A MANUFACTURING COMPANY
A BETTER FUTURE

For the purposes of the project, a fictitious company named BETTER FUTURE was created. It is a manufacturing company that produces three basic office products:

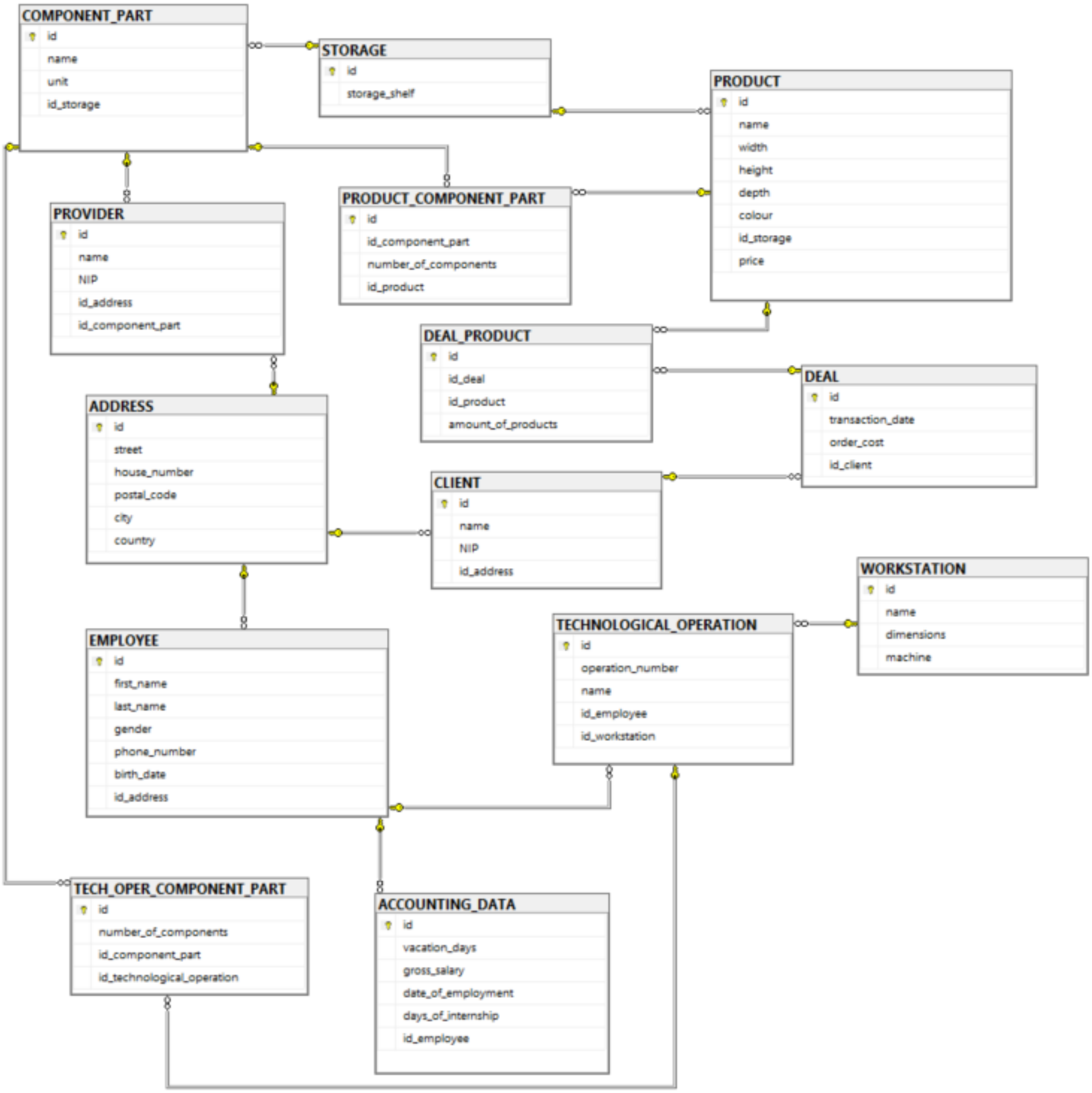
💡 A desk lamp

🪑 An office chair

🗄️ A desk

The products and the components are real. The manufacturing processes are also authentic simplified manufacturing processes for the products mentioned. A database called Better Future was created for the needs of this company. The application of the database will enable the company to record technical data such as information on finished products, components, necessary quantities of these components for the production of one piece of the product along with accounting information, data on customers, employees, suppliers and all transactions. It is necessary for the company to be able to collect information about the staff, technological operations as well as about customers and suppliers and even components necessary for the manufacturing process of the main products.

