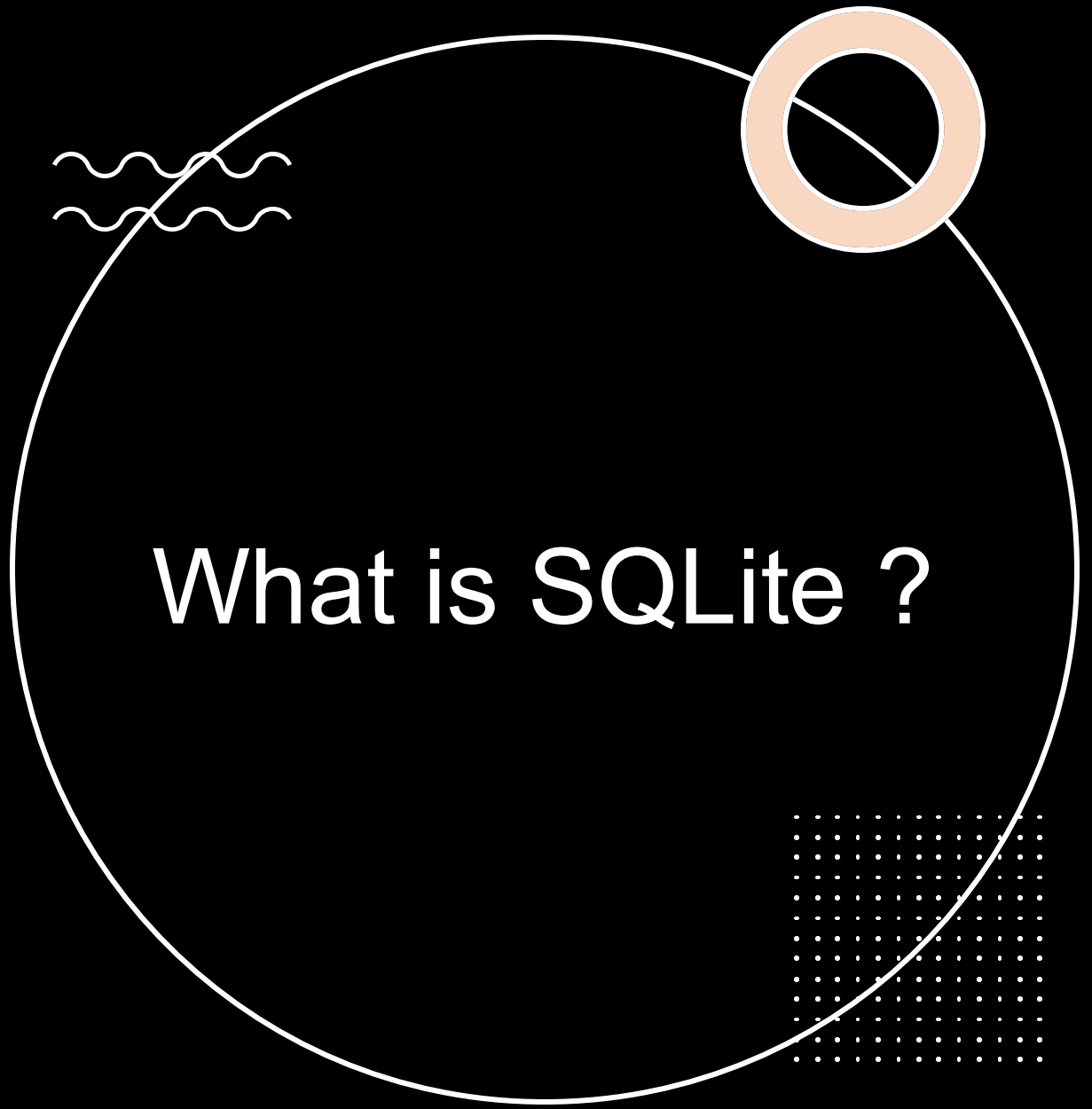
