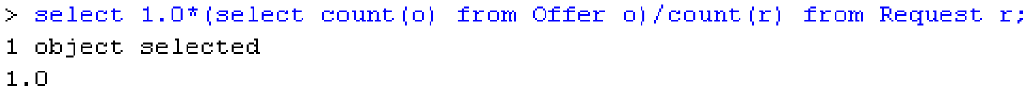
C.1 - Ratio of offers versus requests.

Calculo el número total de ofertas partido del número total de peticiones, todo ello multiplicado por 1.0 para obtener un valor decimal.

select 1.0(select count(o) from Offer o)/count(r) from Request r;*

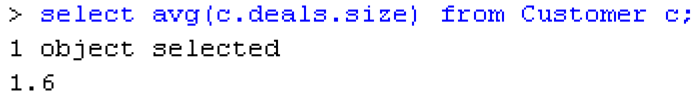


```
> select 1.0*(select count(o) from Offer o)/count(r) from Request r;
1 object selected
1.0
```

C.2 - Average number of offers and request per customer

Realizo la operación media de la cantidad de peticiones y ofertas por cliente.

select avg(c.deals.size) from Customer c;

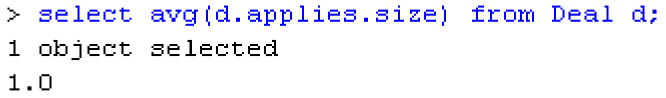


```
> select avg(c.deals.size) from Customer c;
1 object selected
1.6
```

C.3 - Average number of applications per offer or request.

Realizo la operación media de la cantidad de aplicaciones por oferta/petición.

select avg(d.applies.size) from Deal d;

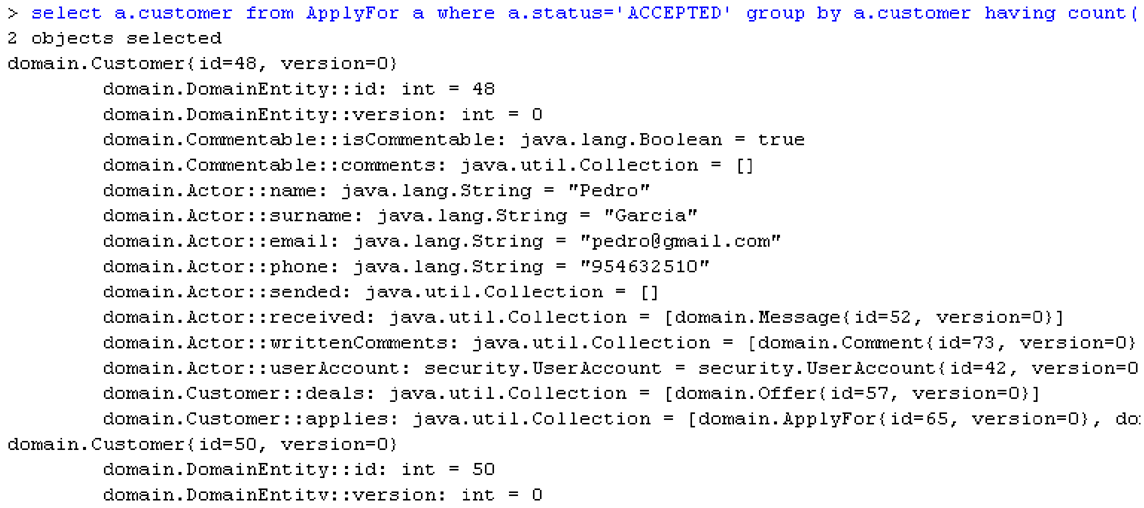


```
> select avg(d.applies.size) from Deal d;
1 object selected
1.0
```

C.4 - The customer who has more applications accepted

Realizo una búsqueda de clientes por aplicación aceptadas, y la comparo con el número de clientes que tienen aplicaciones aceptadas. Cogiendo al que más aplicaciones tenga aceptadas.

select a.customer from ApplyFor a where a.status='ACCEPTED' group by a.customer having count(a)>=all(select count(a2) from ApplyFor a2 where a2.status='ACCEPTED' group by a2.customer);

