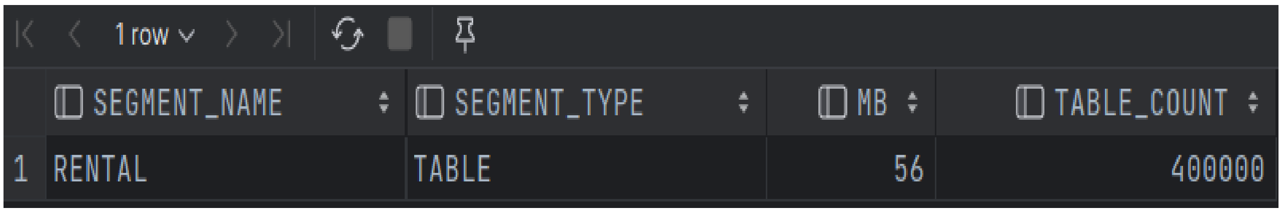
Bewijs Milestone 8

Said khalf **Saif Qudaih**

Student 1:

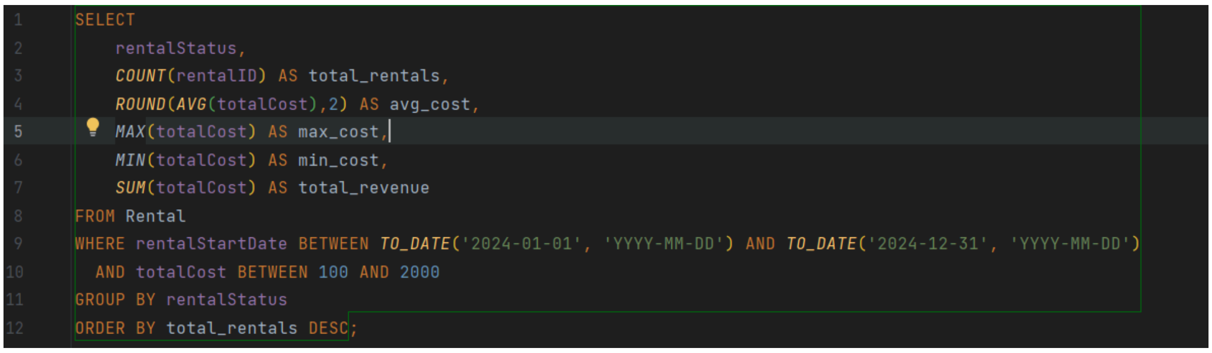
Overzicht vergelijking:

Tabel Rental voor partitionering:

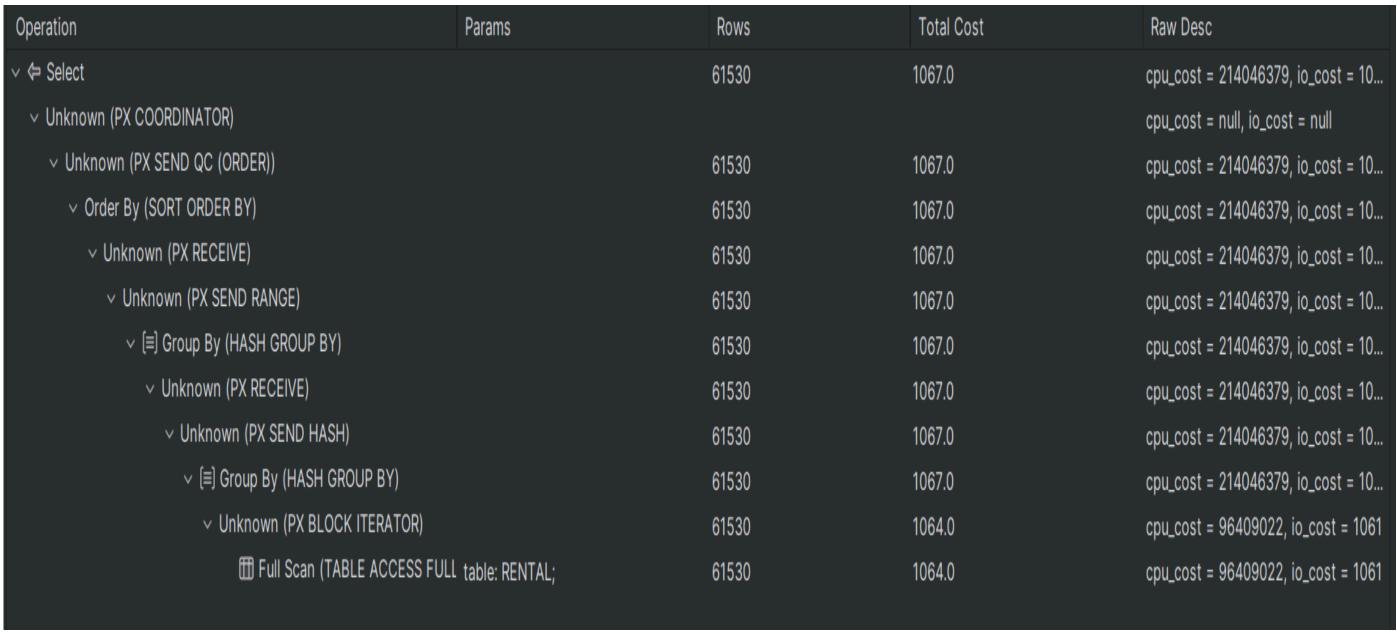


	SEGMENT_NAME	SEGMENT_TYPE	MB	TABLE_COUNT
1	RENTAL	TABLE	56	400000

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;
```

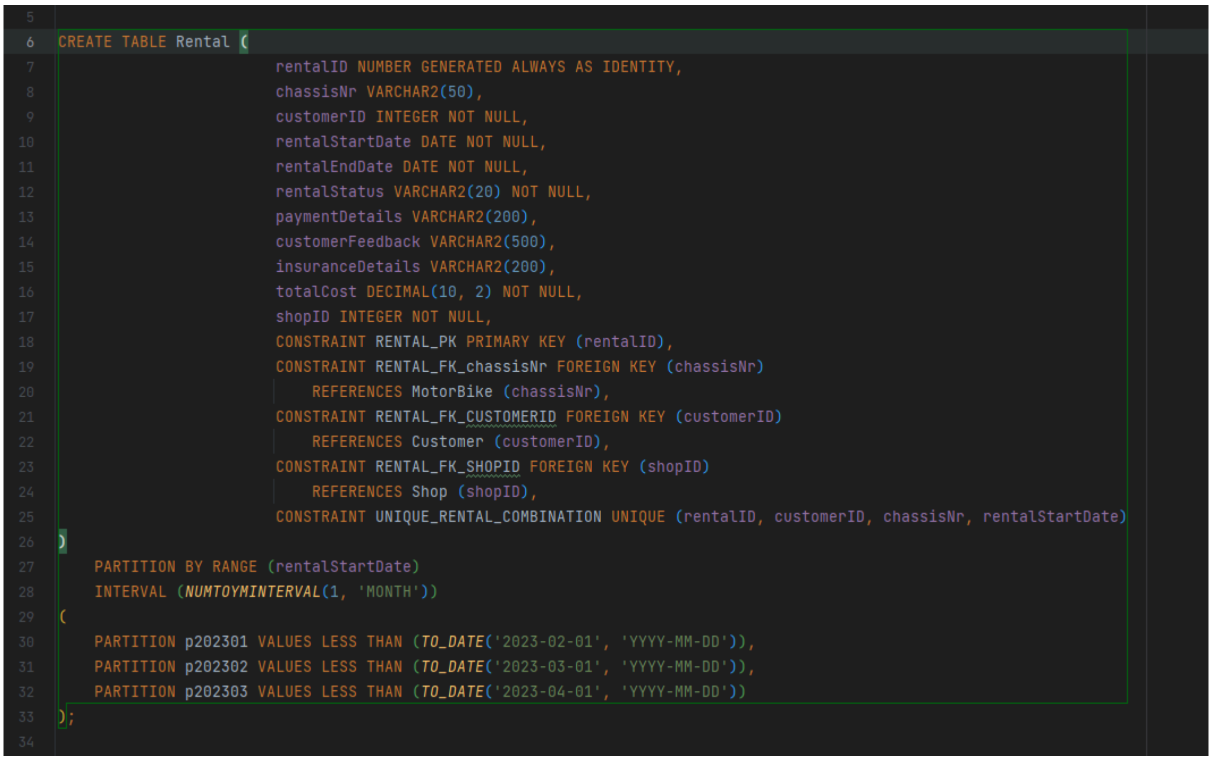


Operation	Params	Rows	Total Cost	Raw Desc
Select		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Unknown (PX COORDINATOR)				cpu_cost = null, io_cost = null
Unknown (PX SEND QC (ORDER))		61530	1067.0	cpu_cost = 214046379, io_cost = 10...
Order By (SORT ORDER BY)		61530	1067.0	cpu_cost = 214046379, 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

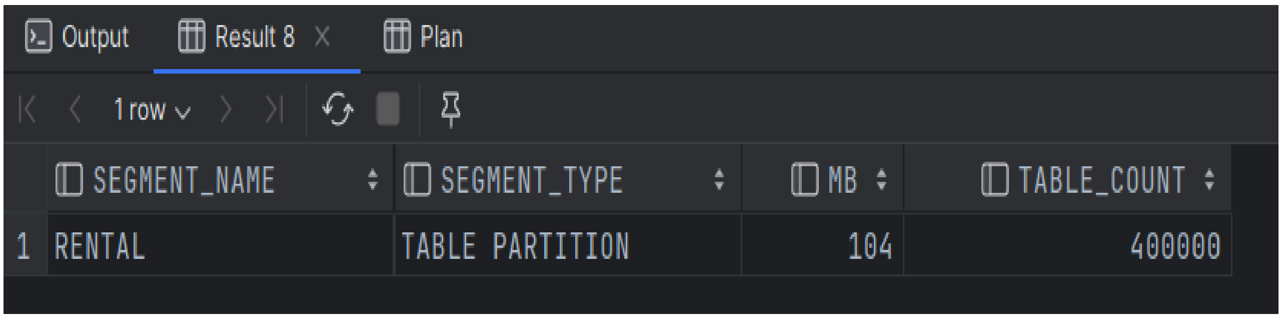
Said khalf **Saif Qudaih**

NA partitionering:



