-- create
CREATE TABLE PURCHESES (id INTEGER PRIMARY KEY, customerId INTEGER, name TEXT NOT NULL, PURCHESES TEXT NOT NULL);

-- insert
INSERT INTO PURCHESES VALUES (1, 123, "daniel", "avocados,bananas");

-- query
SELECT * FROM PURCHESES;

-- What is wrong with the current table definition?
For one you shouldn't give a column the same name as the table - it's a violation of the SQL naming conventions act (geneva 1974). Secondly, the column definition defies the First Normal Form and the following forms, since it uses a repeating group, part of the attributes (namely the name) do not depend on the primary key, and it is possible to better represent the data by using two tables.

-- What anomalies can occur in a table defines this way?
A deletion anomaly can occur if you want to delete a purchase but it's the only purchase the customer has- you will end up deleting the person completely.
An update anomaly can occur if you want to update the name of a customer, since you then need to update their name for all their purchases- if there's an error, you will have a discrepancy.
An insertion anomaly occurs if you would like to add a new costumer to the database, but they have not purchased anything yet- you would not be able to add them since purchases cannot be null.

-- Refactor the code to normalize the data (First Normal Form)

--create
CREATE TABLE Customers (customerId INTEGER PRIMARY KEY, name TEXT NOT NULL);
CREATE TABLE Purchases (id INTEGER PRIMARY KEY, product TEXT NOT NULL PRIMARY KEY, customerId INTEGER);

--insert
INSERT INTO Customers VALUES(123, “daniel”);
INSERT INTO Purchases VALUES(1, “avocados”, 123),
(1, “bananas”, 123);

--query
SELECT * FROM Customer;
SELECT * FROM Purchases;

-- What law did the table comprimize? - (Goto my answer from q1)