Data Base Project For:

TOYS STORE!

Section No. 2

Group No. 205

Group Members

Jana Abdulrahman Alqurashi [redacted]

Ghadi Mohamed Almoqati [redacted]

Ghala Mohammad Albishri [redacted]

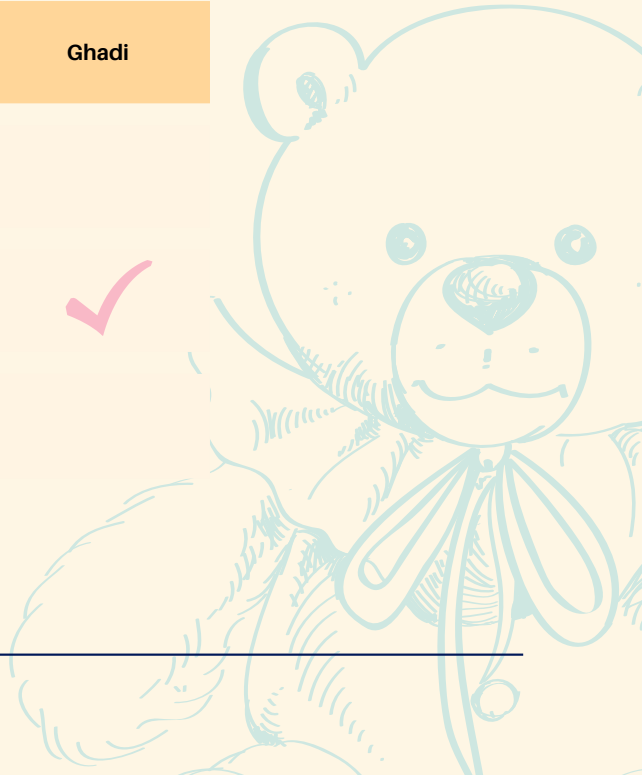
Abeer Motier Alloqmani [redacted]

Maisa Ahmed Alzahrani [redacted]

PHASE 1

Group Tasks Report

Task / Name	Ghala	Maisa	Jana	Abeer	Ghadi
Business Rule	✓	✓	✓		
UML					✓
Chen's notaion				✓	



Description

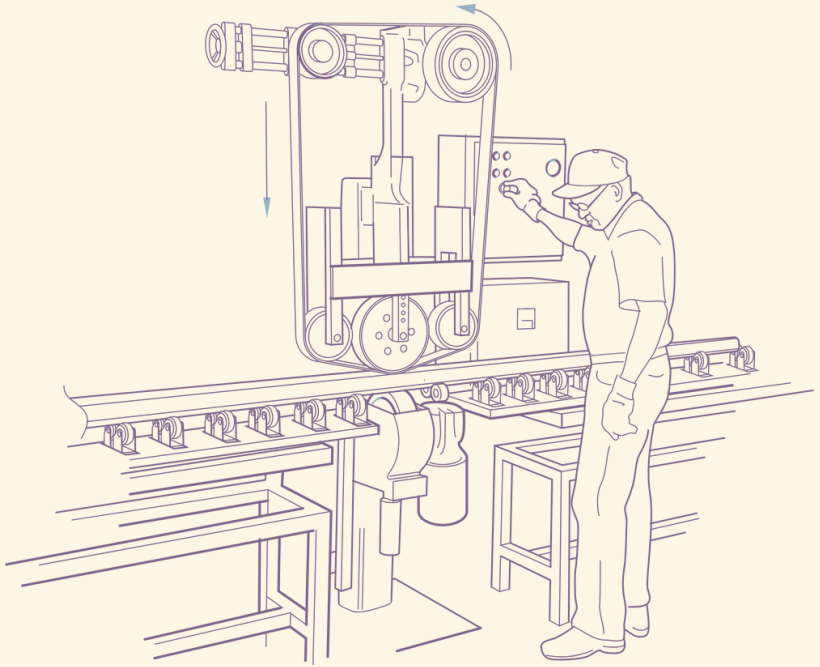
A Toy Store manager asked us to help him with creating a database system for his store and provide it with ER diagrams too, So that he can store and access information easily.
we will design for him a whole database system that can help him keep track of the increasing amount of data and to mange his business efficiently, safely and securely.

Business Rules

The branch has a different attributes such as unique branch Name and an Contact Number and the location. It has a lot of employees, an employee can't work for more than one branch.

Business Rules

Every employee has a name which consists of fname and lName, and a unique identifier number known as EmployeeID, and salary, position and gender . Some employees supervises the machines and make sure it is working well. Each machine can be supervised by many employees.



Business Rules

The Machines produces many toys, each toy has a unique identifier ToyID, and a type, and price. A Toy can be produced by multiple Machines. Every machine has a unique number which is MachineID, a Type, and another attribute for machins Machine Status. Also we will keep a track of the production date of the toys.

Business Rules

other employees are responsible for dealing with customers. Each employee should serves many customers. customers can be served by more than one employee. Some employees are responsible for selling and some are responsible for marketing and advertising directly to the customer.

