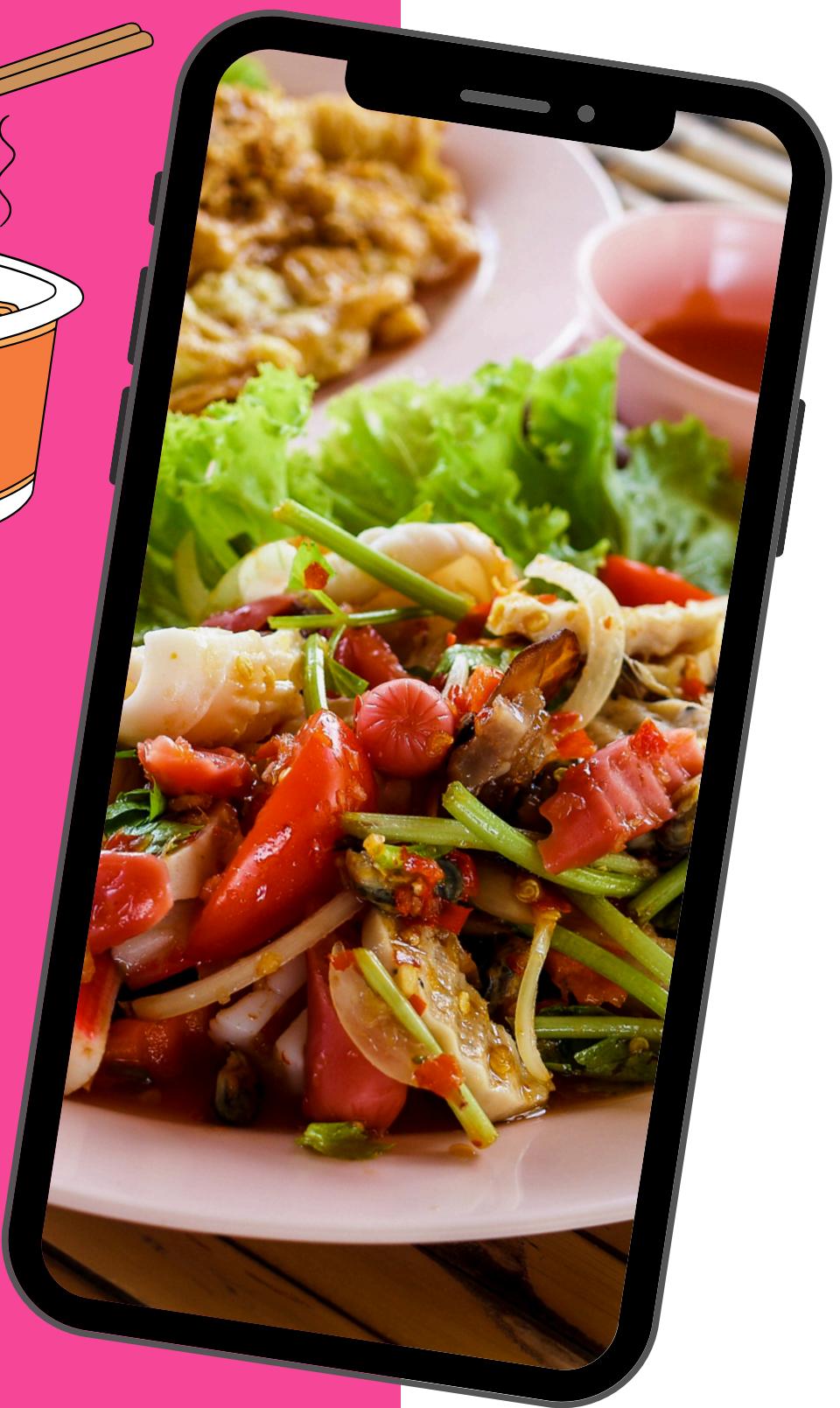
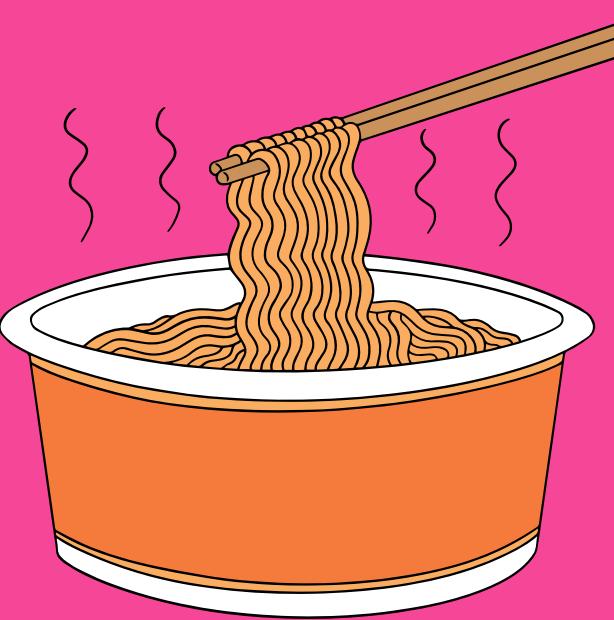
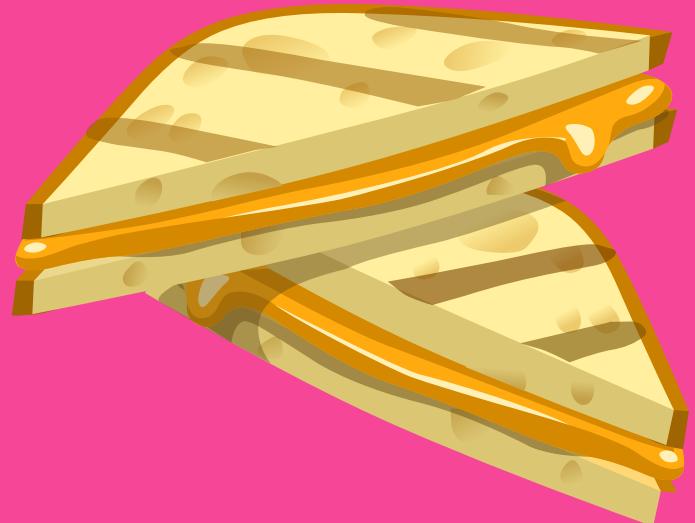


