SEPTEMBER 27, 2024

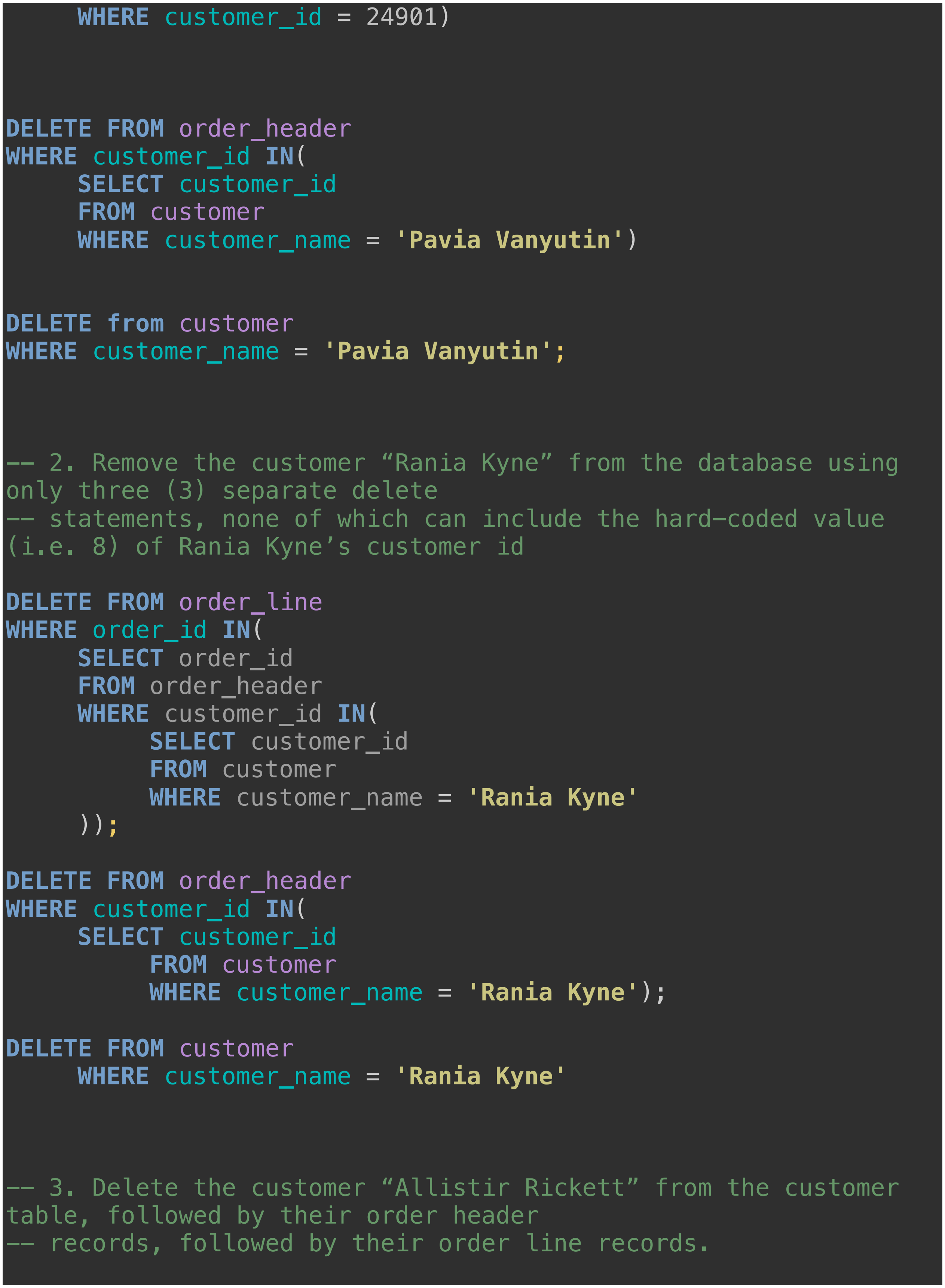
LAB 3

ASSIGNMENT

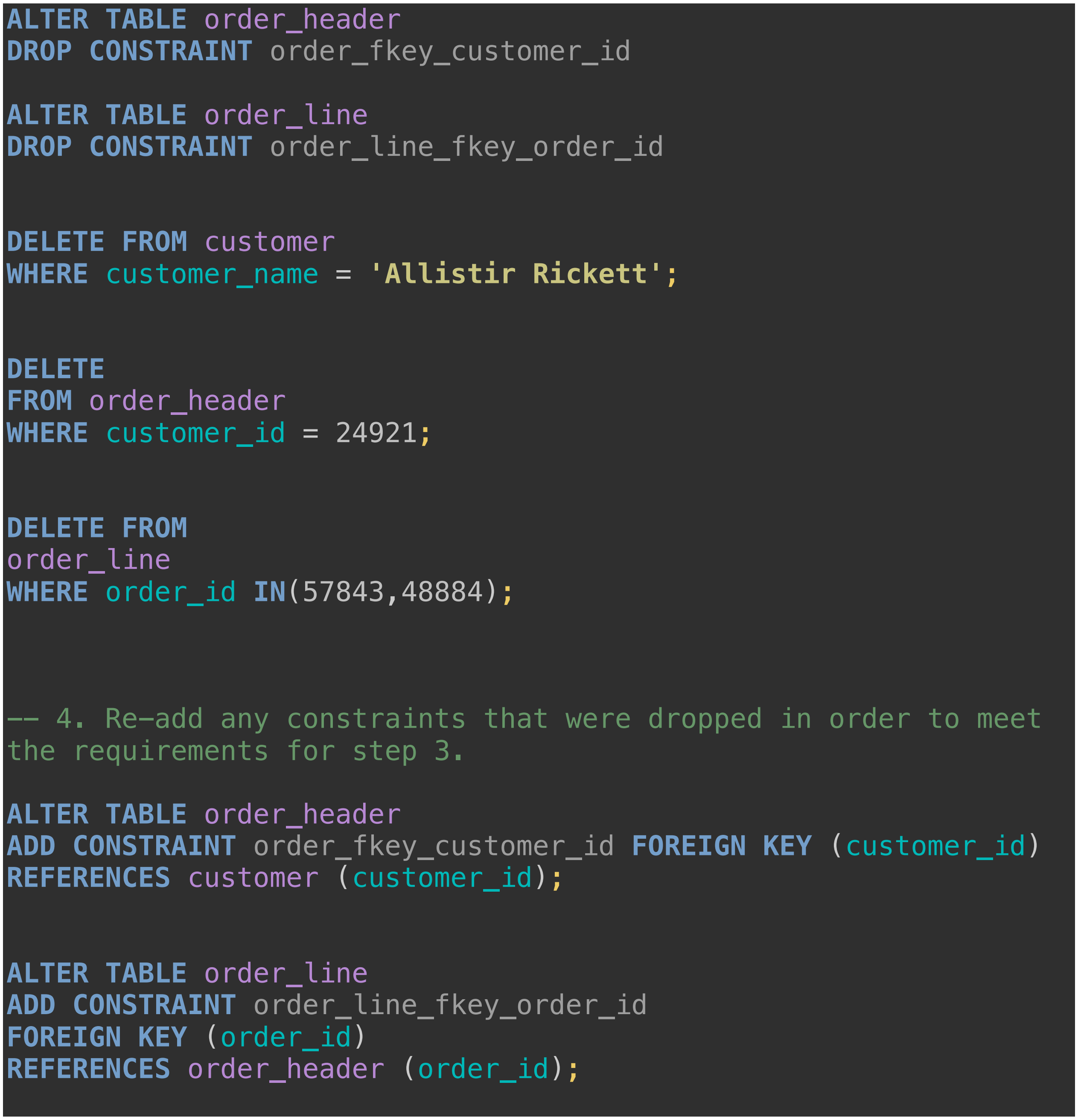
ALI ABBAS LADHA
U1537479
IS 6420-001 Fall 2024 Database Theory/Design

```
-- Ali Abbas Ladha
-- September 27 2024
-- Lab 3


-- Guided Exercise (80% of grade)
-- 1. List names of products that have been ordered. One name
appears exactly one time. Order product names in ascending
order.

SELECT DISTINCT product_name
FROM product
WHERE product_id IN(
     SELECT product_id
     FROM
     order_line)
ORDER BY product_name ASC;


-- 2. List emails of customers that have placed orders after
October 27, 2023. One ID appears exactly one time. Order
customer IDs in ascending order.


SELECT email
FROM customer
WHERE customer_id IN(
     SELECT DISTINCT customer_id
     FROM order_header
     WHERE order_date > '2023-10-27')
ORDER BY customer_id ASC;



-- 3. List all customer names for customers who are from Utah
and whose first name starts with the letter ‘L’.
SELECT customer_name
FROM customer
WHERE state_province ='Utah' AND customer_name LIKE 'L%';



-- 4. List the product name, product price and product price
after 10% discount.

