

# DATABASE TECHNOLOGIES

## Section - I

Create a table Stock with stockid number, stockdet xmltype. Insert values into the stock table as shown below.

```
create table stock(stockid number, stockdet xmltype);

insert into stock values(1, xmltype('<?xml version="1.0"?>
<?xml:stylesheet type="text/xml" href="stock.xml"?>
<portfolio xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <stock exchange="nasdaq">
    <name>new</name>
    <symbol>zzzz</symbol>
    <price dt:dt="number">20.313</price>
  </stock>
  <stock exchange="nyse">
    <name>zack corp</name>
    <symbol>ZCKM</symbol>
    <price dt:dt="number">28.875</price>
  </stock>
  <stock exchange="nasdaq">
    <name>zaffymat inc</name>
    <symbol>ZFFX</symbol>
    <price dt:dt="number">92.250</price>
  </stock>
  <stock exchange="nasdaq">
    <name>zynmergy inc</name>
```

Task completed in 1.504 seconds

ble STOCK created.

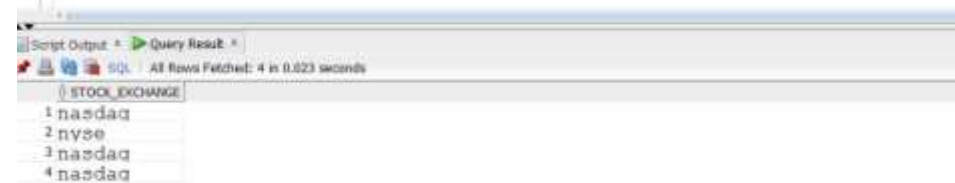
row inserted.

**1. Display stock price whose value is greater than 50.50 using xpath.**

```
WITH
stock_data AS
(select xt.* from stock x, xmltable('portfolio/stock'
    passing x.stockdet columns
    stock_exchange varchar2(10) path '@exchange',
    name varchar2(10) path 'name',
    symbol varchar2(12) path 'symbol',
    price number(10, 4) path 'price')
xt)
select * from stock_data where price>50.50;
```

**Display only the attribute values of the Stock exchange.**

```
WITH
stock_data AS
(select xt.* from stock x, xmltable('portfolio/stock'
passing x.stockdet columns
stock_exchange varchar2(10) path '@exchange',
name varchar2(10) path 'name',
symbol varchar2(12) path 'symbol',
price number(10, 4) path 'price')
xt)
select stock_exchange from stock_data;
```

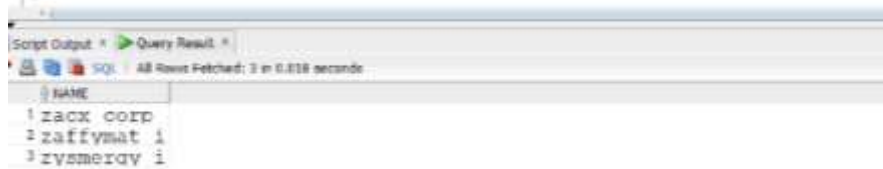


The screenshot shows the SQL Developer interface. The top pane displays the SQL query. The bottom pane shows the query results in a table named 'STOCK\_EXCHANGE' with 4 rows. The first row is '1 nasdaq', the second is '2 nyse', the third is '3 nasdaq', and the fourth is '4 nasdaq'.

| STOCK_EXCHANGE |
|----------------|
| 1 nasdaq       |
| 2 nyse         |
| 3 nasdaq       |
| 4 nasdaq       |

**2. Display the name of the stock whose name starts with “z...”.**

```
WITH
stock_data AS
(select xt.* from stock x, xmltable('portfolio/stock'
passing x.stockdet columns
stock_exchange varchar2(10) path '@exchange',
name varchar2(10) path 'name',
symbol varchar2(12) path 'symbol',
price number(10, 4) path 'price')
xt)
select name from stock_data where name like 'z%';
```



The screenshot shows a database query result window with a tab labeled 'Query Result'. Below the tab, it indicates 'All Rows Fetched: 3 in 0.018 seconds'. The results are displayed in a table with one column named 'NAME'. The table contains three rows of data:

| NAME         |
|--------------|
| 1 ZACK CORP  |
| 2 zaffymat i |
| 3 zysmerdy i |

