

From Raw Data to Decision-Ready Dashboards

Kompleksowy projekt analityczny.

Od przygotowania danych do dashboardów decyzyjnych

Python → SQL → Excel/ Power BI

3 relacyjne tabele, ponad 15 tys. rekordów

Projekt prezentuje pełny proces pracy analityka danych – od przygotowania i modelowania danych, przez analizy, aż po wizualizację wyników.

Bartosz Ceranek | Data Analyst | GitHub: <https://github.com/bceranek>

Executive summary

Projekt powstał jako praktyczne studium przypadku pokazujące **pełny proces pracy analityka danych**. Jego celem było przejście przez wszystkie etapy analizy – od **przygotowania danych**, przez ich **przetwarzanie i analizę**, aż po stworzenie **czytelnych dashboardów decyzyjnych**.

Całość została zaprojektowana w sposób możliwie zbliżony do realnych projektów biznesowych. Dane są **uporządkowane, logicznie połączone** i przygotowane tak, aby mogły być dalej wykorzystywane zarówno w narzędziach BI, jak i w klasycznych raportach.

Projekt skupia się na praktycznym zastosowaniu **analizy danych w obszarze sprzedaży i produktów**. Wyniki zostały przedstawione w formie interaktywnych **dashboardów Power BI**, które umożliwiają szybkie zrozumienie sytuacji oraz identyfikację obszarów wymagających działania.

Zakres projektu i rezultaty

Zakres danych:

- Dane sprzedażowe za pełny rok wygenerowane przez autorski skrypt w Pythonie
- **5 000 unikalnych fv**, każda ma do 5 pozycji
- **20 produktów** podzielone na osobne kategorie i podkategorie
- **20 przedstawicieli handlowych** przypisanych do krajów i dywizji
- **3 rynki**: Polska, Niemcy, Czechy

Rezultaty projektu:

- Biznesowa struktura danych przygotowana pod analizę i raportowanie
- **3 interaktywne dashboardy Power BI**
- (Sprzedaż, Przedstawiciele Handlowi, Produkty)
- Model danych typu **star schema**
- **4 zapytania SQL** przygotowane pod raportowanie w Excelu
- (alternatywa dla Power BI)

Kluczowe pytania biznesowe analizowane w projekcie:

- Którzy przedstawiciele generują najwyższą marżę?
- Które produkty są rentowne, a które budują wyłącznie wolumen?
- Jak kształtuje się sprzedaż w przedziale czasowym i geograficznym?
- Gdzie pojawiają się obszary wymagające uwagi?

Architektura projektu

Rola Pythona w projekcie

Kluczowym elementem projektu jest autorskie **przygotowanie danych w Pythonie**, które stanowi punkt wyjścia dla całej dalszej analizy.

Zamiast gotowych zbiorów, dane sprzedażowe zostały **wygenerowane w sposób kontrolowany i spójny**, tak aby jak najwierniej odzwierciedlały rzeczywiste środowisko biznesowe.

Skrypt w Pythonie odpowiada za:

- generowanie faktur sprzedażowych wraz z **wielopozycyjną strukturą dokumentów**,
- tworzenie powiązanych tabel (sprzedaż, produkty, przedstawiciele handlowi),
- zachowanie **spójności relacyjnej** między danymi (klucze, zakresy, zależności),
- nadanie danym realistycznych cech biznesowych (czas, rynki, kategorie, dywizje).

Python



Baza danych



Power BI (model + DAX + dashboardy)



