Assignment 1

Student information

Matriculation number	Student name
6008526	Wolff Andreas Erhard
6008527	Pegza Ariadna

Exercise 1

```
DELIMITER //

DROP PROCEDURE IF EXISTS `generate_driver_statistics`//

CREATE PROCEDURE generate_driver_statistics(IN iYear INT, IN
week_of_the_year INT)
BEGIN
    CREATE TABLE IF NOT EXISTS DriverRecords(
        `Year` INT,
        Week INT,
        DriverId INT,
        NumOfBookings INT DEFAULT 0,
        FOREIGN KEY (DriverId) REFERENCES Driver(idDriver),
                UNIQUE (DriverId, `Year`, Week)
    );

    INSERT INTO DriverRecords (DriverId, `Year`, Week, NumOfBookings)
    SELECT
        b.Driver_idDriver,
        YEAR(b.BookingTime) AS `Year`,
        WEEK(b.BookingTime) AS Week,
        COUNT(b.idBooking) AS NumOfBookings
    FROM
        Booking b
    WHERE
        YEAR(b.BookingTime) = iYear
        AND WEEK(b.BookingTime) = week_of_the_year
    GROUP BY
        b.Driver_idDriver, WEEK(b.BookingTime), YEAR(b.BookingTime)
        ON DUPLICATE KEY UPDATE
```

```
        NumOfBookings = VALUES(NumOfBookings);

END //
DELIMITER ;
```

Exercise 2

```
CREATE EVENT exec_driver_statistics
ON SCHEDULE EVERY 1 WEEK
STARTS '2019-10-28  00:00:00' + INTERVAL 1 WEEK
DO
    CALL generate_driver_statistics(2019, 22);

SHOW EVENTS;
```

Exercise 3

```
-- USING THE GIVEN VIEW (given in the assigment)
CREATE VIEW travel_statistic AS
SELECT c.idCustomer, c.Surname, c.Firstname, SUM(b.DistanceInKM) AS
SumDistance
FROM  Customer c
JOIN Booking b ON b.Customer_idCustomer = c.idCustomer
GROUP BY c.idCustomer;

-- FUNCTION STARTS HERE
DELIMITER //
CREATE FUNCTION customer_class(customer_id INT) RETURNS CHAR(1)
DETERMINISTIC
BEGIN
  DECLARE customer_group CHAR(1);
  SELECT
    CASE
      WHEN ts.SumDistance >= 0.85 * (SELECT MAX(SumDistance) FROM
travel_statistic) THEN 'A'
      WHEN ts.SumDistance >= 0.40 * (SELECT AVG(SumDistance) FROM
travel_statistic) THEN 'B'
      ELSE 'C'
    END
  INTO customer_group
  FROM travel_statistic ts
  WHERE ts.idCustomer = customer_id;

  RETURN customer_group;
```

```
END //
DELIMITER ;

-- TEST QUERIES
SELECT idCustomer, Surname, Firstname, customer_class(idCustomer) AS
CustomerGroup
FROM travel_statistic
WHERE idCustomer = 19;

SELECT idCustomer, Surname, Firstname, customer_class(idCustomer) AS
CustomerGroup FROM travel_statistic;
```