Queries

Query C/1:

The average, the minimum, the maximum, and the standard deviation of the number of applications per trip.

```
select avg(t.applications.size),
       min(t.applications.size), max(t.applications.size),
       sqrt(sum(t.applications.size * t.applications.size)
       / count(a) - (avg(t.applications.size) *
       avg(t.applications.size)))
  from Trip t, Application a;
```

Explicación:

Esta query muestra el mínimo, el máximo y desviación estándar del número de aplicaciones por viaje. Para ello, buscamos el número de aplicaciones que tiene cada viaje para realizar las operaciones. El promedio (average) se calcula utilizando la función avg(), el mínimo utilizando la función min(), el máximo utilizando la función max() y la desviación típica se calcula utilizando la función matemática facilitada en clase.

Results:

```
1 object selected
[1.2, 1, 2, 0.4]
```

Query C/2:

The average, the minimum, the maximum, and the standard deviation of the number of trips managed per manager

```
select avg(m.trips.size), min(m.trips.size),
       max(m.trips.size),
       sqrt(sum(m.trips.size * m.trips.size) /
            count(t) - (avg(m.trips.size) * avg(m.trips.size)))
 from Manager m, Trip t;
```

Explicación:

Esta query muestra el mínimo, el máximo y desviación estándar del número de viajes gestionados por manager. Para ello, buscamos el número de viajes que ha creado cada

managers para realizar las operaciones. El promedio (average) se calcula utilizando la función avg(), el mínimo (minimum) utilizando la función min(), el máximo (maximum) utilizando la función max() y la desviación típica se calcula utilizando la función matemática facilitada en clase.

Results:

```
1 object selected
[5.0, 3, 7, 2]
```

Query C/3:

The average, the minimum, the maximum, and the standard deviation of the price of the trips.

```
select avg(t.price.amount), min(t.price.amount),
       max(t.price.amount),
       sqrt( sum(t.price.amount * t.price.amount) /
           count(t.price.amount) - (avg(t.price.amount) *
           avg(t.price.amount)))
  from Trip t;
```

Explicación:

Esta query muestra el mínimo, el máximo y desviación estándar del precio de los viajes. Para ello, seleccionamos el precio de cada viaje para realizar las operaciones. El promedio (average) se calcula utilizando la función avg(), el mínimo (minimum) utilizando la función min(), el máximo (maximum) utilizando la función max() y la desviación típica se calcula utilizando la función matemática facilitada en clase.

Results:

```
1 object selected
[84.33699999999999, 16.94, 181.5, 52.53142222517873]
```

Query C/4:

The average, the minimum, the maximum, and the standard deviation of the number trips guided per ranger.

```
select avg(r.trips.size), min(r.trips.size), max(r.trips.size),
       sqrt( sum(r.trips.size * r.trips.size) /
           count(t) - (avg(r.trips.size) * avg(r.trips.size)))
 from Ranger r, Trip t;
```

Explicación:
Esta query muestra el mínimo, el máximo y desviación estándar del número de viajes guiados por cada ranger. Para ello, buscamos el número de viajes que ha guiado cada rangers para realizar las operaciones. El promedio (average) se calcula utilizando la función avg(), el mínimo (minimum) utilizando la función min(), el máximo (maximum) utilizando la función max() y la desviación típica se calcula utilizando la función matemática facilitada en clase.

Results:

```
1 object selected
[5.0, 0, 10, 5.0]
```

Query C/5:

The ratio of applications with status “PENDING”.

```
select concat( 100 * (select count(a)
                        from Application a
                       where a.status = 'PENDING')
              / count(b), '%')
  from Application b;
```

Explicación:
Esta query muestra el ratio de aquellas aplicaciones que tengan como status “PENDING”. Para ello, utilizaremos la función concat para expresarlo en forma de porcentaje en donde la primera parte se cuenta utilizando la función count() y utilizando una query aislada a todas aquellas aplicaciones cuyo estado sea “PENDING”, se divide por el total de aplicaciones y se multiplica por 100 y la segunda parte tiene la cadena ‘%’.

Results:

```
1 object selected
"16.6667%"
```

Query C/6:

The ratio of applications with status “DUE”.

```
select concat( 100 * (select count(a)
                        from Application a
                       where a.status = 'DUE')
              / count(b), '%')
  from Application b;
```

Explicación:

Esta query muestra el ratio de aquellas aplicaciones que tengan como status “DUE”. Para ello, utilizaremos la función concat para expresarlo en forma de porcentaje en donde la primera parte se cuenta utilizando la función count() y utilizando una query aislada a todas aquellas aplicaciones cuyo estado sea “DUE”, se divide por el total de aplicaciones y se multiplica por 100 y la segunda parte tiene la cadena ‘%’.

