UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

22125005 - MAI XUAN BACH

CS423 - Software Testing

HOMEWORK

DATA GENERATION

HCMC – Nov, 2025

fit@hcmus

UNIVERSITY OF SCIENCE

FACULTY OF INFORMATION TECHNOLOGY

Table of contents

Table of contents 2

1. Introduction 3

2. Data Categories and Randomized Ranges 3

2.1. Product Categories 3

id (Primary Key) 3

name 3

slug 3

parent_id (Foreign Key) 4

2.2. Products 4

id (Primary Key) 4

name 4

price 4

description 4

stock 4

category_id (Foreign Key) 5

brand_id (Foreign Key) 5

product_image_id (Foreign Key) 5

co2_rating 5

is_rental 5

is_location_offer 5

2.3. User Accounts 5

id (Primary Key) 5

first_name 5

last_name 5

email 6

password 6

address 6

city 6

state 6

country 6

postcode 6
phone 6
dob (Date of Birth) 6
role 7
enabled 7
2.4. Transactions 7
id (Primary Key) 7
user_id (Foreign Key) 7
invoice_date 7
invoice_number 7
billing_address 7
billing_city 7
billing_state 8
billing_country 8
billing_postcode 8
status 8
total 8
payment_method 8
payment_account_name 8
payment_account_number 8
3. How the Tool Works 9
3.1. Tool Overview 9
3.2. Setup and Installation 9
3.3. Running the Tool 9
3.4 Screenshot - Tool Execution: 10
4. How Tool Works 10
5. Sample Data 12
5.1. Product Categories 12
5.2. Products 13
5.3. User Accounts 14
5.4 Transactions 14
6. Self-Assessment Report 15

fit@hcmus

UNIVERSITY OF SCIENCE

FACULTY OF INFORMATION TECHNOLOGY

1. Introduction

In this assignment, I have developed a custom-built Python tool to generate test data for the Toolshop e-commerce application. This report details the configuration and execution steps taken to generate:

- 50 product categories
- 1000 products
- 50 user accounts
- 1000 transactions

A critical requirement for this task is the quality and realism of the data. The generation process has been strictly configured to ensure all field values are meaningful and contextually accurate, not just random strings.

Tool Used: Custom Python script with the Faker library for realistic data generation.

2. Data Categories and Randomized Ranges

This section details the specific rules, data types, and ranges applied in the data generator

2.1. Product Categories

id (Primary Key)

- Type: Auto-increment Integer
- Range: 1 to 50

name

- Type: String
- Logic: Predefined list of realistic category names
- Examples: Hand Tools, Power Tools, Plumbing, Hammers, Drills, Wrenches

slug

- Type: String (URL-friendly)
- Logic: Derived from the name field. Converts to lowercase and replaces spaces with hyphens
- Result: "Hand Tools" becomes "hand-tools"

parent_id (Foreign Key)

- Type: Integer or NULL
- Logic: To simulate hierarchical category structure:
    - For categories with ID from 1 to 10: `parent_id = NULL` (root categories)
    - For categories with ID from 11 to 50: `parent_id = random integer between 1 and 10` (subcategories)

2.2. Products

id (Primary Key)

- Type: Auto-increment Integer
- Range: 1 to 1000

name

- Type: String
- Logic: Generated using templates combined with word lists (adjectives, tools, brands, sizes)
- Templates:
    - "Professional {adjective} {tool}"
    - "{brand} {tool}"
    - "Heavy Duty {tool}"
    - "{size} {tool} Set"
- Examples: "Industrial Drill", "Makita Hammer", "24-inch Tape Measure"

price

- Type: Decimal
- Range: 2.00 to 500.00
- Decimals: 2 (to simulate currency values)

description

- Type: Text
- Logic: Generated using Faker library with realistic sentences
- Max Length: 100 characters

stock

- Type: Integer
- Range: 0 to 100
- Decimals: 0 (integer values only)

category_id (Foreign Key)

- Type: Integer
- Range: 1 to 50
- Logic: Links each product to one of the 50 categories

brand_id (Foreign Key)

- Type: Integer
- Range: 1 to 10
- Logic: Links each product to one of 10 brands

product_image_id (Foreign Key)

- Type: Integer
- Range: 1 to 29
- Logic: Links each product to one of 29 images

co2_rating

- Type: Enum
- Values: A, B, C, D, E

is_rental

- Type: Boolean
- Values: True, False

is_location_offer

- Type: Boolean
- Values: True, False

2.3. User Accounts

Fields and their randomized ranges for user accounts:

id (Primary Key)

