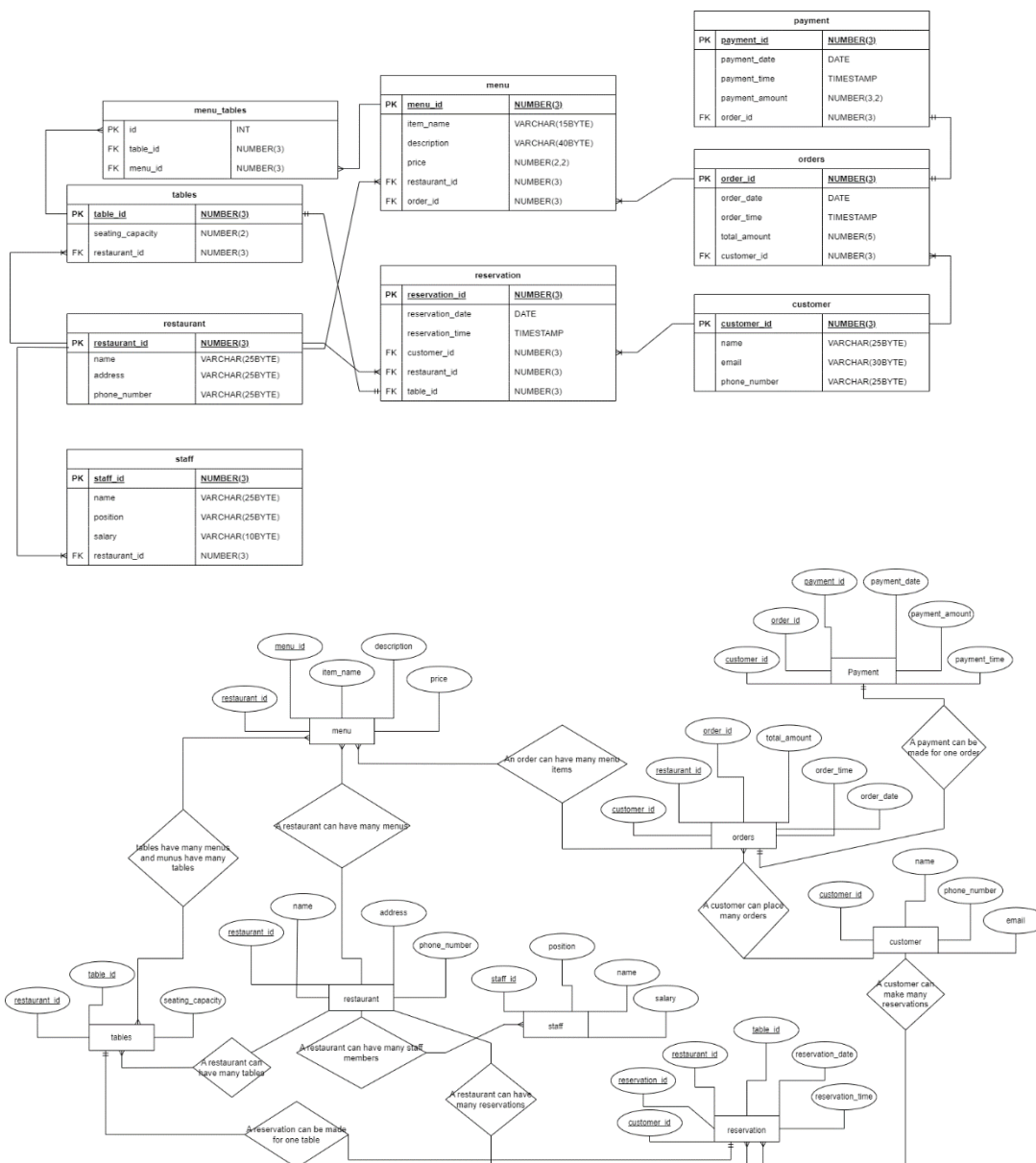


Úvod

Hlavnou úlohou tejto úlohy bolo vytvoriť databázu údajov o reštauráciách, ktorá by umožnila ich správu atď.



