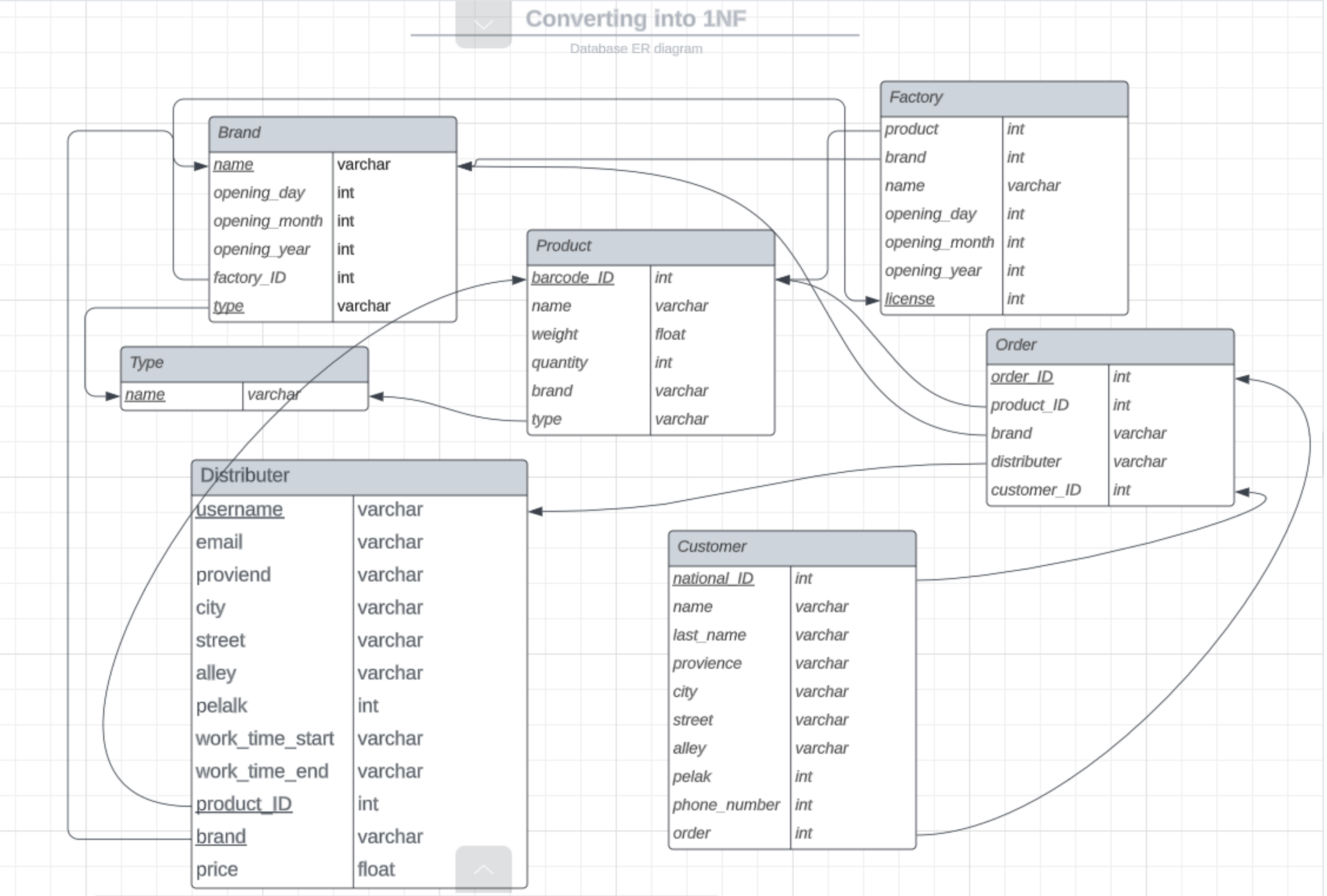
Phase-1

Mostafa Moradi **40031048**

BCNF conversion report

First, we must convert the flat table into 1NF.



