

DB2 for i performance

strumenti e
suggerimenti per
migliorare le
performance

Marco Riva
Sirio Informatica e Sistemi



Ultimo aggiornamento: 31/5/2019

2

Parleremo di...

- il motore del DB2
- concetti chiave sull'ottimizzazione
- modernizzare significa anche migliorare le performance
- suggerimenti per migliorare le performance
- strumenti di analisi



ONL-i Days 2019 - DB2 for i performance

3

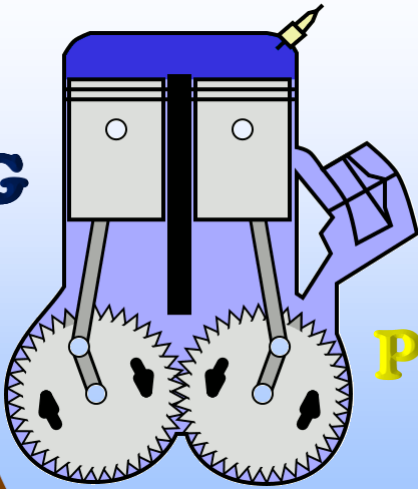
DB2 engine



ONL-i Days 2019 - DB2 for i performance

4

Dentro il motore



SQE

LPG

GSC

MTI

IOA

CQE

AQP

Plan Cache

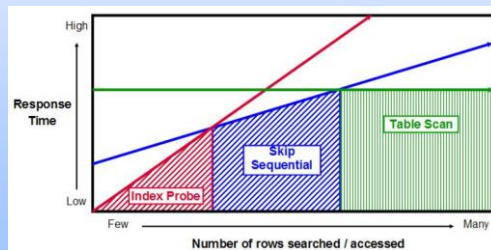
IMO

ONL-i Days 2019 - DB2 for i performance

5

Query optimizer: i costi

- Il motore del DB2 adotta una logica di ottimizzazione basata sui «costi» (*costo = tempo **stimato** per eseguire una richiesta*):
 - vengono valutati uno o più piani di accesso
 - viene scelto quello dal «costo» più basso
 - il piano di accesso viene salvato per un futuro riutilizzo



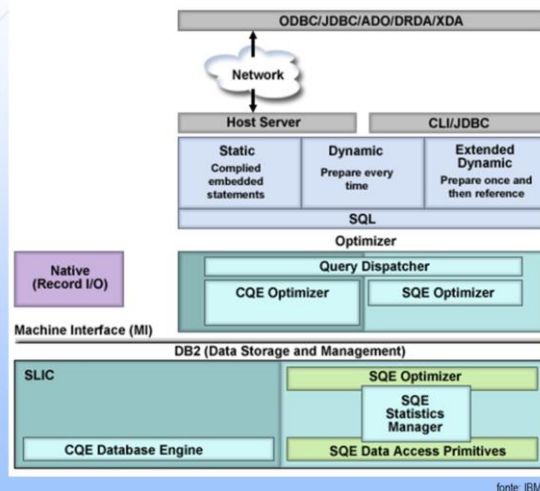
- Lo **scopo primario** dell'ottimizzatore è ridurre il **tempo** e non l'utilizzo delle risorse
- Il **mezzo principale** dell'ottimizzatore è ridurre I/O

ONL-i Days 2019 - DB2 for i performance

fonte: IBM

6

DB architecture da V5R2: CQE vs SQE



► il **Query Dispatcher** ha in carico il lavoro di "smistare" le richieste SQL tra l'ottimizzatore tradizionale CQE (Classic Query Engine) e il nuovo SQE (SQL Query Engine)

► A partire dalla **V5R2** ad ogni release si è allargato il campo di azione di SQE

ONL-i Days 2019 - DB2 for i performance

fonte: IBM

Fino a **6.1** SQE non veniva utilizzato se nella clausola from si indicava un file logico creato con DDS.

SQE da **7.2** ottimizza anche le query native ovverosia non-SQL queries (OpnQryF, RunQry). *Dalla 7.2 SQE elabora praticamente tutte le query*

Query dispatcher - Limitazioni che impongono l'utilizzo di CQE:

- 7.1: https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_71/rzajq/rzajqdispatcher.htm
- 7.2: https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_72/rzajq/rzajqdispatcher.htm
- 7.3: https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_73/rzajq/rzajqdispatcher.htm
- 7.4: https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_74/rzajq/rzajqdispatcher.htm
 - INSERT WITH VALUES statement or the target of an INSERT with subselect statement
 - tables with Read triggers
 - Read-only queries with more than 1000 dataspace, or updatable queries with more than 256 dataspace.
 - DB2® Multisystem tables
 - multi-format logical files (solo per 7.1)
 - non-SQL queries, for example the QQQQry API, Query/400, or OPNQRYF (solo per 7.1, da 7.2 la limitazione riguarda solo QQQQry API)

Opzione in QAQQINI: **SQE_NATIVE_ACCESS** (in 7.2 e 7.3)

Si potrebbe voler continuare ad elaborare le query native con CQE impostando questa opzione a *NO.

IBM advice: Legacy queries that have been running without any issues for many years and over many IBM i releases will now be processed by a completely different query engine. As a result, the run time behavior and results returned can be different for OPNQRYF and Query/400 requests.

Based on the potential impacts on run time performance and query result sets, IBM recommends that you perform extensive functional testing and performance benchmarking of your applications and all reports.

*If you want to avoid the risk of potential impacts, you can set the QAQQINI parameter, SQE_NATIVE_QUERY, to a value of *NO. With a value of *NO, DB2 will limit SQE usage to SQL-based interfaces, making the behavior equivalent to the IBM i 6.1 and 7.1 releases.*

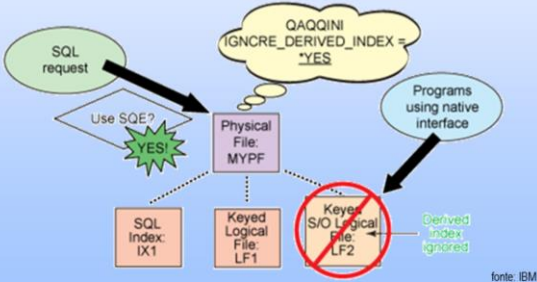
Strategically, it is a better return on investment to spend time converting your OPNQRYF and Query/400 reports to SQL. This will provide you access to the superior functionality and advanced features offered by standard and portable SQL.

CQE vs SQE: ignore derived index

- Impostando a *YES l'opzione **IGNORE_DERIVED_INDEX** del file QAQQINI si consente all'ottimizzatore SQE di ignorare gli indici derivati e processare la query al posto di CQE. Il piano di accesso risultante verrà creato ignorando l'esistenza degli indici derivati

**in 6.1 il default è *YES
da 7.1 non esiste più perché
SQE gestisce anche gli indici
derivati**

Cosa è un indice derivato?
https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_74/rzajq/rzajqderivedindex.htm



ONL-i Days 2019 - DB2 for i performance

Opzioni in QAQQINI: **IGNORE_DERIVED_INDEX**

8

creazione e refresh di una MQT che riepiloga il fatturato per cliente/mese con raffronto anno su anno

SQE riduce enormemente i tempi di esecuzione:
creazione: -78%
refresh: -97%

Cliente	R. mese	Anno	Fatt. anno	Fatt. anno	Delta
			corrente	precedente	
1	1	Generale	37127.31	47829.68	-10702.4
1	2	Febbraio	12779.79	28878.21	-17098.4
1	3	Marzo	38947.1	50907.25	-11960
1	4	Aprile	50084.14	49134.68	949.46
1	5	Maggio	16238.67	39444.43	-23205.8
1	6	Giugno	424.47	49303.5	-48859
1	7	Luglio	0	76959.58	-76959.6
1	8	Agosto	0	52969.26	-52969.3
1	9	Settembre	0	72449.58	-72449.6
1	10	Ottobre	0	22135.26	-22135.3
1	11	Novembre	0	46156.58	-46156.6
1	12	Dicembre	0	46032.46	-46032.5
2	1	Generale	17618.2	20510.58	-2892.38
2	2	Febbraio	5011.88	18869.02	-14857.1
2	3	Marzo	20043.7	11380.4	8663.3
2	4	Aprile	16999.58	31770.49	-14770.9
2	5	Maggio	24602.7	44721.42	-20118.7
2	6	Giugno	14709.55	15857.1	-1057.55
2	7	Luglio	16439.02	31148.4	-14709.4
2	8	Agosto	6667.55	7764.5	-1097
2	9	Settembre	5851.42	28939.7	-27088.3
2	10	Ottobre	0	9794.52	-9794.52
2	11	Novembre	1284.67	14470.35	-13185.7
2	12	Dicembre	0	15044.85	-15044.9

SQE

Creazione: 28 sec

Refresh: 4,6 sec.

CQE

Creazione: 182 sec

Refresh: 133 sec.

1.930.000 record

ONL-i Days 2019 - DB2 for i performance

ONL-i Days

6

9

IOA: Index Only Access

- Questa tecnica consente al DB2 di accedere solo agli indici per restituire il risultato di una query senza accedere ai record della tabella
- Questo metodo può essere utilizzato quando il DB2 determina che tutte le colonne referenziate nella query sono parte della definizione dell'indice

ONL-i Days 2019 - DB2 for i performance

10

IMO: Index Merge Ordering

da 7.2

- SQE in 7.2 (DB2 PTF group level 5) è stato migliorato per poter utilizzare un indice contemporaneamente sia per la selezione dei record (con un operatore diverso da =) sia per l'ordinamento anche se su colonne diverse
- I miglioramenti di questa tecnica si possono osservare soprattutto nel caso in cui il resultset è molto ampio ed è stato richiesto un'ottimizzazione per «first I/O»

ONL-i Days 2019 - DB2 for i performance

SQE - Index Merge Ordering

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/SQE%20-%20Index%20Merge%20Ordering>

11

AQP – Adaptive Query Processing

- ▀ disponibile da IBM i **7.1** (solo **SQE**)
- ▀ lo scopo è migliorare le performance delle query più «pesanti» mentre sono in esecuzione
- ▀ «**adaptive**» significa:
 - ▀ **react in real time**: durante l'esecuzione della query può cambiare il piano di accesso
 - ▀ **learn over time**: «impara» dalla query corrente per migliorare future esecuzioni della stessa query o simili
- ▀ **self-managing**: non è necessario né configurarlo né amministrarlo
- ▀ AQP «entra in gioco» sulle query complesse, con tempi di esecuzione lunghi apportando significativi miglioramenti di performance

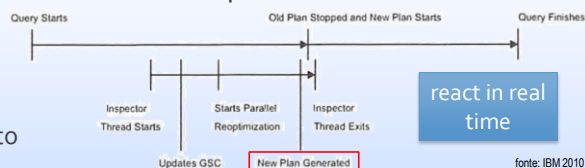
ONL-i Days 2019 - DB2 for i performance

Opzioni in QAQQINI: `ALLOW_ADAPTIVE_QUERY_PROCESSING`

12

AQP - componenti

- current query inspector:** si attiva 2 secondi dopo l'inizio della query e "scova"
 - join che producono "scarsi" risultati (=pochi record)
 - conteggio di righe stimate molto diverse dal conteggio reale
- plan inspector:** entra in gioco dopo il completamento della query per analizzare la "bontà" del piano utilizzato ed aggiornare la GSC
- global statistics cache (GSC):**
 - self-managing
 - contiene statistiche complesse (tabelle multiple o derivate)



ONL-i Days 2019 - DB2 for i performance

13

LPG – Look-ahead Predicate Generation

- **"Power and Magic"** (Mike Cain)
- SQE fornisce una potente strategia per **minimizzare l'I/O** e massimizzare le performance della query
- **Magic**: l'ottimizzatore "riscrive" la query generando clausole di selezione locale
- Tipicamente il **join di tabelle** è lo scenario che può causare numerose operazioni di lettura: se l'ottimizzatore riesce a scovare strategie per diminuire i record da leggere prima del join, le performance della query possono migliorare

da V5R3 (solo SQE)

The DB2
Power!

