Техническое задание на разработку хранилища данных. Этап 1.

Введение.

Сегодня ваш первый день работы инженером по данным в компании такси “Везу и точка”. К вам приходит начальник и говорит тоном, не терпящим возражений: “Что это мы живём без нормальных отчётов? Мне нужно, чтобы ты закончил работу по их созданию”. После чего подмигивает и уходит. От коллеги справа, который работает с операционной системой, в которой ведётся деятельность компании, вы узнаёте, что ваш предшественник начал делать эту задачу, но внезапно решил уволиться.

Открыв папку с документацией, где предшественник всё хранил, вы находите документ, который описывает источник, актуальность которого подтверждает всё тот же коллега справа, а также описание целевой структуры хранилища. Чтобы не терять времени вы решаете использовать её.

Ваша первая задача - настроить ETL процесс, который будет забирать данные из источника и раскладывать их по целевым таблицам в описанную в документации структуру в хранилище.

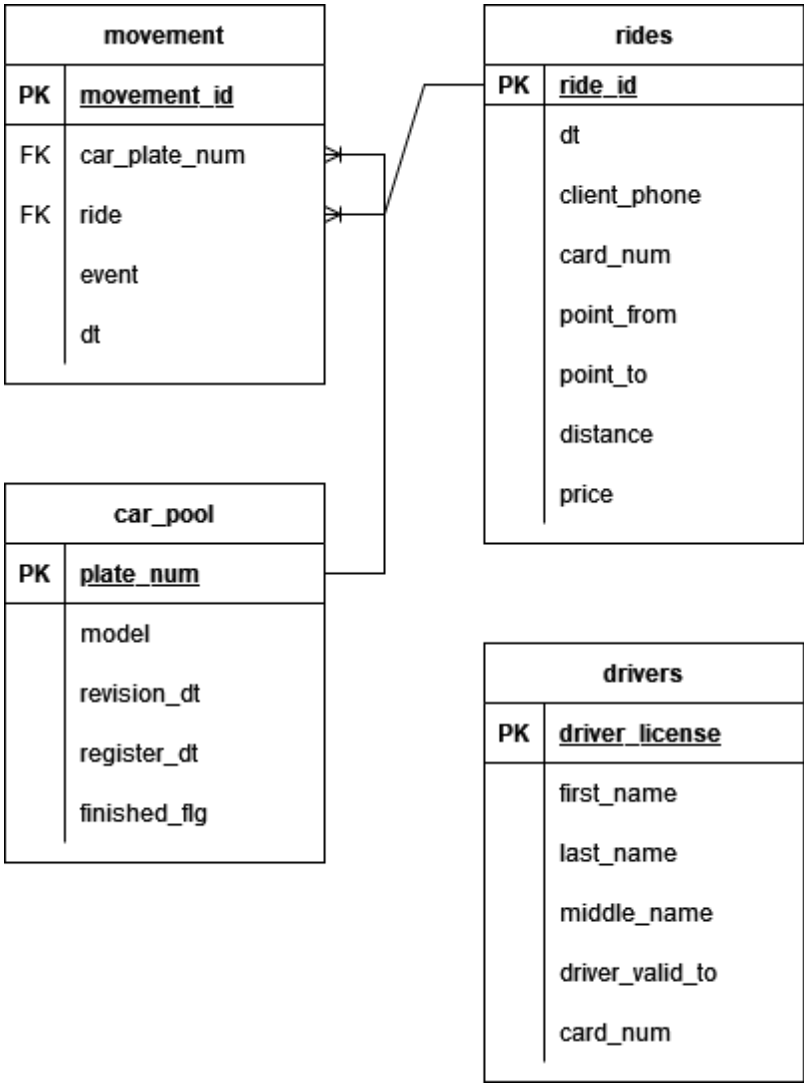
Описание источников данных.

1. Источник данных - СУБД PostgreSQL

Реквизиты подключения к источнику техническим пользователем:

- host: de-edu-db.chronosavant.ru
- port: 5432
- database: taxi
- schema: main
- user: etl_tech_user
- password: etl_tech_user_password

ER-диаграмма.



Описание сущностей.