Food Tastes Best When It's On Time

Hunger Express



Online Food Delivery app



Tables Used

Driver Table

Customers Table

Restaurants Table

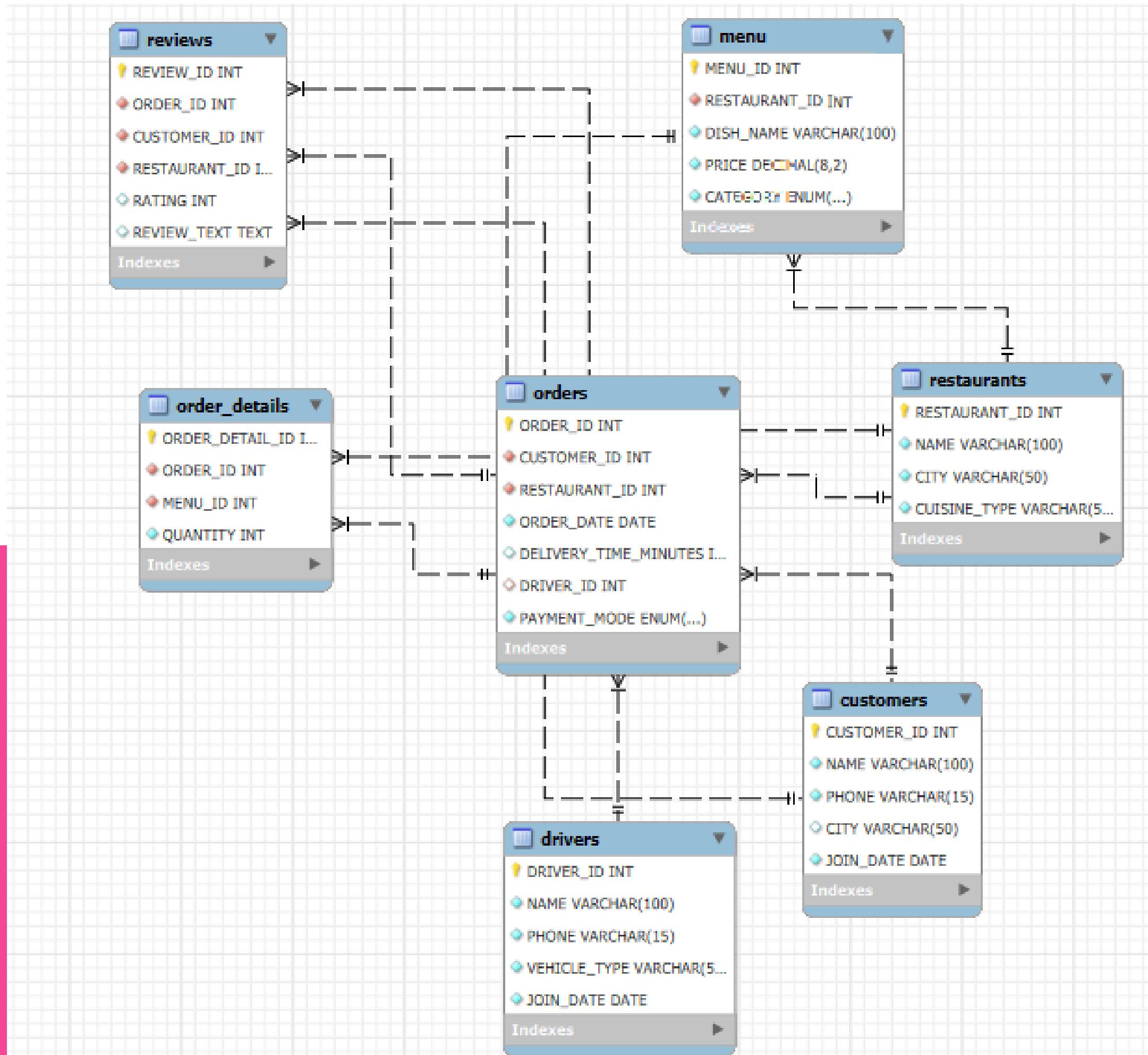
Menu Table

Orders Table

Order_Details Rable

Reviews Table

Entity Relationship Diagram



Scenario

You are building a Database for an online food delivery platform named Hunger Express  , similar to Swiggy or Zomato.



The Platform manages

-  Restaurants and their menus 
-  Customers and their orders 
-  Delivery drivers 
-  Ratings and  Reviews
-  Payments and  Delivery times

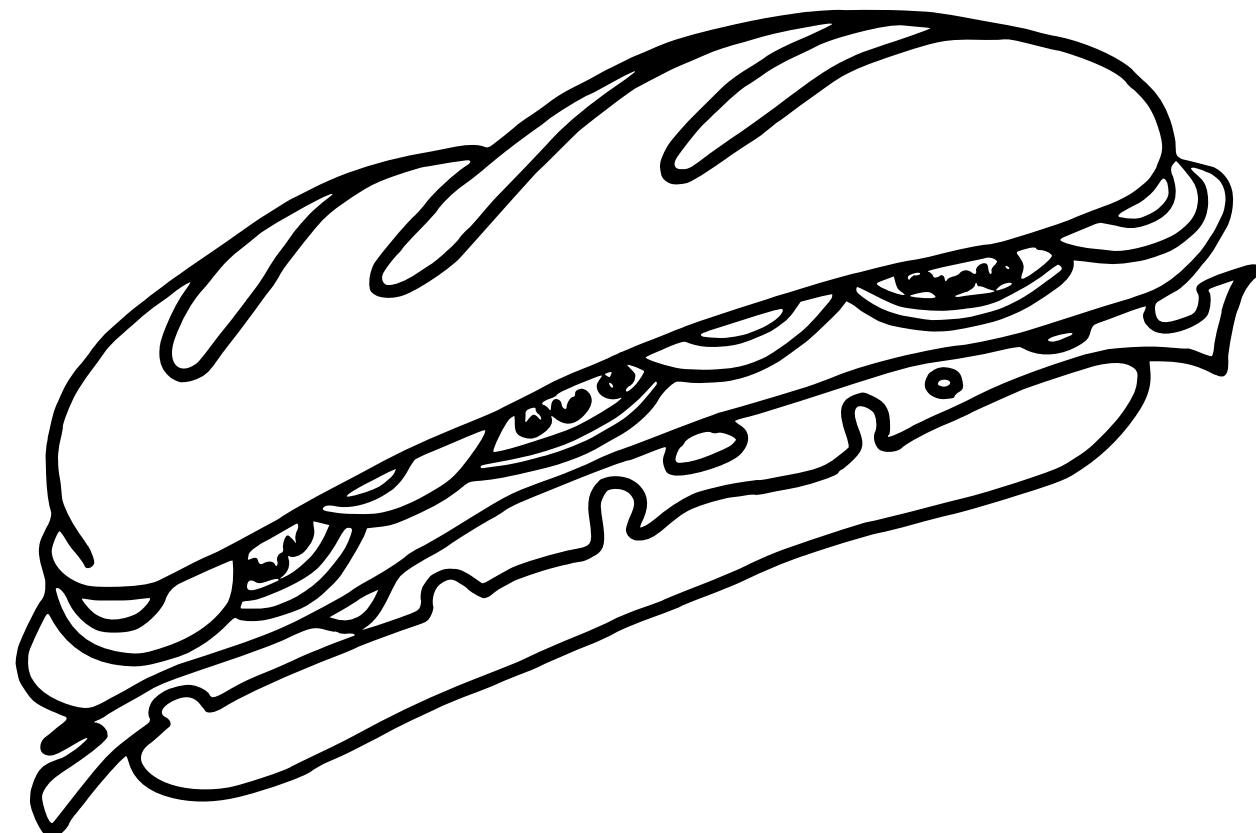
This system will help management

-  Track performance of restaurants
-  Identify customer purchase patterns
-  Monitor delivery performance
-  Improve marketing campaigns



Driver Table

```
CREATE TABLE DRIVERS (
    DRIVER_ID INT PRIMARY KEY
    AUTO_INCREMENT,
    NAME VARCHAR(100) NOT NULL,
    PHONE VARCHAR(15) UNIQUE NOT NULL,
    VEHICLE_TYPE VARCHAR(50) NOT NULL,
    JOIN_DATE DATE NOT NULL
);
```



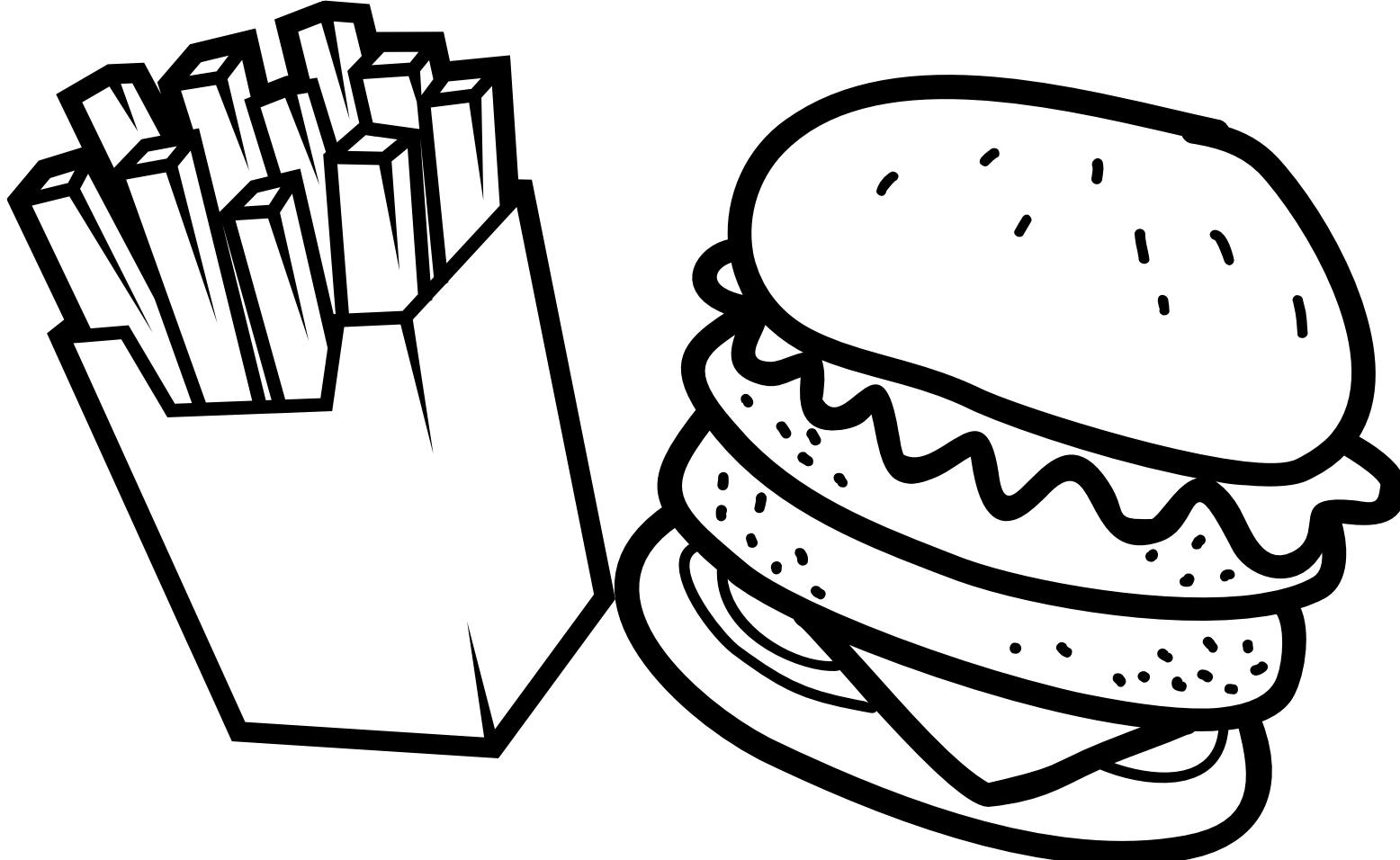
Customers Table

```
CREATE TABLE CUSTOMERS (
    CUSTOMER_ID INT PRIMARY KEY
    AUTO_INCREMENT,
    NAME VARCHAR(100) NOT NULL,
    PHONE VARCHAR(15) UNIQUE NOT NULL,
    CITY VARCHAR(50),
    JOIN_DATE DATE NOT NULL
);
```



