**Пояснения к критериям:**

- Бесплатный доступ: OpenStreetMap, Mapbox и Leaflet предлагают бесплатные решения, тогда как Google Maps и HERE Maps требуют оплаты за доступ к расширенным возможностям или количеству запросов.

- Высокое качество карт: Все сервисы предоставляют карты с высоким качеством и подробными данными, но Google Maps имеет преимущества в обновлении данных (например, с учетом пробок).

- API для разработчиков: Все сервиса предлагают API, с помощью которых можно интегрировать карты в веб-приложения. Однако, Google Maps API имеет плату за использование.

- Возможность кастомизации: Все перечисленные сервисы позволяют кастомизировать карту, добавлять собственные маркеры и элементы.

- Поддержка мобильных устройств: Все сервисы адаптированы для мобильных устройств и поддерживают Responsive Design.

- Режим offline - OpenStreetMap и другие решения, такие как Mapbox и Leaflet, могут работать в режиме offline, в то время как Google Maps требует постоянного подключения к интернету для отображения карт.

- Региональная поддержка: - сервисы поддерживают разные регионы и предоставляют данные на различных языках.

- Ограничения на использование - OpenStreetMap не устанавливает ограничения, в отличие от Google Maps и HERE Maps, которые имеют свои ограничения по количеству запросов и функционалу на бесплатных тарифах.

- Сообщество и поддержка - OpenStreetMap и Leaflet имеют активное сообщество, которое помогает пользователям находить ответы на вопросы. Google и HERE предлагают поддержку через свои учреждения.

- Мультиязычность- поддержка различных языков и культур во всех сервесах.

- Стоимость - OpenStreetMap, Leaflet и Mapbox предлагают бесплатные версии со всеми доступами, в то время как Google Maps и HERE Maps предлагают демо версию бесплатно с ужесточенными ограничениями, а остальное - платное.

Хорошо! Ниже — пример набора SQL-запросов (тестов), которые покрывают все основные таблицы из вашей схемы. Они проверят корректность вставки данных и основные связи между таблицами (через JOIN), а также выборку данных с простыми и сложными условиями. Это удобно использовать для проверки данных после загрузки или в нагрузочном тестировании.

**Тесты (SQL-запросы) для проверки всех таблиц и связей**

**1. Проверка таблицы Logging\_Areas и связанного Timber\_Enterprises**

-- Проверить, что все Logging\_Areas связаны с Timber\_Enterprises

SELECT la.ID\_logging, la.AreaName, te.Timber\_enterpriseName

FROM Logging\_Areas la

JOIN Timber\_Enterprises te ON la.ID\_timber\_enterprise = te.ID\_timber\_enterprise

LIMIT 10;

**2. Проверка таблицы Timber\_Species**

SELECT ID\_timber, TimberName, SpeciesName, Length, Diameter, Quality

FROM Timber\_Species

ORDER BY Quality DESC

LIMIT 10;

**3. Проверка таблицы Timber\_Enterprises**

SELECT ID\_timber\_enterprise, Timber\_enterpriseName

FROM Timber\_Enterprises

LIMIT 10;

**4. Проверка таблицы Products**

SELECT ID\_product, ProductName, Unit

FROM Products

LIMIT 10;

**5. Проверка таблицы Equipments**

SELECT ID\_equipment, EquipmentName, EquipmentCost

FROM Equipments

LIMIT 10;

**6. Проверка таблицы Employees**

SELECT ID\_Employee, EmployeeName, Tariff\_Rate

FROM Employees

LIMIT 10;

**7. Проверка таблицы Roads**

SELECT ID\_road, RoadLength, RoadCapacity, Max\_Speed

FROM Roads

LIMIT 10;

**8. Проверка таблицы Product\_Consumers**

SELECT ID\_consumer, ConsumerName

FROM Product\_Consumers

LIMIT 10;

**9. Проверка таблицы Regulations**

SELECT ID\_regulation, TaxName, Rate

FROM Regulations

LIMIT 10;

**10. Проверка таблицы Enterprises**

SELECT ID\_enterprise, EnterpriseName, Other\_Expenses

FROM Enterprises

LIMIT 10;

**11. Проверка таблицы Raw\_Material\_Needs**

SELECT rmn.ID\_need, e.EnterpriseName, p.ProductName, ts.TimberName, rmn.Raw\_Material\_Volume

