Wikiprint Book

Title: Optymalizacja zapytań SQL

Subject: eDokumenty - elektroniczny system obiegu dokumentów, workflow i CRM - DeployerGuide/Others/SQLPerformanceTips

Version: 1

Date: 05/07/24 16:51:10

Table of Contents

Optymalizacja zapytań SQL

Postgres pozwala na wykonywanie zaawansowanych konstukcji z zagnieżdżonymi zapytaniami SELECT dla których można wykonywać warunki JOIN itd. Jest to bardzo wygodne, jednak należy mieć na względnie kolejność wykonywania zapytań i starać się w podzapytaniach operujących na duzych nieprzefiltrowanych porcjach danych używac jak najmniej JOIN-ów.

Przykładowe zapytanie bez optymalizacji wykonuje się 10sec. Główne obciążenie jest spowodowane wykorzystaniem JOIN na widoku *features_text_view* dla każdego rekordu rcp_cards.

```
SELECT prc_id AS keyval, 'PROCESS' as clsnam, prtpnm , symbol , p.dscrpt , fullnm , name_1 , d.devcid, d.sernum, p.dsexid,
d.name__, dc.decanm AS category,
(SELECT sum(vnetto) FROM fk_elements_view WHERE is_del IS FALSE AND rcp_id IN (SELECT rcp_id FROM rcp_cards rcp WHERE rcp.

(SELECT sum(((extract(EPOCH FROM rcp.rlend_) - extract(EPOCH FROM rcp.rlstrt))/3600)::numeric(12,2)) AS dur
FROM rcp_cards_view rcp JOIN orgtree_view o ON rcp.emp_id = o.usr_id
WHERE rcp.prc_id = p.prc_id) AS czas_pracy,

(SELECT sum(wart) FROM (
SELECT sum(((extract(EPOCH FROM rcp.rlend_) - extract(EPOCH FROM rcp.rlstrt))/3600)::numeric(12,2)) * COALESCE(ftv.data__:
FROM rcp_cards_view rcp JOIN orgtree_view o ON rcp.emp_id = o.usr_id
LEFT JOIN features_text_view ftv ON ftv.tbl_id = rcp.emp_id AND ftv.featid = 326 --139
WHERE rcp.is_del IS FALSE AND rcp.prc_id = p.prc_id GROUP BY rcp.emp_id, ftv.data__) x) AS wartosc_czasu_pracy

FROM processes_view p
INNER JOIN devices d USING (devcid)
LEFT JOIN devices_category dc USING (decaid)
WHERE p.is_del IS NOT TRUE
ORDER BY category
```

Jest to zbędne, gdyż można pierwsze policzyć i pogrupować sumę czasu dla poszczególnych pracowników a później pomnożyć razy ich stawkę pobraną z widoku osobnym zapytaniem które już tylko zwróci tyle rekordów ilu jest pracowników.

```
SELECT prc_id AS keyval, 'PROCESS' as clsnam, prtpnm , symbol , p.dscrpt , fullnm , name_1 , d.devcid, d.sernum, p.dsexid,
d.name__, dc.decanm AS category,
(SELECT sum(vnetto) FROM fk_elements_view WHERE is_del IS FALSE AND rcp_id IN (SELECT rcp_id FROM rcp_cards rcp WHERE rcp.

(SELECT sum(((extract(EPOCH FROM rcp.rlend_) - extract(EPOCH FROM rcp.rlstrt))/3600)::numeric(12,2)) AS dur
FROM rcp_cards_view rcp JOIN orgtree_view o ON rcp.emp_id = o.usr_id
WHERE rcp.prc_id = p.prc_id) AS czas_pracy,

(SELECT sum(COALESCE(ftv.data__::numeric(12,2),0) * dur)::numeric(12,2) FROM (
SELECT sum(((extract(EPOCH FROM rcp.rlend_) - extract(EPOCH FROM rcp.rlstrt))/3600)::numeric(12,2)) AS dur, rcp.emp_id
FROM rcp_cards_view rcp
WHERE rcp.is_del IS FALSE AND rcp.prc_id = p.prc_id GROUP BY rcp.emp_id) x
INNER JOIN users u ON x.emp_id = u.usr_id
LEFT JOIN features_text_view ftv ON ftv.tbl_id = x.emp_id AND ftv.featid = 326 --139
) AS wartosc_czasu_pracy

FROM processes_view p
INNER JOIN devices d USING (devcid)
LEFT JOIN devices_category dc USING (decaid)
WHERE p.is_del IS NOT TRUE
--AND p.dsexid IN (181)
--AND p.devcid = {DEVCID}
ORDER BY category
```