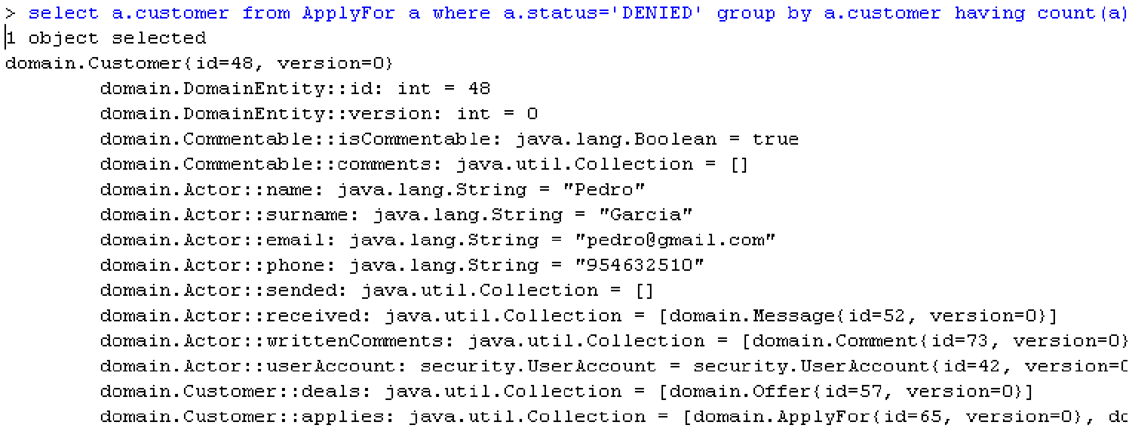


```
> select a.customer from ApplyFor a where a.status='ACCEPTED' group by a.customer having count(
2 objects selected
domain.Customer(id=48, version=0)
        domain.DomainEntity::id: int = 48
        domain.DomainEntity::version: int = 0
        domain.Commentable::isCommentable: java.lang.Boolean = true
        domain.Commentable::comments: java.util.Collection = []
        domain.Actor::name: java.lang.String = "Pedro"
        domain.Actor::surname: java.lang.String = "Garcia"
        domain.Actor::email: java.lang.String = "pedro@gmail.com"
        domain.Actor::phone: java.lang.String = "954632510"
        domain.Actor::sended: java.util.Collection = []
        domain.Actor::received: java.util.Collection = [domain.Message(id=52, version=0)]
        domain.Actor::writtenComments: java.util.Collection = [domain.Comment(id=73, version=0)
        domain.Actor::userAccount: security.UserAccount = security.UserAccount(id=42, version=0
        domain.Customer::deals: java.util.Collection = [domain.Offer(id=57, version=0)]
        domain.Customer::applies: java.util.Collection = [domain.ApplyFor(id=65, version=0), do
domain.Customer(id=50, version=0)
        domain.DomainEntity::id: int = 50
        domain.DomainEntity::version: int = 0
```

C.5 - The customer who has more applications denied

Realizo una búsqueda de clientes por aplicación denegadas, y la comparo con el número de clientes que tienen aplicaciones denegadas. Cogiendo al que más aplicaciones tenga denegadas.

select a.customer from ApplyFor a where a.status='DENIED' group by a.customer having count(a)>=all(select count(a2) from ApplyFor a2 where a2.status='DENIED' group by a2.customer);

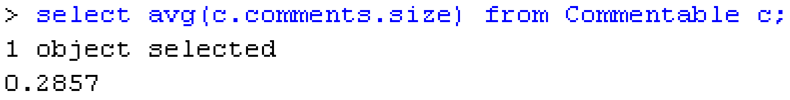


```
> select a.customer from ApplyFor a where a.status='DENIED' group by a.customer having count(a)
1 object selected
domain.Customer{id=48, version=0}
        domain.DomainEntity::id: int = 48
        domain.DomainEntity::version: int = 0
        domain.Commentable::isCommentable: java.lang.Boolean = true
        domain.Commentable::comments: java.util.Collection = []
        domain.Actor::name: java.lang.String = "Pedro"
        domain.Actor::surname: java.lang.String = "Garcia"
        domain.Actor::email: java.lang.String = "pedro@gmail.com"
        domain.Actor::phone: java.lang.String = "954632510"
        domain.Actor::sended: java.util.Collection = []
        domain.Actor::received: java.util.Collection = [domain.Message{id=52, version=0}]
        domain.Actor::writtenComments: java.util.Collection = [domain.Comment{id=73, version=0}
        domain.Actor::userAccount: security.UserAccount = security.UserAccount{id=42, version=C
        domain.Customer::deals: java.util.Collection = [domain.Offer{id=57, version=0}]
        domain.Customer::applies: java.util.Collection = [domain.ApplyFor{id=65, version=0}, dc
```