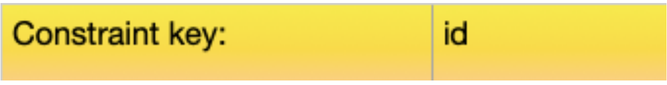
DIAGRAM


COMPONENT_PART
id
name
unit
id_storage
STORAGE
id
storage_shelf
PRODUCT
id
name
width
height
depth
colour
id_storage
price
PROVIDER
id
name
NIP
id_address
id_component_part
PRODUCT_COMPONENT_PART
id
id_component_part
number_of_components
id_product
DEAL_PRODUCT
id
id_deal
id_product
amount_of_products
DEAL
id
transaction_date
order_cost
id_client
ADDRESS
id
street
house_number
postal_code
city
country
CLIENT
id
name
NIP
id_address
WORKSTATION
id
name
dimensions
machine
TECHNOLOGICAL_OPERATION
id
operation_number
name
id_employee
id_workstation
EMPLOYEE
id
first_name
last_name
gender
phone_number
birth_date
id_address
TECH_OPER_COMPONENT_PART
id
number_of_components
id_component_part
id_technological_operation
ACCOUNTING_DATA
id
vacation_days
gross_salary
date_of_employment
days_of_internship
id_employee

TABLES

The database contains 14 tables:

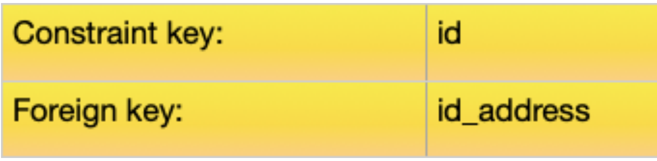
ADDRESS:



Constraint key:	id

The table collects information about employee’s, client’s and provider’s addresses. The relationship between the tables EMPLOYEE, PROVIDER and CLIENT is as follows: An employee, client and provider can have one address, but a specific address can be assigned to many clients. For example, two employees of the company may live at the same address. For example, in case of a married couple or siblings. Each of the above tables (CLIENT, PROVIDER, EMPLOYEE) contains the foreign key id_address referring to a primary key in the ADDRESS table.

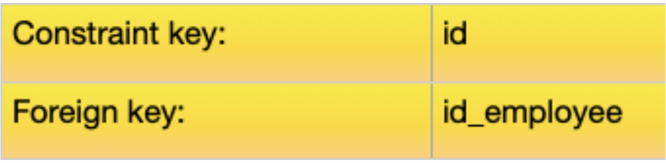
EMPLOYEE:



Constraint key:	id
Foreign key:	id_address

The table collects personal information and contact data of employees. As mentioned above, thanks to the foreign key id_address, it is possible to identify the employee's address.

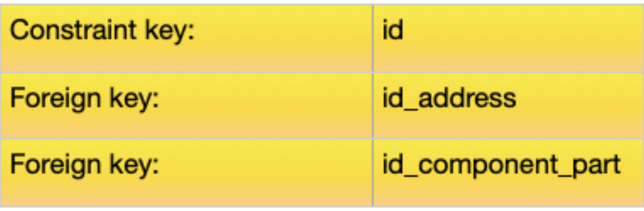
ACCOUNTING DATA:



Constraint key:	id
Foreign key:	id_employee

The table collects accounting data for each employee. It contains data on employees' leaves, monthly salaries, date of employment and days of internship. The foreign key id_employee refers to the primary key in the Employee table. One employee may have many entries in the accounting data, for example when he works on two different positions / functions, but the entry in the ACCOUNTING DATA is for one employee only.

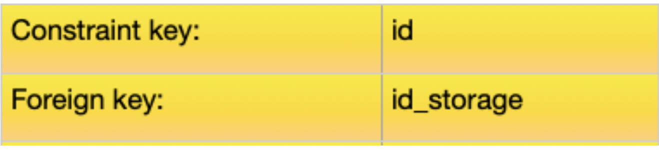
PROVIDER:



Constraint key:	id
Foreign key:	id_address
Foreign key:	id_component_part

The table collects names and data of delivery companies. Foreign key id_component_part allows to define what components are ordered from suppliers. A given component is ordered from one supplier, but one supplier offers multiple components to the company. Likewise in the table EMPLOYEE, thanks to the foreign key id_address, it is possible to identify the provider's address.

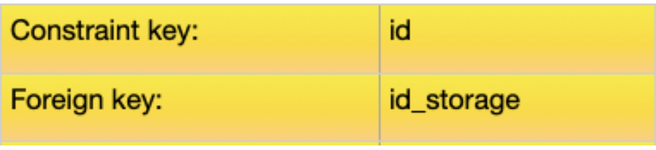
COMPONENT PART:



Constraint key:	id
Foreign key:	id_storage

The table collects data on the name of the component and the unit in which the component is determined for one piece of the finished product. A foreign key id_storage allows you to identify a warehouse location contained in the table STORAGE. It will make it much easier to find the necessary components during the production process.