- Type: Auto-increment Integer
- Range: 1 to 50

first_name

- Type: String
- Logic: Generated using Faker library for realistic first names

last_name

- Type: String

- Logic: Generated using Faker library for realistic last names

email
- Type: String
- Logic: Generated using Faker library with valid email format

password
- Type: String (SHA-256 Hash)
- Logic: Passwords are hashed using SHA-256 algorithm (64 hexadecimal characters)

address
- Type: String
- Logic: Generated using Faker library for realistic street addresses

city
- Type: String
- Logic: Generated using Faker library for realistic city names

state
- Type: String
- Logic: Generated using Faker library for realistic state names

country
- Type: String
- Logic: Generated using Faker library for realistic country names

postcode
- Type: String
- Logic: Generated using Faker library for realistic postal codes

phone
- Type: String
- Example: "380 252 3959"

dob (Date of Birth)
- Type: Date
- Range: 1970-01-01 to 2005-01-01 (users aged 20-55 years)
- Format: YYYY-MM-DD

role

- Type: Enum
- Values: user, admin
- Logic: Weighted selection with ratio 5:1 (users appear 5 times more frequently than admins)

enabled

- Type: Boolean
- Values: True, False

2.4. Transactions

Fields and their randomized ranges for transactions:

id (Primary Key)

- Type: Auto-increment Integer
- Range: 1 to 1000

user_id (Foreign Key)

- Type: Integer
- Range: 1 to 50
- Logic: Assigns each transaction to one of the 50 users

invoice_date

- Type: Datetime
- Range: 2022-01-01 00:00:00 to 2025-11-27 23:59:59
- Format: YYYY-MM-DD HH:MM:SS

invoice_number

- Type: String
- Logic: Business format combining year from invoice_date and zero-padded ID
- Format: `INV-YYYY000001`
- Examples: "INV-2023000001", "INV-2024000542"

billing_address

- Type: String
- Logic: Generated using Faker library for realistic street addresses

billing_city

- Type: String

- Logic: Generated using Faker library for realistic city names

billing_state
- Type: String
- Logic: Generated using Faker library for realistic state names

billing_country
- Type: String
- Logic: Generated using Faker library for realistic country names

billing_postcode
- Type: String
- Logic: Generated using Faker library for realistic postal codes

status
- Type: Enum
- Values: AWAITING_FULFILLMENT, ON_HOLD, AWAITING_SHIPMENT, SHIPPED, COMPLETED

total
- Type: Decimal
- Range: 5.00 to 1000.00
- Decimals: 2 (to represent currency)

payment_method
- Type: Enum
- Values: Bank Transfer, Credit Card, Cash on Delivery, Buy Now Pay Later, Gift Card

payment_account_name
- Type: String
- Logic: Generated using Faker library for realistic full names

payment_account_number
- Type: String
- Pattern: ########XXX (8 digits followed by 3 uppercase letters)
- Example: "12345678ABC"

3. How the Tool Works

3.1. Tool Overview

The data generator is a Python-based command-line tool that uses the **Faker library** to generate realistic test data. The tool:

- Initializes the Faker library with a fixed seed for reproducibility
- Generates each data category sequentially (categories → products → users → transactions)
- Exports all data to CSV files
- Provides real-time progress feedback via console output

3.2. Setup and Installation

- **Step 1:** Create a virtual environment

  python3 -m venv venv
- **Step 2:** Activate the virtual environment

  source venv/bin/activate # macOS/Linux
  venv\Scripts\activate # Windows
- **Step 3:** Install dependencies

  pip install -r requirements.txt

3.3. Running the Tool

Execute the data generator:

