💰 Cost-Efficient DQL Queries

> **Series:** SPANS | **Notebook:** 8 of 8 | **Created:** December 2025

Optimizing Span Queries for Performance and DDU Efficiency

This notebook covers best practices for writing cost-efficient DQL queries, minimizing DDU consumption while maintaining effective observability.

Table of Contents

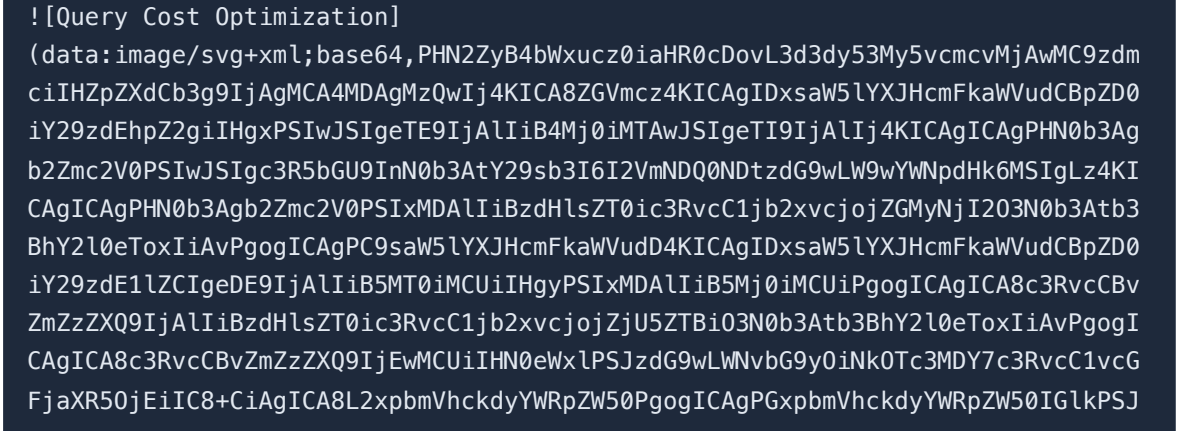
Prerequisites

Before starting this notebook, ensure you have:

- ✅ Completed previous SPANS notebooks (01-07)
- ✅ Understanding of DQL fundamentals
- ✅ Access to span data in your tenant

1. Understanding Query Costs

DQL queries consume DDUs (Davis Data Units) based on several factors:



jb3N0TG93IiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIwJSI+CiAgICAgIDxzdG9wIG9m
ZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMyMmM1NWU7c3RvcC1vcGFjaXR5OjEiIC8+CiAgI
CAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzE2YTM0YTtzdG9wLW9wYW
NpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGllbnQgaWQ9Iml
uZGV4R3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMCUiPgogICAgICA8c3RvcCBv
ZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojNjM2NmYxO3N0b3Atb3BhY2l0eToxIiAvPgogI
CAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiM0ZjQ2ZTU7c3RvcC1vcG
FjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGZpbHRlciBpZD0iY29zdFNoYWR
vdyI+CiAgICAgIDxmZURyb3BTaGFkb3cgZHg9IjIiIGR5PSIyIiBzdGREZXZpYXRpb249IjMiIGZs
b29kLW9wYWNpdHk9IjAuMTUiLz4KICAgIDwvZmlsdGVyPgogICAgPG1hcmtlciBpZD0iY29zdEFyc
m93IiBtYXJrZXJXaWR0aD0iOCIgbWFya2VySGVpZ2h0PSI2IiByZWZYPSI3IiByZWZZPSIzIiBvcm
llbnQ9ImF1dG8iPgogICAgICA8cG9seWdvbiBwb2ludHM9IjAgMCwgOCAzLCAwIDYiIGZpbGw9IiM
2NDc0OGIiLz4KICAgIDwvbWFya2VyPgogIDwvZGVmcz4KCiAgPCEtLSBCYWNrZ3JvdW5kIC0tPgog
IDxyZWN0IHdpZHRoPSI4MDAiIGhlaWdodD0iMzQwIiBmaWxsPSIjZjhmOWZhIiByeD0iMTAiLz4KC
iAgPCEtLSBUaXRsZSAtLT4KICA8dGV4dCB4PSI0MDAiIHk9IjI4IiBmb250LWZhbWlseT0iQXJpYW
wsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTgiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjMzM
zIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5EUUwgUXVlcnkgQ29zdCBPcHRpbWl6YXRpb248L3RleHQ+
CgogIDwhLS0gRERVIENvc3QgRmFjdG9ycyBDb2x1bW4gLS0+CiAgPHRleHQgeD0iMTMwIiB5PSI1O
CIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEzIiBmb250LXdlaW
dodD0iYm9sZCIgZmlsbD0iIzY0NzQ4YiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RERVIENvc3QgRmF
jdG9yczwvdGV4dD4KCiAgPHJlY3QgeD0iMzAiIHk9IjcwIiB3aWR0aD0iMjAwIiBoZWlnaHQ9IjQw
IiByeD0iNiIgZmlsbD0idXJsKCNjb3N0SGlnaCkiIGZpbHRlcj0idXJsKCNjb3N0U2hhZG93KSIvP
gogIDx0ZXh0IHg9IjEzMCIgeT0iOTUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm
9udC1zaXplPSIxMiIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0
ibWlkZGxlIj7wn5OKIERhdGEgU2Nhbm5lZDwvdGV4dD4KCiAgPHJlY3QgeD0iMzAiIHk9IjExOCIg
d2lkdGg9IjIwMCIgaGVpZ2h0PSI0MCIgcng9IjYiIGZpbGw9InVybCgjY29zdEhpZ2gpIiBmaWx0Z
XI9InVybCgjY29zdFNoYWRvdykiLz4KICA8dGV4dCB4PSIxMzAiIHk9IjE0MyIgZm9udC1mYW1pbH
k9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmb250LXdlaWdodD0iYm9sZCIgZml
sbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPuKPse+4jyBUaW1lIFJhbmdlPC90ZXh0PgoK
ICA8cmVjdCB4PSIzMCIgeT0iMTY2IiB3aWR0aD0iMjAwIiBoZWlnaHQ9IjQwIiByeD0iNiIgZmlsb
D0idXJsKCNjb3N0TWVkKSIgZmlsdGVyPSJ1cmwoI2Nvc3RTaGFkb3cpIi8+CiAgPHRleHQgeD0iMT
MwIiB5PSIxOTEiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMiI
gZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj7wn5OL
IEZpZWxkcyBSZXRyaWV2ZWQ8L3RleHQ+CgogIDxyZWN0IHg9IjMwIiB5PSIyMTQiIHdpZHRoPSIyM
DAiIGhlaWdodD0iNDAiIHJ4PSI2IiBmaWxsPSJ1cmwoI2Nvc3RNZWQpIiBmaWx0ZXI9InVybCgjY2
9zdFNoYWRvdykiLz4KICA8dGV4dCB4PSIxMzAiIHk9IjIzOSIgZm9udC1mYW1pbHk9IkFyaWFsLCB
zYW5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUi
IHRleHQtYW5jaG9yPSJtaWRkbGUiPvCflIQgUXVlcnkgQ29tcGxleGl0eTwvdGV4dD4KCiAgPCEtL
SBBcnJvd3MgLS0+CiAgPGxpbmUgeDE9IjIzNSIgeTE9IjE0MCIgeDI9IjI3MCIgeTI9IjE0MCIgc3
Ryb2tlPSIjNjQ3NDhiIiBzdHJva2Utd2lkdGg9IjIiIG1hcmtlci1lbmQ9InVybCgjY29zdEFycm9
3KSIvPgoKICA8IS0tIE9wdGltaXphdGlvbiBTdHJhdGVnaWVzIENvbHVtbiAtLT4KICA8dGV4dCB4
PSI0MDAiIHk9IjU4IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iM
TMiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjNjQ3NDhiIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj
5PcHRpbWl6YXRpb24gU3RyYXRlZ2llczwvdGV4dD4KCiAgPHJlY3QgeD0iMjc1IiB5PSI3MCIgd2l
kdGg9IjI1MCIgaGVpZ2h0PSIzOCIgcng9IjYiIGZpbGw9InVybCgjY29zdExvdykiIGZpbHRlcj0i
dXJsKCNjb3N0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQwMCIgeT0iODUiIGZvbnQtZmFtaWx5PSJBc
mlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9In
doaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4xLiBGaWx0ZXIgRWFybHk8L3RleHQ+CiAgPHRleHQ