FROM Raw\_Material\_Needs rmn

JOIN Enterprises e ON rmn.ID\_enterprise = e.ID\_enterprise

LEFT JOIN Products p ON rmn.ID\_product = p.ID\_product

LEFT JOIN Timber\_Species ts ON rmn.ID\_timber = ts.ID\_timber

LIMIT 10;

**12. Проверка таблицы Productions**

SELECT pr.ID\_production, e.EnterpriseName, p.ProductName, pr.Production\_Volume, pr.ProductCost

FROM Productions pr

JOIN Enterprises e ON pr.ID\_enterprise = e.ID\_enterprise

JOIN Products p ON pr.ID\_product = p.ID\_product

LIMIT 10;

**13. Проверка таблицы Raw\_Material\_Supply**

SELECT rms.ID\_supply, e1.EnterpriseName AS Receiver, e2.EnterpriseName AS Supplier, ts.TimberName, la.AreaName, p.ProductName, rms.Material\_Volume, rms.Material\_Cost

FROM Raw\_Material\_Supply rms

JOIN Enterprises e1 ON rms.ID\_enterprise = e1.ID\_enterprise

JOIN Enterprises e2 ON rms.ID\_enterprise\_resource = e2.ID\_enterprise

LEFT JOIN Timber\_Species ts ON rms.ID\_timber = ts.ID\_timber

LEFT JOIN Logging\_Areas la ON rms.ID\_logging = la.ID\_logging

LEFT JOIN Products p ON rms.ID\_product = p.ID\_product

LIMIT 10;

**14. Проверка таблицы Product\_Delivery**

SELECT pd.ID\_delivery, e.EnterpriseName, p.ProductName, pc.ConsumerName, pd.Delivery\_Volume, pd.Delivery\_Cost

FROM Product\_Delivery pd

JOIN Enterprises e ON pd.ID\_enterprise = e.ID\_enterprise

JOIN Products p ON pd.ID\_product = p.ID\_product

JOIN Product\_Consumers pc ON pd.ID\_consumer = pc.ID\_consumer

LIMIT 10;

**15. Проверка таблицы Transportation\_Routes**

SELECT ID\_route, Distance, Travel\_Time

FROM Transportation\_Routes

LIMIT 10;

**16. Проверка таблицы Employee\_Enterprise**

SELECT ee.ID\_Employee\_Enterprise, e.EnterpriseName, emp.EmployeeName, ee.NCount, ee.Salary

FROM Employee\_Enterprise ee

JOIN Enterprises e ON ee.ID\_enterprise = e.ID\_enterprise

JOIN Employees emp ON ee.ID\_employee = emp.ID\_Employee

LIMIT 10;

**17. Проверка таблицы Enterprise\_Equipment**

SELECT eeq.ID\_Equipment\_Enterprise, e.EnterpriseName, eq.EquipmentName, eeq.NCount, eeq.Depreciation, eeq.Energy, eeq.Other\_Resource\_Sum, eeq.EquipmentCost

FROM Enterprise\_Equipment eeq

JOIN Enterprises e ON eeq.ID\_enterprise = e.ID\_enterprise

JOIN Equipments eq ON eeq.ID\_equipment = eq.ID\_equipment

LIMIT 10;

**18. Проверка таблицы Route\_Roads**

SELECT rr.ID\_route\_road, tr.ID\_route, r.ID\_road, rr.Road\_Order

FROM Route\_Roads rr

JOIN Transportation\_Routes tr ON rr.ID\_route = tr.ID\_route

JOIN Roads r ON rr.ID\_road = r.ID\_road

ORDER BY rr.ID\_route\_road

LIMIT 10;

**19. Проверка таблицы Carriers**

SELECT ID\_carrier, CarrierName

FROM Carriers

LIMIT 10;

**20. Проверка таблицы Tariffs**

SELECT t.ID\_tariff, c.CarrierName, t.Distance, t.Unit, t.TariffCost

FROM Tariffs t

JOIN Carriers c ON t.ID\_carrier = c.ID\_carrier

LIMIT 10;

**Как использовать**

* Выполните каждый запрос после загрузки данных.
* Убедитесь, что данные корректно связаны, нет NULL в ключевых полях.
* Замерьте время выполнения для оценки производительности.

Если нужно — могу помочь с генерацией тестовых данных для этих таблиц или с созданием автоматизированных тестов на Python (например, с использованием pytest + sqlalchemy).

Напишите, если интересно!