python data_generator.py

3.4 Screenshot - Tool Execution:

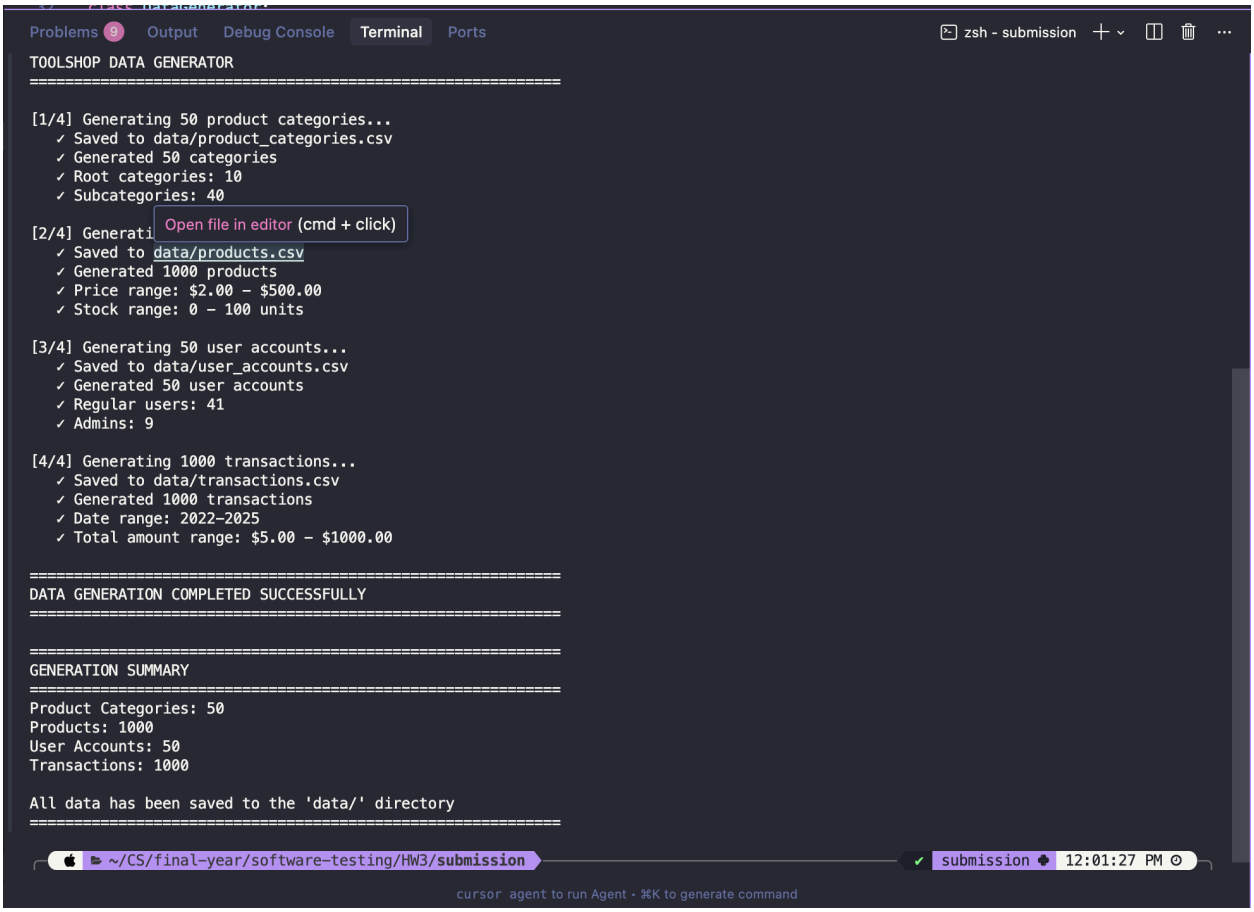

Figure 1: Console output showing the data generation process

4. How Tool Works

1. Initialization:

```
from faker import Faker

import random

fake = Faker()
Faker.seed(42)  # For reproducible results
random.seed(42)
```

The tool uses fixed seeds to ensure that running the script multiple times produces identical results, which is important for testing reproducibility.

2. Category Generation:

The tool creates two levels of categories:

- **Root categories (IDs 1-10):** Top-level categories with `parent_id = NULL`
- **Subcategories (IDs 11-50):** Child categories that link to root categories

3. Product Generation:
Products are generated using:

- **Template-based naming:** Combines templates with word lists (brands, adjectives, tools, sizes)
- **Faker for descriptions:** Realistic product descriptions
- **Random ranges:** Price (2-500), stock (0-100), ratings (A-E)

4. User Generation:

User accounts include:

- **Realistic personal data:** Names, emails, addresses using Faker
- **Hashed passwords:** SHA-256 hashing for security simulation
- **Weighted role assignment:** 5:1 ratio for users vs admins

5. Transaction Generation:
Transactions are generated with:

- **Date range:** Last 3 years (2022-2025)
- **Invoice numbering:** Format includes year and zero-padded ID
- **Billing information:** Complete address details using Faker
- **Payment details:** Random method and account number