geD0iNDAwIiB5PSIxMDAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXpl
PSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5SZ
WR1Y2UgZGF0YSBiZWZvcmUgcHJvY2Vzc2luZzwvdGV4dD4KCiAgPHJlY3QgeD0iMjc1IiB5PSIxMT
QiIHdpZHRoPSIyNTAiIGhlaWdodD0iMzgiIHJ4PSI2IiBmaWxsPSJ1cmwoI2Nvc3RMb3cpIiBmaWx
0ZXI9InVybCgjY29zdFNoYWRvdykiLz4KICA8dGV4dCB4PSI0MDAiIHk9IjEyOSIgZm9udC1mYW1p
bHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZ
mlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPjIuIFVzZSBJbmRleGVkIEZpZWxkczwvdG
V4dD4KICA8dGV4dCB4PSI0MDAiIHk9IjE0NCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcml
mIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9y
PSJtaWRkbGUiPmR0LmVudGl0eS5zZXJ2aWNlLCBzcGFuLmtpbmQsIHRyYWNlLmlkPC90ZXh0PgoKI
CA8cmVjdCB4PSIyNzUiIHk9IjE1OCIgd2lkdGg9IjI1MCIgaGVpZ2h0PSIzOCIgcng9IjYiIGZpbG
w9InVybCgjY29zdExvdykiIGZpbHRlcj0idXJsKCNjb3N0U2hhZG93KSIvPgogIDx0ZXh0IHg9IjQ
wMCIgeT0iMTczIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEi
IGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+My4gT
mFycm93IFRpbWUgUmFuZ2U8L3RleHQ+CiAgPHRleHQgeD0iNDAwIiB5PSIxODgiIGZvbnQtZmFtaW
x5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI
1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Vc2Ugc21hbGxlc3QgcmFuZ2UgbmVlZGVkPC90
ZXh0PgoKICA8cmVjdCB4PSIyNzUiIHk9IjIwMiIgd2lkdGg9IjI1MCIgaGVpZ2h0PSIzOCIgcng9I
jYiIGZpbGw9InVybCgjY29zdExvdykiIGZpbHRlcj0idXJsKCNjb3N0U2hhZG93KSIvPgogIDx0ZX
h0IHg9IjQwMCIgeT0iMjE3IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l
6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRs
ZSI+NC4gU2VsZWN0IE9ubHkgTmVlZGVkIEZpZWxkczwvdGV4dD4KICA8dGV4dCB4PSI0MDAiIHk9I
jIzMiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPS
JyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlVzZSBmaWVsZHMgY29
tbWFuZDwvdGV4dD4KCiAgPHJlY3QgeD0iMjc1IiB5PSIyNDYiIHdpZHRoPSIyNTAiIGhlaWdodD0i
MzgiIHJ4PSI2IiBmaWxsPSJ1cmwoI2Nvc3RMb3cpIiBmaWx0ZXI9InVybCgjY29zdFNoYWRvdykiL
z4KICA8dGV4dCB4PSI0MDAiIHk9IjI2MSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIi
Bmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9
yPSJtaWRkbGUiPjUuIFRhcmdldCBTcGVjaWZpYyBCdWNrZXRzPC90ZXh0PgogIDx0ZXh0IHg9IjQw
MCIgeT0iMjc2IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiI
GZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+ZmV0Y2ggc3
BhbnMsIGJ1Y2tldDp7InByb2QifTwvdGV4dD4KCiAgPCEtLSBBcnJvdyB0byBSZXN1bHQgLS0+CiA
gPGxpbmUgeDE9IjUzMCIgeTE9IjE3NSIgeDI9IjU2NSIgeTI9IjE3NSIgc3Ryb2tlPSIjNjQ3NDhi
IiBzdHJva2Utd2lkdGg9IjIiIG1hcmtlci1lbmQ9InVybCgjY29zdEFycm93KSIvPgoKICA8IS0tI
FJlc3VsdCBDb2x1bW4gLS0+CiAgPHRleHQgeD0iNjgwIiB5PSI1OCIgZm9udC1mYW1pbHk9IkFyaW
FsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEzIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzY
0NzQ4YiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+SW5kZXhlZCBGaWVsZHM8L3RleHQ+CgogIDxyZWN0
IHg9IjU3MCIgeT0iNzAiIHdpZHRoPSIyMjAiIGhlaWdodD0iMTgwIiByeD0iOCIgZmlsbD0idXJsK
CNpbmRleEdyYWQpIiBmaWx0ZXI9InVybCgjY29zdFNoYWRvdykiLz4KCiAgPHRleHQgeD0iNjgwIi
B5PSI5NSIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSg
yNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj7igKIgdHJhY2UuaWQ8L3RleHQ+
CiAgPHRleHQgeD0iNjgwIiB5PSIxMTUiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZvbnQtc2l6Z
T0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+4o
CiIHNwYW4uaWQ8L3RleHQ+CiAgPHRleHQgeD0iNjgwIiB5PSIxMzUiIGZvbnQtZmFtaWx5PSJtb25
vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1h
bmNob3I9Im1pZGRsZSI+4oCiIGR0LmVudGl0eS5zZXJ2aWNlPC90ZXh0PgogIDx0ZXh0IHg9IjY4M
CIgeT0iMTU1IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2
JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPuKAoiBzcGFuLm5hbWU8L3R

