

Melting Moments

The Chocolate Factory

Razan almalki

Retaj Baaqeel

Raghad Murad

Group: 9

Business Rules

Policy: The policy of the chocolate factory is to ensure the production of high-quality chocolate products while prioritizing food safety and customer satisfaction. This includes sourcing premium ingredients, implementing strict quality control measures, and adhering to regulatory standards. The policy aims to establish the chocolate factory as a trusted brand known for its exceptional chocolates.

Procedure: The procedure for chocolate production involves several steps. It starts with the careful selection and measurement of ingredients, including cocoa beans, sugar, milk, and flavors. The ingredients are then processed, mixed, and tempered to create the desired chocolate base. The chocolate is molded into various shapes and sizes, cooled, and solidified. Afterward, the chocolates undergo quality checks for taste, texture, and appearance. Once approved, the chocolates are packaged, labeled with relevant information such as ingredients and expiration dates, and stored in a controlled environment. When orders are received, the packaged chocolates are assigned to a driver for delivery to distributors or customers.

Principle: The principle followed by the chocolate factory is to prioritize sustainability and ethical practices throughout the production process. This involves sourcing ingredients from suppliers who adhere to fair trade practices, and supporting environmentally friendly cultivation methods, and promoting social responsibility within the supply chain. The principle reflects the chocolate factory's commitment to producing delectable chocolates while considering the impact on communities and the environment.

Constraints:

- Quality control measures must be followed during the production process.
 - Regulatory standards and labeling requirements must be adhered to.
 - Allergen information must be accurately labeled on the packaging.
 - Drivers must have a valid license and adhere to safe driving practices.
 - Timely delivery schedules need to be maintained.
- Ingredients and packaging materials must meet quality standards and be stored properly.

Entities:

1. Ingredients: Represents the raw materials used in chocolate production, such as cocoa beans, sugar, milk, and flavors.
2. Chocolate Product: Represents the various types of chocolate products produced by the factory, including bars, truffles, or bonbons.
3. Packaging: Represents the materials used for packaging the chocolate products, such as boxes, wrappers, or containers.
4. Driver: Represents the individuals responsible for delivering the packaged chocolates to distributors or customers.

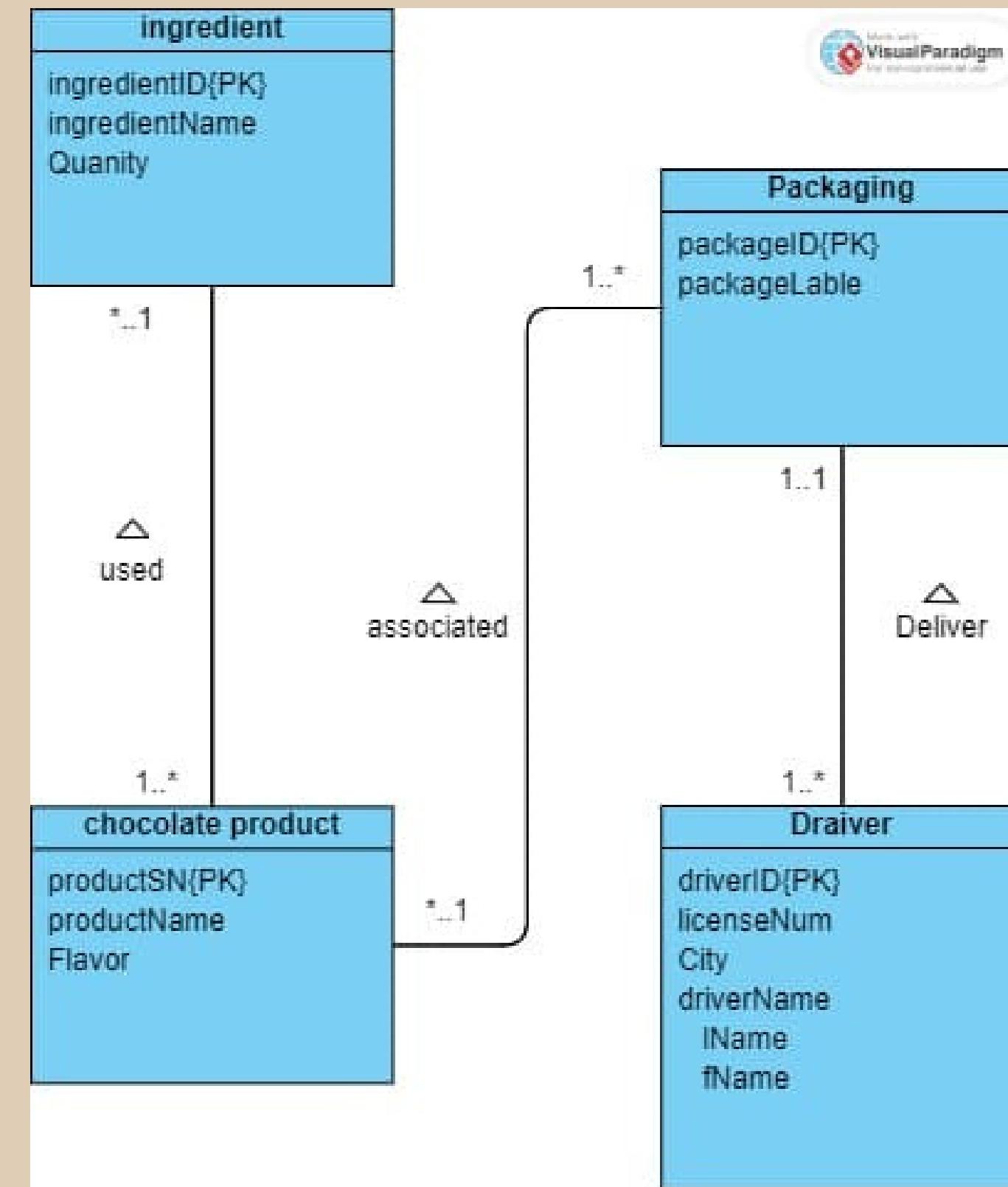
Attributes:

- Ingredients: Ingredient ID, Ingredient Name, Quantity.
- Chocolate Product: Product ID, Product Name, Flavor.
 - Packaging: Packaging ID, Packaging Type.
- Driver: Driver ID, Driver Name, License Number, city.

