Contents

INTRODUCTION TO THE CASE STUDY

Papajon's Pizza is a pizza restaurant chain where customers can choose the varieties of pizza base and the topping(s) with an 'eat in' and 'home delivery' service from several outlets. The recent lockdowns made an impact on the business and revenue for 'eat in' was decreased whereas the revenue for 'home delivery' services got increased. With the current scenario, company's management wish to organise and make home delivery services more efficient. We as a part of IT team have taken the responsibilities to design a data model to process and monitor the requirement mentioned in the case study.

PHASE I

1) Tabular Format to Identify Entities, Attributes and Relationship

After making careful analysis over the case study, below tabular format displays each entity and its attributes along with the relationship table further in the report.

Outlet				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
outlet_id	Outlet Number	The set of all possible outlet numbers	Number: Size 10	Primary Key
phone_no	Phone Number	Possible values of phone numbers	Number: Size 10	Unique
postcode	Postcode	The set of all possible postcodes	Character: Size 6	Not Null
address	Full Address	The set of all possible address across the nation	Character: Size 200	Not Null
name	Full Name	Full name of the outlet with address indicator. Ex: Papajon's Wembley or Papajon's Westminster	Character: Size 50	Not Null
serving_area_postcode [1..*]	Serving area postcode	The set of all possible postcodes for Papajon's serving area	Character: Size 200	Unique

Pizza_Toppings				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
topping_id	Topping ID	The set of all possible topping ID	Number: Size 10	Primary Key
topping_type	Topping Types	The set of all possible topping type	Character: Size 20	Not Null
topping_price	Topping Price	Possible values of topping price	Number: Size 5,2	Not Null

Pizza_base				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
base_id	Pizza Base ID	The set of all possible pizza base ID	Number: Size 10	Primary Key
base_type	Pizza Base Type	The set of all possible base type	Character: 20 Base Type : thin crust, deep pan, stuffed crust	Not Null
base_price	Pizza Base Price	Possible values of base price	Number: Size 5,2	Not Null

Pizza				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
pizza_id	Pizza ID	The set of all possible pizza ID	Number: Size 10	Primary Key
pizza_price	Pizza Price	Possible values of pizza price	Number: Size 5,2	Not Null
qty	Pizza Quantity	Possible values of pizza quantity	Number: Size 10	Not Null

Drivers				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
driver_id	Driver ID	The set of all possible driver IDs	Number: Size 10	Primary Key
driver_name	Driver Name	Full name of the driver	Character: Size 20	Not Null
driver_address	Driver Address	Full address of the driver	Character: Size 200	Not Null
driver_DOB	Driver's Date of Birth	Possible values of driver's date of birth	Date: DD/MM/YYYY	Not Null
driving_license_ number	Driver's License Number	Possible values of driver's driving license	Characters/Number: Size 15	Unique

Customer				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
cust_no	Customer Number	The set of all possible customer ID	Number: Size 10	Primary Key
reg_date	Date of registration	Possible values of customer registration date	Date: DD/MM/YYYY	Not Null
password	Password	The set of all possible password characters	Characters/Number: Size 10	Not Null
mob_number	Contact Number	Possible values of customer contact number	Number: Size 10	Unique
email_id	Email ID	Email address of the registered customer	Characters/Number: Size 50	Unique
last_name	Last Name	Last name of the customer	Characters: Size 50	Not Null
first_name	First Name	First name of the customer	Characters: Size 20	Not Null
cust_address	Customer Full Address	The set of all possible address across the nation	Characters: Size 100	Not Null

Delivery_Trip				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
delivery_id	Delivery ID	The set of all possible delivery IDs	Number: Size 10	Primary Key
departure _time	Departure Time	Possible values of departure time	Time: HH:MM:SS Date: DD/MM/YYYY	Not Null
arrival_time	Arrival Time	Possible values of arrival time	Time: HH:MM:SS Date: DD/MM/YYYY	Nullable

Payment				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
payment_id	Payment ID	The set of all possible payment IDs	Number: Size 10	Primary Key
payment_card_ details	Payment card details	Possible values of payment card	Number: Size 10	Unique
card_number	Card Number	Possible values of card number	Number: Size 16	Unique
cardholder_name	Cardholder Name	The name of Cardholder	Character: Size 50	Not Null
cardholder_ address	Cardholder Address	Address of Cardholder	Character: Size 200	Not Null
expiry_date	Expiry Date	Date of expiry of card	Date: DD/MM/YYYY	Not Null
security_number	Security Number	Security number behind the payment card	Number: Size 3	Not Null
card_issue_date	Card Issue Date	Card issuing date	Date: DD/MM/YYYY	Not Null
card_issue_ number	Card issue Number	Card issuing number	Number: Size 10	Not Null
authorisation_ number	Payment Card Authorisation Number	Possible set of details for authorisation number	Number: Size 15	Not Null
payment_type	Payment Type	The set of all payment types	Character: Size 10 (online/cash)	Not Null
payment_amount	Payment Amount	Possible values of final payment amount	Number: Size 10	Not Null
payment_amount_ after_discount	Payment Amount After Discount	Possible values of payment amount after applying discount	Number: Size 5,2	Nullable

Discount_Voucher				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
voucher_id	Voucher ID	The set of all possible voucher ID	Number: Size 10	Primary Key
vouchers_code	Voucher Code	The set of all vouchers code	Character: Size 10	Unique
description	Voucher Description	The details of the voucher which includes info, T&C	Character: Size 100	Not Null
issue_date	Voucher Issue Date	Possible values of discount voucher issue date	Date: DD/MM/YYYY	Not Null
expire_date	Voucher Expire Date	Possible values of discount voucher expiry date	Date: DD/MM/YYYY	Not Null

Motorbikes				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
motorbikes_id	Motorbike ID	The set of all possible motorbikes ID	Number: Size 10	Primary Key
registration_no	Registration Number	Possible values of bike's registration number	Number: Size 10	Unique
engine_size	Engine Size	Possible values of bike's engine size	Number: Size 2,2	Not Null

