

# Bewijs Milestone 7

---

Saif Qudaih \_INF201

## Overzicht vergelijking:

- **Bewijs M5 voor S1:**

```
PROJECT> BEGIN
-- PKG_SAMEN_MOTORBIKE.BEWIJS_RANDOM_M5;
  PKG_S1_MOTORBIKES.Bewijs_Milestone_M5_S1(3000, 3000, 3000, 40000);
-- PKG_S2_MOTORBIKES.BEWIJS_MILESTONE_M5_S2(30, 30, 30, 0, 30);
  commit;
END;
[2024-05-18 15:48:00] completed in 46 s 535 ms
[2024-05-18 13:47:15] 5.1 generateRandomCustomer(3000) completed in ,436627 seconds.
[2024-05-18 13:47:15] 5.2 generateRandomMotorBikes(3000) completed in ,528738 seconds.
[2024-05-18 13:47:15] 5.3 generateRandomShop(3000) completed in ,288154 seconds.
[2024-05-18 13:48:00] 5.4 generateRandomRental(40000) completed in 45,129878 seconds.
```

- **Bewijs M7 voor S1**

```
PROJECT> BEGIN
  PKG_S1_MOTORBIKES.BEWIJS_MILESTONE_M7_S1(3000, 3000, 3000, 400000);
end;
[2024-05-18 16:15:36] completed in 11 s 881 ms
[2024-05-18 14:15:25] 4.1 generateRandomCustomer_bulk(3000) completed in ,117613 seconds.
[2024-05-18 14:15:25] 4.2 generatedRandomMotorBike_bulk(3000) completed in ,269094 seconds.
[2024-05-18 14:15:25] 4.3 generateRandomShop_bulk(3000) completed in ,132522 seconds.
[2024-05-18 14:15:36] 4.4 generateRandomRental_bulk(400000) completed in 11,135889 seconds.
```

## Conclusie:

Bij milestone 5 duurt het mij ongeveer 46 seconden om 49.000 rijen te maken voor tabellen van S1. In milestone 7 duurt het mij ongeveer 11 seconden om 490,000 rijen te maken. Mijn bulkmethoden zijn dus veel sneller dan mijn gewone procedures .Omdat je de databank maar 1 keer raadpleegt.

# Bewijs Milestone 7

---

## Moeilijkheid:

Bij het uitvoeren van bulk invoegingen voor records die van elkaar afhankelijk zijn, zoals het invoegen van huren die verwijzen naar bestaande Customer, motorBikeChassisNr en shopid , kan het complex zijn om ervoor te zorgen dat de afhankelijkheden van elke record correct worden opgelost. Dit houdt in dat de gegevensintegriteit wordt gehandhaafd en dat er geen inbreuk wordt gemaakt op de beperkingen van externe sleutels.

## Hoe opgelost:

**1:Order of Operations:** Zorg ervoor dat bovenliggende records vóór onderliggende records worden ingevoegd.

Use collections to gather parent records' IDs.

**2: Controlled Bulk Inserts:** Gebruik FORALL voor het bulksgewijs invoegen van afhankelijke records, waarbij gebruik wordt gemaakt van verzamelde ID's voor refererende sleutelreferenties.

# Bewijs Milestone 7

---

## Overzicht functies:

```
PROCEDURE EMPTY_TABLES_S1 ...;

FUNCTION lookup_Customer...;

-- Lookup functie voor het ophalen van chassisnummer op basis van regsNr
FUNCTION lookup_MotorBike...;

FUNCTION lookup_shop...;

-- Lookup function for Rental based on chassisNr and customerID
FUNCTION lookup_rental...;

PROCEDURE add_Customer...;

PROCEDURE add_MotorBike...;

PROCEDURE add_Shop...;

PROCEDURE add_Rental...;

PROCEDURE BEWIJS_M4_S1 ... ;
```

# Bewijs Milestone 7

---

```
-----***** M5 *****-----  
  
FUNCTION random_list_element ...;  
  
FUNCTION random_color ...;  
  
FUNCTION random_availabilityStatus ...;  
  
FUNCTION random_customer_firstname ...;  
  
function random_customer_lastname ...;  
  
FUNCTION random_shop_name ...;  
  
  
FUNCTION random_shop_address ...;  
  
-----*****-----  
  
PROCEDURE generateRandomCustomer...;  
  
PROCEDURE generateRandomMotorBike...;  
  
PROCEDURE generateRandomShop...;  
  
  
PROCEDURE generateRandomRental...;  
  
  
PROCEDURE Bewijs_Milestone_M5_S1...;
```

```
PROCEDURE add_Customer_bulk...;  
  
PROCEDURE add_motorBike_bulk...;  
  
  
PROCEDURE add_shop_bulk...;  
  
PROCEDURE add_rental_bulk...;  
  
  
PROCEDURE generateRandomCustomer_bulk...;  
  
PROCEDURE generatedRandomMotorBike_bulk...;  
  
PROCEDURE generateRandomShop_bulk...;  
  
  
PROCEDURE generateRandomRental_bulk...;  
  
  
PROCEDURE Bewijs_Milestone_M7_S1...;
```