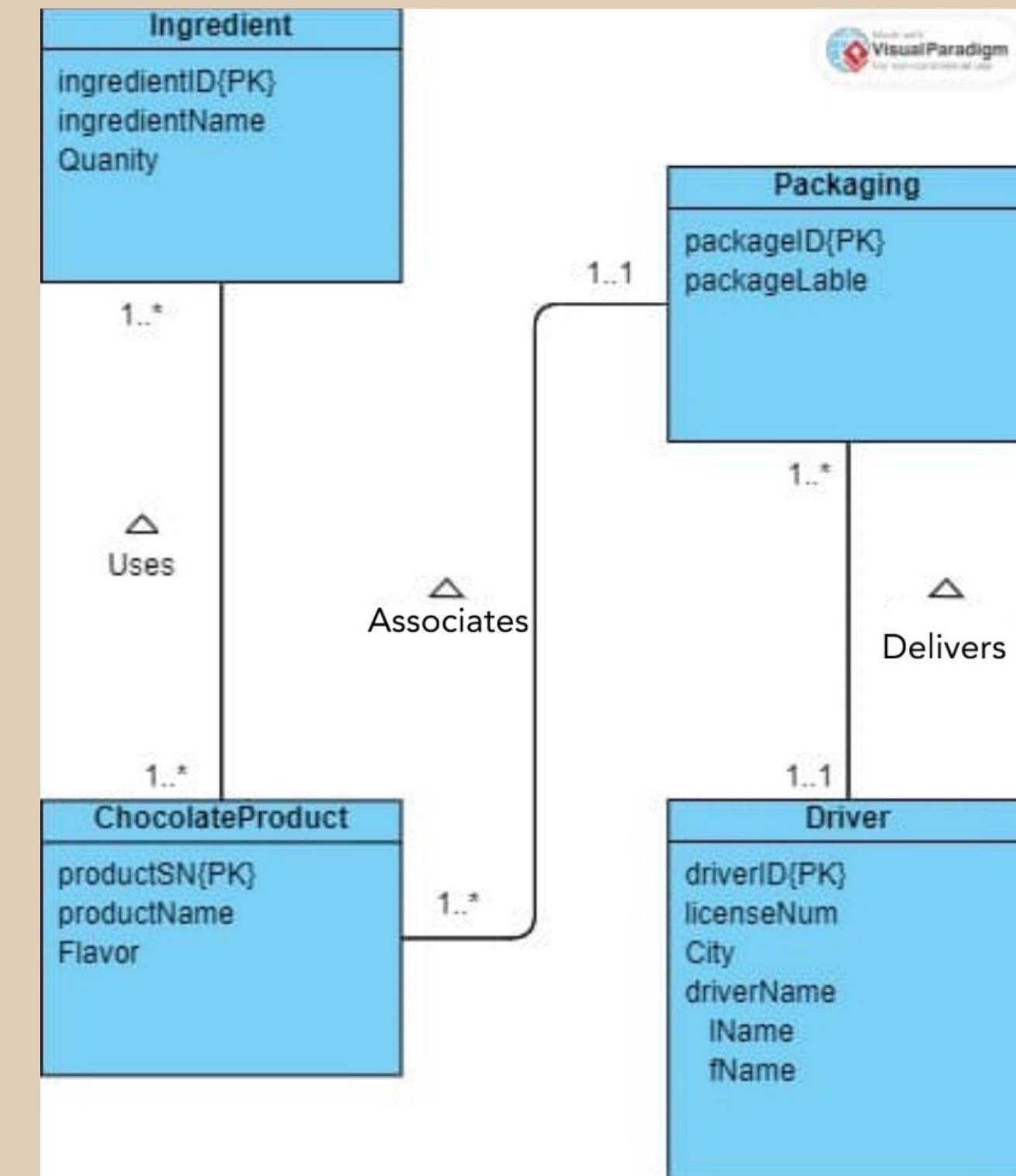
Relationships:

- Ingredients are used in the production of Chocolate Products.
- Chocolate Products are packaged using Packaging materials.
- Drivers are assigned to deliver the packaged Chocolate Products.

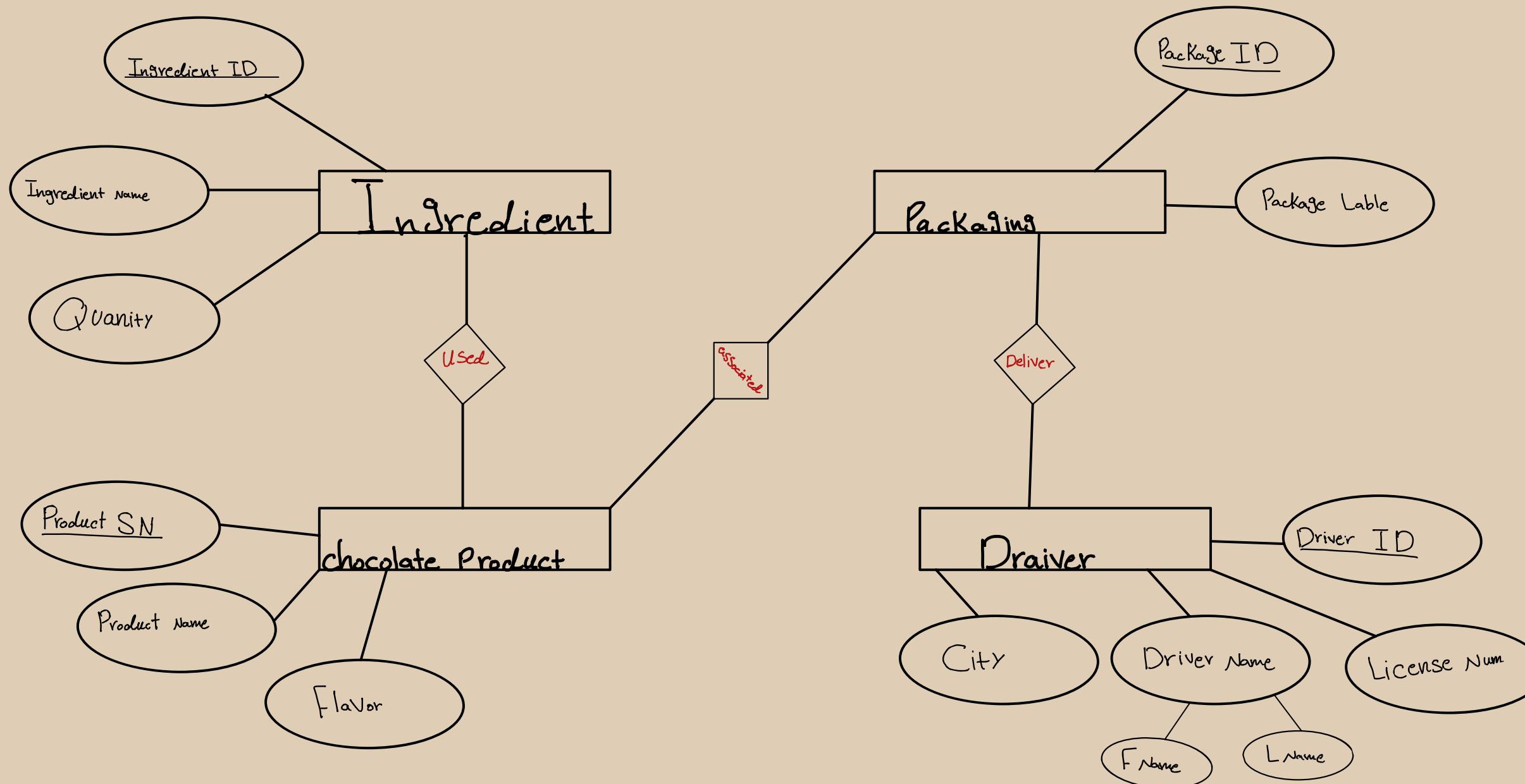
UML Before:



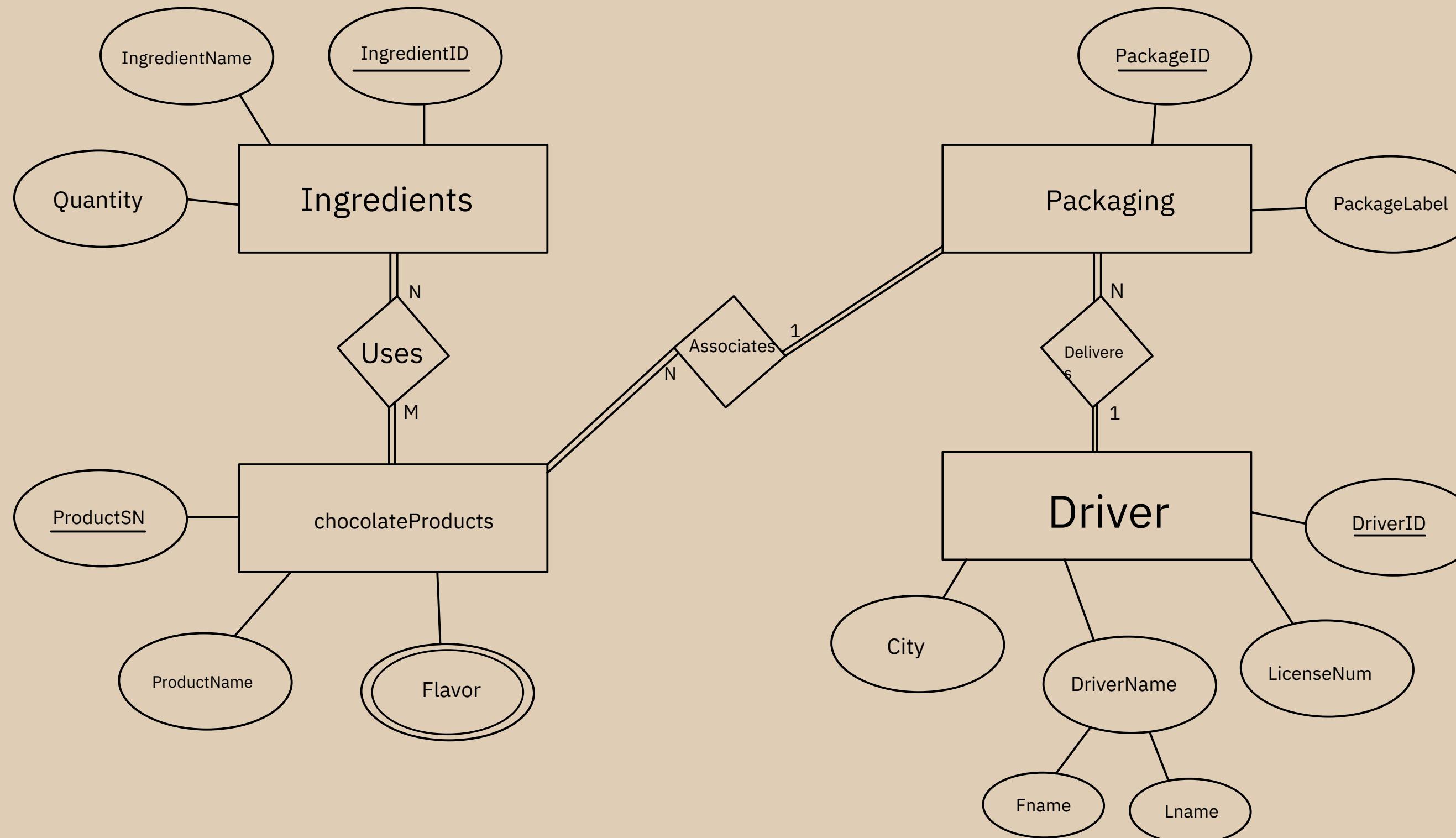
UML After:



Chen notion before:



Chen notion After:

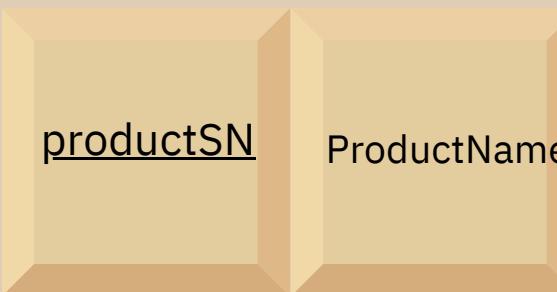


Step 1- mapping regular entity

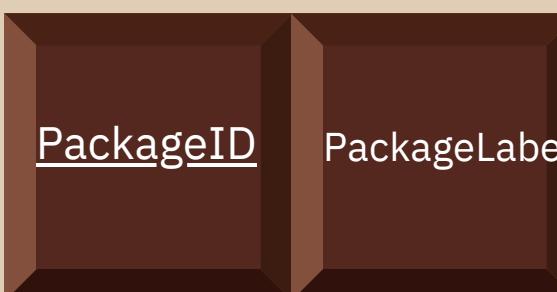
Ingredients Table



chocolateProducts Table



Packaging Table



Driver Table



Step 2- Mapping of Weak Entity Types is Null

Step 3- Mapping of Binary 1:N Relationship Types.

Associates relationship type

chocolateProducts Table

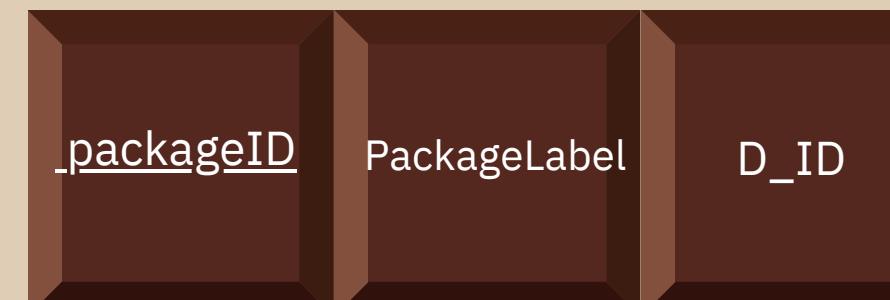


Packaging Table



Delivers relationship type

Packaging Table



Driver Table



Step 4-Mapping of Binary 1:N Relationships Types is Null

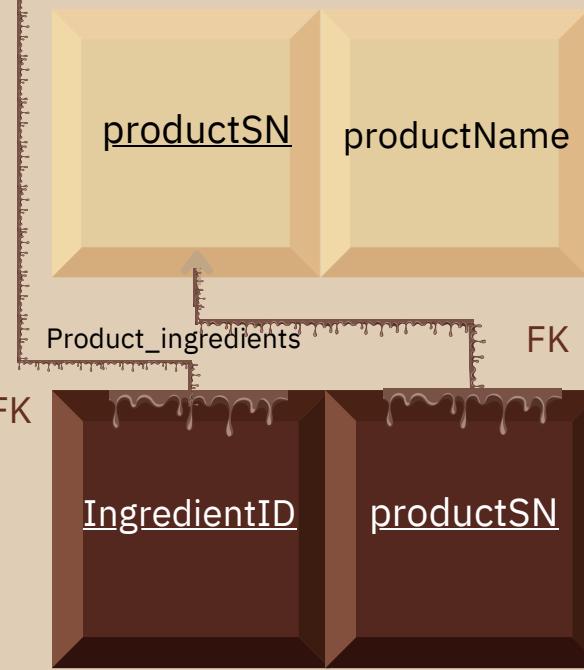
Step 5- Mapping of Binary M:N Relationship Types

Uses relationship type

Ingredients Table

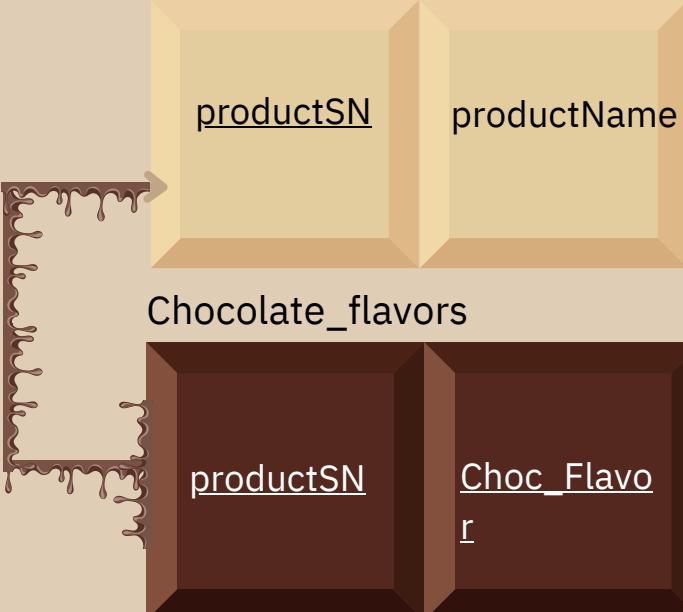


chocolateProducts Table:



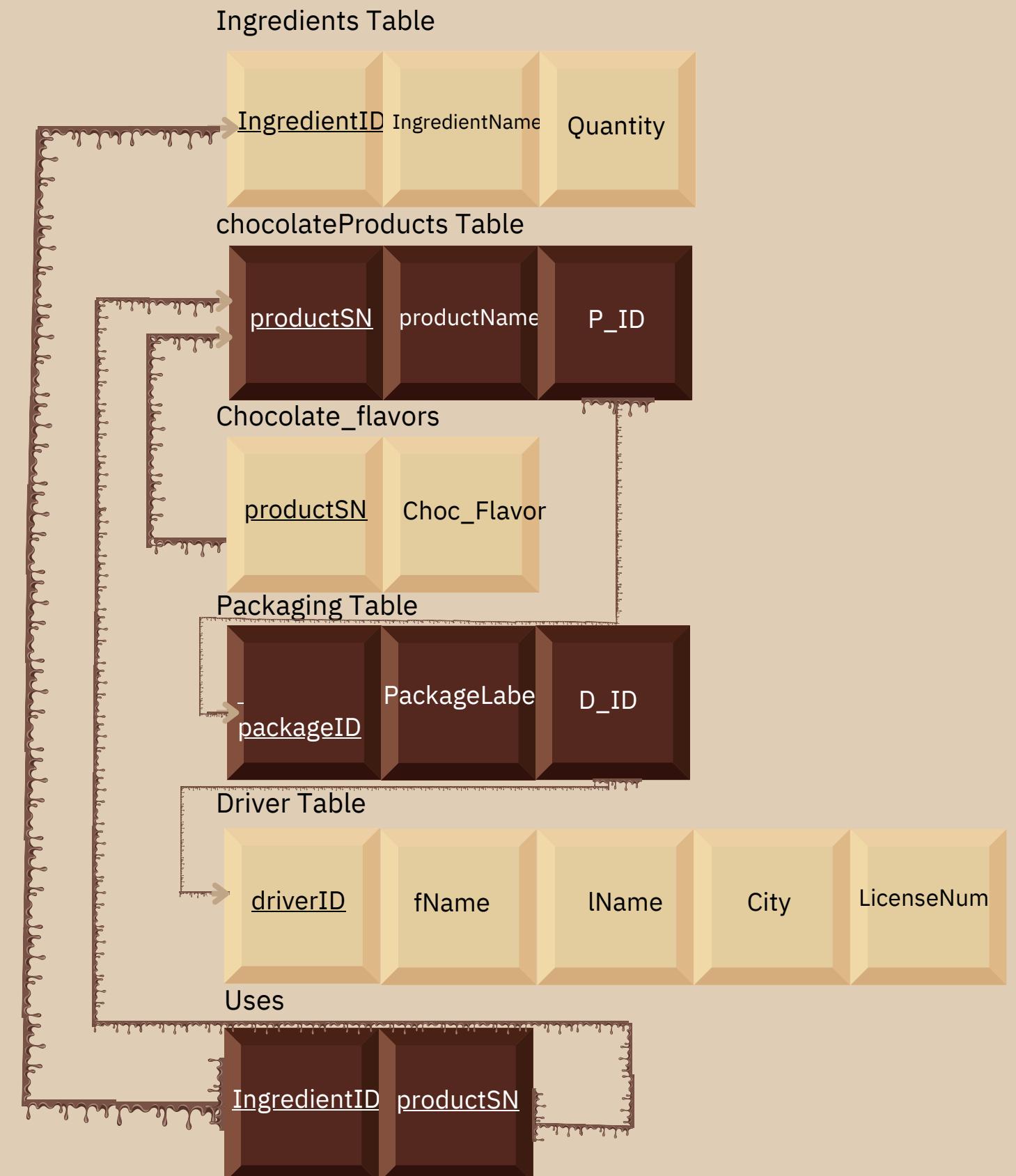
Step 6 - Mapping of Multivalued Attributes

chocolateProducts Table:



Step7- Mapping of N-are Relationships Types is Null

Final Mapping



Normalization

Ingredients Table

IngredientID	IngredientName	Quantity

1-First Normal Form(1NF)

No multivalued attribute so the table is in First Normal Form(1NF)

2- Second Normal Form(2NF)

No partial dependencie,so the table is in Second Normal Form (2NF)

3-Third Normal Form(3NF)

No transitive dependencies, so the table is in Third Normal Form (3NF)

Normalization

ChocolateProducts Table



1-First Normal Form(1NF)

We've covered multivalued attribute in the mapping, so the table is in First Normal Form(1NF)

2- Second Normal Form(2NF)

No partial dependencie,so the table is in Second Normal Form (2NF)

3-Third Normal Form(3NF)

No transitive dependencies, so the table is in Third Normal Form (3NF)

Normalization

Packaging Table



1-First Normal Form(1NF)

No multivalued attribute so the table is in First Normal Form(1NF)

2- Second Normal Form(2NF)

No partial dependencie,so the table is in Second Normal Form (2NF)

3-Third Normal Form(3NF)

No transitive dependencies, so the table is in Third Normal Form (3NF)

Normalization

Driver Table



1-First Normal Form(1NF)

No multivalued attribute so the table is in First Normal Form(1NF)

2- Second Normal Form(2NF)

No partial dependencie,so the table is in Second Normal Form (2NF)

3-Third Normal Form(3NF)

No transitive dependencies, so the table is in Third Normal Form (3NF)



SQL implementation

The creation and the output of the
database and every table



Navigator

SCHEMAS

Filter objects

- chocolatefactory
 - Tables
 - Views
 - Stored Procedures
 - Functions
- coursedb
- productorders
- researchlab
- rrrr
- sys
- unidb
- world

Administration Schemas

Information

No object selected

Query 1 Schemas Schemas 4 Schemas 5

1 CREATE DATABASE `chocolateFactory`;

Limit to 1000 rows

1

SCHEMAS

Filter objects

chocolatefactory

Tables

- ingredient
- Columns
- Indexes
- Foreign Keys
- Triggers

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

CREATE DATABASE `chocolateFactory`;

USE chocolateFactory;

CREATE TABLE Ingredient (

 ingredient_id INT PRIMARY KEY,

 ingredient_name VARCHAR(50),

 quantity INT

);

Administration Schemas

Information

No object selected

SCHEMAS

Filter objects

chocolatefactory

- Tables
 - ingredient
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
- Views
- Stored Procedures
- Functions

coursedb

productorders

researchlab

rrr

sys

unidb

world

1 • `SELECT * FROM chocolatefactory.ingredient;`

Limit to 1000 rows

Administration Schemas

Information

No object selected

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	ingredient_id	ingredient_name	quantity
•	NULL	NULL	NULL

SCHEMAS



Limit to 1000 rows

Filter objects

chocolatefactory

Tables

- ▶ ingredient
- ▶ packaging

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

```
1 CREATE DATABASE `chocolateFactory`;  
2  
3 • USE chocolateFactory;  
4  
5 • CREATE TABLE Ingredient (  
6     ingredient_id INT PRIMARY KEY,  
7     ingredient_name VARCHAR(50),  
8     quantity INT  
9 );  
10  
11 • CREATE TABLE Packaging (  
12     packaging_id INT PRIMARY KEY,  
13     package_label VARCHAR(50)  
14 );
```

Administration Schemas

Information

No object selected

Navigator

Query 1

SQL File 6*

SQL File 4*

SQL File 6*

packaging

SCHEMAS

Filter objects

chocolatefactory

Tables

- ▶ ingredient
- ▶ packaging

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

Administration Schemas

Information

No object selected

1 • `SELECT * FROM chocolatefactory.packaging;`

Limit to 1000 rows

1

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	packaging_id	package_label
•	NUL	NUL

SCHEMAS

Filter objects

chocolatefactory

Tables

- chocolateproduct
- ingredient
- packaging

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

Administration Schemas

Information

No object selected

```
1 CREATE DATABASE `chocolateFactory`;
2
3 • USE chocolateFactory;
4
5 • - CREATE TABLE Ingredient (
6     ingredient_id INT PRIMARY KEY,
7     ingredient_name VARCHAR(50),
8     quantity INT
9 );
10
11 • - CREATE TABLE Packaging (
12     packaging_id INT PRIMARY KEY,
13     package_label VARCHAR(50)
14 );
15
16 • - CREATE TABLE ChocolateProduct (
17     product_id INT PRIMARY KEY,
18     product_name VARCHAR(50),
19     packaging_id INT,
20     CONSTRAINT fk_packaging
21         FOREIGN KEY (packaging_id)
22             REFERENCES Packaging(packaging_id)
23 );
```