ONL-i Days 2019 - DB2 for i performance

14

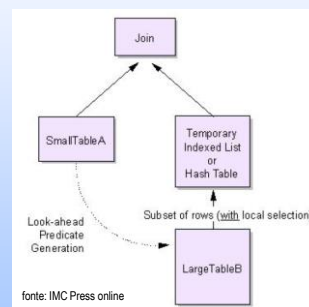
LPG in azione nei join

- A è una tabella piccola con una clausola di selezione
- B è una tabella grande senza clausola di selezione
- La selettività del predicato sulla tabella A è molto alta

```
SELECT *
FROM SmallTableA A, LargeTableB B
WHERE A.Join_Col = B.Join_Col
AND A.Coll IN (112358, 132134)
```

```
SELECT *
FROM SmallTableA A, LargeTableB B
WHERE A.Join_Col = B.Join_Col
AND A.Coll IN (112358, 132134)
AND B.Join_Col IN (value1, value2)
```

ONL-i Days 2019 - DB2 for i performance

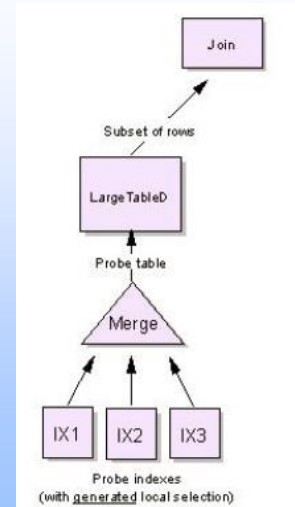


LPG identifica i valori *value1*, *value2* da usare sulla colonna di join per creare una clausola di selezione locale sulla tabella B PRIMA del join

15

LPG in azione con gli indici: index ANDing

- **index ANDing**: è la capacità dell'ottimizzatore di usare più di un indice per identificare ed accedere ai record di una tabella
- Usando indici su colonne-chiave-singole il motore del DB2 può unire la lista dei record identificati da ogni indice e generare una lista combinata dei soli record a cui accedere



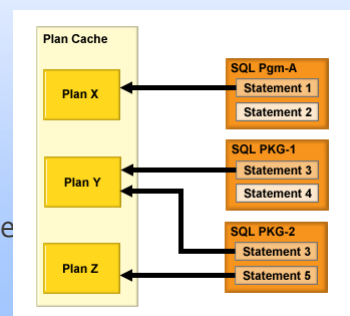
ONL-i Days 2019 - DB2 for i performance

fonte: IBM

16

Plan cache

- è il «contenitore» di tutti i piani di accesso elaborati da SQE
- no COE!
- una volta che il piano di accesso viene creato è disponibile per **TUTTI** gli utenti e per **TUTTE** le query indipendentemente dall'interfaccia con cui vengono richieste
- eventuali indici creati dall'ottimizzatore sono disponibili per **TUTTI**
- da 7.2 il sistema gestisce autonomamente la dimensione massima (default 512 Mb)



ONL-i Days 2019 - DB2 for i performance

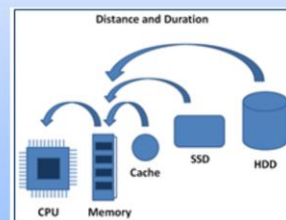
fonte: IBM

17

In-Memory DB enablement

da 7.1

- ▀ disponibile da 7.1 solo per interfacce SQL (no query native, no native record level access)
- ▀ la parte più lenta dell'esecuzione di una query è il tempo per portare i dati da leggere dal disco alla memoria
 - ▀ clausola **KEEP IN MEMORY** per istruzioni **CREATE** o **ALTER TABLE** (da 7.2)
 - ▀ parametro **KEEPINMEM** per comandi **CHGPF** o **CHGLF** (da 7.1)
- ▀ quando in un'istruzione SQL si riferenzia un oggetto con keep in memory = *YES SQE richiede che l'oggetto venga portato in memoria con un processo asincrono rispetto all'esecuzione della query



fonte: IBM

ONL-i Days 2019 - DB2 for i performance

Opzioni in QAQQINI: **MEMORY_POOL_PREFERENCE**

Questa opzione condiziona in quale area di memoria vengono portati gli oggetti con keep in memory = *yes.

Per default gli oggetti del DB2 vengono portati nel pool di memoria del job.

Tramite l'opzione MEMORY_POOL_PREFERENCE è possibile specificare il numero di pool di memoria, oppure *BASE o *JOB.

La gestione degli oggetti del DB2 in-memory è più efficiente se il pool di memoria ha l'opzione di paginazione impostata a *CALC.



19

SELECT: ordine esecuzione

1. ORIGIN
2. SELECTING
3. JOINING
4. GROUPING
5. SELECTING GROUPS
6. PROJECTION
7. ORDERING
8. FETCHING

from
where
join
group
having
select
order
fetch first

DA RICORDARE !

ONL-i Days 2019 - DB2 for i performance

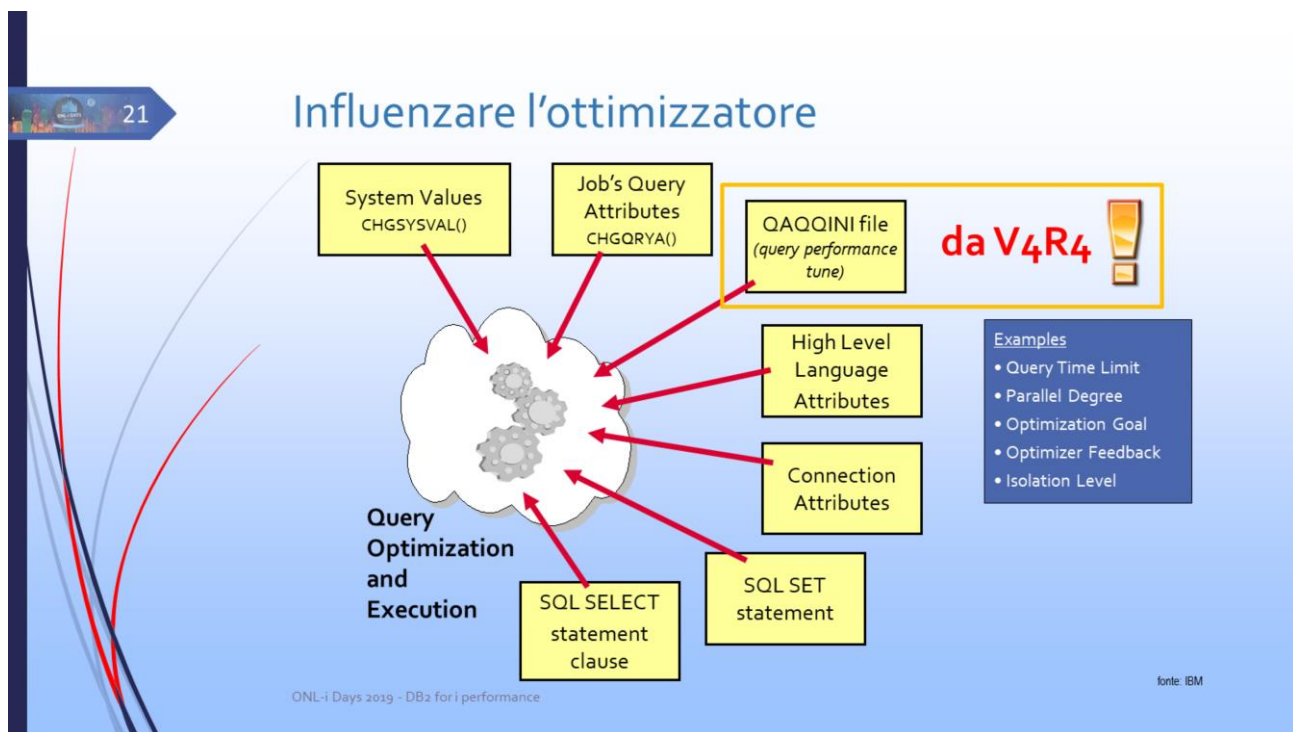




Un determinato piano di accesso può essere pensato come un'intersezione di tutti i fattori che condizionano l'ottimizzazione basata sui costi su uno specifico server su uno specifico disegno di database.

Per capire veramente un piano di accesso e migliorarne le performance, bisogna conoscere e capire tutti i diversi fattori e le impostazioni che condizionano i tempi di ottimizzazione ed esecuzione.

Cambiando uno o più fattori il piano di accesso può cambiare.



QAAQINI query options (da V4R4):

https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_73/rzajq/qryopt.htm

La libreria di default da cui vengono lette le opzioni del file QAAQINI è QUSRSYS. Tramite il comando CHGQRYA è possibile modificarla.

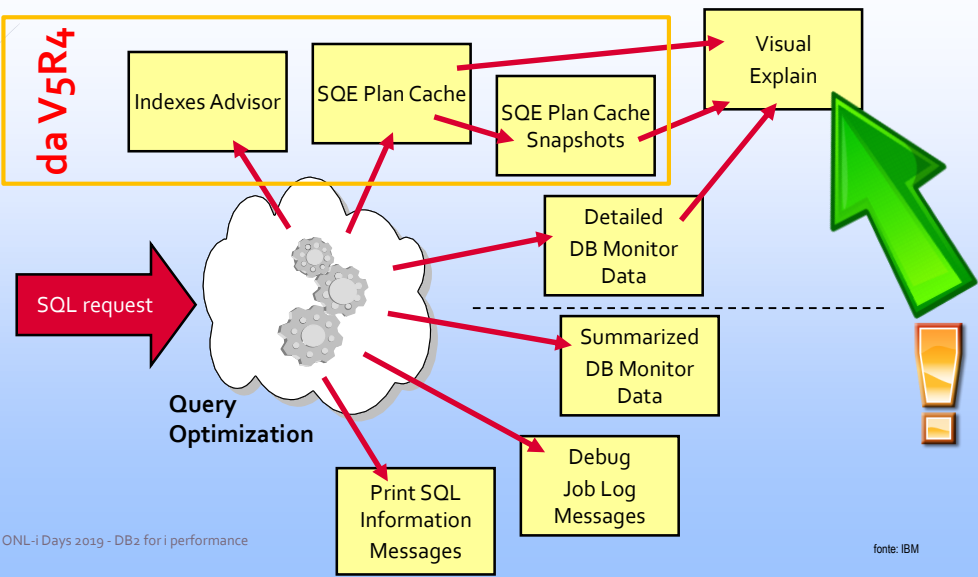
ATTENZIONE! quando si copia (CRTDUPOBJ) il file QAAQINI dal file modello presente nella libreria QSYS bisogna obbligatoriamente copiare i dati, ma soprattutto i trigger ad esso associati, altrimenti il file non verrà utilizzato.

Il file QAAQINI è composto da 3 campi: QPARM (il nome del parametro), QQVAL (il valore del parametro), QQTEXT (un campo descrittivo).

Per aggiornare il file QAAQINI utilizzare istruzioni SQL di INSERT o UPDATE.

Prima dell'introduzione del file QAAQINI il comportamento dell'ottimizzatore poteva essere influenzato utilizzando l'area dati QQOPTIONS.