```sql
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:

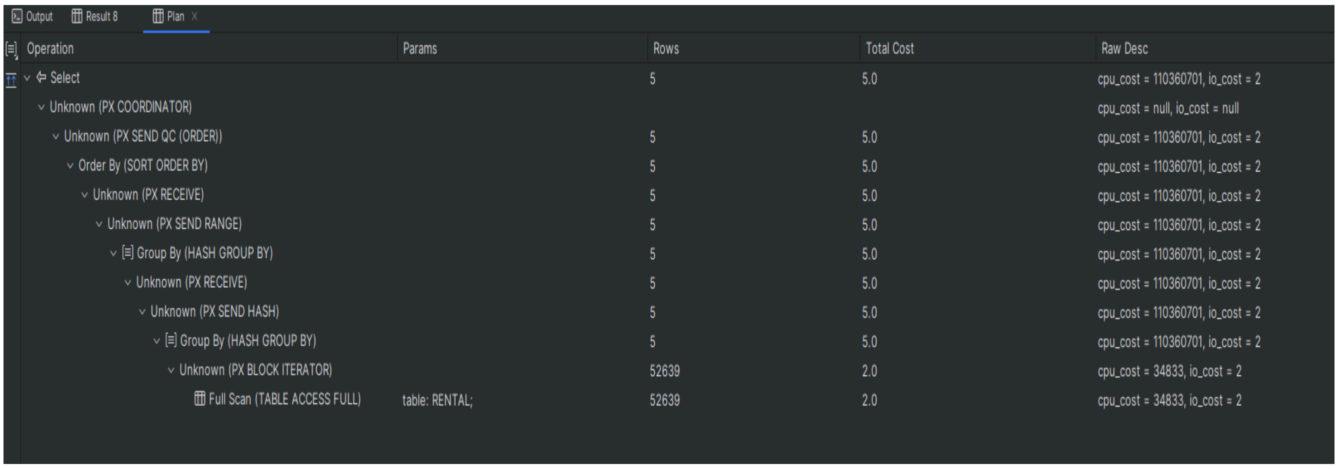


	SEGMENT_NAME	SEGMENT_TYPE	MB	TABLE_COUNT
1	RENTAL	TABLE PARTITION	104	400000

Bewijs Milestone 8

Said khalf **Saif Qudaih**

Explain plan na partitionering



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.

