

### Query C/1

The average, the minimum, the maximum, and the standard deviation of the number of fix-up tasks per user.

```
select max(c.fixUpTasks.size), min(c.fixUpTasks.size), avg(c.fixUpTasks.size),  
sqrt(sum(c.fixUpTasks.size * c.fixUpTasks.size) / count(c.fixUpTasks.size) -  
(avg(c.fixUpTasks.size) * avg(c.fixUpTasks.size))) from Customer c;
```

Se ha calculado las operaciones mencionadas en el enunciado para las chapuzas que tiene asignada cada usuario del sistema.

Resultado: 1 object selected [2, 1, 1.5, 0.5]

### Query C/2:

The average, the minimum, the maximum, and the standard deviation of the number of applications per fix-up task.

```
select max(f.applications.size), min(f.applications.size), avg(f.applications.size),  
sqrt(sum(f.applications.size * f.applications.size) / count(f.applications.size) -  
(avg(f.applications.size) * avg(f.applications.size))) from FixUpTask f;
```

Se ha calculado las operaciones mencionadas en el enunciado para las solicitudes que tiene asignada una chapuza.

Resultado: 1 object selected [2, 0, 1.3333, 0.9428090416999145]

### Query C/3

The average, the minimum, the maximum, and the standard deviation of the maximum price of the fix-up tasks. Devuelve el promedio, el mínimo, el máximo y la desviación estándar del Precio máximo de las tareas de reparación:

```
select avg(f.maxPrice), min(f.maxPrice), max(f.maxPrice),  
sqrt(sum(f.maxPrice*f.maxPrice) / count(f.maxPrice) - (avg(f.maxPrice)  
* avg(f.maxPrice))) from FixUpTask f;
```

El primer valor representa la media, el segundo el mínimo, tercero el máximo y por último la desviación típica.

Resultado: [1382.6666666666667, Quantity: 30.0, Quantity: 4068.0, 1898.8349644511559]

### **Query C/4**

The average, the minimum, the maximum, and the standard deviation of the price offered in the applications. Devuelve el promedio, el mínimo, el máximo y la desviación estándar del Precio ofrecido en las aplicaciones.

```
select avg(a.offeredPrice), max(a.offeredPrice), min(a.offeredPrice),  
sqrt(sum(a.offeredPrice*a.offeredPrice)/count(a.offeredPrice)  
-(avg(a.offeredPrice)*avg(a.offeredPrice))) from Application a group by 'a';
```

El primer valor representa la media, el segundo el mínimo, tercero el máximo y por último la desviación típica.

Resultado: [60.75, Quantity: 80.0, Quantity: 40.0, 19.279198634798075]

### **Query C/5:**

The ratio of pending applications.

```
select (sum(case when a.status='PENDING' then 1.0 else 0 end)/count(*)) from  
Application a;
```

A partir de las solicitudes, en el caso de que una solicitud se encuentre en estado 'pendiente', se suma uno. En el caso contrario cero. Una vez hecho esto se divide entre el número total de solicitudes.

Resultado: 1 object select 0.75

### **Query C/6:**

The ratio of accepted applications.

```
select (sum(case when a.status='ACCEPTED' then 1.0 else 0 end)/count(*)) from  
Application a;
```

A partir de las solicitudes, en el caso de que una solicitud se encuentre en estado 'aceptada', se suma uno. En el caso contrario cero. Una vez hecho esto se divide entre el número total de solicitudes.

Resultado: 1 object select 0.0

### **Query C/7:**

The ratio of rejected applications.

```
select (sum(case when a.status='REJECTED' then 1.0 else 0 end)/count(*)) from  
Application a;
```

A partir de las solicitudes, en el caso de que una solicitud se encuentre en estado 'rechazado', se suma uno. En el caso contrario cero. Una vez hecho esto se divide entre el número total de solicitudes.

Resultado: 1 object select 0.25

### **Query C/8:**

The ratio of pending applications that cannot change its status because their time period's elapsed.

```
select count(a)/(select count(a) from Application a where a.fixUpTask.endMoment <  
CURRENT_DATE and a.status='PENDING')*1.0 from Application a where  
a.status='PENDING';
```

En el numerador se calcula el total de solicitudes cuyo estado es pendiente. En el denominador se obtiene de las solicitudes con estado pendiente, aquellas que la fecha de fin de su chapuza sea anterior a la fecha actual. Obteniendo como resultado la proporción.