Il feedback dell'ottimizzatore



23

Proattivo vs reattivo

- Le strategie di indicizzazione allo scopo di massimizzare le performance sono di due tipi:
 - proattivo**: dall'analisi dell'applicazione costruisco in anticipo gli indici necessari in base al modello dei dati (primary key, unique key, foreign key) e quelli che molto probabilmente soddisferanno la maggior parte delle query (join, selezioni)
 - reattivo**: dall'analisi del feedback dell'ottimizzatore decido di costruire indici specifici per soddisfare le richieste derivanti da query specifiche oppure posso anche decidere di cancellare indici non necessari
- i due metodi non si escludono a vicenda, anzi andrebbero usati iterativamente



ONL-i Days 2019 - DB2 for i performance

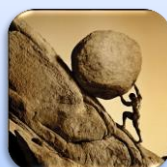
24

In sintesi...

- non esiste una «ricetta facile» da seguire per ottimizzare. Moltissimo dipende da un approccio «**fai, prova ed osserva**» sfruttando la tecnologia F.B.L.
- un «buon ottimizzatore» deve avere il **T.I.C.**



Intuizione



Tenacia



Curiosità

ONL-i Days 2019 - DB2 for i performance

25

Modernizzare per migliorare

ONL-i Days 2019 - DB2 for i performance

26

Modernità che migliorano

- evitare OpnQryF
- evitare I/O nativo
- evitare un numero eccessivo di aperture di file

**USARE EMBEDDED SQL**

- evitare scrittura di file di work temporanei

usare le **CTE** (Common Table Expression)

- non usare le DDS

usare **SQL DDL**

follow up ⇌
lun 11.30 Petris:
Modernizzazione RPG

ONL-i Days 2019 - DB2 for i performance

27

Reverse engineering: genera SQL

- processo per reperire da un oggetto di database creato tramite DDS lo script SQL per ri-crearlo

IBM Access Client Solutions
gestione schemi

ONL-i Days 2019 - DB2 for i performance

Perché reverse engineer?

- migliorare le prestazioni
- IBM ha più volte annunciato che non ci saranno ulteriori sviluppi delle DDS

Metodi disponibili:

- API **QSQGNDLL** "Generate Data Definition Language" (da **V5R1**) interfacciata da Esegui script SQL in via grafica
- procedura SQL **GENERATE_SQL** (da **7.2** e **7.1 TR9**)
 - più facile da usare, consente anche di eseguire conversioni di più oggetti contemporaneamente

Sia la API QSQGNDLL sia la procedura GENERATE_SQL non creano il file e il membro sorgente, ma devono già esistere.

"The IBM i native interfaces are widely used and continue to be supported by IBM. However, these native interfaces are not being enhanced with the same functionality as the strategic DB2 for i interface – SQL."

As a result, IBM i developers need to start adopting and using SQL." (M.Cain, K.Milligan)

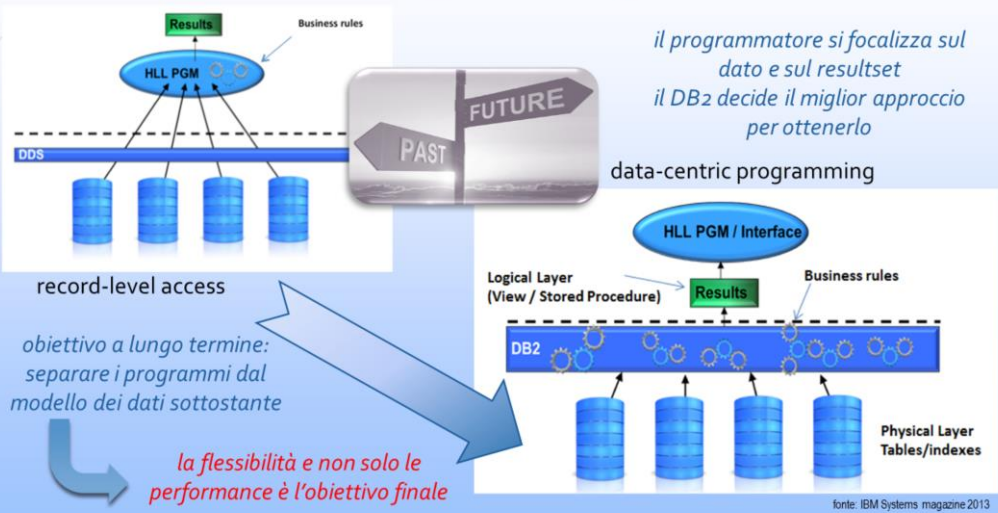
Documentazione procedura SQL GENERATE_SQL:

https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_73/rzajq/rzajqprocgensql.htm

Esempio procedura SQL:

```
call qsys2/generate_sql(
  'ORDIC00F', -- nome tabella da convertire
  '$$LIBFS3', -- nome librerie
  'TABLE',    -- tipo oggetto
  'QSQL',     -- file sorgente
  'MK1TEST',  -- libreria del file sorgente
  'ORDIC00F', -- nome membro sorgente
  replace_option => '1', -- opzione sostituzione membro
  naming_option => 'SYS',
  ccsid_option => '0',
  create_or_replace_option => '1',
  system_name_option => '1',
  qualified_name_option => '0'
);
```

SQL data-centric programming



fonte: IBM Systems magazine 2013

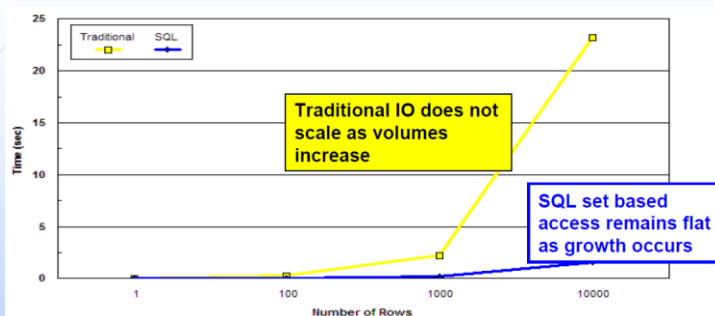
ONL-i Days 2019 - DB2 for i performance

follow up
lun 9.30 Schianca:
Database Design con
Data Studio

Il ruolo del DBE è di vitale importanza per produrre un ottimo modello di dati che consente al DB di «conoscere» il dato e massimizzare le prestazioni. Nella logica di programmazione tradizionale «record-level-access» è il programma (ovvero il programmatore) e non il DB che «conosce» il dato.

29

SQL vs I/O nativo



E' impossibile scrivere dati non validi in una tabella SQL (*create table*) perché i dati vengono controllati a tempo di scrittura.

Mentre i dati dei PF (DDS) sono controllati a tempo di lettura

ONL-i Days 2019 - DB2 for i performance

follow up ➡

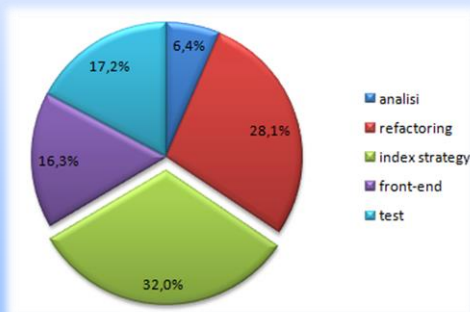
mar 15.00 Duca:
RPG + SQL: l'unione fa
la forza

30

Esempio: da RPG nativo a embedded SQL

- Il peso della **strategia di indicizzazione** per il buon esito del progetto è paragonabile se non superiore allo sforzo di scrittura del "codice RPG"
- Pur con un codice RPG "buono" senza l'analisi di una "buona strategia" di indicizzazione non è detto che si ottengano risultati migliori

decremento tempi di
esecuzione
da -60 a -99%



ONL-i Days 2019 - DB2 for i performance

31

Suggerimenti per lo sviluppo

ONL-i Days 2019 - DB2 for i performance

32

Cause frequenti per performance scarse

- ▀ indici mancanti
- ▀ dati statistici mancanti
- ▀ predicati complessi
- ▀ dati statistici "vecchi" (per indici basati su DDS)
- ▀ modifiche dell'ambiente (dati o sistema)
- ▀ eccessivo I/O
- ▀ riavvii troppo frequenti del sistema

ONL-i Days 2019 - DB2 for i performance

33

Embedded SQL

- massimizzare il riutilizzo degli ODP (soprattutto con SQL dinamici: preparare una sola volta e poi eseguire molte volte)
- effettuare letture/scritture a blocchi di record (multiple row fetch)
- minimizzare le richieste separate di dati al DB2 e privilegiare i join
- ➡ **Think in sets**
- porre massima attenzione al casting dei tipi dati nelle clausole where, nei join e nella proiezione sulle variabili host
- sfruttare la clausola **OPTIMIZE FOR nn ROWS** (da V5R1) che ottimizzare l'esecuzione per restituire più velocemente le prime nn righe

ONL-i Days 2019 - DB2 for i performance

34

Clausole WHERE e JOIN

- evitare se possibile le formule
 - oppure valutare la creazione di indici derivati
- preferire il predicato `LIKE` alla funzione di sottostringa `SUBSTR`
- preferire l'operatore `=` oppure `IN` piuttosto che `<>`
- evitare il casting del tipo dati
- se non richiesto dal disegno applicativo evitare left o right join e privilegiare `inner join`
 - il DB2 ha molte più possibilità di ottimizzare gli inner join piuttosto che i left/right, perché l'ottimizzatore ha la capacità di cambiare l'ordine dell'inner join rispetto a come è stato scritto per massimizzare le performance
 - Da 7.2 sono stati introdotti notevoli miglioramenti anche per gli outer join

ONL-i Days 2019 - DB2 for i performance

SQE - improved performance for Outer Join queries

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20i%20Technology%20Updates/page/SQE%20-%20improved%20performance%20for%20Outer%20Join%20queries>

35

Dati statistici

- Da **V5R2** il DB2 è stato notevolmente migliorato per conservare nell'oggetto tabella i dati statistici necessari all'ottimizzatore
- Tutti i dati statistici sono auto-manutenuti dal DB2 senza la necessità di nessuna operazione da parte del DBA
- Da **7.1** con la tecnologia AQP i dati statistici vengono aggiornati anche durante l'esecuzione della query
- **IMPORTANTISSIMO! Evitare sempre di cancellare gli oggetti tabella, perché andrebbe perso il «prezioso tesoro» dei dati statistici**

ONL-i Days 2019 - DB2 for i performance

36

Indici

- gli indici giocano un ruolo determinante per l'ottimizzatore
- gli indici forniscono:
 - accesso «per chiave»
 - informazioni statistiche sulle chiavi che vengono usate dall'ottimizzatore per scegliere il migliore (=meno costoso) piano di accesso
- IN PRIMIS bisogna focalizzare la propria attenzione sugli indici che coinvolgono le colonne presenti nella clausole di **JOIN** e **WHERE**
- In seconda istanza considerare le clausole di **GROUP BY**
- Quali tipi di indice abbiamo a disposizione nel DB2?

ONL-i Days 2019 - DB2 for i performance

37

Radix index

- **radix index**: ideali per accedere a un piccolo set di record con una chiave ad alta cardinalità
- hanno una struttura ad albero
- il DB2 può accedere con due tipi di operazioni:
 - **probe** (→): quando le colonne che devono essere cercate coincidono con le colonne dell'indice il DB2 può posizionarsi direttamente ai valori ricercati.
P.es. where order = 'B102' and customer = '002'
and item = 'HH-6500'
 - **scan** (↔): quando le colonne che devono essere cercate sono presenti nell'indice, ma non nella sequenza desiderata. Il DB2 deve eseguire una scansione di tutto l'indice sui valori della colonna da cercare.
P.es. where item = 'HH-6500'



CUSTOMER	ORDER	ITEM	RRN
001	B507	AB-2700	0010
001	B607	CD-2000	0005
002	B100	XY-1005	0233
002	B102	AZ-5000	0001
003	B709	HH-6500	0002
004	B043	HH-6500	0077
...	...	...	...