Restaurants Table

```
CREATE TABLE RESTAURANTS (
    RESTAURANT_ID INT PRIMARY KEY
    AUTO_INCREMENT,
    NAME VARCHAR(100) NOT NULL,
    CITY VARCHAR(50) NOT NULL,
    CUISINE_TYPE VARCHAR(50) NOT NULL
);
```



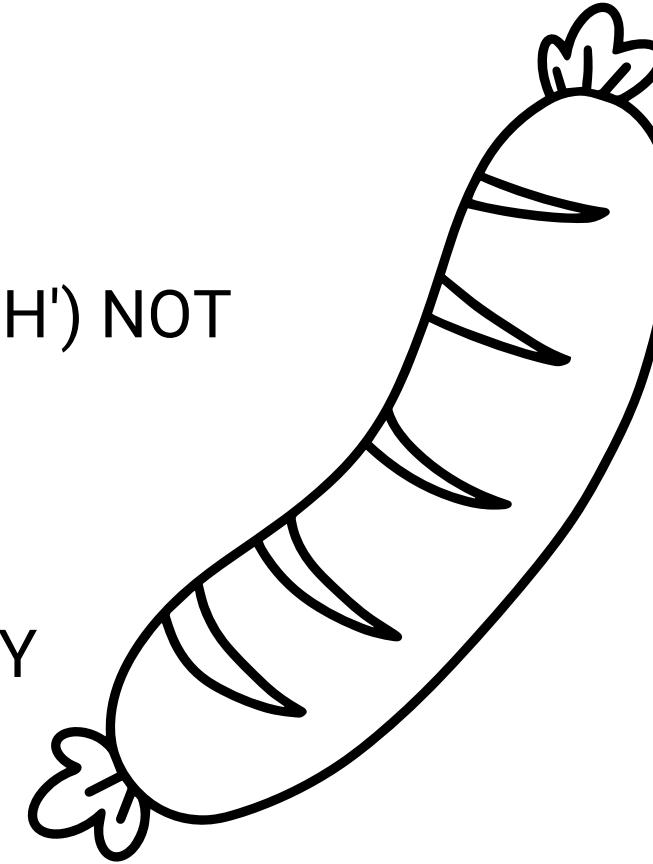
Menu Table

```
CREATE TABLE MENU (
    MENU_ID INT PRIMARY KEY
    AUTO_INCREMENT,
    RESTAURANT_ID INT NOT NULL,
    DISH_NAME VARCHAR(100) NOT NULL,
    PRICE DECIMAL(8,2) NOT NULL CHECK
    (PRICE > 0),
    CATEGORY ENUM('MAIN COURSE',
    'DESSERT', 'BEVERAGE', 'STARTER') NOT
    NULL,
    FOREIGN KEY (RESTAURANT_ID)
    REFERENCES
    RESTAURANTS(RESTAURANT_ID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```



Orders Table

```
CREATE TABLE ORDERS (
    ORDER_ID INT PRIMARY KEY AUTO_INCREMENT,
    CUSTOMER_ID INT NOT NULL,
    RESTAURANT_ID INT NOT NULL,
    ORDER_DATE DATE NOT NULL,
    DELIVERY_TIME_MINUTES INT,
    DRIVER_ID INT,
    PAYMENT_MODE ENUM('UPI', 'CARD', 'CASH') NOT
NULL,
-- FOREIGN KEYS
    CONSTRAINT FK_CUSTOMER FOREIGN KEY
(CUSTOMER_ID) REFERENCES
CUSTOMERS(CUSTOMER_ID),
    CONSTRAINT FK_RESTAURANT FOREIGN KEY
(RESTAURANT_ID) REFERENCES
RESTAURANTS(RESTAURANT_ID),
    CONSTRAINT FK_DRIVER FOREIGN KEY (DRIVER_ID)
REFERENCES DRIVERS(DRIVER_ID)
);
```



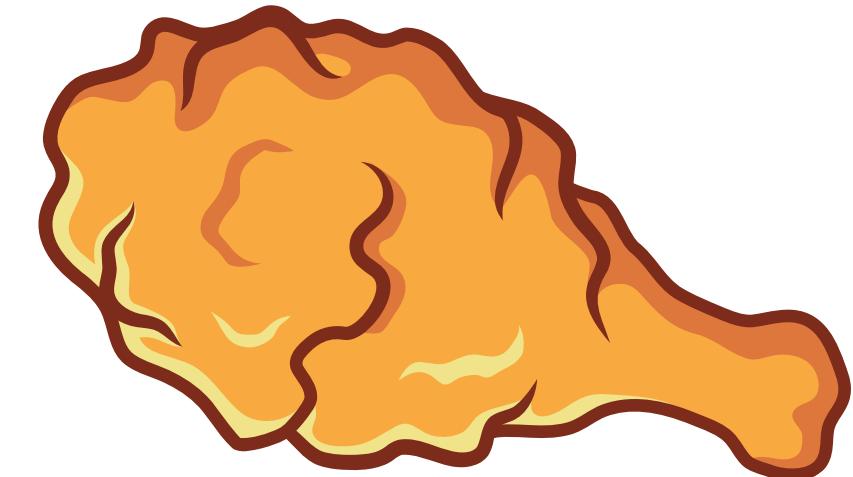
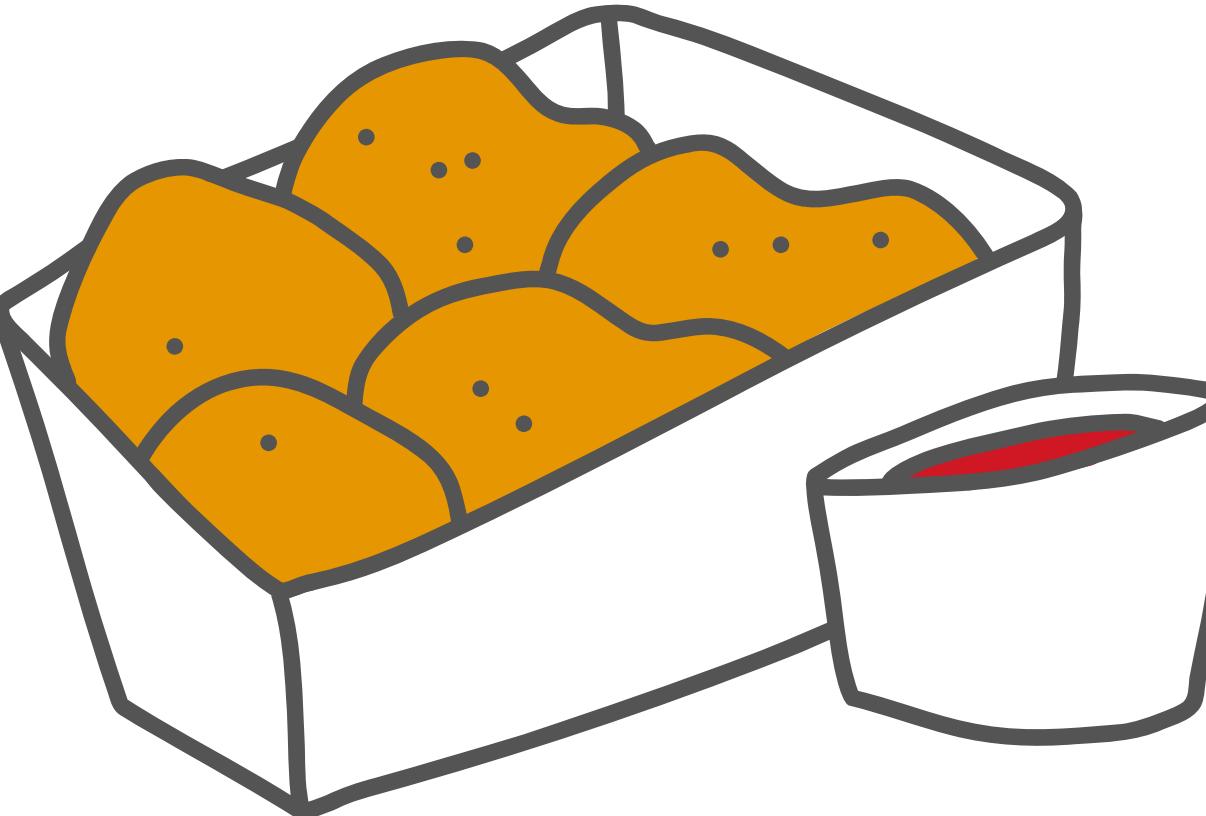
Order_Details Table