leHQ+CiAgPHRleHQgeD0iNjgwIiB5PSIxNzUiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZvbnQt
c2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZ
SI+4oCiIHNwYW4ua2luZDwvdGV4dD4KICA8dGV4dCB4PSI2ODAiIHk9IjE5NSIgZm9udC1mYW1pbH
k9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB
0ZXh0LWFuY2hvcj0ibWlkZGxlIj7igKIgc3Bhbi5zdGF0dXNfY29kZTwvdGV4dD4KICA8dGV4dCB4
PSI2ODAiIHk9IjIxNSIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsb
D0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj7igKIgc3RhcnRfdG
ltZTwvdGV4dD4KCiAgPHRleHQgeD0iNjgwIiB5PSIyNDAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2F
ucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0
ZXh0LWFuY2hvcj0ibWlkZGxlIj5GaWx0ZXIgb24gdGhlc2UgZmlyc3QhPC90ZXh0PgoKICA8IS0tI
ExlZ2VuZC9JbXBhY3QgLS0+CiAgPHJlY3QgeD0iMzAiIHk9IjI5MCIgd2lkdGg9Ijc2MCIgaGVpZ2
h0PSI0MCIgcng9IjYiIGZpbGw9IiNmMGY5ZmYiIHN0cm9rZT0iI2JhZTZmZCIgc3Ryb2tlLXdpZHR
oPSIxIi8+CiAgPHRleHQgeD0iNTAiIHk9IjMxNSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNl
cmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzAzNjlhMSI+Q29zd
CBJbXBhY3Q6PC90ZXh0PgogIDxyZWN0IHg9IjEzNSIgeT0iMzAzIiB3aWR0aD0iMTQiIGhlaWdodD
0iMTQiIHJ4PSIzIiBmaWxsPSJ1cmwoI2Nvc3RIaWdoKSIvPgogIDx0ZXh0IHg9IjE1NSIgeT0iMzE
1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiMw
MzY5YTEiPkhpZ2ggQ29zdDwvdGV4dD4KICA8cmVjdCB4PSIyMzUiIHk9IjMwMyIgd2lkdGg9IjE0I
iBoZWlnaHQ9IjE0IiByeD0iMyIgZmlsbD0idXJsKCNjb3N0TWVkKSIvPgogIDx0ZXh0IHg9IjI1NS
IgeT0iMzE1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZ
pbGw9IiMwMzY5YTEiPk1lZGl1bTwvdGV4dD4KICA8cmVjdCB4PSIzMjAiIHk9IjMwMyIgd2lkdGg9
IjE0IiBoZWlnaHQ9IjE0IiByeD0iMyIgZmlsbD0idXJsKCNjb3N0TG93KSIvPgogIDx0ZXh0IHg9I
jM0MCIgeT0iMzE1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMT
EiIGZpbGw9IiMwMzY5YTEiPk9wdGltaXplZDwvdGV4dD4KICA8dGV4dCB4PSI0ODAiIHk9IjMxNSI
gZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmaWxsPSIjMDM2
OWExIj7wn5KhIEZpbHRlciBvbiBpbmRleGVkIGZpZWxkcyBmaXJzdCDihpIgdGhlbiBub24taW5kZ
XhlZCDihpIgdGhlbiBwcm9jZXNzPC90ZXh0Pgo8L3N2Zz4K)

Cost Optimization Priorities

Priority	Strategy	Impact
1	Filter early	High — reduces data scanned
2	Use indexed fields	High — leverages optimized indexes
3	Limit time range	High — reduces data volume
4	Select only needed fields	Medium — reduces transfer
5	Query specific buckets	Medium — targets data
6	Use aggregations	Low — summarizes results

---

2. Filter Early Pattern

The **most important** optimization: Apply filters as early as possible in the query pipeline.

```dql
// ✅ EFFICIENT: Filter BEFORE any processing
fetch spans
| filter span.kind == "server"           // Filter first!
| filter dt.entity.service == "SERVICE-6C36694E683AD694"  // Further narrow
| filter span.status_code == "error"     // Even more specific
| fields start_time, span.name, duration
| limit 100
```

```dql
// ❌ INEFFICIENT: Processing before filtering scans more data
// This pattern wastes resources - shown for comparison only
fetch spans
| fieldsAdd duration_ms = duration / 1000000  // Processing first (bad)
| summarize {avg_duration = avg(duration_ms)}, by:{dt.entity.service,
span.name}
| filter dt.entity.service == "SERVICE-6C36694E683AD694"  // Too late!
| limit 100
```

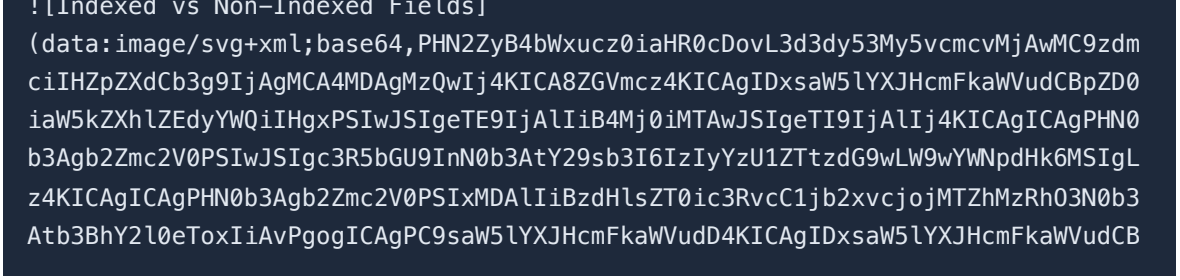
```dql
// ✅ EFFICIENT: Combine multiple filter conditions
fetch spans
| filter span.kind == "server"
        and isNotNull(dt.entity.service)
        and duration > 100ms
| fields start_time, span.name, duration
| sort duration desc
| limit 50
```

---

3. Indexed Fields for Performance

Filter on **indexed fields** for best query performance. These fields have optimized storage that allows fast filtering.