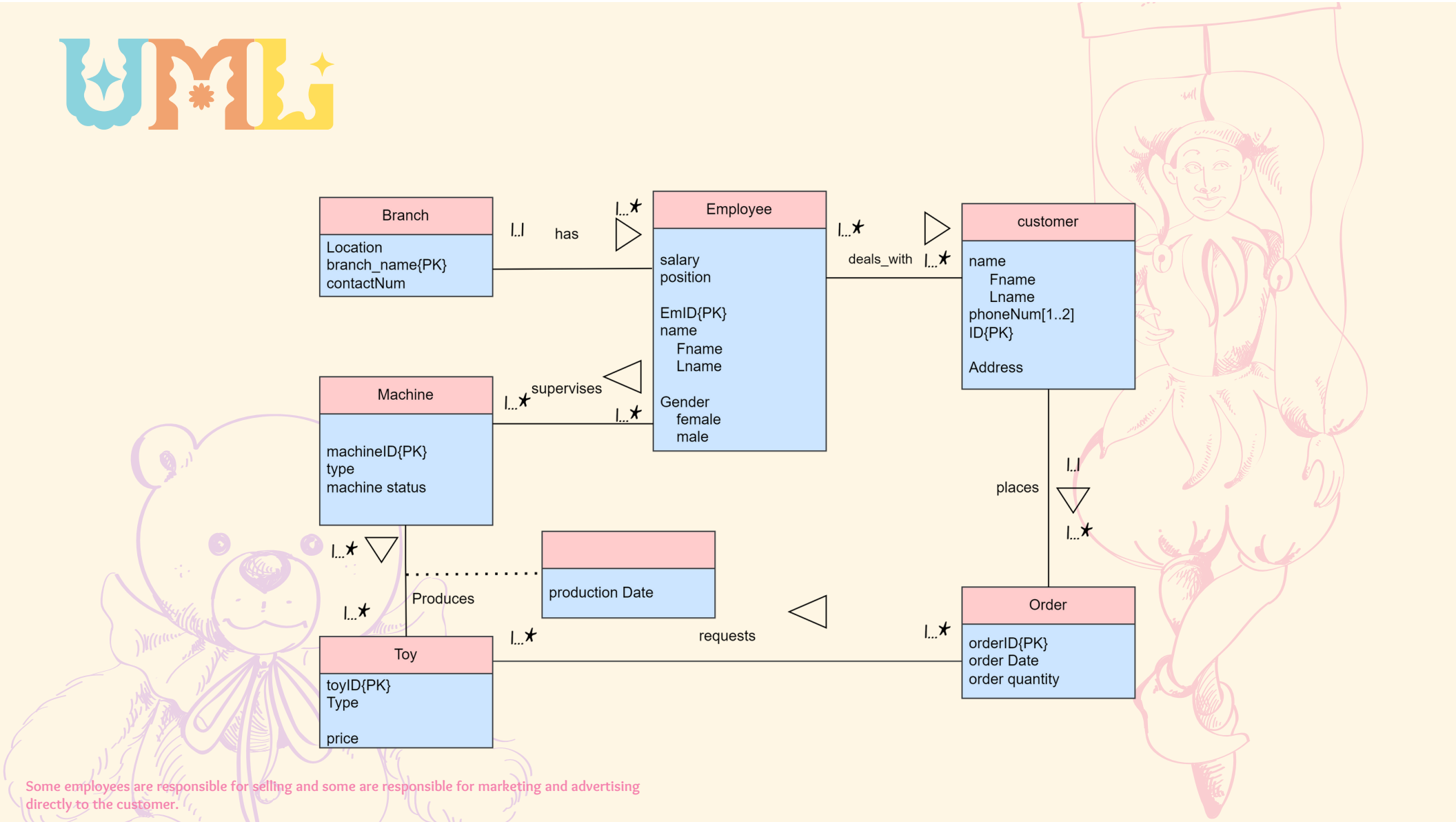
Customer can place many orders, each order should be placed by one and only one customer.

Each customer has a name(fname, lname) , ID, address, and can have one or two phone numbers .

Business Rules

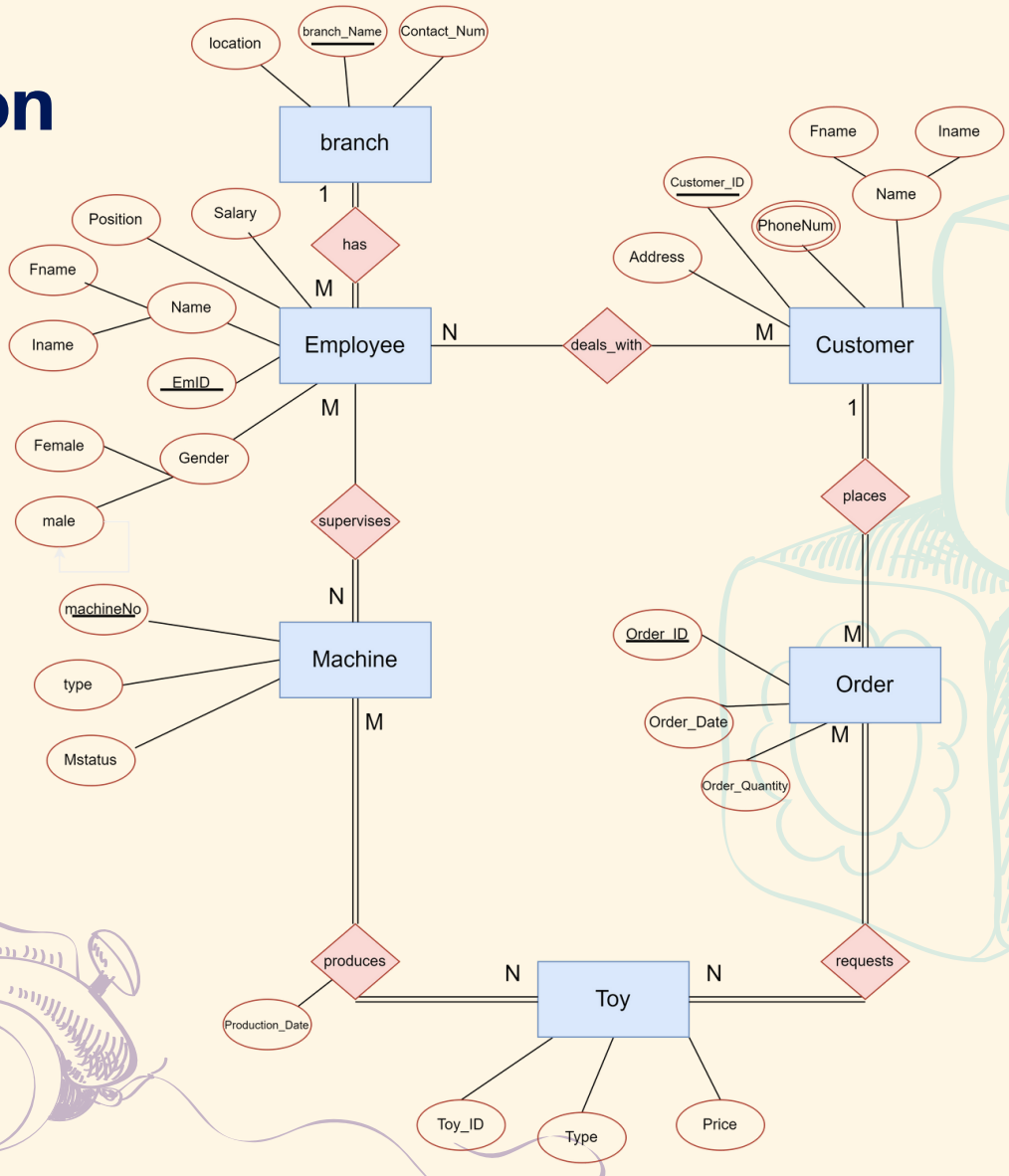
An order has a unique orderID and Order date and a Order quantity.

