

ENTREGA 3

JPQL STATEMENTS

Diseño y Pruebas



Grupo 20:

José Ángel Domínguez Espinaco

Daniel Lozano Portillo

José Joaquín Rodríguez Pérez

María Ruiz Gutiérrez

Miguel Ternero Algarín

Laura Vera Recacha

Índice

Introducción	3
Consultas	3

Introducción

En este documento se expondrán las consultas JPQL que se han realizado para el proyecto de la manera en la que se nos indica en el documento de requisitos.

Consultas

- **Query C/1**

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

- Consulta: `select max(t.applicationsFor.size), min(t.applicationsFor.size), avg(t.applicationsFor.size) , stddev(t.applicationsFor.size) from Trip t;`

- Descripción: En esta consulta estamos usando las funciones max, min, avg y stddev para calcular el máximo número de solicitudes que tienen las excursiones, el mínimo número de solicitudes que tienen las excursiones, la media de las solicitudes que tienen las excursiones y su desviación estándar. En la clase Trip hay una colección de solicitudes y desde esa clase cogemos el número de solicitudes que tienen las excursiones y así poder hacer el máximo, el mínimo, la media y la desviación típica. Esta consulta te devuelve un solo objeto con los datos explicados.

- Resultado:

```
> select max(t.applicationsFor.size), min(t.applicationsFor.size), avg(t.applicationsFor.size) ,  
r.size) from Trip t;  
1 object selected  
[3, 0, 1.4, 1.0198]
```

- **Query C/2**

- Título: The average, the minimum, the maximum, and the standard deviation of the number of trips managed per manager.

- Consulta: `select max(m.trips.size), min(m.trips.size), avg(m.trips.size), stddev(m.trips.size) from Manager m;`

- Descripción: En esta consulta estamos usando las funciones max, min, avg y stddev para calcular el máximo número de excursiones que maneja un manager, el mínimo número de excursiones que maneja un manager, la media de las excursiones que maneja un manager y su desviación estándar. En la clase Manager hay una colección de excursiones y desde esa clase cogemos el número de excursiones que tienen cada uno de los managers y así poder hacer el máximo, el mínimo, la media y la desviación estándar. Esta consulta te devuelve un solo objeto con los datos explicados.

- Resultado:

```
> select max(m.trips.size), min(m.trips.size), avg(m.trips.size), stddev(m.trips.size) from Manager m;
1 object selected
[2, 0, 1.0, 0.6325]
```

- **Query C/3**

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

- Consulta: select min(t.price), max(t.price), avg(t.price), stddev(t.price) from Trip t;

- Descripción: En esta consulta estamos usando las funciones max, min, avg y stddev para calcular el precio máximo de las excursiones, el precio mínimo, la media de los precios y su desviación estándar. En la clase Trip tenemos un atributo precio y desde esa clase cogemos el precio que tiene cada una de las excursiones y así poder hacer el máximo, el mínimo, la media y la desviación estándar. Esta consulta te devuelve un solo objeto con los datos explicados.

- Resultado:

```
> select min(t.price), max(t.price), avg(t.price), stddev(t.price) from Trip t;
1 object selected
[12.3, 55.5, 33.56, 15.347390657698135]
```

- **Query C/4**

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

- Consulta: select avg(r.trips.size),min(r.trips.size),max(r.trips.size), stddev(r.trips.size) from Ranger r;

- Descripción: En esta consulta estamos usando las funciones max, min, avg y stddev para calcular el número de trips máximo, mínimo y la desviación estándar que son guiados por los Rangers. En la clase Ranger tenemos un atributo que es la lista de trips que guían y en función del tamaño de dicha lista (puesto que el tamaño depende de la cantidad de trips que está guiando dicho Ranger) realizamos el máximo, el mínimo, la media y la desviación estándar. Esta consulta te devuelve un solo objeto con los datos explicados.

- Resultado:

```
> select avg(r.trips.size),min(r.trips.size),max(r.trips.size), stddev(r.trips.size) from Ranger r;
1 object selected
[1.0, 0, 2, 0.6325]
```

- **Query C/5**

- Título: The ratio of applications with status “PENDING”.

- Consulta: `select (select count(a) from ApplicationFor a where a.status='PENDING') *1.0 / count(a) from ApplicationFor a;`

- Descripción: Esta query consta de dos consultas, la primera cuenta el número de applicationFor que tienen como estado pendiente y se multiplica por 1.0 para que el resultado de la división sea un double, posteriormente a la multiplicación se hace una división entre el número total de ApplicationFor.