Commonly Indexed Span Fields



pZD0ibm9uSW5kZXhlZEdyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjAlIj4KICAg
ICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6I2Y1OWUwYjtzdG9wLW9wYWNpd
Hk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZDk3Nz
A2O3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmF
kaWVudCBpZD0iaGVhZGVyR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMCUiPgog
ICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMTQ5NmZmO3N0b3Atb3BhY
2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMwYT
Y0YmM7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGZpbHRlciB
pZD0iaWR4U2hhZG93Ij4KICAgICAgPGZlRHJvcFNoYWRvdyBkeD0iMSIgZHk9IjEiIHN0ZERldmlh
dGlvbj0iMiIgZmxvb2Qtb3BhY2l0eT0iMC4xNSIvPgogICAgPC9maWx0ZXI+CiAgPC9kZWZzPgoKI
CA8IS0tIEJhY2tncm91bmQgLS0+CiAgPHJlY3Qgd2lkdGg9IjgwMCIgaGVpZ2h0PSIzNDAiIGZpbG
w9IiNmOGY5ZmEiIHJ4PSIxMCIvPgoKICA8IS0tIFRpdGxlIC0tPgogIDx0ZXh0IHg9IjQwMCIgeT0
iMjgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxOCIgZm9udC13
ZWlnaHQ9ImJvbGQiIGZpbGw9IiMzMzMiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkluZGV4ZWQgdnMgT
m9uLUluZGV4ZWQgU3BhbiBGaWVsZHM8L3RleHQ+CiAgPHRleHQgeD0iNDAwIiB5PSI0OCIgZm9udC
1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmaWxsPSIjNjY2IiB0ZXh
0LWFuY2hvcj0ibWlkZGxlIj5GaWx0ZXIgb24gaW5kZXhlZCBmaWVsZHMgZmlyc3QgZm9yIG9wdGlt
YWwgcXVlcnkgcGVyZm9ybWFuY2U8L3RleHQ+CgogIDwhLS0gSW5kZXhlZCBGaWVsZHMgU2VjdGlvb
iAtLT4KICA8cmVjdCB4PSIzMCIgeT0iNzAiIHdpZHRoPSIzNjAiIGhlaWdodD0iMjUwIiByeD0iOC
IgZmlsbD0iI2ZmZiIgc3Ryb2tlPSIjMjJjNTVlIiBzdHJva2Utd2lkdGg9IjIiIGZpbHRlcj0idXJ
sKCNpZHhTaGFkb3cpIi8+CiAgPHJlY3QgeD0iMzAiIHk9IjcwIiB3aWR0aD0iMzYwIiBoZWlnaHQ9
IjM1IiByeD0iOCIgZmlsbD0idXJsKCNpbmRleGVkR3JhZCkiLz4KICA8dGV4dCB4PSIyMTAiIHk9I
jkzIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIGZvbnQtd2
VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+4pyTIElORVVYRUQ
gKEZhc3QgRmlsdGVyaW5nKTwvdGV4dD4KCiAgPCEtLSBJbmRleGVkIGZpZWxkIHJvd3MgLS0+CiAg
PHRleHQgeD0iNTAiIHk9IjEzMCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxM
iIgZmlsbD0iIzMzMyI+dHJhY2UuaWQ8L3RleHQ+CiAgPHRleHQgeD0iMjAwIiB5PSIxMzAiIGZvbn
QtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZmlsbD0iIzY0NzQ4YiI
+UHJpbWFyeSB0cmFjZSBpZGVudGlmaWVyPC90ZXh0PgoKICA8bGluZSB4MT0iNDUiIHkxPSIxNDUi
IHgyPSIzNzUiIHkyPSIxNDUiIHN0cm9rZT0iI2UyZThmMCIgc3Ryb2tlLXdpZHRoPSIxIi8+CgogI
Dx0ZXh0IHg9IjUwIiB5PSIxNjUiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMT
IiIGZpbGw9IiMzMzMiPnNwYW4uaWQ8L3RleHQ+CiAgPHRleHQgeD0iMjAwIiB5PSIxNjUiIGZvbnQ
tZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZmlsbD0iIzY0NzQ4YiI+
SW5kaXZpZHVhbCBzcGFuIGlkZW50aWZpZXI8L3RleHQ+CgogIDxsaW5lIHgxPSI0NSIgeTE9IjE4M
CIgeDI9IjM3NSIgeTI9IjE4MCIgc3Ryb2tlPSIjZTJlOGYwIiBzdHJva2Utd2lkdGg9IjEiLz4KCi
AgPHRleHQgeD0iNTAiIHk9IjIwMCIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSI
xMiIgZmlsbD0iIzMzMyI+ZHQuZW50aXR5LnNlcnZpY2U8L3RleHQ+CiAgPHRleHQgeD0iMjAwIiB5
PSIyMDAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZmlsb
D0iIzY0NzQ4YiI+U2VydmljZSBlbnRpdHkgSUQ8L3RleHQ+CgogIDxsaW5lIHgxPSI0NSIgeTE9Ij
IxNSIgeDI9IjM3NSIgeTI9IjIxNSIgc3Ryb2tlPSIjZTJlOGYwIiBzdHJva2Utd2lkdGg9IjEiLz4
KCiAgPHRleHQgeD0iNTAiIHk9IjIzNSIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXpl
PSIxMiIgZmlsbD0iIzMzMyI+c3Bhbi5uYW1lPC90ZXh0PgogIDx0ZXh0IHg9IjIwMCIgeT0iMjM1I
iBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiM2ND
c0OGIiPk9wZXJhdGlvbiBuYW1lPC90ZXh0PgoKICA8bGluZSB4MT0iNDUiIHkxPSIyNTAiIHgyPSI
zNzUiIHkyPSIyNTAiIHN0cm9rZT0iI2UyZThmMCIgc3Ryb2tlLXdpZHRoPSIxIi8+CgogIDx0ZXh0
IHg9IjUwIiB5PSIyNzAiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMTIiIGZpb
Gw9IiMzMzMiPnNwYW4ua2luZDwvdGV4dD4KICA8dGV4dCB4PSIyMDAiIHk9IjI3MCIgZm9udC1mYW
1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmaWxsPSIjNjQ3NDhiIj5zZXJ