Results:

```
1 object selected
"16.6667%"
```

Query C/7:

The ratio of applications with status “ACCEPTED”.

```
select concat( 100 * (select count(a)
                        from Application a
                       where a.status = 'ACCEPTED')
              / count(b), '%')
  from Application b;
```

Explicación:

Esta query muestra el ratio de aquellas aplicaciones que tengan como status “ACCEPTED”. Para ello, utilizaremos la función concat para expresarlo en forma de porcentaje en donde la primera parte se cuenta utilizando la función count() y utilizando una query aislada a todas aquellas aplicaciones cuyo estado sea “ACCEPTED”, se divide por el total de aplicaciones y se multiplica por 100 y la segunda parte tiene la cadena ‘%’.

Results:

```
1 object selected
"33.3333%"
```

Query C/8:

The ratio of applications with status “CANCELLED”.

```
select concat( 100 * (select count(a)
                            from Application a
                            where a.status = 'CANCELLED')
              / count(b), '%')
       from Application b;
```

Explicación:

Esta query muestra el ratio de aquellas aplicaciones que tengan como status “CANCELLED”. Para ello, utilizaremos la función concat para expresarlo en forma de porcentaje en donde la primera parte se cuenta utilizando la función count() y utilizando una query aislada a todas aquellas aplicaciones cuyo estado sea “CANCELLED”, se divide por el total de aplicaciones y se multiplica por 100 y la segunda parte tiene la cadena ‘%’.

Results:

```
1 object selected
"16.6667%"
```

Query C/9:

The ratio of trips that have been cancelled versus the total number of trips that have been organised.

```
select concat( 100 * (select count(t)
                            from Trip t
                            where t.cancelledReason is not null)
              / count(r), '%')
       from Trip r;
```

Explicación:

Esta query muestra el ratio de aquellas viajes que se hayan cancelado respecto al total organizados. Para ello, utilizaremos la función concat para expresarlo en forma de porcentaje en donde la primera parte se cuenta utilizando la función count() y utilizando una query aislada a todas aquellos viajes que tengan algún razón de cancelación indicando si el atributo cancelledReason de la clase Trip es no nula, se divide por el total de viajes y se multiplica por 100 y la segunda parte tiene la cadena ‘%’.

Results:

```
1 object selected
"10.0000%"
```

Query C/10:

The listing of trips that have got at least 10% more applications than the average, ordered by number of applications.

```
select t
       from Trip t
       where t.applications.size >= 1.1
             * (select avg(r.applications.size)
                       from Trip r)
       order by t.applications.size;
```

Explicación:

Esta query muestra un listado de aquellos viajes que tengan al menos un 10% más de aplicaciones que el promedio, ordenados por el número de aplicaciones. Para ello, seleccionaremos aquellos viajes cuyo número de aplicaciones sea mayor o igual al 10% que el resto, es decir, multiplicando las aplicaciones de cada viaje por 1.1 y, mediante una query aislada, por el promedio de los viajes ordenados por el número de aplicaciones.

