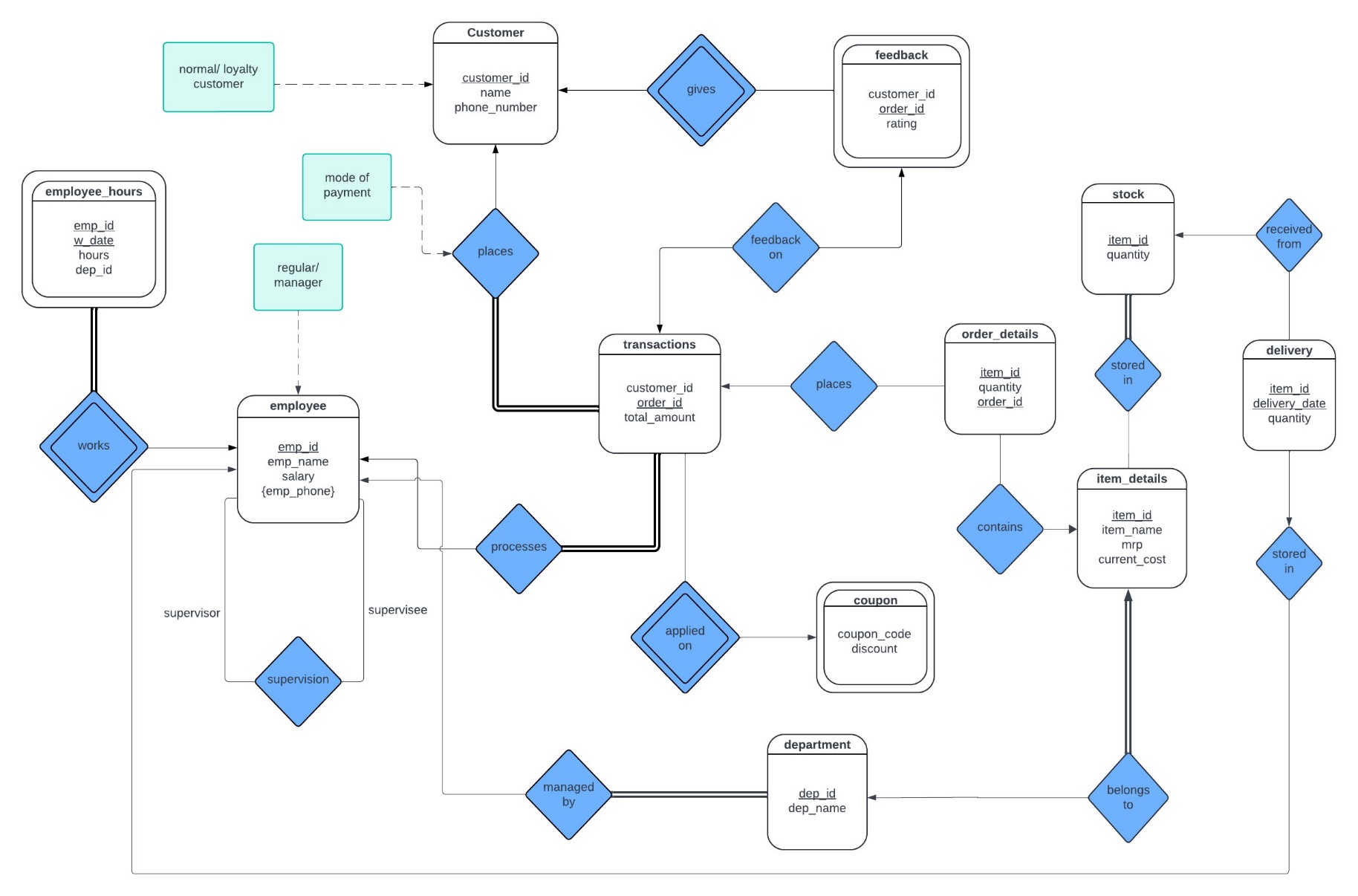