2ZXIsIGNsaWVudCwgaW50ZXJuYWwuLi48L3RleHQ+CgogIDxsaW5lIHgxPSI0NSIgeTE9IjI4NSIg
eDI9IjM3NSIgeTI9IjI4NSIgc3Ryb2tlPSIjZTJlOGYwIiBzdHJva2Utd2lkdGg9IjEiLz4KCiAgP
HRleHQgeD0iNTAiIHk9IjMwNSIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMi
IgZmlsbD0iIzMzMyI+c3Bhbi5zdGF0dXNfY29kZTwvdGV4dD4KICA8dGV4dCB4PSIyMDAiIHk9IjM
wNSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmaWxsPSIj
NjQ3NDhiIj5vaywgZXJyb3I8L3RleHQ+CgogIDwhLS0gTm9uLUluZGV4ZWQgRmllbGRzIFNlY3Rpb
24gLS0+CiAgPHJlY3QgeD0iNDEwIiB5PSI3MCIgd2lkdGg9IjM2MCIgaGVpZ2h0PSIxNDUiIHJ4PS
I4IiBmaWxsPSIjZmZmIiBzdHJva2U9IiNmNTllMGIiIHN0cm9rZS13aWR0aD0iMiIgZmlsdGVyPSJ
1cmwoI2lkeFNoYWRvdykiLz4KICA8cmVjdCB4PSI0MTAiIHk9IjcwIiB3aWR0aD0iMzYwIiBoZWln
aHQ9IjM1IiByeD0iOCIgZmlsbD0idXJsKCNub25JbmRleGVkR3JhZCkiLz4KICA8dGV4dCB4PSI1O
TAiIHk9IjkzIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTQiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+4pqgIE5
PTi1JTkRFWEVEIChTbG93ZXIpPC90ZXh0PgoKICA8IS0tIE5vbi1pbmRleGVkIGZpZWxkIHJvd3Mg
LS0+CiAgPHRleHQgeD0iNDMwIiB5PSIxMzAiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZvbnQtc
2l6ZT0iMTIiIGZpbGw9IiMzMzMiPnVybC5wYXRoPC90ZXh0PgogIDx0ZXh0IHg9IjU2MCIgeT0iMT
MwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiM
2NDc0OGIiPlJlcXVpcmVzIGZ1bGwgc2NhbjwvdGV4dD4KCiAgPGxpbmUgeDE9IjQyNSIgeTE9IjE0
NSIgeDI9Ijc1NSIgeTI9IjE0NSIgc3Ryb2tlPSIjZTJlOGYwIiBzdHJva2Utd2lkdGg9IjEiLz4KC
iAgPHRleHQgeD0iNDMwIiB5PSIxNjUiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZvbnQtc2l6ZT
0iMTIiIGZpbGw9IiMzMzMiPmh0dHAudXJsPC90ZXh0PgogIDx0ZXh0IHg9IjU2MCIgeT0iMTY1IiB
mb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZpbGw9IiM2NDc0
OGIiPlVzZSBhZnRlciBpbmRleGVkIGZpbHRlcnM8L3RleHQ+CgogIDxsaW5lIHgxPSI0MjUiIHkxP
SIxODAiIHgyPSI3NTUiIHkyPSIxODAiIHN0cm9rZT0iI2UyZThmMCIgc3Ryb2tlLXdpZHRoPSIxIi
8+CgogIDx0ZXh0IHg9IjQzMCIgeT0iMjAwIiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXN
pemU9IjEyIiBmaWxsPSIjMzMzIj5kYi5zdGF0ZW1lbnQ8L3RleHQ+CiAgPHRleHQgeD0iNTYwIiB5
PSIyMDAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZmlsb
D0iIzY0NzQ4YiI+Q3VzdG9tIGF0dHJpYnV0ZXM8L3RleHQ+CgogIDwhLS0gQmVzdCBQcmFjdGljZS
BCb3ggLS0+CiAgPHJlY3QgeD0iNDEwIiB5PSIyMzAiIHdpZHRoPSIzNjAiIGhlaWdodD0iOTAiIHJ
4PSI4IiBmaWxsPSIjMWUyOTNiIiBmaWx0ZXI9InVybCgjaWR4U2hhZG93KSIvPgogIDx0ZXh0IHg9
IjU5MCIgeT0iMjU1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iM
TIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjOTRhM2I4IiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj
5CZXN0IFByYWN0aWNlIFBhdHRlcm48L3RleHQ+CiAgPHRleHQgeD0iNDI1IiB5PSIyODAiIGZvbnQ
tZmFtaWx5PSJtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiMyMmM1NWUiPmZldGNoPC90
ZXh0PgogIDx0ZXh0IHg9IjQ2MCIgeT0iMjgwIiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250L
XNpemU9IjEwIiBmaWxsPSIjZjhmYWZjIj5zcGFuczwvdGV4dD4KICA8dGV4dCB4PSI0MjUiIHk9Ij
I5NSIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzE0OTZmZiI
+fCBmaWx0ZXI8L3RleHQ+CiAgPHRleHQgeD0iNDg1IiB5PSIyOTUiIGZvbnQtZmFtaWx5PSJtb25v
c3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiMyMmM1NWUiPnNwYW4ua2luZCA9PSAic2VydmVyI
jwvdGV4dD4KICA8dGV4dCB4PSI2NzAiIHk9IjI5NSIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm
9udC1zaXplPSIxMCIgZmlsbD0iIzY0NzQ4YiI+4oaQIGluZGV4ZWQ8L3RleHQ+CiAgPHRleHQgeD0
iNDI1IiB5PSIzMTAiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9
IiMxNDk2ZmYiPnwgZmlsdGVyPC90ZXh0PgogIDx0ZXh0IHg9IjQ4NSIgeT0iMzEwIiBmb250LWZhb
WlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIjZmJiZjI0Ij5jb250YWlucyh1cm
wucGF0aCwgImFwaSIpPC90ZXh0PgogIDx0ZXh0IHg9IjY3MCIgeT0iMzEwIiBmb250LWZhbWlseT0
ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIj7ihpAgdGhlbiBub24taW5k
ZXhlZDwvdGV4dD4KPC9zdmc+Cg==)

````
> 💡 **Tip:** When possible, filter on `dt.entity.service` rather than
`service.name` - the entity ID is more reliably indexed.