SELECT product_name,
     product_price,
     ROUND(product_price * 0.9,2) AS discounted_price
FROM product


-- 5. List the number of products with a prices below $50.

SELECT COUNT(*) AS number_of_products
FROM product
WHERE product_price < 50

-- 6. List name and price for all products that have been
purchased on order 63589. Use a subquery and IN to implement
this query.
SELECT product_name, product_price
FROM product
WHERE product_id IN(
     SELECT product_id
     FROM order_line
     WHERE order_id = 63589
)

-- 7. List the order id and the total quantity of items for each
order after January 1, 2024. Sort the result by the total
quantity descending.
SELECT order_id, SUM(quantity)
FROM order_line
WHERE order_id IN(
     SELECT order_id
     FROM order_header
     WHERE order_date > '2024-01-01'
)
GROUP BY order_id
ORDER BY SUM(quantity) DESC;


-- Challenge Exercise - Part 1 (10% of grade)
-- 1. Select all rows from the product table, but add a column
called is_best_seller where 1 indicates
-- products have orderd in at least 5 orders, and 0 indicates
products that have been ordered 4 or
-- less times. The result should have the “best selling” product
first, then those are not best
-- sellers. Within these two groups, sort by state/province
descending, then city ascending.
```

```sql
SELECT product_name, product_line, product_price,
product_status, customer.state_province, customer.city,
     CASE WHEN COUNT(product_id) >= 5 THEN 1 ELSE 0
     END AS is_best_seller
FROM product
LEFT JOIN order_line
USING(product_id)
LEFT JOIN order_header
USING(order_id)
LEFT JOIN customer
USING(customer_id)
GROUP BY 1,2,3,4,5,6
ORDER BY is_best_seller DESC, state_province DESC, city ASC;




-- 2. List the order id, date and total dollar amount for the
top 15 orders by dollar amount. Sort the
-- result by the total amount descending, then the date
ascending. (hint: you will need to join
-- tables to get product price and quantity)

SELECT oh.order_id,
oh.order_date,
(p.product_price * ol.quantity) AS total_amount
FROM order_line AS ol
LEFT JOIN product AS p
USING(product_id)
LEFT JOIN order_header AS oh
USING(order_id)
WHERE (p.product_price * ol.quantity) IS NOT NULL
ORDER BY total_amount DESC, oh.order_date ASC
LIMIT 15;


-- Challenge Exercise – Part 2 (10% of grade)
-- 1. Remove the customer “Pavia Vanyutin” from the database.

DELETE FROM order_line
WHERE order_id IN(
     SELECT order_id
     FROM order_header
```



```
     WHERE customer_id = 24901)



DELETE FROM order_header
WHERE customer_id IN(
     SELECT customer_id
     FROM customer
     WHERE customer_name = 'Pavia Vanyutin')


DELETE from customer
WHERE customer_name = 'Pavia Vanyutin';



-- 2. Remove the customer “Rania Kyne” from the database using
only three (3) separate delete
-- statements, none of which can include the hard-coded value
(i.e. 8) of Rania Kyne’s customer id

DELETE FROM order_line
WHERE order_id IN(
     SELECT order_id
     FROM order_header
     WHERE customer_id IN(
          SELECT customer_id
          FROM customer
          WHERE customer_name = 'Rania Kyne'
     ));

DELETE FROM order_header
WHERE customer_id IN(
     SELECT customer_id
          FROM customer
          WHERE customer_name = 'Rania Kyne');

DELETE FROM customer
     WHERE customer_name = 'Rania Kyne'



-- 3. Delete the customer “Allistir Rickett” from the customer
table, followed by their order header
-- records, followed by their order line records.
```



```
ALTER TABLE order_header
DROP CONSTRAINT order_fkey_customer_id

ALTER TABLE order_line
DROP CONSTRAINT order_line_fkey_order_id


DELETE FROM customer
WHERE customer_name = 'Allistir Rickett';


DELETE
FROM order_header
WHERE customer_id = 24921;


DELETE FROM
order_line
WHERE order_id IN(57843,48884);



-- 4. Re-add any constraints that were dropped in order to meet
the requirements for step 3.

ALTER TABLE order_header
ADD CONSTRAINT order_fkey_customer_id FOREIGN KEY (customer_id)
REFERENCES customer (customer_id);


ALTER TABLE order_line
ADD CONSTRAINT order_line_fkey_order_id
FOREIGN KEY (order_id)
REFERENCES order_header (order_id);
```