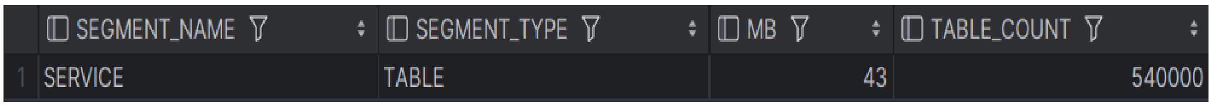
De substantiële vermindering van de totale kosten suggereert dat partition pruning effectief het gescande gegevensvolume beperkt, wat bijdraagt aan verbeterde query prestaties.

Bewijs Milestone 8

Said khalf **Saif Qudaih**

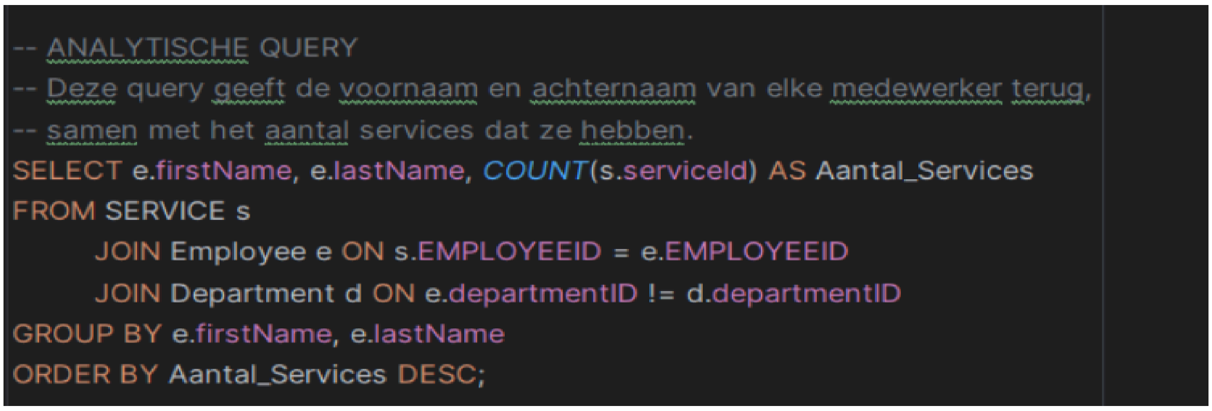
Student 2

Voor Materialized View:



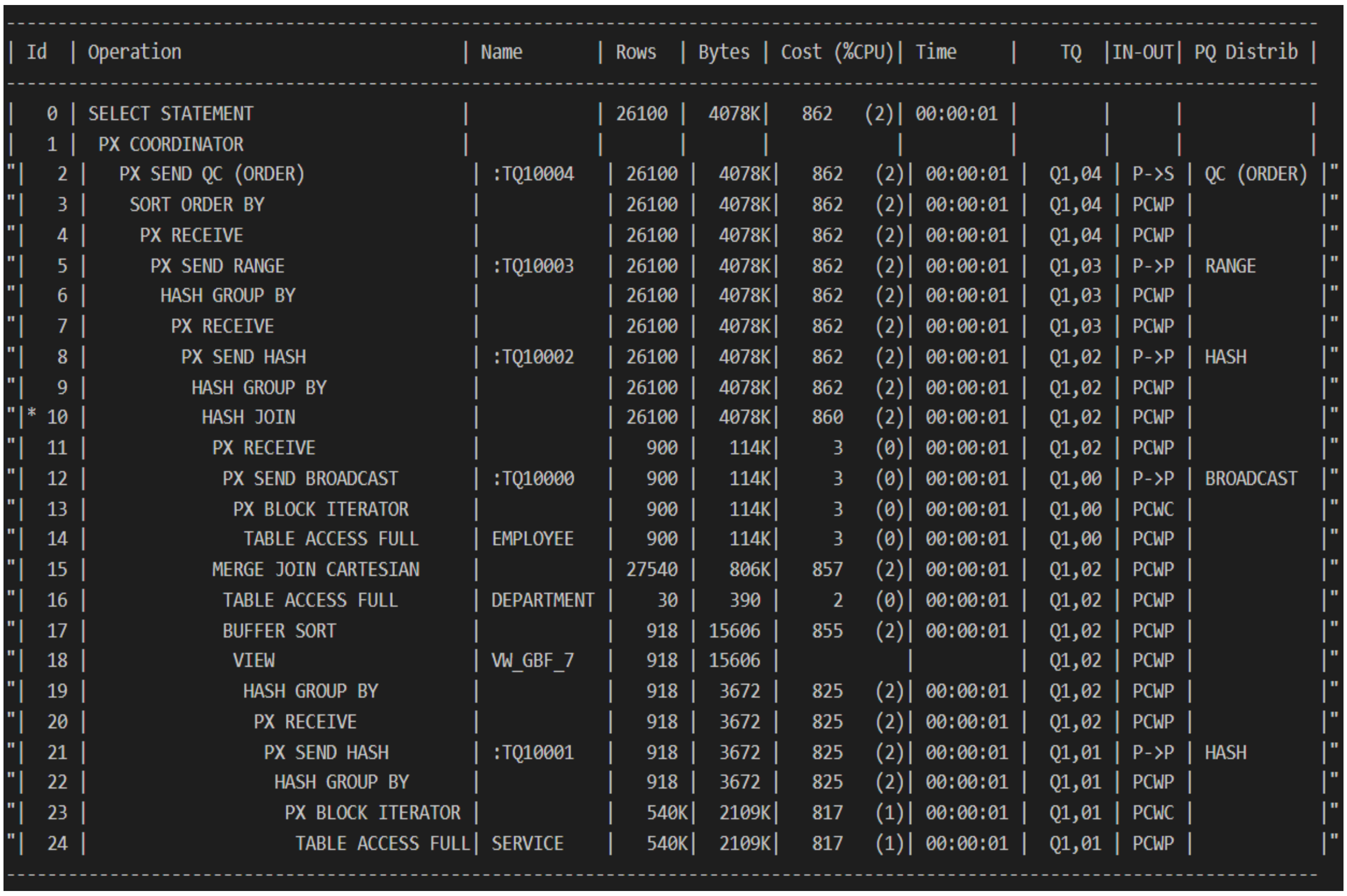
	SEGMENT_NAME	SEGMENT_TYPE	MB	TABLE_COUNT
1	SERVICE	TABLE	43	540000

Queries:



```sql
-- ANALYTISCHE QUERY
-- Deze query geeft de voornaam en achternaam van elke medewerker terug,
-- samen met het aantal services dat ze hebben.
SELECT e.firstName, e.lastName, COUNT(s.serviceId) AS Aantal_Services
FROM SERVICE s
        JOIN Employee e ON s.EMPLOYEEID = e.EMPLOYEEID
        JOIN Department d ON e.departmentID != d.departmentID
GROUP BY e.firstName, e.lastName
ORDER BY Aantal_Services DESC;
```

Explain plan:

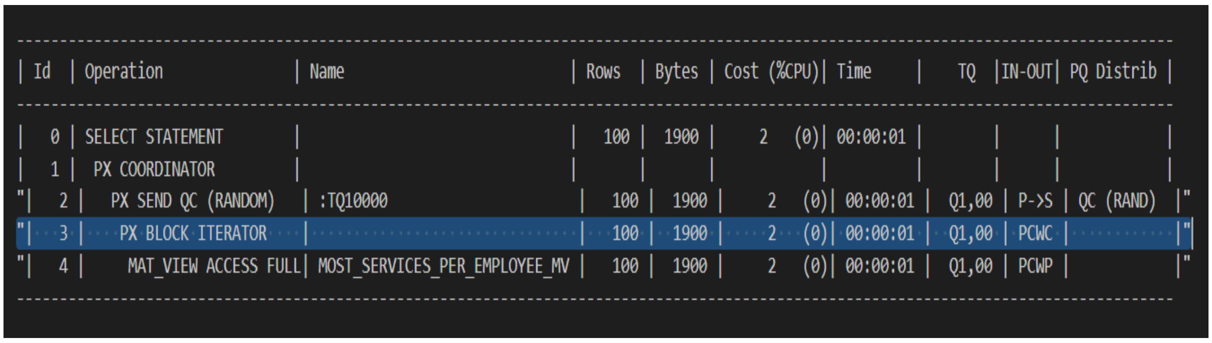


```
--------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                      | Name       | Rows  | Bytes | Cost (%CPU)| Time     |    TQ  |IN-OUT| PQ Distrib |
--------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |            | 26100 |  4078K|   862   (2)| 00:00:01 |        |      |            |
|   1 |  PX COORDINATOR                |            |       |       |            |          |        |      |            |
"|   2 |   PX SEND QC (ORDER)            | :TQ10004   | 26100 |  4078K|   862   (2)| 00:00:01 |  Q1,04 | P->S | QC (ORDER) |"
"|   3 |    SORT ORDER BY                |            | 26100 |  4078K|   862   (2)| 00:00:01 |  Q1,04 | PCWP |            |"
"|   4 |     PX RECEIVE                  |            | 26100 |  4078K|   862   (2)| 00:00:01 |  Q1,04 | PCWP |            |"
"|   5 |      PX SEND RANGE              | :TQ10003   | 26100 |  4078K|   862   (2)| 00:00:01 |  Q1,03 | P->P | RANGE      |"
"|   6 |       HASH GROUP BY             |            | 26100 |  4078K|   862   (2)| 00:00:01 |  Q1,03 | PCWP |            |"
"|   7 |        PX RECEIVE               |            | 26100 |  4078K|   862   (2)| 00:00:01 |  Q1,03 | PCWP |            |"
"|   8 |         PX SEND HASH            | :TQ10002   | 26100 |  4078K|   862   (2)| 00:00:01 |  Q1,02 | P->P | HASH       |"
"|   9 |          HASH GROUP BY          |            | 26100 |  4078K|   862   (2)| 00:00:01 |  Q1,02 | PCWP |            |"
"|* 10 |           HASH JOIN             |            | 26100 |  4078K|   860   (2)| 00:00:01 |  Q1,02 | PCWP |            |"
"|  11 |            PX RECEIVE           |            |   900 |   114K|     3   (0)| 00:00:01 |  Q1,02 | PCWP |            |"
"|  12 |             PX SEND BROADCAST   | :TQ10000   |   900 |   114K|     3   (0)| 00:00:01 |  Q1,00 | P->P | BROADCAST  |"
"|  13 |              PX BLOCK ITERATOR  |            |   900 |   114K|     3   (0)| 00:00:01 |  Q1,00 | PCWC |            |"
"|  14 |               TABLE ACCESS FULL | EMPLOYEE   |   900 |   114K|     3   (0)| 00:00:01 |  Q1,00 | PCWP |            |"
"|  15 |            MERGE JOIN CARTESIAN |            | 27540 |   806K|   857   (2)| 00:00:01 |  Q1,02 | PCWP |            |"
"|  16 |             TABLE ACCESS FULL   | DEPARTMENT |    30 |   390 |     2   (0)| 00:00:01 |  Q1,02 | PCWP |            |"
"|  17 |             BUFFER SORT         |            |   918 | 15606 |   855   (2)| 00:00:01 |  Q1,02 | PCWP |            |"
"|  18 |              VIEW               | VW_GBF_7   |   918 | 15606 |            |          |  Q1,02 | PCWP |            |"
"|  19 |               HASH GROUP BY     |            |   918 |  3672 |   825   (2)| 00:00:01 |  Q1,02 | PCWP |            |"
"|  20 |                PX RECEIVE       |            |   918 |  3672 |   825   (2)| 00:00:01 |  Q1,02 | PCWP |            |"
"|  21 |                 PX SEND HASH    | :TQ10001   |   918 |  3672 |   825   (2)| 00:00:01 |  Q1,01 | P->P | HASH       |"
"|  22 |                  HASH GROUP BY  |            |   918 |  3672 |   825   (2)| 00:00:01 |  Q1,01 | PCWP |            |"
"|  23 |                   PX BLOCK ITERATOR |        |   540K|  2109K|   817   (1)| 00:00:01 |  Q1,01 | PCWC |            |"
"|  24 |                    TABLE ACCESS FULL| SERVICE |   540K|  2109K|   817   (1)| 00:00:01 |  Q1,01 | PCWP |            |"
--------------------------------------------------------------------------------------------------------------------------
```

