Run the General Programme (with seeded data)

**In MySQL shell:**

**Connect:**

\sql

\connect [root@127.0.0.1:3306](mailto:root@127.0.0.1:3306)

**Load report:**

1. *Fresh DB*

DROP DATABASE IF EXISTS mamma\_mia;

CREATE DATABASE mamma\_mia CHARACTER SET utf8mb4;

USE mamma\_mia;

1. *Tables & constraints*

\source "C:\Users\Charp\Documents\eu-pizza-system\02\_schema\00\_schema\_from\_model.sql"

\source "C:\Users\Charp\Documents\eu-pizza-system\02\_schema\01\_checks.sql"

\source "C:\Users\Charp\Documents\eu-pizza-system\02\_schema\02\_alter\_for\_reports.sql"

1. *Views*

\source "C:\Users\Charp\Documents\eu-pizza-system\04\_views\_funcs\views.sql"

1. *Functions & procedure*

\source "C:\Users\Charp\Documents\eu-pizza-system\04\_views\_funcs\functions\fn\_pizza\_price.sql"

\source "C:\Users\Charp\Documents\eu-pizza-system\04\_views\_funcs\functions\fn\_is\_birthday.sql"

\source "C:\Users\Charp\Documents\eu-pizza-system\04\_views\_funcs\functions\fn\_customer\_pizza\_count.sql"

\source "C:\Users\Charp\Documents\eu-pizza-system\04\_views\_funcs\functions\sp\_validate\_and\_lock\_discount\_code.sql"

1. *Delivery helper views*

\source "C:\Users\Charp\Documents\eu-pizza-system\04\_views\_funcs\queries\_delivery.sql"

1. *BIG seed (pizzas, ingredients, customers, addresses, drivers, zones, codes)*

\source "C:\Users\Charp\Documents\eu-pizza-system\03\_data\big\_seed.sql"

1. *Presentation orders (25+ orders over last 3 months + backdated statuses)*

\source "C:\Users\Charp\Documents\eu-pizza-system\03\_data\presentation\_orders.sql"

\source "C:\Users\Charp\Documents\eu-pizza-system\03\_data\backdate\_and\_status.sql"

1. *force one birthday today + one promo use today*

\source "C:\Users\Charp\Documents\eu-pizza-system\03\_data\presentation\_hotfix.sql"

**Sanity check:**

SELECT COUNT(\*) AS pizzas FROM pizzas;

SELECT COUNT(\*) AS customers FROM customers;

SELECT COUNT(\*) AS orders FROM orders;

SELECT MIN(placed\_at) AS first\_order, MAX(placed\_at) AS last\_order FROM orders;

SELECT status, COUNT(\*) FROM orders GROUP BY status;

SELECT \* FROM v\_menu\_pizzas LIMIT 5;

**Run the report in powershell**

cd C:\Users\Charp\Documents\eu-pizza-system

python app\reports.py

**Place an order:**

python app\place\_order.py --customer-id 1 --address-id 1 --postal-code 6211AA `

--pizzas "[{""pizza\_id"":1,""size"":""M"",""qty"":1}]" `

--drinks "[{""drink\_id"":1,""qty"":1}]" `

--desserts "[]"

**Re run:**

python app\reports.py

Re using a one-time code example:

**MySQL:**

\sql

\connect root@127.0.0.1:3306

USE mamma\_mia;

1. *sanity: confirm the one-time code exists and is single-use*

SELECT code, single\_use, redeemed\_at, redeemed\_by\_customer

FROM discount\_codes

WHERE code='ONETIME15';

1. *pick a valid customer + address + postal code (copy the 3 values you see)*

SELECT c.customer\_id, a.address\_id, a.postal\_code

FROM customers c

JOIN addresses a USING (customer\_id)

LIMIT 1;

**PowerShell:**

cd C:\Users\Charp\Documents\eu-pizza-system

python app\place\_order.py --customer-id 1 --address-id 1 --postal-code 6211AA `

--pizzas "[{""pizza\_id"":1,""size"":""M"",""qty"":1}]" `

--drinks "[]" --desserts "[]" `

--discount-code ONETIME15

-> Print order placed

-> Do it again, prints Order failed – rolled back

Negative Ingredient price

**MySQL:**

\sql

\connect root@127.0.0.1:3306

USE mamma\_mia;

1. *Try inserting a bad ingredient (should fail)*

INSERT INTO ingredients(name, cost\_per\_unit, unit, is\_meat, is\_animal\_product)

VALUES ('Bad Powder', -1.00, 'kg', 0, 0);

1. *Try making an existing one negative (should fail)*

UPDATE ingredients

SET cost\_per\_unit = -0.01

WHERE name = 'Dough';

1. *Sanity: confirm nothing bad slipped in*

SELECT name, cost\_per\_unit

FROM ingredients

WHERE name IN ('Bad Powder','Dough');