REQUEST 1: CREATE / UPDATE / DELETE QUERIES

- CREATE a new fashion exhibition with the following attributes:
    - id: 51.
    - duration: 60.
    - description: An exclusive exhibition highlighting contemporary and classic haute couture pieces from leading fashion designers around the world.
    - location_address: 151 Elegant Rd Barcelona Spain.
    - title: Barcelona Haute Couture Showcase.

CODE:

```
CREATE (fe:fashion_exhibition {
  id: 51,
  duration: 60,
  local_address: "151 Elegant Rd Barcelona Spain",
  title: "Barcelona Haute Couture Showcase",
  description: "An exclusive exhibition highlighting contemporary and classic haute couture pieces from leading fashion designers around the world."
})
```

- CREATE the relationship between the previous fashion exhibition and the following models. The models who partake to such exhibition are:
    - Model with personal_id = 10;
    - Model with personal_id = 20;
    - Model with personal_id = 30;
    - Model with personal_id = 40;
    - Model with personal_id = 50;

CODE:

```
MATCH (f:fashion_exhibition { id: 51}),
   (m1:fashion_model {personal_id:10}),
   (m2:fashion_model {personal_id:20}),
   (m3:fashion_model {personal_id:30}),
   (m4:fashion_model {personal_id:40}),
   (m5:fashion_model {personal_id:50})

CREATE (m1)-[:partake_to]->(f),
    (m2)-[:partake_to]->(f),
    (m3)-[:partake_to]->(f),
    (m4)-[:partake_to]->(f),
    (m5)-[:partake_to]->(f)
```

- UPDATE the stage name of the designer with personal id equal to 10 to “Noir Éternel”

  CODE:
  ```
  MATCH (fd:fashion_designer {personal_id: 10})
  SET fd.stage_name = "Noir Éternel"
  ```

- UPDATE: the price per square meter of the fabric with id equal to 25 to 135.67.

  CODE:
  ```
  MATCH (fa:fabric {id: 25})
  SET fa.price_per_square_meter = 9.67
  ```

- DELETE the relationship between the fashion model with personal id equal to 34 and the fashion exhibition, whose duration is less than 90 minutes.

  CODE:
  ```
  MATCH (fm:fashion_model)-[r:partake_to]->(fe:fashion_exhibition)
  WHERE fm.personal_id=34 AND fe.duration<90
  DELETE r
  ```

- DELETE all the yellow dresses and the respective relationships found in the whole graphs.

  CODE:
  ```
  MATCH (dr:dress)
  WHERE dr.colour = "Yellow"
  DETACH DELETE dr
  ```

REQUEST 2: A total of 10 queries of the following complexities

3 queries with at least 2 nodes in the MATCH statement and conditions

Query 1: return the fashion models which have size S and height smaller than 170:

CODE:

```
MATCH (fm:fashion_model)
WHERE fm.size= "S" AND fm.height<170
RETURN fm
```

Query 2: Match the fashion designers Evelyn and return her relationships with the fashion models.

CODE:

```
MATCH (fd:fashion_designer)-[m:manages]->(fm)
WHERE fd.name="Evelyn"
RETURN fd, m, fm
```

Query 3: Find all the models that partake to the fashion exhibition Berlin Fashion Spotlight, return the exhibition, relationship and the models.

CODE:

```
MATCH (fe:fashion_exhibition)<-[p:partake_to]-(fm)
WHERE fe.title="Berlin Fashion Spotlight"
RETURN fe, p, fm
```

2 queries with at least 2 nodes in the MATCH statement, conditions and aggregation **without** a WITH statement

Query 4: Count the number of models that can be assign to the generation Z (born after year 2000).

CODE:

```
MATCH (fm:fashion_model)-[:partake_to]-()
WHERE fm.date_of_birth>="2000-01-01"
RETURN COUNT(DISTINCT fm) AS models_generation_z
```

Query 5: Find the number of fashion exhibitions located in Milan and their average duration.

```
MATCH (fe:fashion_exhibition)

WHERE  fe.location_address =~ '.*Milan.*'

RETURN COUNT(*) AS number_Milan_exhibitions,

    AVG(fe.duration) AS Milan_avg_exhibition_duration
```

2 queries with at least 2 nodes in the MATCH statement, conditions and a WITH statement

Query 6: Find the fashion models who partake to at least 7 fashion exhibitions. Return the couple (fashion model, number of participations).

CODE:

```
MATCH (fm:fashion_model)-[:partake_to]->(fe:fashion_exhibition)

WITH fm.personal_id AS model_id, COUNT(*) AS count_exhibitions

WHERE count_exhibitions>=7

RETURN model_id, count_exhibitions
```

Query 7: Find the number of clothes, grouped by size, made up of at least denim fabric.

CODE:

```
MATCH (dr:dress)-[:composed_of]->(fa:fabric)

WHERE fa.name = "Denim"

WITH dr.size AS size, COUNT(*) AS n_dresses_per_size

RETURN size, n_dresses_per_size
```

2 queries with at least 3 nodes in the MATCH statement, conditions and multiple WITH statements

Query 8: For each fashion designer, find the total number of participations of his models in exhibitions.

CODE:

```
MATCH (fm:fashion_model)-[:partake_to]->(fe:fashion_exhibition)

WITH fm, COUNT(fe) AS n_exhibitions_per_model

MATCH (fd:fashion_designer)-[:manages]->(fm)

WITH fd.personal_id AS designer_id,
    SUM(n_exhibitions_per_model) AS total_exhibitions
ORDER BY designer_id
WHERE total_exhibitions > 15
RETURN designer_id, total_exhibitions
```

Query 9: Show the id of the 5 fashion exhibitions with the most number of dresses shown.

CODE:

```
MATCH (fm:fashion_model)-[:wears]->(dr:dress)
WITH fm, COUNT(dr) AS n_dresses_per_model
MATCH (fm)-[:partake_to]->(fe: fashion_exhibition)
WITH fe, sum(n_dresses_per_model) AS n_dresses_per_exhibition
ORDER BY n_dresses_per_exhibition DESC
RETURN fe.id as fashion_exhibition_id, n_dresses_per_exhibition
LIMIT 5
```

1 query with the shortestPath(...) function

Query: Finds the shortest path between the fashion model whose "id" is 6 and the one that has "id" equal to 7. Set an upper bound of 10 steps.

CODE:

```
MATCH path = shortestPath((f1:fashion_model {personal_id:7})-[*..10]-
(f2:fashion_model {personal_id:6}))
RETURN path
```