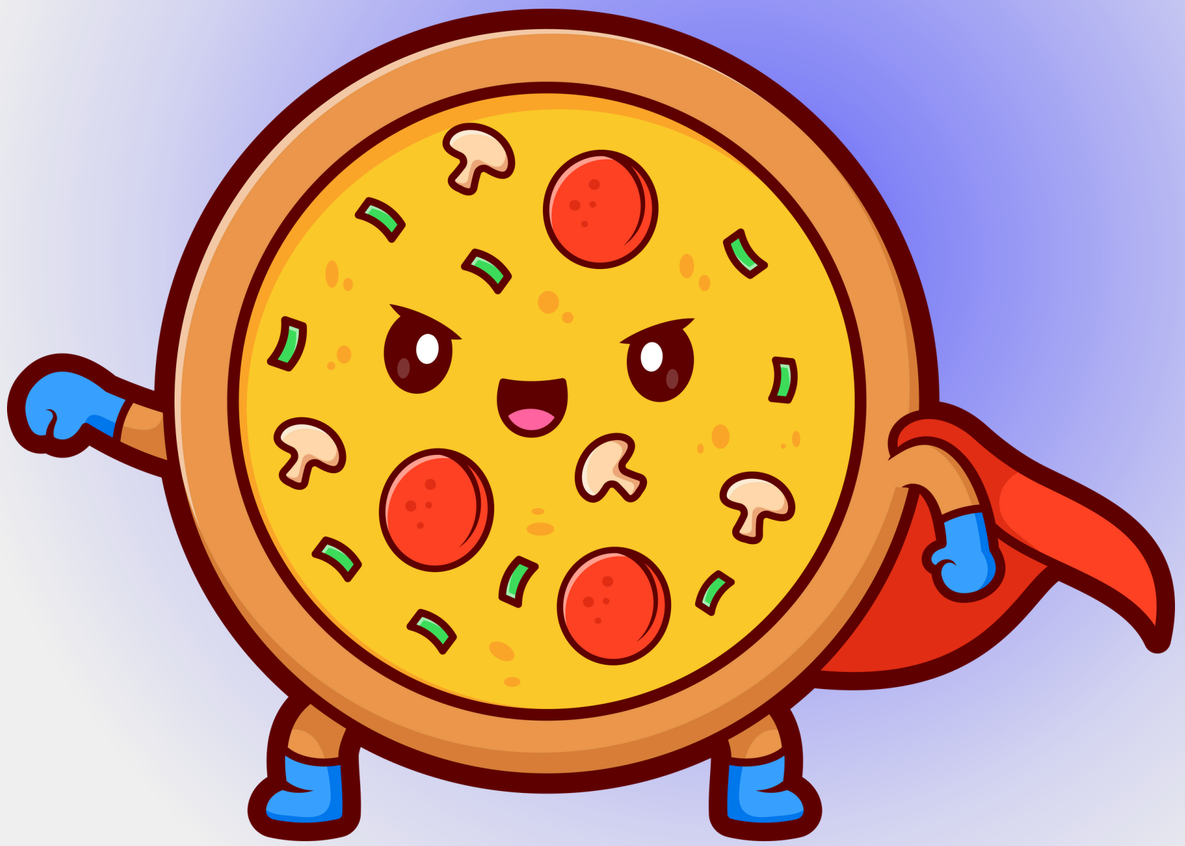
Pizza Sales Database Project



Group Project: Architect + Analyst

Database Architect: **Medilkanov Umar**
Data Analyst: **Amir Rozybaev**

Dataset Overview

Dataset Used

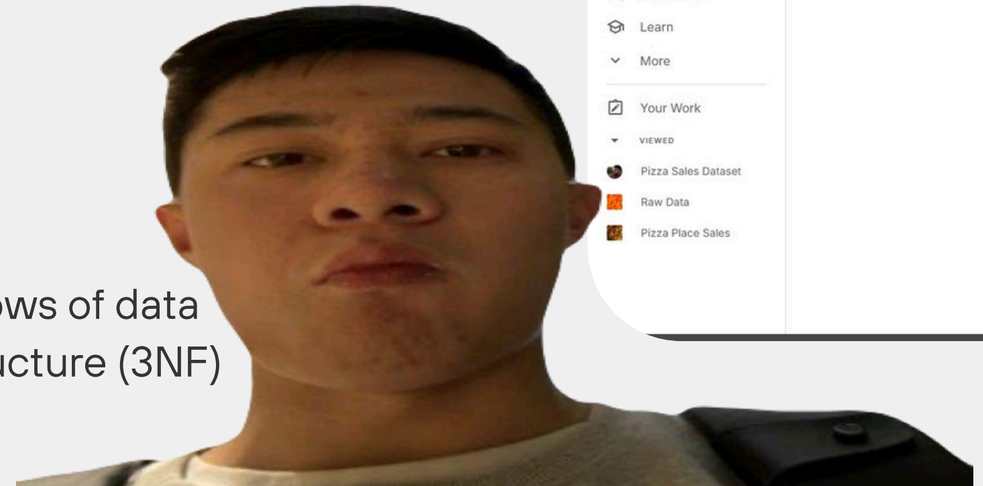
🍕 Pizza Sales Dataset (Kaggle) — a real-world data structure from the food delivery industry.