PRODUCT:



Constraint key:	id
Foreign key:	id_storage

The table collects data of the name, dimensions of a given product, price and thanks to the connection with table STORAGE, a warehouse space. The table also collects data on the prices of finished goods

STORAGE :

Constraint key:	id

The table collects data on the markings of the shelves in the warehouse.

TECHNOLOGICAL_OPERATION:

Constraint key:	id
Foreign key:	id_employee
Foreign key:	id_workstation

The table gathers information about technological operations in the production process. It collects data on the operation number and its name. Thanks to the relationship with the table EMPLOYEE and the table WORKSTATION, it is possible to assign an employee responsible for a particular process and to assign a workstation where a process is carried out.

TECH_OPER_COMPONENT_PART:

Constraint key:	id
Foreign key:	id_component_part
Foreign key:	id_technological_operation

This table is an intermediate table between table TECHNOLOGICAL_OPERATION and COMPONENT PART. It is necessary due to the many-to-many relationship between these tables. Multiple components may be used in an operation and a given component may be used for multiple operations. The table also collects data on the number of components necessary for a given technological operation

PRODUCT_COMPONENT PART:

Constraint key:	id
Foreign key:	id_component_part
Foreign key:	id_product

This table is an intermediate table between table PRODUCT and table COMPONENT PART. Many of the same components may be used to manufacture a finished product and many products may require the same component item. The table collects data on the number of components contained in one finished product

CLIENT:

Constraint key:	id
Foreign key:	id_address

The table collects personal information and contact data on clients. Thanks to the foreign key id_address, it is possible to identify the client's address.

WORKSTATION:

Constraint key:	id

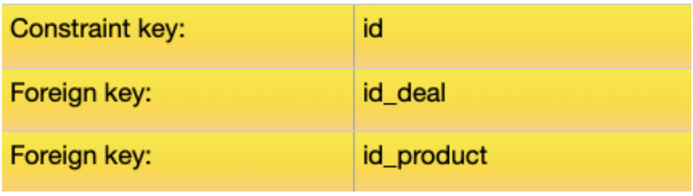
The table collects data on workstations and the dimensions of these workstations together with the machines used.

DEAL:

Constraint key:	id
Foreign key:	id_client

The table collects data of transactions – the date of purchase, the cost of the entire order and, thanks to the foreign key id_client to identify which customer has placed the order.

DEAL_PRODUCT:



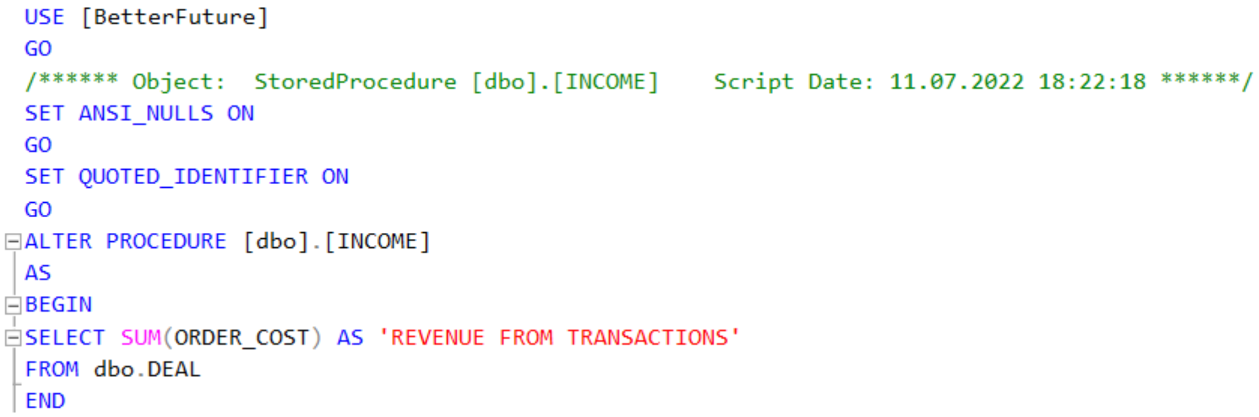
Constraint key:	id
Foreign key:	id_deal
Foreign key:	id_product

The table is an intermediate table between DEAL and PRODUCT tables. It is necessary because many products can appear in the order and many orders can be for the same product. The table collects data on the number of items purchased in a particular transaction.