Druhá časť zadania

1. 2 pohľady s netriviálnym selektom nad jednou tabuľkou (nestačí použiť iba vymenovanie stĺpcov, treba použiť "niečo navyše" napr.: vstavané funkcie)

-- Vytvorenie pohľadu, ktorý počíta počet položiek menu v každej reštaurácii

```
CREATE VIEW menu_items_count AS
SELECT restaurant.name, COUNT(menu.menu_id) AS num_items
FROM restaurant
JOIN menu ON restaurant.restaurant_id = menu.restaurant_id
GROUP BY restaurant.name;
```

-- Vytvorenie pohľadu, ktorý zobrazuje celkovú sumu objednanú každým zákazníkom

```
CREATE VIEW customers_ordered_amount AS
SELECT customer.name, SUM(orders.total_amount) AS ordered_amount
FROM customer
JOIN orders ON customer.customer_id = orders.customer_id
GROUP BY customer.name;
```

View created.

View created.

NAME	NUM_ITEMS
Chinese Palace	3
Italiano	2

Download CSV

2 rows selected.

NAME	ORDERED_AMOUNT
Emma Davis	28
Alice Smith	33
Charlie Brown	17
Diana Lee	26
Bob Johnson	27

Download CSV

5 rows selected.

2. 3 pohľady so spájaním tabuliek(1x spojenie 3 tabuliek, 1x outer join, 1x spojenie aspoň 2 tabuliek);

-- Vytvorenie zobrazenia, ktoré kombinuje informácie o rezerváciách s informáciami o zákazníkoch a stoloch

```
CREATE VIEW reservation_info AS
SELECT reservation.reservation_id, reservation.reservation_date,
reservation.reservation_time,
customer.name AS customer_name, tables.seating_capacity
FROM reservation
JOIN customer ON reservation.customer_id = customer.customer_id
JOIN tables ON reservation.table_id = tables.table_id;
```

-- Vytvorenie zobrazenia, ktoré kombinuje informácie o položkách menu a objednávkach s informáciami o zákazníkoch

```
CREATE VIEW menu_orders AS
SELECT menu.menu_id, menu.item_name, menu.price,
orders.order_date, orders.order_time,
orders.total_amount, customer.name AS customer_name
FROM menu
LEFT JOIN orders ON menu.order_id = orders.order_id
LEFT JOIN customer ON orders.customer_id = customer.customer_id;
```

-- Vytvorenie zobrazenia, ktoré kombinuje informácie o platbách a objednávkach s informáciami o zákazníkoch

```
CREATE VIEW payment_info AS
SELECT payment.payment_id, payment.payment_date,
payment.payment_time,
orders.order_date, customer.name AS customer_name
FROM payment
JOIN orders ON payment.order_id = orders.order_id
JOIN customer ON orders.customer_id = customer.customer_id;
```

View created.

View created.

RESERVATION_ID	RESERVATION_DATE	RESERVATION_TIME	CUSTOMER_NAME	SEATING_CAPACITY
1	05-JAN-22	05-JAN-22 07.00.00.000000 PM	Alice Smith	4
2	05-JAN-22	05-JAN-22 07.00.00.000000 PM	Bob Johnson	6
3	06-JAN-22	06-JAN-22 06.30.00.000000 PM	Charlie Brown	2
4	06-JAN-22	06-JAN-22 06.30.00.000000 PM	Diana Lee	4
5	07-JAN-22	07-JAN-22 08.00.00.000000 PM	Emma Davis	8

Download CSV

5 rows selected.