```
CREATE TABLE ORDER_DETAILS (
    ORDER_DETAIL_ID INT PRIMARY KEY
AUTO_INCREMENT,
    ORDER_ID INT NOT NULL,
    MENU_ID INT NOT NULL,
    QUANTITY INT NOT NULL CHECK
(QUANTITY > 0),
    CONSTRAINT FK_ORDER FOREIGN KEY
(ORDER_ID) REFERENCES
ORDERS(ORDER_ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CONSTRAINT FK_MENU FOREIGN KEY
(MENU_ID) REFERENCES MENU(MENU_ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

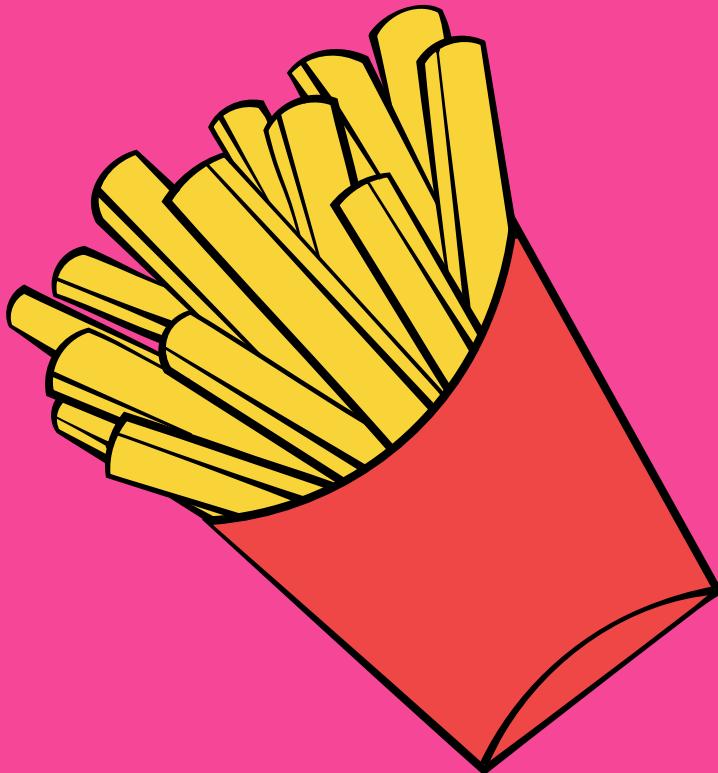


Reviews Table

```
CREATE TABLE REVIEWS (
    REVIEW_ID INT PRIMARY KEY AUTO_INCREMENT,
    ORDER_ID INT NOT NULL,
    CUSTOMER_ID INT NOT NULL,
    RESTAURANT_ID INT NOT NULL,
    RATING INT CHECK (RATING BETWEEN 1 AND 5),
    REVIEW_TEXT TEXT,
    CONSTRAINT FK REVIEW ORDER FOREIGN KEY (ORDER_ID)
        REFERENCES ORDERS(ORDER_ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CONSTRAINT FK REVIEW CUSTOMER FOREIGN KEY (CUSTOMER_ID)
        REFERENCES CUSTOMERS(CUSTOMER_ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CONSTRAINT FK REVIEW RESTAURANT FOREIGN KEY (RESTAURANT_ID)
        REFERENCES RESTAURANTS(RESTAURANT_ID)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

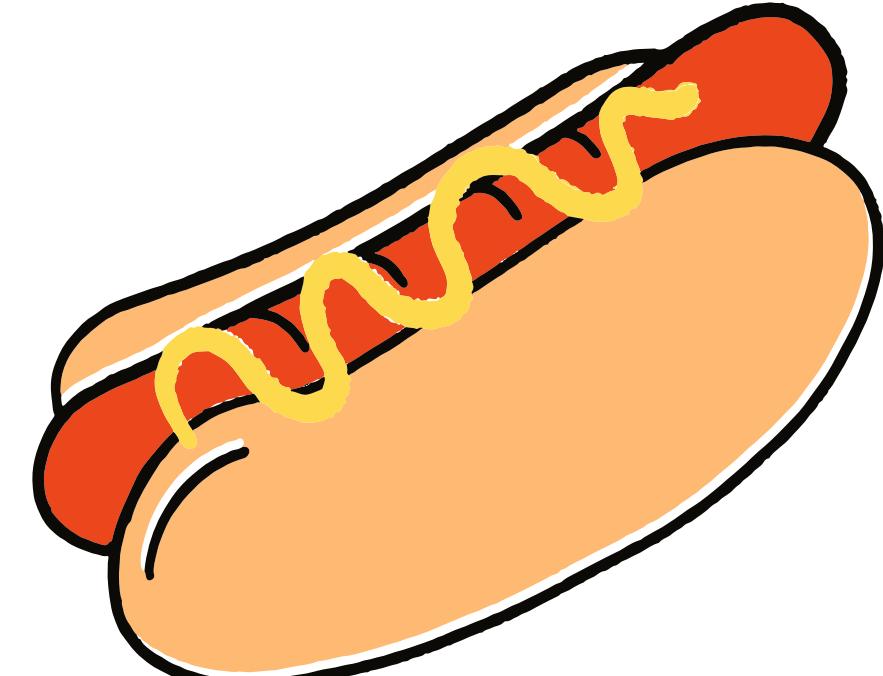
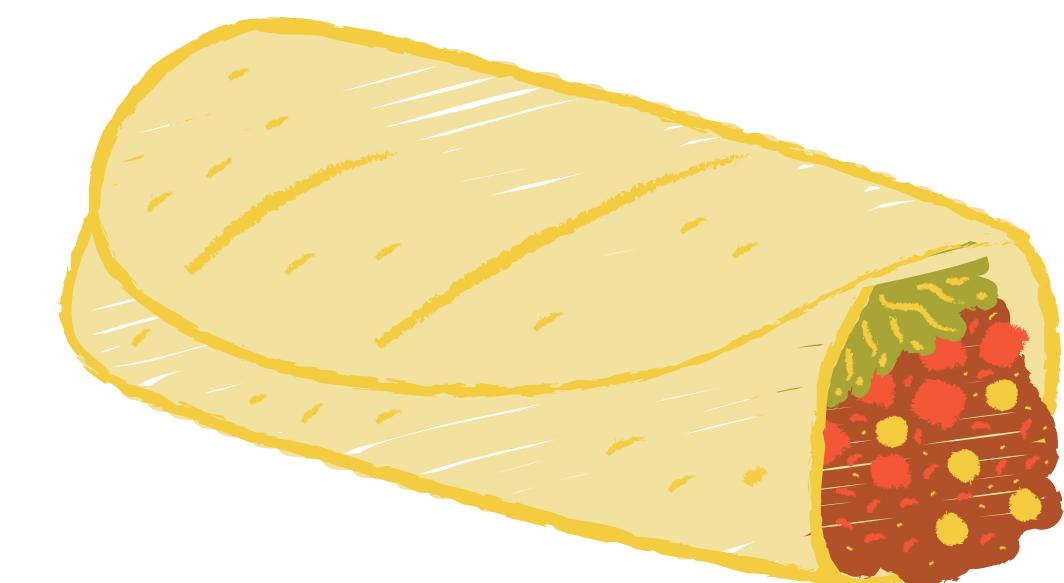


List the top 3 most-ordered dishes in each city.

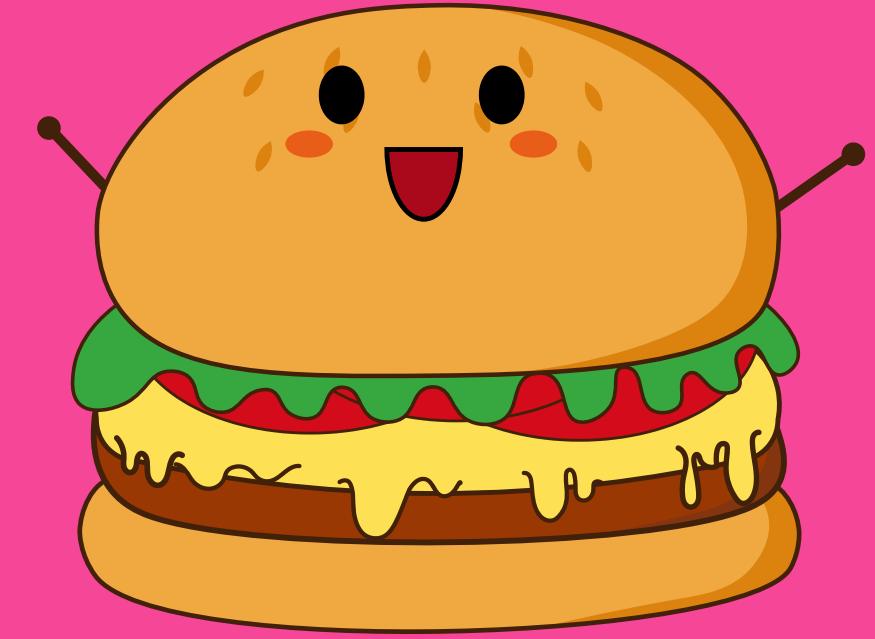


CITY	DISH_NAME
BANGALORE	KUNG PAO CHICKEN
BANGALORE	VEGETABLE SPRING ROLLS
BANGALORE	SWEET CORN SOUP
BANGALORE	CHILLI CHICKEN
BANGALORE	JASMINE TEA
BANGALORE	CHILL SHRIMP
BHUBANESWAR	BUTTER CHICKEN
BHUBANESWAR	PANEER TIKKA