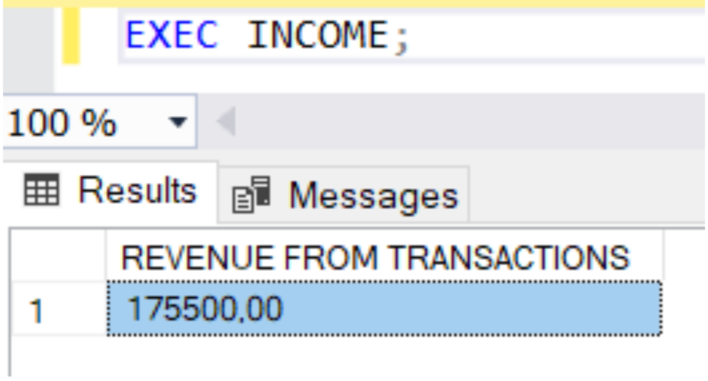
SORTED PROCEDURES

INCOME:

The procedure allows you to calculate the amount of money from products ordered by customers. Thanks to it, you can easily get the company's income result. The procedure code and its result are shown below:



```sql
USE [BetterFuture]
GO
/****** Object:  StoredProcedure [dbo].[INCOME]    Script Date: 11.07.2022 18:22:18 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[INCOME]
AS
BEGIN
SELECT SUM(ORDER_COST) AS 'REVENUE FROM TRANSACTIONS'
FROM dbo.DEAL
END
```

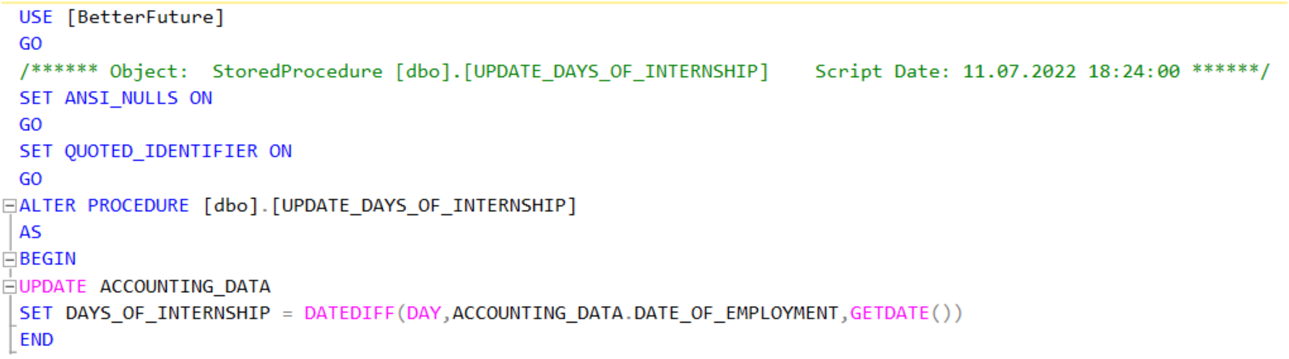


```
EXEC INCOME;
```

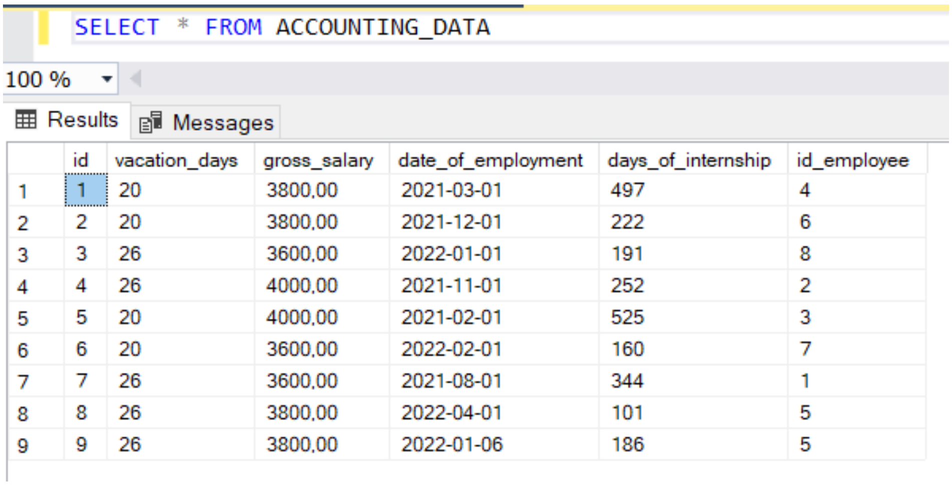
	REVENUE FROM TRANSACTIONS
1	175500,00

UPDATE_DAYS_OF_INTERNSHIP:

The procedure allows you to calculate the number of days of internship in table ACCOUNTING_DATA, taking into account the current date and the date of employment. The procedure code and its result are shown below: (procedure call on 07/11/2022)



```
USE [BetterFuture]
GO
/****** Object:  StoredProcedure [dbo].[UPDATE_DAYS_OF_INTERNSHIP]    Script Date: 11.07.2022 18:24:00 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UPDATE_DAYS_OF_INTERNSHIP]
AS
BEGIN
UPDATE ACCOUNTING_DATA
SET DAYS_OF_INTERNSHIP = DATEDIFF(DAY,ACCOUNTING_DATA.DATE_OF_EMPLOYMENT,GETDATE())
END
```