Resultado: 1 object selected 1.5

### **Query C/9**

The listing of customers who have published at least 10% more fix-up tasks than the average, ordered by number of applications.

```
select c, c.fixUpTasks.size from Customer c, FixUpTask f  
where c.fixUpTasks.size>(select avg(c.fixUpTasks.size)*1.1 from Customer c)  
order by count(f.applications.size) desc;
```

Representa el cliente y el nº de fix up task que tiene dicho cliente. Devuelve la lista de clientes que han publicado al menos un 10% más de tareas de reparación que el promedio, ordenado por número de solicitudes.

Resultado: 1 object selected [domain.Customer{id=1817, version=0}, 2]

## Query C/10

The listing of handy workers who have got accepted at least 10% more applications than the average, ordered by number of applications.

```
select h from HandyWorker h where h.applications.size/ (select  
avg(h1.applications.size) from HandyWorker h1)>=1.1 order by h.applications.size;
```

Seleccionamos los manitas que tengan más solicitudes aceptadas que el 10% de la media ordenados por número de solicitudes.

### Resultado:

1 object selected

```
domain.HandyWorker{id=1936, version=0}  
  domain.DomainEntity::id: int = 1936  
  domain.DomainEntity::version: int = 0  
  domain.Actor::name: java.lang.String = "Rosa"  
  domain.Actor::middleName: java.lang.String = null  
  domain.Actor::surname: java.lang.String = "Santacruz Alejo"  
  domain.Actor::photo: java.lang.String = null  
  domain.Actor::email: java.lang.String = "rossanale@mail.com"  
  domain.Actor::phoneNumber: java.lang.String = null  
  domain.Actor::address: java.lang.String = null  
  domain.Actor::isSuspicious: boolean = false  
  domain.Actor::socialProfiles: java.util.Collection = []  
  domain.Actor::messageBoxes: java.util.Collection =  
[domain.MessageBox{id=1884, version=0}, domain.MessageBox{id=1885, v  
ersion=0}, domain.MessageBox{id=1886, version=0}, domain.MessageBox{id=1887,  
version=0}]  
  domain.Actor::userAccount: security.UserAccount =  
security.UserAccount{id=1892, version=0}  
  domain.HandyWorker::score: double = 0.0  
  domain.HandyWorker::make: java.lang.String = "Handy Worker 1 S.A."  
  domain.HandyWorker::endorsements: java.util.Collection =  
[domain.Endorsement{id=1940, version=0}]  
  domain.HandyWorker::tutorial: java.util.Collection = []  
  domain.HandyWorker::applications: java.util.Collection =  
[domain.Application{id=1951, version=0}, domain.Application{id  
=1952, version=0}, domain.Application{id=1953, version=0}]  
  domain.HandyWorker::finder: domain.Finder = domain.Finder{id=1928,  
version=0}  
  domain.HandyWorker::curriculum: domain.Curriculum =  
domain.Curriculum{id=1844, version=0}
```

## **Query B/1**

Display the minimum, the maximum, the average, and the standard deviation of the number of complaints per fix-up task.

```
select max(f.complaints.size), min(f.complaints.size), avg(f.complaints.size),  
sqrt(sum(f.complaints.size * f.complaints.size) / count(f.complaints.size) -  
(avg(f.complaints.size) * avg(f.complaints.size))) from FixUpTask f;
```

Recorre todas las fixUpTaks mirando el tamaño de la lista de complaints de cada una y devuelve los valores de número (tamaño) máximo de complaints en una fixUpTasks, mínimo, medio y desviación típica.

Resultado: 1 object selected [2, 0, 1.0, 0.8164965805194777]

## **Query B/2**

Display the minimum, the maximum, the average, and the standard deviation of the number of notes per referee report.

```
select max(r.notes.size), min(r.notes.size), avg(r.notes.size), sqrt(sum(r.notes.size *  
r.notes.size) / count(r.notes.size) - (avg(r.notes.size) * avg(r.notes.size))) from Report r;
```

Recorre todas los Reports mirando el tamaño de la lista de notas de cada una y devuelve los valores de número (tamaño) máximo de notas en un Reports, mínimo, medio y desviación típica.

