SQL：

CREATE DATABASE IF NOT EXISTS restaurant\_db;

USE restaurant\_db;

CREATE TABLE menu

(

dish\_id INT PRIMARY KEY AUTO\_INCREMENT,

dish\_name VARCHAR(50) NOT NULL,

price DECIMAL(10,2) NOT NULL,

description TEXT,

profit\_per\_dish DECIMAL(10,2)

);

CREATE TABLE orders

(

item\_id INT AUTO\_INCREMENT PRIMARY KEY,

order\_id VARCHAR(20) NOT NULL,

order\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

table\_id INT NOT NULL COMMENT '桌号',

dish\_id INT COMMENT '菜品ID',

quantity INT NOT NULL COMMENT '数量',

subtotal DECIMAL(10,2) NOT NULL COMMENT '小计',

profit DECIMAL(10,2) NOT NULL DEFAULT 0 COMMENT '单份利润',

create\_date DATE NOT NULL DEFAULT (CURRENT\_DATE()),

FOREIGN KEY (dish\_id) REFERENCES menu(dish\_id),

INDEX idx\_order\_time (order\_time),

INDEX idx\_create\_date (create\_date),

INDEX idx\_order\_id (order\_id)

);

CREATE TABLE sales\_statistics

(

dish\_id INT PRIMARY KEY,

total\_sales INT NOT NULL DEFAULT 0,

total\_revenue DECIMAL(10,2) NOT NULL DEFAULT 0,

total\_profit DECIMAL(10,2) NOT NULL DEFAULT 0,

FOREIGN KEY (dish\_id) REFERENCES menu(dish\_id)

);

DELIMITER $$

CREATE TRIGGER after\_menu\_insert

AFTER INSERT ON menu

FOR EACH ROW

BEGIN

INSERT INTO sales\_statistics (dish\_id, total\_sales, total\_revenue, total\_profit)

VALUES (NEW.dish\_id, 0, 0, 0);

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER update\_sales\_trigger

AFTER INSERT ON orders

FOR EACH ROW

BEGIN

UPDATE sales\_statistics

SET total\_sales = total\_sales + NEW.quantity,

total\_revenue = total\_revenue + NEW.subtotal,

total\_profit = total\_profit + NEW.profit

WHERE dish\_id = NEW.dish\_id;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER update\_order\_profit

BEFORE INSERT ON orders

FOR EACH ROW

BEGIN

SET NEW.profit = (

SELECT profit\_per\_dish FROM menu WHERE dish\_id = NEW.dish\_id

) \* NEW.quantity;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER before\_menu\_delete

BEFORE DELETE ON menu

FOR EACH ROW

BEGIN

UPDATE orders SET dish\_id = NULL WHERE dish\_id = OLD.dish\_id;

DELETE FROM sales\_statistics WHERE dish\_id = OLD.dish\_id;

END$$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE update\_menu\_price(

IN p\_dish\_id INT,

IN p\_new\_price DECIMAL(10,2)

)

BEGIN

IF p\_new\_price <= 0 THEN

SIGNAL SQLSTATE '45000';

END IF;

UPDATE menu SET price = p\_new\_price WHERE dish\_id = p\_dish\_id;

END$$

DELIMITER ;

CREATE OR REPLACE VIEW total\_restaurant\_profit AS

SELECT SUM(profit) AS total\_profit FROM orders;

INSERT INTO menu (dish\_name, price, profit\_per\_dish, description) VALUES

('宫保鸡丁', 38.00, 15.00, '传统川菜，鸡肉鲜嫩，花生酥脆'),

('鱼香肉丝', 36.00, 14.00, '酸甜可口，具有鱼香味但不含鱼肉'),

('糖醋排骨', 42.00, 18.00, '色泽红亮，酸甜开胃'),

('麻婆豆腐', 18.00, 8.00, '麻辣鲜香，豆腐嫩滑'),

('龙井虾仁', 68.00, 30.00, '浙菜经典，虾仁鲜嫩配以龙井茶清香'),

('扬州炒饭', 28.00, 12.00, '配料丰富，口感多样'),

('西湖牛肉羹', 22.00, 10.00, '细腻滑嫩，营养丰富'),

('珍珠奶茶', 12.00, 6.00, 'Q弹珍珠配以浓郁奶茶');