Таблица main.rides

Заказы, поступающие от клиентов.

PK	NAME	TYPE	DESCRIPTION
PK	ride_id	INTEGER	Идентификатор заказа-поездки. Автоинкремент.
	dt	TIMESTAMP(0)	Дата и время заказа.
	client_phone	CHAR(18)	Номер телефона клиента.

	card_num	CHAR(19)	Карта клиента *(обратите внимание, при разных заказах одного и того же клиента может быть разной).*
	point_from	VARCHAR(200)	Начальный адрес.
	point_to	VARCHAR(200)	Конечный адрес.
	distance	NUMERIC(5,2)	Дистанция поездки.
	price	NUMERIC(7,2)	Цена поездки.

Таблица main.movement

Статусы машин, направленных на заказ. Каждый статус добавляется новой строкой с временно́й меткой. При успешно выполненном заказе по одному заказу будет три записи - подача машины (READY), начало поездки (BEGIN) и окончание поездки (END). Отмена заказа (CANCEL) возможна только до его начала, при этом завершения заказа (END) не будет.

PK	**NAME**	**TYPE**	**DESCRIPTION**
PK	movement_id	INTEGER	Идентификатор факта перемещения. Автоинкремент.
FK	car_plate_num	CHAR(9)	Номер машины.
FK	ride	INTEGER	Ссылка на заказ-поездку.
	event	VARCHAR(6)	Произошедшее событие: READY - машина подана BEGIN - начало выполнения заказа CANCEL - отмена заказа (при таком статусе END не будет) END - завершение заказа (если он начал выполняться)
	dt	TIMESTAMP(0)	Дата и время события.

Таблица main.car_pool

Машины, зарегистрированные в автопарке. Машина регистрируется при первом участии ее в любой поездке. Каждые три дня машина должна проходить осмотр, отметка о проведении осмотра *обновляется* в таблице. Если машина больше не работает с компанией - обновляется соответствующий флаг (в данном кейсе таких ситуаций нет).

PK	**NAME**	**TYPE**	**DESCRIPTION**
PK	plate_num	CHAR(9)	Номер автомобиля

	model	VARCHAR(30)	Марка автомобиля
	revision_dt	DATE	Дата последнего осмотра
	register_dt	TIMESTAMP(0)	Дата первой регистрации машины в парке
	finished_flg	CHAR(1)	Машина списана (больше не работает в парке): Y/N

Таблица main.drivers
Водители, работающие в компании. Водители могут добавляться независимо от поездок. Иногда у водителя может меняться номер карты, это изменение фиксируется обновлением соответствующего поля.

PK	NAME	TYPE	DESCRIPTION
PK	driver_license	CHAR(12)	Номер водительского удостоверения
	first_name	VARCHAR(20)	Фамилия
	last_name	VARCHAR(20)	Имя
	middle_name	VARCHAR(20)	Отчество
	driver_valid_to	DATE	Срок действия водительского удостоверения
	card_num	CHAR(19)	Номер карты водителя
	update_dt	TIMESTAMP(0)	Дата и время обновления записи