B.1 - Average number of comments per actor, offer, or request

Realizo la operación media del número de comentarios de actores, ofertas y peticiones.

select avg(c.comments.size) from Commentable c;

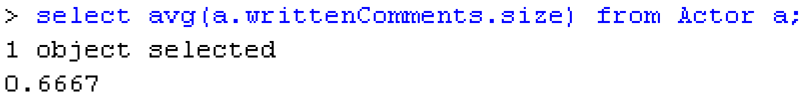


```
> select avg(c.comments.size) from Commentable c;
1 object selected
0.2857
```

B.2 - Average number of comments posted by administrators and customers

Realizo la operación media del número de comentarios escritos por administradores y clientes.

select avg(a.writtenComments.size) from Actor a;

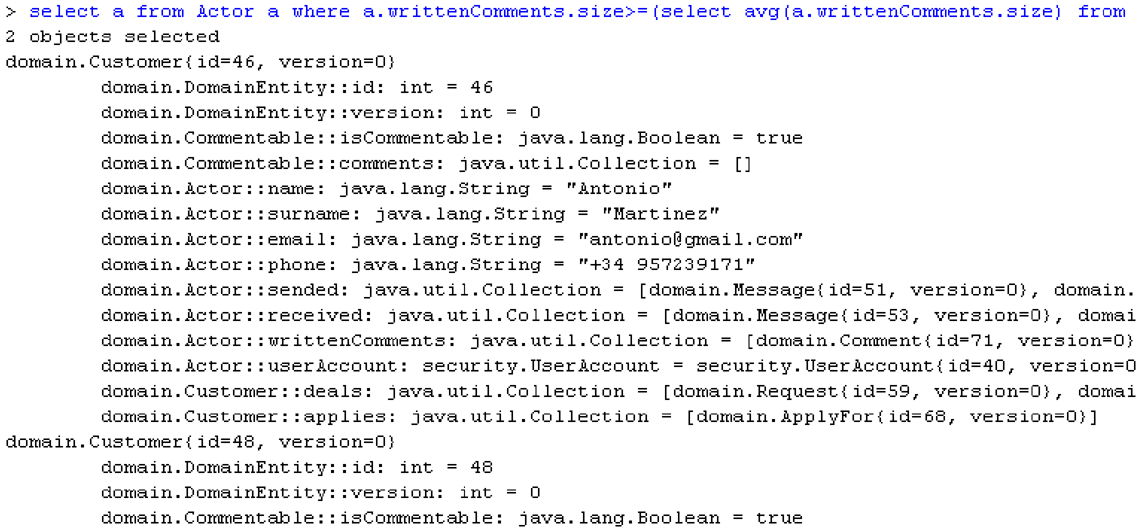


```
> select avg(a.writtenComments.size) from Actor a;
1 object selected
0.6667
```

B.3 - The actors who have posted ±10% the average number of comments per actor.

Realizo una búsqueda de los actores comparando el número de comentarios escritos con la media de comentarios escritos por actor, cogiendo al que supera el 10% de la media.