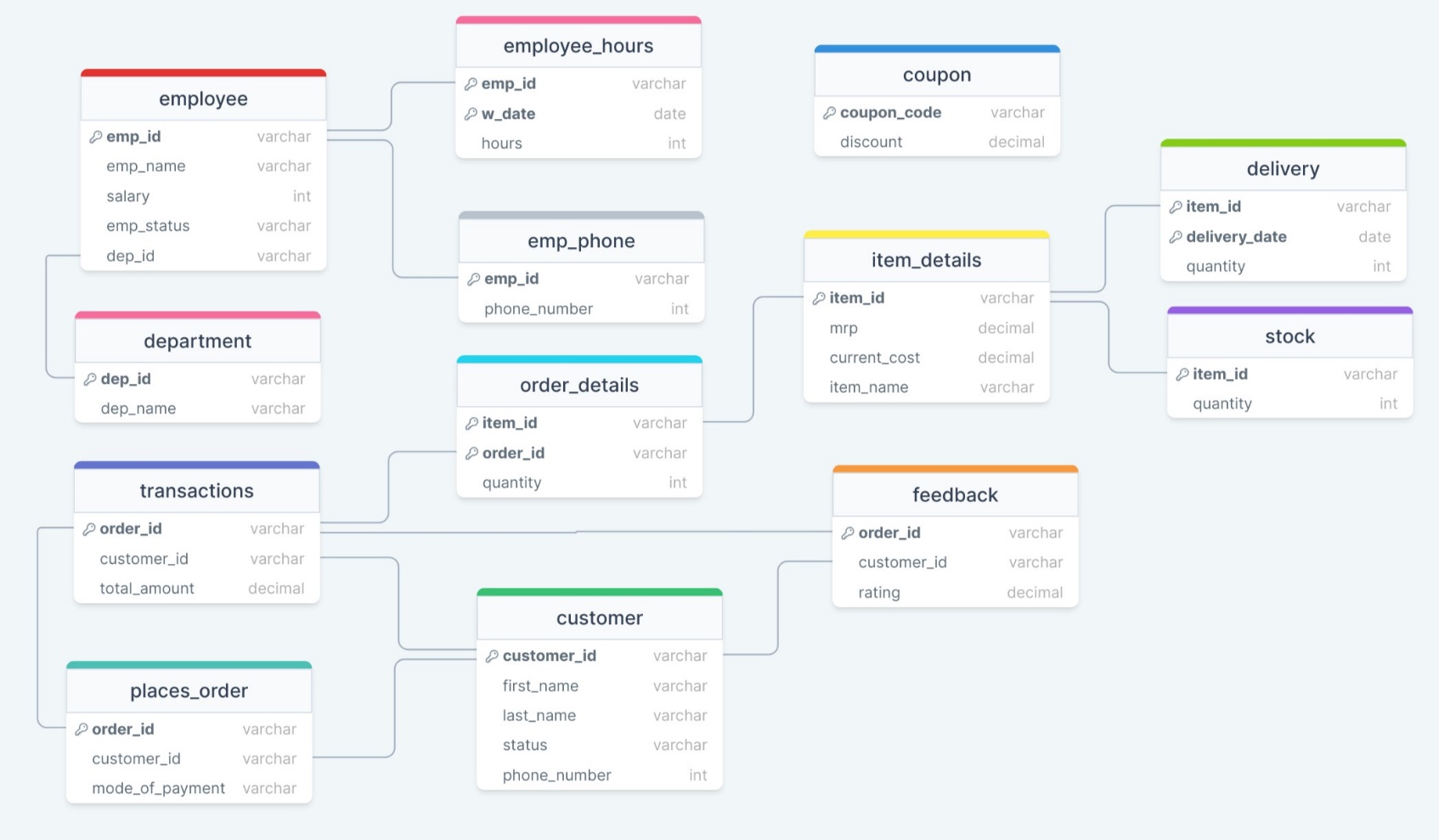
# **Entity Relationship Diagram**