**3. Display the all the elements of the stock.**

The screenshot displays the Oracle SQL Developer environment. The main window is divided into three panes:

- SQL Worksheet:** Contains the following SQL query:
 

```
SET LONG 5000
SELECT x.stockdet.getClobVal()
FROM stock x;
```
- Script Output / Query Result:** Shows the execution status: "All Rows Fetched: 1 in 0.053 seconds". Below this, the XML output is displayed:
 

```
1 | </stock>
```
- Value:** A pane on the right showing the full XML result of the query:
 

```
<?xml version="1.0"?>
<?xml:stylesheet type="text/xsl" href="stock.xsl"?>
<portfolio xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <stock exchange="nasdaq">
    <name>new</name>
    <symbol>zzzz</symbol>
    <price dt:dt="number">20.313</price>
  </stock>
  <stock exchange="nyse">
    <name>zacx corp</name>
    <symbol>ZCXM</symbol>
    <price dt:dt="number">28.875</price>
  </stock>
  <stock exchange="nasdaq">
    <name>zaffymat inc</name>
    <symbol>ZFFX</symbol>
    <price dt:dt="number">92.250</price>
  </stock>
  <stock exchange="nasdaq">
    <name>zysmergy inc</name>
    <symbol>ZYSZ</symbol>
    <price dt:dt="number">20.313</price>
  </stock>
</portfolio>
```

4. Display only the symbol and price values.

```
WITH
stock_data AS
(select xt.* from stock x, xmltable('portfolio/stock'
passing x.stockdet columns
symbol varchar2(12) path 'symbol',
price number(10, 4) path 'price')
xt)
select symbol, price from stock_data;
```

Script Output \* Query Result \* Query Result 1 \*

SQL All Rows Fetched: 4 in 0.03 seconds

|   | SYMBOL | PRICE  |
|---|--------|--------|
| 1 | ZZZZ   | 20.313 |
| 2 | ZCKM   | 28.875 |
| 3 | ZFFX   | 92.25  |
| 4 | ZYSZ   | 20.313 |

5. Change the price value of stock whose name is “zysmergy inc”.

```
=update stock set stockdet = updatexml(stockdet,'portfolio/stock[name="zysmergy inc"]',
xmltype('<stock exchange="nasdaq">
<name>zysmergy inc</name>
<symbol>ZYSZ</symbol>
<price>100</price>
</stock>
'));
```

Script Output \* Query Result \* Query Result 1 \*

Task completed in 0.206 seconds

>>Query Run In:Query Result

1 row updated.

## Section - II

**Convert RDBMS table into XMLtype. For the following Restaurant data. Create tables for restaurant and menu as per xml document shown below. Copy the restaurant information into the 'rest\_xmltab' using PL/SQL.**

```
create or replace type coordinate as object(  
  latitude varchar2(30),  
  longitude varchar2(30));
```

```
CREATE OR REPLACE TYPE addr AS OBJECT (  
  street varchar2(30),  
  city varchar2(20),  
  country varchar2(30),  
  coordinates coordinate  
);
```

```
create or replace type map_zoom as object  
(minZoomLevel number(3),  
  maxZoomLevel number(3));
```

Script Output \* Query Result \* Query Result 1 \*  
Task completed in 0.083 seconds

Type COORDINATE compiled

Type ADDR compiled

Type MAP\_ZOOM compiled



```
*create table restaurant(  
  name varchar2(30),  
  logo varchar2(30),  
  address addr,  
  map map_zoom,  
  telephone varchar2(13),  
  url varchar2(60),  
  description VARCHAR2(1000)  
);
```

Script Output Query Result Query Result 1  
Task completed in 0.096 seconds

Table RESTAURANT created.

```
insert into restaurant values(  
  'dinner',  
  'content/logo.png',  
  addr('1567 Broadway', 'New York', 'United States', coordinate('40.75930175423635', '-73.98525953292847')),  
  map_zoom(12, 17),  
  '+2129189999',  
  'http://www.diner.com',  
  'It takes guts to open a restaurant in Times Square that lacks a theme or  
  strives to be classier than a carnival. Granted, Diner offers an audacious voyeurs paradise in  
  its soaring staircase that rises alongside a suspended abstract sculpture of a fork. Mayson  
  Bacon Jr. has succeeded finding a perfect balance between class and bacon.'  
);
```

Script Output \* Query Result \* Query Result 1 \*  
Task completed in 0.073 seconds

1 row inserted.

```
create table menu(  
  category_id varchar2(12) primary key,  
  name varchar2(60),  
  icon varchar2(100)  
);  
  
create table dish(  
  dish_name varchar2(30),  
  dish_icon varchar2(30),  
  description varchar2(1000),  
  category_id varchar2(12));
```

Script Output \* Query Result \* Query Result 1 \*

Task completed in 0.135 seconds

Table MENU created.