Bewijs Milestone 8

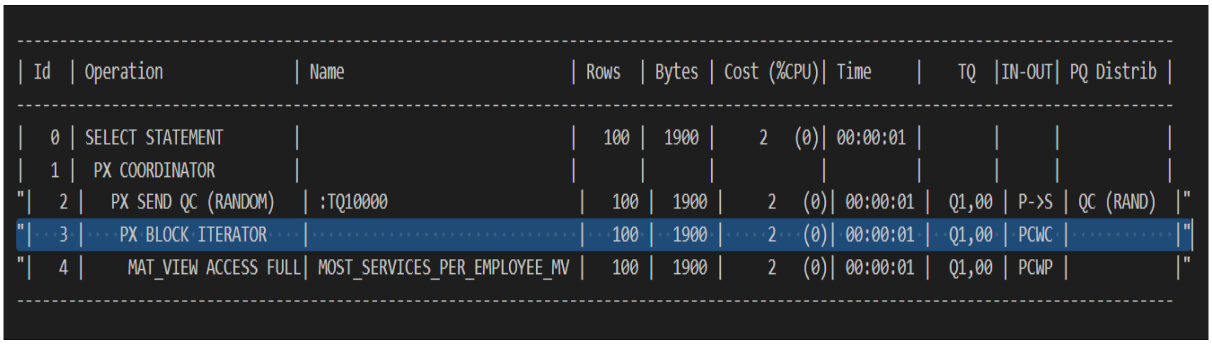
Said khalf **Saif Qudaih**

Explain Plan na Materialized View:



```
--------------------------------------------------------------------------------------------------------------------------
| Id  | Operation              | Name                        | Rows  | Bytes | Cost (%CPU)| Time     |    TQ  |IN-OUT| PQ Distrib |
--------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |                             |   100 |  1900 |     2   (0)| 00:00:01 |        |      |            |
|   1 |  PX COORDINATOR        |                             |       |       |            |          |        |      |            |
"|   2 |   PX SEND QC (RANDOM)  | :TQ10000                    |   100 |  1900 |     2   (0)| 00:00:01 |  Q1,00 | P->S | QC (RAND)  |"
"|   3 |    PX BLOCK ITERATOR   |                             |   100 |  1900 |     2   (0)| 00:00:01 |  Q1,00 | PCWC |            |"
"|   4 |     MAT_VIEW ACCESS FULL| MOST_SERVICES_PER_EMPLOYEE_MV |   100 |  1900 |     2   (0)| 00:00:01 |  Q1,00 | PCWP |            |"
--------------------------------------------------------------------------------------------------------------------------
```

Na nieuwe data:



```
--------------------------------------------------------------------------------------------------------------------------
| Id  | Operation              | Name                        | Rows  | Bytes | Cost (%CPU)| Time     |    TQ  |IN-OUT| PQ Distrib |
--------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |                             |   100 |  1900 |     2   (0)| 00:00:01 |        |      |            |
|   1 |  PX COORDINATOR        |                             |       |       |            |          |        |      |            |
"|   2 |   PX SEND QC (RANDOM)  | :TQ10000                    |   100 |  1900 |     2   (0)| 00:00:01 |  Q1,00 | P->S | QC (RAND)  |"
"|   3 |    PX BLOCK ITERATOR   |                             |   100 |  1900 |     2   (0)| 00:00:01 |  Q1,00 | PCWC |            |"
"|   4 |     MAT_VIEW ACCESS FULL| MOST_SERVICES_PER_EMPLOYEE_MV |   100 |  1900 |     2   (0)| 00:00:01 |  Q1,00 | PCWP |            |"
--------------------------------------------------------------------------------------------------------------------------
```

Conclusie:

Als je kunt vergelijken de eerste Explain plan met de tweede, dan kunt je zien dat de cost is veel minder. Dus in de materialized view die ik heb gebruikt bewijst dat de cost is minder.
En toen ik mijn database opnieuw vulde of extra data toevoegde, bleef het resultaat van de materialized view hetzelfde als ervoor.
Een nadeel van een materialized view is dat deze gerefreshed moet worden wanneer de originele data verandert. Als de data vaak verandert, is een materialized view geen goede keuze.