```
WITH DISH_RANK AS (
    SELECT R.CITY CITY,M.DISH_NAME DISH_NAME,
    COUNT(OD.ORDER_ID) AS TIMES_ORDERED,
    RANK() OVER (PARTITION BY R.CITY ORDER BY
    COUNT(OD.ORDER_ID) DESC) AS RANKING
    FROM ORDER_DETAILS OD
    JOIN MENU M ON OD.MENU_ID = M.MENU_ID
    JOIN RESTAURANTS R ON M.RESTAURANT_ID =
    R.RESTAURANT_ID
    JOIN ORDERS O ON OD.ORDER_ID = O.ORDER_ID
    GROUP BY R.CITY, M.DISH_NAME
)
SELECT CITY, DISH_NAME
FROM DISH_RANK
WHERE RANKING <= 3
ORDER BY CITY, RANKING;
```



Find customers who spent more than ₹5000 in the last month.



CUSTOMER_NAME	TOTAL_SPENT
NIKITA PANDEY	10300.00
PRANSHU SHUKLA	9000.00
JIT SARKAR	6000.00

Because food tastes best when it's on time!

Identify customers who placed orders from more than 5 different restaurants.

CUSTOMER_NAME
ANSHIKA RANAUT
CHIRAG CHAUHAN

```
SELECT C.NAME AS CUSTOMER_NAME  
FROM CUSTOMERS C  
JOIN ORDERS O ON C.CUSTOMER_ID =  
O.CUSTOMER_ID  
GROUP BY C.CUSTOMER_ID, C.NAME  
HAVING COUNT(DISTINCT  
O.RESTAURANT_ID) > 5  
ORDER BY COUNT(DISTINCT  
O.RESTAURANT_ID) DESC;
```

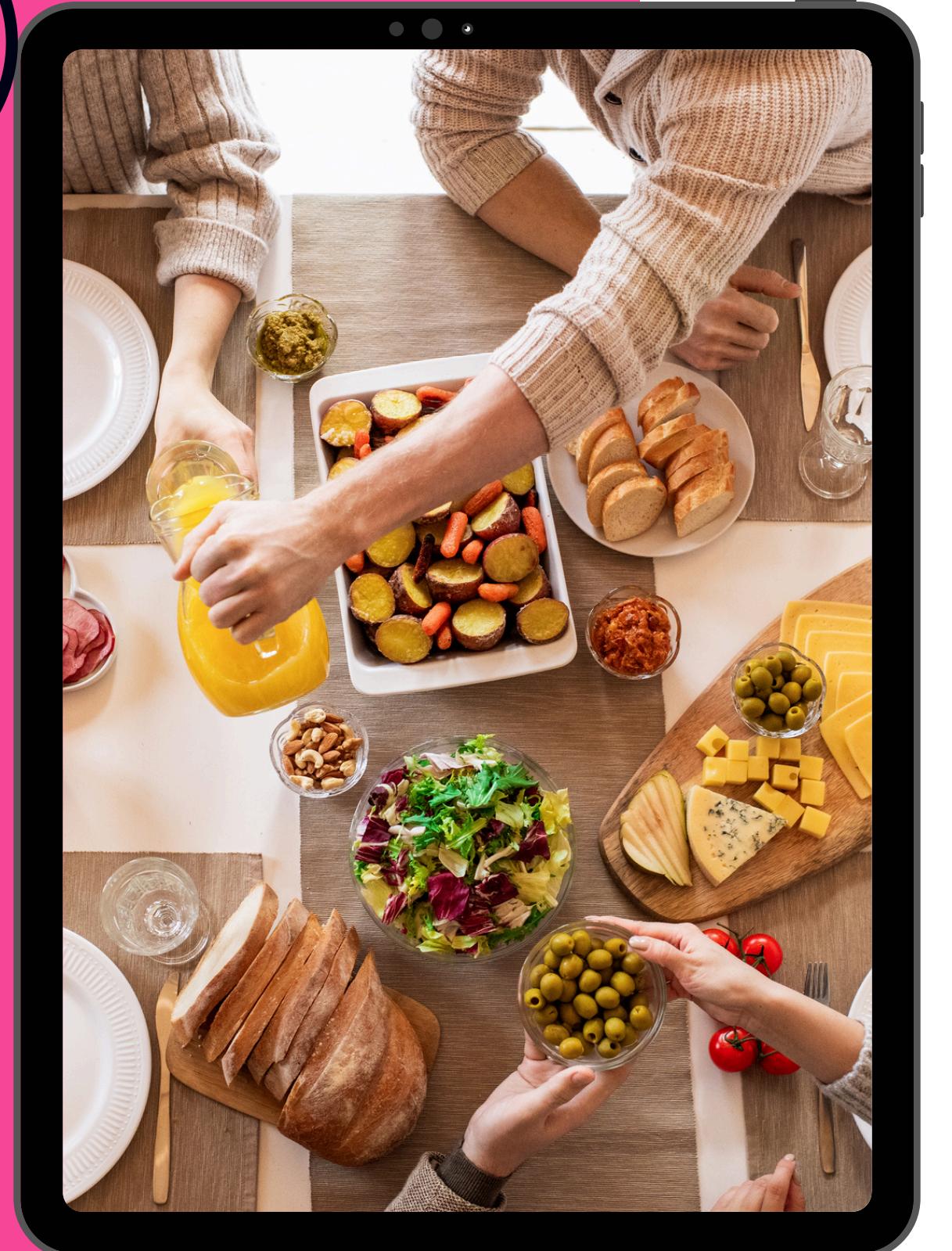


Show repeat customers (placed more than 3 orders in the last 2 months).



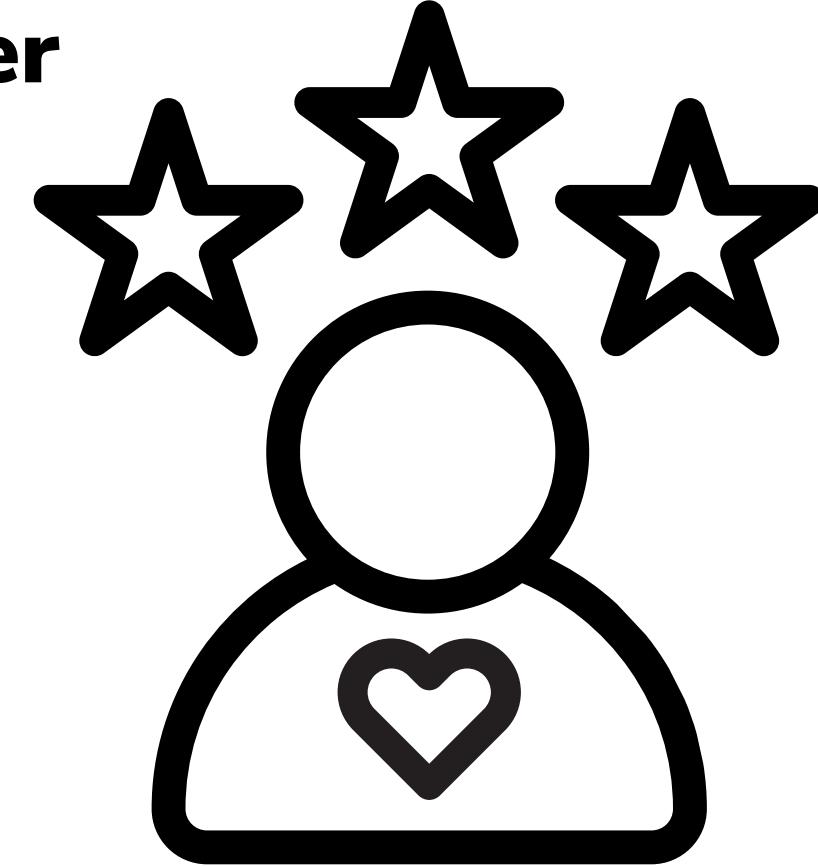
```
SELECT C.NAME AS CUSTOMER_NAME  
FROM CUSTOMERS C  
JOIN ORDERS O  
ON C.CUSTOMER_ID = O.CUSTOMER_ID  
WHERE O.ORDER_DATE BETWEEN  
'2025-07-01' AND '2025-08-31'  
GROUP BY C.NAME  
HAVING COUNT(O.ORDER_ID) > 3;
```

	CUSTOMER_NAME
→	NIKITA PANDEY



List customers who have never given a review.

