

Milestone 8 Bewijs in PDF

Maxim Derboven - INF203

Overzicht vergelijking:

Tabel info voor partitionering:

	SEGMENT_NAME	SEGMENT_TYPE	MB	TABLE_COUNT
1	PERFORMANCES	TABLE	19	502505

Query:

```
-- Gemiddelde dagen tussen start en einde van film performances (die starten op een bepaalde dagen) (in dagen) per theater
SELECT /*+ FULL(P) */ T.THEATER_ID, T.NAME, ROUND(AVG(CAST(P.ENDTIME as DATE)-CAST(P.STARTTIME as DATE))) AS "Gemiddelde dagen per performance"
FROM THEATERS T
    JOIN HALLS H on H.THEATER_ID = T.THEATER_ID
    JOIN PERFORMANCES P on P.HALL_ID = H.HALL_ID
WHERE P.STARTTIME BETWEEN TO_DATE('01-02-2018 00:00', 'DD-MM-YYYY HH24:MI') AND TO_DATE('01-02-2018 23:59', 'DD-MM-YYYY HH24:MI')
GROUP BY T.THEATER_ID, T.NAME, TO_CHAR(P.STARTTIME, 'YYYY-MM');
```

Explain plan

Operation	Params	Rows	Total Cost	Raw Desc
Select		13	660.0	cpu_cost = 209131666, io_cost = 653
Group By (HASH GROUP BY)		13	660.0	cpu_cost = 209131666, io_cost = 653
Hash Join		13	659.0	cpu_cost = 179522657, io_cost = 653
Full Scan (TABLE ACCESS FULL)	table: PERFORMANCES;	13	653.0	cpu_cost = 148450191, io_cost = 648
Merge Join		1005	6.0	cpu_cost = 30370017, io_cost = 5
Index Scan (TABLE ACCESS BY INDEX R)	table: THEATERS;	25	2.0	cpu_cost = 24743, io_cost = 2
Full Index Scan (INDEX FULL SCAN)	index: THEATER_PK;	25	1.0	cpu_cost = 12121, io_cost = 1
Sort (SORT JOIN)		1005	4.0	cpu_cost = 30345274, io_cost = 3
Full Scan (TABLE ACCESS FULL)	table: HALLS;	1005	3.0	cpu_cost = 286857, io_cost = 3

NA partitionering:

Partitie script + uitleg partitie sleutel

Ik partitioneer de performances elke maand op hun start tijd ('starttime'). De eerste performance vindt plaats op 2018/01/01

```
DROP TABLE PERFORMANCES CASCADE CONSTRAINTS PURGE;
CREATE TABLE PERFORMANCES
(
    performance_id INTEGER GENERATED ALWAYS AS IDENTITY CONSTRAINT fk_performance PRIMARY KEY,
    movie_id       INTEGER NOT NULL CONSTRAINT fk_movie_performance REFERENCES MOVIES(MOVIE_ID) ON DELETE CASCADE,
    hall_id        INTEGER NOT NULL CONSTRAINT fk_hall_performance REFERENCES HALLS(HALL_ID) ON DELETE CASCADE,
    starttime      TIMESTAMP NOT NULL,
    endtime        TIMESTAMP NOT NULL
);
PARTITION BY RANGE (starttime)
    INTERVAL (NUMTOYMINTERVAL(1, 'MONTH'))
(
    PARTITION p0 VALUES LESS THAN (TO_DATE('2018/02/01', 'YYYY/MM/DD')),
    PARTITION p1 VALUES LESS THAN (TO_DATE('2018/03/01', 'YYYY/MM/DD'))
);
```

Tabel info NA partitionering:

	SEGMENT_NAME	SEGMENT_TYPE	MB	TABLE_COUNT
1	PERFORMANCES	TABLE PARTITION	21	502505

Query: → moet dezelfde zijn

Explain plan na partitionering

Operation	Params	Rows	Total Cost	Raw Desc
▼ Select		12	10.0	cpu_cost = 60796824, io_cost = 8
▼ [H] Group By (HASH GROUP BY)		12	10.0	cpu_cost = 60796824, io_cost = 8
▼ Hash Join		12	9.0	cpu_cost = 31188045, io_cost = 8
▼ Unknown (PARTITION RANGE SINGLE)		12	3.0	cpu_cost = 115728, io_cost = 3
Full Scan (TABLE ACCESS FULL)	table: PERFORMANCES;	12	3.0	cpu_cost = 115728, io_cost = 3
▼ Merge Join		1005	6.0	cpu_cost = 30370017, io_cost = 5
▼ Index Scan (TABLE ACCESS BY INDEX F	table: THEATERS;	25	2.0	cpu_cost = 24743, io_cost = 2
Full Index Scan (INDEX FULL SCAN)	index: THEATHER_PK;	25	1.0	cpu_cost = 12121, io_cost = 1
▼ Sort (SORT JOIN)		1005	4.0	cpu_cost = 30345274, io_cost = 3
Full Scan (TABLE ACCESS FULL)	table: HALLS;	1005	3.0	cpu_cost = 286857, io_cost = 3

Conclusie:

De costs dalen omdat hij sneller doorheen de data kan door de bepaalde partitie grenzen.
De tijd zal daarom ook afnemen. Hoe meer partities hoe groter de database wordt.