Order				
Attributes	**Domain Name**	**Meaning**	**Domain Definition**	**Constraints**
order_id	Order ID	The set of all possible order ID	Number: Size 10	Primary Key
unreg_cust_name	Unregistered Customer Full Name	Full name of unregistered customer	Character: Size 20	Not Null
unreg_cust_phone_no	Unregistered Customer Phone Number	Possible values of unregistered customer contact number	Number: Size 10	Unique
order_datetime	Order Date and Time	Possible values of order date and time	Time: HH:MM:SS Date: DD/MM/YYYY	Not Null
order_amt	Order Amount	Possible values of order amount	Number: Size 5,2	Not Null
order_status	Order Status	Possible set of Order status	Character: Size 10	Not Null
collection_time	Order Collection Time	Possible values of order collection time	Time: HH:MM:SS Date: DD/MM/YYYY	Not Null
advance_booking_datetime	Order Advance Booking Date and Time	Possible values of advance order date and time	Time: HH:MM:SS Date: DD/MM/YYYY	Nullable
order_mode	Order Mode	The set of all possible order mode	Number: Size 10	Not Null
delivery_address	Order Delivery Address	Customer full address	Character: Size 200	Nullable
delivery_est_time	Order Delivery Estimated Time	Possible values of order estimated delivery time	Time: HH:MM:SS Date: DD/MM/YYYY	Not Null
order_delivery_time	Actual Order Delivery Time	Possible values of actual order delivery time	Time: HH:MM:SS Date: DD/MM/YYYY	Not Null

Existing Relationships among Entities			
Relationship Name	**Type**	**Entity A**	**Entity B**
places	Binary	Customer	Order
allocate_to	Binary	Order	Outlet
prepares	Binary	Outlet	Pizza
has_a	Binary	Order	Pizza
determines	Ternary	Pizza_topping	Pizza
determines	Ternary	Pizza_base	Pizza
determines	Ternary	Pizza_size	Pizza
available_at	Ternary	Pizza_topping	Pizza
available_at	Ternary	Pizza_base	Pizza
available_at	Ternary	Pizza_size	Pizza
complimentary_to	Binary	Discount_voucher	Customer
late_delivery_discount	Binary	Order	Discount_voucher
apply_to	Binary	Discount_voucher	Payment
confirms	Binary	Payment	Order
include	Binary	Delivery_Trip	Order
assign_to	Binary	Delivery_Trip	Drivers
involve	Binary	Motorbikes	Delivery_trip
use_available	Binary	Drivers	Motorbikes

2) Assumptions

Before designing the EERD, below are some assumptions that has been considered –

1. If there is a pizza, there must be a topping.
   Outlet → Topping (1..*)
2. At least one topping, one base, one size is required to determine a pizza.
3. Outlet should have at least one pizza (base, topping, size)
   Outlet → Pizza (1..*)
4. One outlet in one service area should have at least one pizza
5. Order should have at least one pizza
   Order → Pizza (1..*)
6. For a pizza available in system has a possibility that there is no order for that pizza.
   Pizza → Order (0..*)
7. There is a possibility for order, that there is no delivery. Example: during festive season, with high volume of orders.

8. No delivery trip is possible without bike and driver.
9. Possibility of bike and driver not getting any delivery trip reason being they are newly introduced in the system and no delivery assigned to both.
10. Biker can choose from many available bike options.
11. Bike should have at least one driver to use, and bike can be used by multiple drivers (not at a same time).
12. Driver should choose at least one bike or any available bikes.
13. Each discount voucher is uniquely assigned to the customer.
14. One outlet should have at least one order.

3) Conceptual E-ERD

The EERD is designed using Draw.io tool [2]. Relationships are determined in order to relate the entities using lines and labels. More detail is layered by adding key attributes of entities. Cardinality shows the relationship is 0-1,1-1, 1-many or many-to-many.