MENU_ID	ITEM_NAME	PRICE	ORDER_DATE	ORDER_TIME	TOTAL_AMOUNT	CUSTOMER_NAME
1	Spaghetti	8.9	05-JAN-22	05-JAN-22 07.15.00.000000 PM	33	Alice Smith
2	Pizza	4.5	05-JAN-22	05-JAN-22 07.15.00.000000 PM	33	Alice Smith
3	Egg rolls	6.9	05-JAN-22	05-JAN-22 07.15.00.000000 PM	27	Bob Johnson
4	Kung Pao Chicken	8.7	05-JAN-22	05-JAN-22 07.15.00.000000 PM	27	Bob Johnson
5	Kung Lao Meat	8.9	05-JAN-22	05-JAN-22 07.15.00.000000 PM	27	Bob Johnson

Download CSV

PAYMENT_ID	PAYMENT_DATE	PAYMENT_TIME	ORDER_DATE	CUSTOMER_NAME
1	05-JAN-22	05-JAN-22 07.30.00.000000 PM	05-JAN-22	Alice Smith
2	05-JAN-22	05-JAN-22 07.30.00.000000 PM	05-JAN-22	Bob Johnson
3	06-JAN-22	06-JAN-22 07.00.00.000000 PM	06-JAN-22	Charlie Brown
4	06-JAN-22	06-JAN-22 07.00.00.000000 PM	06-JAN-22	Diana Lee
5	06-APR-22	06-JAN-22 07.50.00.000000 PM	06-JAN-22	Diana Lee

Download CSV

5 rows selected.

3. 2 pohľady s použitím agregáčnych funkcií alebo zoskupenia;

-- Vytvorenie pohľadu, ktorý zobrazuje celkové tržby pre každú reštauráciu

```
CREATE VIEW total_sales AS
SELECT restaurant.name, SUM(menu.price) AS total_sales
FROM restaurant
JOIN menu ON restaurant.restaurant_id = menu.restaurant_id
GROUP BY restaurant.name;
```

-- vytvorte pohľad, ktorý spočíta počet zamestnancov na jednotlivých pozíciách v každej reštaurácii

```
CREATE VIEW staff_count AS
SELECT restaurant.name, staff.position, COUNT(staff.staff_id) AS
staff_count
FROM restaurant
JOIN staff ON restaurant.restaurant_id = staff.restaurant_id
GROUP BY restaurant.name, staff.position;
```

View created.

View created.

NAME	TOTAL_SALES
Chinese Palace	24.5
Italiano	13.4

Download CSV

2 rows selected.

NAME	POSITION	STAFF_COUNT
Italiano	Waiter	1
Chinese Palace	Manager	1
The Red Pepper	Manager	1
Italiano	Chef	1
The Red Pepper	Waitress	1

Download CSV

5 rows selected.

4. 1 pohľad s použitím množinových operácií;

-- Vytvorenie zobrazenia, ktoré kombinuje informácie
o zamestnancoch a zákazníkoch

```
CREATE VIEW staff_customer AS  
SELECT staff_id, name, position, salary  
FROM staff  
UNION  
SELECT customer_id, name, 'Customer', NULL  
FROM customer;
```

View created.

STAFF_ID	NAME	POSITION	SALARY
1	Alice Smith	Customer	-
1	John Smith	Manager	5000
2	Bob Johnson	Customer	-
2	Jane Doe	Waitress	2000
3	Charlie Brown	Customer	-
3	Mario Rossi	Chef	3000
4	Diana Lee	Customer	-
4	Luigi Bianchi	Waiter	2500
5	Emma Davis	Customer	-
5	Li Xu	Manager	6000

Download CSV

10 rows selected.