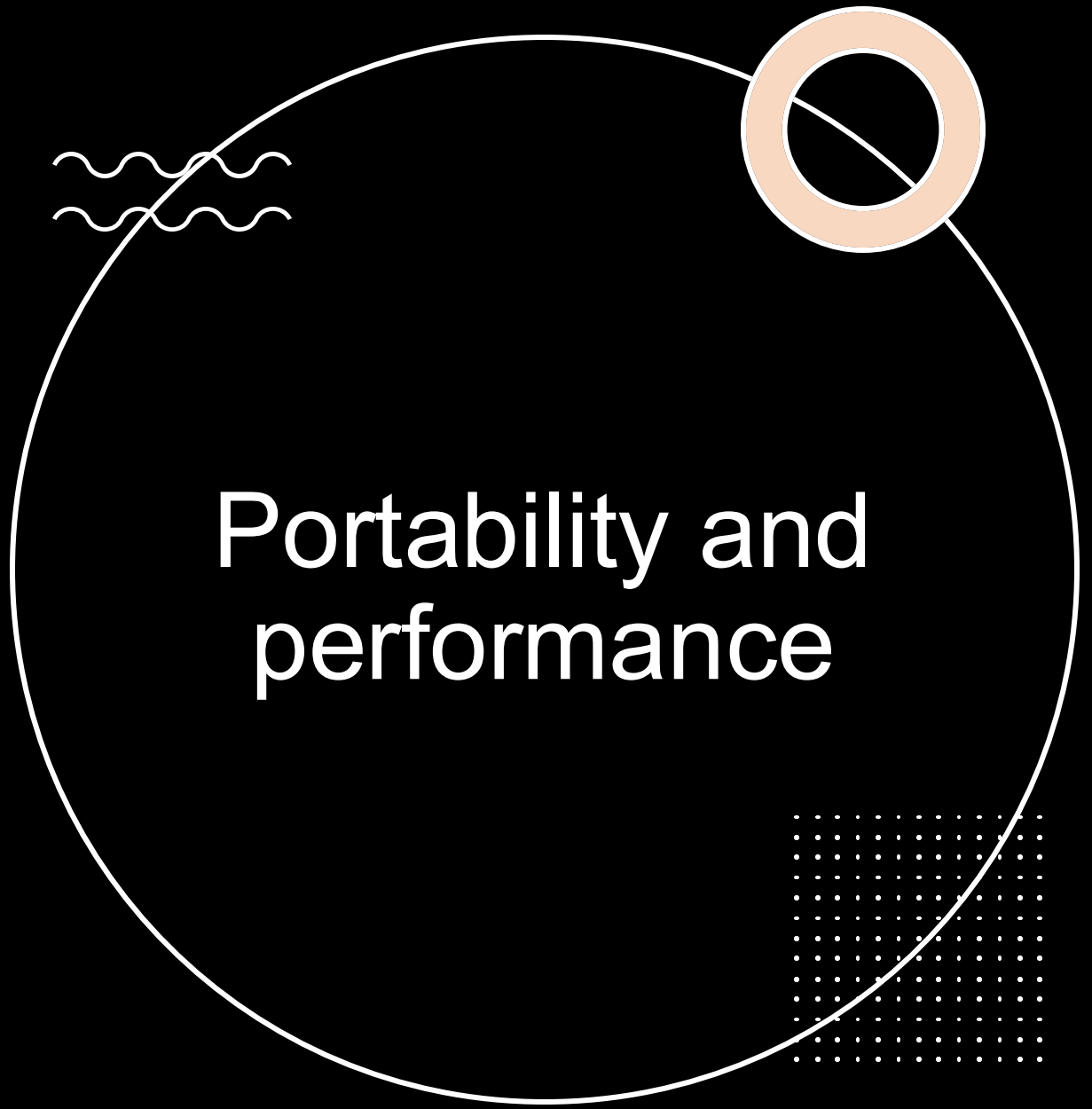
Final Projet

