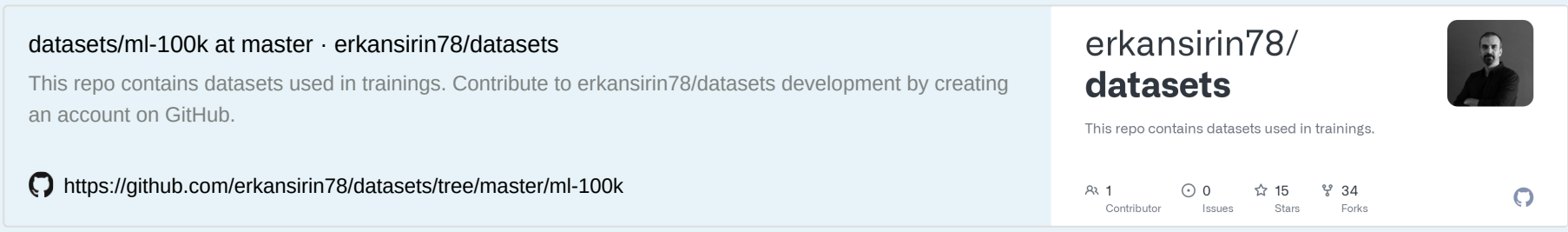
Apache Hive Partition ve Bucketing Uygulaması Çözümü

Movielens, Partition & Bucketing

▼ **Görev 1: Github repoda bulunan u.data ve u.item veri setlerini Hive'a tablo olarak yükleyiniz.**



▼ **Adım 1: datasets klasörüne veri setlerini indiriniz ve inceleyiniz.**

- u.data

```
wget -P ~/datasets/ https://raw.githubusercontent.com/erkansirin78/datasets/master/ml-100k/u.data
```

`head ~/datasets/u.data`

- u.item

```
wget -P ~/datasets/ https://raw.githubusercontent.com/erkansirin78/datasets/master/ml-100k/u.item
```

`head ~/datasets/u.item`

▼ **Adım 2:** Beeline'ı açınız ve `movielens` adında yeni bir veri tabanı oluşturunuz

`beeline -u jdbc:hive2://localhost:10000`

`CREATE DATABASE movielens;`

▼ **Adım 3: u.data** içeriğini inceleyiniz ve ratings adında veriye uygun bir tablo oluşturunuz.

```
create table if not exists movielens.ratings (
user_id int,
item_id int,
rating int,
rating_time bigint)
row format delimited
fields terminated by '\t'
lines terminated by '\n'
stored as textfile
tblproperties('skip.header.line.count'='1');
```

▼ **Adım 4: Localden ratings tablosuna veriyi yükleyiniz ve tabloyu inceleyiniz.**

`load data local inpath '/home/train/datasets/u.data' into table movielens.ratings;`

```
select * from movielens.ratings limit 4;
+------------------+------------------+-----------------+----------------------+
| ratings.user_id  | ratings.item_id  | ratings.rating  | ratings.rating_time  |
+------------------+------------------+-----------------+----------------------+
| 196              | 242              | 3               | 881250949            |
| 186              | 302              | 3               | 891717742            |
| 22               | 377              | 1               | 878887116            |
| 244              | 51               | 2               | 880606923            |
+------------------+------------------+-----------------+----------------------+
```

```
select count(1) from movielens.ratings;
+---------+
|   _c0   |
+---------+
| 100000  |
+---------+



select count(distinct user_id) from movielens.ratings;
+------+
| _c0  |
+------+
| 943  |
+------+
```

▼ Adım 5 u.item verisine uygun movies adında bir tablo oluşturunuz.

```
create table if not exists  movielens.movies (
    movieid int,
    movietitle string,
    releasedate string,
    videoreleasedate string,
    IMDbURL string,
    unknown tinyint,
    Action tinyint,
    Adventure tinyint,
    Animation tinyint,
    Childrens tinyint,
    Comedy tinyint,
    Crime tinyint,
    Documentary tinyint,
    Drama tinyint,
    Fantasy tinyint,
    FilmNoir tinyint,
    Horror tinyint,
    Musical tinyint,
    Mystery tinyint,
    Romance tinyint,
    SciFi tinyint,
    Thriller tinyint,
    War tinyint,
    Western  tinyint)
    row format delimited
fields terminated by '|'
lines terminated by '\n'
stored as textfile
tblproperties('skip.header.line.count'='1');
```

