare de coloane

Demo

Urmează: De ce ne trebuie indecși?

De ce ne trebuie indecși?

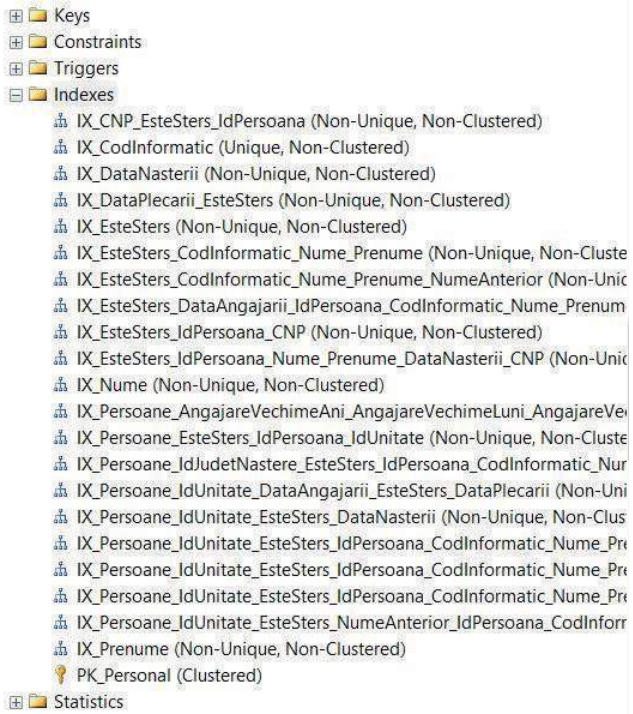
Pentru că îi cere planul de execuție…

Pentru fiecare query, planul de execuție poate să ceară crearea unor indecși diferiți.

Soluția cea mai la îndemână: Missing index details, creare index.

Rezultatul: câte index pentru fiecare query.

De ce ne trebuie indecși?



De ce ne trebuie indecși?

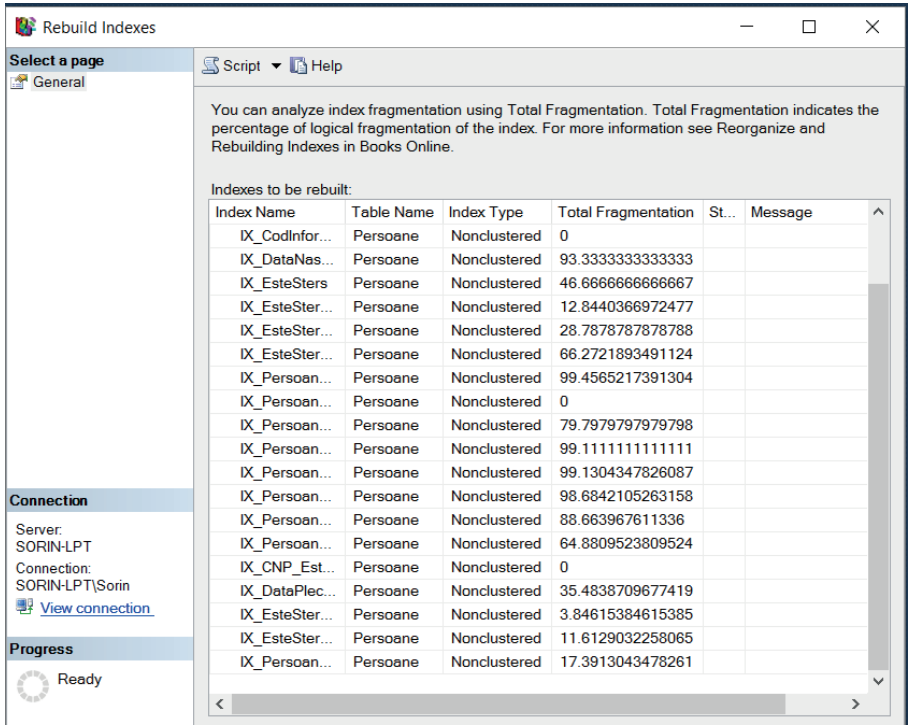
Crearea și întreținerea multor indecși nu e o soluție.

Inflația de indecși duce la performanțe scăzute, contrar rolului lor.

Operațiile de INSERT, UPDATE, DELETE au efect asupra tuturor indecșilor, care trebuie updatați. Costul update-ului a 20 de indecși este mult mai mare decât a 2 indecși. Este ca și cum am updata 20 de tabele.