Navigator

Query 1

SQL File 6*

SQL File 4*

SQL File 6*

chocolateproduct

SCHEMAS

Filter objects

chocolatefactory

Tables

- ▶ chocolateproduct
- ▶ ingredient
- ▶ packaging

Views

Stored Procedures

Functions

coursedb**productorders****researchlab****rrrr****sys****unidb****world**

Administration Schemas

Information

No object selected



Limit to 1000 rows

1 • `SELECT * FROM chocolatefactory.chocolateproduct;`

	product_id	product_name	packaging_id
●	HULL	HULL	HULL

SCHEMAS

 Filter objects

chocolatefactory

Tables

- ▶ chocolateproduct
- ▶ driver
- ▶ ingredient
- ▶ packaging

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

Administration Schemas

Information

No object selected

```
8     quantity INT
9 );
10
11 • ◻ CREATE TABLE Packaging (
12     packaging_id INT PRIMARY KEY,
13     package_label VARCHAR(50)
14 );
15
16 • ◻ CREATE TABLE ChocolateProduct (
17     product_id INT PRIMARY KEY,
18     product_name VARCHAR(50),
19     packaging_id INT,
20     CONSTRAINT fk_packaging
21         FOREIGN KEY (packaging_id)
22             REFERENCES Packaging(packaging_id)
23 );
24
25 • ◻ CREATE TABLE Driver (
26     driver_id INT PRIMARY KEY,
27     first_name VARCHAR(50),
28     last_name VARCHAR(50),
29     city VARCHAR(50),
30     driver_key VARCHAR(50)
31 );
```

SCHEMAS



Limit to 1000 rows



Filter objects

chocolatefactory

Tables

- chocolateproduct
- driver
- ingredient
- packaging

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

1 • SELECT * FROM chocolatefactory.driver;

Administration

Schemas

Information

Result Grid



Filter Rows:

Edit:



Export/Import:



	driver_id	first_name	last_name	city	driver_key
•	NULL	NULL	NULL	NULL	NULL

No object selected

SCHEMAS

Filter objects

chocolatefactory

Tables

- chocolateproduct
- driver
- ingredient
- packaging

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

Administration

Schemas

Information

No object selected

```
20      CONSTRAINT fk_packaging
21          FOREIGN KEY (packaging_id)
22              REFERENCES Packaging(packaging_id)
23      );
24
25 • ◻ CREATE TABLE Driver (
26     driver_id INT PRIMARY KEY,
27     first_name VARCHAR(50),
28     last_name VARCHAR(50),
29     city VARCHAR(50),
30     driver_key VARCHAR(50)
31 );
32
33 • ◻ CREATE TABLE Uses (
34     ingredient_id INT,
35     product_id INT,
36     CONSTRAINT fk_ingredient
37         FOREIGN KEY (ingredient_id)
38             REFERENCES Ingredient(ingredient_id),
39     CONSTRAINT fk_product
40         FOREIGN KEY (product_id)
41             REFERENCES ChocolateProduct(product_id),
42     PRIMARY KEY (ingredient_id, product_id)
43 );
```

SCHEMAS

Filter objects

chocolatefactory

- Tables
 - chocolateproduct
 - driver
 - ingredient
 - packaging
 - uses
- Views
- Stored Procedures
- Functions

coursedb

productorders

researchlab

rrrrr

sys

unidb

world

1 • SELECT * FROM chocolatefactory.usess;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

ingredient_id	product_id
HULL	HULL

No object selected

SCHEMAS



Limit to 1000 rows

Filter objects

chocolatefactory

Tables

- ▶ chocolateproduct
- ▶ driver
- ▶ ingredient
- ▶ packaging
- ▶ uses

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrrr

sys

unidb

world

Administration Schemas

Information

No object selected

```
28     last_name VARCHAR(50),  
29     city VARCHAR(50),  
30     driver_key VARCHAR(50)  
31   );  
32  
33 • ◻ CREATE TABLE Uses (  
34     ingredient_id INT,  
35     product_id INT,  
36     CONSTRAINT fk_ingredient  
37         FOREIGN KEY (ingredient_id)  
38             REFERENCES Ingredient(ingredient_id),  
39     CONSTRAINT fk_product  
40         FOREIGN KEY (product_id)  
41             REFERENCES ChocolateProduct(product_id),  
42     PRIMARY KEY (ingredient_id, product_id)  
43   );  
44  
45 • ◻ CREATE TABLE ChocolateFlavours (  
46     product_id INT,  
47     chocolate_flavour VARCHAR(50),  
48     CONSTRAINT fk_chocolate_product  
49         FOREIGN KEY (product_id)  
50             REFERENCES ChocolateProduct(product_id)  
51   );
```

SCHEMAS

 Filter objects

Limit to 1000 rows

chocolatefactory

Tables

- ▶ chocolateflavours
- ▶ chocolateproduct
- ▶ driver
- ▶ ingredient
- ▶ packaging
- ▶ uses

Views

Stored Procedures

Functions

- ▶ coursedb
- ▶ productorders
- ▶ researchlab
- ▶ rrrr
- ▶ sys
- ▶ unidb
- ▶ world

1 • SELECT * FROM chocolatefactory.chocolateflavours;

Result Grid



Filter Rows:



Export:



Wrap Cell Content:



product_id

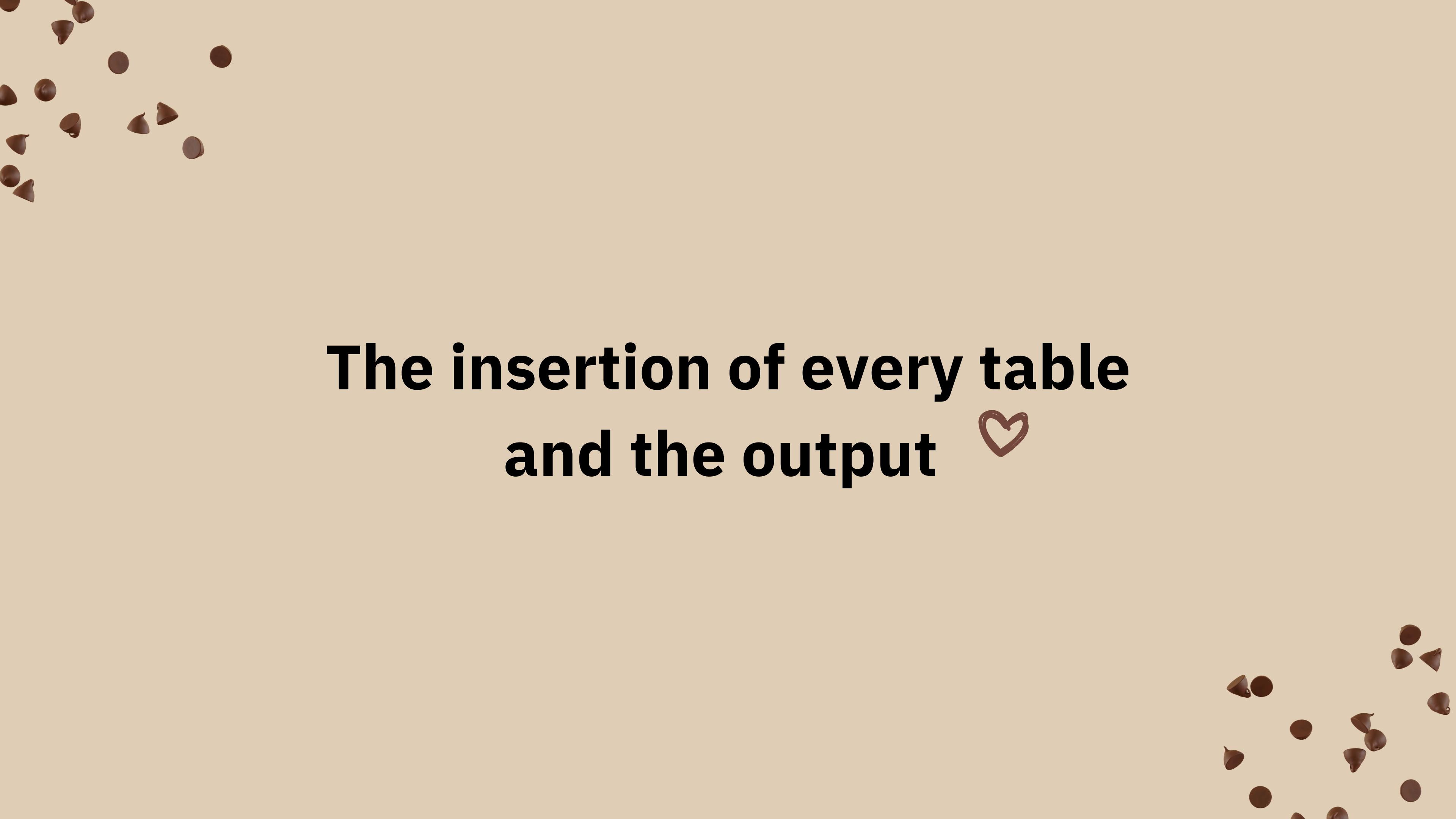
chocolate_flavour

Administration

Schemas

Information

No object selected



The insertion of every table
and the output ❤

```
Tables
  ► chocolateflavours
  ► chocolateproduct
  ► driver
  ► ingredient
  ► packaging
  ► uses
Views
Stored Procedures
Functions
coursedb
productorders
researchlab
rrrr
sys
unidb
world

35   product_id INT,
36   CONSTRAINT fk_ingredient
37       FOREIGN KEY (ingredient_id)
38           REFERENCES Ingredient(ingredient_id),
39   CONSTRAINT fk_product
40       FOREIGN KEY (product_id)
41           REFERENCES ChocolateProduct(product_id),
42   PRIMARY KEY (ingredient_id, product_id)
43 );
44
45 • ⓧ CREATE TABLE ChocolateFlavours (
46     product_id INT,
47     chocolate_flavour VARCHAR(50),
48     CONSTRAINT fk_chocolate_product
49         FOREIGN KEY (product_id)
50             REFERENCES ChocolateProduct(product_id)
51 );
52
53 • INSERT INTO Ingredient (ingredient_id, ingredient_name, quantity)
54     VALUES (1, 'Cocoa Powder', 100),
55             (2, 'Sugar', 200),
56             (3, 'Milk', 150),
57             (4, 'Vanilla Extract', 50);
```

Administration Schemas

Information

No object selected

Output

Action Output

Schemas

Filter objects

chocolatefactory

- Tables
 - chocolateflavours
 - chocolateproduct
 - driver
 - ingredient
 - packaging
 - uses
- Views
- Stored Procedures
- Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

1 • | SELECT * FROM chocolatefactory.ingredient;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	ingredient_id	ingredient_name	quantity
▶	1	Cocoa Powder	100
	2	Sugar	200
	3	Milk	150
	4	Vanilla Extract	50
		NONE	NONE

No object selected

ingredient 1

Apply

Revert

```
uses
views
Stored Procedures
Functions
sedb
ductorders
archlab
b
d
tion Schemas
ect selected
45 • CREATE TABLE ChocolateFlavours (
46     product_id INT,
47     chocolate_flavour VARCHAR(50),
48     CONSTRAINT fk_chocolate_product
49         FOREIGN KEY (product_id)
50             REFERENCES ChocolateProduct(product_id)
51 );
52
53 • INSERT INTO Ingredient (ingredient_id, ingredient_name, quantity)
54     VALUES (1, 'Cocoa Powder', 100),
55             (2, 'Sugar', 200),
56             (3, 'Milk', 150),
57             (4, 'Vanilla Extract', 50);
58
59 • INSERT INTO Packaging (packaging_id, package_label)
60     VALUES (1, 'Box'),
61             (2, 'Wrapper'),
62             (3, 'Jar');
```

Output

Action Output

#	Time	Action	Message	Duration / Feti
---	------	--------	---------	-----------------

SCHEMAS



Limit to 1000 rows



Filter objects

chocolatefactory

Tables

- ▶ chocolateflavours
- ▶ chocolateproduct
- ▶ driver
- ▶ ingredient
- ▶ packaging
- ▶ uses

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrr

sys

unidb

world

1 • SELECT * FROM chocolatefactory.packaging;

Administration

Schemas

Information

No object selected

Result Grid | Filter Rows: _____ | Edit: Export/Import: | Wrap Cell Content:

	packaging_id	package_label
▶	1	Box
	2	Wrapper
	3	Jar
*	NULL	NULL

Stored Procedures

Functions

sedb

uctorders

archlab

b

d

tion Schemas

ect selected

```
54     VALUES (1, 'Cocoa Powder', 100),  
55         (2, 'Sugar', 200),  
56         (3, 'Milk', 150),  
57         (4, 'Vanilla Extract', 50);  
58  
59 • INSERT INTO Packaging (packaging_id, package_label)  
60     VALUES (1, 'Box'),  
61         (2, 'Wrapper'),  
62         (3, 'Jar');  
63  
64 • INSERT INTO ChocolateProduct (product_id, product_name, packaging_id)  
65     VALUES (1, 'Dark Chocolate Bar', 1),  
66         (2, 'Milk Chocolate Bar', 1),  
67         (3, 'White Chocolate Bar', 1),  
68         (4, 'Assorted Chocolates', 2),  
69         (5, 'Chocolate Spread', 3);
```

Output

Action Output

#	Time	Action	Message	Duration / F...
57	21:56:49	SELECT * FROM chocolatefactory.packaging U...	3 row(s) returned	0.000 sec / 0
58	21:56:49	INSERT INTO ChocolateProduct (product_id, p...	Success! 5 row(s) inserted	0.000 sec / 0

SCHEMAS

Filter objects

chocolatefactory

Tables

- ▶ chocolateflavours
- ▶ chocolateproduct
- ▶ driver
- ▶ ingredient
- ▶ packaging
- ▶ uses

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world



1 • **SELECT * FROM chocolatefactory.chocolateproduct;**

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	product_id	product_name	packaging_id
▶	1	Dark Chocolate Bar	1
	2	Milk Chocolate Bar	1
	3	White Chocolate Bar	1
	4	Assorted Chocolates	2
	5	Chocolate Spread	3
	HULL	HULL	HULL

Administration Schemas

Information

No object selected

```
61          (2, 'Wrapper'),  
62          (3, 'Jar');  
63  
64 •    INSERT INTO ChocolateProduct (product_id, product_name, packaging_id)  
65      VALUES (1, 'Dark Chocolate Bar', 1),  
66          (2, 'Milk Chocolate Bar', 1),  
67          (3, 'White Chocolate Bar', 1),  
68          (4, 'Assorted Chocolates', 2),  
69          (5, 'Chocolate Spread', 3);  
70
```

```
71 •    INSERT INTO Driver (driver_id, first_name, last_name, city, driver_key)  
72      VALUES (1, 'John', 'Doe', 'New York', '1234'),  
73          (2, 'Jane', 'Smith', 'Los Angeles', '5678'),  
74          (3, 'David', 'Johnson', 'Chicago', '9012');  
75
```

Output

Action Output

#	Time	Action	Message	Duration
59	21:58:29	SELECT * FROM chocolatefactory.chocolateprod...	5 row(s) returned	0.000 s

SCHEMAS

Filter objects

chocolatefactory

Tables

- ▶ chocolateflavours
- ▶ chocolateproduct
- ▶ driver
- ▶ ingredient
- ▶ packaging
- ▶ uses

Views

Stored Procedures

Functions

- ▶ coursedb
- ▶ productorders
- ▶ researchlab
- ▶ rrrrr
- ▶ sys
- ▶ unidb
- ▶ world



1 • SELECT * FROM chocolatefactory.driver;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	driver_id	first_name	last_name	city	driver_key
▶	1	John	Doe	New York	1234
▶	2	Jane	Smith	Los Angeles	5678
▶	3	David	Johnson	Chicago	9012
◀	NULL	NULL	NULL	NULL	NULL