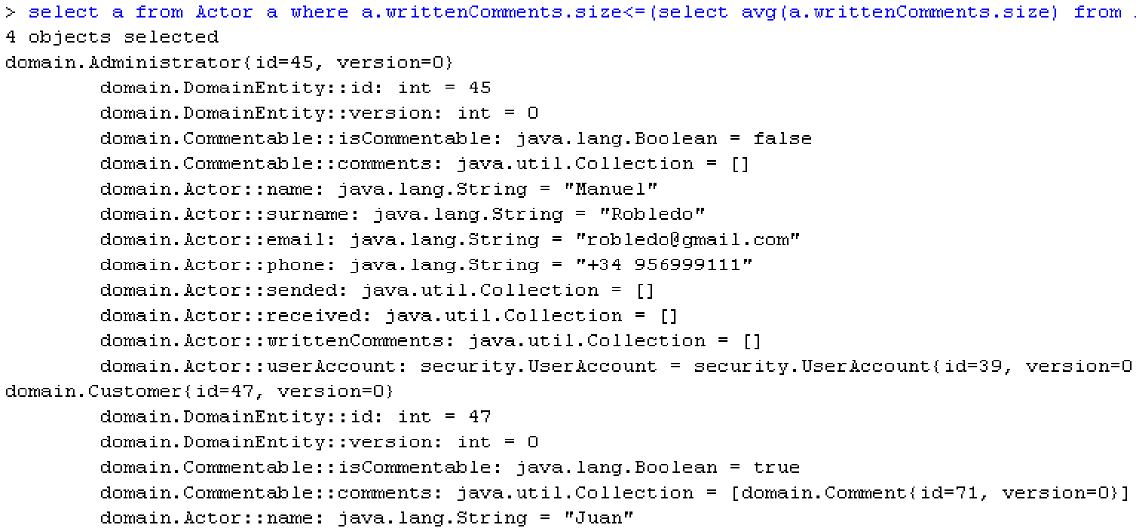
*select a from Actor a where a.writtenComments.size>=(select avg(a.writtenComments.size) from Actor a)*0.1;*



```
> select a from Actor a where a.writtenComments.size>=(select avg(a.writtenComments.size) from
2 objects selected
domain.Customer{id=46, version=0}
        domain.DomainEntity::id: int = 46
        domain.DomainEntity::version: int = 0
        domain.Commentable::isCommentable: java.lang.Boolean = true
        domain.Commentable::comments: java.util.Collection = []
        domain.Actor::name: java.lang.String = "Antonio"
        domain.Actor::surname: java.lang.String = "Martinez"
        domain.Actor::email: java.lang.String = "antonio@gmail.com"
        domain.Actor::phone: java.lang.String = "+34 957239171"
        domain.Actor::sended: java.util.Collection = [domain.Message{id=51, version=0}, domain.
        domain.Actor::received: java.util.Collection = [domain.Message{id=53, version=0}, domai
        domain.Actor::writtenComments: java.util.Collection = [domain.Comment{id=71, version=0}
        domain.Actor::userAccount: security.UserAccount = security.UserAccount{id=40, version=0
        domain.Customer::deals: java.util.Collection = [domain.Request{id=59, version=0}, domai
        domain.Customer::applies: java.util.Collection = [domain.ApplyFor{id=68, version=0}]
domain.Customer{id=48, version=0}
        domain.DomainEntity::id: int = 48
        domain.DomainEntity::version: int = 0
        domain.Commentable::isCommentable: java.lang.Boolean = true
```

Realizo una búsqueda de los actores comparando el número de comentarios escritos con la media de comentarios escritos por actor, cogiendo al que supera el 10% de la media.

> *select a from Actor a where a.writtenComments.size<=(select avg(a.writtenComments.size) from Actor a)*0.1;*