```
SELECT DISTINCT(C.NAME)  
CUSTOMER_NAME  
FROM CUSTOMERS C  
JOIN REVIEWS R  
ON C.CUSTOMER_ID =  
R.CUSTOMER_ID  
WHERE R.REVIEW_TEXT IS NULL  
OR R.REVIEW_TEXT = ";
```

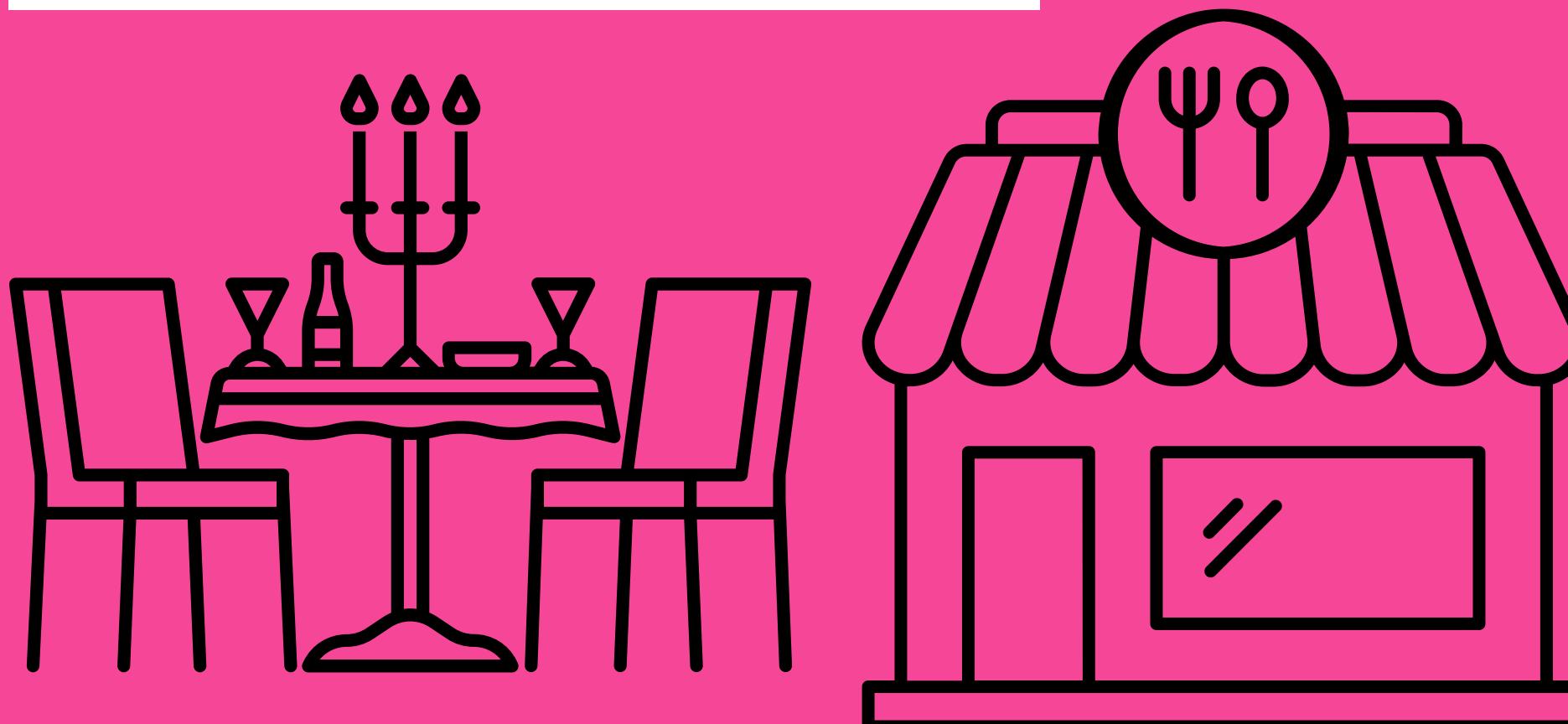


CUSTOMER_NAME
TULSI BOPCHE
AYSUH RAWAT
YASH SHARMA
RAHUL SHARMA
ANSHIKA RANAUT
HIMA MAZUMDAR
VENUGOPAL IYER
BALVINDAR SINGH

Find the restaurant with the highest number of orders in the last quarter.

RESTAURANT_NAME
DRAGON WOK
SWEET CRAVINGS CAFE
BELLA ITALIA
GRILL & CHILL
TANDOORI NIGHTS
LA PASTA FRESCA

```
WITH MAX_ORDERS AS(
SELECT RT.NAME RESTAURANT_NAME,
COUNT(O.ORDER_ID) TOTAL_ORDERS,
RANK() OVER(ORDER BY COUNT(O.ORDER_ID)
DESC)RNK
FROM RESTAURANTS RT
JOIN ORDERS O
ON RT.RESTAURANT_ID=O.RESTAURANT_ID
WHERE QUARTER(O.ORDER_DATE) =
QUARTER(CURDATE() - INTERVAL 1 QUARTER)
AND YEAR(O.ORDER_DATE) = YEAR(CURDATE() -
INTERVAL 1 QUARTER)
GROUP BY RT.NAME
)
SELECT RESTAURANT_NAME FROM MAX_ORDERS
WHERE RNK=1;
```





List restaurants whose average rating is below 3.5.

```
SELECT RT.NAME AS RESTAURANT_NAME  
FROM RESTAURANTS RT  
JOIN REVIEWS R  
ON  
RT.RESTAURANT_ID=R.RESTAURANT_ID  
GROUP BY RT.NAME  
HAVING AVG(R.RATING)  
<3.5;
```



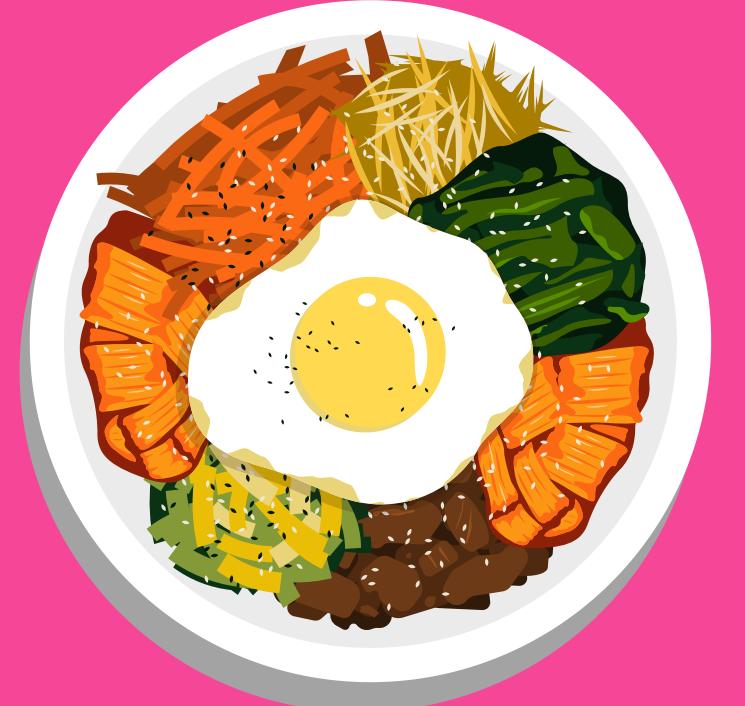
	RESTAURANT_NAME
▶	SWEET CRAVINGS CAFE
	LA PASTA FRESCA

Identify the most popular cuisine type across all cities.