Table DISH created.

```

insert into menu values('category1', 'Soups', 'content/soups_icon.png');
insert into menu values('category2', 'Vegetarian', 'content/vegetarian_icon.png');
insert into menu values('category3', 'Sea Food', 'content/sea_food_icon.png');
insert into menu values('category4', 'Steaks', 'content/steaks_icon.png');
insert into menu values('category5', 'Desserts', 'content/desserts_icon.png');
insert into menu values('category6', 'Drinks', 'content/drinks_icon.png');

insert into dish values('The King of Prawns', '', 'Deep-fried king prawns and Sweet Chilisauce.', 'category1');
insert into dish values('Super Hot Jalapeno Pops', '', 'Deep-fried jalapenos filled with cheese. Served with garlic mayonnaise and out SuperHot');
insert into dish values('Muchos Nachos', '', 'Tortillaa chips, melted cheddar cheese, salsa sauce, jalapeno slices, sour cream and gu');
insert into dish values('Just Fries', '', 'Salted fries with bbq sauce', 'category1');
insert into dish values('Blue Cheese and Fries', '', 'Seasoned Criss Cuts with blue cheesesauce.', 'category1');
insert into dish values('Sweet Fries', '', 'Sweet potato fries with our special mustardsauce.', 'category1');
insert into dish values('THE Ceasar Salad', '', 'A mixed green salad with Caesar dressing, Parmesan cheese, croutons and arugula with a choi');
insert into dish values('All Veggies', '', 'A mixed green salad with cherry tomato, cucumber, fresh pineapple, orange, marinated red or');
insert into dish values('Fruity Fish Salad', '', 'A mixed green salad with salmon fillet, cherry tomato, cucumber, fresh pineapple, orange, m');
insert into dish values('Master Muffin', '', 'Chocolate muffin with white-chock chips, served with vanilla whipped cream.', 'category5');
insert into dish values('The Big Apple', '', 'Apple pie with pecan nuts and raisins, served with vanilla ice cream and chocolate sauce.', 'category5');
insert into dish values('Shake it, shake it', '', 'A giant milk shake with a choice of bananachocolate, chocolate, strawberry-vanilla or mint-');
insert into dish values('Super Cheesecake', '', 'A huge slice of cheesecake served with home of whipped cream and Cold brew and chocolate sauce.').

```

Script Output \* Query Result \* Query Result 1 \*

Task completed in 0.265 seconds

1 row inserted.

## 1. Retrieve the restaurant information into xml form using PL/SQL code

The screenshot displays a SQL IDE with a query window on the left and a 'View Value' window on the right.

**SQL Query:**

```

DECLARE
  l_xmltype XMLTYPE;
BEGIN
  SELECT XMLELEMENT("restaurant",
    XMLAGG(
      XMLELEMENT("info",
        XMLFOREST(
          e.name AS "name",
          e.logo AS "logo",
          e.address AS "address",
          e.map AS "map",
          e.telephone AS "telephone",
          e.url AS "url",
          e.description AS "description"
        )
      )
    )
  INTO l_xmltype FROM restaurant e;
  INSERT INTO xml_restaurant VALUES (1, l_xmltype);
end;
select * from xml_restaurant;