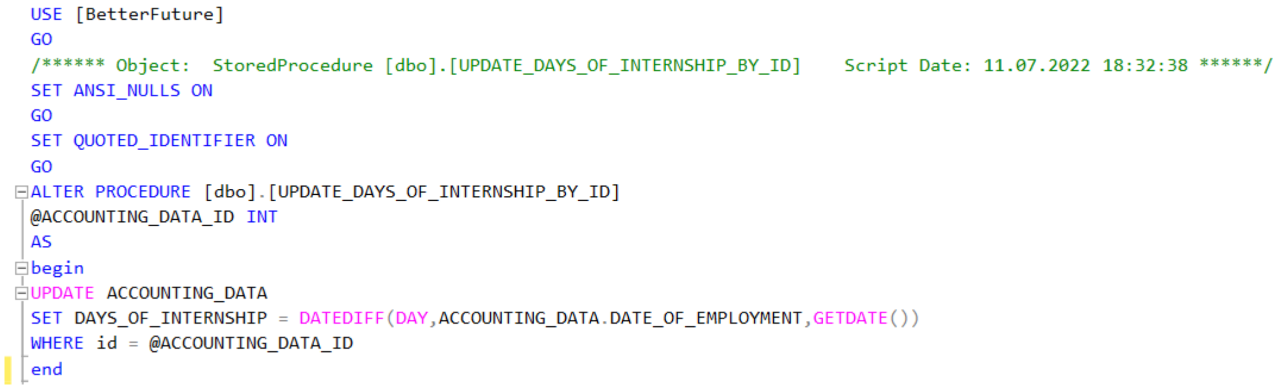


```
SELECT * FROM ACCOUNTING_DATA
```

	id	vacation_days	gross_salary	date_of_employment	days_of_internship	id_employee
1	1	20	3800,00	2021-03-01	497	4
2	2	20	3800,00	2021-12-01	222	6
3	3	26	3600,00	2022-01-01	191	8
4	4	26	4000,00	2021-11-01	252	2
5	5	20	4000,00	2021-02-01	525	3
6	6	20	3600,00	2022-02-01	160	7
7	7	26	3600,00	2021-08-01	344	1
8	8	26	3800,00	2022-04-01	101	5
9	9	26	3800,00	2022-01-06	186	5

UPDATE_DAYS_OF_INTERNSHIP_BY_ID:

This procedure is similar to the above, but includes the “id” parameter in the ACCOUNTING_DATA table.

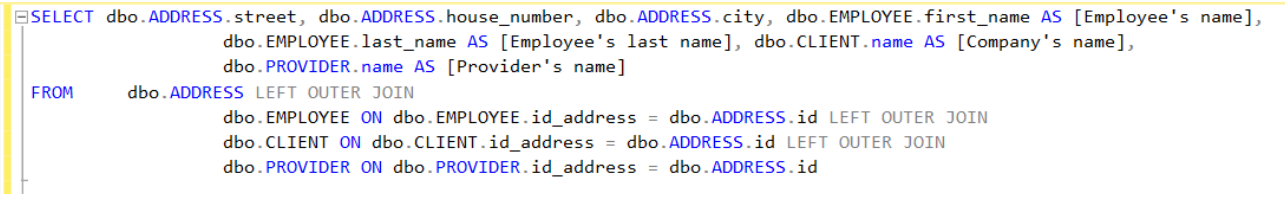


```
USE [BetterFuture]
GO
/****** Object:  StoredProcedure [dbo].[UPDATE_DAYS_OF_INTERNSHIP_BY_ID]    Script Date: 11.07.2022 18:32:38 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UPDATE_DAYS_OF_INTERNSHIP_BY_ID]
@ACCOUNTING_DATA_ID INT
AS
begin
UPDATE ACCOUNTING_DATA
SET DAYS_OF_INTERNSHIP = DATEDIFF(DAY,ACCOUNTING_DATA.DATE_OF_EMPLOYMENT,GETDATE())
WHERE id = @ACCOUNTING_DATA_ID
end
```