A customer can order at least one toy. each order requests many toys, the toys can be requested by multiple orders



Some employees are responsible for selling and some are responsible for marketing and advertising directly to the customer.

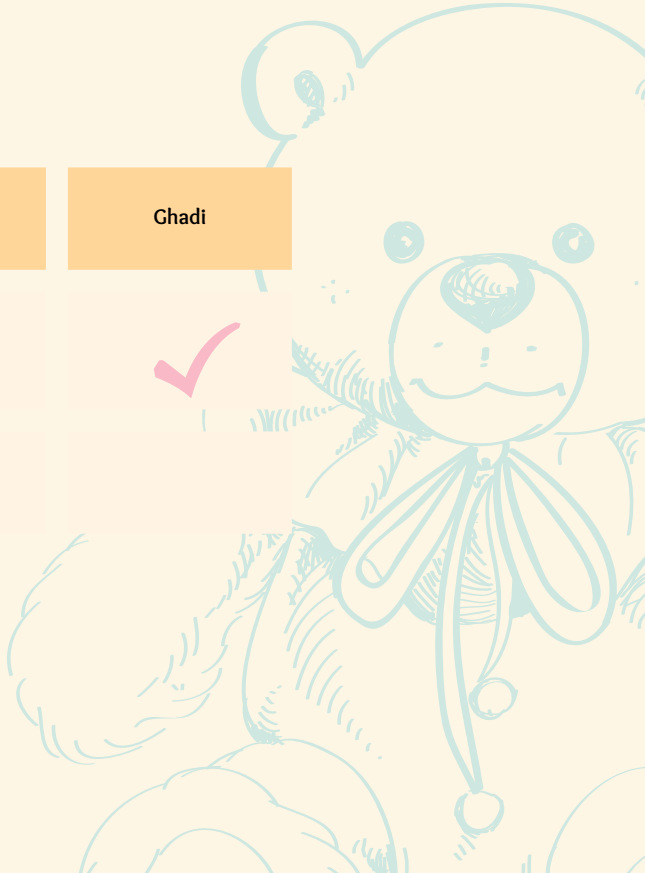
Chen's notaion




PHASE 2

Group Tasks Report

Task / Name	Ghala	Maisa	Jana	Abeer	Ghadi
Mapping of ER into Relational Schema	✓		✓		✓
Normalization		✓		✓	



Relational Schema Mapping

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of weak Entity Entity Types
(No weak entities in the ER Model).

Step 3: Mapping of binary 1:1 Relationship Types
(No binary relationship in the ER Model).

Step 4: Mapping of binary 1:N Relationship Types

Step 5: Mapping of binary M:N Relationship Types

Step 6: Mapping of Multivalued attributes

Step 7: Mapping of N-ary telationship Types
(No n-ary relationship in the ER Model).

Relational Schema Mapping

Step 1: Mapping of Regular Entity Types

BRANCH

branch_name	Contact_Num	location

EMPLOYEE

Fname	Lname	EmployeeID	Salary	Position	Gender

MACHINE

MachineID	Type	Machine_status

TOY

ToyID	Type	Price

CUSTOMER

CostomerID	Fname	Lname	Address

ORDER

OrderID	Order_date	Order_quantity

Relational Schema Mapping

Step 4: Mapping of Binary 1:N Relationship

Employee

Fname	Lname	EmployeeID	female	salary	position	branchN

Machine

machineID	type	machineStatus

branch

Location	branch_name	ContactNum

Customer

Address	CustomerID	Fname	Lname

Order

orderID	orderDate	orderQuantity	customID

Toy

ToyID	ProductionDate	Type	Price

Relational Schema Mapping

Step 5: Mapping of Binary M:N Relationship

SUPERVISES

EmpID	MID	→ Machine(MachineID)

↓ Employee(EmployeeID)

PRODUCES

toyID	MID	productuion_Date

↓ Toy(ToyID) ↓ Machine(MachineID)

REQUESTS

toyID	orderID	→ Order(orderID)

↓ Toy(ToyID)

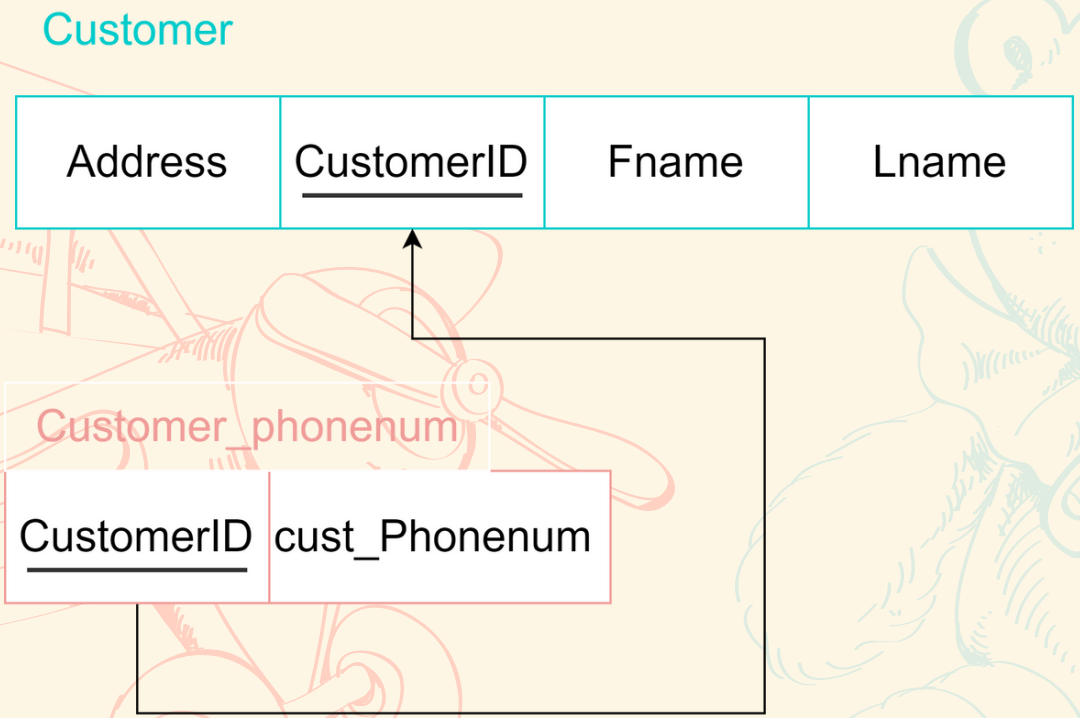
deals_with

custID	empID	→ Employee(EmployeeID)

