

AD – учетная запись

1. Склонировать репозитории на свою локальную машину
2. Создать в каждом репозитории ветку для разработки – develop
3. Переключиться на ветку develop
4. Добавить файл .gitignore, содержащий:

```
.idea/*  
.bsp/*  
target/*  
project/*  
*.iml  
  
!project/build.properties  
!project/plugins.sbt
```

5. Написать код в каждом репозитории, согласно заданиям ниже.
6. datamarts-<AD>: Написание спарк-приложения:
Используемые версии зависимостей:

ThisBuild / scalaVersion := "2.11.8"

val sparkVersion = "2.3.2"

libraryDependencies ++= Seq(

"org.apache.spark" %% "spark-core" % sparkVersion % "provided",

"org.apache.spark" %% "spark-sql" % sparkVersion % "provided",

"org.postgresql" % "postgresql" % "42.3.3"

)

Написать два приложения (scala класса)

1. Загрузка данных из реляционной БД на кластер Hadoop.
Необходимо загрузить данные из базы
postgresql://XXX
в hive базу school_de таблицы:
bookings.seats -> school_de.seats_<AD>
bookings.flights_v -> school_de.flights_v_<AD> (загрузить с разбивкой на партиции по
актуальной дате вылета))
В коде для сдачи доступ к postgres осуществлять из под ТУЗ tech_repbgd_local
 2. Выполнение запросов из практического задания v.0.1 используя распределенные
вычисления spark, записав результат в таблицу school_de.results_<AD>
Для решения задачи использовать Spark DataFrame API, без использования spark.sql("...")
7. dags-<AD>: Создать два дага
1. Для загрузки данных задачи п. 8.1 (***доступ к postgres осуществлять из под ТУЗ
tech_repbgd_local, получив кредиты из Vault***)
 2. Для расчёта данных задачи п. 8.2
- Оба дага должны иметь регламент запуска None и запускаться из под ТУЗ
tech_schoolde_bgd_ms***

P.S. ***Доступ к postgres для разработки производить из под личной УЗ, база керберезированная.***