În timp, indecșii se fragmentează, iar acest fapt se reflectă în scăderea performanțelor.

De ce ne trebuie indecși?



Index Name	Table Name	Index Type	Total Fragmentation	St...	Message
IX_CodInfor...	Persoane	Nonclustered	0		
IX_DataNas...	Persoane	Nonclustered	93.3333333333333		
IX_EsteSters	Persoane	Nonclustered	46.6666666666667		
IX_EsteSter...	Persoane	Nonclustered	12.8440366972477		
IX_EsteSter...	Persoane	Nonclustered	28.7878787878788		
IX_EsteSter...	Persoane	Nonclustered	66.2721893491124		
IX_Persoan...	Persoane	Nonclustered	99.4565217391304		
IX_Persoan...	Persoane	Nonclustered	0		
IX_Persoan...	Persoane	Nonclustered	79.7979797979798		
IX_Persoan...	Persoane	Nonclustered	99.1111111111111		
IX_Persoan...	Persoane	Nonclustered	99.1304347826087		
IX_Persoan...	Persoane	Nonclustered	98.6842105263158		
IX_Persoan...	Persoane	Nonclustered	88.663967611336		
IX_Persoan...	Persoane	Nonclustered	64.8809523809524		
IX_CNP_Est...	Persoane	Nonclustered	0		
IX_DataPlec...	Persoane	Nonclustered	35.4838709677419		
IX_EsteSter...	Persoane	Nonclustered	3.84615384615385		
IX_EsteSter...	Persoane	Nonclustered	11.6129032258065		
IX_Persoan...	Persoane	Nonclustered	17.3913043478261		

De ce ne trebuie indecși?

E mai bine să modificăm query-ul pe baza indecșilor existenți, până când obținem cea mai buna performanță.

Numai dacă nu există altă soluție, creăm un nou index.

Întrebarea ar fi: De ce ne trebuie atâția indecși?

Urmează: Am o problemă de performanță. Unde mă uit?

Am o problemă de performanță. Unde mă uit?

De ce se insistă atât de mult pe performanță?

Lucrurile merg bine, nu sunt erori în bazele de date sau în aplicații, toată lumea este mulțumită.

Chiar așa este?

Am o problemă de performanță. Unde mă uit ?

Coadă la ghișeu.

Am o problemă de performanță. Unde mă uit?

La ghișeu, un funcționar care lucrează la un calculator. Din când în când pare că îngheață, cu ochii în monitor. Așteaptă să se termine ceva acolo, pe ecran, pentru ca apoi să treacă mai departe.

Acel ceva este o problemă de performanță nerezolvată, dintr-o bază de date sau dintr-o aplicație.

Am o problemă de performanță. Unde mă uit?

Ce înseamnă o secundă de așteptare a răspunsului aplicației, în cifre:

- Pentru a zecea persoană de la coadă, înseamnă zece secunde din timpul ei.
- Pentru toate cele zece persoane, timpul cumulat de este de 55 de secunde (1+2+3+4+5+6+7+8+9+10).
- La un volum de numai 100 de persoane pe zi (10 cozi) înseamnă 550 de secunde adică 9 minute și 10 secunde.
- La 3 ghișee pe unitate înseamnă 1650 de secunde, adică 27 minute și 30 de secunde.
- Dacă aplicația este folosită la nivel național, cu cate 3 ghișee / județ, ajungem la 69300 de secunde, adică 19 ore și 15 minute. Pe zi.

Am o problemă de performanță. Unde mă uit?

Ce înseamnă o secundă de așteptare a răspunsului aplicației, în cifre:

- 19 ore și 15 minute pe zi, la nivel național, pentru că există o problemă de performanță nerezolvată, dintr-o bază de date sau dintr-o aplicație.
- Și este doar o secundă…

Am o problemă de performanță. Unde mă uit?

Resurse utilizate de SQL Server pentru un query:

Disc. Fizic sau virtual. Viteza de acces. Număr de citiri, scrieri.
CPU. 1, 2, 8, 16 procesoare.
Memorie RAM alocată serverului SQL.
Timp fizic necesar query-ului pentru a fi executat.

Între anumite limite, CPU, memoria RAM și discul pot fi extinse.
Asupra timpului nu putem interveni.

Dar putem interveni asupra query-ului.

Am o problemă de performanță. Unde mă uit?

