Milestone 8 Bewijs in PDF

Matthias Vermeiren - TIFLE

Overzicht vergelijking:

```
BEGIN
    DBMS_STATS.GATHER_TABLE_STATS( OWNNAME: 'PROJECT',  TABNAME: 'ARTISTS');
    DBMS_STATS.GATHER_TABLE_STATS( OWNNAME: 'PROJECT',  TABNAME: 'RECORDING_STUDIOS');
    DBMS_STATS.GATHER_TABLE_STATS( OWNNAME: 'PROJECT',  TABNAME: 'ROOMS');
    DBMS_STATS.GATHER_TABLE_STATS( OWNNAME: 'PROJECT',  TABNAME: 'EQUIPMENT');
END;

-- Table size
SELECT segment_name, segment_type, sum(bytes/1024/1024) MB,
       (SELECT COUNT(*) FROM EQUIPMENT)  as table_count
FROM DBA_SEGMENTS
WHERE SEGMENT_NAME = 'EQUIPMENT'
GROUP BY segment_name, segment_type;
```

Tabel info voor partitionering:

	SEGMENT_NAME	SEGMENT_TYPE	MB	TABLE_COUNT
1	EQUIPMENT	TABLE	72	400000

Query:

```
-- Overview of average rent per hour for equipment where the recording studios are located in Brussel.
SELECT s.STUDIO_NAME, s.ADDRESS, r.ROOM_NAME, ROUND(AVG(e.RENTPERHOUR)) AS "Average Rent Per Hour"
FROM RECORDING_STUDIOS S
    JOIN ROOMS R ON s.STUDIO_CODE = R.RECORDING_STUDIOS_STUDIO_CODE
    JOIN EQUIPMENT e ON e.ROOMS_ROOM_CODE = r.ROOM_CODE
WHERE upper(S.LOCATION) = 'BRUSSEL'
GROUP BY s.STUDIO_NAME, s.ADDRESS, r.ROOM_NAME
ORDER BY s.STUDIO_NAME, r.ROOM_NAME;
```

Explain plan

Output | Result 28 | Plan

Operation	Params	Rows	Total Cost	Raw Desc
Select		400	2475.0	cpu_cost = 277437961, io_cost = 2466
Order By (SORT ORDER BY)		400	2475.0	cpu_cost = 277437961, io_cost = 2466
Group By (HASH GROUP BY)		400	2475.0	cpu_cost = 277437961, io_cost = 2466
Hash Join		4211	2473.0	cpu_cost = 213655725, io_cost = 2466
Full Scan (TABLE ACCESS FULL)	table: ROOMS;	800	3.0	cpu_cost = 251607, io_cost = 3
Merge Join (MERGE JOIN CARTESIAN)		80000	2470.0	cpu_cost = 204684118, io_cost = 2463
Full Scan (TABLE ACCESS FULL)	table: RECORDING_STUDIOS;	1	3.0	cpu_cost = 42807, io_cost = 3
Sort (BUFFER SORT)		400000	2467.0	cpu_cost = 204641311, io_cost = 2460
Full Scan (TABLE ACCESS FULL)	table: EQUIPMENT;	400000	2467.0	cpu_cost = 204641311, io_cost = 2460

NA partitionering:

Partitie script + uitleg partitie sleutel

```
-- Partitionering
DROP TABLE EQUIPMENT CASCADE CONSTRAINTS PURGE;
CREATE TABLE EQUIPMENT
(       equipment_code  INTEGER GENERATED ALWAYS AS IDENTITY,
        rentperhour     NUMBER(3),  --M6
        equipmentname   VARCHAR2(30 CHAR),  --M6
        mixing_console  VARCHAR2(4 CHAR),
        monitors        VARCHAR2(10 CHAR),
        hardware        VARCHAR2(50),
        daw             VARCHAR2(10 CHAR),
        software        VARCHAR2(50 CHAR),
        synths          VARCHAR2(50 CHAR),
        vocal_mic       VARCHAR2(10),
        rooms_room_code INTEGER NOT NULL,
        ro_rec_stu_code INTEGER NOT NULL
)
    PARTITION BY RANGE(rooms_room_code)
    INTERVAL(50)
    (
        partition equipment_50 VALUES LESS THAN (50)
    );
```