Administration Schemas

Information

No object selected

Filter objects

chocolatefactory

Tables

- chocolateflavours
- chocolateproduct
- driver
- ingredient
- packaging
- uses

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

Administration Schemas

Information

No object selected

```
66      (2, 'Milk Chocolate Bar', 1),  
67      (3, 'White Chocolate Bar', 1),  
68      (4, 'Assorted Chocolates', 2),  
69      (5, 'Chocolate Spread', 3);  
70  
71 • INSERT INTO Driver (driver_id, first_name, last_name, city, driver_key)  
72     VALUES (1, 'John', 'Doe', 'New York', '1234'),  
73             (2, 'Jane', 'Smith', 'Los Angeles', '5678'),  
74             (3, 'David', 'Johnson', 'Chicago', '9012');  
75  
76 • INSERT INTO Uses (ingredient_id, product_id)  
77     VALUES (1, 1),  
78             (2, 1),  
79             (2, 2),  
80             (3, 2),  
81             (4, 3),  
82             (1, 4),  
83             (2, 4),  
84             (3, 4),  
85             (4, 4),  
86             (1, 5),  
87             (2, 5),  
88             (3, 5);  
89
```

Schemas

Filter objects

chocolatefactory

- Tables
 - chocolateflavours
 - chocolateproduct
 - driver
 - ingredient
 - packaging
 - uses
- Views
- Stored Procedures
- Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

Administration Schemas

Information

No object selected

1 • SELECT * FROM chocolatefactory.usages;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	ingredient_id	product_id
1	1	
2	1	
2	2	
3	2	
4	3	
1	4	
2	4	
3	4	
4	4	
1	5	
2	5	
3	5	
*	NULL	NULL

uses 1 × Apply Revert

```
81      (4, 3),  
82      (1, 4),  
83      (2, 4),  
84      (3, 4),  
85      (4, 4),  
86      (1, 5),  
87      (2, 5),  
88      (3, 5);  
89  
90 • |INSERT INTO ChocolateFlavours (product_id, chocolate_flavour)  
91     VALUES (1, 'Dark'),  
92             (2, 'Milk'),  
93             (3, 'White'),  
94             (4, 'Assorted'),  
95             (5, 'Hazelnut');
```

Output

Action Output

#	Time	Action	Message	Duration
63	22:02:06	SELECT * FROM chocolatefactory.users LIMIT 0, ...	12 row(s) returned	0.000

SCHEMAS



Limit to 1000 rows

Filter objects

chocolatefactory

Tables

- ▶ chocolateflavours
- ▶ chocolateproduct
- ▶ driver
- ▶ ingredient
- ▶ packaging
- ▶ uses

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrrr

sys

unidb

world

1 • SELECT * FROM chocolatefactory.chocolateflavours;

Result Grid | Filter Rows: _____

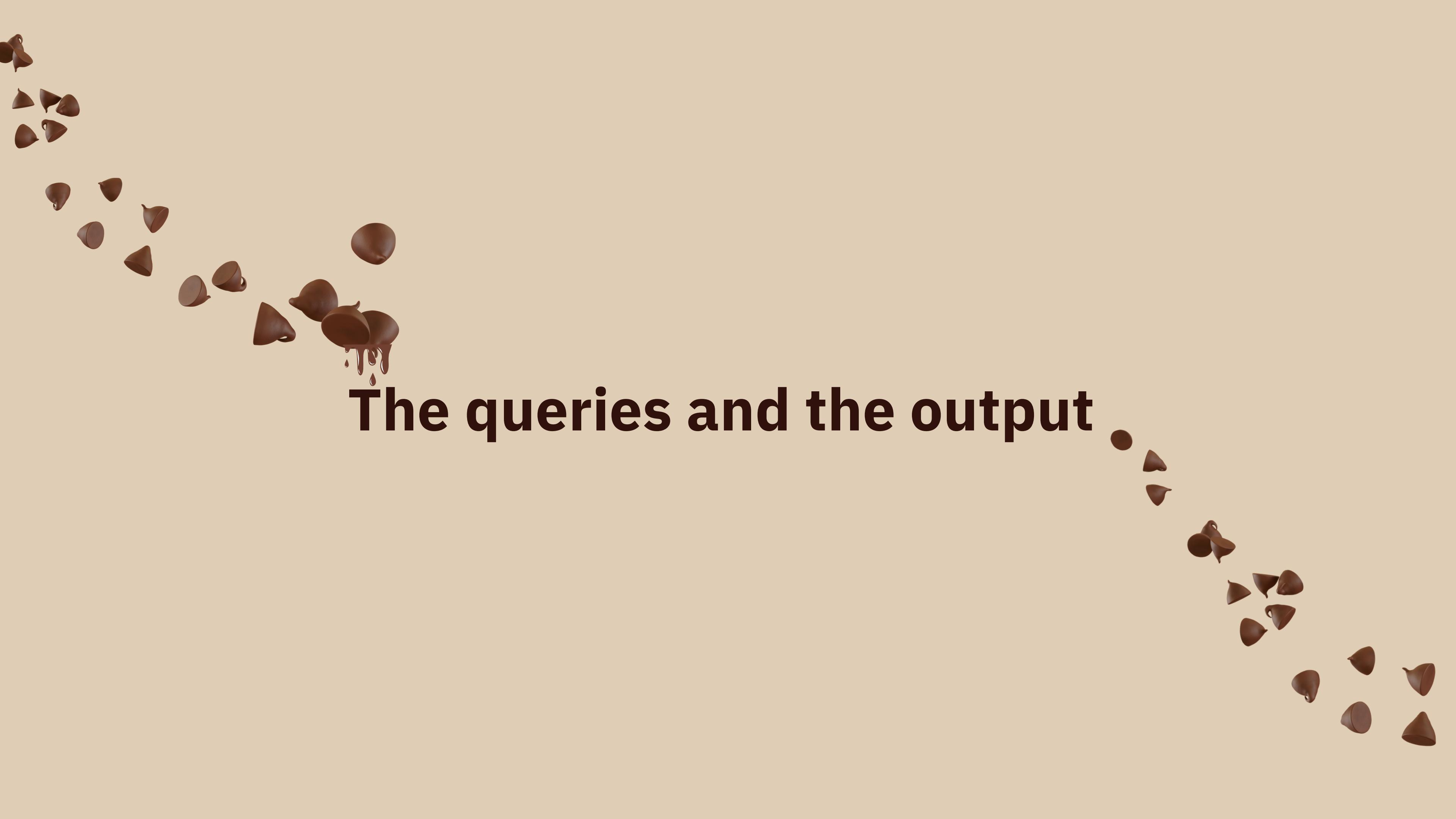
Export: Wrap Cell Content:

	product_id	chocolate_flavour
▶	1	Dark
	2	Milk
	3	White
	4	Assorted
	5	Hazelnut

Administration Schemas

Information

No object selected



The queries and the output .

