FLEXIBLE SEARCH QUERY

- it’s a hybris built-in query language based on SQL syntax.
- It enables us to search the records from the database using item types.
- In flexible search queries, we never use database table names.

We always use item types which will be mapped to a corresponding tables by Hybris.

Flexible search query execution has 2 phases

1) Pre-parsing phase
In this phase, Hybris converts the flexible search query into SQL query.

2) Executing the SQL converted query
In this phase,converted SQL query will be executed by Hybris.

Flexible search query is cached by deafault in hybris cache. if multiple sources fire same query,it is picked from cache.

We can extract the same data using flexibleSearchService through beanshell.flexibleSearchService is oob spring bean.

Sample Example :

```
select {pk} , {code} , {name[de]} from {Product}
```

Select query in Orders table :

```
select * from {Order}
```

Query to fetch Orders Placed between 2 date :

```
SELECT * FROM {order as o }

      WHERE (to_char({o:creationtime}, 'YYYY-mm-dd hh24:mi:ss') >=  ('2010-12-28 23:59:59') )
and (to_char({o:creationtime}, 'YYYY-mm-dd hh24:mi:ss') <=  ('2020-01-17 10:00:00') )
```

Query to show joins :

```
SELECT

       {os.code},
     {o:code} as Hybris_OrdNbr
        FROM {order as o
        join orderStatus as os on {o:status} = {os:pk}

            }
```

Query to understand left join and right join :

```
select * from
{Order as o

join OrderStatus as os on {os.pk} = {o.status}

left join PaymentStatus as pts  on {pts.pk} = {o.paymentStatus}



join cmssite as cms on {o.site}={cms.pk}



JOIN address as ad on {o.deliveryAddress} = {ad.pk}
join Country as c ON {ad.country} = {c.pk}
}



where
{o.versionId} is null
```

Order by Query :

```
SELECT {o.CODE},{o.CREATIONTIME}

        ,{os.code}
          FROM {order as o
          join orderStatus as os on {o:status} = {os:pk}

                } ORDER BY {o.creationtime} DESC // latest date data will come first
```

And and OR condition :

```
SELECT {o.CODE},{o.CREATIONTIME}

        ,{os.code}
          FROM {order as o
          join orderStatus as os on {o:status} = {os:pk}

                }

              where ({os.code}='COMPLETED' OR {os.code}='CREATED')
              ORDER BY {o.creationtime} DESC
```

Exists query in Hybris :

```
SELECT * FROM {order as o} where  EXISTS ( {{select {os.pk} from {OrderStatus as os} where {os.code}='COMPLETED' }} )
```

Like Statement :

```
SELECT {o.code},{os.code} FROM {order as o JOIN OrderStatus as os on {o.status}={os.pk}} where {os.code} like '%ETED'   //here % is called as WildCard
```

Wild card matching a single character :

```
SELECT {o.code},{os.code} FROM {order as o JOIN OrderStatus as os on {o.status}={os.pk}} where {os.code} like 'COMP_ETED'
```

<> OR != keywords :

```
SELECT {o.code},{os.code} FROM {order as o JOIN OrderStatus as os on {o.status}={os.pk}} where {os.code} <> 'COMPLETED'
```

CONCAT Keyword:

```
SELECT CONCAT({o.code},{os.code}) FROM {order as o JOIN OrderStatus as os on {o.status}={os.pk}} where {os.code} <> 'COMPLETED'
```

CatalogVersion Query :

```
select {p.code},{cv.version} from {Product! as p  join catalogVersion as cv on {p.catalogVersion}={cv.pk}} where {cv.version}='Online'
```

When you fire a select statement say : select * from {Product} , it gives search results from this type as well as from it's subtypes , if you don't want sub types , than just give this ! with item type.

Flexible search query is cached by deafault in hybris cache. if multiple sources fire same query,it is picked from cache.

We can extract the same data using flexibleSearchService through beanshell.flexibleSearchService is oob spring bean.