ONL-i Days 2019 - DB2 for i performance

fonte: IBM

- **encoded vector index:** ideali per accedere a un set più consistente di record con un chiave a bassa cardinalità
- questa **tecnologia è proprietaria di IBM**, sviluppata dai laboratori di ricerca per superare i limiti della tecnologia degli indici bitmap presenti in altri db
- EVI ha una **symbol table** che «assegna» un codice per ogni valore distinto della chiave e un **vettore** che «abbina» questi codici agli RRN della tabella.
- si può creare solo tramite SQL


```
CREATE ENCODED
VECTOR INDEX myevi
ON sales(state)...
```

The diagram illustrates the internal structure of an Encoded Vector Index (EVI). It consists of two main components: a Symbol Table and a Vector table.

Symbol Table: This table maps Key Values to Codes and Counts. It has three columns: Key Value, Code, and Count.

Key Value	Code	Count
Arizona	1	5000
Arkansas	2	7300

Vector: This table maps Codes to Include/Exclude flags and Sum values. It has three columns: Code, Include, and Sum.

Code	Include	Sum
1	1500	2005
2	3200	450

The diagram shows that the Symbol Table and Vector table are linked by the Code column. The Symbol Table's Count column represents the total number of records for a given Key Value, while the Vector table's Include and Sum columns provide more granular information about the distribution of records across different categories (e.g., by state).

```
CREATE ENCODED
VECTOR INDEX myevi
ON sales(state)...
```

Symbol Table					Vector	RRN
Key Value	Code	Count	Include Sum()	Include Sum()		
Arizona	1	5000	1500	2005	1	1
Arkansas	2	7300	3200	450	2	2
...					3	3
Wisconsin	49	340	575	1200	4	4
Wyoming	50	2760	210	0	5	5
					6	6
					7	7
					8	8
					9	9
					...	...

optional

fonte: IBM

fonte: IBM

Opzioni in QAQQINI: ALLOW EVI ONLY ACCESS

Alta cardinalità: un elevato numero di valori distinti

Bassa cardinalità: un piccolo numero di valori distinti

41

Derived e sparse indexes

da 6.1

- sia radix index sia EVI possono includere **chiavi derivate**
- una **chiave derivata** significa che l'indice è costruito sul risultato di una funzione applicata alla colonna chiave

```
CREATE INDEX clienti_nome  
ON clienti (UPPER(rag_soc))
```

- se alla definizione dell'indice si aggiunge anche una clausola **where** si ottiene quello che viene definito **sparse index**

```
CREATE INDEX clienti_attivi  
ON clienti (CL_ID)  
WHERE stato = 'A'
```

N.B. solo da 7.1 l'ottimizzatore sfrutta i sparse index. In 6.1 di fatto costituiscono solo un metodo per rimpiazzare tramite SQL le viste logiche con regole di selezione create con le DDS

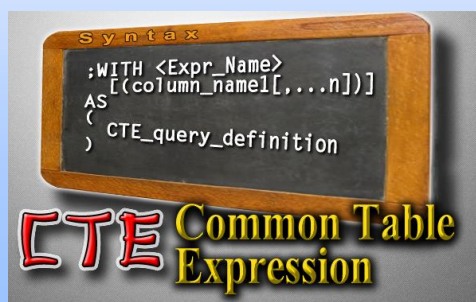
ONL-i Days 2019 - DB2 for i performance

42

Common Table Expression (CTE)

da V5R4

- Le CTE sono uno strumento molto potente per gestire uno o più resultset volatili in una sola istruzione SQL
- Le CTE consentono anche di scrivere istruzioni SQL ricorsive (p.es. esplosione distinta base)
- Le CTE sono **sempre** da **prediligere** rispetto ad altre soluzioni che utilizzano tabelle temporanee



ONL-i Days 2019 - DB2 for i performance

43

CTE: esempio

```

with DINFO (DEPTNO, AVGSALARY, EMPCOUNT) as
  (select OTHERS.WORKDEPT, AVG(OTHERS.SALARY), count(*)
   from DB2SAMPLE/EMPLOYEE OTHERS
   group by OTHERS.WORKDEPT),
 DINFOMAX AS
  (select MAX(AVGSALARY) AS AVGMAX
   from DINFO)
select THIS_EMP.EMPNO as "Codice",
       dec(THIS_EMP.SALARY, 10, 0) as "StipendioImpiegato",
       dec(DINFO.AVGSALARY, 10, 0) as "SalarioMedioDip",
       DINFO.EMPCOUNT as "NumeroImpiegati",
       dec(DINFOMAX.AVGMAX, 10, 0) as "MaxSalarioMedio"
from DB2SAMPLE/EMPLOYEE THIS_EMP, DINFO, DINFOMAX
where THIS_EMP.JOB = 'SALESREP'
      and THIS_EMP.WORKDEPT = DINFO.DEPTNO;

```

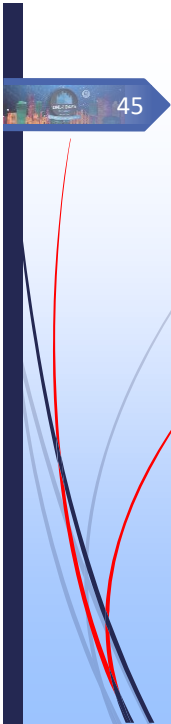
ONL-i Days 2019 - DB2 for i performance

44

Materialized Query Table (MQT) **da V5R3**

- La definizione di una MQT è basata sul risultato di un'istruzione di select
- Si può pensare a una MQT come a una vista (view) con i dati "materializzati"
- La MQT contiene una **copia** dei dati che sono il risultato della query
- I dati **non** vengono **aggiornati automaticamente**, ma bisogna eseguire l'istruzione **refresh table**
- Le MQT possono essere utilizzate dal DB2 per migliorare le performance delle query

ONL-i Days 2019 - DB2 for i performance



MQT creazione

CREATE TABLE -MQT name- **AS**

(SELECT -grouping columns-
-aggregate/summary columns-
FROM -table(s)
WHERE -selection columns-
GROUP BY -grouping columns-
ORDER BY -ordering columns-

← Query that defines and populates MQT

DATA INITIALLY IMMEDIATE
DATA INITIALLY DEFERRED

← Data is inserted into the MQT when it is created
← Data is not inserted into the MQT when it is created

REFRESH DEFERRED
REFRESH IMMEDIATE

← Data in the MQT can be refreshed at any time using the REFRESH TABLE statement
← Not currently supported

ENABLE QUERY OPTIMIZATION
DISABLE QUERY OPTIMIZATION

← MQT can be used for query optimization
← MQT cannot be used for query optimization but the MQT can be queried directly

MAINTAINED BY USER
MAINTAINED BY SYSTEM

← MQT is maintained by the user with INSERT, DELETE, UPDATE, or REFRESH TABLE
← Not currently supported

ONL-i Days 2019 - DB2 for i performance

46

MQT: perché?

- Migliorare le performance
- In particolare quando ci sono query complesse (join multipli, funzioni di aggregazione...) su una grande mole di dati
- Una “buona idea” per usare le MQT come *tabelle aggregate* e/o *tabelle denormalizzate* (soprattutto in ambito della business intelligence)
- Nella query “complessa” non è strettamente necessario fare riferimento esplicito alla MQT, perché è l’ottimizzatore del DB2 che userà l’MQT se lo riterrà opportuno

ONL-i Days 2019 - DB2 for i performance

Opzioni in QAQQINI: `MATERIALIZED_QUERY_TABLE_REFRESH_AGE`,
`MATERIALIZED_QUERY_TABLE_USAGE`

47

Tabella conversione date/1

- E' una tabella "calendario" che contiene un record per ogni giorno in uno specifico range di date.
- Ogni record contiene la stessa data rappresentata in diversi modi (data giuliana, data "vera", giorno della settimana, data YYYYMMDD, data DDMMYY, excel...)
- Risolve il problema di convertire le date «legacy» in campi data «veri» per sfruttare tutte le potenzialità del DB2 sui campi data
- Dal redbook SG24-7214-01 "Getting Started with DB2 Web Query for i" (appendix B p. 542-552)
Per gli iscritti a ONL-i Days in allegato lo script localizzato per l'Italia e arricchito: MK1DATECNV.ZIP



ONL-i Days 2019 - DB2 for i performance

48

Tabella conversione date/2



LEGACY_ORDER_HEADER file →

ORDER	CUST	CRDOAT	SHPOAT	SHIPVIA
1168336	P8311	1022005	7232005	FedEx
7073849	I7472	1022005	9032005	Pick Up
1168336	P8311	1022006	7232006	FedEx
7073849	I7472	1022006	9032006	Pick Up
1168336	P8311	1022007	7232007	FedEx
7073849	I7472	1022007	9032007	Pick Up
1730932	E9747	1042005	10122005	LPS
8211860	P3751	1042005	8242005	Pick Up
1730932	E9747	1042006	10122006	LPS
8211860	P3751	1042006	8242006	Pick Up
1730932	E9747	1042007	10122007	LPS
8211860	P3751	1042007	8242007	Pick Up

Date conversion table ↓

Join (RI constraints, SQL view, synonym join, report join)

DC_DATE	DC_JDE_JULIAN_DATE	DC_MDYY_DEC	DC_MDYY_CHAR	DC_YEAR	DC_DOW	DC_DOY	DC_WOY	DC_QOY	DC
2005-01-01	106001	1042006	1012006	2006	7	1	1	1	1
2005-01-02	106002	1022006	1022006	2006	1	2	1	1	1
2005-01-03	106003	1032006	1032006	2006	2	3	1	1	1
2005-01-04	106004	1042006	1042006	2006	3	4	1	1	1
2005-01-05	106005	1052006	1052006	2006	4	5	1	1	1
2005-01-06	106006	1062006	1062006	2006	5	6	1	1	1
2005-01-07	106007	1072006	1072006	2006	6	7	1	1	1
2005-01-08	106008	1082006	1082006	2006	7	8	2	1	1
2005-01-09	106009	1092006	1092006	2006	1	9	2	1	1
2005-01-10	106010	1102006	1102006	2006	2	10	2	1	1
2005-01-11	106011	1112006	1112006	2006	3	11	2	1	1
2005-01-12	106012	1122006	1122006	2006	4	12	2	1	1

ONL-i Days 2019 - DB2 for i performance

fonte: IBM

49

Tabelle di inclusione/esclusione

- Non bisogna sottovalutare la potenza dei join ben costruiti con gli indici appropriati sulle colonne di join
- Esempio tipico: una tabella di anagrafica grande con un campo «flag annullamento» che può assumere un valore booleano
- Si possono ottenere risultati molto più performanti con un exception join tra la tabella anagrafica e una tabella che contiene l'elenco degli ID dei record annullati piuttosto che una clausola where sul campo «flag annullamento» della tabella anagrafica

