Bewijs Milestone 8 S1

Saif Qudaih

Student 1:

Overzicht vergelijking:

Tabel Rental voor partitionering:

1 row

	SEGMENT_NAME	SEGMENT_TYPE	MB	TABLE_COUNT
1	RENTAL	TABLE	56	400000

Stap 2: analyse voor optimalisatie:

```sql
SELECT
    rentalStatus,
    COUNT(rentalID) AS total_rentals,
    ROUND(AVG(totalCost),2) AS avg_cost,
    MAX(totalCost) AS max_cost,
    MIN(totalCost) AS min_cost,
    SUM(totalCost) AS total_revenue
FROM Rental
WHERE rentalStartDate BETWEEN TO_DATE('2024-01-01', 'YYYY-MM-DD') AND TO_DATE('2024-12-31', 'YYYY-MM-DD')
  AND totalCost BETWEEN 100 AND 2000
GROUP BY rentalStatus
ORDER BY total_rentals DESC;
```

Operation	Params	Rows	Total Cost	Raw Desc
Select		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Unknown (PX COORDINATOR)				cpu_cost = null, io_cost = null
Unknown (PX SEND QC (ORDER))		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Order By (SORT ORDER BY)		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Unknown (PX RECEIVE)		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Unknown (PX SEND RANGE)		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Group By (HASH GROUP BY)		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Unknown (PX RECEIVE)		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Unknown (PX SEND HASH)		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Group By (HASH GROUP BY)		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Unknown (PX BLOCK ITERATOR)		61530	1064.0	cpu_cost = 96409022, io_cost = 1061
Full Scan (TABLE ACCESS FULL	table: RENTAL;	61530	1064.0	cpu_cost = 96409022, io_cost = 1061

Bewijs Milestone 8 S1

Saif Qudaih

NA partitionering:

```oracle
CREATE TABLE Rental (
                        rentalID NUMBER GENERATED ALWAYS AS IDENTITY,
                        chassisNr VARCHAR2(50),
                        customerID INTEGER NOT NULL,
                        rentalStartDate DATE NOT NULL,
                        rentalEndDate DATE NOT NULL,
                        rentalStatus VARCHAR2(20) NOT NULL,
                        paymentDetails VARCHAR2(200),
                        customerFeedback VARCHAR2(500),
                        insuranceDetails VARCHAR2(200),
                        totalCost DECIMAL(10, 2) NOT NULL,
                        shopID INTEGER NOT NULL,
                        CONSTRAINT RENTAL_PK PRIMARY KEY (rentalID),
                        CONSTRAINT RENTAL_FK_chassisNr FOREIGN KEY (chassisNr)
                            REFERENCES MotorBike (chassisNr),
                        CONSTRAINT RENTAL_FK_CUSTOMERID FOREIGN KEY (customerID)
                            REFERENCES Customer (customerID),
                        CONSTRAINT RENTAL_FK_SHOPID FOREIGN KEY (shopID)
                            REFERENCES Shop (shopID),
                        CONSTRAINT UNIQUE_RENTAL_COMBINATION UNIQUE (rentalID, customerID, chassisNr, rentalStartDate)
)
    PARTITION BY RANGE (rentalStartDate)
    INTERVAL (NUMTOYMINTERVAL(1, 'MONTH'))
(
    PARTITION p202301 VALUES LESS THAN (TO_DATE('2023-02-01', 'YYYY-MM-DD')),
    PARTITION p202302 VALUES LESS THAN (TO_DATE('2023-03-01', 'YYYY-MM-DD')),
    PARTITION p202303 VALUES LESS THAN (TO_DATE('2023-04-01', 'YYYY-MM-DD'))
);
```

Tabel Rental NA partitionering:

Output | Result 8 | Plan

1 row

	SEGMENT_NAME	SEGMENT_TYPE	MB	TABLE_COUNT
1	RENTAL	TABLE PARTITION	104	400000

Bewijs Milestone 8 S1

Saif Qudaih

Explain plan na partitionering

Output | Result 8 | Plan

Operation	Params	Rows	Total Cost	Raw Desc
Select		5	5.0	cpu_cost = 110360701, io_cost = 2
Unknown (PX COORDINATOR)				cpu_cost = null, io_cost = null
Unknown (PX SEND QC (ORDER))		5	5.0	cpu_cost = 110360701, io_cost = 2
Order By (SORT ORDER BY)		5	5.0	cpu_cost = 110360701, io_cost = 2
Unknown (PX RECEIVE)		5	5.0	cpu_cost = 110360701, io_cost = 2
Unknown (PX SEND RANGE)		5	5.0	cpu_cost = 110360701, io_cost = 2
Group By (HASH GROUP BY)		5	5.0	cpu_cost = 110360701, io_cost = 2
Unknown (PX RECEIVE)		5	5.0	cpu_cost = 110360701, io_cost = 2
Unknown (PX SEND HASH)		5	5.0	cpu_cost = 110360701, io_cost = 2
Group By (HASH GROUP BY)		5	5.0	cpu_cost = 110360701, io_cost = 2
Unknown (PX BLOCK ITERATOR)		52639	2.0	cpu_cost = 34833, io_cost = 2
Full Scan (TABLE ACCESS FULL)	table: RENTAL;	52639	2.0	cpu_cost = 34833, io_cost = 2

Conclusie:

De partitionering van de Rental-tabel heeft de query prestaties aanzienlijk verbeterd door de totale kosten te verlagen van 1.067,0 naar 5,0. Het aantal geraadpleegde rijen is ook afgenomen van 61.530 naar 52.639, wat duidt op efficiëntere gegevensopvraging.

Dit geeft aan dat partitionering op rentalStartDate de query heeft geoptimaliseerd, waarschijnlijk door een betere afstemming met de filter voorwaarden van de query en efficiëntere gegevens scanning.

De substantiële vermindering van de totale kosten suggereert dat partition pruning effectief het gescande gegevensvolume beperkt, wat bijdraagt aan verbeterde query prestaties.