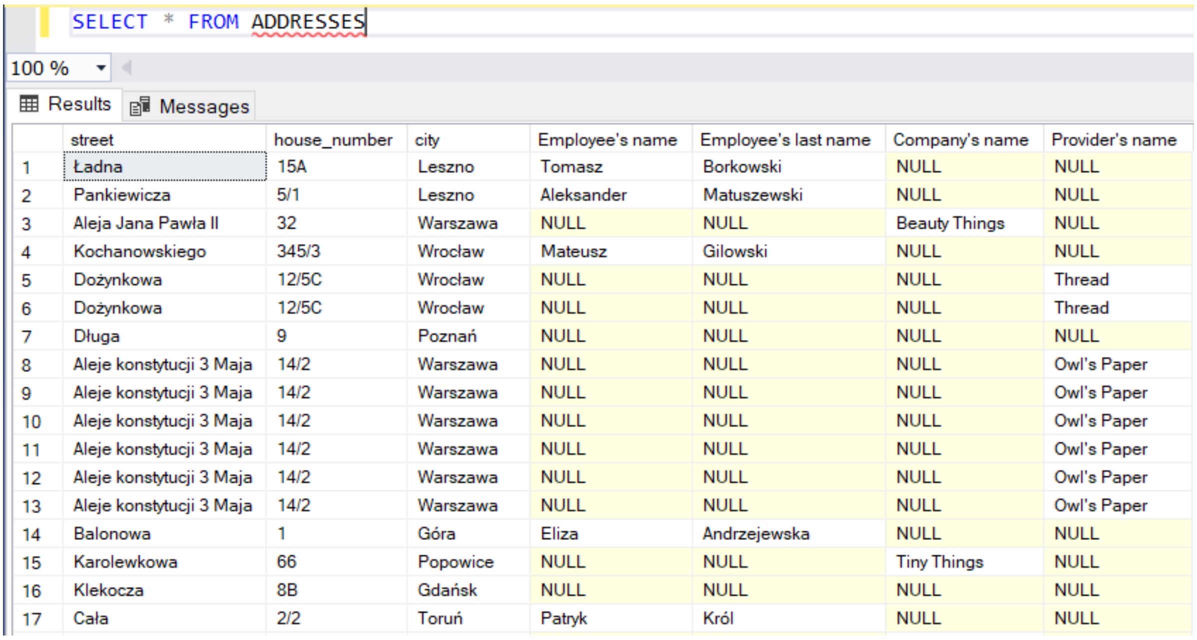
VIEWS

ADDRESSES:

The view shows a list of addresses in the company's database along with the names of customers, employees and suppliers. A code and a fragment of the result is shown below:



```
SELECT dbo.ADDRESS.street, dbo.ADDRESS.house_number, dbo.ADDRESS.city, dbo.EMPLOYEE.first_name AS [Employee's name],
                  dbo.EMPLOYEE.last_name AS [Employee's last name], dbo.CLIENT.name AS [Company's name],
                  dbo.PROVIDER.name AS [Provider's name]
FROM     dbo.ADDRESS LEFT OUTER JOIN
                  dbo.EMPLOYEE ON dbo.EMPLOYEE.id_address = dbo.ADDRESS.id LEFT OUTER JOIN
                  dbo.CLIENT ON dbo.CLIENT.id_address = dbo.ADDRESS.id LEFT OUTER JOIN
                  dbo.PROVIDER ON dbo.PROVIDER.id_address = dbo.ADDRESS.id
```