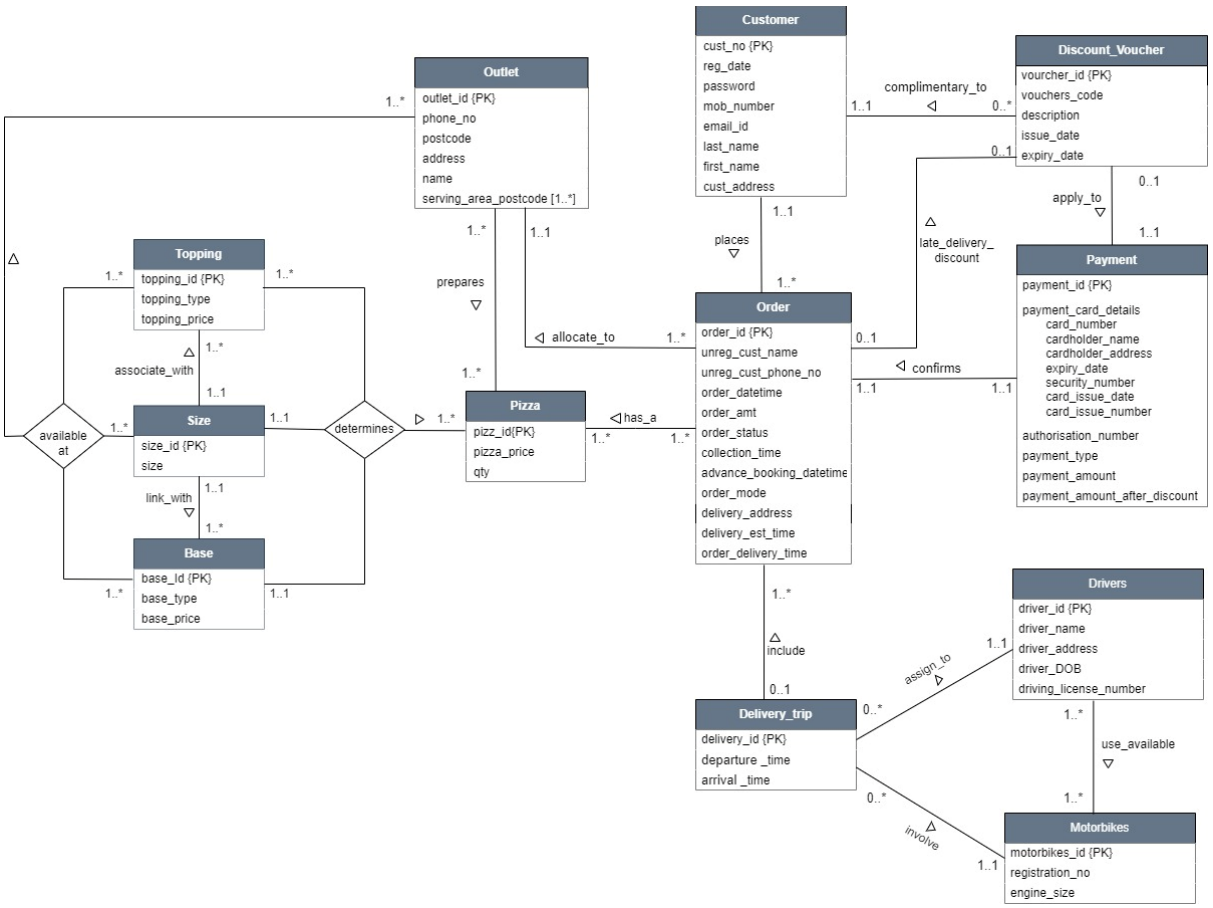

Figure 1: EERD model

PHASE II

We have decided to work with Oracle SQL to design relational schema equivalent to Logical EERD (Figure 1). This part will turn the derived logical model into a relational model. The steps to transform a relational model are as follows:

1) Map Entity to Relation (Relational table)

- To convert logical entities to relational tables, the attributes of strong entities are mapped to relational tables and the relational table is named against the entity name. One of the attributes of the strong entities is chosen as the relational table's primary key.
- The properties of a weak entity are mapped to a relational table with a primary key and a foreign key, which is generated from the primary key of the associated strong entities.
- If the entity has a composite attribute, then only the simple attributes of that composite attribute has been included in the relational table.

After doing a thorough analysis, the following relation tables are derived from the conceptual diagram. The payment entity has a composite attribute which is converted into a simple attribute in the relational table.

customers(*cust_no*, reg_date, password, mob_number, email_id, first_name, last_name, cust_address)

orders(*order_id*, unreg_cust_name, unreg_cust_phone_no, order_datetime, order_amt, order_status, collection_time, advance_booking_datetime, order_mode, delivery_address, delivery_est_time, order_delivery_time)

outlet(*oulet_id*, phone_no, postcode, address, name, serving_area_postcode)

pizza(*pizza_id*, pizza_price, qty)

topping(*topping_id*, topping_type, topping_price)

base(*base_id*, base_type, base_price)

size(*size_id* **,** size**)**

payment(*payment_id*, card_number, cardholder_name, cardholder_address, expiry_date, security_number, card_issue_date, card_issue_number, authorization_number, payment_type, payment_amount, payment_amount_after_discount**)**

discount_voucher(*vourcher_id* **,** vouchers_code, description, issue_date, expiry_date**)**

delivery_trip (*delivery_id*, departure_time, arrival_time**)**

drivers (*driver_id*, driver_name, driver_address, driver_dob, driving_license_number**)**

motorbikes (*motorbikes_id*, registration_no, engine_size**)**

2) Map Entity relationship to Relation

After mapping the Entities to the relational table, the relationship between two entities to the relational tables is mapped as followed:

places(*cust_no*, *order_id*)

has_a(*order_id*, *pizz_id*)

allocate_to(*order_id, outlet_id*)

prepare(*oulet_id*, *pizza_id*)

determines(*pizza_id, topping_id, size_id, base_Id*)

associate_with(*size_id, topping_id*)

link_with(*size_id, base_Id*)

availabile_at(*outlet_id, topping_id, size_id, base_Id*)

prepare(*outlet_id, pizza_id*)

confirm(*payment_id, order_id*)

include(*delivery_id, order_id*)

apply_to(*payment_id, voucher_id*)

late_delivery_discount(*order_id, voucher_id*)

complementary_to(*cust_no, voucher_id*)

assign_to(*delivery_id, driver_id*)

involve(*delivery_id, motorbikes_id*)

use_available(*driver_id, motorbikes_id*)

3) Resulting Schema

The relationships between two rational tables have been be handled as below:

1. **One-to-One relationship:** The relational table at the mandatory end of the relation should be amended by passing the primary key of the other table as a foreign key.

2. **One-to-Many Relationship:** The relational table at the * (Many) end of the relation should be amended by passing the primary key of the other table's primary key as a foreign key.

3. **Many-to-Many Relationship:** The new relational table should be created using the primary keys of both tables and include the required attribute.

Based on the relationship rules, the resulting schema has been derived as follows:

customers (***cust_no***, reg_date, password, mob_number, email_id, first_name, last_name, cust_address)

orders (***order_id,*** unreg_cust_name, unreg_cust_phone_no, order_datetime, order_amt, order_status, collection_time, advance_booking_datetime, order_mode, delivery_address, delivery_est_time, order_delivery_time, ***oulet_id, delivery_id, cust_no)***

outlet (***oulet_id***, phone_no, postcode, address, name, serving_area_postcode)

pizza (***pizza_id***, qty, pizza_price***, base _id***, ***size_id***)

size (*size_id* , size**)**

topping (***topping_id***, topping_type, topping_price***, size_id***)

base (***base _id***, base_type, base_price***, size_id***)

payment (***payment_id,*** card_number, cardholder_name, cardholder_address, expiry_date, security_number, card_issue_date, card_issue_number, authorization_number, payment_type, payment_amount, payment_amount_after_discount, ***order_id,*** , ***voucher_id***)

discount_voucher(***voucher_id,*** vouchers_code, description, issue_date, expiry_date, ***cust_no, order_id***)

delivery_trip (***delivery_id,*** departure_time, arrival_time, orders, ***driver_id, motorbikes_id***)

motorbikes (***motorbikes_id,*** registration_no***,*** engine_size)

drivers (***driver_id,*** driver_name, driver_address, driver_dob, driving_license_number)

driver_motorbikes (***d_m_mapping_id***, ***driver_id, motorbikes_id***)

pizza_topping (***p_t_mapping_id***, ***topping_id, pizza_id***)

pizza_outlet ***(p_out_mapping_id, outlet_id, pizza_id***)

pizza_order (***p_o_mapping_id***, ***order_id, pizza_id***)

available_at (***o_t_s_b_id***, ***topping_id, size_id, outlet_id, base_id***)