```dql
// ✅ FAST: Filter on indexed fields
fetch spans
| filter isNotNull(dt.entity.service)    // Indexed
| filter span.kind == "server"           // Indexed
| filter span.status_code == "error"     // Indexed
| fields start_time, span.name, duration
| limit 100
```

```dql
// ⚠️ SLOWER: Filter on non-indexed field requires full scan
fetch spans
| filter contains(url.path, "checkout")  // Not indexed - slower
| limit 100
```

```dql
// ✅ BETTER: Combine indexed filter first, then non-indexed
fetch spans
| filter span.kind == "server"           // Indexed - reduces data first
| filter isNotNull(dt.entity.service)    // Indexed - further reduces
| filter contains(url.path, "checkout")  // Non-indexed on smaller dataset
| limit 100
```

---

## 4. Field Selection

Select only the fields you need - avoid fetching all attributes.

```dql
// ✅ EFFICIENT: Select only required fields
fetch spans
| filter span.kind == "server"
| fields start_time,
         dt.entity.service,
         span.name,
         duration,
         span.status_code
| limit 100
```
````

```dql
// ❌ INEFFICIENT: Fetching all fields (default behavior)
// Returns ALL fields for each span - more data transfer
fetch spans
| filter isNotNull(dt.entity.service)
| limit 5
```

```dql
// ✅ EFFICIENT: Compute fields only when needed
fetch spans
| filter span.kind == "server"
| filter duration > 500ms
| fieldsAdd duration_ms = duration / 1000000
| fields start_time, dt.entity.service, span.name, duration_ms
| sort duration_ms desc
| limit 50
```

---

5. Time Range Optimization

Always use the smallest time range that meets your needs.

kbGUiPlNob3J0ZXIgdGltZSByYW5nZXMgPSBsb3dlciBERFUgY29zdCB8IFVzZSB0aGUgbWluaW11
bSByYW5nZSBuZWVkZWQ8L3RleHQ+CgogIDwhLS0gQ29zdCBTY2FsZSBWaXN1YWxpemF0aW9uIC0tP
gogIDxyZWN0IHg9IjUwIiB5PSI3MCIgd2lkdGg9IjcwMCIgaGVpZ2h0PSIxMDAiIHJ4PSI4IiBmaW
xsPSIjZmZmIiBzdHJva2U9IiNlMmU4ZjAiIHN0cm9rZS13aWR0aD0iMiIgZmlsdGVyPSJ1cmwoI3R
yU2hhZG93KSIvPgoKICA8IS0tIFRpbWUgcmFuZ2UgYmFycyAtIHByb2dyZXNzaXZlIHdpZHRoIHNo
b3dzIHJlbGF0aXZlIGNvc3QgLS0+CiAgPHJlY3QgeD0iNzAiIHk9IjkwIiB3aWR0aD0iNTAiIGhla
WdodD0iMzAiIHJ4PSI0IiBmaWxsPSJ1cmwoI2xvd0Nvc3RHcmFkKSIvPgogIDx0ZXh0IHg9Ijk1Ii
B5PSIxMTAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9
udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj41bTwvdGV4
dD4KICA8dGV4dCB4PSI5NSIgeT0iMTQwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiI
GZvbnQtc2l6ZT0iMTAiIGZpbGw9IiMyMmM1NWUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPiQ8L3RleH
Q+CgogIDxyZWN0IHg9IjE0MCIgeT0iOTAiIHdpZHRoPSI4MCIgaGVpZ2h0PSIzMCIgcng9IjQiIGZ
pbGw9InVybCgjbG93Q29zdEdyYWQpIi8+CiAgPHRleHQgeD0iMTgwIiB5PSIxMTAiIGZvbnQtZmFt
aWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiI
GZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4xaDwvdGV4dD4KICA8dGV4dCB4PSIxOD
AiIHk9IjE0MCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiB
maWxsPSIjMjJjNTVlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4kJDwvdGV4dD4KCiAgPHJlY3QgeD0i
MjQwIiB5PSI5MCIgd2lkdGg9IjEzMCIgaGVpZ2h0PSIzMCIgcng9IjQiIGZpbGw9InVybCgjbWVkQ
29zdEdyYWQpIi8+CiAgPHRleHQgeD0iMzA1IiB5PSIxMTAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2
Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB
0ZXh0LWFuY2hvcj0ibWlkZGxlIj4yNGg8L3RleHQ+CiAgPHRleHQgeD0iMzA1IiB5PSIxNDAiIGZv
bnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2Y1OWUwY
iIgdGV4dC1hbmNob3I9Im1pZGRsZSI+JCQkPC90ZXh0PgoKICA8cmVjdCB4PSIzOTAiIHk9IjkwIi
B3aWR0aD0iMTgwIiBoZWlnaHQ9IjMwIiByeD0iNCIgZmlsbD0idXJsKCNoaWdoQ29zdEdyYWQpIi8
+CiAgPHRleHQgeD0iNDgwIiB5PSIxMTAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIg
Zm9udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvc
j0ibWlkZGxlIj43ZDwvdGV4dD4KICA8dGV4dCB4PSI0ODAiIHk9IjE0MCIgZm9udC1mYW1pbHk9Ik
FyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjZWY0NDQ0IiB0ZXh0LWFuY2h
vcj0ibWlkZGxlIj4kJCQkPC90ZXh0PgoKICA8cmVjdCB4PSI1OTAiIHk9IjkwIiB3aWR0aD0iMTQw
IiBoZWlnaHQ9IjMwIiByeD0iNCIgZmlsbD0idXJsKCNoaWdoQ29zdEdyYWQpIi8+CiAgPHRleHQge
D0iNjYwIiB5PSIxMTAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPS
IxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4
zMGQ8L3RleHQ+CiAgPHRleHQgeD0iNjYwIiB5PSIxNDAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fu
cy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iI2VmNDQ0NCIgdGV4dC1hbmNob3I9Im1pZGRsZ
SI+JCQkJCQ8L3RleHQ+CgogIDwhLS0gQXJyb3cgc2hvd2luZyBwcm9ncmVzc2lvbiAtLT4KICA8cG
F0aCBkPSJNNzAsMTU1IEw3MzAsMTU1IiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiI
gZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXJsKCNhcnJvdykiLz4KICA8dGV4dCB4PSI0MDAiIHk9
IjE2OCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsP
SIjNjQ3NDhiIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5JbmNyZWFzaW5nIEREVSBDb3N0IOKGkjwvdG
V4dD4KCiAgPCEtLSBVc2UgQ2FzZSBHdWlkZSAtLT4KICA8cmVjdCB4PSI1MCIgeT0iMTg1IiB3aWR
0aD0iMzQwIiBoZWlnaHQ9IjEyMCIgcng9IjgiIGZpbGw9IiNmZmYiIHN0cm9rZT0iI2UyZThhMCIg
c3Ryb2tlLXdpZHRoPSIyIiBmaWx0ZXI9InVybCgjdHJTaGFkb3cpIi8+CiAgPHRleHQgeD0iMjIwI
iB5PSIyMDgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMiIgZm
9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IiMzMzMiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlJlY29tbWV
uZGVkIFRpbWUgUmFuZ2VzPC90ZXh0PgoKICA8cmVjdCB4PSI2NSIgeT0iMjIwIiB3aWR0aD0iMTIi
IGhlaWdodD0iMTIiIHJ4PSIyIiBmaWxsPSJ1cmwoI2xvd0Nvc3RHcmFkKSIvPgogIDx0ZXh0IHg9I
jg1IiB5PSIyMzAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMC
IgZmlsbD0iIzMzMyI+UmVhbC10aW1lIGRlYnVnZ2luZzogNS0xNW08L3RleHQ+CgogIDxyZWN0IHg