Results:

```
2 objects selected

domain.Trip{id=7151, version=0}
   domain.DomainEntity::id: int = 7151
   domain.DomainEntity::version: int = 0
   domain.Trip::ticker: java.lang.String = "170101-AAAA"
   domain.Trip::title: java.lang.String = "Trip 1"
   domain.Trip::description: java.lang.String = "This is trip description"
   domain.Trip::requirements: java.util.Collection = ["Requirement 1"]
   domain.Trip::publicationDate: java.util.Date= <<2017-02-01 12:00:00.0>>
   domain.Trip::startDateTrip: java.util.Date = <<2017-03-01 12:00:00.0>>
   domain.Trip::endDateTrip: java.util.Date = <<2017-04-01 12:00:00.0>>
   domain.Trip::cancelledReason: java.lang.String = null
   domain.Trip::price: domain.Money = domain.Money@4ab9bc66
   domain.Trip::stages: java.util.Collection =
          [domain.Stage{id=7045, version=0},
           domain.Stage{id=7046, version=0},
           domain.Stage{id=7047, version=0},
           domain.Stage{id=7048, version=0},
```

```
            domain.Stage{id=7049, version=0}]
    domain.Trip::category: domain.Category =
            domain.Category{id=7032, version=0}
    domain.Trip::registers:java.util.Collection =
           [domain.Register{id=7202, version=0}]
    domain.Trip::legalText: domain.LegalText =
            domain.LegalText{id=7075, version=0}
    domain.Trip::notes: java.util.Collection =
           [domain.Note{id=7161, version=0},
            domain.Note{id=7170, version=0}]

    domain.Trip::auditRecords: java.util.Collection =
           [domain.AuditRecord{id=7173, version=0}]
    domain.Trip::sponsorships:java.util.Collection =
           [domain.Sponsorship{id=7180, version=0}]
    domain.Trip::survivalClasses:java.util.Collection =
           [domain.SurvivalClass{id=7182, version=0}]
    domain.Trip::manager: domain.Manager =
           domain.Manager{id=7024, version=0}
    domain.Trip::ranger: domain.Ranger =
           domain.Ranger{id=7026, version=1}
    domain.Trip::stories: java.util.Collection =
           [domain.Story{id=7192, version=0}]
    domain.Trip::finders: java.util.Collection =
           [domain.Finder{id=7086, version=0},
            domain.Finder{id=7087, version=0}]
    domain.Trip::applications: java.util.Collection =
           [domain.Application{id=7224, version=0},
            domain.Application{id=7234, version=0}]

domain.Trip{id=7153, version=0}
    domain.DomainEntity::id: int = 7153
    domain.DomainEntity::version: int = 0
    domain.Trip::ticker: java.lang.String = "170101-AAAC"
    domain.Trip::title: java.lang.String = "Trip 3"
    domain.Trip::description: java.lang.String = "This is trip description"
    domain.Trip::requirements: java.util.Collection =
           ["Requirement 1",
            "Requirement 2",
            "Requirement 3"]
    domain.Trip::publicationDate:java.util.Date = <<2017-02-01 12:00:00.0>>
    domain.Trip::startDateTrip: java.util.Date = <<2017-03-01 12:00:00.0>>
    domain.Trip::endDateTrip: java.util.Date = <<2017-04-01 12:00:00.0>>
    domain.Trip::cancelledReason: java.lang.String = null
    domain.Trip::price: domain.Money = domain.Money@3d51572
    domain.Trip::stages: java.util.Collection =
```

```
        [domain.Stage{id=7054, version=0},
         domain.Stage{id=7055, version=0},
         domain.Stage{id=7056, version=0}]
domain.Trip::category: domain.Category =
         domain.Category{id=7034, version=0}
domain.Trip::registers: java.util.Collection =
        [domain.Register{id=7204, version=0}]
domain.Trip::legalText: domain.LegalText =
         domain.LegalText{id=7077, version=0}
domain.Trip::notes: java.util.Collection =
        [domain.Note{id=7163, version=0},
         domain.Note{id=7168, version=0}]
domain.Trip::auditRecords: java.util.Collection =
        [domain.AuditRecord{id=7175, version=0}]
domain.Trip::sponsorships: java.util.Collection = []
domain.Trip::survivalClasses: java.util.Collection =
        [domain.SurvivalClass{id=7184, version=0}]
domain.Trip::manager: domain.Manager =
         domain.Manager{id=7024, version=0}
domain.Trip::ranger: domain.Ranger =
         domain.Ranger{id=7026, version=1}
domain.Trip::stories: java.util.Collection =
         [domain.Story{id=7194, version=0}]
domain.Trip::finders: java.util.Collection =
         [domain.Finder{id=7086, version=0},
          domain.Finder{id=7087, version=0}]
domain.Trip::applications: java.util.Collection =
         [domain.Application{id=7225, version=0},
          domain.Application{id=7235, version=0}]
```

Query C/11:

A table with the number of times that each legal text's been referenced.

```
select l.title, l.trips.size
  from LegalText l;
```

Explicación:

Esta query muestra el número de veces que cada LegalText ha sido referenciado. Para ello, consultamos en la tabla LegalText, mostrado el título (title) de cada LegalText y el número de viajes referenciados (trips.size).

Results:

```
10 objects selected
["title1", 1]
["title2", 1]
["title3", 1]
["title4", 1]
["title5", 1]
["title6", 1]
["title7", 1]
["title8", 1]
["title9", 1]
["title10", 1]
```

Query B/1:

The minimum, the maximum, the average, and the standard deviation of the number of notes per trip.

```
select min(t.notes.size), max(t.notes.size), avg(t.notes.size),
       sqrt(sum(t.notes.size * t.notes.size) / count(n) -
       (avg(t.notes.size) * avg(t.notes.size)))
  from Trip t, Note n;
```

Explicación:

Esta query muestra el mínimo, el máximo y desviación estándar del número de notas por viaje. Para ello, buscamos el número de notas que tiene cada viaje para realizar las operaciones. El promedio (average) se calcula utilizando la función avg(), el mínimo (minimum) utilizando la función min(), el máximo (maximum) utilizando la función max() y la desviación típica se calcula utilizando la función matemática facilitada en clase.