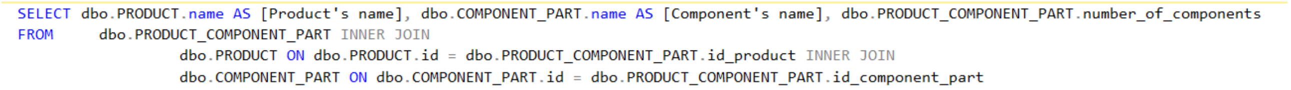


```
SELECT * FROM ADDRESSES
```

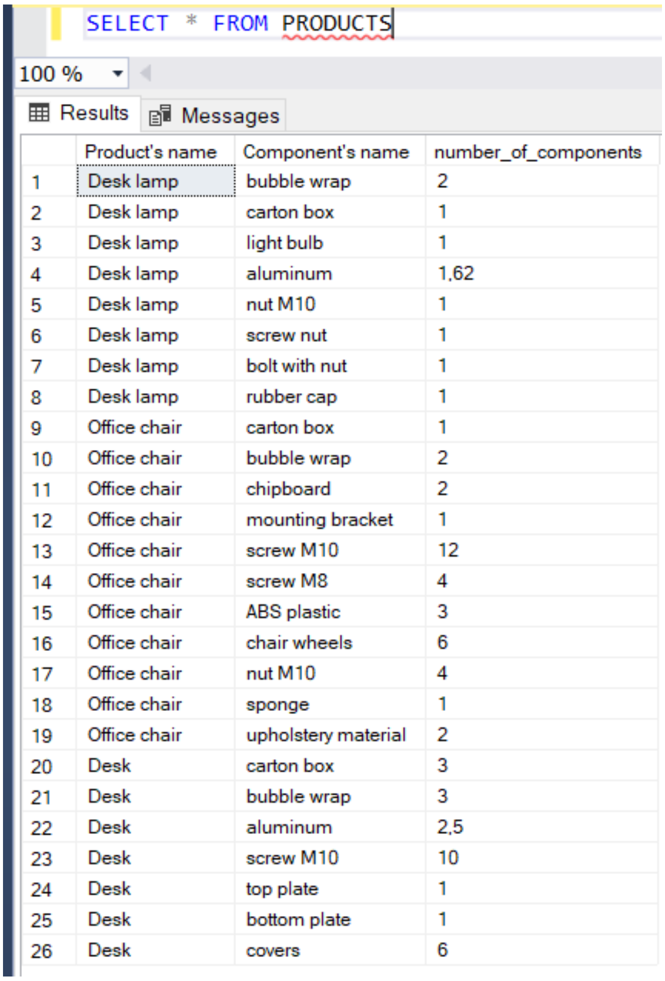
	street	house_number	city	Employee's name	Employee's last name	Company's name	Provider's name
1	Ładna	15A	Leszno	Tomasz	Borkowski	NULL	NULL
2	Pankiewicza	5/1	Leszno	Aleksander	Matuszewski	NULL	NULL
3	Aleja Jana Pawła II	32	Warszawa	NULL	NULL	Beauty Things	NULL
4	Kochanowskiego	345/3	Wrocław	Mateusz	Gilowski	NULL	NULL
5	Dożynkowa	12/5C	Wrocław	NULL	NULL	NULL	Thread
6	Dożynkowa	12/5C	Wrocław	NULL	NULL	NULL	Thread
7	Długa	9	Poznań	NULL	NULL	NULL	NULL
8	Aleje konstytucji 3 Maja	14/2	Warszawa	NULL	NULL	NULL	Owl's Paper
9	Aleje konstytucji 3 Maja	14/2	Warszawa	NULL	NULL	NULL	Owl's Paper
10	Aleje konstytucji 3 Maja	14/2	Warszawa	NULL	NULL	NULL	Owl's Paper
11	Aleje konstytucji 3 Maja	14/2	Warszawa	NULL	NULL	NULL	Owl's Paper
12	Aleje konstytucji 3 Maja	14/2	Warszawa	NULL	NULL	NULL	Owl's Paper
13	Aleje konstytucji 3 Maja	14/2	Warszawa	NULL	NULL	NULL	Owl's Paper
14	Balonowa	1	Góra	Eliza	Andrzejewska	NULL	NULL
15	Karolewkowa	66	Popowice	NULL	NULL	Tiny Things	NULL
16	Klekocza	8B	Gdańsk	NULL	NULL	NULL	NULL
17	Cała	2/2	Toruń	Patryk	Król	NULL	NULL

PRODUCTS:

The view shows the products with their ingredients and their quantity. The code and the result are shown below:



```
SELECT dbo.PRODUCT.name AS [Product's name], dbo.COMPONENT_PART.name AS [Component's name], dbo.PRODUCT_COMPONENT_PART.number_of_components
FROM      dbo.PRODUCT_COMPONENT_PART INNER JOIN
              dbo.PRODUCT ON dbo.PRODUCT.id = dbo.PRODUCT_COMPONENT_PART.id_product INNER JOIN
              dbo.COMPONENT_PART ON dbo.COMPONENT_PART.id = dbo.PRODUCT_COMPONENT_PART.id_component_part
```