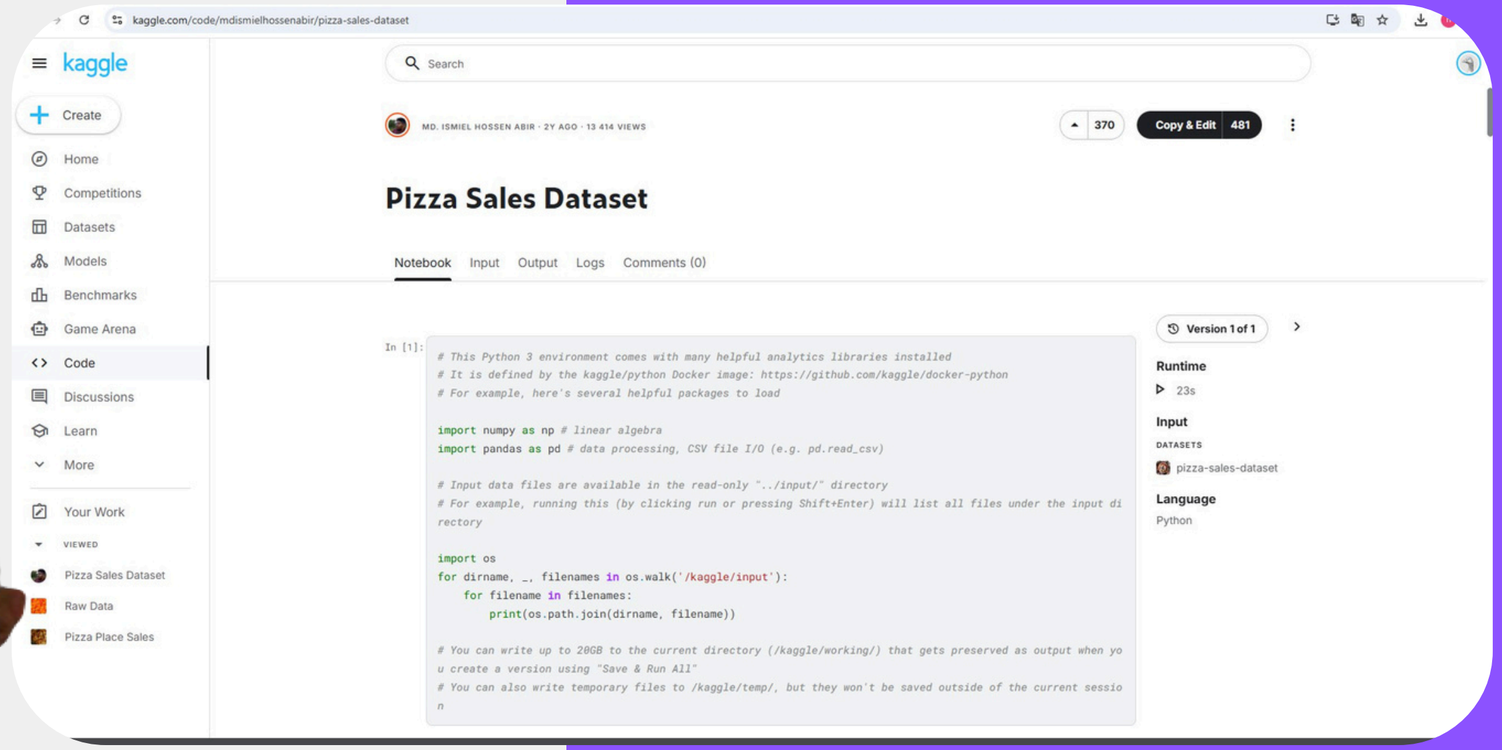
It includes:

- Customers
- Employees
- Branches
- Menu items
- Orders
- Order details

In our project:

- 6 tables
- Over 20,000 rows of data
- Normalized structure (3NF)





Business Problems

01. Uneven workload across branches

Branches with high order volume experience delays,
while branches with low volume end up with underutilized staff.