Resultado: 1 object selected [1, 0, 0.3333, 0.47140452067318056]

## **Query B/3**

Display the ratio of fix-up tasks with a complaint.

```
select count(f) / (select count(f) from FixUpTask f where f.complaints is not empty) * 1.0  
from FixUpTask f;
```

Devuelve por un lado el número de fixUpTask con complaint y por otro el de fixUpTasks total.

Resultado: 1 object selected 1.5

## Query B/4

The top-three customers in terms of complaints.

*select c from Customer c order by c.complaints.size desc;*

Nota: En los servicios limitaremos a los 3 primeros elementos de la lista que devuelve la query usando paginación.

Mira el número de complaints de cada Customer y las devuelve ordenadas de forma descendente.

### Resultado:

2 objects selected

```
domain.Customer{id=1819, version=0}
  domain.DomainEntity::id: int = 1819
  domain.DomainEntity::version: int = 0
  domain.Actor::name: java.lang.String = "Lorena"
  domain.Actor::middleName: java.lang.String = "María"
  domain.Actor::surname: java.lang.String = "Revuelta Berbel"
  domain.Actor::photo: java.lang.String =
"https://media.aweita.larepublica.pe/678x508/aweita/imagen/2018/02/08/noticia-p
ersona-positiva.png"
  domain.Actor::email: java.lang.String = "lorrevber@mail.com"
  domain.Actor::phoneNumber: java.lang.String = "+34689945823"
  domain.Actor::address: java.lang.String = "C/ Villa Crepúsculo, 1 (Dos
Hermanas, Sevilla)"
  domain.Actor::isSuspicious: boolean = true
  domain.Actor::socialProfiles: java.util.Collection =
[domain.SocialProfile{id=1895, version=0}, domain.SocialProfile{id
=1896, version=0}]
  domain.Actor::messageBoxes: java.util.Collection =
[domain.MessageBox{id=1864, version=0}, domain.MessageBox{id=1865, v
ersion=0}, domain.MessageBox{id=1866, version=0}, domain.MessageBox{id=1867,
version=0}]
  domain.Actor::userAccount: security.UserAccount =
security.UserAccount{id=1820, version=0}
  domain.Customer::score: double = 0.0
  domain.Customer::endorsements: java.util.Collection =
[domain.Endorsement{id=1940, version=0}]
  domain.Customer::fixUpTasks: java.util.Collection =
[domain.FixUpTask{id=1943, version=0}]
  domain.Customer::complaints: java.util.Collection =
[domain.Complaint{id=1947, version=0}]
domain.Customer{id=1817, version=0}
  domain.DomainEntity::id: int = 1817
```

```

domain.DomainEntity::version: int = 0
domain.Actor::name: java.lang.String = "Francisco"
domain.Actor::middleName: java.lang.String = null
domain.Actor::surname: java.lang.String = "Berrocal Ribas"
domain.Actor::photo: java.lang.String = null
domain.Actor::email: java.lang.String = "frabelrib@mail.com"
domain.Actor::phoneNumber: java.lang.String = null
domain.Actor::address: java.lang.String = null
domain.Actor::isSuspicious: boolean = false
domain.Actor::socialProfiles: java.util.Collection = []
domain.Actor::messageBoxes: java.util.Collection =
[domain.MessageBox{id=1860, version=0}, domain.MessageBox{id=1861, v
ersion=0}, domain.MessageBox{id=1862, version=0}, domain.MessageBox{id=1863,
version=0}]
domain.Actor::userAccount: security.UserAccount =
security.UserAccount{id=1818, version=0}
domain.Customer::score: double = 0.0
domain.Customer::endorsements: java.util.Collection =
[domain.Endorsement{id=1938, version=0}, domain.Endorsement{id=19
39, version=0}]
domain.Customer::fixUpTasks: java.util.Collection =
[domain.FixUpTask{id=1941, version=0}, domain.FixUpTask{id=1942, ve
rsion=0}]
domain.Customer::complaints: java.util.Collection = []

```

## **Query B/5**

The top-three handy workers in terms of complaints.

```
select h from HandyWorker h join h.applications a join a.fixUpTask f group by h order
by count(f.complaints.size) desc;
```

Nota: En los servicios limitaremos a los 3 primeros elementos de la lista que devuelve la query usando paginación.