By Lilian BASTARD NAMINZO



What is SQLite ?

- 11th database engine DB-Engines
- Software in C
- Relational and transactionnal db engine
- Serverless -> local file
- Simple to integrate into apps



Portability and performance

- All in one file -> easy to attach
- No complex setup
- Cross-systems (Windows, Mac, Linux...)
- Lightweight

Data integrity and portability

- ACID :
  - Atomicity : all-or-nothing operations (commit or abort)
  - Consistency : data always valid, if rule transgressed -> abort
  - Isolation : transactions are independant
  - Durability : changes are permanent + data not stored in volatile memory

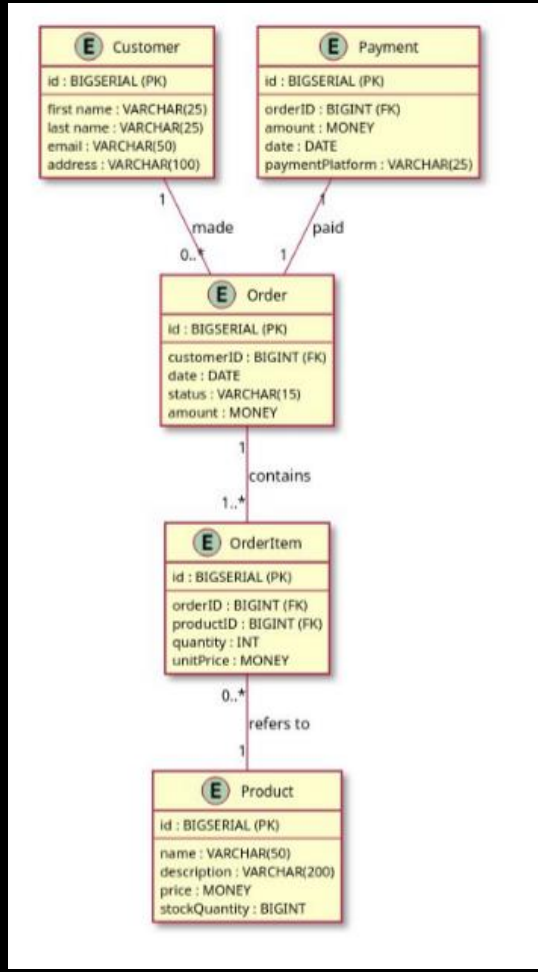
Wide application range

- Mobile applications (in iOS AND Android)
  - Cache, app data, settings all without connexion
- Desktop apps
  - Web browsers cookies & preferences
- Web projects
  - Useful to easily save & acces data (SQLite 3 API)
- IoT devices
  - Lightweight => manage data logs and config

What to retain ?

- Serverless
- Reliable
- Lightweight
- Very used for local storage

My SQL implementation Diagram



Queries - All orders from a specific customer

Retrieves every order that a customer made, using their email address (could also be with id)

```
SELECT
c.firstName,
c.lastName,
o.id AS orderID,
o.date AS orderDate,
o.amount AS totalAmount
FROM
customer AS c
JOIN
"order" AS o ON c.id = o.customerID
WHERE
c.email = 'jean.dupont@email.com';
```

Queries – All items of a specific order

Retrieves all items included in an order using its order id.

```
SELECT
oi.orderID,
p.name AS productName,
oi.quantity,
oi.unitPrice
FROM
orderitem AS oi
JOIN
product AS p ON oi.productID = p.id
WHERE
oi.orderID = 1;
```

Queries – All products with low stocks

Retrieves every products with a stock less than 15

```
SELECT
p.name AS productName,
p.stockQuantity
FROM
product AS p
WHERE
p.stockQuantity < 15
ORDER BY
p.stockQuantity ASC;
```

Queries – Most trending products

Retrieve the orders that are the most popular to the customers (best sellers)

```
SELECT
p.id,
p.name,
SUM(oi.quantity) AS totalOrdered
FROM
product AS p
JOIN
orderitem AS oi ON p.id = oi.productID
GROUP BY
p.id, p.name
ORDER BY
totalOrdered DESC
```

Queries – Average order value

Average price of an order (according to the order made by all the clients)

```
SELECT
ROUND(AVG(o.amount), 2) AS averageOrderValue
FROM
"order" AS o;
```