```
SELECT RT.CUISINE_TYPE CUISINE_TYPE  
FROM RESTAURANTS RT  
JOIN ORDERS O  
ON RT.RESTAURANT_ID=O.RESTAURANT_ID  
GROUP BY RT.CUISINE_TYPE  
ORDER BY COUNT(*) DESC  
LIMIT 1;
```



	CUISINE_TYPE
▶	AMERICAN

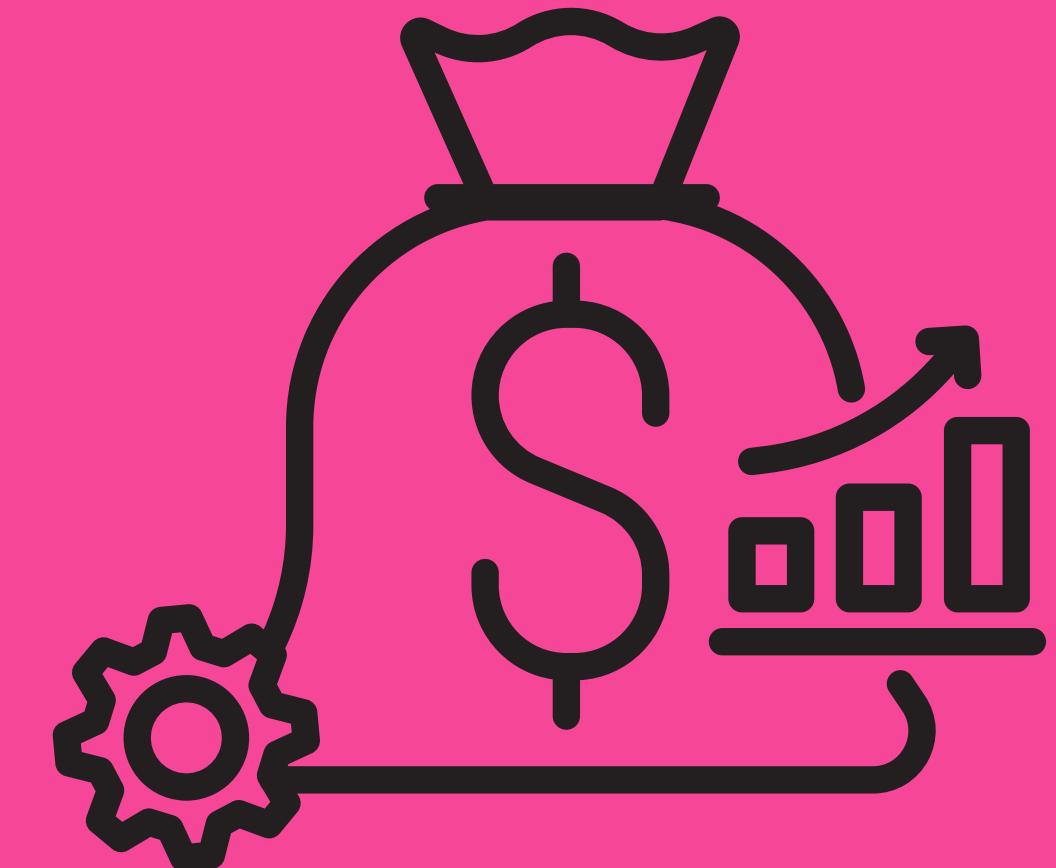


Calculate total revenue for each restaurant in the last month.



```
SELECT RT.NAME, CONCAT('₹ ',  
SUM(M.PRICE * OD.QUANTITY)) AS  
LAST_MONTH_REVENUE  
FROM RESTAURANTS RT  
JOIN MENU M ON RT.RESTAURANT_ID =  
M.RESTAURANT_ID  
JOIN ORDER_DETAILS OD ON M.MENU_ID =  
OD.MENU_ID  
JOIN ORDERS O ON O.ORDER_ID =  
OD.ORDER_ID  
WHERE O.ORDER_DATE BETWEEN '2025-08-  
01' AND '2025-08-31'  
GROUP BY RT.NAME;
```

	NAME	LAST_MONTH_REVENUE
▶	BELLA ITALIA	₹ 9500.00
	DRAGON WOK	₹ 8340.00
	BURGER HUB	₹ 2400.00
	SPICE OF INDIA	₹ 6070.00
	SWEET CRAVINGS CAFE	₹ 3600.00



Find restaurants that have at least one dish priced above ₹1000.



	NAME
▶	SPICE OF INDIA
	BELLA ITALIA
	DRAGON WOK
	BURGER HUB
	SWEET CRAVINGS CAFE
	TANDOORI NIGHTS
	LA PASTA FRESCA
	GOLDEN CHOPSTICKS

```
SELECT DISTINCT(RT.NAME) FROM RESTAURANTS RT  
JOIN MENU M  
ON RT.RESTAURANT_ID=M.RESTAURANT_ID  
WHERE M.PRICE>1000
```





Calculate average delivery time by driver.

```
SELECT D.NAME  
DRIVER_NAME,ROUND(AVG(O.  
DELIVERY_TIME_MINUTES),2)  
AS AVG_DELIVERY_TIME  
FROM DRIVERS D  
JOIN ORDERS O  
ON D.DRIVER_ID=O.DRIVER_ID  
GROUP BY D.NAME  
ORDER BY  
AVG_DELIVERY_TIME;
```

	DRIVER_NAME	Avg_Delivery_Time
▶	VIKAS YADAV	32.50
	MANOJ SINGH	34.29
	SURESH REDDY	34.86
	KARAN MEHTA	35.83
	RAHUL VERMA	36.57
	ARJUN DAS	37.71
	ROHIT NAIR	38.33
	RAVI KUMAR	39.29

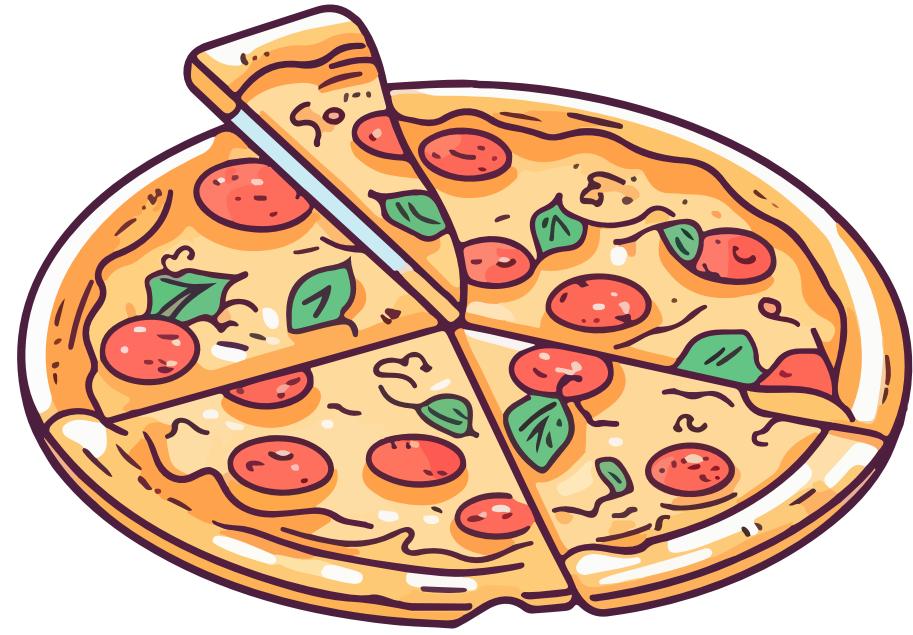
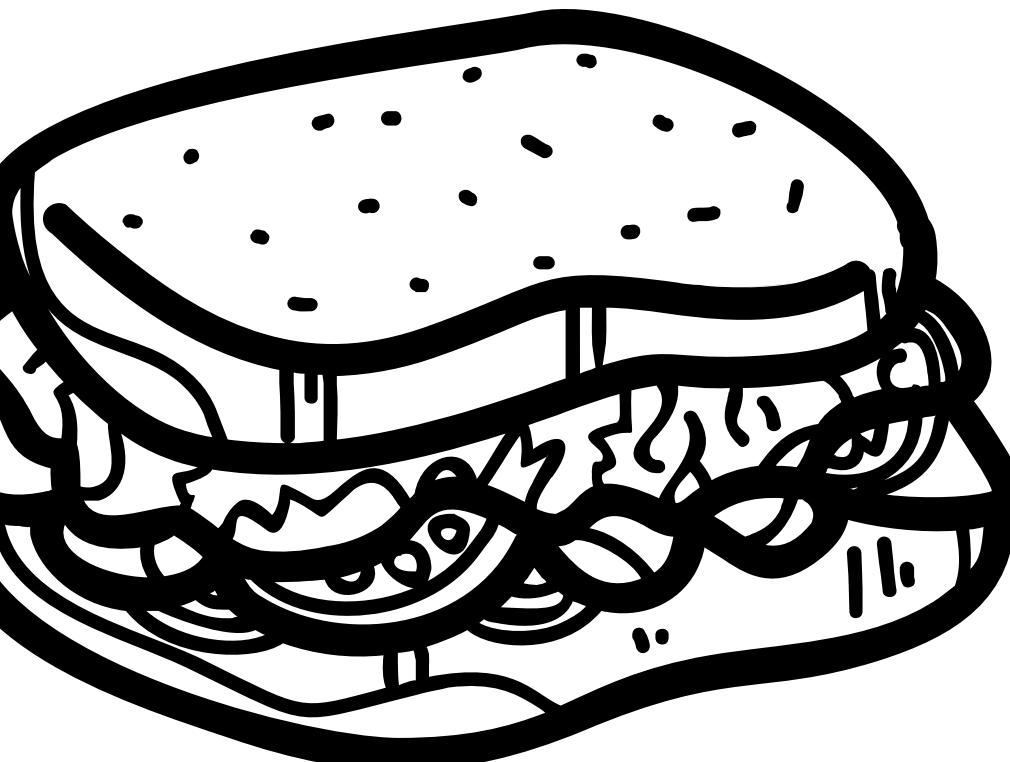