Filter objects

- chocolatefactory
- coursedb
- productorders
- researchlab
- rrrr
- sys
- unidb
- world

```
91     VALUES (1, 'Dark'),  
92             (2, 'Milk'),  
93             (3, 'White'),  
94             (4, 'Assorted'),  
95             (5, 'Hazelnut');  
96  
97 • | SELECT ingredient_name FROM Ingredient;  
98
```

Result Grid | Filter Rows: _____ | Export: Wrap Cell Content:

	ingredient_name
▶	Cocoa Powder
	Sugar
	Milk
	Vanilla Extract

administration Schemas

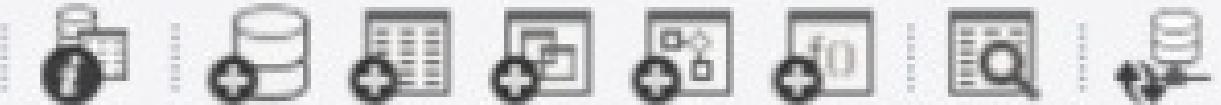
information

platefactory
edb
ctorders
rchlab

```
95      (3, "Hazelnut"),  
96  
97 •   SELECT ingredient_name FROM Ingredient;  
98  
99 •   SELECT p.product_name, pk.package_label  
100    FROM ChocolateProduct p  
101    JOIN Packaging pk ON p.packaging_id = pk.packaging_id;  
102
```

Result Grid | Filter Rows: _____ | Export: Wrap Cell Content:

	product_name	package_label
▶	Dark Chocolate Bar	Box
	Milk Chocolate Bar	Box
	White Chocolate Bar	Box
	Assorted Chocolates	Wrapper
	Chocolate Spread	Jar



Query 1

SQL File 6*

SQL File 4*

SQL File 6* X

```
101     JOIN Packaging pk ON p.packaging_id = pk.packaging_id;
102
103 •   SELECT p.product_name, COUNT(*) AS total_ingredients
104     FROM ChocolateProduct p
105     JOIN Uses u ON p.product_id = u.product_id
106     GROUP BY p.product_name
107     HAVING COUNT(*) >= 3;
108
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	product_name	total_ingredients
▶	Assorted Chocolates	4
	Chocolate Spread	3

```
105      JOIN Uses u ON p.product_id = u.product_id
106      GROUP BY p.product_name
107      HAVING COUNT(*) >= 3;
108
109 •   SELECT product_name, packaging_id
110   FROM ChocolateProduct
111   ORDER BY product_name ASC;
112
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	product_name	packaging_id
▶	Assorted Chocolates	2
	Chocolate Spread	3
	Dark Chocolate Bar	1
	Milk Chocolate Bar	1
	White Chocolate Bar	1

Schemas

SCHEMAS

Filter objects

chocolatefactory
coursedb
productorders
researchlab
rrrr
sys
unidb
world

109 • `SELECT product_name, packaging_id`
110 `FROM ChocolateProduct`
111 `ORDER BY product_name ASC;`
112
113 • `SELECT p.product_name, cf.chocolate_flavour`
114 `FROM ChocolateProduct p`
115 `LEFT JOIN ChocolateFlavours cf ON p.product_id = cf.product_id;`
116

Result Grid | Filter Rows: Export: Wrap Cell Content:

	product_name	chocolate_flavour
▶	Dark Chocolate Bar	Dark
	Milk Chocolate Bar	Milk
	White Chocolate Bar	White
	Assorted Chocolates	Assorted
	Chocolate Spread	Hazelnut

Administration Schemas

Information

No object selected

SCHEMAS



Limit to 1000 rows



```
115      LEFT JOIN ChocolateFlavours cf ON p.product_id = cf.product_id;
116
117 •   SELECT p.product_name
118     FROM ChocolateProduct p
119     JOIN Uses u ON p.product_id = u.product_id
120     JOIN Ingredient i ON u.ingredient_id = i.ingredient_id
121    WHERE i.ingredient_name = 'Cocoa Powder';
122
```

Result Grid | Filter Rows:

Export: Wrap Cell Content:

	product_name
▶	Dark Chocolate Bar
	Assorted Chocolates
	Chocolate Spread

Administration

Schemas

Information

No object selected

SCHEMAS

- Filter objects
- chocolatefactory
- coursedb
- productorders
- researchlab
- rrrrr
- sys
- unidb
- world

120 JOIN Ingredient i ON u.ingredient_id = i.ingredient_id
121 WHERE i.ingredient_name = 'Cocoa Powder';
122
123 • SELECT pk.package_label
124 FROM ChocolateProduct p
125 JOIN Packaging pk ON p.packaging_id = pk.packaging_id
126 WHERE p.product_name = 'Milk Chocolate Bar';
127

Result Grid | Filter Rows: Export: Wrap Cell Content:

	package_label
▶	Box

Administration Schemas

Information

No object selected

Filter objects

chocolatefactory
coursedb
productorders
researchlab
rrrr
sys
unidb
world

129 FROM ChocolateProduct p
130 JOIN Packaging pk ON p.packaging_id = pk.packaging_id
131 WHERE p.product_name = 'Milk Chocolate Bar';
132
133 • SELECT *
134 FROM Driver
135 WHERE city = 'New York';
136

Result Grid | Filter Rows: _____ | Edit: | Export/Import: | Wrap Cell Content: |

	driver_id	first_name	last_name	city	driver_key
▶	1	John	Doe	New York	1234
◀	NUL	NUL	NUL	NUL	NUL

Filter objects

- chocolatefactory
- coursedb
- productorders
- researchlab
- rrrr
- sys
- unidb
- world

```
134     FROM Driver
135     WHERE city = 'New York';
136
137 •   SELECT cf.chocolate_flavour, COUNT(*) AS product_count
138     FROM ChocolateFlavours cf
139     JOIN ChocolateProduct p ON cf.product_id = p.product_id
140     GROUP BY cf.chocolate_flavour;
141
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	chocolate_flavour	product_count
▶	Dark	1
	Milk	1
	White	1
	Assorted	1
	Hazelnut	1

Administration Schemas

Information

No object selected

Navigator Query 1 SQL File 6 SQL File 4 SQL File 6 Ingredient

SCHEMAS

Filter objects

chocolatefactory
coursedb
productorders
researchlab
rrr
sys
unidb
world

1 • **SELECT * FROM chocolatefactory.ingredient;**

2

3 • **UPDATE Ingredient**

4 **SET quantity = 120**

5 **WHERE ingredient_id = 1;**

6

Result Grid | Filter Rows: _____ | Edit: | Export/Import: | Wrap Cell Content: |

	ingredient_id	ingredient_name	quantity
▶	1	Cocoa Powder	120
	2	Sugar	200
	3	Milk	150
	4	Vanilla Extract	50
	NULL	NULL	NULL

Administration Schemas

Information

No object selected

Filter objects

chocolatefactory
coursedb
productorders
researchlab
rrrr
sys
unidb
world

```
1 •   SELECT * FROM chocolatefactory.driver;
2
3 •   UPDATE Driver
4     SET last_name = 'Almalki'
5     WHERE driver_id = 1;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	driver_id	first_name	last_name	city	driver_key
▶	1	John	Almalki	New York	1234
	2	Jane	Smith	Los Angeles	5678
	3	David	Johnson	Chicago	9012
*	NUL	NUL	NUL	NUL	NUL

administration Schemas

Schema:
chocolatefactory

chocolatefactory

Tables

- chocolateflavours
- chocolateproduct
- driver
- ingredient
- packaging
- uses

Views

Stored Procedures

Functions

coursedb

productorders

researchlab

rrr

sys

unidb

world

2

3 • DELETE FROM Uses

4 WHERE ingredient_id = 2 AND product_id = 1;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	ingredient_id	product_id
1	1	1
2	2	2
3	2	2
4	3	3
1	4	4
2	4	4
3	4	4
4	4	4
1	5	5
2	5	5
3	5	5
	HULL	HULL

Administration Schemas

Information

Schema:
chocolatefactory

chocolatefactory

- Tables
 - chocolateflavours
 - chocolateproduct
 - driver
 - ingredient
 - packaging
 - uses
- Views
- Stored Procedures
- Functions

coursedb
productorders
researchlab
rrrr
sys
unidb
world

Administration Schemas

Information:

Schema: chocolatefactory

2

3 • DELETE FROM Driver

4 WHERE driver_id = 3;

Result Grid | Filter Rows: _____ | Edit: | Export/Import: | Wrap Cell Content: |

	driver_id	first_name	last_name	city	driver_key
▶	1	John	Almalki	New York	1234
▶	2	Jane	Smith	Los Angeles	5678
◀	HULL	HULL	HULL	HULL	HULL

Task Distribution

Phase 1:

	Razan	Retaj	Raghad
Business rules	✓		
Chen's notation		✓	
UML			✓
PowerPoint	✓		

Phase 2:

	Razan	Retaj	Raghad
Mapping	✓	✓	
Normalization			✓
Presentation	✓		

Phase 3:

	Razan	Retaj	Raghad
SQL file	✓	✓	✓
Pdf file	✓		