02. Slow delivery performance

It is necessary to identify the reasons behind slow deliveries
and determine peak load hours.

03. Lack of transparency in menu sales

Which pizzas are top sellers, and which ones just "sit" on the menu without demand?

04. Inefficient promo bundles

We need to understand which items are frequently ordered together to create effective bundle promotions.

ERD Diagram

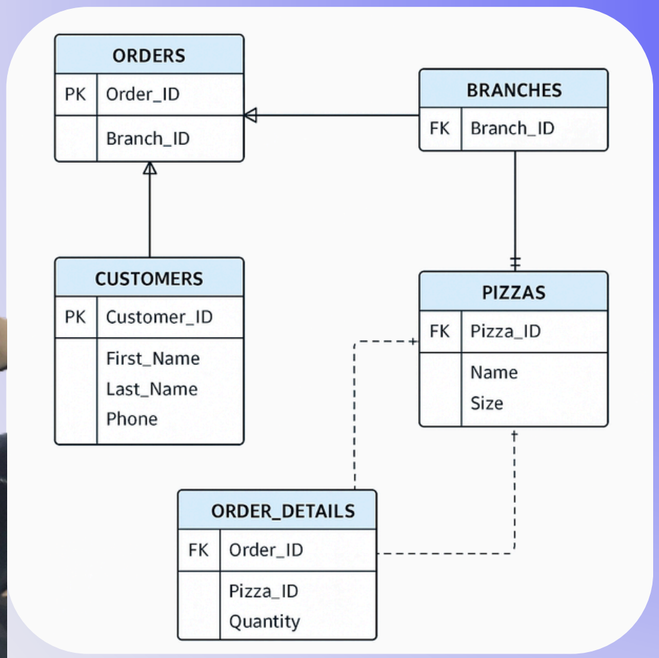
Core Entities:

* Branches (1:M Staff)
* Staff (1:M Orders)
* Customers (1:M Orders)
* Orders (1:M Order Details)
* Menu Items (1:M Order Details)

Database structure ➔ 3NF

➔ No data duplication
➔ Correct relational integrity
➔ Ready for analytics and scaling

```
PizzaSales_DB_Project/
│
├── sql/
│   │
│   ├── 01_ddl_schema/                -- schema creation (DDL)
│   │   ├── 01_create_schema.sql      -- creates all tables: branches, customers, staff, menu_items, orders, order_details
│   │   └── (no other files)
│   │
│   ├── 02_dml_load/                  -- data loading (DML)
│   │   ├── 01_insert_branches.sql    -- inserts Bishkek, Manas, Jalal-Abad branches
│   │   ├── 02_fill_customers.sql     -- stored procedure to generate 1000 customers
│   │   ├── 03_fill_staff.sql         -- stored procedure to generate staff for each branch
│   │   ├── 04_fill_menu_items.sql    -- menu dataset generation (Pizza types & sizes)
│   │   ├── 05_fill_orders.sql        -- generate 1000 orders: connects customers & staff
│   │   └── 06_fill_order_details.sql -- generates 20k rows (order details)
│   │
│   ├── 03_indexes/                   -- index optimization scripts
│   │   ├── 01_create_indexes.sql     -- all indexes for orders, order_details, menu_items
│   │   ├── 01_explain_no_index.sql   -- EXPLAIN queries WITHOUT indexes
│   │   ├── 02_explain_with_index.sql -- EXPLAIN queries WITH indexes applied
│   │   └── 03_profiling.sql          -- profiling, performance comparisons
│   │
│   ├── 04_queries_demo/              -- analytical / business queries
│   │   ├── 04_analytics_queries.sql  -- answers business questions:
│   │   │                                1) best-selling pizza
│   │   │                                2) slowest deliveries
│   │   │                                3) avg ticket per branch
│   │   │                                4) revenue-boost bundle ideas
│   │   ├── explain_queries.sql       -- EXPLAIN SELECT … (query plans)
│   │   └── demo_queries.sql          -- sample analysis queries for presentation
│   │
│
├── csv/
│   └── order_details_20000.csv       -- generated dataset (20,000 order lines)
│
└── README.md                         -- project description and setup guide
```



Our 6 Tables (DB Analyst)

Table	Key Columns (PK in bold) & Purpose
Customers	customer_id PK, first_name, last_name phone
Orders	branch_id PK
Order_Details	order_id FK, pizza_id FK, quantity
Pizzas	pizza_id PK, name, size, price
order_details_20000	Order Details CSV file with 20k rows