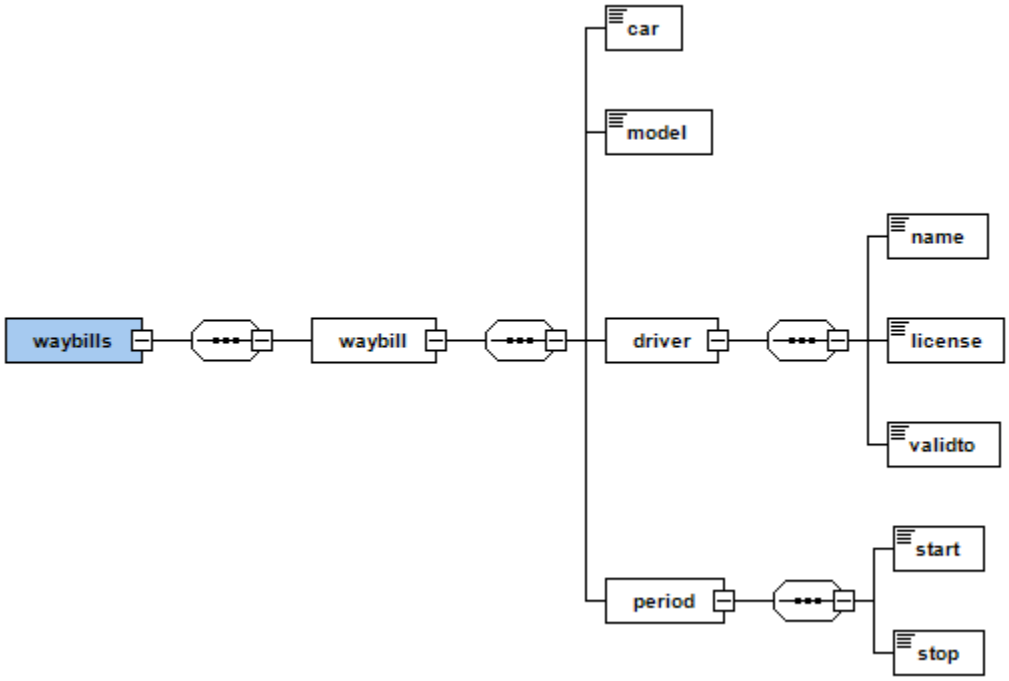
2. Источник данных - файловые выгрузки.

Реквизиты подключения к FTP over TLS техническим пользователем:

- host: de-edu-db.chronosavant.ru
- port: 21
- user: etl_tech_user
- password: etl_tech_user_password

Каталог waybills содержит путевые листы, в одном файле один путевой лист. Путевые листы содержат информацию о том, какой водитель в какой период времени на какой машине работал.
Файлы выгружаются по факту появления. Маска имени waybill_NNNNNN.xml, номер NNNNNN возрастает по порядку начиная с 1.

Структура файла:



Каталог payments содержит бухгалтерские выписки о поступлении денег от клиентов на счет компании. Файлы выгружаются раз в полчаса. Маска имени payment_YYYY-MM-DD_HH-MI.csv.

Структура файла (разделители - табуляция):

Поле	**Формат**	**Длина**
Дата и время	DD.MM.YYYY HH24:MI:SS	19
Номер карты	NNNNNNNNNNNNNNN	16
Сумма платежа	##N.NN	4-6

Требования к хранилищу данных.

На первом этапе от вас требуется создать детальный слой хранилища данных. В детальном слое обычно создаются таблицы, описывающие каждую бизнес-сущность, существующую в компании. В нашем случае это будут: заказы (поездки), клиенты, машины, водители и платежи. Также необходима техническая таблица, связывающая водителей и машины - в нашем случае это информация о путевых листах.

Приемник данных (хранилище) - СУБД PostgreSQL

Реквизиты подключения к хранилищу техническим пользователем:

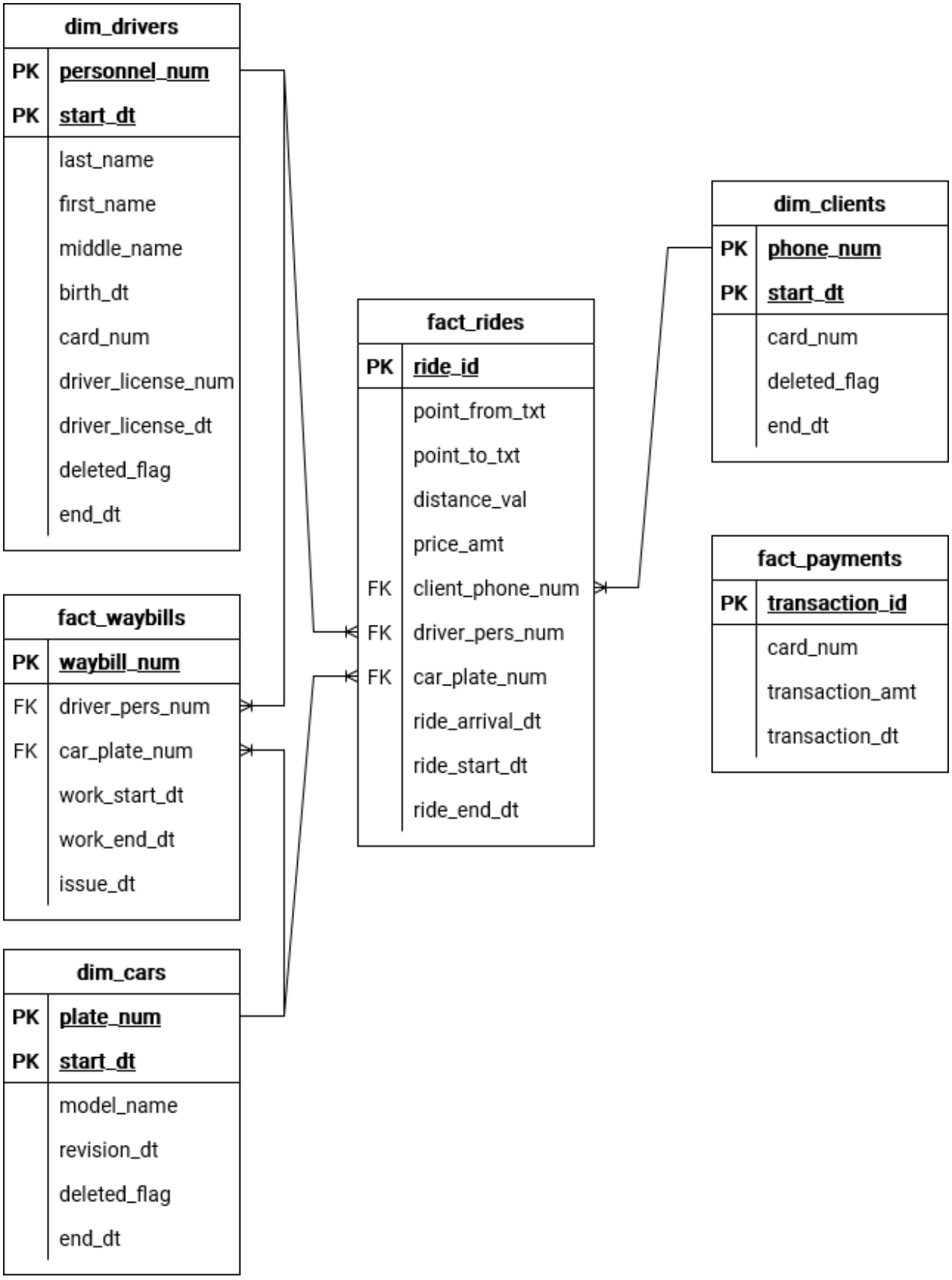
- host: de-edu-db.chronosavant.ru
- port: 5432
- database: dwh
- schema: dwh_<имя_города_вашей_команды>
- user: dwh_<имя_города_вашей_команды>
- password: <пароль_вашей_команды>

ER-диаграмма целевого хранилища данных.

Обратите внимание что вам необходимо строго следовать правилам именования всех объектов. Типизацию выбирайте из соображений разумной потребности (например, поле last_name вряд ли стоит назначать типом DATE, разумнее выбрать VARCHAR. Длину поля тоже имеет смысл сделать 50 символов, а не 25000, это, кстати, может быть поводом для отдельного исследования).

В хранилищах данных, как правило, не назначаются ограничения целостности типа unique, pk, fk (из-за их большой ресурсоемкости), поэтому проверка constraints не предусмотрена.

Возможно создание дополнительных объектов, если это обосновано. Все технические таблицы, если они вам потребуются, должны быть промаркированы префиксом work_. Целесообразность использования технических таблиц будет рассмотрена экспертами при оценке.


dim_drivers
PK personnel_num
PK start_dt
last_name
first_name
middle_name
birth_dt
card_num
driver_license_num
driver_license_dt
deleted_flag
end_dt
fact_waybills
PK waybill_num
FK driver_pers_num
FK car_plate_num
work_start_dt
work_end_dt
issue_dt
dim_cars
PK plate_num
PK start_dt
model_name
revision_dt
deleted_flag
end_dt
fact_rides
PK ride_id
point_from_txt
point_to_txt
distance_val
price_amt
FK client_phone_num
FK driver_pers_num
FK car_plate_num
ride_arrival_dt
ride_start_dt
ride_end_dt
dim_clients
PK phone_num
PK start_dt
card_num
deleted_flag
end_dt
fact_payments
PK transaction_id
card_num
transaction_amt
transaction_dt