4) Primary key and foreign keys

The list of primary keys and foreign keys of the resulting schema is explained in the following tables

Table 1: List of Foreign keys

Referencing Table	Referencing Column (s)	Target Table	Target Column (s)
orders	cust_no	customers	cust_no
orders	outlet_id	outlet	outlet_id
orders	delivery_id	delivery_trip	delivery_id
pizza	base_id	base	base_id
pizza	size_id	size	size_id
topping	size_id	size	size_id
base	size_id	size	size_id
payment	order_id	order	order_id
payment	voucher_id	discount_voucher	voucher _id
discount_voucher	cust_no	customers	cust_no
discount_voucher	order_id	order	order_id
delivery_trip	driver_id	drivers	driver_id
delivery_trip	motorbikes_id	motorbikes	motorbikes_id
driver_motorbikes	driver_id	drivers	driver_id
driver_motorbikes	motorbikes_id	motorbikes	motorbikes_id
pizza_topping	topping_id	topping	topping_id
pizza_topping	pizza_id	pizza	pizza_id
pizza_outlet	outlet_id,	outlet	outlet_id,
pizza_outlet	pizza_id	pizza	pizza_id
pizza_order	order_id	orders	order_id
pizza_order	pizza_id	pizza	pizza_id
available_at	topping_id	topping	topping_id
available_at	size_id	size	size_id
available_at	outlet_id	outlet	outlet_id
available_at	base_id	base	base_id

Table 2: List of Primary Keys

Primary Keys	
customers	**cust_no**
orders	**order_id**
outlet	**outlet_id**
pizza	**pizza_id**
topping	**topping_id**
base	**base_id**
size	**size_id**
payment	**payment_id**
discount_voucher	**voucher_id**
delivery_trip	**delivery_id**
drivers	**driver_id**
motorbikes	**motorbikes_id**

5) Relational Diagram

Due to clogging of the content, Figure 2 is added on page 15 of this report. To visualize relational diagram more clearly, PDF attachment is provided in appendix section of this report.

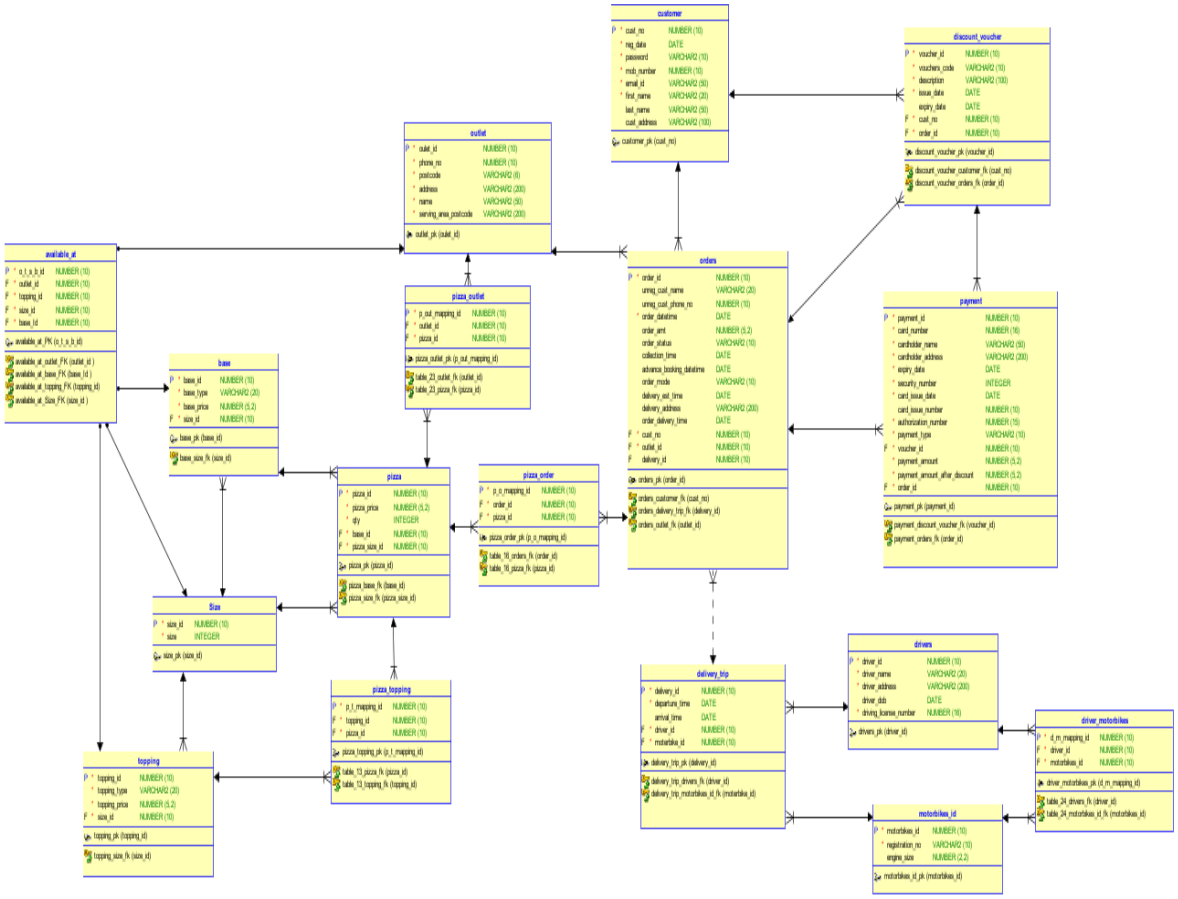

Figure 2: Relation Schema

REFERENCE

[1] Oracle.com. 2022. [online] Available at: <https://www.oracle.com/uk/database/sqldeveloper/>