Simptome ale unei probleme de performanță:

- aplicațiile merg greu
- uneori apar erori de timeout
- serverul SQL ocupă 99% din procesor și / sau din memorie
- IO este foarte mare

Am o problemă de performanță. Unde mă uit?

Unde mă uit?

- Activity Monitor
- SQL Server Profiler
- Execution Plan

Se pot găsi cauzele problemelor care apar ACUM.

Am o problemă de performanță. Unde mă uit?

Dar dacă problemele sunt intermitente?

În SQL Server există Rapoarte statistice, la nivel de instanță SQL sau la nivel de bază de date.

Urmează: Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Demo

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Stadii ale unui produs software:

analiză

dezvoltare (DEV)

testare la dezvoltator (QA)

pre-producție (staging) (STAGE)

producție (PROD)

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Pentru fiecare stadiu există câte un mediu, fiecare fiind configurat potrivit scopului său.

Dezvoltare (DEV):

- număr de înregistrări relativ mic
- încărcare scăzută a serverului SQL
- număr mic de utilizatori
- concurență scăzută
- utilizare intermitentă a bazelor de date
- număr mic de servere SQL, pentru aplicații distribuite (1 - 4)
- fără aplicații sau baze de date satelit

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Pentru fiecare stadiu există câte un mediu, fiecare fiind configurat potrivit scopului său.

Testare (QA):

- număr de înregistrări mediu
- încărcare scăzută a serverului SQL (testarea urmărește în principal depistarea erorilor)
- număr mic de utilizatori
- concurență medie (artificială)
- utilizare intermitentă a bazelor de date
- număr mic de servere SQL, pentru aplicații distribuite
- poate avea aplicații sau baze de date satelit, cu încărcare minimă

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Pentru fiecare stadiu există câte un mediu, fiecare fiind configurat potrivit scopului său.

Pre-producție (STAGE):

- număr de înregistrări ridicat
- încărcare medie a serverului SQL
- număr mic de utilizatori
- concurență medie (artificială)
- utilizare continuă a bazelor de date
- număr mic de servere SQL, pentru aplicații distribuite
- are aplicații sau baze de date satelit, cu încărcare medie

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Pentru fiecare stadiu există câte un mediu, fiecare fiind configurat potrivit scopului său.

Producție (PROD):

- număr de înregistrări ridicat
- încărcare ridicată a serverului SQL
- număr mare de utilizatori
- concurență mare (reală)
- utilizare continuă a bazelor de date
- număr mare de servere SQL, pentru aplicații distribuite (zeci, sute)
- are aplicații sau baze de date satelit, cu încărcare mare

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Pentru fiecare stadiu există câte un mediu, fiecare fiind configurat potrivit scopului său.

	DEV	QA	STAGE	PROD
număr de înregistrări	Relativ mic	Mediu	Ridicat	Ridicat
încărcarea serverului SQL	Scăzută	Scăzută	Medie	Mare
număr de utilizatori	Mic	Mic	Mic	Mare
concurență	Scăzută	Medie (artificială)	Medie (artificială)	Mare (reală)
utilizarea bazelor de date	Intermitentă	Intermitentă	Continuă	Continuă
număr de servere SQL, pentru aplicații distribuite	Mic	Mic	Mic	Mare (zeci, sute)
aplicații sau baze de date satelit	Fără	Poate, cu încărcare minimă	Da, cu încărcare medie	Da, cu încărcare mare

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Atunci când se face trecerea software-ului de pe un mediu pe altul …

Apar erori!

Scheme de baze de date diferite.
Lipsesc obiecte în noul mediu (tabele, coloane, proceduri stocate, funcții, view-uri, useri)
Procedurile stocate, funcțiile, view-urile au versiuni diferite. Cod neactualizat
Coloanele sau variabilele au tipuri de date diferite
Setări de collation diferite.
Probleme de securitate: GRANT-uri pierdute odată cu DROP-uirea obiectelor

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Atunci când se face trecerea software-ului de pe un mediu pe altul …

Apar anomalii!

Lipsesc înregistrări din nomenclatoare sau din tabele de bază.
Exista înregistrările, dar au alte id-uri (în special dacă sunt coloane de tip Identity).
Triggeri uitați sau neactualizați
Tabele cu denumiri identice în scheme diferite.
SELECT col1, col2 FROM Tabela1 vs. SELECT col1, col2 FROM Schema1.Tabela1.
Folosire incorectă a funcțiilor SQL built-in.

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Atunci când se face trecerea software-ului de pe un mediu pe altul …

Query-urile durează foarte mult!