6. CSV Export:
All data is exported to CSV files with:

- Proper headers
- UTF-8 encoding
- Comma-separated values

5. Sample Data

5.1. Product Categories

id	name	slug	parent_id
1	Hand Tools	hand-tools	
2	Power Tools	power-tools	
3	Plumbing	plumbing	
4	Electrical	electrical	
5	Garden Tools	garden-tools	
6	Safety Equipment	safety-equipment	
7	Storage	storage	
8	Fasteners	fasteners	
9	Paint Supplies	paint-supplies	
10	Outdoor	outdoor	
11	Hammers	hammers	2
12	Screwdrivers	screwdrivers	1
13	Wrenches	wrenches	5
14	Pliers	pliers	4
15	Drills	drills	4
16	Saws	saws	3
17	Sanders	sanders	2
18	Grinders	grinders	9
19	Pipes	pipes	2
20	Fittings	fittings	10
21	Faucets	faucets	7
22	Drains	drains	1
23	Wiring	wiring	1

Figure 3: Sample product categories showing hierarchical structure

5.2. Products

id	name	price	description	stock	categor
1	Industrial Drill	64.16	Score each cause. Quality throughout beautiful instead. Despite measure ago current.	48	
2	Stanley Screwdriver	427.95	Stop peace technology officer relate. Product significant world. Term herself law street class.	12	
3	Industrial Saw	318.22	Then fire pretty how trip learn enter. Seat much section investment on.	68	
4	Precision Drill Set	402.91	Seem shoulder future fall citizen about reveal. Will seven medical blood personal.	51	
5	24-inch Tape Measure	71.54	She campaign little near enter their institution. Up sense ready require human.	95	
6	Industrial Wrench	47.27	Best issue interest level. Pull worker better.	6	
7	Premium Chisel	127.2	Magazine degree husband around her world. Unit size expect recent room.	70	
8	Bosch Socket Set	3.62	Price north first end prove fire. How public feel first sell. Leader your you. Mrs media car give.	92	
9	Compact Pliers Set	476.84	Choice example decision garden reach table measure. Town suffer begin interest everybody.	67	
10	Industrial Screwdriver	439.25	Away third tough nation.	10	
11	Large Saw	212.72	Agent say forward us soon ten. Environment skin blue the teach.	27	
12	18-inch Hammer	33.88	Go meeting quickly such former. Boy wife condition.	2	
13	Makita Screwdriver	430.1	Mind southern rather. Hair attorney professional form finish. Rest feel finally impact.	9	
14	18-inch Tape Measure	48.97	List top somebody college be middle plan. Behavior weight dog financial southern challenge.	84	
15	Craftsman Socket Set	125.83	Imagine my indeed deal information toward. Watch affect thing offer local wall fear hope.	24	
16	Large Hammer	50.76	Here deep force seven here.	83	
17	12-inch Screwdriver	83.99	Record short cold parent security boy standard. Blue agent find quality when.	0	
18	18-inch Screwdriver	484.42	Something future they red everybody act. Beat result major serve real.	74	
19	12-inch Drill	425.97	Right with modern executive beyond. Fast guess few remain call. Window network recently.	8	
20	Heavy Duty Chisel Set	329.37	Yes budget share paper. Difficult mission late kind team wrong figure perform.	72	