▼ Adım 6: Localden movies tablosuna veriyi yükleyiniz ve tabloyu inceleyiniz.

```
load data local inpath '/home/train/datasets/u.item' into table movielens.movies;
```

```
select movieid, movietitle, releasedate from  movielens.movies limit 5;
+----------+------------------+--------------+
| movieid  |    movietitle    | releasedate  |
+----------+------------------+--------------+
| 1        | ToyStory(1995)   | 01-Jan-1995  |
| 2        | GoldenEye(1995)  | 01-Jan-1995  |
| 3        | FourRooms(1995)  | 01-Jan-1995  |
| 4        | GetShorty(1995)  | 01-Jan-1995  |
| 5        | Copycat(1995)    | 01-Jan-1995  |
+----------+------------------+--------------+

select count(1) from movielens.movies;
+-------+
|  _c0  |
+-------+
| 1682  |
+-------+
```

▼ **Görev 2:** İş kullanıcıları bazı sorgulamalar yapmayı ve bu sorguların mümkün olduğu kadar kısa süre içinde sonuçlanmasını talep etmektedir. İş kullanıcılarının söz konusu ihtiyacını karşılamak üzere Hive üzerinde gerekli veri organizasyonunu yapınız.

▼ **Adım 1:** Aylık olarak en popüler (en çok oylanan, en yüksek ortalama puanı alan) filmler belirlenmek istenmektedir. Buna göre tabloyu tasarlayıp (partition ve bucketing), oluşturunuz.

İş kullanıcıları aylık olarak filmleri sorguladığı için yıl ve aya göre partition yapıp film adlarını da bucket yaparsam sorgu performansını arttırabiliriz.

```
create table if not exists movielens.movie_ratings (
user_id int,
rating int,
rating_time bigint,
movieid int,
movietitle string,
videoreleasedate string,
imdburl string)
partitioned by (review_year int, review_month int)
clustered by (movietitle) into 4 buckets
stored as orc;
```

▼ **Adım 2: Dinamik Partitioning ayarlayınız.**

```
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.enforce.bucketing=true;
```

▼ **Adım 3:** İki tablo verilerini tasarladığınız tabloya yükleyiniz.

```
insert overwrite table movielens.movie_ratings PARTITION(review_year, review_month)
select user_id,
rating,
rating_time,
movieid,
movietitle,
videoreleasedate,
imdburl,
YEAR(from_unixtime(rating_time, 'yyyy-MM-dd')) as review_year,
MONTH(from_unixtime(rating_time, 'yyyy-MM-dd')) as review_month
from movielens.ratings r join movielens.movies m on r.item_id = m.movieid;
```

▼ **Adım 4: Oluşturduğunuz tabloyu kontrol ediniz.**

▼ Gözlem sayısı nedir?

```
select count(1) from movielens.movie_ratings;
+---------+
|   _c0   |
+---------+
| 100000  |
+---------+
```

▼ Partitionları listeleyiniz

```
show partitions movielens.movie_ratings;
+-----------------------------------+
|             partition             |
+-----------------------------------+
| review_year=1997/review_month=10  |
| review_year=1997/review_month=11  |
| review_year=1997/review_month=12  |
| review_year=1997/review_month=9   |
| review_year=1998/review_month=1   |
| review_year=1998/review_month=2   |
| review_year=1998/review_month=3   |
| review_year=1998/review_month=4   |
+-----------------------------------+
```

▼ Tablo özelliklerini listeleyiniz

```
describe movielens.movie_ratings;
+--------------------------+------------+----------+
|         col_name         | data_type  | comment  |
+--------------------------+------------+----------+
| user_id                  | int        |          |
| rating                   | int        |          |
| rating_time              | bigint     |          |
| movieid                  | int        |          |
| movietitle               | string     |          |
| videoreleasedate         | string     |          |
| imdburl                  | string     |          |
| review_year              | int        |          |
| review_month             | int        |          |
|                          | NULL       | NULL     |
| # Partition Information  | NULL       | NULL     |
| # col_name               | data_type  | comment  |
| review_year              | int        |          |
| review_month             | int        |          |
+--------------------------+------------+----------+
```

