

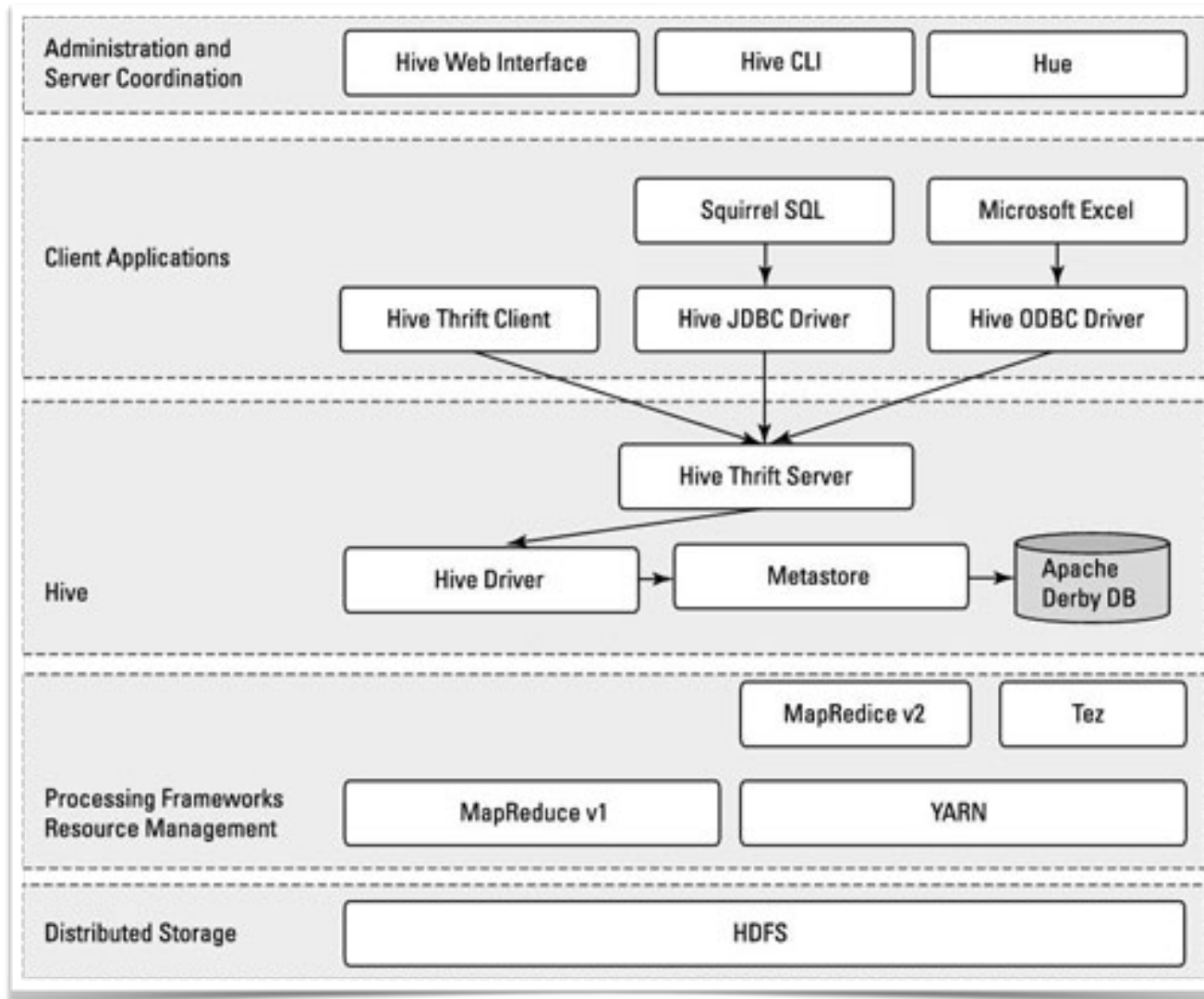


# HIVE

## Практика

[NEWPROLAB.COM](http://NEWPROLAB.COM)