- Resultado:

```
> select (select count(a) from ApplicationFor a where a.status='PENDING') *1.0 / count(a) from ApplicationFor a;
1 object selected
0.28571
```

- **Query C/6**

- Título: The ratio of applications with status “DUE”.

- Consulta: `select (select count(a) from ApplicationFor a where a.status='DUE') *1.0 / count(a) from ApplicationFor a;`

- Descripción: Esta query consta de dos consultas, la primera cuenta el número de applicationFor que tienen como estado debido y se multiplica por 1.0 para que el resultado de la división sea un double, posteriormente a la multiplicación se hace una división entre el número total de ApplicationFor.

- Resultado:

```
> select (select count(a) from ApplicationFor a where a.status='DUE') *1.0 / count(a) from ApplicationFor a;
1 object selected
0.14286
```

- **Query C/7**

- Título: The ratio of applications with status “ACCEPTED”.

- Consulta: `select (select count(c) from ApplicationFor c where c.status='ACCEPTED') *1.0 / count(r) from ApplicationFor r;`

- Descripción: Esta query consta de dos consultas, la primera cuenta el número de applicationFor que tienen como estado aceptado y se multiplica por 1.0 para que el resultado de la división sea un double, posteriormente a la multiplicación se hace una división entre el número total de ApplicationFor.

- Resultado:

```
> select (select count(c) from ApplicationFor c where c.status='ACCEPTED') *1.0 / count(r) from ApplicationFor r;
1 object selected
0.28571
```

- **Query C/8**

- Título: The ratio of applications with status “CANCELLED”.

- Consulta: select (select count(c) from ApplicationFor c where c.status='CANCELLED') *1.0 / count(r) from ApplicationFor r;

- Descripción: Esta query consta de dos consultas, la primera cuenta el número de applicationFor que tienen como estado cancelado y se multiplica por 1.0 para que el resultado de la división sea un double, posteriormente a la multiplicación se hace una división entre el número total de ApplicationFor.

- Resultado:

```
> select (select count(c) from ApplicationFor c where c.status='CANCELLED') *1.0 / count(r) from ApplicationFor r;
1 object selected
0.14286
```

- **Query C/9**

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

- Consulta: select (select count(c) from Trip c where c.cancelled=true) * 1.0 / count(r) from Trip r where r.publicationDate is not null;

- Descripción: Esta query consta de dos consultas, la primera consulta lo que hace es contar el número de trips que han sido canceladas y se multiplica por 1.0 para que posteriormente el resultado sea un double, luego de la multiplicación se hace la segunda consulta que lo que hace es contar el número de trips que han sido organizadas.

- Resultado:

```
> select (select count(c) from Trip c where c.cancelled=true) * 1.0 / count(r) from Trip r where r.publicationDate is not null;
1 object selected
0.75
```

- **Query C/10**

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

- Consulta: select t from Trip t where t.applicationsFor.size > 1.1*(select avg(t.applicationsFor.size) from Trip t) order by t.applicationsFor.size;

- Descripción: Esta consulta nos muestra las Trips que tiene al menos un 10% más ApplicationsFor de la media de ApplicationsFor que tiene cada Trip y todo ordenado por el número de ApplicationsFor. Por tanto está compuesta de dos consultas, una que nos devuelve la media de ApplicationsFor por Trip, lo que nos devuelve lo multiplicamos por 1.1 para hacerle el 10% y consultamos la Trips que tienen más applicationsFor del número que nos devuelve la consulta anterior.