Desde la clase HandyWorker va hasta la clase FixUpTask mediante join y agrupa por cada handy worker el número total de complaints que tiene relacionadas en el conjunto de fixUpTasks.

### **Resultado:**

```

2 objects selected
domain.HandyWorker{id=1912, version=0}
  domain.DomainEntity::id: int = 1912
  domain.DomainEntity::version: int = 0
  domain.Actor::name: java.lang.String = "Rosa"

```

```
domain.Actor::middleName: java.lang.String = null
domain.Actor::surname: java.lang.String = "Santacruz Alejo"
domain.Actor::photo: java.lang.String = null
domain.Actor::email: java.lang.String = "rossanale@mail.com"
domain.Actor::phoneNumber: java.lang.String = null
domain.Actor::address: java.lang.String = null
domain.Actor::isSuspicious: boolean = false
domain.Actor::socialProfiles: java.util.Collection = []
domain.Actor::messageBoxes: java.util.Collection =
[domain.MessageBox{id=1862, version=0}, domain.MessageBox{id=1863, version=0},
domain.MessageBox{id=1864, version=0}, domain.MessageBox{id=1865, version=0}]
domain.Actor::userAccount: security.UserAccount =
security.UserAccount{id=1870, version=0}
domain.HandyWorker::score: double = 0.0
domain.HandyWorker::make: java.lang.String = "Handy Worker 1 S.A."
domain.HandyWorker::endorsements: java.util.Collection =
[domain.Endorsement{id=1916, version=0}]
domain.HandyWorker::tutorial: java.util.Collection = []
domain.HandyWorker::applications: java.util.Collection =
[domain.Application{id=1927, version=0}, domain.Application{id=1928, version=0},
domain.Application{id=1929, version=0}]
domain.HandyWorker::finder: domain.Finder = domain.Finder{id=1904,
version=0}
domain.HandyWorker::curriculum: domain.Curriculum =
domain.Curriculum{id=1822, version=0}
domain.HandyWorker{id=1913, version=0}
domain.DomainEntity::id: int = 1913
domain.DomainEntity::version: int = 0
domain.Actor::name: java.lang.String = "Esteban"
domain.Actor::middleName: java.lang.String = "Coralino"
domain.Actor::surname: java.lang.String = "Riesco Motos"
domain.Actor::photo: java.lang.String =
"http://s1.1zoom.me/b5050/425/GTA_5_Men_Trevor_Philips_Formal_shirt_518191_
1920x1080.jpg"
domain.Actor::email: java.lang.String = "estriemot@mail.com"
domain.Actor::phoneNumber: java.lang.String = "+34653384191"
domain.Actor::address: java.lang.String = "C/ San Fransokyo, 83 (Espartinas,
Sevilla)"
domain.Actor::isSuspicious: boolean = true
domain.Actor::socialProfiles: java.util.Collection =
[domain.SocialProfile{id=1878, version=0}]
domain.Actor::messageBoxes: java.util.Collection =
[domain.MessageBox{id=1866, version=0}, domain.MessageBox{id=1867, version=0},
domain.MessageBox{id=1868, version=0}, domain.MessageBox{id=1869, version=0}]
domain.Actor::userAccount: security.UserAccount =
security.UserAccount{id=1871, version=0}
domain.HandyWorker::score: double = 0.0
```



```
domain.HandyWorker::make: java.lang.String = "Handy Worker 2 S.A."
domain.HandyWorker::endorsements: java.util.Collection =
[domain.Endorsement{id=1914, version=0}, domain.Endorsement{id=1915, version=0}]
domain.HandyWorker::tutorial: java.util.Collection = [domain.Tutorial{id=1826,
version=0}, domain.Tutorial{id=1828, version=0}]
domain.HandyWorker::applications: java.util.Collection =
[domain.Application{id=1926, version=0}]
domain.HandyWorker::finder: domain.Finder = domain.Finder{id=1905,
version=0}
domain.HandyWorker::curriculum: domain.Curriculum =
domain.Curriculum{id=1812, version=0}
```