# Hive архитектура



# Списки объектов

- `show databases;`
- `show tables [in 'db_name']`
- `show partitions <tbl_name>`
- `show create table <tbl_name>`

# Таблицы

- Regular

```
CREATE TABLE users  
(  
  user_id int,  
  age int,  
  gender string,  
  occupation string,  
  zip string  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '|';
```

# Таблицы

- External

```
create external table genre
```

```
(  
  name string,  
  id int  
)
```

```
ROW FORMAT DELIMITED
```

```
FIELDS TERMINATED BY '|'
```

```
location '/user/anton.pilipenko/npl/';
```

# Скрипты

```
CREATE TABLE users  
(  
  user_id int,  
  age int,  
  gender string,  
  occupation string,  
  zip string  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '|'   
STORED AS TEXTFILE;
```

# Скрипты

```
create table movies
```

```
(  
  movie_id int, movie_title string, release_date string,  
  video_release_date string, IMDbURL string, unknown int,  
  Action int, Adventure int, Animation int, Childrens int,  
  Comedy int, Crime int, Documentary int, Drama int,  
  Fantasy int, FilmNoir int,  
  Horror int, Musical int, Mystery int, Romance int, SciFi  
  int, Thriller int, War int, Western int  
)
```

```
row format delimited
```

```
fields terminated by '\|'
```

```
STORED AS TEXTFILE;;
```

# Скрипты

```
CREATE TABLE rating  
(  
  user_id int,  
  item_id int,  
  rating int  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
STORED AS TEXTFILE;
```



# Скрипты

```
create external table genre  
(  
    name string,  
    id int  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '|'   
location '/user/anton.pilipenko/npl/';
```

# Загрузка данных

```
load data <local> inpath  
'/user/anton.pilipenko/npl/rating.data'  
into table rating;
```

# Задания

получить средний возраст пользователей

# Задания

получить средний возраст пользователей

```
select avg(age)  
from users;
```

# Задания

для пользователей старше 21 года  
получить статистику количества  
пользователей в разрезе возраста только  
для тех групп в которых более 10 человек

# Задания

```
select count(*) cnt, age from users  
where age > 21  
group by age  
having count(*) > 10  
order by age;
```

# Задания

получить название, минимальную, максимальную и среднюю оценки комедий, отсортированные по убыванию среднего рейтинга

# Задания

```
Select      min(rating) min_rating,  
            max(rating) max_rating,  
            avg(rating) avg_rating,  
            m.movie_title  
from rating r  
join movies m on (r.item_id = m.movie_id)  
where m.comedy = 1  
group by m.movie_title  
order by avg_rating desc
```



# Виртуальные поля

```
select INPUT__FILE__NAME,  
BLOCK__OFFSET__INSIDE__FILE,  
m.*  
from movies m;
```

# Партиционирование

```
CREATE TABLE rating_parted(  
  user_id int,  
  item_id int)  
partitioned by (rating int)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t';
```

# Партиционирование

```
insert into table rating_parted partition  
(rating=1)
```

```
select t.user_id, t.item_id from rating t  
where t.rating=1;
```

# Выгрузки данных

```
insert overwrite local directory '/home/apilipenko/result.csv'  
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\';  
select /*+ MAPJOIN(dbl) */ t.ban,  
t.subscriber_no,  
t.imsi,  
lag(t.imsi) over (partition by t.ban, t.subscriber_no order by  
eff_date_time) old_imsi,  
t.eff_date_time,  
t.exp_date_time  
from subscriber_sim t  
join (select ban, subscriber_no from subscriber_sim where  
eff_date_time >= '2015-07-27' and eff_date_time < '2015-08-03'  
group by ban, subscriber_no ) dbl on (dbl.ban = t.ban and  
dbl.subscriber_no = t.subscriber_no);
```