Brand	
name	varchar
opening_day	int
opening_month	int
opening_year	int
factory_ID	int
type	varchar

Factory	
product	int
brand	int
name	varchar
opening_day	int
opening_month	int
opening_year	int
license	int

Product	
barcode_ID	int
name	varchar
weight	float
quantity	int
brand	varchar
type	varchar

Type	
name	varchar

Order	
order_ID	int
product_ID	int
brand	varchar
distributer	varchar
customer_ID	int

Distributer	
username	varchar
email	varchar
proviend	varchar
city	varchar
street	varchar
alley	varchar
pelalk	int
work_time_start	varchar
work_time_end	varchar
product_ID	int
brand	varchar
price	float

Customer	
national_ID	int
name	varchar
last_name	varchar
provience	varchar
city	varchar
street	varchar
alley	varchar
pelak	int
phone_number	int
order	int

As you can see in the table, the” Brand” and” Factory” tables had date field in which contained multi-valued data. Also, the” customer” table and the address field got split into its parts. So, they got split into three parts as day, month and the year so it can fit in the First-Normal-Form.

now we must move forward to the second normal form. To ensure that a table is in second normal form (2NF), the following conditions must be met:

1. The table must already be in first normal form (1NF).
2. There should be no partial dependencies in the table, i.e., no non-key attributes should depend on only a portion of the primary key.
3. All non-key attributes should depend on the entire primary key.

To reach to this aim, we apply:

Brand:

Name --> opening_day, opening_month, opening_year, factory_ID, type

Factory:

License --> product, brand, name, opening_day, opening_month, opening_year

Name, opening_day, opening_month, opening_year --> license, brand, product

Order:

Order_ID --> product_ID, brand, distributer, customer_ID

Product_ID, brand, customer_ID --> distributer

Customer_ID -->order_ID

Product:

Barcode_ID , brand--> name, weight, quantity, type

Type:

Name --> name

Distributer:

Username --> email, province, city, street, alley, Pelak, work_time_start, work_time_end, product_ID, brand

Username, prduct_ID, brand --> price

Customer:

National_ID --> name, last_name, provience, city, street, alley, pelak, phone_number, order

So here we can see some conflicts in which won't let the database to be in 2NF. These conflicts are in tables {brand, distributer}; because there are some subkeys form the candidate key in which we can reach to other attributes. To move into 2NF, we have to remove these partial dependencies in this way:

Brand:

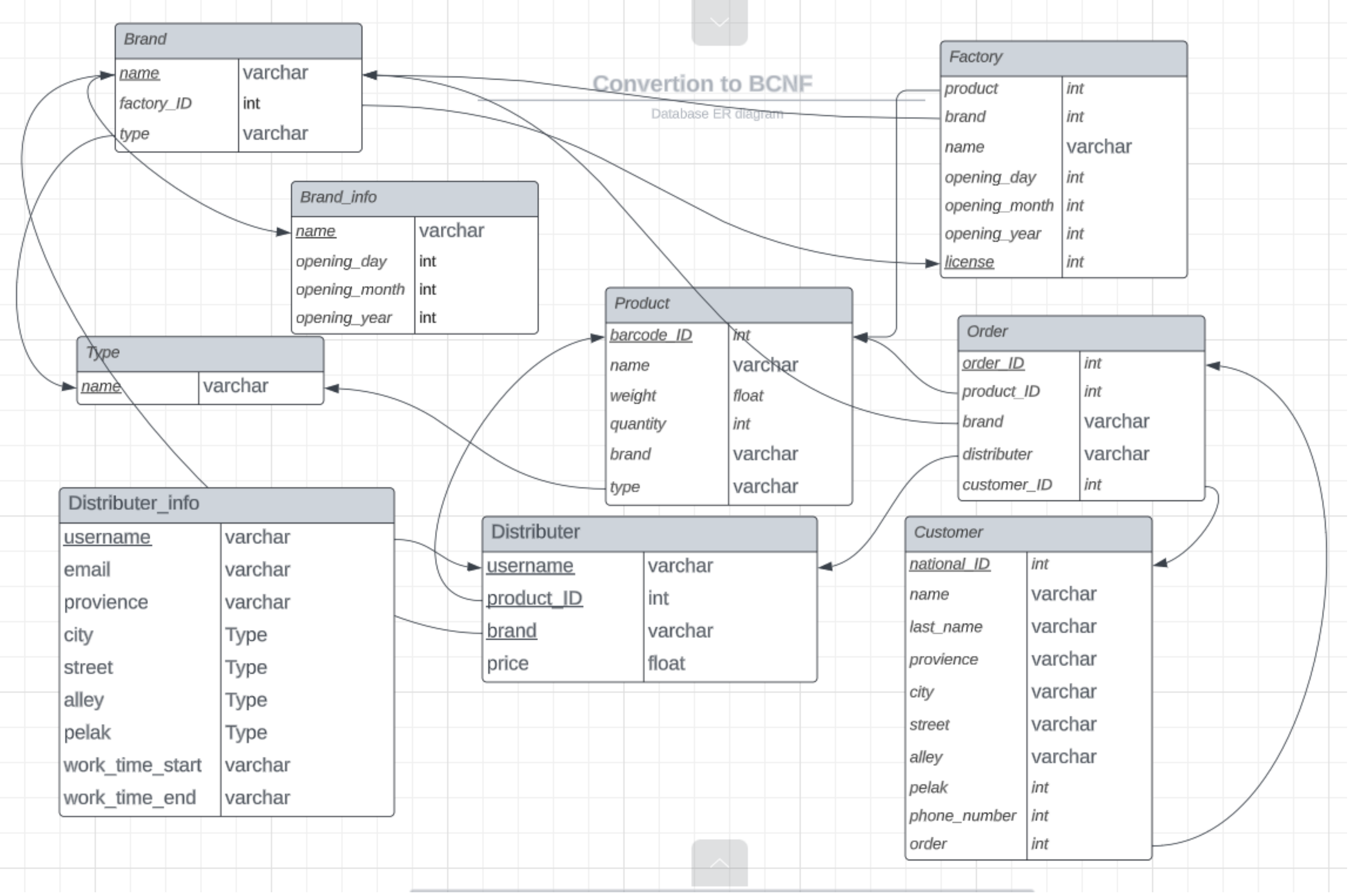
Brand_info(name, opening_day, opening_month, opening_year);