9IjY1IiB5PSIyNDIiIHdpZHRoPSIxMiIgaGVpZ2h0PSIxMiIgcng9IjIiIGZpbGw9InVybCgjbG93
Q29zdEdyYWQpIi8+CiAgPHRleHQgeD0iODUiIHk9IjI1MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzY
W5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjMzMzIj5SZWNlbnQgaW52ZXN0aWdhdGlvbj
ogMS02aDwvdGV4dD4KCiAgPHJlY3QgeD0iNjUiIHk9IjI2NCIgd2lkdGg9IjEyIiBoZWlnaHQ9IjE
yIiByeD0iMiIgZmlsbD0idXJsKCNtZWRDb3N0R3JhZCkiLz4KICA8dGV4dCB4PSI4NSIgeT0iMjc0
IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiMzM
zMiPkRhaWx5IHBhdHRlcm5zOiAyNGg8L3RleHQ+CgogIDxyZWN0IHg9IjY1IiB5PSIyODYiIHdpZH
RoPSIxMiIgaGVpZ2h0PSIxMiIgcng9IjIiIGZpbGw9InVybCgjaGlnaENvc3RHcmFkKSIvPgogIDx
0ZXh0IHg9Ijg1IiB5PSIyOTYiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1z
aXplPSIxMCIgZmlsbD0iIzMzMyI+V2Vla2x5L2hpc3RvcmljYWw6IDdkKyAodXNlIGFnZ3JlZ2F0a
W9ucyEpPC90ZXh0PgoKICA8IS0tIERRTCBQYXR0ZXJuIEJveCAtLT4KICA8cmVjdCB4PSI0MTAiIH
k9IjE4NSIgd2lkdGg9IjM0MCIgaGVpZ2h0PSIxMjAiIHJ4PSI4IiBmaWxsPSIjMWUyOTNiIiBmaWx
0ZXI9InVybCgjdHJTaGFkb3cpIi8+CiAgPHRleHQgeD0iNTgwIiB5PSIyMDgiIGZvbnQtZmFtaWx5
PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMiIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpb
Gw9IiM5NGEzYjgiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkV4cGxpY2l0IFRpbWUgRmlsdGVyIFBhdH
Rlcm48L3RleHQ+CgogIDx0ZXh0IHg9IjQyNSIgeT0iMjM1IiBmb250LWZhbWlseT0ibW9ub3NwYWN
lIiBmb250LXNpemU9IjEwIiBmaWxsPSIjMjJjNTVlIj5mZXRjaDwvdGV4dD4KICA8dGV4dCB4PSI0
NjAiIHk9IjIzNSIgZm9udC1mYW1pbHk9Im1vbm9zcGFjZSIgZm9udC1zaXplPSIxMCIgZmlsbD0iI
2Y4ZmFmYyI+c3BhbnM8L3RleHQ+CiAgPHRleHQgeD0iNDI1IiB5PSIyNTUiIGZvbnQtZmFtaWx5PS
Jtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiMxNDk2ZmYiPnwgZmlsdGVyPC90ZXh0Pgo
gIDx0ZXh0IHg9IjQ4NSIgeT0iMjU1IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9
IjEwIiBmaWxsPSIjZmJiZjI0Ij5zdGFydF90aW1lID49IG5vdygpIC0gMWg8L3RleHQ+CiAgPHRle
HQgeD0iNDI1IiB5PSIyNzUiIGZvbnQtZmFtaWx5PSJtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIG
ZpbGw9IiMxNDk2ZmYiPnwgZmlsdGVyPC90ZXh0PgogIDx0ZXh0IHg9IjQ4NSIgeT0iMjc1IiBmb25
0LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXNpemU9IjEwIiBmaWxsPSIjZjhmYWZjIj5zcGFuLmtp
bmQgPT0gInNlcnZlciI8L3RleHQ+CiAgPHRleHQgeD0iNDI1IiB5PSIyOTUiIGZvbnQtZmFtaWx5P
SJtb25vc3BhY2UiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiMxNDk2ZmYiPnwgc3VtbWFyaXplPC90ZX
h0PgogIDx0ZXh0IHg9IjUwNSIgeT0iMjk1IiBmb250LWZhbWlseT0ibW9ub3NwYWNlIiBmb250LXN
pemU9IjEwIiBmaWxsPSIjZjhmYWZjIj57Y291bnQoKX0sIGJ5OnsuLi59PC90ZXh0Pgo8L3N2Zz4K
)


````
```dql
// Query with explicit time filter (last 15 minutes)
fetch spans
| filter start_time >= now() - 15m
| filter span.kind == "server"
| summarize {span_count = count()}, by:{dt.entity.service}
| sort span_count desc
| limit 20
```

```dql
// Narrow time range for troubleshooting specific incident
fetch spans
| filter start_time >= now() - 30m
| filter span.kind == "server"
````
````

```
| filter span.status_code == "error"
| fields start_time, dt.entity.service, span.name, span.status_message
| sort start_time desc
| limit 100
```

```dql
// Use aggregations for longer time ranges to reduce output
fetch spans
| filter span.kind == "server"
| summarize {
    span_count = count(),
    error_count = countIf(span.status_code == "error"),
    avg_duration_ms = avg(duration) / 1000000
  }, by:{dt.entity.service}
| sort span_count desc
| limit 25
```