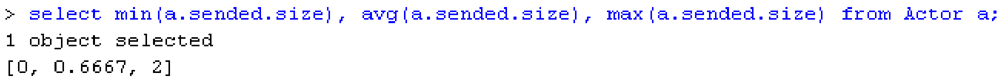


```
> select a from Actor a where a.writtenComments.size<=(select avg(a.writtenComments.size) from .
4 objects selected
domain.Administrator{id=45, version=0}
        domain.DomainEntity::id: int = 45
        domain.DomainEntity::version: int = 0
        domain.Commentable::isCommentable: java.lang.Boolean = false
        domain.Commentable::comments: java.util.Collection = []
        domain.Actor::name: java.lang.String = "Manuel"
        domain.Actor::surname: java.lang.String = "Robledo"
        domain.Actor::email: java.lang.String = "robledo@gmail.com"
        domain.Actor::phone: java.lang.String = "+34 956999111"
        domain.Actor::sended: java.util.Collection = []
        domain.Actor::received: java.util.Collection = []
        domain.Actor::writtenComments: java.util.Collection = []
        domain.Actor::userAccount: security.UserAccount = security.UserAccount{id=39, version=0
domain.Customer{id=47, version=0}
        domain.DomainEntity::id: int = 47
        domain.DomainEntity::version: int = 0
        domain.Commentable::isCommentable: java.lang.Boolean = true
        domain.Commentable::comments: java.util.Collection = [domain.Comment{id=71, version=0}]
        domain.Actor::name: java.lang.String = "Juan"
```

A.1 - The minimum, the average, and the maximum number of messages sent per actor

Realizo la operación de mínimo, media y máximo de los mensajes enviados por actor.

select min(a.sended.size), avg(a.sended.size), max(a.sended.size) from Actor a;

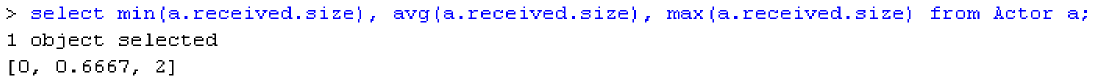


```
> select min(a.sended.size), avg(a.sended.size), max(a.sended.size) from Actor a;
1 object selected
[0, 0.6667, 2]
```

A.2 - The minimum, the average, and the maximum number of messages received per actor

Realizo la operación de mínimo, media y máximo de los mensajes recibidos por actor.

select min(a.received.size), avg(a.received.size), max(a.received.size) from Actor a;

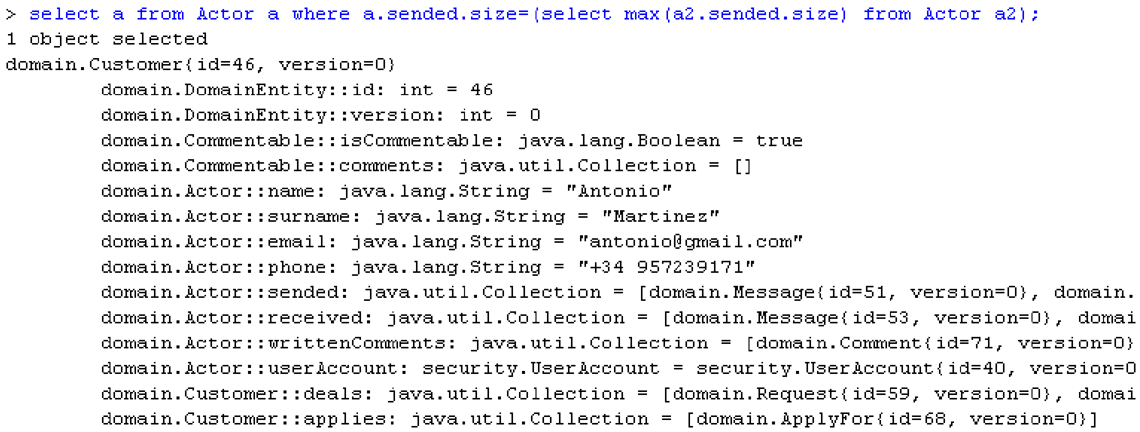


```
> select min(a.received.size), avg(a.received.size), max(a.received.size) from Actor a;
1 object selected
[0, 0.6667, 2]
```

A.3 - The actors who have sent more messages

Realizo una búsqueda de los actores comparando el número de mensajes enviados con el número máximo de mensajes enviados por actor, y cogiendo al que tenga el máximo número de mensajes.

select a from Actor a where a.sended.size=(select max(a2.sended.size) from Actor a2);