**Relational Schema**



# **Tables gathered from ERD**

Department (dep\_name, **dep\_id**)

Stock (**item\_id**, quantity)

Transactions (customer\_id, **order\_id**, total\_amount)

Coupon (**coupon\_code**, discount)

places\_order (customer\_id, **order\_id**, mode\_of\_payment)

Order\_details (**item\_id**, quantity, **order\_id**)

Delivery (quantity, **item\_id, delivery\_date**)

Customer (name, **customer\_id**, status, phone\_number)

Item\_details (**item\_id**, mrp, current\_cost, item\_name)

Feedback (**order\_id**, customer\_id, rating)

Employee (emp\_name, **emp\_id**, salary, emp\_status, phone\_number, dep\_id)

Employee\_hours (**emp\_id**, hours, dep\_id, **w\_date**)

# **Normalisation**

**Functional dependencies:**

Department table: dep\_id → dep\_name

Stock table: item\_id → quantity

Transactions table: order\_id → customer\_id, total\_amount

Coupon table: coupon\_code → discount

Places\_order table: order\_id → customer\_id, mode\_of\_payment

Order\_details table: (order\_id, item\_id) → quantity

Delivery table: (item\_id, delivery\_date) → quantity

Customer table: customer\_id → name, status, phone number

Item\_details table: item\_id → mrp, current\_cost, item\_name

Feedback table: order\_id → customer\_id, rating

Employee table: emp\_id → emp\_name, salary, emp\_status, phone\_number, dep\_name

Employee\_hours table: (emp\_id, w\_date) → hours, dep\_id

# **1NF:**

To convert the above schema to 1NF, we have to ensure that all the attributes have atomic values.

The tables Department, Stock, Transactions, Coupon, Places\_order, Order\_details, Delivery, Item\_details, Feedback, Employee\_hours have attributes with atomic values. Thus they are in 1NF.

In the **Employee** table, the **phone\_number** is a multi-valued attribute.

We can create a new table Emp\_phone to store multiple phone numbers of the employees

**Emp\_phone(emp\_id, phone\_number)**

In the **customer** table, name is a complex attribute which can be divided into first name and last name.

**Customer(first\_name, last\_name, customer\_id, status, phone\_number)**

**After applying 1NF:**

Department table: dep\_id → dep\_name

Stock table: item\_id → quantity

Transactions table: order\_id → customer\_id, total\_amount

Coupon table: coupon\_code → discount

Places\_order table: order\_id → customer\_id, mode\_of\_payment

Order\_details table: (order\_id, item\_id) → quantity

Delivery table: (item\_id, delivery\_date) → quantity

Customer table: customer\_id → first\_name, last\_name, status, phone\_number

Item\_details table: item\_id → mrp, current\_cost, item\_name

Feedback table: order\_id → customer\_id, rating

Employee table: emp\_id → emp\_name, salary, emp\_status, dep\_name

Emp\_phone table: emp\_id →phone\_number

Employee\_hours table: (emp\_id, w\_date) → hours, dep\_id

## **2NF:**

To convert the given schema to 2NF, we need to ensure that all non-key attributes in each table depend on the entire primary key of that table.

The non-key attributes in the tables Department, Stock, Transactions, Coupon, Places\_order, Order\_details, Delivery, customer, Item\_details, Feedback, employee, emp\_phone depend on the entire primary key. Thus, these tables are in 2NF form.

**Order\_details** Table**:**

Composite Primary Key: (order\_id, item\_id)

Non-Prime Attributes: quantity

quantity is dependent on the entire composite key (order\_id, item\_id) and not on any single part of it, so this table is in 2NF.

**Delivery** Table**:**

Composite Primary Key: (item\_id, delivery\_date)

Non-Prime Attributes: quantity

quantity is dependent on the entire composite key (item\_id, delivery\_date) and not on any single part of it, so this table is in 2NF.

**Employee\_hours** table**:**

dep\_id can be found using only emp\_id which is only a part of the primary attribute. This table is not in 2NF, so we convert to 2NF by shifting dep\_id to the **Employee** table.

Since emp\_id has a total participation with employee, we add it as an attribute to the employee table

**Employee table:**

**emp\_id → emp\_name, salary, emp\_status, phone\_number, dep\_id, dep\_name**

**After 2NF:**

**Department table**: dep\_id → dep\_name

**Stock table**: item\_id → quantity

**Transactions table**: order\_id → customer\_id, total\_amount

**Coupon table**: coupon\_code → discount

**Places\_order table**: order\_id → customer\_id, mode\_of\_payment

**Order\_details table**: (order\_id, item\_id) → quantity

**Delivery table**: (item\_id, delivery\_date) → quantity

**Customer table**: customer\_id → first\_name, last\_name, status, phone\_number

**Item\_details table**: item\_id → mrp, current\_cost, item\_name

**Feedback table**: order\_id → customer\_id, rating

**Employee table**: emp\_id → emp\_name, salary, emp\_status, dep\_id, dep\_name

**Emp\_phone table**: emp\_id →phone\_number

**Employee\_hours table**: (emp\_id, w\_date) → hours

# **3NF**

To convert the schema to 3NF, we need to eliminate transitive dependencies. A transitive dependency occurs when a non-key attribute depends on another non-key attribute rather than the primary key.