Results:

```
1 object selected
[0, 2, 1.2, 0.8717797887081347]
```

Query B/2:

The minimum, the maximum, the average, and the standard deviation of the number of audit records per trip.

```
select min(t.auditRecords.size), max(t.auditRecords.size),
       avg(t.auditRecords.size),
       sqrt(sum(t.auditRecords.size * t.auditRecords.size) / count(a)
          - (avg(t.auditRecords.size) * avg(t.auditRecords.size)))
 from Trip t, AuditRecord a;
```

Explicación:

Esta query muestra el mínimo, el máximo y desviación estándar del número de auditRecords por viaje. Para ello, buscamos el número de AuditRecord realizados para cada viaje para realizar las operaciones. El promedio (average) se calcula utilizando la función avg(), el mínimo (minimum) utilizando la función min(), el máximo (maximum) utilizando la función max() y la desviación típica se calcula utilizando la función matemática facilitada en clase.

Results:

```
1 object selected
[0, 1, 0.7, 0.458257569495584]
```

Query B/3:

The ratio of trips with an audit record.

```
select concat ( 100 * ( select count(t)
                          from Trip t
                         where t.auditRecords is not empty )
       / count(r), '%')
  from Trip r;
```

Explicación:

Esta query muestra el ratio de aquellos viajes que tengan realizado un auditRecord. Para ello, utilizaremos la función concat para expresarlo en forma de porcentaje en donde la primera parte se cuenta utilizando la función count() y utilizando una query aislada a todas aquellas AuditRecords cuyas referencias no esten vacías, se divide por el total de viajes y se multiplica por 100 y la segunda parte tiene la cadena ‘%’.

Results:

```
1 object selected
"70.0000%"
```

Query B/4:

The ratio of rangers who have registered their curricula.

```
select concat( 100 * ( select count(r)
                         from Ranger r
                        where r.curriculum is not null)
       / count(a), '%')
 from Ranger a;
```

Explicación:

Esta query muestra el ratio de aquellos rangers que tengan registrado sus curriculums. Para ello, utilizaremos la función concat para expresarlo en forma de porcentaje en donde la primera parte se cuenta utilizando la función count() y utilizando una query aislada a todos aquellos curriculums cuya referencia no sea nula, se divide por el total de rangers y se multiplica por 100 y la segunda parte tiene la cadena '%'.

Results:

```
1 object selected
"100.0000%"
```

Query B/5:

The ratio of rangers whose curriculum's been endorsed.

```
select concat ( 100 * ( select count(r)
                         from Ranger r
                        where r.curriculum.endorserRecords is not empty)
       / count(a), '%')
  from Ranger a;
```

Explicación:

Esta query muestra el ratio de aquellos rangers que tengan registrado algún currículum aprobado (endorser). Para ello, utilizaremos la función concat para expresarlo en forma de

porcentaje en donde la primera parte se cuenta utilizando la función count() y utilizando una query aislada a todos aquellos curriculums cuya referencia a endorserRecords no esté vacía, se divide por el total de rangers y se multiplica por 100 y la segunda parte tiene la cadena '%'.

Results:

```
1 object selected
"50.0000%"
```

Query B/6:

The ratio of suspicious managers.

```
select concat( 100 * ( select count(m)
                         from Manager m
                        where m.suspicious is true)
       / count(a), '%')
  from Manager a;
```

Explicación:

Esta query muestra el ratio de aquellos managers que sean sospechosos (suspicious). Para ello, utilizaremos la función concat para expresarlo en forma de porcentaje en donde la primera parte se cuenta utilizando la función count() y utilizando una query aislada a todos aquellos managers cuyo atributo suspicious sea true, se divide por el total de managers y se multiplica por 100 y la segunda parte tiene la cadena '%'.

Results:

```
1 object selected
"0.0000%"
```

Query B/7:

The ratio of suspicious rangers.

```
select concat ( 100 * ( select count(r)
                         from Ranger r
                        where r.suspicious is true)
       /count(a), '%')
  from Ranger a;
```

Explicación:

Esta query muestra el ratio de aquellos rangers que sean sospechosos (suspicious). Para ello, utilizaremos la función concat para expresarlo en forma de porcentaje en donde la primera parte se cuenta utilizando la función count() y utilizando una query aislada a todos aquellos rangers cuyo atributo suspicious sea true, se divide por el total de rangers y se multiplica por 100 y la segunda parte tiene la cadena '%'.

Results:

```
1 object selected
"0.0000%"
```