```
WITH AVERAGE_DELIVERY_TIME AS(  
SELECT D.NAME  
DRIVER_NAME,ROUND(AVG(O.DELIVERY_TIME_MINU  
TES),2) AS AVG_DELIVERY_TIME,  
RANK()OVER(ORDER BY  
ROUND(AVG(O.DELIVERY_TIME_MINUTES),2)) AS  
RNK  
FROM DRIVERS D  
JOIN ORDERS O  
ON D.DRIVER_ID=O.DRIVER_ID  
GROUP BY D.NAME  
)  
SELECT DRIVER_NAME FROM  
AVERAGE_DELIVERY_TIME  
WHERE RNK=1;
```



Identify the fastest driver in terms of average delivery time.

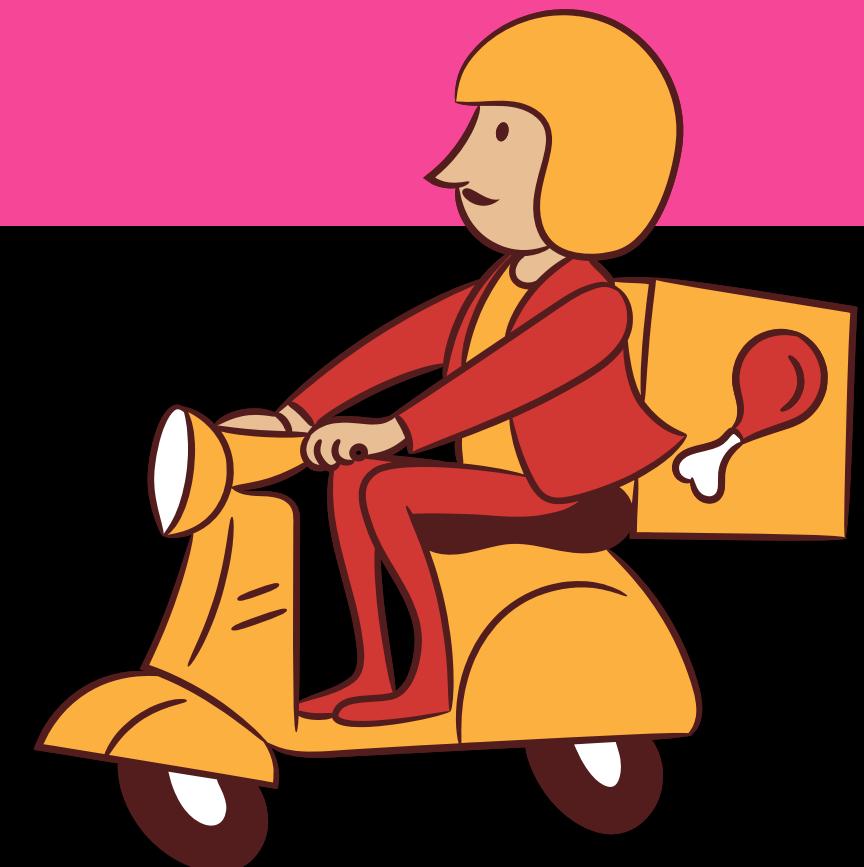
	DRIVER_NAME
▶	VIKAS YADAV



NAME	LATE_DELIVERY_PERCENTAGE
RAVI KUMAR	28.57%
AMIT SHARMA	22.22%
SURESH REDDY	28.57%
RAHUL VERMA	28.57%
MANOJ SINGH	14.29%
ARJUN DAS	42.86%
KARAN MEHTA	33.33%
VIKAS YADAV	33.33%

**Find the percentage of late deliveries
(e.g., >40 minutes) per driver.**

```
SELECT D.NAME,
CONCAT(ROUND(COUNT(CASE WHEN
O.DELIVERY_TIME_MINUTES>40 THEN 1
END)*100.0/COUNT(O.ORDER_ID),2),'%') AS
LATE_DELIVERY_PERCENTAGE
FROM DRIVERS D
JOIN ORDERS O
ON D.DRIVER_ID=O.DRIVER_ID
GROUP BY D.NAME;
```



List dishes that were never ordered in the last 6 months.

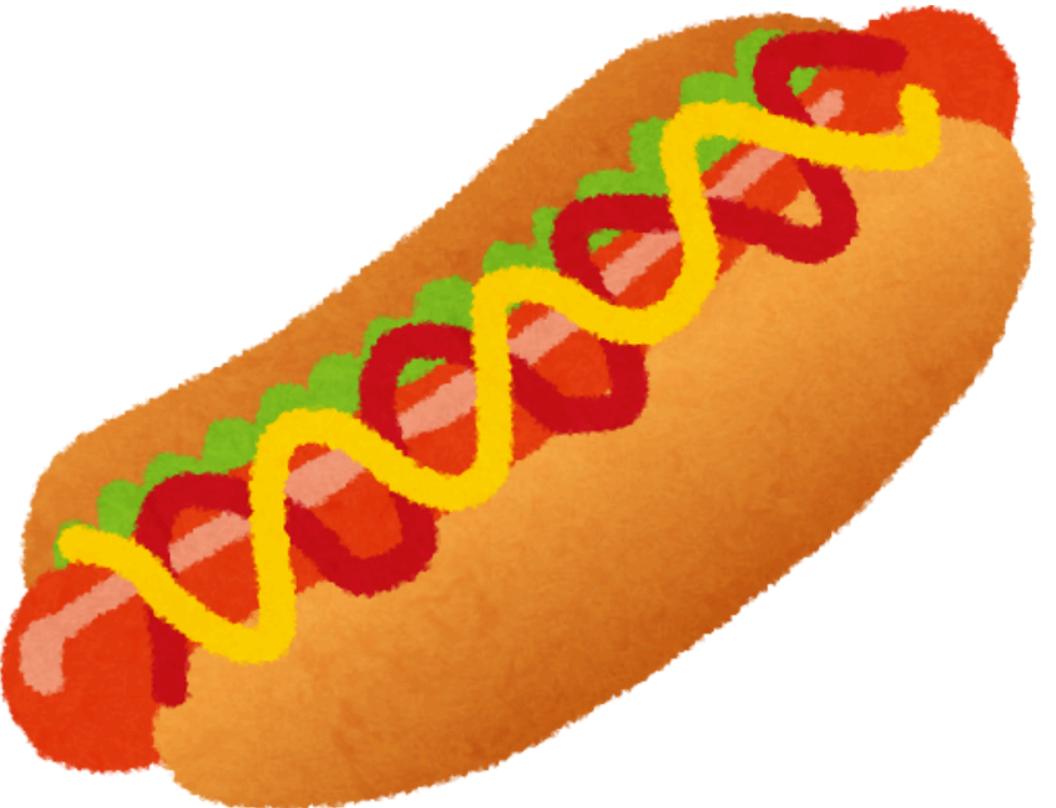
```
SELECT DISTINCT(M.DISH_NAME)
FROM MENU M
LEFT JOIN ORDER_DETAILS OD ON M.MENU_ID =
OD.MENU_ID
LEFT JOIN ORDERS O ON OD.ORDER_ID = O.ORDER_ID
    AND O.ORDER_DATE >= DATE_SUB(CURDATE(),
INTERVAL 6 MONTH)
WHERE O.ORDER_ID IS NULL;
```

	DISH_NAME
▶	BUTTER CHICKEN
	PANEER TIKKA
	GULAB JAMUN
	SPECIAL EGG CHICKEN BIRYANI
	CHAMPARAN MUTTON
	MARGHERITA PIZZA
	PASTA ALFREDO
	TIRAMISU



For each restaurant, find the most profitable dish (highest total sales amount).

RESTAURANT_NAME	DISH_NAME	TOTAL_SALES
BELLA ITALIA	MARGHERITA PIZZA	₹ 27000.00
BURGER HUB	BROWNIE SUNDAE	₹ 22000.00
DRAGON WOK	KUNG PAO CHICKEN	₹ 36000.00
GOLDEN CHOPSTICKS	KUNG PAO CHICKEN	₹ 3800.00
GOLDEN CHOPSTICKS	SPRING ROLLS	₹ 3800.00
GRILL & CHILL	BUFFALO WINGS	₹ 8400.00
HEAVENLY BAKES	SOURDOUGH SANDWICH	₹ 5000.00
LA PASTA FRESCA	TIRAMISU	₹ 14000.00



WITH DISHREVENUE AS (

SELECT

R.RESTAURANT_ID,

R.NAME AS RESTAURANT_NAME,

M.MENU_ID,

M.DISH_NAME,

SUM(OD.QUANTITY * M.PRICE) AS TOTAL_SALES

FROM MENU M

JOIN RESTAURANTS R ON M.RESTAURANT_ID = R.RESTAURANT_ID

JOIN ORDER_DETAILS OD ON M.MENU_ID = OD.MENU_ID

JOIN ORDERS O ON OD.ORDER_ID = O.ORDER_ID

GROUP BY R.RESTAURANT_ID, R.NAME, M.MENU_ID, M.DISH_NAME

),

RANKEDDISHES AS (

SELECT *,

**RANK() OVER (PARTITION BY RESTAURANT_ID ORDER BY
TOTAL_SALES DESC) AS RANKING**

FROM DISHREVENUE

)

SELECT RESTAURANT_NAME,DISH_NAME,

CONCAT('₹ ', TOTAL_SALES) AS TOTAL_SALES

FROM RANKEDDISHES

WHERE RANKING = 1

ORDER BY RESTAURANT_NAME;