Statisticile sunt diferite, planurile de execuție alese pot fi altele decât cele de pe DEV, iar query-ul a fost scris și optimizat cu un anumit plan de execuție.
Lipsesc indecși sau sunt fragmentați.
Query-urile conțin iteratori care au funcționat bine pe un număr mic de înregistrări, dar care nu funcționează bine pe un număr mare de înregistrări
Instrucțiunile folosesc mai multe date decât au nevoie. Înregistrări si coloane extrase și prelucrate în plus.

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Atunci când se face trecerea software-ului de pe un mediu pe altul …

Query-urile durează foarte mult! Apar erori!

SELECT * FROM tabela nu are ce căuta în producție.

Pot sa fie adăugate coloane de alte procese.

```
INSERT INTO Tabela2
SELECT * FROM Tabela1
```

Dacă în Tabela2 a fost adăugată o coloană de un alt proces decât cel de deployment curent, atunci va apare o eroare.

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Reguli de deployment

Prerechizite.

pregătire mediu de instalare

scripturi ajutătoare

kit de instalare

instrumente de instalare.

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Reguli de deployment

Plan, procedură detaliată. Cine, ce face, când, cum, cu ce și de ce. (Exemplu)

Estimare durata de deployment și în caz de reușită și în caz de eșec.

Backup!

- backup de baze de date,
- backup de aplicații,
- backup de mașini virtuale.

Verificare backup! RESTORE VERIFYONLY FROM DISK='c:\....bak‘ (exemplu)

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Reguli de deployment

În plan să nu existe nici un Point of No Return!

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Reguli de deployment

Să existe un plan de restaurare în caz de eșec a instalării în producție. Verificat și actualizat după fiecare instalare!

Aducerea mediului în starea de dinaintea începerii instalării – stabilă și consistentă
Instrucțiuni clare de refacere completa a mediului inițial. La fel ca la planul de instalare, să conțină cine, ce face, când, cum, cu ce și de ce.
Restaurarea datelor, inclusiv din bazele de date ale aplicațiilor conexe.
Recuperare date.

Pe mediul de dezvoltare funcționează fără probleme. Ce s-a întâmplat în producție?

Reguli de deployment

Verificare . Și în caz de reușită, și în caz de eșec.

Scripturi cu elemente care trebuie verificate (număr de înregistrări, scripturi de verificare a calității datelor).

Schema Compare, Data Compare din Visual Studio

Alte tooluri și metode de verificare

Urmează: Pierderi de date

Pierderi de date

. . .

Pierderi de date

Datele sunt cele mai importante

Datele sunt ale clientului / beneficiarului.

Baza de date poate fi restaurată din backup, din scripturi de creare a structurilor sau dintr-o altă bază de date similară.

Tabelele, procedurile stocate, funcțiile, view-urile, userii, joburile, toate obiectele pot fi restaurate.

Aplicațiile pot fi recompilate din surse, reinstalate

Pierderi de date

Datele sunt cele mai importante

Datele pierdute pot fi restaurate din backup, până la momentul backup-ului.

Datele pierdute mai pot fi recuperate de pe suporturi externe, dacă există, cu un cost mare și cu posibile inconsistențe. (DVD, SAN, NAS, stick-uri, carduri, documente primare)

Restul datelor sunt definitiv pierdute.

Pierderi de date

Consecințe:

utilizatori frustrați, pentru că munca lor s-a pierdut, și nu din vina lor
imposibilitatea desfășurării activităților uzuale până la stabilizarea sistemului
pierderea încrederii în stabilitatea sistemului și în consistența datelor.
pierderea credibilității departamentului de IT și personal a celui responsabil de gestionarea bazelor de date. Pierderea de date este cel mai mare coșmar al DBA-ului și a dezvoltatorului SQL.
recuperarea datelor este un mare consumator de resurse prețioase (materiale, umane, de timp) și este neproductivă.

Pierderi de date

Pierderile de date se produc prin:

1. ștergere - datele dispar.

- DELETE FROM table.
- TRUNCATE TABLE
- DROP TABLE
- DROP DATABASE

Pierderi de date

Pierderile de date se produc prin:

2. alterare - datele există, dar sunt inutilizabile, pentru că nu mai sunt reale și consistente. Cea mai periculoasă situație, pentru că nu este vizibilă imediat.

- UPDATE table fără clauza WHERE.
- INSERT sau UPDATE într-o sursă greșită.
- INSERT sau UPDATE dintr-o sursă greșită.