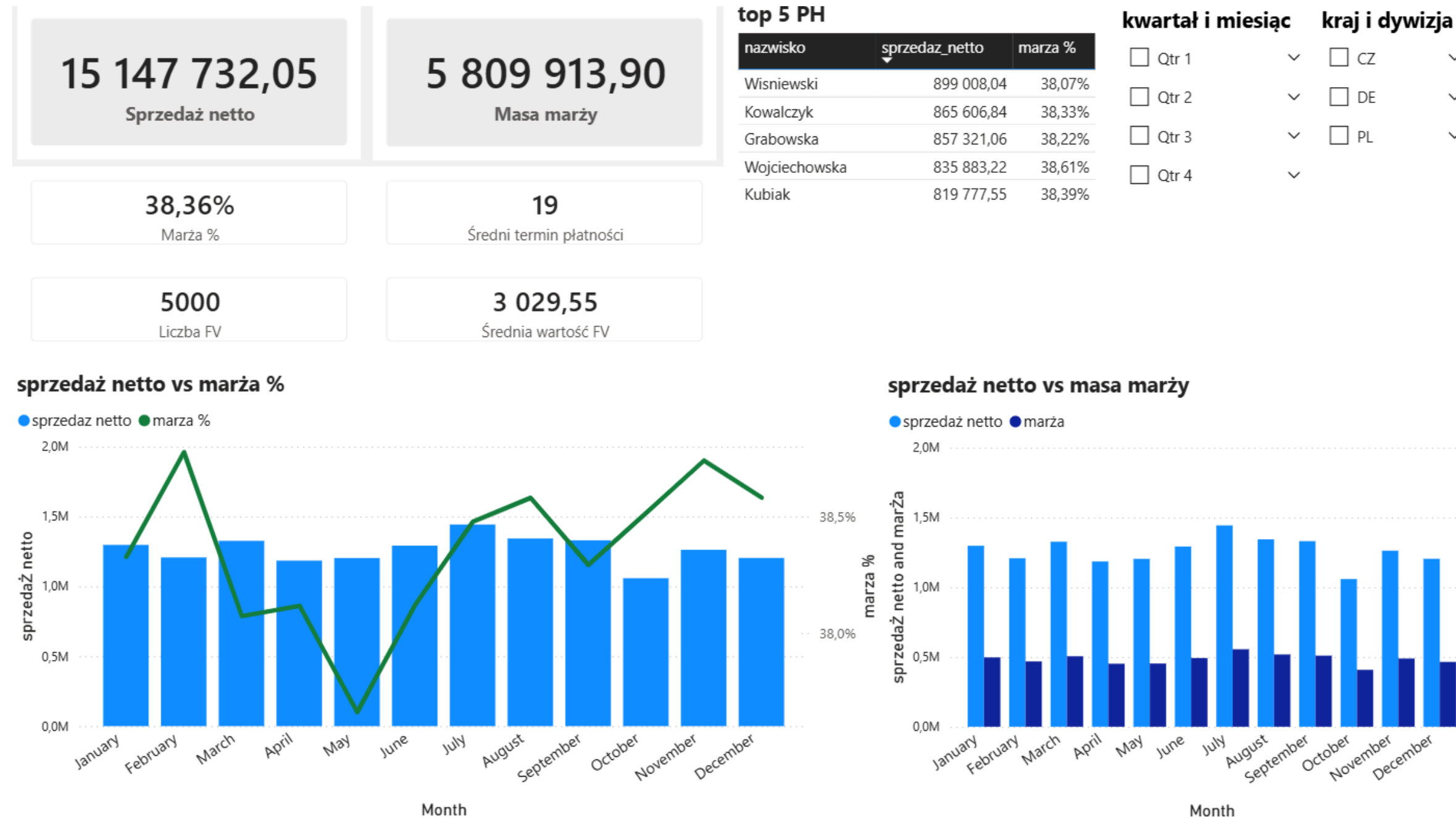
SQL (zapytania raportowe)



Excel

```
14 def build_tabela_produkt(df_dane_produkt):|
15
16     df_tabela_produkt = df_dane_produkt.copy()
17
18     # kategoria i podkategoria
19
20     df_tabela_produkt["s1"] = df_tabela_produkt["nazwa_produktu"].str.split().str[0]
21     unikalne_kategorie = df_tabela_produkt["s1"].unique()
22     map_kategoria = {kat: str(i+1).zfill(2) for i, kat in enumerate(unikalne_kategorie)}
23     #enumerate nadaje index do kazdego elementu listy, zfill(2) dodaje 0 na poczatku jesli jest potrzeba
24
25     df_tabela_produkt["kategoria"] = df_tabela_produkt["s1"].map(map_kategoria)
26
27     df_tabela_produkt["s2"] = df_tabela_produkt["nazwa_produktu"].str.split().str[0] +
28     "_" + df_tabela_produkt["nazwa_produktu"].str.split().str[1]
29     unikalne_podkategorie = df_tabela_produkt["s2"].unique()
30     map_podkategoria = {pod: str(i + 1).zfill(2) for i, pod in enumerate(unikalne_podkategorie)}
31
32     df_tabela_produkt["podkategoria"] = df_tabela_produkt["s2"].map(map_podkategoria)
33
```