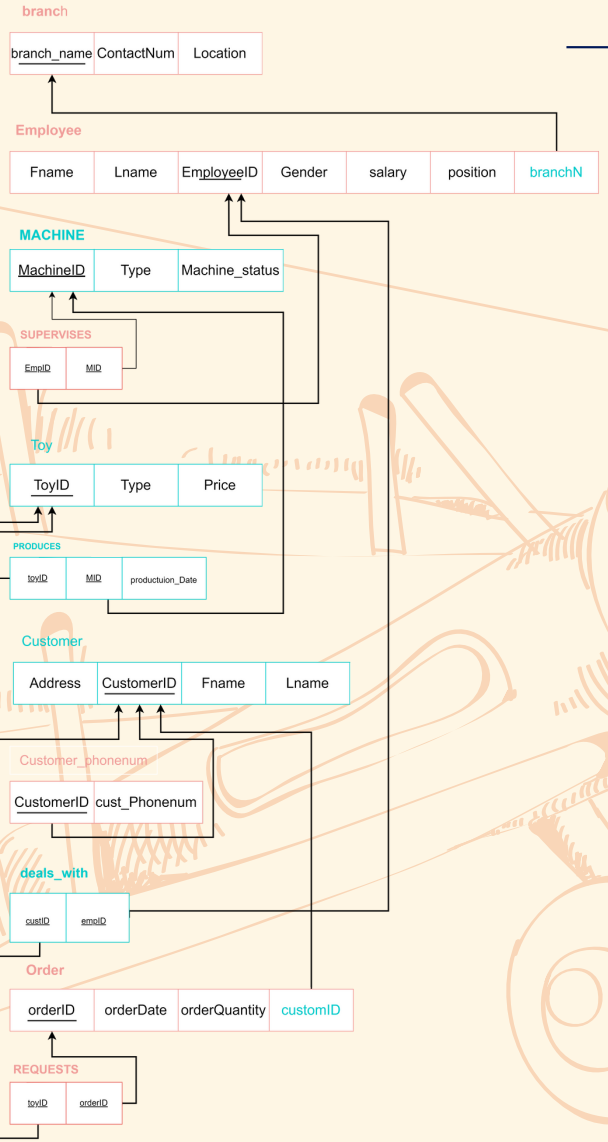
↓ Customer(CustomerID)

Relational Schema Mapping

Step 6 : Mapping of Multivalued Attributes



Final Mapping



Relational Schema Mapping

FINAL MAPPING

https://drive.google.com/file/d/1IJWXWSVBxJ0ikiA3SLNdiCjf7XJ4flBU/view?usp=sharing

Normalization

NF1:

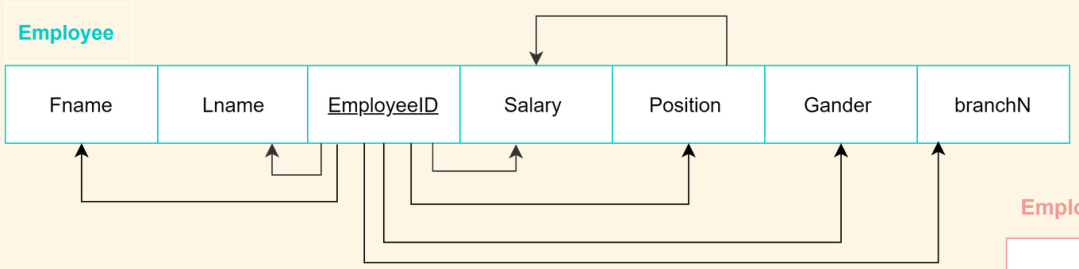
Is there any malti-value attribute ?no
Is there any repeating groups?no

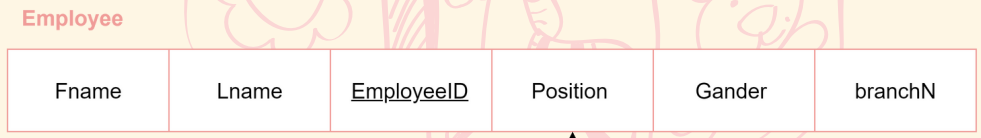
NF2:

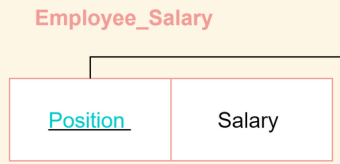
Is it in 1NF? yes
is there any partial dependencies? No

NF3:

Is it in 2NF? Yes
Is there any transitive dependencies? Yes the salary depends on position not only the employee I'd







Normalization

NF1:

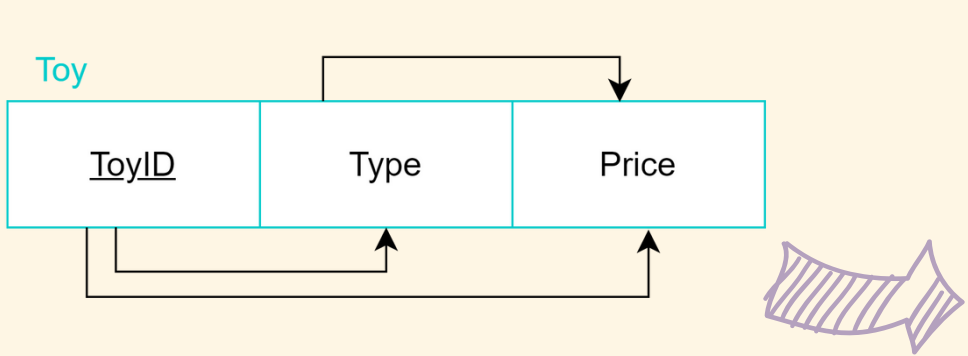
Is there any malti-value attribute ?no
Is there any repeating groups?no

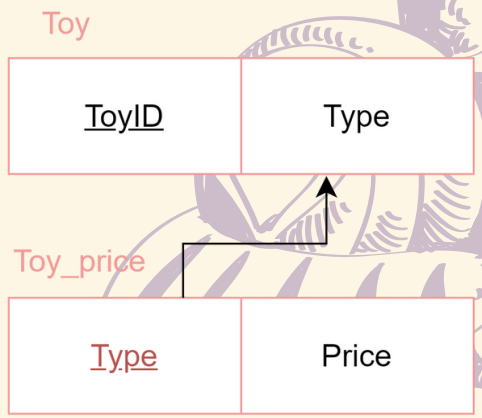
NF2:

Is it in 1NF? yes
is there any partial dependencies? No

NF3:

Is it in 2NF? Yes
Is there any transitive dependencies? Yes the price of the toy depends on both type of toy and ToyID





Normalization

NF1:

Is there any malti-value attribute ?no
Is there any repeating groups?no

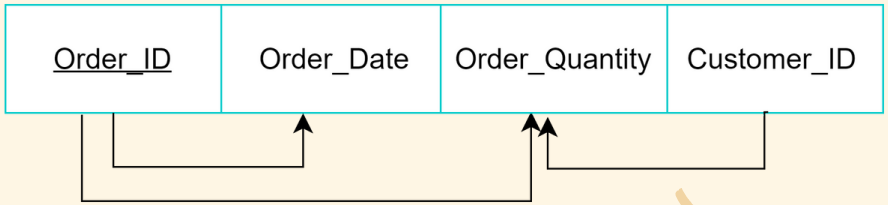
NF2:

Is it in 1NF? yes
is there any partial dependencies? No

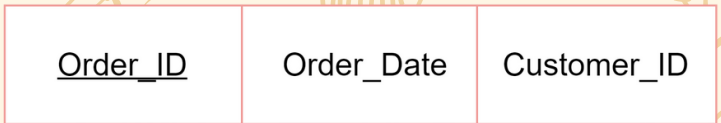
NF3:

Is it in 2NF? Yes
Is there any transitive dependencies? Yes , there is transitive dependency on the customer ID, so we move customerID and quantity to new table

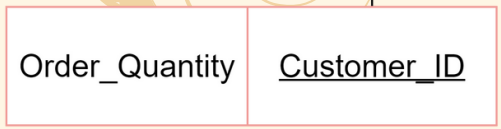
Order



Order



Order_Quantity



Normalization

NF1:

Is there any multi-value attribute ?no
Is there any repeating groups?no

NF2:

Is it in 1NF? yes
is there any partial dependencies? No

NF3:

Is it in 2NF? Yes
Is there any transitive dependencies? NO
Is it in 3NF? Yes

SUPERVISES

EmpID	MID

branch

Location	branch_name	ContactNum

REQUESTS

toyID	orderID

PRODUCES

toyID	MID	Productuion_Date

Normalization

NF1:

Is there any multi-value attribute ?no
Is there any repeating groups?no

NF2:

Is it in 1NF? yes
is there any partial dependencies? No

NF3:

Is it in 2NF? Yes
Is there any transitive dependencies? NO
Is it in 3NF? Yes

Machine

machineID	type	machineStatus

CUSTOMER

Address	CustomerID	Fname	Lname

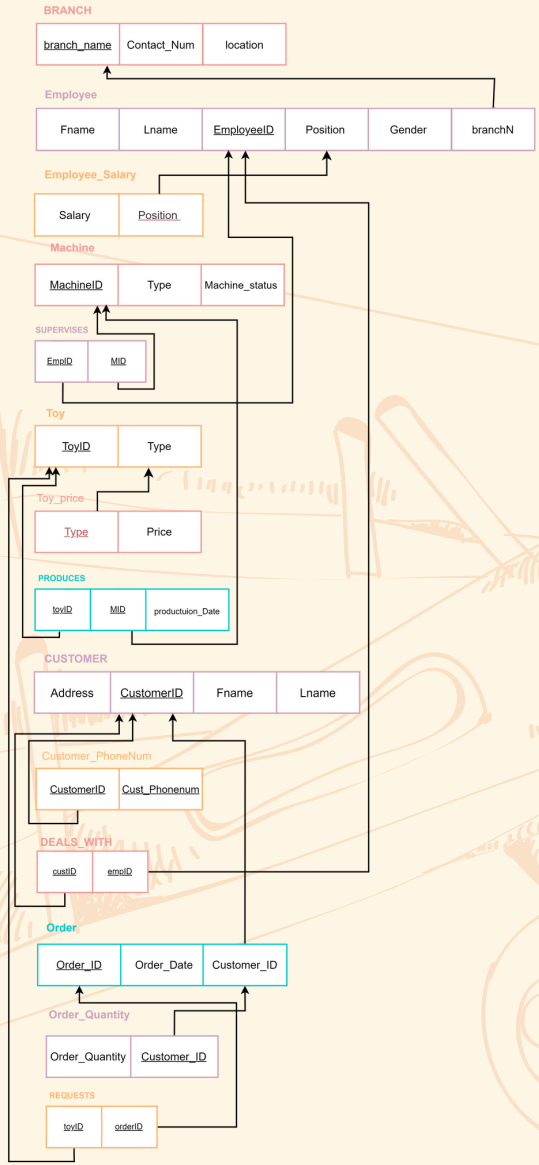
deals_with

custID	empID

CUSTOMER_PHONENUM

CustomerID	Cust_Phonenum

Mapping After Normalization



Mapping After Normalization

https://drive.google.com/file/d/1VkPTXIl0zZrmgLEdCyYvyGlONz2xKNXk/view?usp=sharing

PHASE 3

Group Tasks Report

Task / Name	Ghala	Maisa	Jana	Abeer	Ghadi
creating schema and tables	✓	✓	✓	✓	✓
inserting data into tabels	✓	✓	✓	✓	✓
applying commands: update , delete, where, group by, order by, having , subquery, join	✓	✓	✓	✓	✓

CREATE, SCHEMA, TAPLE

Creating the schema

```
CREATE SCHEMA IF NOT EXISTS `ToyStore`;
USE `ToyStore`;
```

Creating 'branch' table

```
CREATE TABLE branch
(branch_name        VARCHAR(30) NOT NULL,
 contact_num        INT(10),
 location     VARCHAR(50),
 CONSTRAINT branch__PK PRIMARY KEY (branch_name)
);
```

Creating 'employee' table

```
CREATE TABLE employee
(Fname        VARCHAR(15),
 Lname        VARCHAR(15),
 EmployeeID   INT(10) NOT NULL,
 position     VARCHAR(30),
 Gender       VARCHAR(6) CHECK (Gender IN ('female' , 'male')),
 StoreN       VARCHAR(30) ,
 CONSTRAINT EmployeeID_PK PRIMARY KEY (EmployeeID),
 CONSTRAINT StoreN_FK FOREIGN KEY (StoreN) REFERENCES store(store_name) ON DELETE CASCADE ON UPDATE CASCADE ,
 INDEX (position) -- add an index on the 'position' column
);
```