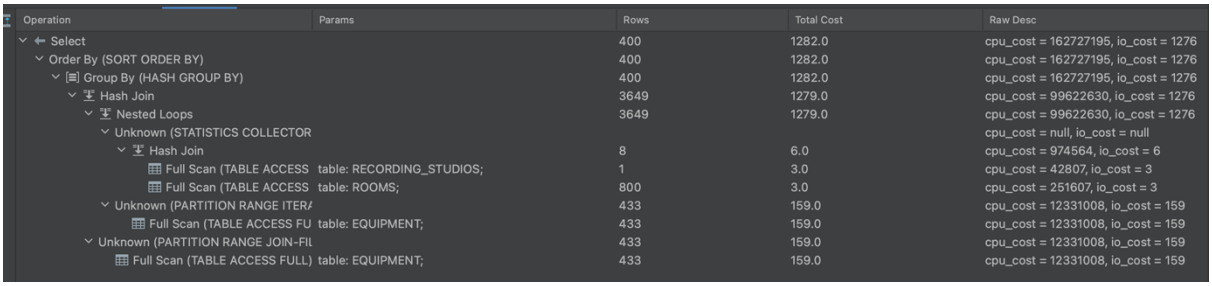
```
ALTER TABLE equipment ADD CHECK ( rentperhour BETWEEN 1 AND 200 );

ALTER TABLE equipment
    ADD CHECK ( mixing_console IN ( 'API', 'AVID', 'NEVE', 'SSL' ) );

ALTER TABLE equipment
    ADD CHECK ( monitors IN ( 'Adam', 'Barefoot', 'Dynaudio', 'Focal', 'Genelec') );

ALTER TABLE equipment
    ADD CHECK ( daw IN ( 'Ableton', 'Cubase', 'Logic', 'ProTools' ) );

ALTER TABLE equipment
    ADD CHECK ( vocal_mic IN ( 'AKG', 'Neumann', 'Shure', 'Sony' ) );

ALTER TABLE equipment
    ADD CONSTRAINT equipment_pk PRIMARY KEY ( equipment_code,
                                              rooms_room_code,
                                              ro_rec_stu_code );
ALTER TABLE equipment
    ADD CONSTRAINT equipment_rooms_fk FOREIGN KEY ( rooms_room_code,
                                                    ro_rec_stu_code )
        REFERENCES rooms ( room_code,
                           recording_studios_studio_code )
            ON DELETE CASCADE;

alter session set "_partition_large_extents"= false;
```

Tabel info NA partitionering:

	SEGMENT_NAME	SEGMENT_TYPE	MB	TABLE_COUNT
1	EQUIPMENT	TABLE PARTITION	80.125	400000

Query: → moet dezelfde zijn

Explain plan na partitionering



Operation	Params	Rows	Total Cost	Raw Desc
Select		400	1282.0	cpu_cost = 162727195, io_cost = 1276
Order By (SORT ORDER BY)		400	1282.0	cpu_cost = 162727195, io_cost = 1276
Group By (HASH GROUP BY)		400	1282.0	cpu_cost = 162727195, io_cost = 1276
Hash Join		3649	1279.0	cpu_cost = 99622630, io_cost = 1276
Nested Loops		3649	1279.0	cpu_cost = 99622630, io_cost = 1276
Unknown (STATISTICS COLLECTOR				cpu_cost = null, io_cost = null
Hash Join		8	6.0	cpu_cost = 974564, io_cost = 6
Full Scan (TABLE ACCESS	table: RECORDING_STUDIOS;	1	3.0	cpu_cost = 42807, io_cost = 3
Full Scan (TABLE ACCESS	table: ROOMS;	800	3.0	cpu_cost = 251607, io_cost = 3
Unknown (PARTITION RANGE ITER/		433	159.0	cpu_cost = 12331008, io_cost = 159
Full Scan (TABLE ACCESS FU	table: EQUIPMENT;	433	159.0	cpu_cost = 12331008, io_cost = 159
Unknown (PARTITION RANGE JOIN-FIL		433	159.0	cpu_cost = 12331008, io_cost = 159
Full Scan (TABLE ACCESS FULL)	table: EQUIPMENT;	433	159.0	cpu_cost = 12331008, io_cost = 159

Conclusie:

De kosten van de resources zijn lager na het partitioneren, aangezien de database niet meer door de gehele databank moet zoeken. Na partitionering zijn de kosten van de resources gedaald met -48,20%. Het geheugen is echter toegenomen met 11,28%. Hieruit kan worden afgeleid dat we via partitionering met een trade-off zitten tussen een daling van de cost van de resources en een toename in het geheugengebruik.