# Bewijs Milestone 7

---

## S2: Said Khalaf \_INF201

### Overzicht vergelijking tussen M5 en M7:

- **Bewijs M5 voor S2**

```
PROJECT> BEGIN
-- PKG_SAMEN_MOTORBIKE.BEWIJS_RANDOM_M5;
--PKG_S1_MOTORBIKES.Bewijs_Milestone_M5_S1(3000, 3000, 3000, 40000);
PKG_S2_MOTORBIKES.BEWIJS_MILESTONE_M5_S2(30, 30, 30, 60, 30);
commit;
END;
[2024-05-20 15:22:09] completed in 5 m 29 s 957 ms
[2024-05-20 13:16:42] 5.1 generateRandomMotorBikes(30) completed in .048272 seconds.
[2024-05-20 13:16:42] 5.2 generateRandomDepartment(30) completed in .008625 seconds.
[2024-05-20 13:16:42] 5.3 generateRandomMaintenance(30) completed in .027175 seconds.
[2024-05-20 13:16:42] 5.4 generateRandomEmployee(60) completed in .12702 seconds.
Total services inserted: 54000
[2024-05-20 13:22:11] 5.5 generateRandomService(30) completed in 329.595688 seconds.
```

- **Bewijs M7 voor S2**

```
PROJECT> BEGIN
--PKG_S1_MOTORBIKES.BEWIJS_MILESTONE_M7_S1(3000, 3000, 3000, 400000);
PKG_S2_MOTORBIKES.BEWIJS_MILESTONE_M7_S2(30,30, 30, 30, 600);
commit ;
end;
[2024-05-20 20:44:09] completed in 10 s 870 ms
[2024-05-20 18:44:00] 7.1 generateRandomDepartment_bulk(30) completed in .006397 seconds.
[2024-05-20 18:44:00] 7.2 generatedRandomMaintenances_bulk(30) completed in .006337 seconds.
[2024-05-20 18:44:00] 7.2 generatedRandomMaintenances_bulk(30) completed in .028728 seconds.
[2024-05-20 18:44:00] 7.3 generateRandomEmployee_bulk(30) completed in .012511 seconds.
[2024-05-20 18:44:11] 7.4 generateRandomService_bulk(600) completed in 10.637184 seconds.
```

# Bewijs Milestone 7

---

## Conclusie

Bij Milestone 5 duurt het bijna 4m 52s om 54.000 rijen te maken voor mijn eigen tabellen. Integendeel duurt het ongeveer 11s in Milestone 7 om 540.000 te kunnen maken.

Mijn bulk methodes zijn dus veel sneller dan mijn gewone procedures. Omdat de databank maar 1 keer raadpleegt.

## Mogelijkheid

Bijvoorbeeld, wanneer meerdere records worden ingevoegd in de Service- en Employee-tabellen, kan ik moeite hebben met het efficiënt invoegen van grote hoeveelheden data. **Een oplossing** voor dit probleem is het gebruik van bulkoperaties, zoals de FORALL-instructie in PL/SQL, die de prestaties aanzienlijk kan verbeteren door het aantal context wisselingen tussen de PL/SQL- en SQL-engines te minimaliseren.

# Bewijs Milestone 7

---

## Overzicht functies

```
--Lookup functies--  
-- Employee Lookup method for departmentId based on departmentName  
function lookup_DepartmentId...;  
  
-- Employee Lookup method for maintenancId based on maintenanceDescription and maintenanceDate  
function lookup_MaintenancId...;  
  
-- Private opzoekfuncties to look up the employeeId based on firstName and lastName  
function lookup_EmployeeId...;  
  
-- Private function to look up the customerId based on email  
function lookup_CustomerId...;  
  
procedure add_Department...;  
  
-- Private procedure om een rij toe te voegen aan de Maintenance tabel  
procedure add_Maintenance...;  
  
-- Private procedure om een rij toe te voegen aan de Employee tabel  
procedure add_Employee...;  
  
-- Private procedure om een rij toe te voegen aan de Service tabel  
procedure add_Service...;  
  
PROCEDURE BEWIJS_M4_S2 ...;
```

# Bewijs Milestone 7

```
----- M5 -----  
  
FUNCTION random_list_element_DepartmentName ...;  
FUNCTION random_department_location ...;  
FUNCTION random_department_tasks ...;  
FUNCTION random_maintenanceDescription ...;  
FUNCTION random_maintenanceType ...;  
FUNCTION random_employee_firstname ...;  
FUNCTION random_employee_lastname ...;  
FUNCTION random_employee_position ...;  
FUNCTION random_service_Details ...;  
  
----- M5 procedure -----  
  
PROCEDURE generateRandomDepartment...;  
PROCEDURE generateRandomMaintenance...;  
PROCEDURE generateRandomEmployee...;  
PROCEDURE generateRandomService...;  
  
PROCEDURE BEWIJS_MILESTONE_M5_S2...;
```

```
----- M7 -----  
  
PROCEDURE add_Department_bulk...;  
PROCEDURE add_Maintenance_bulk...;  
PROCEDURE add_Employee_bulk...;  
PROCEDURE add_Service_bulk...;  
  
PROCEDURE generateRandomDepartment_bulk...;  
PROCEDURE generateRandomMaintenance_bulk...;  
PROCEDURE generateRandomEmployee_bulk...;  
PROCEDURE generateRandomService_bulk...;  
  
PROCEDURE Bewijs_Milestone_M7_S2...;
```