Задание №1

В некотором банке внедрили новую frontend-систему для работы с клиентами, а так же обновили и саму базу данных. Большую часть данных успешно были перенесены из старых БД в одну новую централизованную БД. Но в момент переключения со старой системы на новую возникли непредвиденные проблемы в ETL-процессе, небольшой период (конец 2017 начало 2018 года) так и остался в старой базе. Старую базу отключили, а не выгруженные данные сохранили в csv-файлы. Недавно банку потребовалось построить отчёт по 101 форме. Те данные что остались в csv-файлах тоже нужны. Загрузить их в новую БД не получиться из-за архитектурных и управленческих сложностей, нужно рассчитать витрину отдельно. Но для этого сначала нужно загрузить исходные данные из csv-файлов в детальный слой (DS) хранилища в СУБД Oracle / PostgreSQL.

**Задача 1.2**

После того как детальный слой «DS» успешно наполнен исходными данными из файлов – нужно рассчитать витрины данных в слое «DM»: витрину оборотов и витрину 101-й отчётной формы.

Для этого вам сперва необходимо построить витрину оборотов «DM.DM\_ACCOUNT\_TURNOVER\_F». А именно, посчитать за каждый день января 2018 года кредитовые и дебетовые обороты по счетам с помощью Oracle-пакета dm.fill\_account\_turnover\_f или с помощью аналогичной PostgreSQL-процедуры.

Затем вы узнаёте от Аналитика в банке, что пакет (или процедуру) расчёта витрины 101-й формы «dm.fill\_f101\_round\_f» необходимо доработать. Необходимо сделать расчёт полей витрины «dm.dm\_f101\_round\_f» по формулам:

* *«BALANCE\_OUT\_RUB»*  
  для счетов с CHARACTERISTIC = 'A' и currency\_code '643' рассчитать *BALANCE\_OUT\_RUB* = BALANCE\_IN\_RUB - TURN\_CRE\_RUB + TURN\_DEB\_RUB;  
    
  для счетов с CHARACTERISTIC = 'A' и currency\_code '810' рассчитать *BALANCE\_OUT\_RUB* = BALANCE\_IN\_RUB - TURN\_CRE\_RUB + TURN\_DEB\_RUB;  
    
  для счетов с CHARACTERISTIC = 'P' и currency\_code '643' рассчитать *BALANCE*\_OUT\_RUB = BALANCE\_IN\_RUB + TURN\_CRE\_RUB - TURN\_DEB\_RUB;  
    
  для счетов с CHARACTERISTIC = 'P' и currency\_code '810' рассчитать *BALANCE\_OUT\_RUB* = BALANCE\_IN\_RUB + TURN\_CRE\_RUB - TURN\_DEB\_RUB;
* *«BALANCE\_OUT\_VAL»*для счетов с CHARACTERISTIC = 'A' и currency\_code не '643' и не '810' рассчитать *BALANCE\_OUT\_VAL* = BALANCE\_IN\_VAL - TURN\_CRE\_VAL + TURN\_DEB\_VAL;  
    
  для счетов с CHARACTERISTIC = 'P' и currency\_code не '643' и не '810' рассчитать *BALANCE\_OUT\_VAL* = BALANCE\_IN\_VAL + TURN\_CRE\_VAL - TURN\_DEB\_VAL;
* *«BALANCE\_OUT\_TOTAL»*рассчитать *BALANCE\_OUT\_TOTAL* как BALANCE\_OUT\_VAL + BALANCE\_OUT\_RUB

Обратите внимание, что в предоставленных вам пакетах (процедурах) есть процедура логирования, под них нужно создать соответствующие таблицы или реализовать собственный процесс логирования расчёта витрин – это будет только плюсом.

Примечания:

* К заданию прилагаются артефакты Oracl и PostgreSQL – это пакеты и процедуры наполнения витрин соответственно. Применяйте те, что подходят под вашу БД;
* Для витрин данных и логирования не забудьте создать схему «DM»;
* В процедурах логирования генерируется id строки на основании последовательности. Этот sequence предварительно нужно создать (как в Oracle, так и в Postgres).   
  Касательно Oracle ниже 19-й версии последовательности с автогенерацией создать нельзя, нужно дополнительно создавать триггер, который будет её увеличивать;
* Расчёт витрин в Oracle можно сделать c помощью заданий (Job Sheduler), в PostgreSQL с помощью расширения «pg\_cron» (придётся повозится с настройкой немного).   
  Так же вы можете реализовать запуск расчёта с помощью Python / Scala / Java.