Brand (name, factory_ID, type);

Distributer:

Distributer_info(username, email, province, city, street, alley, pelak, work_time_start,
work_time_end);

Distribuer(username, product_ID, brand, price);



Brand	
name	varchar
factory_ID	int
type	varchar

Factory	
product	int
brand	int
name	varchar
opening_day	int
opening_month	int
opening_year	int
license	int

Brand_info	
name	varchar
opening_day	int
opening_month	int
opening_year	int

Product	
barcode_ID	int
name	varchar
weight	float
quantity	int
brand	varchar
type	varchar

Order	
order_ID	int
product_ID	int
brand	varchar
distributer	varchar
customer_ID	int

Type	
name	varchar

Distributer_info	
username	varchar
email	varchar
provience	varchar
city	Type
street	Type
alley	Type
pelak	Type
work_time_start	varchar
work_time_end	varchar

Distributer	
username	varchar
product_ID	int
brand	varchar
price	float

Customer	
national_ID	int
name	varchar
last_name	varchar
provience	varchar
city	varchar
street	varchar
alley	varchar
pelak	int
phone_number	int
order	int

Tables brand and distributor had some partial dependencies so to fix this conflict we split the tables into two parts so that there exist no partial dependencies.

Now we go through the next normal form, the 3NF; To ensure that a table is in Third Normal Form (3NF), the following conditions must be checked:

4. The table must be in Second Normal Form (2NF).
5. Every non-prime attribute (i.e., attribute that is not part of any candidate key) must be directly dependent on the primary key.
6. There should be no transitive dependency in the table.