[2] draw.io. [online] Available at: <https://drawio-app.com/>

APPENDIX

1) EERD Drafts

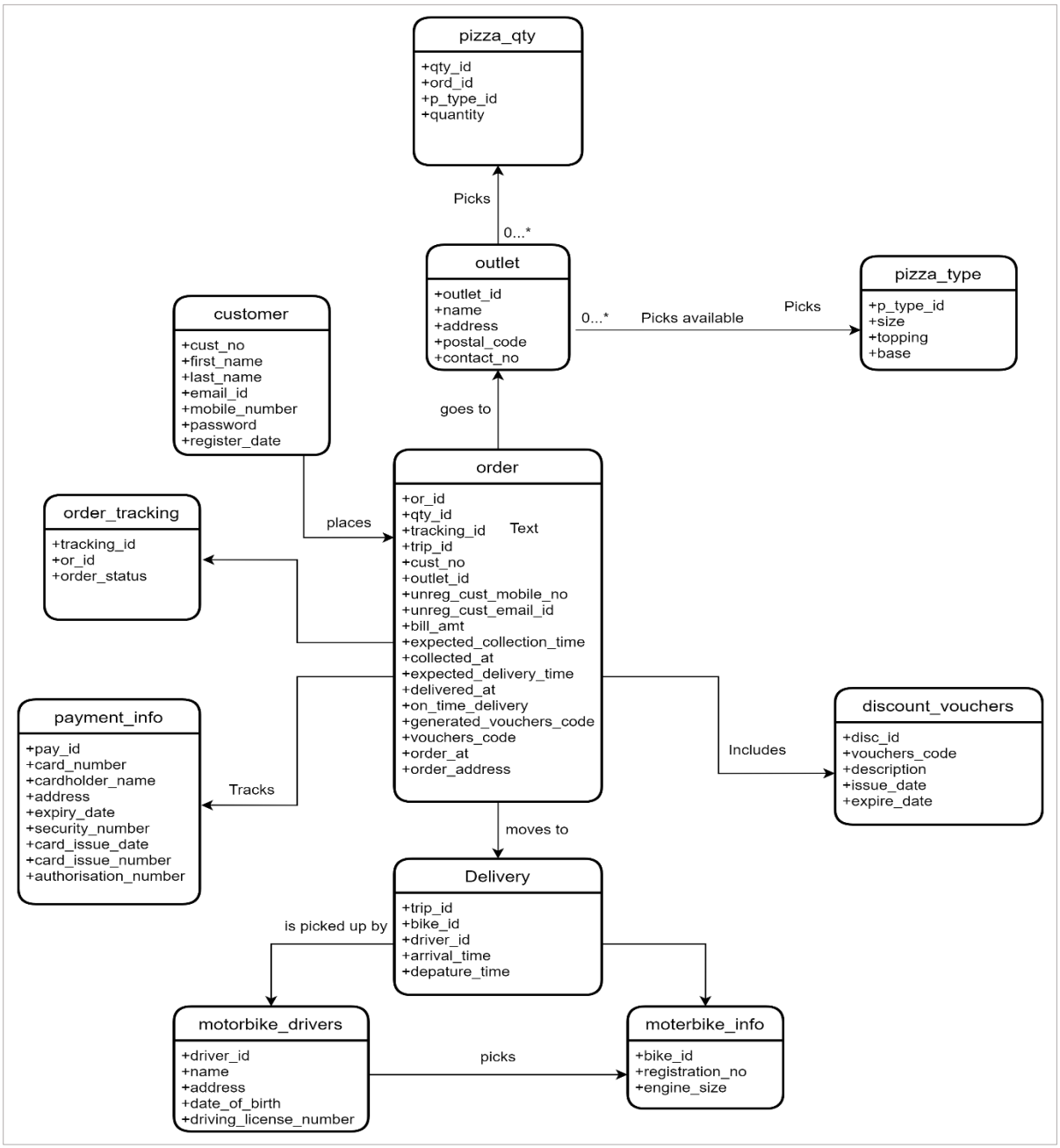

Figure 3: EERD Draft Version 1

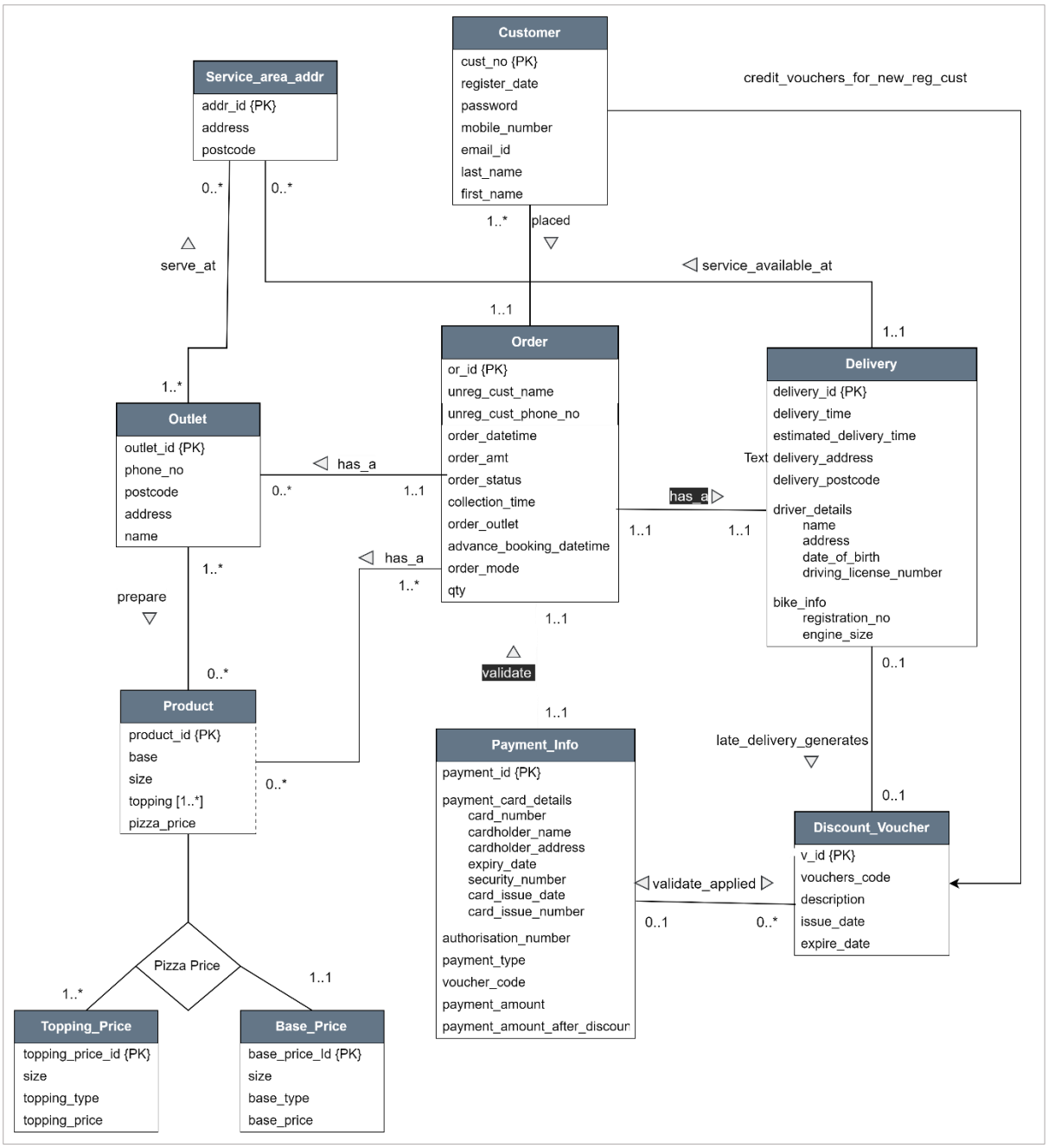

Figure 4: EERD Draft Version 2

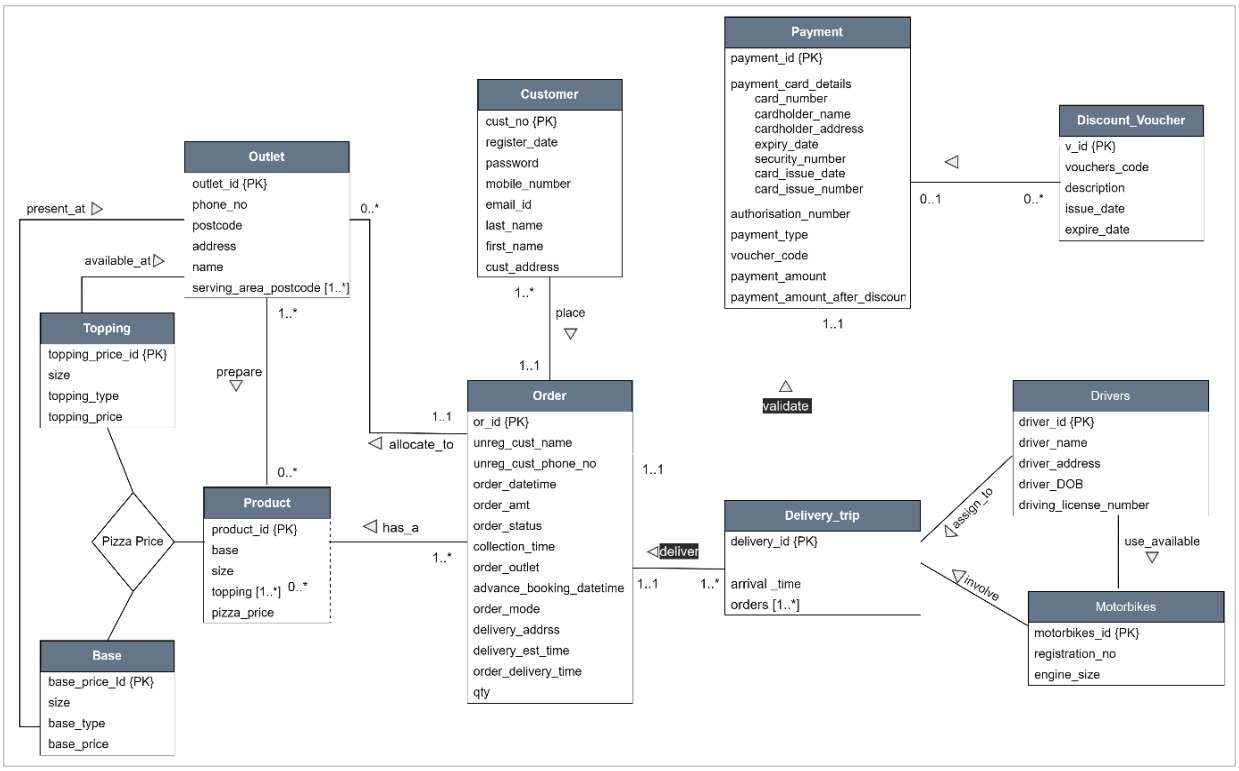

Figure 5: EERD Draft Version 3

2) PDF for Relational Diagram

Relational_Diagram.pdf

3) DDL Script

```
CREATE TABLE base (
   base_id    NUMBER(10) NOT NULL,
   base_type  VARCHAR2(20) NOT NULL,
   base_price NUMBER(5, 2) NOT NULL,
   size_id    NUMBER(10) NOT NULL
);
ALTER TABLE base ADD CONSTRAINT
base_pk PRIMARY KEY ( base_id );
```

```
CREATE TABLE customer (
   cust_no      NUMBER(10) NOT NULL,
   reg_date     DATE NOT NULL,
   password     VARCHAR2(10) NOT NULL,
   mob_number   NUMBER(10) NOT NULL,
   email_id     VARCHAR2(50) NOT NULL,
   first_name   VARCHAR2(20) NOT NULL,
   last_name    VARCHAR2(50),
   cust_address VARCHAR2(100)
);
ALTER TABLE customer ADD CONSTRAINT
customer_pk PRIMARY KEY ( cust_no );
```

```
CREATE TABLE delivery_trip (
   delivery_id    NUMBER(10) NOT NULL,
   departure_time DATE NOT NULL,
   arrival_time   DATE,
   driver_id      NUMBER(10) NOT NULL,
   moterbike_id   NUMBER(10) NOT NULL
);

ALTER TABLE delivery_trip ADD
CONSTRAINT delivery_trip_pk PRIMARY
KEY ( delivery_id );
```

```
CREATE TABLE discount_voucher (
   voucher_id  NUMBER(10) NOT NULL,
   vouchers_code VARCHAR2(10) NOT NULL,
   description  VARCHAR2(100) NOT NULL,
   issue_date  DATE NOT NULL,
   expiry_date  DATE,
   cust_no  NUMBER(10) NOT NULL,
   order_id  NUMBER(10) NOT NULL
);

ALTER TABLE discount_voucher ADD
CONSTRAINT discount_voucher_pk PRIMARY
KEY ( voucher_id );
```

```
CREATE TABLE driver_motorbikes (
   d_m_mapping_id NUMBER(10) NOT
NULL,
   driver_id      NUMBER(10) NOT NULL,
   motorbikes_id  NUMBER(10) NOT NULL
);

ALTER TABLE driver_motorbikes ADD
CONSTRAINT driver_motorbikes_pk
PRIMARY KEY ( d_m_mapping_id );
```

```
CREATE TABLE drivers (
   driver_id  NUMBER(10) NOT NULL,
   driver_name  VARCHAR2(20) NOT NULL,
   driver_address  VARCHAR2(200) NOT NULL,
   driver_dob  DATE,
  driving_license_number NUMBER(16) NOT
NULL
);
ALTER TABLE drivers ADD CONSTRAINT
drivers_pk PRIMARY KEY ( driver_id );
```

```
CREATE TABLE motorbikes_id (
   motorbikes_id   NUMBER(10) NOT NULL,
   registration_no  VARCHAR2(10) NOT
NULL,
   engine_size  NUMBER(2, 2)
);

ALTER TABLE motorbikes_id