Creating 'employee_salary' table

```
CREATE TABLE employee_salary
(salary     DECIMAL(7,2) CHECK (salary>1000.00),
 position   VARCHAR(30),
  CONSTRAINT position_PK PRIMARY KEY (position),
  CONSTRAINT position_FK FOREIGN KEY (position) REFERENCES employee(position) ON DELETE CASCADE
  ON UPDATE CASCADE
);
```

CREATE, SCHEMA, TAPLE

```
CREATE TABLE machine
(MachineID INT(5) NOT NULL,
Type VARCHAR(25),
Machine_status  VARCHAR(30),
CONSTRAINT MCHINE_PK PRIMARY KEY (MachineID),
INDEX (Type));
```

Creating 'machine' table

Creating 'supervises' table

```
CREATE TABLE supervises
(EmpID INT(10) NOT NULL,
MID INT(5) NOT NULL,
CONSTRAINT suoervises_PK PRIMARY KEY (EmpID ,MID),
CONSTRAINT supervises_FK1 FOREIGN KEY (EmpID) REFERENCES employee (EmployeeID) ON DELETE CASCADE,
CONSTRAINT supervises_FK2 FOREIGN KEY (MID) REFERENCES machine (MachineID) ON DELETE CASCADE);
```

```
CREATE TABLE toy(
toyID  INT (5) NOT NULL ,
type   VARCHAR(25) NOT NULL,
CONSTRAINT toy_PK1 PRIMARY KEY(toyID),
iNDEX (type)
);
```

Creating 'toy' table

CREATE, SCHEMA, TAPLE

Creating 'toy_price' table

```
CREATE TABLE toy_price(
type VARCHAR (25) NOT NULL ,
price DECIMAL (5,2) NOT NULL,
CONSTRAINT toy_price_PK1 PRIMARY KEY(type),
CONSTRAINT toy_price_FK1 FOREIGN KEY (type) REFERENCES toy(type)
ON DELETE CASCADE
ON UPDATE CASCADE
);
```

Creating 'produces' table

```
CREATE TABLE produces(
toyID  INT(5)NOT NULL ,
MID    INT(5)NOT NULL ,
production_Date DATE NOT NULL,
CONSTRAINT produces_PK PRIMARY KEY (toyID ,MID),
CONSTRAINT produces_FK1 FOREIGN KEY (toyID) REFERENCES toy(toyID)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT produces_FK2 FOREIGN KEY (MID) REFERENCES machine(MachineID)
ON DELETE CASCADE
ON UPDATE CASCADE
);
```

Creating 'customer' table

```
CREATE TABLE customer (
    CustomerID INT(10) NOT NULL,
    Fname VARCHAR(15),
    Lname VARCHAR(15),
    Address VARCHAR(50) NOT NULL,
    CONSTRAINT CUSTOMER_PK PRIMARY KEY (CustomerID)
);
```

CREATE, SCHEMA, TAPLE

```
CREATE TABLE Customer_phonenum (
    CustomerID INT(10) NOT NULL,
    Cust_Phonenum INT(10) NOT NULL,
    CONSTRAINT Customer_phonenum_PK PRIMARY KEY (Cust_Phonenum , CustomerID),
    CONSTRAINT Customer_phonenum_FK FOREIGN KEY (CustomerID)
        REFERENCES customer (CustomerID)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);
```

Creating 'customer_phonenum' table

```
CREATE TABLE deals_with (
    CustID INT(10) NOT NULL,
    EmpID INT(10) NOT NULL,
    CONSTRAINT deal_with_PK PRIMARY KEY (CustID , EmpID),
    CONSTRAINT deal_with_FK1 FOREIGN KEY (CustID) REFERENCES customer (CustomerID)
 ON UPDATE CASCADE
 ON DELETE CASCADE,
    CONSTRAINT deal_with_FK2 FOREIGN KEY (EmpID) REFERENCES employee (EmployeeID)
 ON UPDATE CASCADE
 ON DELETE CASCADE

);
```

Creating 'deals_with' table

```
CREATE TABLE `order` (
    Order_ID INT(10) NOT NULL,
    Order_Date DATE NOT NULL,
    Customer_ID INT(10) NOT NULL,
    CONSTRAINT order_PK PRIMARY KEY (Order_ID),
    CONSTRAINT ORDER_FK FOREIGN KEY (Customer_ID) REFERENCES customer (CustomerID) ON UPDATE CASCADE
);
```

Creating 'order' table

CREATE, SCHEMA, TAPLE

Creating 'order_quantity' table

```
CREATE TABLE order_quantity (
    Customer_ID INT(10) NOT NULL,
    order_quantity INT (100) NOT NULL,
    CONSTRAINT order_quantity_PK PRIMARY KEY (Customer_ID),
    CONSTRAINT order_quantity_FK FOREIGN KEY (Customer_ID) REFERENCES `order` (Customer_ID) ON UPDATE CASCADE
);
```

Creating 'requests' table

```
CREATE TABLE requests (
    toyID INT(5) NOT NULL,
    orderID INT(10) NOT NULL,
    CONSTRAINT requests_PK PRIMARY KEY (toyID, orderID),
    CONSTRAINT requests_FK1 FOREIGN KEY (toyID) REFERENCES toy (toyID) ON UPDATE CASCADE,
    CONSTRAINT requests_FK2 FOREIGN KEY (orderID) REFERENCES `order` (Order_ID) ON UPDATE CASCADE
);
```

insert

```sql
INSERT INTO branch
VALUES ('happy world' , 0568901726,'makkah-alzaher'),
('Toy Oasis' , 0238901326,'riyadh-arimal'),
('Toytopia' , 0138931326,'makkah-alzaher'),
('Wonder World' , 0928901326,'madina-albarakah'),
('Playful Pals' , 0498901326,'makkah-alzaher');
SELECT * FROM branch;
```

branch_name	contact_num	location
happy world	568901726	makkah-alzaher
Playful Pals	498901326	makkah-alzaher
Toy Oasis	238901326	riyadh-arimal
Toytopia	138931326	makkah-alzaher
Wonder World	928901326	madina-albarakah
NULL	NULL	NULL

```sql
INSERT INTO employee
VALUES ('Sara' , 'Albishri', 443000001,'Toy Demonstrator' ,'female' , 'Wonder World'),
 ('Yaser' , 'Alqurashi', 443000002,'Online Sales Specialist' ,'male' , 'Toytopia'),
 ('Mayar' , 'Almoqati', 443000003,'Store Manager' ,'female' , 'Playful Pals'),
 ('Faisal' , 'Alloqmani', 443000004,'Marketing Specialist' ,'male' , 'Toytopia'),
 ('Rawan' , 'Alzahrani', 443000005,'Cashier' ,'female' , 'happy world');
 SELECT * FROM employee;
```