ONL-i Days 2019 - DB2 for i performance

50

Strumenti di analisi

ONL-i Days 2019 - DB2 for i performance

51

Registrazione dati per benchmark

- ▀ esempio UPBCMK
- ▀ pro: scelgo esattamente quali eventi intercettare e quali dati registrare
- ▀ contro: richiede la modifica del sorgente
- ▀ si consiglia di sfruttare le direttive di compilazione condizionale
- ▀ *Per gli iscritti a ONL-i Days in allegato un service program di esempio per registrare dati di benchmark: setRegLog.zip*

ONL-i Days 2019 - DB2 for i performance

52

Messaggi di debug

- ▀ Eseguire statement SQL con il debug attivo (**STRDBG**) causa la registrazione nel joblog di diversi messaggi inviati dall'ottimizzatore
- ▀ Questi messaggi forniscono principalmente informazioni e suggerimenti sugli indici utilizzati o consigliati
- ▀ Leggere bene in questi messaggi il testo di secondo livello

ONL-i Days 2019 - DB2 for i performance

Esegui script SQL

- Al termine di ogni istruzione SQL viene restituito l’esito e il tempo di esecuzione

```
[ Tue May 21 14:25:49 CEST 2019 ] Esegui selezionato...
(Me: ...
Fatt: ...
P.F.: ...
sum: ...
✓ Istruzione eseguita con esito positivo (3.229 ms = 3,229 sec)
```

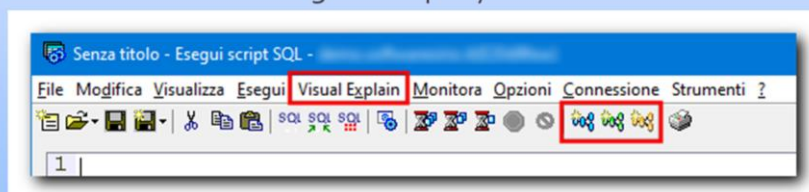
follow up ⇄
mar 15.00 Pagano:
ACS principali
funzionalità

54

Visual explain

da V4R5

- Visual Explain è LO strumento principale da utilizzare per il «query tuning»
- è disponibile in Esegui script SQL e fornisce numerose informazioni sull'esecuzione di uno statement SQL
- E' possibile eseguire la query e contestualmente ottenere le informazioni di Visual Explain oppure si possono solo visualizzare le informazioni senza eseguire la query



ONL-i Days 2019 - DB2 for i performance

E' uno strumento ricco di informazioni.

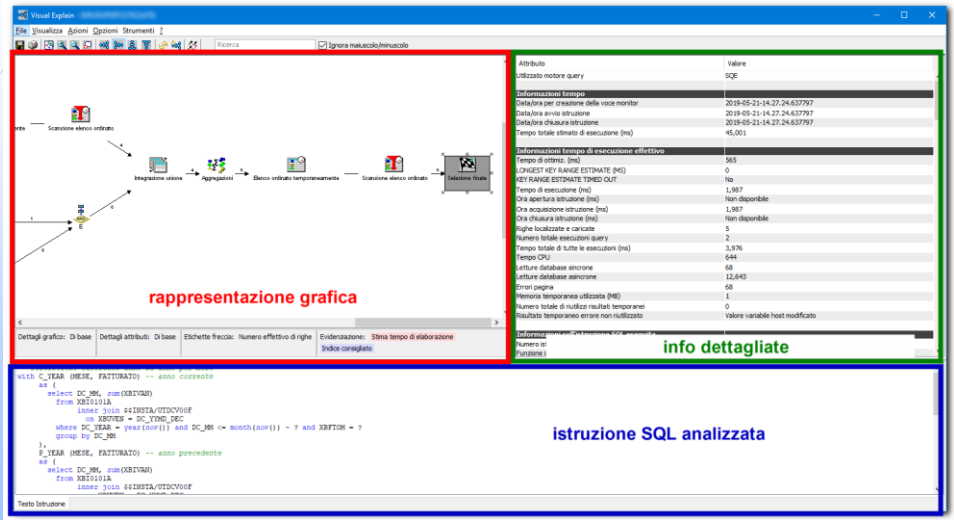
Per chi vuole affrontare seriamente la sfida dell'ottimizzazione delle performance di una query, è assolutamente indispensabile.

Bisogna allenarsi ad usarlo e esplorare tutte le informazioni che fornisce.

Il redbook *DB2 UDB for AS/400 Visual Explain - Illustrating the Secrets* anche se datato 2001 è una buona lettura e guida pratica.

55

Visual explain: esempio



ONL-i Days 2019 - DB2 for i performance

56

Visual Explain: icone principali/1



Scansione tabella indica che tutte le righe nella tabella sono state impaginate e che i criteri di selezione sono stati applicati a ogni riga. Su queste righe vengono richiamati i criteri di selezione.



Sonda indice (*index probe*) indica che i criteri di selezione i cui predicati corrispondevano alle colonne chiave principali dell'indice sono stati utilizzati per sondare direttamente l'indice. La selezione sonda minimizzerà il numero di voci chiave che devono essere elaborate e impaginate nella memoria principale.



Scansione elenco numero righe (RRN) indica che l'intero elenco numero di righe verrà sottoposto a scansione e tutte le voci saranno elaborate. La scansione di un elenco numero di righe può fornire un elevato numero di salvataggi per i dati tabella associati all'elenco numero di righe temporaneo. Poiché la struttura dati dell'elenco numero di righe temporaneo garantisce che i numeri di righe vengano ordinati, viene eseguito un attento mirroring del layout del numero di righe dei dati della tabella assicurando che l'impaginazione sulla tabella non analizzerà due volte la stessa pagina di dati.



Integrazione bitmap indica che più bitmap o oggetti elenco numero di righe sono stati integrati e/o combinati per formare una bitmap finale o un oggetto elenco numero di righe.



Sonda tabella indica che i dati contenuti in una tabella devono essere elaborati e selezionati per questa query. La tabella viene sondata utilizzando una chiave derivata dal numero ordinale o dal numero record relativo associato a ciascuna riga nella tabella.

ONL-i Days 2019 - DB2 for i performance

57

Visual Explain: icone principali/2



Tabella casuale distinta temporanea indica che una tabella casuale distinta temporanea è stata creata per eseguire l'elaborazione casuale. Una tabella casuale distinta rappresenta una struttura di dati identica alla tabella casuale temporanea con la sola eccezione dei dati duplicati che saranno compressi fuori dalla tabella temporanea creata. La tabella casuale risultante può quindi essere utilizzata per eseguire operazioni distinte o aggregate per la query



Scansione tabella casuale indica che l'intera tabella casuale temporanea verrà sottoposta a scansione e tutte le voci in essa contenute saranno elaborate.

ONL-i Days 2019 - DB2 for i performance

58

Visual explain: info

- Posizionando il mouse su un'icona compare un tooltip con le info più importanti
- Nell'area laterale destra vengono mostrate tutte le info per l'icona selezionata
- Con clic destro su un'icona compare il menu contestuale

ONL-i Days 2019 - DB2 for i performance

Integrazione bitmap

Tempo cumulativo (ms)	1.171,275
Costo CPU (ms)	2,175
costo I/E (ms)	0,0
Conteggio I/E	0,0

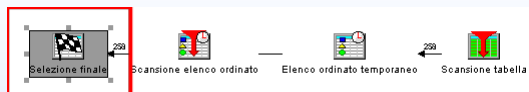
Attributo	Valore
Nome tabella, nome tabella di ba...	
Nome della tabella in	TRAMO
Libreria della tabella in consulta...	\$\$LIBF
Membro della tabella in consulta...	TRAMO
Nome lungo della tabella in cons...	TRAMO
Libreria lunga della tabella in co...	\$\$LIBF

Definizione tabella

- Dati statistici
- Descrizione tabella
- Visualizza Indici
- Visualizza Tabelle query materializzate
- Visualizza correlati
- 2

59

Visual explain: info selezione finale



- **Utilizzato motore query:** indica se viene usato CQE o SQE
- **Tempo totale stimato di esecuzione (ms)**
- Letture database
- Testo istruzione e Valori variabile host
- Libreria e tabella opzioni query
- Utilizzo tabella query materializzata (MQT)
- Utilizzo AQP

ONL-i Days 2019 - DB2 for i performance

60

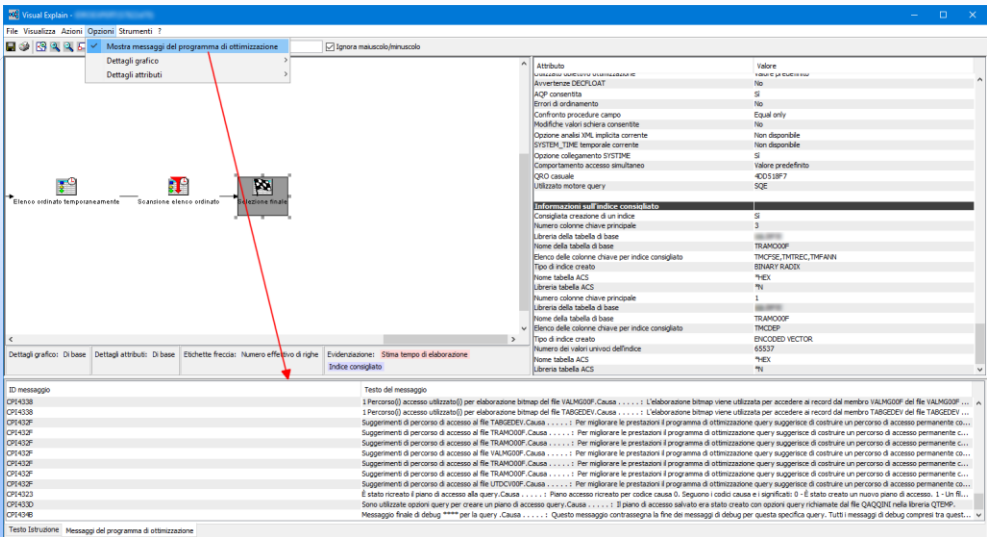
Visual explain: info selezione passi intermedi

- tempo di elaborazione parziale e cumulativo (ms)
- righe tabella e righe selezionate
- % selettività
- **costo CPU vs costo I/E**
- **indici consigliati**

ONL-i Days 2019 - DB2 for i performance

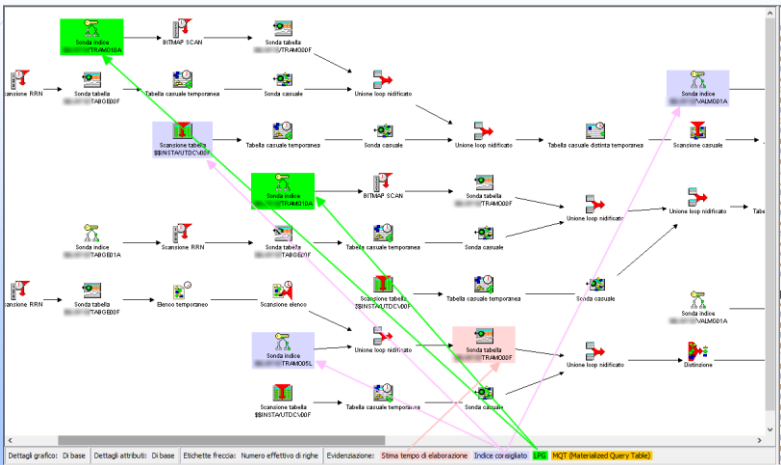
61

Visual explain: messaggi ottimizzatore



62

Visual explain: evidenziazione suggerimenti



ONL-i Days 2019 - DB2 for i performance

Index advisor da V5R4 è sempre attivo

► Dal menu Azioni scegliere la voce Advisor. Visualizza tutti gli indici consigliati e permette di crearli direttamente

Advisor delle statistiche e dell'indice

Advisor dell'indice | Advisor statistiche

I seguenti indici sono stati consigliati:

Crea	Nome tabella	Schema	Tipo indice	Colonne	Sequenza ordinamento
☒	TRANCOOP	DB/ADM	Radice binaria	THCPSE, THNREC, THFANN	Nessuno (Ordina per valore esadecimale)
☒	TRANCOOP	DB/ADM	Radice binaria	THNDEP	Nessuno (Ordina per valore esadecimale)
☒	VALMCOOP	DB/ADM	Radice binaria	VACART	Nessuno (Ordina per valore esadecimale)
☒	UTDCOOP	DB/ADM	Radice binaria	DC_YM, DC_DD, DC_YEAR, DC_YMD_DEC	Nessuno (Ordina per valore esadecimale)
☒	TABCOOP	DB/ADM	Radice binaria	TAMNIN, TACHIA	Nessuno (Ordina per valore esadecimale)
☒	TRANCOOP	DB/ADM	Radice binaria	THCPSE, THNREC, THNDEP, THFANN	Nessuno (Ordina per valore esadecimale)
☒	TRANCOOP	DB/ADM	Radice binaria	THCPSE, THNREC, THNDEP, THFANN	Nessuno (Ordina per valore esadecimale)

Visualizza SQL Crea...