```

**View Value:**

```

<restaurant>
  <info>
    <name>diner</name>
    <logo>content/logo.png</logo>
    <address>
      <STREET>1567 Broadway</STREET>
      <CITY>New York</CITY>
      <COUNTRY>United States</COUNTRY>
      <COORDINATES>
        <LATITUDE>40.75930175423635</LATITUDE>
        <LONGITUDE>-73.98525953252847</LONGITUDE>
      </COORDINATES>
    </address>
    <map>
      <MINZOOMLEVEL>12</MINZOOMLEVEL>
      <MAXZOOMLEVEL>17</MAXZOOMLEVEL>
    </map>
    <telephone>+2129189999</telephone>
    <url>http://www.diner.com</url>
    <description>It takes guts to open a restaurant in Times Square that lacks a theme or strives to be classier than a carnival. Granted, Diner offers an audacious voyeurs paradise in its soaring staircase that rises alongside a suspended abstract sculpture of a fork. Mayson Bacon Jr. has succeeded finding a perfect balance between class and bacon.</description>
  </info>
</restaurant>

```

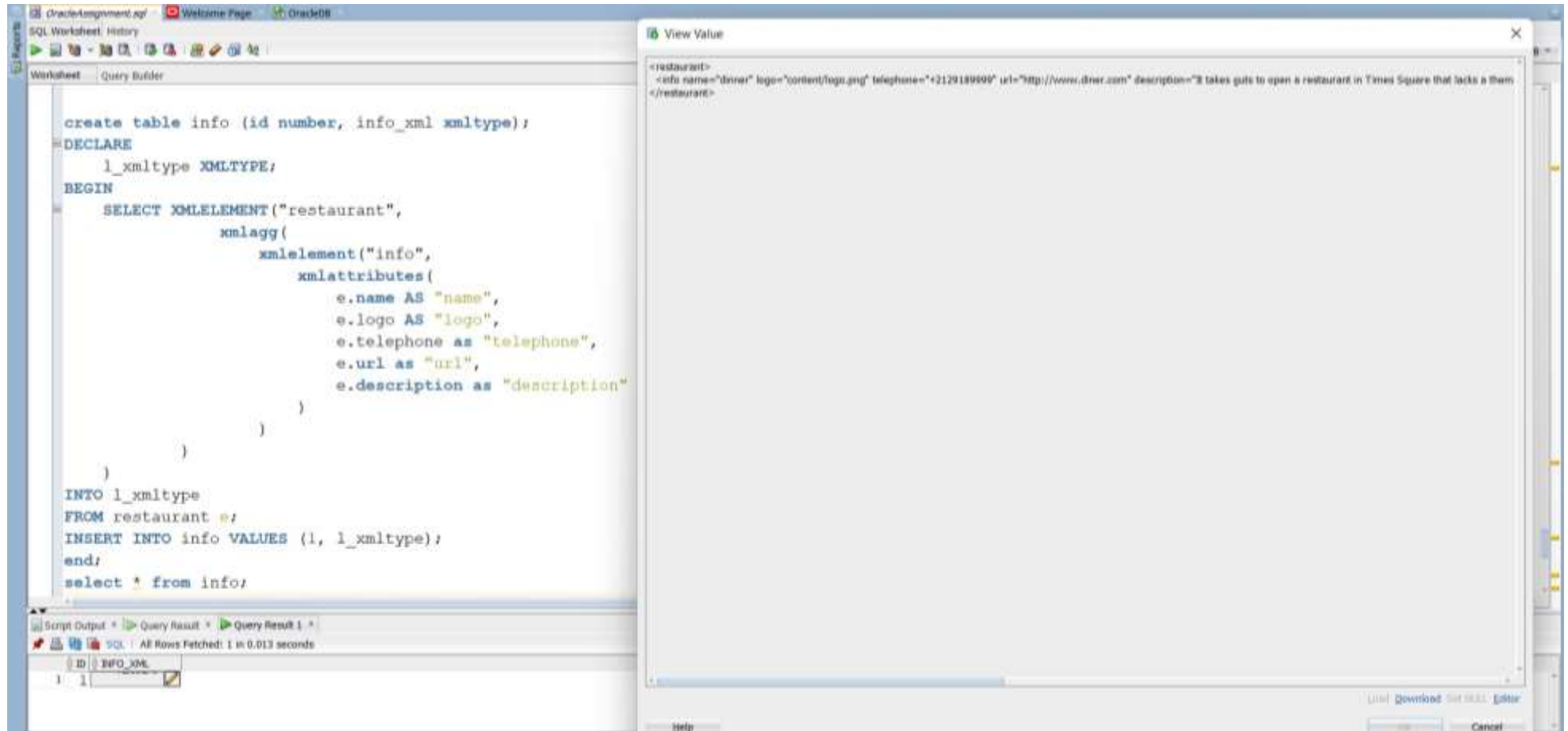
**Script Output:**

Query Result 1: 1 in 0.011 seconds

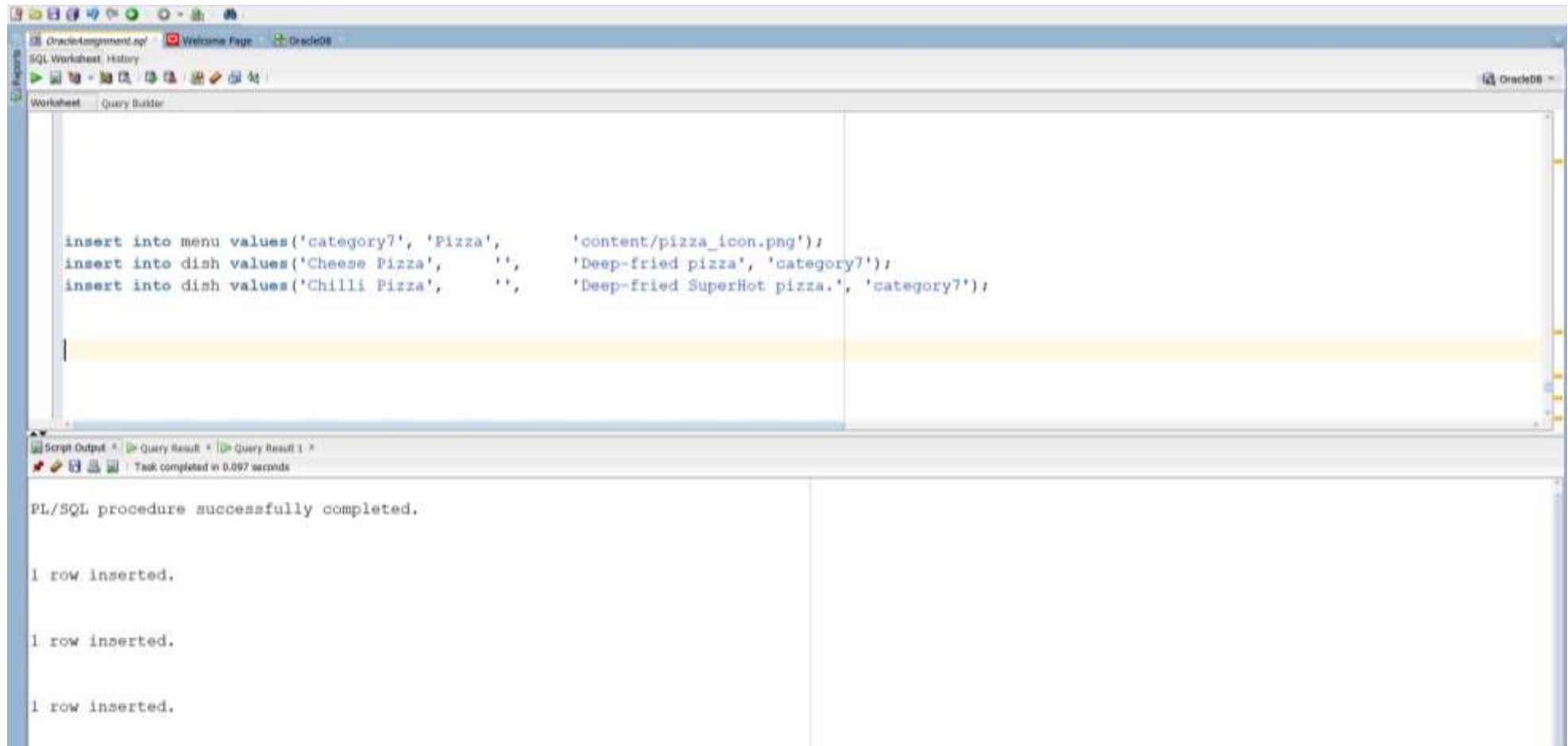
| ID | XML_DATA |
|----|----------|
| 1  | 1        |

## 2. Retrieve the menu details into xml column





4. Insert a new category element with a set of dish names.



5. Update “category5” of dish name “Super Cheesecake” into “Super Cheese Sauce”.