- Resultado:

```
> select t from Trip t where t.applicationsFor.size > 1.1*(select avg(t.applicationsFor.size) from Trip t) order by t.applicationsFor.size;
2 objects selected
domain.Trip(id=6150, version=0)
  domain.DomainEntity::id: int = 6150
  domain.DomainEntity::version: int = 0
  domain.Trip::ticker: java.lang.String = "170112-XXXX"
  domain.Trip::title: java.lang.String = "title 1"
  domain.Trip::description: java.lang.String = "description 1"
  domain.Trip::price: double = 12.3
  domain.Trip::requirementsExplorers: java.util.Collection = ["requirementsExplorers 1", "requirementsExplorers 2"]
  domain.Trip::publicationDate: java.util.Date = <<2017-02-02 22:45:00.0>>
  domain.Trip::startDate: java.util.Date = <<2017-02-03 22:00:00.0>>
  domain.Trip::finishDate: java.util.Date = <<2017-02-04 22:00:00.0>>
  domain.Trip::reasonWhy: java.lang.String = "reasonWhy 1"
  domain.Trip::cancelled: boolean = true
  domain.Trip::ranger: domain.Ranger = domain.Ranger(id=6083, version=1)
  domain.Trip::manager: domain.Manager = domain.Manager(id=6112, version=0)
  domain.Trip::classes: java.util.Collection = [domain.SurvivalClass(id=6175, version=0)]
  domain.Trip::stories: java.util.Collection = [domain.Story(id=6180, version=0)]
  domain.Trip::applicationsFor: java.util.Collection = [domain.ApplicationFor(id=6185, version=0), domain.ApplicationFor(id=6191, version=0)]
  domain.Trip::auditRecords: java.util.Collection = [domain.AuditRecord(id=6165, version=0)]
  domain.Trip::notes: java.util.Collection = [domain.Note(id=6160, version=0)]
  domain.Trip::categories: java.util.Collection = [domain.Category(id=6106, version=0)]
  domain.Trip::sponsorships: java.util.Collection = [domain.Sponsorship(id=6170, version=0)]
  domain.Trip::stages: java.util.Collection = [domain.Stage(id=6151, version=0)]
  domain.Trip::legalText: domain.LegalText = domain.LegalText(id=6096, version=0)
  domain.Trip::tags: java.util.Collection = [domain.Tag(id=6076, version=0)]
domain.Trip(id=6158, version=0)
  domain.DomainEntity::id: int = 6158
  domain.DomainEntity::version: int = 0
  domain.Trip::ticker: java.lang.String = "150112-DDDD"
  domain.Trip::title: java.lang.String = "title 5"
  domain.Trip::description: java.lang.String = "description 5"
  domain.Trip::price: double = 55.5
  domain.Trip::requirementsExplorers: java.util.Collection = ["requirementsExplorers 8", "requirementsExplorers 9"]
  domain.Trip::publicationDate: java.util.Date = null
  domain.Trip::startDate: java.util.Date = <<2014-02-21 22:00:00.0>>
  domain.Trip::finishDate: java.util.Date = <<2015-08-04 22:00:00.0>>
  domain.Trip::reasonWhy: java.lang.String = "reasonWhy 5"
  domain.Trip::cancelled: boolean = false
  domain.Trip::ranger: domain.Ranger = domain.Ranger(id=6083, version=1)
  domain.Trip::manager: domain.Manager = domain.Manager(id=6116, version=0)
  domain.Trip::classes: java.util.Collection = [domain.SurvivalClass(id=6179, version=0)]
  domain.Trip::stories: java.util.Collection = [domain.Story(id=6184, version=0)]
  domain.Trip::applicationsFor: java.util.Collection = [domain.ApplicationFor(id=6188, version=0), domain.ApplicationFor(id=6189, version=0), domain.ApplicationFor(id=6190, version=0)]
  domain.Trip::auditRecords: java.util.Collection = [domain.AuditRecord(id=6169, version=0)]
  domain.Trip::notes: java.util.Collection = []
  domain.Trip::categories: java.util.Collection = [domain.Category(id=6110, version=0)]
  domain.Trip::sponsorships: java.util.Collection = [domain.Sponsorship(id=6174, version=0)]
  domain.Trip::stages: java.util.Collection = [domain.Stage(id=6159, version=0)]
  domain.Trip::legalText: domain.LegalText = domain.LegalText(id=6096, version=0)
  domain.Trip::tags: java.util.Collection = [domain.Tag(id=6080, version=0), domain.Tag(id=6076, version=0)]
```

- **Query C/11**

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

- Consulta: select count(t) from Trip t group by t.legalText;

- Descripción: Esta query nos devuelve el número de veces que un LegalText ha sido referenciado. Para ello contamos el número de Trip que tienen ese LegalText, esto lo hacemos agrupando las Trip por los legalTexts.

