# Milestone 8 Bewijs in PDF

Sam Rotthier - TIFL

## Overzicht vergelijking:

### Tabel info voor partitionering:

Query:

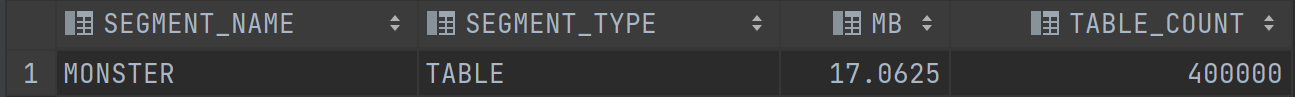
*select segment\_name,segment\_type, sum(bytes/1024/1024) MB ,*

*(select COUNT(\*) FROM monster) as table\_count*

*from dba\_segments*

*where segment\_name= ‘MONSTER’*

*group by segment\_name,segment\_type;*



### **Query:**

*SELECT p.playerid, m.monstername, ROUND(AVG(m."level")) AS "Average\_time\_played"*

*FROM player p*

*JOIN team t ON p.playerid = t.PLAYER\_PLAYERID*

*JOIN monster m ON m.team\_teamid = t.teamid*

*WHERE t.teamid < 25*

*GROUP BY p.playerid, m.monstername*

*ORDER BY p.playerid;*

### **Explain plan**

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

## **NA partitionering:**

### Partitie script + uitleg partitie sleutel

*DROP TABLE monster CASCADE CONSTRAINTS PURGE;*

*CREATE TABLE monster (*

*monsterid INTEGER GENERATED ALWAYS AS IDENTITY*

*CONSTRAINT software\_pk PRIMARY KEY,*

*monstername VARCHAR2(500) NOT NULL,*

*health INTEGER NOT NULL,*

*"level" INTEGER,*

*canevolve CHAR(1),*

*team\_teamid INTEGER NOT NULL*

*)*

*PARTITION BY RANGE (team\_teamid)*

*INTERVAL (50)*

*(*

*partition team\_50 VALUES LESS THAN (50)*

*);*

*ALTER TABLE monster*

*ADD CONSTRAINT monster\_team\_fk FOREIGN KEY (team\_teamid)*

*REFERENCES team ( teamid );*

### Tabel info NA partitionering:

Query:

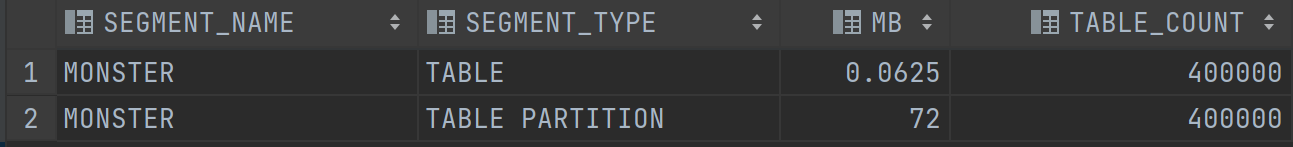
*select segment\_name,segment\_type, sum(bytes/1024/1024) MB ,*

*(select COUNT(\*) FROM monster) as table\_count*

*from dba\_segments*

*where segment\_name= ‘MONSTER’*

*group by segment\_name,segment\_type;*



### Explain plan na partitionering

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

### **Blooper:**

Voor:

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

Na:

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

### **Conclusie:**

In de blooper kunnen we zien dat wanneer je een index op een verkeerd attribuut zet (of een verkeerde filter toepast) dat dit tegenovergesteld werkt en de query langer laat duren.  
Dit komt doordat het programma nu over meerdere files gaat moeten zoeken in plaats van (bij een goede partitionering) minder.

Door de blooper zie je duidelijk hoe de theorie de voor en nadelen van een partitie bespreekt.

Bij een goede parititie (en filtering) gaat het syteem net over minder files moeten zoeken en hierdoor veel tijd winnen.