Informazioni sull'indice consigliato

Consigliata creazione di un indice	SI
Numero colonne chiave principale	3
Libreria della tabella di base	DB/ADM
Nome della tabella di base	TRANCOOP
Nome della tabella di base	THNDEP
Nome della tabella ACS	THN
Numero colonne chiave principale	1
Libreria della tabella di base	DB/ADM
Nome della tabella di base	TRANCOOP
Nome della tabella di base	THNDEP
Nome della tabella ACS	THN

```
/* Creazione di TRANCOOP_IDX [Indice]
Durante la creazione di questo indice, la connessione al database deve avere una sequenza di ordinamento 'HEX'.
create encoded vector index TRANCOOP_IDX
on TRANCOOP (THNDEP asc) with 65537 distinct values unit any keep in MEMORY no;
/* Impostazione testo etichetta per TRANCOOP_IDX */
label on index TRANCOOP_IDX is 'Indice generato da Advisor dell'indice';
```


64

Index advisor

- Dalla gestione schemi, tramite clic destro su un oggetto tabella si possono visualizzare tutte le voci di index advisor suggerite dall'ottimizzatore

Tabella per cui è stato consigliato l'indice	Schema sistema	Chiavi consigliate	Indipendente dall'ordine delle chiavi principali	Tipo di indice consigliato	Ultimo consigliato per utilizzo query	Numero di volte consigliato per utilizzo query	Tempo creazione indice stimato	Motivo consigliato	Dimensione pagina logica consigliata	Stima query più costosa	Stima media query	Righe in tabella quando consigliato
...	...	TMTDOR, TMTREC, TMCNOA, TMHREG, TMARE...	TMTDOR, TMTREC...	Radice binaria	27/05/2019 18:24:08	461472	00:02:17	Selezione riga	64	1	0.0000	9060090
...	...	TMTREC, TMFANN	TMTDOR, TMTREC...	Radice binaria	27/05/2019 23:34:48	249009	00:01:07	Selezione riga	64	193	0.2371	9022090
...	...	TMTDOR, TMHREG, TMAREG, TMORE1, TMGRE...	TMTDOR, TMHRE...	Radice binaria	27/05/2019 15:57:54	168472	00:01:29	Selezione riga	64	1	0.0000	9059090
...	...	TMTNOV, TMTDOR, TMORE1, TMHURE, TMHRE...	TMTNOV, TMTDO...	Radice binaria	27/05/2019 18:30:43	99922	00:02:01	Selezione riga	64	1	0.0000	9059090
...	...	TMTDOR, TMORE1, TMHURE, TMTREC	TMTDOR, TMORE...	Radice binaria	27/05/2019 18:30:43	64854	00:00:53	Ordinamento/...	64	716	0.2639	8970090

- Si può interrogare direttamente la tabella `qsys2/sysixadv` con le informazioni di Index advisor:

```
select * from QSYS2/SYSIXADV
where LAST ADVISED >= timestamp('2019-06-10', '00.00.00');
```

ONL-i Days 2019 - DB2 for i performance

Index advisor: un consiglio non un obbligo

Per valutare se e quali consigli seguire ricordarsi bene che:

- è basato su una singola istruzione
- assume che esista un'alta selettività
- assume che usare un indice per accedere ai dati è la soluzione migliore

Creare un indice per volta ed eseguire nuovi test anche comparativi

Va usato da chi conosce il modello dei dati e il disegno applicativo

Massima attenzione agli indici consigliati su colonne coinvolte nei JOIN o nella clausola WHERE

Ricordarsi che l'indice non serve all'ottimizzatore solo a scopo di selezione dei record per chiave anche per i dati statistici necessari a «conoscere» il dato contenuto nelle tabelle

65

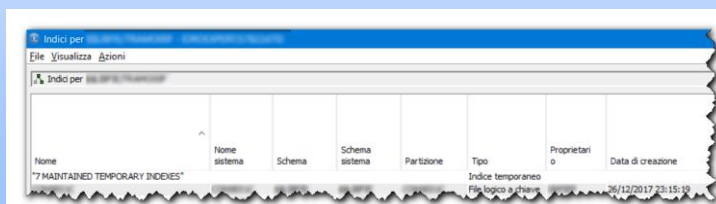
Index advisor e MTI

da V5R4

- MTI: **M**aintained **T**emporary **I**ndexes
- sono creati ed utilizzati da SQE al fine di ottimizzare le query, se non esiste un indice permanente soddisfacente
- all'IPL vengono cancellati
- visualizzando gli indici collegati ad una tabella viene anche mostrato il numero di indici temporanei
- sono mantenuti aggiornati dal DB2
- sono usati solo per la selezione e non per i dati statistici

self tuning

ONL-i Days 2019 - DB2 for i performance



The screenshot shows the 'Indici per' window in the DB2 GUI. It displays a table of maintained temporary indexes. The table has columns for Name, Nome sistema, Schema, Schema sistema, Partizione, Tipo, Proprietario, and Data di creazione. The first row shows '7 MAINTAINED TEMPORARY INDEXES'.

Nome	Nome sistema	Schema	Schema sistema	Partizione	Tipo	Proprietario	Data di creazione
7 MAINTAINED TEMPORARY INDEXES							

Auto-valutazione degli indici (index evaluator)

Nome sistema	Schema sistema	Tipo	Ultimo utilizzo query	Ultimo utilizzo statistiche query	Conteggio utilizzo query	Conteggio utilizzo statistiche query	Ultimo utilizzo	Conteggio giorni di utilizzo	Data di reimpostazione conteggio giorni di utilizzo
FILELOGIC	DB2SPD	File logico a chiave	27/05/2019 18:23:09	27/05/2019 18:23:09	353.644	470.723	27/05/2019 00:00:00	113	
FILELOGIC	DB2SPD	File logico a chiave	24/05/2019 16:43:54	27/05/2019 21:40:19	103.755	463.261	24/05/2019 00:00:00	33	
FILELOGIC	DB2SPD	File logico a chiave	27/05/2019 16:00:54	27/05/2019 18:23:09	5.013	464.826	27/05/2019 00:00:00	147	
FILELOGIC	DB2SPD	File logico a chiave	27/05/2019 18:14:11	27/05/2019 18:23:09	5.013	468.409	27/05/2019 00:00:00	166	
FILELOGIC	DB2SPD	File logico a chiave	27/05/2019 23:58:57	27/05/2019 23:58:57	3.045	3.697	28/05/2019 00:00:00	171	
FILELOGIC	DB2SPD	File logico a chiave	27/05/2019 23:51:42	27/05/2019 23:58:57	856	4.469	28/05/2019 00:00:00	154	
FILELOGIC	DB2SPD	File logico a chiave	27/05/2019 15:12:19	28/05/2019 00:00:05	804	472.522	27/05/2019 00:00:00	142	
FILELOGIC	DB2SPD	File logico a chiave	27/05/2019 15:07:58	28/05/2019 00:00:05	537	476.360	27/05/2019 00:00:00	141	

- Dalla gestione schemi visualizzando gli indici collegati a una tabella si possono esaminare i dati statistici di utilizzo dell'indice
- Questi dati statistici si riferiscono all'uso di SQE
Attenzione che l'accesso nativo non viene considerato

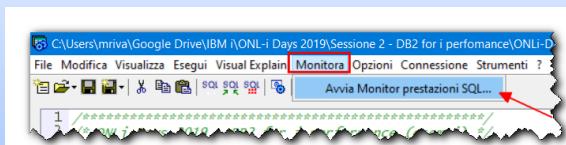
ONL-i Days 2019 - DB2 for i performance

The Index Evaluator's query usage metrics can be accessed programmatically by querying one of the following catalog views in QSYS2: `SYSINDEXSTAT`, `SYSPARTITIONINDEXES`, or `SYSTABLEINDEXSTAT`.
The DSPFD command does not return the query usage metrics for indexes or keyed logical files.

67

Database monitor

- Il database monitor è uno strumento che raccoglie informazioni durante l'esecuzione delle query
- Si avvia da esegui script SQL con «Avvia monitor prestazioni SQL...» oppure con il comando `STRDBMON`

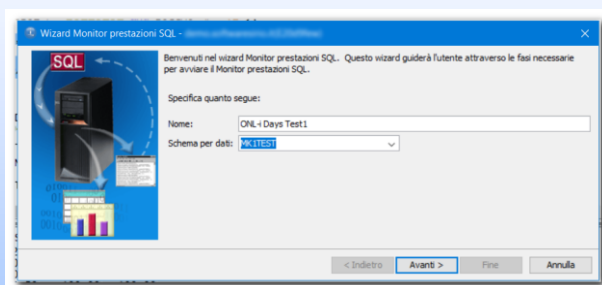


ONL-i Days 2019 - DB2 for i performance

68

Monitor prestazioni SQL: raccolta dati/1

- Assegnare un nome e scegliere una libreria in cui memorizzare i dati raccolti



ONL-i Days 2019 - DB2 for i performance

Monitor prestazioni SQL: raccolta dati/2

Impostare filtri per la raccolta dati

Wizard Monitor prestazioni SQL - [Vedere informazioni...](#)

Per impostare la quantità di dati raccolti, specificare quali filtri utilizzare. Quando si forniscono dei filtri, solo le situazioni corrispondenti ai valori filtro specificati vengono prelevate.