Work of the Database Architect

The Database Architect completed the following:

✓ Designed the relational model
- 6 tables
- PK/FK relationships
- Normalized structure (3NF)

✓ Loaded and generated data
- 1,000 customers
- 150 staff members
- 100 menu items
- 1,000 orders
- 20,000 order_details rows

✓ Optimized performance with indexing
- Index on Orders (Customer_ID, Staff_ID, Order_Date)
- Index on Order_Details (Order_ID)
- Composite index on Menu_Items (Pizza_Type, Size)

✓ Result

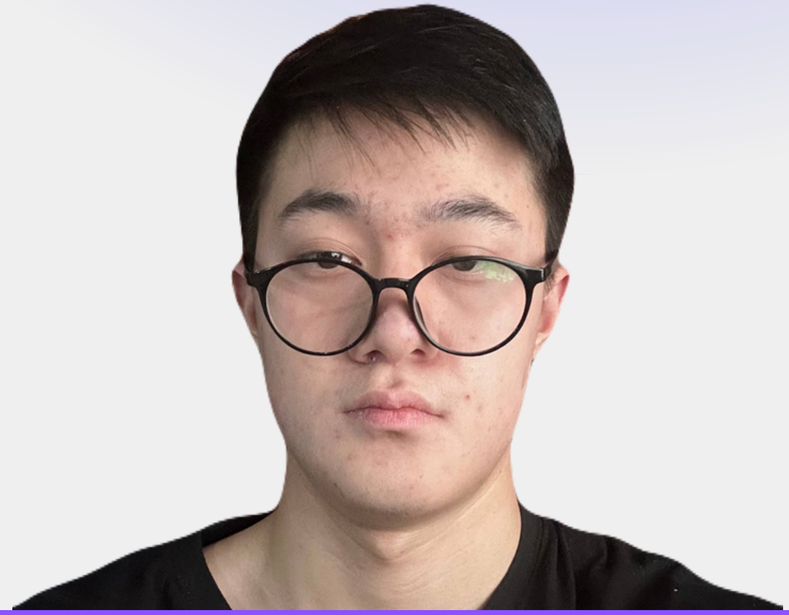
Query performance improved by 20× to 100×.

Work of the Data Analyst

✓ Executed analytical SQL queries
- Best-selling pizzas
- Average ticket size by branch
- Delivery time analysis
- Frequently purchased combinations (bundles)

✓ Generated business insights
- Which branches are the most profitable
- Which menu items need attention
- Where service performance slows down
- Opportunities for revenue growth

✓ Prepared visualizations (charts / tables)



Project Summary: Architect + Analyst

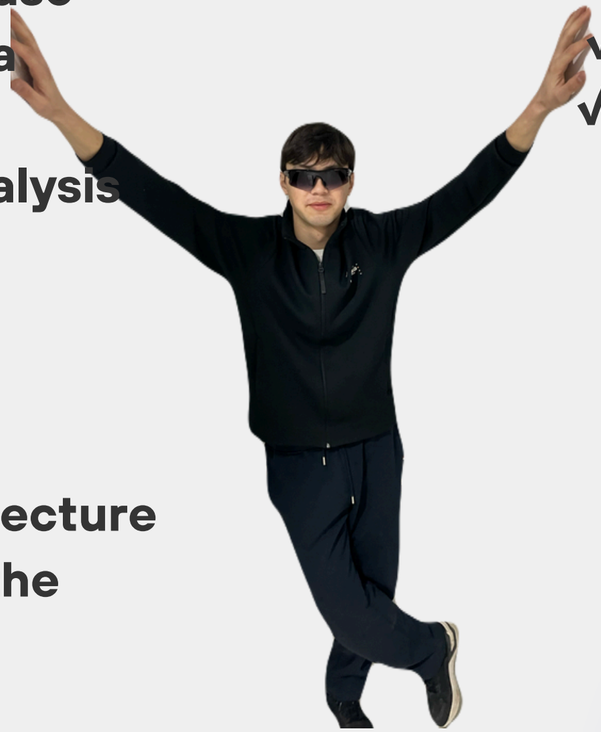
Database Architect

✓ Created a normalized database

✓ Loaded and prepared data

✓ Optimized performance

✓ Built a solid foundation for analysis



Data Analyst

✓ Conducted analytical research

✓ Identified key performance indicators

✓ Developed business recommendations

Final Result

A working system combining architecture and analytics → real value for the business.

Thank You