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 ginde van film performances (die starten op een bepaalde dagen) (in dagen) per theather

SELECT /*+ FULL(P) */ T.THEATHER_ID,T.NAME, ROUND(AVG(CAST(P.ENDTIME as DATE)-CAST(P.STARTTIME AS DATE))) AS "Gemiddelde dagen per performance"

FROM THEATHERS T

JOIN HALLS H on H.THEATHER_ID = T.THEATHER_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.THEATHER_ID, T.NAME, TO_CHAR(P.STARTTIME,'YYYY-MM');
```

Explain plan

Operation	Params	Rows	Total Cost	Raw Desc
✓ ← Select				cpu_cost = 209131666, io_cost = 653
✓ [≡] Group By (HASH GROUP BY)				cpu_cost = 209131666, io_cost = 653
✓ 斐 Hash Join		13	659.0	cpu_cost = 179522657, io_cost = 653
■ Full Scan (TABLE ACCESS FULL)				cpu_cost = 148450191, io_cost = 648
✓				cpu_cost = 30370017, io_cost = 5
Index Scan (TABLE ACCESS BY INDEX R				cpu_cost = 24743, io_cost = 2
Full Index Scan (INDEX FULL SCAN)	index: THEATHER_PK;			cpu_cost = 12121, io_cost = 1
✓ Sort (SORT JOIN)				cpu_cost = 30345274, io_cost = 3
Ⅲ 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 vondt plaats op 2018/01/01

```
DROP TABLE PERFORMANCES

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'))

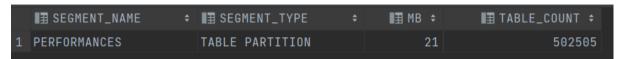
(

PARTITION p0 VALUES LESS THAN (TO_DATE('2018/03/01', 'YYYY/MM/DD')),

PARTITION p1 VALUES LESS THAN (TO_DATE('2018/03/01', 'YYYY/MM/DD'))

);
```

Tabel info NA partitionering:



Query: \rightarrow moet dezelfde zijn

Explain plan na partitionering



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.