Se si desidera limitare la quantità di dati raccolti specificare quali filtri utilizzare:

- ☐ Numero iniziale di record:
- ☐ Tempo di esecuzione minimo stimato per query:
- ☐ Memoria temporanea minima stimata:
- ☐ Profilo utente o gruppo corrente:
- ☐ Limiti governi della query:
- ☐ Risultato dell'istruzione o codice SQL:
- ☐ Valori variabile host:

Registri client

- ☐ Stringa di account:
- ☐ Nome applicazione:
- ☐ Nome programma:
- ☐ Utente client:
- ☐ Stazione di lavoro:

☐ Istruzioni che accedono a tali oggetti:

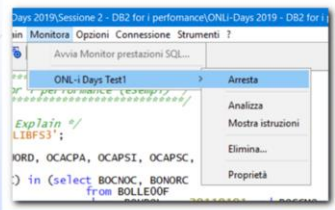
Schema	Nome

Attività da monitorare

- ☒ Raccogli solo emissioni monitor per attività utente
- ☐ Raccogli emissioni monitor per utente e attività sistema

< Indietro Avanti > Fine Annulla

Monitor prestazioni SQL: raccolta dati/3



- eseguire la query
- arrestare il monitor
- analizzare i dati del monitor

The screenshot shows the 'ONL-i Days Test1' monitor with the 'Analisi dati prestazioni SQL' (SQL Performance Analysis) window open. The window displays various performance metrics and their values. The 'Categorie' (Categories) dropdown is set to 'Panoramica' (Overview). The table below lists the metrics and their values:

Metrica	Valore
Quantità di lavoro richiesto?	0,000
Quanti sono le opz. fornite al prog. di ottimizzazione?	0,002198
Quale implementazione ha usato il prog. di ottimiz.?	1,00
Quanti sono i tipi di istruzioni SQL richieste?	1,00
Informazioni varie	1 < Seleziona un prospetto >
Media righe restituite	0
Tempo di esecuzione medio	15 < Seleziona un prospetto >
Grado parallelo medio utilizzato	0
Grado parallelo massimo	1 < Seleziona un prospetto >
SQLC	0
Denominazione sistema	1 < Seleziona un prospetto >
Denominazione SQL	0
Istruzioni Aperture univoche	1 < Seleziona un prospetto >
Aperture complete	1 < Seleziona un prospetto >
Pseudoaperture	0
Scansioni tabella	1 < Seleziona un prospetto >
Media MQT utilizzate	0,000
Media indici utilizzati	0,000
Indici completi creati	0

I dati vengono scritti in una tabella QZG...

AQP: i record 5005 e 5007 documentano l'attività di AQP

Centro prestazioni SQL

da V_5R_4 [illegible]

Disponibile in IBM Access Client Solutions

Consente di interrogare tutti i piani di accesso delle query eseguite dall'ottimizzatore
SQE

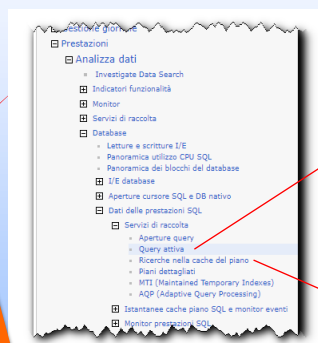
L'elenco può essere filtrato per vari criteri (data, ora, utente, oggetti coinvolti...)

Dall'elenco delle esecuzioni si può facilmente accedere al Visual Explain corrispondente

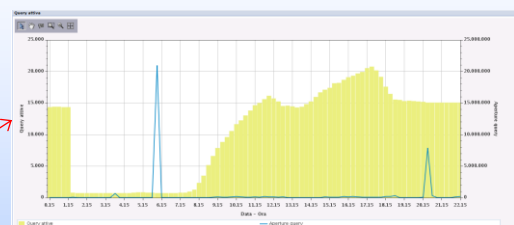
72

Navigator for i – analisi prestazioni

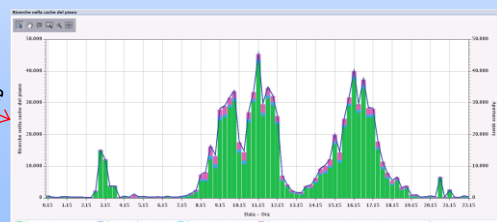
[http://\[indirizzo\]:2004/ibm/console/login.do?action=secure](http://[indirizzo]:2004/ibm/console/login.do?action=secure)



query active



plan cache

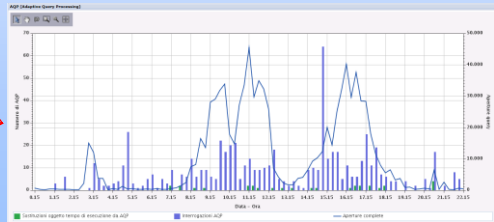
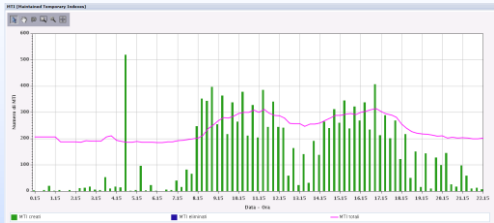
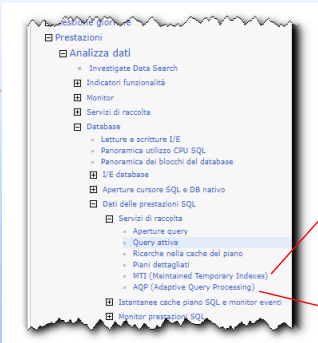


follow up
mar 11.30
Pacchiarini:
Strumenti di analisi
health & performance

ONL-i Days 2019 - DB2 for i performance

73

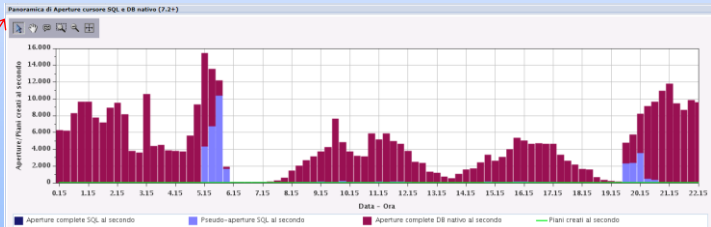
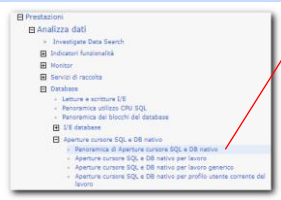
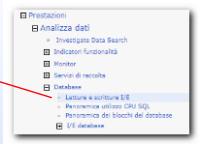
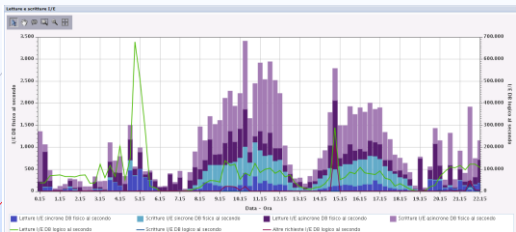
Performance: AQP e MTI



ONL-i Days 2019 - DB2 for i performance

74

Performance: tipo di accesso al dato



ONL-i Days 2019 - DB2 for i performance



il catalogo del DB2 è una ricca fonte di informazioni sulla struttura del database, sul suo contenuto e sul suo utilizzo.

Molte informazioni sono disponibili anche tramite il comando DSPFD, ma le viste del catalogo sono più ricche e più comode da interrogare

76

Statistiche su tabelle e membri (partizioni)

- contengono informazioni statistiche sul contenuto della tabella (numero record, record cancellati, dimensione) e sui dati di accesso (numero aperture, chiusure, letture fisiche/logiche, aggiornamenti...)
- aiutano a capire quali sono le tabelle più "sensibili" e che meritano una maggior attenzione per l'analisi dell'ottimizzazione
 - SYSPARTITIONSTAT: per statistiche membri
 - SYSTABLESTAT: per statistiche tabelle

TABLE	LAST_USED	DAYS_USED_COUNT	NUMBER_ROWS	NUMBER_DELETED_ROWS	Data_Size (Mb)	NUMBER_DISTINCT_INDEXES	OPEN_OPERATIONS	CLOSE_OPERATIONS	INSERT_OPERATIONS	UPDATE_OPERATIONS	DELETE_OPERATIONS	LOGICAL_READS	PHYSICAL_READS
2019-05-31	175	0	7147394	0	1840	11	25171653	25172334	101661	13430807	226	20457734450	0
2019-05-31	175	0	7989135	0	2544	6	4101275	4101273	90454	1078449	222	2169034318	0
2019-05-31	175	0	144464	0	54	39	205137711	2060109604	903	284	0	2259524682	0
2019-05-31	175	0	0	0	0	4	18567466	18567465	0	0	0	0	0
2019-05-31	175	35	0	0	0	8	146555	146579	35	1	0	1593	0
2019-05-31	175	1696	48	0	0	8	146415	146410	0	0	0	341199	0
2019-05-31	175	9425	0	3	3	9	146713	146707	259	115	0	108939845	0
2019-05-31	175	24	0	0	0	6	146415	146409	0	0	0	29011	0
2019-05-31	175	2160	0	0	0	10	146972	146967	0	0	0	187826654	0
2019-05-31	175	26	0	0	0	9	146414	146408	0	0	0	72768140	0
2019-05-31	175	13235	0	3	3	13	147098	147092	7	37	0	367024475	0
2019-05-31	175	925	52	0	4	4	125269	125269	8	3	0	55209	0
2019-05-20	15	4	9	0	0	3	0	0	0	0	0	0	0
2019-05-11	0	0	0	0	0	0	0	0	0	0	0	0	0

```
/* STATISTICHE SU TABELLE E MEMBRI (PARTIZIONI) */
select SYSTEM_TABLE_NAME TABLE,
       date(LAST_USED_TIMESTAMP) LAST_USED, DAYS_USED_COUNT,
       NUMBER_ROWS, NUMBER_DELETED_ROWS,
       DECIMAL(DATA_SIZE/1048576, 9, 0) as "Data_Size (Mb)",
       NUMBER_DISTINCT_INDEXES, OPEN_OPERATIONS, CLOSE_OPERATIONS,
       INSERT_OPERATIONS, UPDATE_OPERATIONS, DELETE_OPERATIONS,
       LOGICAL_READS, PHYSICAL_READS, CLEAR_OPERATIONS,
       COPY_OPERATIONS, REORGANIZE_OPERATIONS, INDEX_BUILDS
from QSYS2/SYSPARTITIONSTAT -- per statistiche membri
-- from QSYS2/SYSTABLESTAT -- per statistiche file
where TABLE_SCHEMA = 'DB2SAMPLE'
order by SYSTEM_TABLE_NAME;
```

77

Stato di «salute» del DB2

- Informazioni generali sull'intero database (numero schemi, tabelle...)
`call qsys2/Health_Database_Overview(1, '%', null, null, null);`
- informazioni sulle attività di tutti gli oggetti nello schema DB2SAMPLE per un massimo di 10 oggetti per ogni attività (inserimenti, aggiornamenti, cancellazioni, letture fisiche/logiche...)
`call qsys2/Health_Activity(1, 0, 'DB2SAMPLE', '%', 10, null, null, null);`
- informazioni sui limiti dimensionali di tutti gli oggetti il cui nome comincia con E nello schema DB2SAMPLE per un massimo di 2 oggetti per ogni tipologia di limite
`call qsys2/Health_Size_Limits(1, 0, 'DB2SAMPLE', 'E%', 2, null, null, null);`

ONL-i Days 2019 - DB2 for i performance

Index advisor e MTI

- Informazioni sugli indici consigliati dove un MTI è stato utilizzato a partire dall'ultimo IPL
- e' importante tenere sotto osservazione gli MTI usati più spesso e più recentemente, perché potrebbero essere buoni candidati per essere promossi a indici residenti