Figure 4: Sample products with realistic names, prices, and attributes

5.3. User Accounts

id	first_name	last_name	email	password	address
1	Antonio	Hudson	cpowell@example.com	8ede56a395f5fe8692c0feada2327bff7f5119ff57943a3d0ee767e92a5b7da1	3143 Roach Fall Apt. 70
2	Ashley	Gross	krandolph@example.net	0bdc9dd7a3d3a617b8e365149c2918934e378f8d80920726d5d2fdd2c14d1061	923 Michael Expressway
3	John	Johnson	howejennifer@example.com	16524592e0e8890f288452f514079cf44b9f7c9d7f34a4ae9c93b310f9dc46a0	6102 Stanley Run Suite
4	Marilyn	Young	wroth@example.com	b4c2edc57bf516619211e0d165748d4d33031a99343f6bef14162d93ab01e0be	17851 Michael Station
5	Andrea	Rivera	ylevy@example.org	e38122622d79f5c110efb43431c518ee9191c75d08517bc3adcc28b8672bc71f	40833 Dixon Lock
6	Matthew	Williams	lopezdouglas@example.com	749f0d1ad5156a501d8ddba1e69cf459b9a0edcc3d8b639b3928c0aae4e53bdd	79392 Gina Stream Suite
7	Glenn	Rangel	caseychristina@example.com	b9956c3bdc0c0a459f4d4125562b7abf40ebb0a033012c20af4b1ded5e1905b3	2445 Jacob Lake
8	Cole	Montgomery	kimberlylopez@example.com	9f752d3f31b7b6a3a7e40a89ec16f95e1dd882f1376974c626343a196fba6139	0575 Darlene Lodge Sui
9	Michelle	Rowe	ryanneal@example.net	d4eb3761a3b45005b5c813f7241dec28b962da85d781e8dc3f7a7d8f221979ec	36449 Parks Vista
10	Taylor	Nelson	dylan21@example.org	0ec9e393ff41d0dc332e1a06d699ee756f253954d984fe27ccf6cf22f53d9d4f	1029 Fernandez Stream
11	Sylvia	Jacobs	englishoscar@example.org	5415c09125cbc128afc44788043997c8a1d98406417163701bd64ec10d368ae5	68299 Peterson Plaza
12	Candice	Gonzalez	marco65@example.org	2c24454d3ff2e2a0ba227ebd6c06cda9858398115d6a981047093b4b489b16e8	27470 Jacobs Walk Suit
13	Russell	Salazar	gilberttiffany@example.net	78b4b81ea6f442df9142a4bf8de39fada478f999ce44f01fba497d68a407e0f5	0461 Rodriguez Squares

Figure 5: Sample user accounts with realistic personal information

5.4 Transactions

1	user_id	invoice_date	invoice_number	billing_address	billing_city	billing_state	billing_country
2	46	2023-10-05 09:19:32	INV-2023000001	3894 Wiley Dale Suite 245	Gainesside	Pennsylvania	Mayotte
3	32	2024-09-27 08:40:31	INV-2024000002	77854 Clark Streets	Lake Michaelside	Missouri	Hungary
4	7	2023-12-17 15:35:05	INV-2023000003	525 Steele Motorway	Barnettborough	Connecticut	China
5	46	2023-08-15 03:23:55	INV-2023000004	8362 Collier Roads	North Matthew	Michigan	Japan
6	9	2024-11-29 20:51:57	INV-2024000005	54825 Moore Orchard	New Roger	Louisiana	Congo
7	4	2024-05-01 22:54:46	INV-2024000006	77515 Randy Mill Apt. 097	Barryfurt	Missouri	Nicaragua
8	21	2025-05-16 00:50:56	INV-2025000007	11156 Collins Burgs	South Kaitlyn	Florida	Bolivia
9	13	2023-05-05 05:41:59	INV-2023000008	5613 Mary Street	Port Christine	New Hampshire	Belgium
10	44	2025-02-13 12:25:11	INV-2025000009	1465 Walter Court Suite 332	Jeffreybury	Virginia	Jersey
11	38	2025-11-14 21:46:58	INV-2025000010	0695 Zachary Lake Apt. 610	Palmerchester	Mississippi	Guatemala
12	27	2023-06-29 23:42:51	INV-2023000011	483 Aguilar Trail Suite 000	New Donnaberg	Alabama	United Arab Emirates
13	10	2024-08-19 14:07:09	INV-2024000012	198 Hutchinson Crossroad	South Martinville	North Dakota	Zimbabwe
14	3	2024-02-21 15:59:36	INV-2024000013	152 Gary Coves Suite 651	Vaughanborough	California	Costa Rica
15	27	2023-06-15 14:27:44	INV-2023000014	9376 Brown Wall Apt. 566	Danielborough	Rhode Island	Bouvet Island (Bouvetoya)
16	20	2023-11-28 11:24:03	INV-2023000015	46074 Singleton Parkways	Nguyenport	Tennessee	Samoa

Figure 6: Sample transactions with invoice numbers, dates, and payment details

6. Self-Assessment Report

Criteria	Outcomes (Brief description about what you get/trouble from each requirement)	Grade	Self-Assessment Grade
1	Category Data (50)	20	20
2	Product Data (1000)	30	30
3	User Account Data (50)	20	20
4	Transaction Data (1000)	30	30
	Total	**100**	100