---

6. Aggregation Efficiency

Use aggregations to summarize data instead of retrieving raw records. Simpler aggregations are faster.

```dql
// ✅ EFFICIENT: Basic aggregations
fetch spans
| filter span.kind == "server"
| summarize {
    requests = count(),
    errors = countIf(span.status_code == "error")
  }, by:{dt.entity.service}
| sort requests desc
| limit 20
```

```dql
// ✅ EFFICIENT: Summarize at source with error rate calculation
fetch spans
| filter span.kind == "server"
| summarize {
    total_requests = count(),
    error_count = countIf(span.status_code == "error"),
    p50_duration_ms = percentile(duration, 50) / 1000000,
    p95_duration_ms = percentile(duration, 95) / 1000000
  }, by:{dt.entity.service}
```

```
| fieldsAdd error_rate_pct = (error_count * 100.0) / total_requests
| sort total_requests desc
| limit 20
```

```dql
// ⚠️ MORE EXPENSIVE: Many percentiles are costlier
// Use only the percentiles you actually need
fetch spans
| filter span.kind == "server"
| summarize {
    p50 = percentile(duration, 50) / 1000000,
    p75 = percentile(duration, 75) / 1000000,
    p90 = percentile(duration, 90) / 1000000,
    p95 = percentile(duration, 95) / 1000000,
    p99 = percentile(duration, 99) / 1000000
  }, by:{dt.entity.service}
| limit 20
```

```dql
// Time-bucketed aggregations for trends
fetch spans
| filter span.kind == "server"
| fieldsAdd time_bucket = bin(start_time, 5m)
| summarize {
    request_count = count(),
    avg_duration_ms = avg(duration) / 1000000
  }, by:{time_bucket, dt.entity.service}
| sort time_bucket desc, request_count desc
| limit 100
```

---

7. High-Cardinality Grouping

Avoid grouping by high-cardinality fields (fields with many unique values).

```dql
// ❌ BAD: Grouping by high-cardinality field
// trace.id could have millions of unique values!
fetch spans
| summarize {trace_span_count = count()}, by:{trace.id}
| limit 10
```

```dql
```

```
// ✅ GOOD: Group by lower-cardinality fields
fetch spans
| summarize {span_count = count()}, by:{dt.entity.service, span.name}
| sort span_count desc
| limit 20
```

```dql
// ✅ GOOD: If you need trace analysis, filter first
fetch spans
| filter span.status_code == "error"      // Reduce first
| filter start_time >= now() - 15m        // Narrow time
| summarize {
    span_count = count(),
    services = collectDistinct(dt.entity.service)
  }, by:{trace.id}
| sort span_count desc
| limit 20
```

---

8. Production Query Patterns

Optimized query templates for common production use cases.

```dql
// Production Pattern: Service Health Dashboard
// Optimized for dashboard refresh (low DDU)
fetch spans, bucket: {"default_spans"}
| filter span.kind == "server"
| summarize {
    requests = count(),
    errors = countIf(span.status_code == "error"),
    avg_latency_ms = avg(duration) / 1000000
  }, by:{dt.entity.service}
| fieldsAdd error_rate = (errors * 100.0) / requests
| sort requests desc
| limit 20
```

```dql
// Production Pattern: Error Investigation
// Fast targeted query for debugging
fetch spans, bucket: {"default_spans"}
| filter span.kind == "server"
| filter span.status_code == "error"
| filter start_time >= now() - 30m
```

```
| fields start_time,
         dt.entity.service,
         span.name,
         span.status_message,
         trace.id
| sort start_time desc
| limit 50
```

```dql
// Production Pattern: Latency Trend Analysis
// Using bin() for time-series without makeTimeseries
fetch spans, bucket: {"default_spans"}
| filter span.kind == "server"
| filter isNotNull(dt.entity.service)
| fieldsAdd time_bucket = bin(start_time, 5m)
| summarize {
    request_count = count(),
    p95_latency_ms = percentile(duration, 95) / 1000000
  }, by:{time_bucket}
| sort time_bucket desc
| limit 50
```

```dql
// Production Pattern: Top Slow Operations
// Focus on actionable data
fetch spans, bucket: {"default_spans"}
| filter span.kind == "server"
| filter duration > 1s
| summarize {
    occurrence_count = count(),
    avg_duration_ms = avg(duration) / 1000000,
    max_duration_ms = max(duration) / 1000000
  }, by:{dt.entity.service, span.name}
| sort avg_duration_ms desc
| limit 20
```

```dql
// Production Pattern: Well-optimized comprehensive query
// Follows all optimization rules
fetch spans
| filter start_time >= now() - 1h          // 1. Appropriate time range
| filter isNotNull(dt.entity.service)      // 2. Filter early (indexed)
| filter span.kind == "server"             // 2. Additional early filter
(indexed)
| filter span.status_code == "error"       // 3. Indexed field
```

```
| fields start_time, span.name, duration, http.route  // 4. Only needed
fields
| summarize {
    error_count = count(),
    avg_duration_ms = avg(duration) / 1000000
  }, by:{span.name, http.route}            // 5. Low cardinality
| sort error_count desc
| limit 20                                 // 6. Limited results
```

---

9. Performance Checklist

Use this checklist before running production queries:

```
Query Optimization Checklist:

□ 1. Time range - Is it as short as possible for my needs?
□ 2. Filter early - Are filters immediately after fetch?
□ 3. Indexed fields - Am I filtering on indexed fields first?
      • trace.id, span.id, dt.entity.service
      • span.name, span.kind, span.status_code
      • start_time
□ 4. Field selection - Am I selecting only needed fields?
□ 5. Low cardinality - Are my group-by fields low cardinality?
□ 6. Result limits - Do I have appropriate limits?
□ 7. Bucket targeting - Am I querying specific buckets if known?
```

Summary

In this notebook, you learned:

✅ **Query cost factors** and optimization priorities
✅ **Filter early pattern** to reduce data scanned
✅ **Indexed fields** for faster query execution
✅ **Field selection** to minimize data transfer
✅ **Time range optimization** for cost control
✅ **Aggregation efficiency** for summarized results
✅ **High-cardinality grouping** pitfalls to avoid
✅ **Production patterns** for common use cases
✅ **Performance checklist** for query review

Series Complete! 🎉

You have completed the **Spans & Distributed Tracing** notebook series!

What You've Learned:

1. **Fundamentals** – Span structure and distributed tracing concepts
2. **Querying** – DQL syntax for effective span analysis
3. **Troubleshooting** – Error detection and root cause analysis
4. **Topology** – Service dependencies and flow visualization
5. **Analytics** – Advanced metrics and trend analysis
6. **Security** – Security monitoring and compliance with spans
7. **Buckets & Pipeline** – Data architecture, OpenPipeline, and governance
8. **Cost Optimization** – Efficient query patterns and indexed fields

Next Steps:

- Apply these patterns to your own Dynatrace environment
- Build dashboards using the optimized query patterns
- Configure alerts based on span analytics
- Set up OpenPipeline for data optimization
- Explore Davis AI integration for intelligent analysis