5. 2 pohľady s použitím vnorených selektov;

-- Vytvorenie pohľadu, ktorý počíta počet rezervácií v každej reštaurácii

```
CREATE VIEW restaurant_reservations AS
SELECT r.restaurant_id, r.name, COUNT(reservation_id) AS
num_reservations
FROM restaurant r
JOIN reservation re ON r.restaurant_id = re.restaurant_id
GROUP BY r.restaurant_id, r.name;
```

-- Vytvorenie zobrazenia, ktoré zobrazuje celkovú sumu, ktorú každý zákazník utratil

```
CREATE VIEW customer_orders AS
SELECT c.customer_id, c.name, SUM(m.price) AS total_amount
FROM customer c
JOIN orders o ON c.customer_id = o.customer_id
JOIN menu m ON o.order_id = m.order_id
GROUP BY c.customer_id, c.name;
```

View created.

View created.

RESTAURANT_ID	NAME	NUM_RESERVATIONS
3	Chinese Palace	1
2	Italiano	2
1	The Red Pepper	2

Download CSV

3 rows selected.

CUSTOMER_ID	NAME	TOTAL_AMOUNT
1	Alice Smith	13.4
2	Bob Johnson	24.5

Download CSV

2 rows selected.

6. vytvorte 1 sekvenciuna generovanie primárnych kľúčov a trigger, ktorý bude vkladat' hodnoty do príslušných tabuliek

-- Vytvorenie sekvencie na generovanie ID reštaurácií

```
CREATE SEQUENCE restaurant_seq  
START WITH 6  
INCREMENT BY 1  
NOCACHE  
NOCYCLE;
```

-- Vytvorenie spúšťača, ktorý automaticky priradí nové ID reštaurácie pri vložení novej reštaurácie

```
CREATE OR REPLACE TRIGGER restaurant_trigger  
BEFORE INSERT ON restaurant  
FOR EACH ROW  
BEGIN  
    SELECT restaurant_seq.NEXTVAL  
    INTO :new.restaurant_id  
    FROM dual;  
END;  
/
```

```
INSERT INTO restaurant (name, address, phone_number) VALUES  
( 'Te Restaurant', 'Te Location', '555-4742');  
SELECT * FROM restaurant;
```

RESTAURANT_ID	NAME	ADDRESS	PHONE_NUMBER
1	The Red Pepper	123 Main St	555-1234
2	Italiano	456 Oak St	555-5678
3	Chinese Palace	789 Elm St	555-9012
4	Indian Kaif	456 Elm St	555-3712
5	The best Place	Moz-Ar St	555-7015
6	Te Restaurant	Te Location	555-4742

[Download CSV](#)

7. vytvorte 1 ľubovoľný triggerokrem typu triggera uvedeného vyššie (trigger musí obsahovať buď podmienku, cyklus alebo výnimku).

```
-- Vytvorenie spúšťača, ktorý aktualizuje ceny položiek menu pri
zmene celkovej sumy objednávky
CREATE OR REPLACE TRIGGER menu_price_update
AFTER UPDATE OF total_amount ON orders
FOR EACH ROW
BEGIN
    UPDATE menu
    SET price = price * :new.total_amount / :old.total_amount
    WHERE order_id = :new.order_id;
END;
/

INSERT INTO orders (order_id, total_amount) VALUES (6, 10);
INSERT INTO menu (menu_id, order_id, item_name, price)
VALUES (6, 6, 'Burger', 8.5);

-- Check the initial price of the menu item
SELECT price FROM menu WHERE order_id = 6 AND item_name =
'Burger';

-- Update the total amount of the order
UPDATE orders SET total_amount = 5 WHERE order_id = 6;

-- Check if the trigger has updated the price of the menu item
SELECT price FROM menu WHERE order_id = 6 AND item_name =
'Burger';
```

PRICE
8.5
Download CSV
1 row(s) updated.
PRICE
4.25
Download CSV

Záver

Netriviálne selekty	2/2
Spájanie tabuliek	3/3
Použitie agregáčnych funkcií alebo zoskupenia	2/2
Množinová operácia	2/2
Vnorené selekty	2/2
Sekvencia + trigger	2/2
Trigger	2/2
Celkovo	15/15