Фактовые таблицы:

- fact_rides
- fact_payments
- fact_waybills

Таблицы-измерения:

- dim_clients
- dim_cars
- dim_drivers

Заполнение фактовой таблицы fact_rides должно производиться только для завершенных поездок. То есть в источнике для поездки должен стоять либо статус END, либо статус CANCEL. Незавершенные поездки недопустимы. Правила заполнения полей следующие:

- ride_arrival_dt - время прибытия машины на заказ (статус READY),
- ride_start_dt - время начала заказа (статус BEGIN, если заказ был отменен - то NULL),
- ride_end_dt - время завершения или отмены заказа (статус END или CANCEL).

Во всех таблицах измерений должны быть корректно заполнены версии, то есть

- поля, определяющие ключ
- start_dt
- end_dt.

Знания и технологии, которые вам пригодятся.

Во-первых, рекомендуется почитать про хранилища данных. Начните отсюда - https://panoply.io/data-warehouse-guide/ , обратите внимание на понятие фактовых таблиц и таблиц измерений.

Вам нужно изучить два полезных понятия, используемых в хранилищах:

- форматы хранения таблиц измерений (SCD), рекомендуем формат SCD2. Начать можно с *английской* википедии https://en.wikipedia.org/wiki/Slowly_changing_dimension , но с точки зрения практического опыта там есть некорректные вещи, например, “перехлест” версий и null в поле end_dt.
- понятие об инкрементальной загрузке (не надо каждый раз выгружать весь источник, выгружайте только изменения).

Что от вас требуется?

- Сервер, который выполняет процессы загрузки - это ваш компьютер. Инструмент исполнения процессов - python.

- При этом целевой инструмент *для подключения и загрузки* данных - python. Целевой инструмент *для обработки и преобразования* данных - SQL.
- **Ваша разработка должна запускаться ежедневно, “захватывать” данные из источника и “укладывать” их в определенном формате в хранилище данных. Попутно можете проводить необходимые преобразования.**

Обратите внимание, что защищенными с точки зрения информационной безопасности считаются только СУБД источника, FTP сервер источника, если обмен данными с ним происходит через TLS, а также СУБД хранилища.

Критерии оценки.

К оцениванию проекта невозможно применить некую объективную шкалу оценки (например, 50 строк кода это лучше чем 20 строк кода, или пять таблиц в отчете лучше, чем три). Поэтому проект будет оцениваться экспертной оценкой по шести показателям. В качестве экспертов выступают менторы. Оценка выставляется аргументировано и может обсуждаться, но не изменяться.

У ментора есть право добавить дополнительные баллы за сложные решения в проекте (не сложное решение простой задачи, а именно решение сложной задачи), или вычесть баллы за грубые нарушения (например, хардкод).

Критерий	**Диапазон**	**Комментарий**
Структурированность кода	0..5	Оценивается читаемость кода (отступы, табуляции),чистота, комментирование, разделение на отдельные файлы логических блоков, как python, так и SQL.
Качество обработки инкремента	0..5	Оценивается насколько сильно вы нагружаете источник при ежедневной загрузке данных.
Общая сложность проекта	0..5	Оценивается каким образом вы разобрались с основными технологиями хранения данных в хранилищах данных и каким образом вы их реализовали. Также оценивается применение целевых инструментов для соответствующих задач.
Качество получаемого результата	0..5	Оценивается реализация всех перечисленных в задании требований. Оценивается качество данных, загруженных в таблицы (отсутствие дублей, корректность истории, соответствие источнику).

Безопасность данных	0..5	Проект оценивается с точки зрения информационной безопасности - есть ли потенциальные угрозы утечки данных.
Дополнительные баллы за сложные или плохие решения	-5..5	Эксперты обсудят ваш проект, и если встретятся оригинальные решения сложных задач - баллы будут добавлены. Если обнаружатся грубые ошибки или попытки обмануть систему - баллы будут сняты.