ADD CONSTRAINT motorbikes_id_pk
PRIMARY KEY ( motorbikes_id );
```

```
CREATE TABLE orders (
   order_id  NUMBER(10) NOT NULL,
   unreg_cust_name  VARCHAR2(20),
   unreg_cust_phone_no  NUMBER(10),
   order_datetime  DATE NOT NULL,
   order_amt  NUMBER(5, 2),
   order_status  VARCHAR2(10),
   collection_time  DATE,
  order_outlet  NUMBER(10) NOT NULL,
   advance_booking_datetime DATE,
   order_mode  VARCHAR2(10),
   delivery_est_time  DATE,
   delivery_address  VARCHAR2(200),
   order_delivery_time  DATE,
   cust_no  NUMBER(10) NOT NULL,
   outlet_id  NUMBER(10) NOT NULL,
   delivery_id  NUMBER(10)
);

ALTER TABLE orders
ADD CONSTRAINT orders_pk
PRIMARY KEY ( order_id );
```

```
CREATE TABLE outlet (
   oulet_id  NUMBER(10) NOT NULL,
   phone_no  NUMBER(10) NOT NULL,
   postcode  VARCHAR2(6) NOT NULL,
   address  VARCHAR2(200) NOT NULL,
   name  VARCHAR2(50) NOT NULL,
 serving_area_postcode      VARCHAR2(200)
NOT NULL
);

ALTER TABLE outlet
ADD CONSTRAINT outlet_pk
PRIMARY KEY ( oulet_id );
```

```
CREATE TABLE payment (
   payment_id  NUMBER(10) NOT NULL,
   card_number  NUMBER(16) NOT NULL,
 cardholder_name  VARCHAR2(50) NOT NULL,
 cardholder_address      VARCHAR2(200)      NOT
NULL,
   expiry_date  DATE NOT NULL,
  security_number  INTEGER NOT NULL,
   card_issue_date  DATE NOT NULL,
   card_issue_number  NUMBER(10),
  authorization_number      NUMBER(15)      NOT
NULL,
  payment_type  VARCHAR2(10) NOT NULL,
   voucher_id  NUMBER(10) NOT NULL,
 payment_amount  NUMBER(5, 2) NOT NULL,
payment_amount_after_discount  NUMBER(5,  2)
NOT NULL,
 order_id  NUMBER(10) NOT NULL
);

ALTER TABLE payment ADD CONSTRAINT
payment_pk PRIMARY KEY (payment_id);
```

```
CREATE TABLE pizza (
   pizza_id  NUMBER(10) NOT NULL,
   pizza_price  NUMBER(5, 2) NOT NULL,
   qty  INTEGER NOT NULL,
   base_id  NUMBER(10) NOT NULL,
   pizza_size_id NUMBER(10) NOT NULL
);

ALTER TABLE pizza ADD CONSTRAINT
pizza_pk PRIMARY KEY ( pizza_id );
```

```
CREATE TABLE pizza_order (
   p_o_mapping_id  NUMBER(10) NOT NULL,
   order_id  NUMBER(10) NOT NULL,
   pizza_id  NUMBER(10) NOT NULL
);

ALTER TABLE pizza_order ADD CONSTRAINT
pizza_order_pk PRIMARY KEY ( p_o_mapping_id
);
```

```
CREATE TABLE pizza_outlet (
  p_out_mapping_id NUMBER(10) NOT NULL,
  outlet_id NUMBER(10) NOT NULL,
  pizza_id NUMBER(10) NOT NULL
);

ALTER TABLE pizza_outlet ADD CONSTRAINT pizza_outlet_pk PRIMARY KEY ( p_out_mapping_id );
```

```