Fname	Lname	EmployeeID	position	Gender	branchN
Sara	Albishri	443000001	Toy Demonstrator	female	Wonder World
Yaser	Alqurashi	443000002	Online Sales Specialist	male	Toytopia
Mayar	Almoqati	443000003	Store Manager	female	Playful Pals
Faisal	Alloqmani	443000004	Marketing Specialist	male	Toytopia
Rawan	Alzahrani	443000005	Cashier	female	happy world
NULL	NULL	NULL	NULL	NULL	NULL

```sql
INSERT INTO employee_salary
VALUES (4500.00 ,'Cashier'),
(80000.00 ,'Marketing Specialist'),
(10000.00 ,'Store Manager'),
(7000.00 ,'Online Sales Specialist'),
(5000.00 ,'Toy Demonstrator');
SELECT * FROM employee_salary;
```

salary	position
4500.00	Cashier
80000.00	Marketing Specialist
7000.00	Online Sales Specialist
10000.00	Store Manager
5000.00	Toy Demonstrator
NULL	NULL

Insert

```
INSERT INTO machine
VALUES
(125,'Injection Molding Machine','Operational'),
(134,'CNC Machine','Under Maintenance'),
(441,'Laser Engraving Machine','Under Maintenance'),
(361,'Extrusion Machine','Operational'),
(111,'CNC Machine','Under Maintenance');
SELECT * FROM machine;
```

MachineID	Type	Machine_status
111	CNC Machine	Under Maintenance
125	Injection Molding Machine	Operational
134	CNC Machine	Under Maintenance
361	Extrusion Machine	Operational
441	Laser Engraving Machine	Under Maintenance
NULL	NULL	NULL

```
INSERT INTO supervises
VALUES (443000001,125),
(443000002,111),
(443000003,361),
(443000004,441),
(443000005,134);
SELECT * FROM supervises;
```

EmpID	MID
443000002	111
443000001	125
443000005	134
443000003	361
443000004	441
NULL	NULL

```
INSERT INTO toy VALUES
(10000,'plastic_doll'),
(11000,'sand_bucket'),
(12345,'lego'),
(54321,'fabric_doll'),
(11123,'rubber_ball');
SELECT * FROM toy;
```

toyID	type
54321	fabric_doll
12345	lego
10000	plastic_doll
11123	rubber_ball
11000	sand_bucket
NULL	NULL

Insert

```
INSERT INTO toy_price VALUES
('plastic_doll',30.99),
('sand_bucket',15.50),
('lego',35.00),
('fabric_doll',25.99),
('rubber_ball',10.25);
SELECT * FROM toy_price;
```

type	price
fabric_doll	25.99
lego	35.00
plastic_doll	30.99
rubber_ball	10.25
sand_bucket	15.50
NULL	NULL

```
INSERT INTO produces VALUES
(10000,125,'2023-12-1'),
(11000,134,'2023-12-2'),
(12345,441,'2023-11-15'),
(54321,361,'2023-9-5'),
(11123,111,'2023-1-20');
SELECT * FROM produces;
```

toyID	MID	production_Date
10000	125	2023-12-01
11000	134	2023-12-02
11123	111	2023-01-20
12345	441	2023-11-15
54321	361	2023-09-05
NULL	NULL	NULL

```
INSERT INTO customer
VALUES
(131, 'Reem', 'Saleh', 'Alaziziah'),
(111, 'Farah', 'Khaled', 'Alawali'),
(224, 'Layla', 'Alhaitham', 'Alshoqiyah'),
(298, 'Mona', 'Ahmed', 'Alsharea'),
(188, 'Amal', 'Hassan', 'Alnaseem');
```

CustomerID	Fname	Lname	Address
111	Farah	Khaled	Al awali
131	Reem	Saleh	Al aziziah
188	Amal	Hassan	Al naseem
224	layla	Alhaitham	Al shoqiyah
298	Mona	Ahmed	Al sharea

Insert

```sql
INSERT INTO Customer_phonenum
VALUES
(131, 0589590032),
(111, 0500211024),
(224, 0505052239),
(298, 0589590032),
(188, 0552002179),
(111, 0589590032);
```

	CustomerID	Cust_Phonenum
	111	500211024
	111	500930322
	131	589590032
	188	552002179
	224	505052239
	298	589590032

```sql
INSERT INTO deals_with
VALUES (131 ,443000004),
(111 ,443000005),
(224 ,443000004),
(298 ,443000005),
(188 ,443000005);
SELECT * FROM deals_with;
```

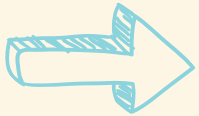
CustID	EmpID
131	443000004
224	443000004
111	443000005
188	443000005
298	443000005
NULL	NULL

```sql
INSERT INTO `order`
VALUES
(101,'2023-5-10',131),
(503,'2023-4-15',111),
(107,'2023-5-20',224),
(211,'2023-5-25',298),
(119,'2024-5-23',188);
SELECT * FROM `order`;
```

Order_ID	Order_Date	Customer_ID
101	2023-05-10	131
107	2023-05-20	224
119	2024-05-23	188
211	2023-05-25	298
503	2023-04-15	111
NULL	NULL	NULL

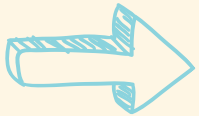
Insert

```
INSERT INTO order_quantity
VALUES
(131,5),
(111,15),
(224,2),
(298,3),
(188,4);
SELECT * FROM order_quantity;
```



