

BikeStore: SQL-Powered Retail Business Intelligence

Texnik Topshiriq va Loyiha Hujjatnomasi

1. Loyihaning Umumiy Maqsadi (Business Goal)

Ushbu loyihaning asosiy maqsadi **BikeStore** chakana savdo tarmog'ining CSV formatidagi xom ma'lumotlarini **SQL Server** muhitiga integratsiya qilish, ularni tozalash va normalizatsiyalangan BI (Business Intelligence) qatlamini yaratishdir.

Asosiy vazifalar:

- Ma'lumotlar Ombori:** Xom ma'lumotlarni ishonchli SQL strukturasiga o'tkazish.
- Analitik Hisobotlar (Views):** Biznes qarorlar qabul qilish uchun tayyor ko'rinishlar.
- Avtomatlashtirilgan Tahlil (Stored Procedures):** Parametrli va qayta ishlatiladigan hisob-kitob algoritmlari.
- KPI Monitoring:** Kompaniya samaradorligini o'lchovchi ko'rsatkichlar.
- Operatsion Avtomatlashtirish:** SQL Agent Job orqali kundalik ma'lumotlarni yangilash.

2. Kiruvchi Ma'lumotlar (Data Inputs)

Loyiha quyidagi 9 ta asosiy CSV fayllar asosida quriladi:

Fayl nomi	Tavsifi
products.csv	Mahsulotlar ro'yxati (velosipedlar, aksessuarlar)
brands.csv	Mahsulot brendlari
categories.csv	Mahsulot kategoriyalari
customers.csv	Mijozlar haqidagi ma'lumotlar
stores.csv	Filiallar (do'konlar) joylashuvi
staffs.csv	Xodimlar va boshqaruv iyerarxiyasi
orders.csv	Buyurtmalar haqida umumiy ma'lumot
order_items.csv	Buyurtma tarkibidagi mahsulotlar va narxlar
stocks.csv	Ombor qoldiqlari (har bir do'kon kesimida)

3. Tizim Arxitekturası (3-Qatlamli Model)

Ma'lumotlar oqimi quyidagi uchta mantiqiy qatlam orqali o'tadi:

1. Staging Layer (stg_...):

- Xom ma'lumotlarni CSV'dan 1:1 ko'chiradi.
- Cheklovlar (constraints) minimal darajada bo'ladi.
- Ma'lumotlar turi asosan VARCHAR formatida saqlanadi.

2. Core Layer (dbo yoki core):

- Tozalangan va normalizatsiyalangan ma'lumotlar.
- Primary Key (PK) va Foreign Key (FK) bog'liqliklari to'liq o'rnatilgan.
- Ma'lumot turlari optimallashtirilgan (Date, Decimal, Int).

3. Reporting Layer (vw_... , rpt_...):

- Biznes foydalanuvchilar uchun tushunarli View'lar.
- KPI jadvallari va Stored Procedure natijalari.

4. Loyiha Tuzilmasi (Folder Structure)

Dasturchi uchun qulaylik yaratish maqsadida loyiha quyidagi papka iyerarxiyasida tashkil etilishi lozim:

```
C:\BikeStore_BI\  
├─ data\                # Barcha CSV fayllar  
├─ sql\                 # SQL skriptlar (tartib bilan)  
│   ├── 01_setup.sql  
│   ├── 02_staging.sql  
│   ├── 03_etl_load.sql  
│   ├── 04_core_schema.sql  
│   ├── 05_views.sql  
│   ├── 06_procedures.sql  
│   └─ 07_automation.sql  
├─ screenshots\        # Diagrammalar va natijalar  
└─ docs\                # Hujjatlar (.pdf, .docx)
```

5. Bosqichma-bosqich Amalga Oshirish Rejasi

STEP 0: Muhitni Tayyorlash (Setup)

- **Vazifa:** SQL Serverda `BikeStore` ma'lumotlar bazasini yaratish va papkalar tizimini sozlash.
- **Natija:** SSMS orqali bazaga ulanish imkoniyati mavjud.

