1. Create 5 new entity with properties (with names: Raj, Sam, Diya, Ali, Arna) where Raj and Sam works for sales and Diya, Ali and Arna works for support team.

```
CREATE (RajC:EMPLOYEE{name:"Raj", age: 38}),
(SamC:EMPLOYEE{name:"Sam", age: 25}),
(AliC:EMPLOYEE{name:"Ali", age: 45}),
(DiyaC:EMPLOYEE{name:"Diya", age: 40}),
(ArnaC:EMPLOYEE{name:"Arna", age: 32}),

(Sa:DEPARTMENT{shortName:"Sales", longName:"Sales Team"}),
(Su:DEPARTMENT{shortName:"Support", longName:"Support Team"}),

(RajC)-[:MEMBER_OF]->(Sa),
(SamC)-[:MEMBER_OF]->(Sa),
(AliC)-[:MEMBER_OF]->(Su),
(DiyaC)-[:MEMBER_OF]->(Su),
(ArnaC)-[:MEMBER_OF]->(Su)
```

2. Add constraint to ensure name property is NOT NULL

```
CREATE CONSTRAINT FOR (e:EMPLOYEE) REQUIRE e.name IS NOT NULL
```

3. Display the names and age of all employees with designation HR

```
match(e:EMPLOYEE)-[:MEMBER_OF]->(d:DEPARTMENT)
where d.shortName='HR'
return e.name,e.age
```

4. Rename designation HR as HRD

```
match(d:DEPARTMENT)
where d.shortName='HR'
set d.shortName='HRD'
```

5. Display names of employees in ascending order that starts from ‘S’

```
match(e:EMPLOYEE)
where e.name STARTS WITH 'S'
return e.name,e.age
order by e.name ASC
```

6. Count number of Employees in each department

```
match(e:EMPLOYEE)-[:MEMBER_OF]->(d:DEPARTMENT)
WITH d.shortName as Dept,count(e) as EmployeeCount
return Dept,EmployeeCount
```

7. Display the details of top 3 employees who are having maximum age.

```
match(e:EMPLOYEE)
return e.name,e.age
order by e.age DESC
limit 3
```

8. Set new property Bonus to all the employees with HR bonus as 20k, Sales: 20 K, Finance: 30k and Support 15K.

```
match(c:COMPANY)-[r:EMPLOYS]->(e:EMPLOYEE)-[:MEMBER_OF]-
>(d:DEPARTMENT)
set r.bonus=
CASE d.shortName
when 'HRD' then 20000
when 'FN' then 30000
END
```

NOTE: I ADDED ONLY FOR HR AND FINANCE, SIMILARLY WE HAVE TO DO IT FOR SALES AND SUPPORT

9. Create two club's music (properties: genre) and fitness (properties: type)

```
create (mC:CLUB{name:'Music',genre:'Music'}),
(fC:CLUB{name:'Fitness',type:'Fitness'})
```

10. Every employee should be part of at least one club. Record date of joining.

```
match(e:EMPLOYEE)
where e.name in ['James Lemmon','Jenny Lane','Josh Simmons','Bob Jones']
match(c:CLUB)
where c.name='Fitness'
create (e)-[:BELONGS_TO]->(c)

 match(e:EMPLOYEE)
 where e.name in ['James Lemmon','Jenny Lane','Josh Simmons','Bob Jones']
 match(c:CLUB)
 where c.name='Fitness'
 create (e)-[:BELONGS_TO]->(c)

 match(e:EMPLOYEE)
 where e.name in ['Raj','Diya','Ali','Arna']
 match(c:CLUB)
 where c.name='Fitness'
 create (e)-[:BELONGS_TO]->(c)
```

11. How many employees are part two clubs

```
match(e:EMPLOYEE)-[:BELONGS_TO]->(c:CLUB)
with e.name as Name,count(c) as ClubCount
where ClubCount=2
return Name
```

12. Display the details of employee who draws salary more than 50K.

```
match(c:COMPANY)-[r:EMPLOYS]->(e:EMPLOYEE)
where r.salary>50000
return e.name,e.age
```