MTI_USED	MTI_CREATED	TABLE_NAME	TABLE_SCHEMA	KEY_COLUMNS_ADVISED	LEADING_COLUMN_KEYS
429091042	1	DEPOSITO	SC	DEPOSITO, SOC, NOMEFILE	DEPOSITO, SOC, NOMEFILE
160709660	0	FLAG_FORNABITUALE	SC	FLAG_FORNABITUALE, FLAG_ANNULLAMENTO, CODICE_ARTICOLO	FLAG_FORNABITUALE, FLAG_ANNULLAMENTO, CODICE_ARTICOLO
159082082	1	FLAG_STATO_LISTINO	SC	FLAG_STATO_LISTINO, FLAG_FORNABITUALE, FLAG_ANNULLAMENTO, CODICE_ARTICOLO, DATA_ENTRATA_V...	FLAG_STATO_LISTINO, FLAG_FORNABITUALE, FLAG_ANNULLAMENTO, CODICE_ARTICOLO, DATA_ENTRATA_V...
44147851	1	FLAG_ANNULLAMENTO	SC	FLAG_ANNULLAMENTO, FLAG_FORNABITUALE, CODICE_ARTICOLO, DATA_ENTRATA_V...	FLAG_ANNULLAMENTO, FLAG_FORNABITUALE, CODICE_ARTICOLO, DATA_ENTRATA_V...
44147453	0	FLAG_FORNABITUALE	SC	FLAG_FORNABITUALE, FLAG_ANNULLAMENTO, CODICE_ARTICOLO	FLAG_FORNABITUALE, FLAG_ANNULLAMENTO, CODICE_ARTICOLO
9163669	19	DEP, ARTICOLO, DISPRO	SC	DEP, ARTICOLO, DISPRO	DEP, ARTICOLO, DISPRO
1169987	196	YR_FLAGANNULLAMENTO, YR_FILIALE, YR_TIPOPAGAMENTO, YR_NUMRATE	SC	YR_FLAGANNULLAMENTO, YR_FILIALE, YR_TIPOPAGAMENTO, YR_NUMRATE	YR_FLAGANNULLAMENTO, YR_FILIALE, YR_TIPOPAGAMENTO, YR_NUMRATE
1006634	5	BOTD01, BOCDEP, BONLOT, BOCART, BOFANN	SC	BOTD01, BOCDEP, BONLOT, BOCART, BOFANN	BOTD01, BOCDEP, BONLOT, BOCART, BOFANN
618814	9	DEP, ARTICOLO, DISPRO	SC	DEP, ARTICOLO, DISPRO	DEP, ARTICOLO, DISPRO
531187	28	TP_CLI_FOR_POTENZ, COD_INDIRIZZO_DI_SPEDIZIONE, NR_PROGRESSIVO_RUBRICA, COD_CLI_FOR_POTENZ	SC	TP_CLI_FOR_POTENZ, COD_INDIRIZZO_DI_SPEDIZIONE, NR_PROGRESSIVO_RUBRICA, COD_CLI_FOR_POTENZ	TP_CLI_FOR_POTENZ, COD_INDIRIZZO_DI_SPEDIZIONE, NR_PROGRESSIVO_RUBRICA, COD_CLI_FOR_POTENZ
380986	23	DEP, ARTICOLO, DISPRO	SC	DEP, ARTICOLO, DISPRO	DEP, ARTICOLO, DISPRO
367617	4	BOFTOM, BOTD01, BONINT, BONFAT, BONBOL, BOCIVA	SC	BOFTOM, BOTD01, BONINT, BONFAT, BONBOL, BOCIVA	BOFTOM, BOTD01, BONINT, BONFAT, BONBOL, BOCIVA
323116	4	NBTD0C, NBCNBO, NBCNFA, NBNFAT, NBNBOL	SC	NBTD0C, NBCNBO, NBCNFA, NBNFAT, NBNBOL	NBTD0C, NBCNBO, NBCNFA, NBNFAT, NBNBOL
320302	17	BONLOT, BOTD01, BOCDEP, BOCART, BOFANN	SC	BONLOT, BOTD01, BOCDEP, BOCART, BOFANN	BONLOT, BOTD01, BOCDEP, BOCART, BOFANN
267129	168	OBSOLETE	SC	OBSOLETE	OBSOLETE

```
with last_ipl(ipl_time) as
  (select job_entered_system_time
   from table(qsys2.job_info(job_status_filter => '*ACTIVE',
                             job_user_filter   => 'QSYS')) x
   where job_name = '000000/QSYS/SCPF')
select mti_used, mti_created, table_name, table_schema,
       key_columns_advised, leading_column_keys, index_type,
       varchar_format(last_advised, 'DD/MM/IYYY - HH24.MI')
       last_advised, times_advised,
       varchar_format(first_advised, 'DD/MM/IYYY - HH24.MI')
       first_advised
from last_ipl, qsys2.sysixadv i
where last_mti_used > ipl_time or
      last_mti_used_for_stats > ipl_time
order by mti_used desc;
```

79

Istruzioni SQL più lunghe

- individuare le istruzioni SQL più lunghe attualmente in esecuzione può rivelarsi un'indagine utile per scovare i lavori che maggiormente impegnano il DB2
- vengono reperite informazioni sia sul job sia sullo specifico statement SQL in esecuzione

Q_JOB_NAME	AUTHORIZATION_NAME	CPU_TIME	RUN_PRIORITY	V_SQL_STATEMENT_TEXT	SQL_STMT_DURATION
468117/QUSER/QZDASOINIT	QZDASOINIT	2066620	20	SELECT * FROM (SELECT TRIM(A.CODARTICOLO) AS ArticoloId, TRIM(A.DESCRIZIONE) AS Descrizio...	1,420540
468273/QUSER/QZDASOINIT	QZDASOINIT	3167	20	-- description: Find the longest executing SQL statements WITH active_user_jobs(q_job_na...	0,007070

ONL-i Days 2019 - DB2 for i performance

```

with active_user_jobs(q_job_name, authorization_name,
                      cpu_time, run_priority)
as (select job_name, authorization_name,
          cpu_time, run_priority
    from table(qsys2.active_job_info()) x
   where job_type <> 'SYS')
select q_job_name, authorization_name,
       cpu_time, run_priority, v_sql_statement_text,
       abs(CURRENT TIMESTAMP -
v_sql_stmt_start_timestamp)
       AS sql_stmt_duration, b.* FROM active_user_jobs,
       table(qsys2.get_job_info(q_job_name)) b
  where v_sql_stmt_status = 'ACTIVE'
 order by sql_stmt_duration desc;

```

80

Plan cache services

► DUMP_PLAN_CACHE_TOPN

estrae dal plan cache le prime n query più lunghe

```
call DUMP_PLAN_CACHE_TOPN('MYLIB', 'TOP20',  
20);
```

► DUMP_PLAN_CACHE_PROPERTIES

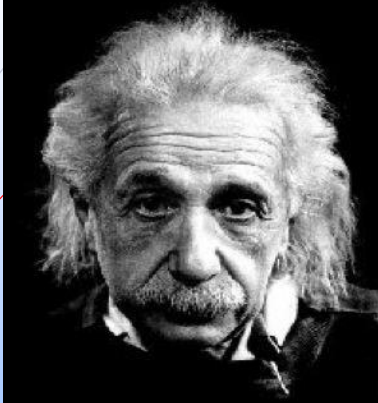
estrae le proprietà del plan cache

```
call  
QSYS2/DUMP_PLAN_CACHE_PROPERTIES('MYLIB','PCP  
ROP');
```

ONL-i Days 2019 - DB2 for i performance

81

... tutti rischiamo momenti di follia...



La definizione di
follia è insistere nel
fare la stessa cosa...

**aspettandosi risultati
differenti!**

Albert Einstein

82

Bibliografia/1

- *Accessing Data Using SQL Views*, di Dan Cruikshank, 1-nov-2007
- *Achieve Memory-Making SQL Performance with DB2 In-Memory Support*, di Kent Milligan, 28-ott-2011, <https://www.mcpressonline.com/analytics-cognitive/db2/techtip-achieve-memory-making-sql-performance-with-db2-in-memory-support>
- *API to convert files to SQL*, di Simon Hutchinson, 6-mag-2015, <https://www.rpgpgm.com/2015/05/api-to-convert-files-to-sql.html>
- *Application Modernization – Db2 for i Style*, di Kent Milligan, Mike Cain, 2011, <https://www.ibm.com/developerworks/community/wikis/form/anonymous/api/wiki/bb5007a3-d52b-44e4-a2bc-4e56b454ffof/page/831ee31b-e632-495b-a18a-37028a5cfb66/attachment/9of45224-31fa-41ad-ba97-4243647a511b/media/app%20modernization%20Db2%20for%20i%20style.pdf>
- *Converting DDS files to SQL DDL using SQL*, di Simon Hutchinson, 13-mag-2015, <https://www.rpgpgm.com/2015/05/converting-dds-files-to-sql-ddl-using.html>
- *Database Modernization: Architectural Design*, di Jim Ritchhart, mar-2013, http://ibmsystemsmag.com/ibmi/developer/modernization/dbmodernization_part4/
- *DB2 for i Adaptive Query Processing (AQP)*, di Mike Cain, 2010
- *DB2 for i Autonomic Index Advice*, di Mike Cain, 2010
- *DB2 for i 7.2 overview*, di Kent Milligan, 20-apr-2015, https://www-356.ibm.com/partnerworld/wps/servlet/ContentHandler/stg_ast_sys-ibm-db2-for-i-7-2-overview

ONL-i Days 2019 - DB2 for i performance

83

Bibliografia/2

- *DB2 for i Indexing - Keys to Success*, di Mike Cain, 2010
- *DB2 for I Technical forum*, di Mike Cain, 2010
- *DB2 UDB for AS/400 Visual Explain*, di Hernando Bedoya, Mark Walas, 10-gen-2001, redbook REDP0505
- *IBM DB2 for i Advanced SQL Concepts*, di Mike Cain, 18-ott-2012, corso OLB77/IT
- *IBM DB2 for i indexing methods and strategies*, di Mike Cain, Kent Milligan, 2011, IBM white paper, https://www-356.ibm.com/partnerworld/wps/servlet/ContentHandler/stg_ast_sys_wp_db2_i_indexing_methods_strategies
- *Index Advisor Surfaces Maintained Temporary Indexes Activity on DB2 for i5/OS*, di Scott Forstie, 12-ago-2008, <http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/tips0624.html?Open#contents>
- *The Power and magic of LPG*, di Mike Cain, 13-feb-2005
- *Using the QAQQINI File*,
- *Understanding and Using the QAQQINI Query Options File*,

ONL-i Days 2019 - DB2 for i performance

84

Riferimenti



- E-mail aziendale: mriva@sirio-is.it



- Blog: www.markonetools.it
- Facebook: <https://www.facebook.com/markonetools/>
- E-mail: info@markonetools.it



- LinkedIn: www.linkedin.com/in/marco-riva-5975785a

ONL-i Days 2019 - DB2 for i performance

87

Marco Riva

1973

Lentate sul Seveso (MB)



ONL-i Days 2019 - DB2 for i performance

88

► **1998-oggi:** Sirio Informatica e Sistemi SpA

- programmatore
- analista
- capo progetto
- istruttore
- consulente tecnico per il laboratorio di sviluppo
- analista performance



ONL-i Days 2019 - DB2 for i performance

89

- 1998: corso introduttivo per AS/400 programmazione RPG/400 (C.I.A.M.)
- 2002: Corso DB2 UDB SQL avanzato (Enza Marvulli)
- 2010: Technical forum DB2 for i (Mike Cain)
- 2011: corso RPG IV avanzato (Luciano Garibaldi)
- 2011: database modernization workshop (Daniel Cruikshank)
- 2012: DB2 advanced SQL next level (Mike Cain)
- 2017: workshop Web Services con RPG e SQL (FAQ400)
- 2017: workshop Node.js e IBM i (FAQ400)

ONL-i Days 2019 - DB2 for i performance



90

- metodi scambio dati tra IBM i e PC
- RPG IV e ILE
- SQL
- embedded SQL in RPG
- DB2 performance
- Rational Developer for i
- Power Systems (PTF, gestione sicurezza, exit point)
- XML

ONL-i Days 2019 - DB2 for i performance