Customer_ID	order_quantity
111	15
131	5
188	4
224	2
298	3
NULL	NULL

```
INSERT INTO requests
VALUES
(10000, 101),
(11000, 503),
(12345,107),
(54321, 211),
(11123,119);
SELECT * FROM requests;
```



toyID	orderID
10000	101
12345	107
11123	119
54321	211
11000	503
NULL	NULL

Update

before

after

updating the employees's salary if thier salary less than 5000 adding 2000 to it

salary	position
4500.00	Cashier
80000.00	Marketing Specialist
7000.00	Online Sales Specialist
10000.00	Store Manager
5000.00	Toy Demonstrator
NULL	NULL

```
-- update 1
UPDATE employee_salary
SET salary = salary +2000
WHERE salary <5000;
select * from employee_salary;
```

salary	position
6500.00	Cashier
80000.00	Marketing Specialist
7000.00	Online Sales Specialist
10000.00	Store Manager
5000.00	Toy Demonstrator
NULL	NULL

updating the toys's prices if it was equal or less than 20 and equal or less than 30 by multiplying it by 1.5

type	price
fabric_doll	25.99
lego	35.00
plastic_doll	30.99
rubber_ball	10.25
sand_bucket	15.50
NULL	NULL

```
-- update 2
UPDATE toy_price
SET price = price *1.5
WHERE price <=20 AND price <=30;
select * from toy_price;
```

type	price
fabric_doll	25.99
lego	35.00
plastic_doll	30.99
rubber_ball	23.07
sand_bucket	23.25
NULL	NULL

Delete

before

after

here in toy table we deleted the row by selecting a specific type ='lego' from 'toy' table

toyID	type
54321	fabric_doll
12345	lego
10000	plastic_doll
11123	rubber_ball
11000	sand_bucket
NULL	NULL

```
-- delete 1
DELETE FROM toy
WHERE type = 'lego';
SELECT * FROM toy;
```

toyID	type
54321	fabric_doll
10000	plastic_doll
11123	rubber_ball
11000	sand_bucket
NULL	NULL

here in customer_phonenum table we deleted the rows of customer's phone numbers by selecting the ID

CustomerID	Cust_Phonenum
111	500211024
111	500930322
131	589590032
188	552002179
224	505052239
298	589590032

```
-- delete 2
DELETE FROM Customer_phonenum
WHERE
    CustomerID = 111;
SELECT * FROM customer_phonenum;
```

CustomerID	Cust_Phonenum
131	589590032
188	552002179
224	505052239
298	589590032
NULL	NULL

Where

here we selected the names of the branches that its location in 'makkah-alzaher'

```
SELECT branch_name AS branchesInMakkah
FROM branch
WHERE
    location = 'makkah-alzaher';
```

branchesInMakkah
happy world
Playful Pals
Toytopia

here we selected the machineIDs for specific machine status (operational)

```
select  MachineID AS OperationalMachines
FROM machine
WHERE Machine_status = 'Operational';
```

OperationalMachines
125
361

here we selected the the IDs of CNC Machines type

```
SELECT MachineID AS MachinesIDs_of_CNCType
FROM machine
WHERE Type ='CNC Machine';
```

MachinesIDs_of_CNCType
111
134

Group by

here we used 'Group by' command to show the number of female employees

```
SELECT COUNT(Gender) AS number_of_female_employees
FROM employee
WHERE Gender IN ('female')
GROUP BY Gender;
```

number_of_female_employees
3

we use the'Group by' to group the toy_price table by type and use the MAX function

-- to find the maximum price of toys in each type

```
SELECT type AS Toy_Type, MAX(price) as max_price
FROM toy_price
GROUP BY type;
```

Toy_Type	max_price
fabric_doll	25.99
plastic_doll	30.99
rubber_ball	23.07
sand_bucket	23.25

Having

Show the sum of order quantity for each customer but only if the sum value is greater than 2

```
SELECT Customer_ID, SUM(order_quantity) as total_quantity
FROM order_quantity
GROUP BY Customer_ID
HAVING SUM(order_quantity) > 2;
```

Customer_ID	total_quantity
111	15
131	5
188	4
298	3

we retrieve the number of employees from 'employee' table in each position using the COUNT() function

```
SELECT employee.position, COUNT(*) as num_employees
FROM employee
GROUP BY employee.position
HAVING COUNT(*) >= 1;
```

position	num_employees
Cashier	1
Marketing Specialist	1
Online Sales Specialist	1
Store Manager	1
Toy Demonstrator	1

Order by

here We arranged the requests details(orderID and toyID) in descending order by toyID

```
SELECT *
FROM toystore.requests
order by toyID DESC;
```

toyID	orderID
54321	211
11123	119
11000	503
10000	101
NULL	NULL

here We arranged the produces table in ascending order by production _Date

```
SELECT *
FROM toystore.produces
ORDER BY production_Date;
```

toyID	MID	production_Date
11123	111	2023-01-20
54321	361	2023-09-05
10000	125	2023-12-01
11000	134	2023-12-02
NULL	NULL	NULL

Subqueries

retrieve customers who made orders for toys produced by machines under maintenance:

```sql
-- 9- (subquery) find the customers who made orders for toys produced by machines under maintenance:
SELECT *
FROM customer
WHERE CustomerID IN (
    SELECT Customer_ID
    FROM `order`
    WHERE Order_ID IN (
        SELECT orderID
        FROM requests
        WHERE toyID IN (
            SELECT toyID
            FROM produces
            WHERE MID IN (
                SELECT MachineID
                FROM machine
                WHERE Machine_status = 'Under Maintenance')
        )
    )
);
```

CustomerID	Fname	Lname	Address
188	Amal	Hassan	Alnaseem
111	Farah	Khaled	Alawali
NULL	NULL	NULL	NULL

Join

retrieve information about orders, and customers who bought it with the type and its price

```
SELECT o.Order_ID, CONCAT( c.Fname, " ",c.Lname) AS CustomerName,
       t.type AS ToyType, tp.price AS ToyPrice, od.order_quantity AS Quantity
FROM `order` o
JOIN customer c ON o.Customer_ID = c.CustomerID
JOIN requests r ON o.Order_ID = r.orderID
JOIN toy t ON r.toyID = t.toyID
JOIN toy_price tp ON t.type = tp.type
JOIN order_quantity od ON o.Customer_ID = od.Customer_ID;
```

Order_ID	CustomerName	ToyType	ToyPrice	Quantity
101	Reem Saleh	plastic_doll	30.99	5
119	Amal Hassan	rubber_ball	23.07	4
211	Mona Ahmed	fabric_doll	25.99	3
503	Farah Khaled	sand_bucket	23.25	15


TOY
THANK YOU