CREATE TABLE pizza_topping (
   p_t_mapping_id NUMBER(10) NOT NULL,
   topping_id NUMBER(10) NOT NULL,
   pizza_id NUMBER(10) NOT NULL
);

ALTER TABLE pizza_topping ADD CONSTRAINT pizza_topping_pk PRIMARY KEY ( p_t_mapping_id );
```

```
CREATE TABLE "Size" (
   size_id NUMBER(10) NOT NULL,
   "size" INTEGER NOT NULL
);

ALTER TABLE "Size" ADD CONSTRAINT size_pk PRIMARY KEY ( size_id );
```

```
CREATE TABLE topping (
   topping_id NUMBER(10) NOT NULL,
   topping_type VARCHAR2(20) NOT NULL,
   topping_price NUMBER(5, 2) NOT NULL,
   size_id NUMBER(10) NOT NULL
);
ALTER TABLE topping ADD CONSTRAINT topping_pk PRIMARY KEY ( topping_id );
```

```
ALTER TABLE base
ADD CONSTRAINT base_size_fk
FOREIGN KEY ( size_id )
REFERENCES "Size" ( size_id );
```

```
ALTER TABLE delivery_trip
ADD CONSTRAINT delivery_trip_drivers_fk
FOREIGN KEY ( driver_id )
REFERENCES drivers ( driver_id );
```

```
ALTER TABLE delivery_trip
ADD CONSTRAINT
delivery_trip_motorbikes_id_fk
FOREIGN KEY ( moterbike_id )
REFERENCES motorbikes_id (
motorbikes_id );
```

```
ALTER TABLE discount_voucher
ADD CONSTRAINT
discount_voucher_customer_fk
FOREIGN KEY ( cust_no )
REFERENCES customer ( cust_no );
```

```
ALTER TABLE discount_voucher
ADD CONSTRAINT
discount_voucher_orders_fk
FOREIGN KEY ( order_id )
REFERENCES orders ( order_id );
```

```
ALTER TABLE orders
ADD CONSTRAINT orders_customer_fk
FOREIGN KEY ( cust_no )
REFERENCES customer ( cust_no );
```

```
ALTER TABLE orders
ADD CONSTRAINT orders_delivery_trip_fk
FOREIGN KEY ( delivery_id )
REFERENCES delivery_trip ( delivery_id );
```

```
ALTER TABLE orders
ADD CONSTRAINT orders_outlet_fk FOREIGN KEY ( outlet_id )
REFERENCES outlet ( oulet_id );
```

```
ALTER TABLE payment
ADD CONSTRAINT
payment_discount_voucher_fk
FOREIGN KEY ( voucher_id )
REFERENCES discount_voucher (
voucher_id );
```

```
ALTER TABLE payment
ADD CONSTRAINT payment_orders_fk
FOREIGN KEY ( order_id )
REFERENCES orders ( order_id );
```

```
ALTER TABLE pizza
ADD CONSTRAINT pizza_base_fk
FOREIGN KEY ( base_id )
REFERENCES base ( base_id );
ALTER TABLE pizza_topping
ADD CONSTRAINT table_13_pizza_fk
FOREIGN KEY ( pizza_id )
REFERENCES pizza ( pizza_id );
```

```
ALTER TABLE pizza
ADD CONSTRAINT pizza_size_fk
FOREIGN KEY ( pizza_size_id )
REFERENCES "Size" ( size_id );
ALTER TABLE pizza_topping
ADD CONSTRAINT table_13_topping_fk
FOREIGN KEY ( topping_id )
REFERENCES topping ( topping_id );
```