▼ Review Year ve Review Month olarak kaç unique değer vardır?

```
select distinct (review_year, review_month) from movielens.movie_ratings;
+--------------------------+
|           _c0            |
+--------------------------+
| {"col1":1997,"col2":9}   |
| {"col1":1997,"col2":10}  |
| {"col1":1997,"col2":11}  |
| {"col1":1997,"col2":12}  |
| {"col1":1998,"col2":1}   |
| {"col1":1998,"col2":2}   |
| {"col1":1998,"col2":3}   |
| {"col1":1998,"col2":4}   |
+--------------------------+
```

▼ **Görev 3: İstenen analizler için gerekli sorguları oluşturup yorumlayınız.**

▼ **Adım 1:** 1998 yılının Nisan ayında en çok puanlanan 20 filmini bulunuz.

```
select  count(*) total_count,  movietitle
from movielens.movie_ratings
where review_year=1998 AND review_month=4
group by movietitle order by total_count desc limit 20;

+--------------+------------------------------+
| total_count  |          movietitle          |
+--------------+------------------------------+
| 63           | Titanic(1997)                |
| 52           | AirForceOne(1997)            |
| 50           | Contact(1997)                |
| 49           | FullMonty,The(1997)          |
| 49           | StarWars(1977)               |
| 42           | GoodWillHunting(1997)        |
| 41           | LiarLiar(1997)               |
| 41           | EnglishPatient,The(1996)     |
| 39           | AsGoodAsItGets(1997)         |
| 39           | ConspiracyTheory(1997)       |
| 37           | Scream(1996)                 |
| 36           | ToyStory(1995)               |
| 36           | Fargo(1996)                  |
| 36           | ReturnoftheJedi(1983)        |
| 35           | L.A.Confidential(1997)       |
| 34           | ChasingAmy(1997)             |
| 34           | Godfather,The(1972)          |
| 33           | Braveheart(1995)             |
| 33           | StarshipTroopers(1997)       |
| 33           | SilenceoftheLambs,The(1991)  |
+--------------+------------------------------+
```

▼ **Adım 2:** 1998 yılının Nisan ayında oylanan filmlerden en yüksek ortalama puana sahip 20 filmi bulunuz.

```
select avg(rating) as avg_rating, count(*) total_count,  movietitle
from movielens.movie_ratings
where review_year=1998 AND review_month=4
group by movietitle order by avg_rating desc limit 20;

+-------------+--------------+-----------------------------------------+
| avg_rating  | total_count  |                movietitle               |
+-------------+--------------+-----------------------------------------+
| 5.0         | 3            | CelluloidCloset,The(1995)               |
| 5.0         | 1            | Boys,Les(1997)                          |
| 5.0         | 1            | Flirt(1995)                             |
| 5.0         | 1            | FreeWilly2:TheAdventureHome(1995)       |
| 5.0         | 1            | DeltaofVenus(1994)                      |
| 5.0         | 1            | CutthroatIsland(1995)                   |
| 5.0         | 1            | DunstonChecksIn(1996)                   |
| 5.0         | 2            | Diexueshuangxiong(Killer,The)(1989)     |
| 5.0         | 1            | Lassie(1994)                            |
| 5.0         | 1            | Innocents,The(1961)                     |
| 5.0         | 1            | Stalingrad(1993)                        |
| 5.0         | 1            | FearofaBlackHat(1993)                   |
| 5.0         | 1            | Trust(1990)                             |
| 5.0         | 1            | BoxingHelena(1993)                      |
| 5.0         | 1            | DavyCrockett,KingoftheWildFrontier(1955)  |
| 5.0         | 1            | BitterSugar(AzucarAmargo)(1996)         |
| 5.0         | 1            | BlueSky(1994)                           |
| 5.0         | 1            | Daylight(1996)                          |
| 5.0         | 2            | Prefontaine(1997)                       |
| 5.0         | 1            | 8Seconds(1994)                          |
+-------------+--------------+-----------------------------------------+
```