The tables Department, Stock, Transactions, Coupon, Places\_order, Order\_details, Delivery, customer, Item\_details, Feedback, employee\_hours, emp\_phone do not have any transitive dependencies. No non-key attributes depend on another non-key attribute.

Thus, they are in 3NF form.

Employee Table (3NF Version):

- **Primary Key**: emp\_id

- **Attributes**: emp\_name, salary, emp\_status, phone\_number, dep\_id, dep\_name

- dep\_id and dep\_name are non-prime attributes. However, we can derive dep\_name from dep\_id. employee table is not in 3nf form. We remove dep\_name to eliminate the transitive dependency.

**After 3NF:**

**Department table**: dep\_id → dep\_name

**Stock table**: item\_id → quantity

**Transactions table**: order\_id → customer\_id, total\_amount

**Coupon table**: coupon\_code → discount

**Places\_order table**: order\_id → customer\_id, mode\_of\_payment

**Order\_details table**: (order\_id, item\_id) → quantity

**Delivery table**: (item\_id, delivery\_date) → quantity

**Customer table**: customer\_id → first\_name, last\_name, status, phone\_number

**Item\_details table**: item\_id → mrp, current\_cost, item\_name

**Feedback table**: order\_id → customer\_id, rating

**Employee table**: emp\_id → emp\_name, salary, emp\_status, dep\_id

**Emp\_phone table**: emp\_id →phone\_number

**Employee\_hours table**: (emp\_id, w\_date) → hours

# **BCNF**

To convert the schema to BCNF, we need to ensure that all functional dependencies are dependencies on the primary key, and there are no non-trivial functional dependencies between non-key attributes.

There are no non-trivial functional dependencies between non-key attributes in the current schema, so it is already in BCNF.

# DDL

create table department(

dep\_name varchar(20),

dep\_id varchar(20),

primary key(dep\_id)

);

create table item\_details(

item\_id varchar(20),

item\_name varchar(50),

mrp number(6,2),

curr\_cost number(6,2),

quantity int,

primary key(item\_id),

);

create table customer(

first\_name varchar(20),

last\_name varchar(20),

customer\_id varchar(20),

phone\_number varchar(10),

status varchar(20),

primary key(customer\_id)

);

create table transactions(

customer\_id varchar(20),

order\_id varchar(20),

total\_amount number(5,2),

primary key(order\_id),

foreign key(customer\_id) references customer\_details

);

create table order\_details(

order\_id varchar(20),

item\_id varchar(20),

quantity int,

foreign key(order\_id) references transactions,

foreign key(item\_id) references item\_details,

primary key(order\_id, item\_id)

);

create table feedback(

order\_id varchar(20),

customer\_id varchar(20),

rating number(1,1),

foreign key(order\_id) references transactions,

foreign key(customer\_id) references customer,

primary key(order\_id)

);

create table employee(

emp\_name varchar(20),

emp\_id varchar(20),

emp\_status varchar(10),

salary int,

dep\_id varchar(20),

primary key(emp\_id),

foreign key(dep\_id) references department

);

create table employee\_hours(

emp\_id varchar(20),

hours int,

w\_date date,

foreign key(emp\_id) references employee\_details,

primary key(emp\_id, w\_date)

);

create table delivery(

quantity int,

item\_id varchar(20),

delivery\_date date,

primary key(quantity, item\_id, delivery\_date),

foreign key(item\_id) references item\_details

)

create table stock(

quantity int,

item\_id varchar(20),

primary key(item\_id),

foreign key(item\_id) references item\_details

)

create table coupon(

discount number(1,2),

coupon\_code varchar(20),

primary key(coupon\_code)

)

create table emp\_phone(

phone\_number int,

emp\_id varchar(20),

primary key(emp\_id),

foreign key(emp\_id) references employee

)

create table places\_order(

order\_id varchar(20),

customer\_id varchar(20),

mode\_of\_payment varchar(20),

primary key(order\_id),

foreign key(order\_id) references transactions,

foreign key(customer\_id) references customer

)