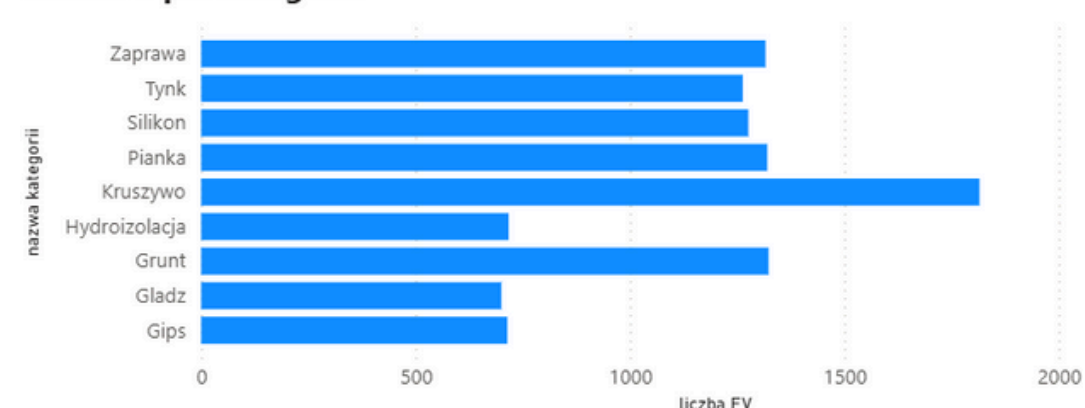
Power BI – dashboard ogólny (KPI i trendy)



W Power BI dane z bazy zostały przekształcone w **3 interaktywne dashboardy** wspierające codzienne decyzje biznesowe. Dashboardy pozwalają szybko przejść od ogólnych **KPI do analizy** wyników według rynku, **przedstawiciela handlowego i produktu**.

Power BI – widoki detaliczne (PH i produkty)

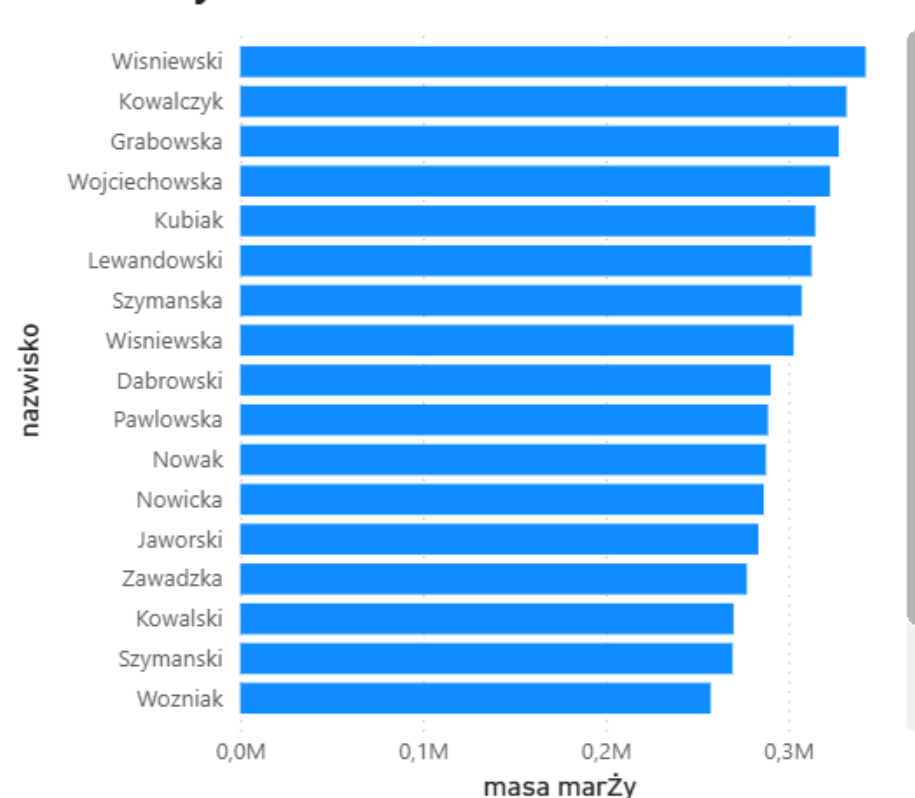
liczba FV per kategoria



top 5 – najwyższa masa marży

nazwa produktu	sprzedaż	marża %	marża
Tynk silikonowy elewacyjny	2 566 219,80	46,84%	1 202 051,72
Hydroizolacja dwuskładnikowa	2 984 085,65	33,91%	1 012 027,51
Tynk akrylowy elewacyjny	2 142 095,68	35,17%	753 429,28
Zaprawa tynkarska wewnętrzna	617 468,45	46,81%	289 060,94
Gładz szpachlowa finiszowa	870 493,00	30,66%	266 916,00

masa marży



Kluczowe obszary analityczne raportu:

- **Monitorowanie KPI:** sprzedaż netto, masa marży i marża % w ujęciu czasowym i regionalnym
- Analiza trendów: **porównanie sprzedaży i rentowności** w czasie, wykrywanie odchyłeń
- **Wyniki PH:** rankingi i porównania efektywności handlowców na rynkach PL/DE/CZ
- Produkty: identyfikacja produktów rentownych oraz tych budujących głównie wolumen

Warstwa techniczna raportu (Power BI):

- **Power Query:** przygotowanie i transformacja danych (typy danych, czyszczenie, kolumny pomocnicze)
- Model danych: układ gwiazdy (**star schema**) z tabelą faktów i wymiarami
- **DAX: miary** do obliczeń rentowności, rankingów oraz średnich wartości dokumentów
- Interaktywność: **filtry**, **drill-through** oraz zależności między wizualizacjami

Rola SQL w projekcie

Zapytania SQL zostały napisane jako niezależna warstwa raportowa, niezależna od Power BI. Na ich podstawie powstały **widoki agregujące dane** sprzedażowe, **wykorzystywane** bezpośrednio do raportów w Excelu.

Widoki obejmują m.in.:

- **agregacje** sprzedaży i marży w ujęciu czasowym,
- **podział** według **rynku**, dywizji i kanału sprzedaży,
- zestawienia produktowe (**Top / Bottom**).

```
2 with poziom_fv as(  
3     SELECT  
4         s.ph_id,  
5         s.sales_id,  
6         CASE  
7             WHEN strftime('%m', s.data_sprzedazy) <= '03'  
8                 THEN date(strftime('%Y', s.data_sprzedazy) || '-01-01')  
9             WHEN strftime('%m', s.data_sprzedazy) <= '06'  
10                THEN date(strftime('%Y', s.data_sprzedazy) || '-04-01')  
11             WHEN strftime('%m', s.data_sprzedazy) <= '09'  
12                THEN date(strftime('%Y', s.data_sprzedazy) || '-07-01')  
13             ELSE date(strftime('%Y', s.data_sprzedazy) || '-10-01')  
14         END AS kwartal,  
15         min(s.klient_id) as klient_id,  
16         round(sum(s.wartosc_netto), 2) as wartosc_netto_fv,  
17         round(sum(s.koszt_wlasny), 2) as koszt_wlasny_fv,  
18         COUNT(*) as ilosc_pozycji  
19     FROM tabela_sprzedaz s  
20     GROUP BY s.ph_id, s.sales_id, kwartal  
21 ),  
22  
23 +poziom_ph as(  
36  
37     SELECT  
38         ph_id,  
39         ph.imie || ' ' || ph.nazwisko as imie_nazwisko,  
40         kwartal,  
41         suma_wartosci_netto,  
42         suma_kosztow_wlasnych,  
43         ilosc_fv,  
44         sr_wartosc_fv,  
45         sr_ilosc_poz_na_fv,  
46         liczba_klientow  
47     FROM poziom_ph  
48     JOIN tabela_ph ph using(ph_id)  
49     ORDER by ph.ph id, kwartal
```


Excel – raportowanie oparte na SQL

Raporty w Excelu zostały przygotowane bezpośrednio **na podstawie zapytań SQL**, bez dodatkowych przeliczeń w arkuszu.

Zastosowanie **tabel przestawnych i prostych rankingów** umożliwia szybką analizę wyników sprzedaży i marży na poziomie przedstawiciela handlowego.

Excel pełni tu **rolę lekkiej, operacyjnej alternatywy** dla **Power BI** w środowiskach, gdzie raportowanie odbywa się głównie w arkuszach.

[illegible]

Podsumowanie

Projekt pokazuje **pełny proces** pracy analitycznej — od **przygotowania** i **modelowania** danych, przez **analizę**, aż po **raportowanie** i **wizualizację** wyników.

Dane zostały zaprojektowane w **Pythonie i SQL**, a następnie wykorzystane równolegle w **Power BI oraz Excelu**, bez powielania logiki obliczeń. Takie podejście odzwierciedla sposób pracy spotykany w **praktyce biznesowej**.

Kod źródłowy i dokumentacja:

GitHub: <https://github.com/bceranek>