STEP 1: Ma'lumotlarni Tahlil Qilish (Data Understanding)

- **Vazifa:** Har bir CSV ustunlarini va ularning o'zaro bog'liqligini aniqlash.
- **Bog'lanishlar:**
 - `orders.customer_id` → `customers.customer_id`
 - `order_items.product_id` → `products.product_id`
 - `staffs.manager_id` → `staffs.staff_id` (Self-reference)

STEP 2: Core Sxema Dizayni

- **Vazifa:** Final jadvallarni yaratish. Ma'lumot turlarini aniq belgilash (`Decimal(10,2)` , `BIT` , `DATE`).
- **Natija:** Database Diagram'da barcha jadvallar PK/FK orqali bog'langan bo'ladi.

STEP 3: Staging va Ma'lumotlarni Yuklash (ETL)

- **Vazifa:** `BULK INSERT` yordamida ma'lumotlarni staging jadvallariga o'tkazish.
- **Tozalash:** `TRY_CONVERT()` yordamida xatolarni tekshirish va dublikatlarni o'chirish (`ROW_NUMBER()`).

6. Analitik Qatlam (BI Layer)

6.1. Reusable Views (Kamida 6 ta)

View Nomi	Maqsadi
<code>vw_StoreSalesSummary</code>	Do'konlar kesimida tushum va buyurtmalar soni.
<code>vw_TopSellingProducts</code>	Eng ko'p sotilayotgan mahsulotlar reytingi.
<code>vw_InventoryStatus</code>	Qoldig'i kamayib qolgan mahsulotlar (Stock Alert).
<code>vw_StaffPerformance</code>	Xodimlarning sotuv samaradorligi.
<code>vw_RegionalTrends</code>	Shaharlar bo'yicha sotuv dinamikasi.
<code>vw_SalesByCategory</code>	Kategoriyalar bo'yicha foyda va tushum.

6.2. Stored Procedures (Kamida 4 ta)

1. `sp_CalculateStoreKPI` : Muayyan do'kon va vaqt oralig'i uchun KPI hisoblash.
2. `sp_GenerateRestockList` : Omborlarni to'ldirish uchun zarur mahsulotlar ro'yxati.
3. `sp_CompareSalesYearOverYear` : Yillik sotuv o'sishini solishtirish.
4. `sp_GetCustomerProfile` : Mijozning sotib olish tarixi va portreti.

7. KPI Ko'rsatkichlari va Formularlar

KPI Nomi	Formula / Izoh	Biznes Ahamiyati
Total Revenue	$\Sigma (Qty * Price * (1 - Discount))$	Umumiy moliyaviy holat.
AOV	Total Revenue / Total Orders	Mijozning o'rtacha xarid hajmi.
Inventory Turnover	Sotilgan mahsulotlar / O'rtacha qoldiq	Ombor samaradorligi.
Gross Profit	Revenue - Cost (mavjud bo'lsa)	Sof daromad tahlili.

8. Avtomatlashtirish (SQL Server Agent)

Ma'lumotlar yangiligini ta'minlash uchun quyidagi **Job Flow** sozlanadi:

- Step 1:** Staging jadvallarini tozalash (TRUNCATE).
- Step 2:** Yangi CSV fayllarni yuklash (BULK INSERT).
- Step 3:** Ma'lumotlarni transformatsiya qilib Core qatlamga yuklash.
- Step 4:** Analitik protseduralarni ishga tushirish.
- Step 5:** Natijalarni audit jadvallariga yozish.

9. Topshirish Paketi (Deliverables)

Loyiha yakunida quyidagilar taqdim etilishi shart:

- final_submission.sql** : Barcha bazaviy obyektlar va skriptlar bitta faylda.
- Screenshots:** DB Diagramma, Job History va natijalar namunalari.
- Hujjatlar:** Ushbu texnik topshiriqning to'liq PDF va Word versiyasi.
- Taqdimot:** Loyihaning 5-7 slayddan iborat prezentatsiyasi.

10. Sifat Nazorati (Quality Checklist)

- ☐ Barcha jadvallarda PK/FK bog'liqliklari mavjud.
- ☐ Staging import jarayonida xatoliklar yo'q.
- ☐ View'lar va Protseduralar dinamik (hardcode yo'q).
- ☐ SQL Agent Job muvaffaqiyatli ishlayapti.
- ☐ Hujjat tushunarli va ketma-ketlikka asoslangan.