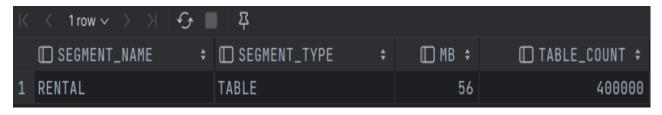
Bewijs Milestone 8 S1

Saif Qudaih

Student 1:

Overzicht vergelijking:

Tabel Rental voor partitionering:



Stap 2: analyse voor optimalisatie:



Operation	Params	Rows	Total Cost	Raw Desc
y ← 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
v 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:

```
CREATE TABLE Rental (

rentalID NUMBER GENERATED ALWAYS AS IDENTITY,
chassisNr VARCHAR2(50),
customerID INTEGER NOT NULL,
rentalEndDate DATE NOT NULL,
rentalEndDate DATE NOT NULL,
rentalEndDate DATE NOT NULL,
rentalEndDate DATE NOT NULL,
customerFeedDack VARCHAR2(200),
customerFeedDack VARCHAR2(200),
totalCost DECIMAL(10, 2) NOT NULL,
shopID INTEGER NOT NULL,
CONSTRAINT RENTAL_PK PRIMARY KEY (rentalID),
customerFeedDack VARCHAR2(200),
totalCost DECIMAL(10, 2) NOT NULL,
shopID INTEGER NOT NULL,
constraint Rental_PK primary Key (rentalID),
customerID REFERENCES MotomBike (chassisNr)

REFERENCES Notomer (customerID),
constraint Rental_FK_construction FOREIGN KEY (customerID)
REFERENCES Shop (shopID)
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-NH-DD')),
PARTITION p202302 VALUES LESS THAN (TO_DATE('2023-03-01', 'YYYY-NH-DD')),
PARTITION p202303 VALUES LESS THAN (TO_DATE('2023-04-01', 'YYYY-NH-DD'))

PARTITION p202303 VALUES LESS THAN (TO_DATE('2023-04-01', 'YYYY-NH-DD'))

PARTITION p202303 VALUES LESS THAN (TO_DATE('2023-04-01', 'YYYY-NH-DD'))
```

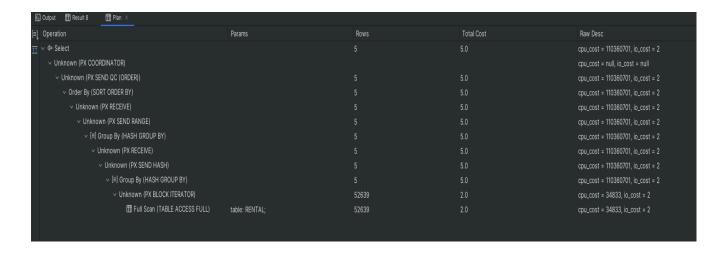
Tabel Rental NA partitionering:



Bewijs Milestone 8 S1

Saif Qudaih

Explain plan na partitionering



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.