ALTER TABLE *pizza_order* ADD CONSTRAINT *table_16_orders_fk* FOREIGN KEY (*order_id*) REFERENCES orders (*order_id*);	ALTER TABLE *pizza_order* ADD CONSTRAINT *table_16_pizza_fk* FOREIGN KEY (*pizza_id*) REFERENCES pizza (*pizza_id*);
ALTER TABLE *pizza_outlet* ADD CONSTRAINT *table_23_outlet_fk* FOREIGN KEY (*outlet_id*) REFERENCES outlet (*oulet_id*);	ALTER TABLE *pizza_outlet* ADD CONSTRAINT *table_23_pizza_fk* FOREIGN KEY (*pizza_id*) REFERENCES pizza (*pizza_id*);
ALTER TABLE *driver_motorbikes* ADD CONSTRAINT *table_24_drivers_fk* FOREIGN KEY (*driver_id*) REFERENCES drivers (*driver_id*);	ALTER TABLE *driver_motorbikes* ADD CONSTRAINT *table_24_motorbikes_id_fk* FOREIGN KEY (*motorbikes_id*) REFERENCES *motorbikes_id* (*motorbikes_id*);
ALTER TABLE *topping* ADD CONSTRAINT *topping_size_fk* FOREIGN KEY (*size_id*) REFERENCES "*Size*" (*size_id*);	CREATE TABLE available_at (o_t_s_b_id NUMBER(10) NOT NULL, "outlet_id " NUMBER(10) NOT NULL, topping_id NUMBER(10) NOT NULL, "size_id " NUMBER(10) NOT NULL, "base_Id " NUMBER(10) NOT NULL)
ALTER TABLE available_at ADD CONSTRAINT available_at_base_fk FOREIGN KEY ("base_Id ") REFERENCES base (base_id)	ALTER TABLE available_at ADD CONSTRAINT available_at_outlet_fk FOREIGN KEY ("outlet_id ") REFERENCES outlet (oulet_id)
ALTER TABLE available_at ADD CONSTRAINT available_at_size_fk FOREIGN KEY ("size_id ") REFERENCES "Size" (size_id)	ALTER TABLE available_at ADD CONSTRAINT available_at_topping_fk FOREIGN KEY (topping_id) REFERENCES topping (topping_id)