- Resultado:

```
> select count(t) from Trip t group by t.legalText;
4 objects selected
2
1
1
1
```

- **Query B/1**

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

- Consulta: `select min(t.notes.size), max(t.notes.size), avg(t.notes.size), stddev(t.notes.size) from Trip t;`

- Descripción: En esta consulta devolvemos el mínimo, máximo, media y desviación típica de las notas que tienen las Trips. Para ello usamos las funciones definidas en jpql (min, max, avg y stddev). Esas operaciones se las aplicamos al número de notes que tiene cada Trip.

- Resultado:

```
> select min(t.notes.size), max(t.notes.size), avg(t.notes.size), stddev(t.notes.size) from Trip t;
1 object selected
[0, 2, 1.0, 0.6325]
```

- **Query B/2**

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

- Consulta: `select min(t.auditRecords.size),max(t.auditRecords.size),avg(t.auditRecords.size), stddev(t.auditRecords.size) from Trip t;`

- Descripción: La consulta selecciona el número mínimo, máximo, la media y la desviación estándar de auditRecords que tiene un Trip de entre todos los Trips.

- Resultado:

```
> select min(t.auditRecords.size),max(t.auditRecords.size),avg(t.auditRecords.size), stddev(t.auditRecords.size) from Trip t;
1 object selected
[0, 2, 1.0, 0.6325]
```


- **Query B/3**

- Título: The ratio of trips with an audit record.

- Consulta: `select count(r)/(select count(t) from Trip t)*1.0 from Trip r where r.auditRecords.size>0;`

- Descripción: La consulta selecciona el número de Trips que tengan auditRecords y los divide entre el número total de trips.

- Resultado:

```
> select count(r)/(select count(t) from Trip t)*1.0 from Trip r where r.auditRecords.size>0;
1 object selected
0.8
```

- **Query B/4**

- Título: The ratio of rangers who have registered their curricula.

- Consulta: `select count(r)/(select count(s) from Ranger s)*1.0 from Ranger r where r.curricula!=null;`

- Descripción: La consulta selecciona el número de Rangers que tengan curricula y los divide entre el número total de Rangers.

- Resultado:

```
> select count(r)/(select count(s) from Ranger s)*1.0 from Ranger r where r.curricula!=null;
1 object selected
0.6
```

- **Query B/5**

- Título: The ratio of rangers whose curriculum's been endorsed.

- Consulta: `select (select count(c) from Ranger r join r.curricula c where c.endorserRecords.size>0)*1.0/count(c) from Curricula c;`

- Descripción: Esta consulta está compuesta de dos consultas la primera cuenta todas los ranger que tengan en su curricula algún EndorseRecord y en la otra se calcula el número total de Curriculas que hay que a su vez equivale al número de Ranger. Después éstos se dividen y así se calcula el ratio. Se ha multiplicado el dividendo * 1.0 para que el resultado sea un double.

- Resultado:

```
> select (select count(c) from Ranger r join r.curricula c where c.endorserRecords.size>0)*1.0/count(c) from Curricula c
;
1 object selected
0.6
```

- **Query B/6**

- Título: The ratio of suspicious managers.

- Consulta: `select (select count(m1) from Manager m1 where m1.suspicious=false)*1.0/count(m) from Manager m ;`

- Descripción: Esta consulta está compuesta por dos consultas, la que hace referencia al objeto m te cuenta el número total de Manager y la otra que hace referencia al objeto m1 te cuenta el número de Manager que son sospechosos. Después se calcula el ratio de esta forma: $m1 * 1.0 / m$. Se multiplica por 1.0 para que el dividendo sea double y al hacer la división también lo sea.

- Resultado:

- ```
> select (select count(m1) from Manager m1 where m1.suspicious=false)*1.0/count(m) from Manager m ;
1 object selected
0.6
```

- **Query B/7**

- Título: The ratio of suspicious rangers.

- Consulta: `select (select count(r1) from Ranger r1 where r1.suspicious=false)*1.0/count(r) from Ranger r ;`

- Descripción: Esta consulta está compuesta por dos consultas, la que hace referencia al objeto r te cuenta el número total de Ranger y la otra que hace referencia al objeto r1 te cuenta el número de Ranger que son sospechosos. Después se calcula el ratio de esta forma  $r1 * 1.0 / r$ . Se multiplica por 1.0 para que el dividendo sea double y al hacer la división también lo sea.

- Resultado:

- ```
> select (select count(r1) from Ranger r1 where r1.suspicious=false)*1.0/count(r) from Ranger r ;
1 object selected
0.4
```