Pierderi de date

Cauzele pierderilor de date sunt:

analiza superficială

testarea insuficientă sau incompletă a proceselor și a fluxurilor

nestăpânirea proceselor automate. Joburi, triggeri, servicii Windows, servicii web.

intervențiile directe asupra datelor, prin comenzi SQL sau prin executarea unor proceduri stocate, fără a ști exact ce se întâmplă

Pierderi de date

Cauzele pierderilor de date sunt:

efectuarea de operații pe bazele de date sau pe tabele la momente nepotrivite sau în ordine greșită.

orientarea unei alte versiuni de aplicație către baza de date din producție

o prea mare siguranță de sine.

Pierderi de date

Posibilități de evitare a pierderilor de date:

Documentați-vă asupra proceselor și a operațiilor efectuate de acestea.

Nu presupuneți că lucrurile vor decurge după cum vă așteptați. Asigurați-vă!

Verificați contextul în care efectuați operațiile. E posibil ca ceva să se fi schimbat față de ultima operație similară pe care ați făcut-o, fără să fiți informat.

Pierderi de date

Posibilități de evitare a pierderilor de date:

Daca vi se solicită intervenții directe asupra datelor (de obicei corecții minore), puteți refuza. Este alegerea voastră și responsabilitatea acțiunilor ulterioare vă aparține în totalitate. Un NU spus când trebuie vă poate scuti de foarte multe probleme.

Înainte de a face orice, ascultați-vă vocea interioară. Dacă vă dă semnale de alarmă, chiar dacă nu puteți defini exact unde ar putea fi o problemă, mai bine amânați și cercetați până când sunteți siguri.

Pierderi de date

. . .

Urmează: Cum să îmi fac viața de dezvoltator SQL mai ușoară

Cum să îmi fac viața de dezvoltator SQL mai ușoară

- Codare
- Stil de programare
- Arhitectură
- Administrare baze de date
- Out of computer

Cum să îmi fac viața de dezvoltator SQL mai ușoară

Codare

Folosiți Pascal case pentru denumiri de obiecte din bazele de date și variabile.

Ex. LocalitateaDeDomiciliu, TelefonPersonal, TelefonDeServiciu

Folosiți comentarii, fără a face excese.

Generați cod, daca e posibil, în loc să îl scrieți.

Folosiți facilitățile din SSMS. Intellisense, drag and drop, cursoare multiple, shortcuturi.

Cum să îmi fac viața de dezvoltator SQL mai ușoară

Codare

Scrieți cod așa cum v-ar place vouă să îl găsiți.

Lăsați în urma voastră cod mai bun decât cel existent.

Cum să îmi fac viața de dezvoltator SQL mai ușoară

Stil de programare

- Evitați iteratorii, pe cât posibil, în favoarea operațiilor bulk.
- Nu se pun reguli de business în view-uri.
- Nu se hardcodează valori care pot fi găsite printr-o interogare.
- Folositi obiecte care exista deja. Funcții, view-uri, proceduri stocate, baze de date.
- Subquery vs CTE. Înlănțuire

Cum să îmi fac viața de dezvoltator SQL mai ușoară

Stil de programare

- Outputuri imediate folosind tabelele virtuale inserted și deleted
- Variabile tabel vs. tabele temporare
- Folosiți view-uri de sistem
- Folosiți facilități SQL native (ROWNUMBER, paginare cu OFFSET, IIF)

Cum să îmi fac viața de dezvoltator SQL mai ușoară

Arhitectură

- Folosiți informații care există deja în alte baze de date.
- Încercați să nu dezvoltați "insule informatice".
- Think Big

Cum să îmi fac viața de dezvoltator SQL mai ușoară

Administrare

Backup. Backup si verificare

Puneti la treabă alte servere SQL.

- Registered servers.
- Query executat remote.

Cum să îmi fac viața de dezvoltator SQL mai ușoară

Out of computer

Informați-vă. Documentație, legislatie, discuții cu specialiștii din domeniu, cu utilizatorii finali.

Colaborați.

Scrieti documentatie de analiza.

Intretineti un folder cu documentatie despre incidente. Descrieti problema intampinata, cauzele pe care le-ati identificat, modul de rezolvare, pas cu pas. Includeti scripturile si orice alte tooluri folosite.

Cum să îmi fac viața de dezvoltator SQL mai ușoară

Out of computer

- Nu ezitați să spuneți NU când este cazul.

Întrebări și răspunsuri

Tips & tricks

Demo

Vă mulțumesc!