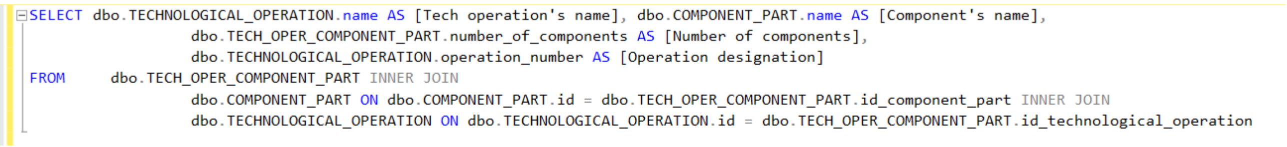


```
SELECT * FROM PRODUCTS
```

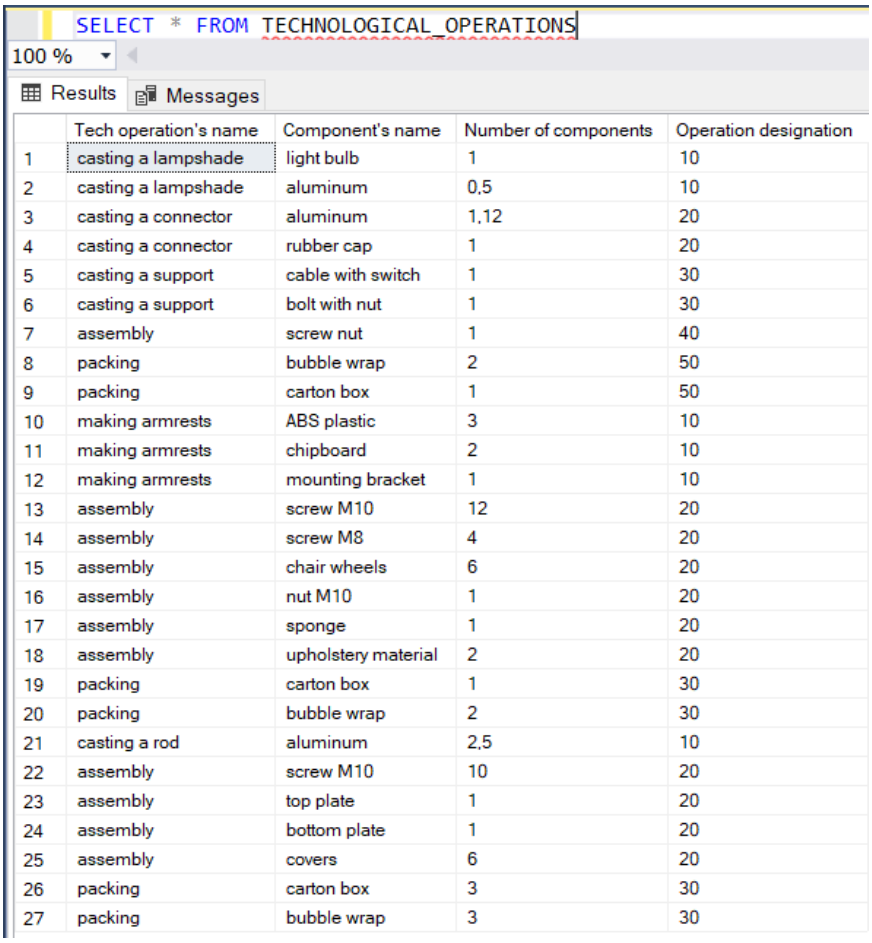
	Product's name	Component's name	number_of_components
1	Desk lamp	bubble wrap	2
2	Desk lamp	carton box	1
3	Desk lamp	light bulb	1
4	Desk lamp	aluminum	1,62
5	Desk lamp	nut M10	1
6	Desk lamp	screw nut	1
7	Desk lamp	bolt with nut	1
8	Desk lamp	rubber cap	1
9	Office chair	carton box	1
10	Office chair	bubble wrap	2
11	Office chair	chipboard	2
12	Office chair	mounting bracket	1
13	Office chair	screw M10	12
14	Office chair	screw M8	4
15	Office chair	ABS plastic	3
16	Office chair	chair wheels	6
17	Office chair	nut M10	4
18	Office chair	sponge	1
19	Office chair	upholstery material	2
20	Desk	carton box	3
21	Desk	bubble wrap	3
22	Desk	aluminum	2,5
23	Desk	screw M10	10
24	Desk	top plate	1
25	Desk	bottom plate	1
26	Desk	covers	6

TECHNOLOGICAL_OPERATIONS:

The view shows a list of technological operations along with numerical designations of these operations and the necessary components with their number. The code and the result are shown below:



```
SELECT dbo.TECHNOLOGICAL_OPERATION.name AS [Tech operation's name], dbo.COMPONENT_PART.name AS [Component's name],
                  dbo.TECH_OPER_COMPONENT_PART.number_of_components AS [Number of components],
                  dbo.TECHNOLOGICAL_OPERATION.operation_number AS [Operation designation]
FROM      dbo.TECH_OPER_COMPONENT_PART INNER JOIN
                  dbo.COMPONENT_PART ON dbo.COMPONENT_PART.id = dbo.TECH_OPER_COMPONENT_PART.id_component_part INNER JOIN
                  dbo.TECHNOLOGICAL_OPERATION ON dbo.TECHNOLOGICAL_OPERATION.id = dbo.TECH_OPER_COMPONENT_PART.id_technological_operation
```



```
SELECT * FROM TECHNOLOGICAL_OPERATIONS
```

	Tech operation's name	Component's name	Number of components	Operation designation
1	casting a lampshade	light bulb	1	10
2	casting a lampshade	aluminum	0,5	10
3	casting a connector	aluminum	1,12	20
4	casting a connector	rubber cap	1	20
5	casting a support	cable with switch	1	30
6	casting a support	bolt with nut	1	30
7	assembly	screw nut	1	40
8	packing	bubble wrap	2	50
9	packing	carton box	1	50
10	making armrests	ABS plastic	3	10
11	making armrests	chipboard	2	10
12	making armrests	mounting bracket	1	10
13	assembly	screw M10	12	20
14	assembly	screw M8	4	20
15	assembly	chair wheels	6	20
16	assembly	nut M10	1	20
17	assembly	sponge	1	20
18	assembly	upholstery material	2	20
19	packing	carton box	1	30
20	packing	bubble wrap	2	30
21	casting a rod	aluminum	2,5	10
22	assembly	screw M10	10	20
23	assembly	top plate	1	20
24	assembly	bottom plate	1	20
25	assembly	covers	6	20
26	packing	carton box	3	30
27	packing	bubble wrap	3	30

Source of images used in the project:
https://www.flaticon.com/