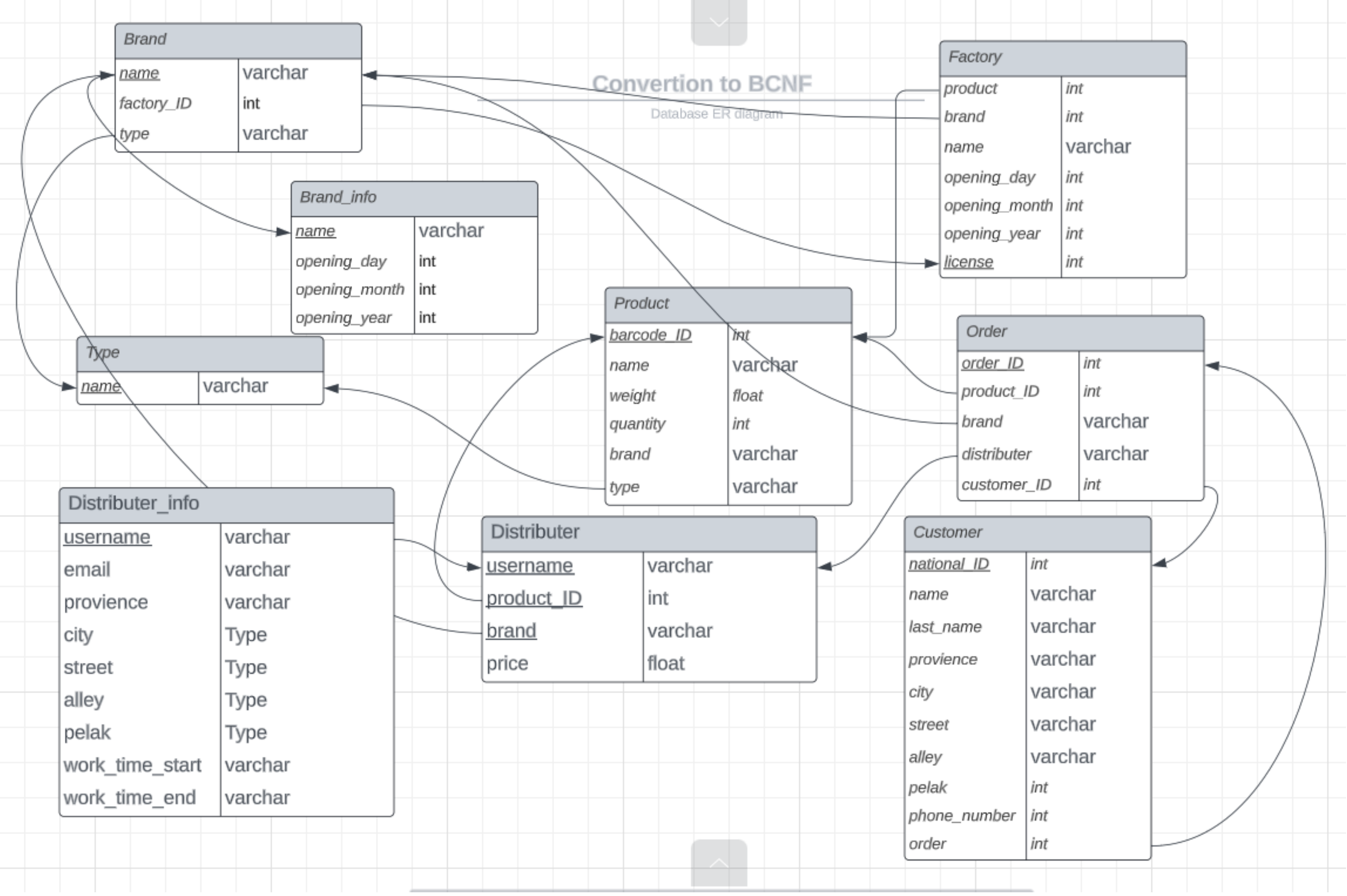
As it is visible in the diagram above, while converting into 2NF, the expected conditions for the Third-normal-form have been created too. The diagram is in 3NF already!

Finally, the last normal form, BCNF; To ensure that a table is in Boyce-Codd Normal Form (BCNF), the following conditions must be checked:

7. Every determinant (i.e., every candidate key) of the table must be a unique and irreducible set of attributes.

8. Every non-prime attribute (i.e., attribute that is not part of any candidate key) must be fully functionally dependent on every candidate key in the table.

Same like the 3NF, without any need for further changes, the diagram is in BCNF too! Because it fits all the conditions for BCNF. So, the expected diagram for the database would be like this:



Convertion to BCNF

Database ER diagram

Brand	
name	varchar
factory_ID	int
type	varchar

Factory	
product	int
brand	int
name	varchar
opening_day	int
opening_month	int
opening_year	int
license	int

Brand_info	
name	varchar
opening_day	int
opening_month	int
opening_year	int

Product	
barcode_ID	int
name	varchar
weight	float
quantity	int
brand	varchar
type	varchar

Type	
name	varchar

Order	
order_ID	int
product_ID	int
brand	varchar
distributer	varchar
customer_ID	int

Distributer_info	
username	varchar
email	varchar
provience	varchar
city	Type
street	Type
alley	Type
pelak	Type
work_time_start	varchar
work_time_end	varchar

Distributer	
username	varchar
product_ID	int
brand	varchar
price	float

Customer	
national_ID	int
name	varchar
last_name	varchar
provience	varchar
city	varchar
street	varchar
alley	varchar
pelak	int
phone_number	int
order	int