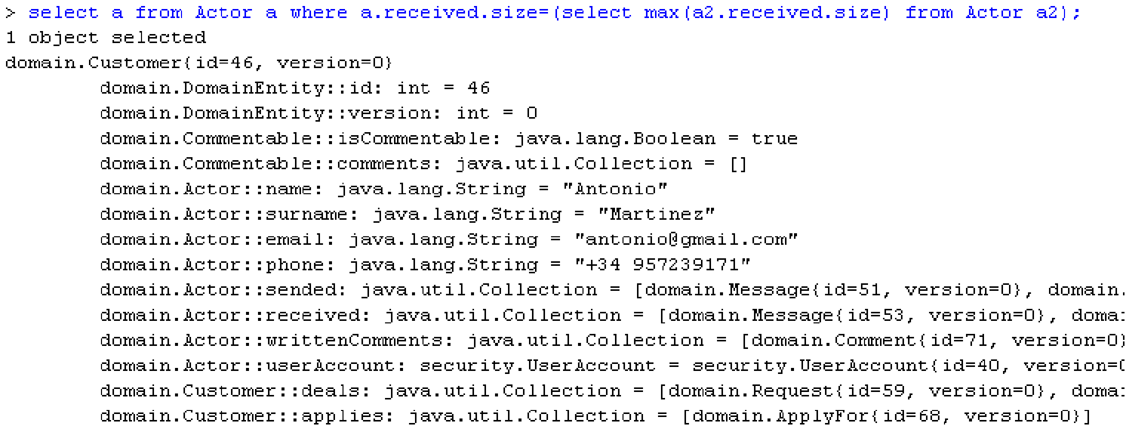


```
> select a from Actor a where a.sended.size=(select max(a2.sended.size) from Actor a2);
1 object selected
domain.Customer{id=46, version=0}
        domain.DomainEntity::id: int = 46
        domain.DomainEntity::version: int = 0
        domain.Commentable::isCommentable: java.lang.Boolean = true
        domain.Commentable::comments: java.util.Collection = []
        domain.Actor::name: java.lang.String = "Antonio"
        domain.Actor::surname: java.lang.String = "Martinez"
        domain.Actor::email: java.lang.String = "antonio@gmail.com"
        domain.Actor::phone: java.lang.String = "+34 957239171"
        domain.Actor::sended: java.util.Collection = [domain.Message{id=51, version=0}, domain.
        domain.Actor::received: java.util.Collection = [domain.Message{id=53, version=0}, domai
        domain.Actor::writtenComments: java.util.Collection = [domain.Comment{id=71, version=0}
        domain.Actor::userAccount: security.UserAccount = security.UserAccount{id=40, version=0
        domain.Customer::deals: java.util.Collection = [domain.Request{id=59, version=0}, domai
        domain.Customer::applies: java.util.Collection = [domain.ApplyFor{id=68, version=0}]
```

A.4 - The actors who have got more messages

Realizo una búsqueda de los actores comparando el número de mensajes recibidos con el número máximo de mensajes recibidos por actor, y cogiendo al que tenga el máximo número de mensajes.

> *select a from Actor a where a.received.size=(select max(a2.received.size) from Actor a2);*



```
> select a from Actor a where a.received.size=(select max(a2.received.size) from Actor a2);
1 object selected
domain.Customer{id=46, version=0}
        domain.DomainEntity::id: int = 46
        domain.DomainEntity::version: int = 0
        domain.Commentable::isCommentable: java.lang.Boolean = true
        domain.Commentable::comments: java.util.Collection = []
        domain.Actor::name: java.lang.String = "Antonio"
        domain.Actor::surname: java.lang.String = "Martinez"
        domain.Actor::email: java.lang.String = "antonio@gmail.com"
        domain.Actor::phone: java.lang.String = "+34 957239171"
        domain.Actor::sended: java.util.Collection = [domain.Message{id=51, version=0}, domain.
        domain.Actor::received: java.util.Collection = [domain.Message{id=53, version=0}, doma
        domain.Actor::writtenComments: java.util.Collection = [domain.Comment{id=71, version=0}
        domain.Actor::userAccount: security.UserAccount = security.UserAccount{id=40, version=
        domain.Customer::deals: java.util.Collection = [domain.Request{id=59, version=0}, doma
        domain.Customer::applies: java.util.Collection = [domain.ApplyFor{id=68, version=0}]
```