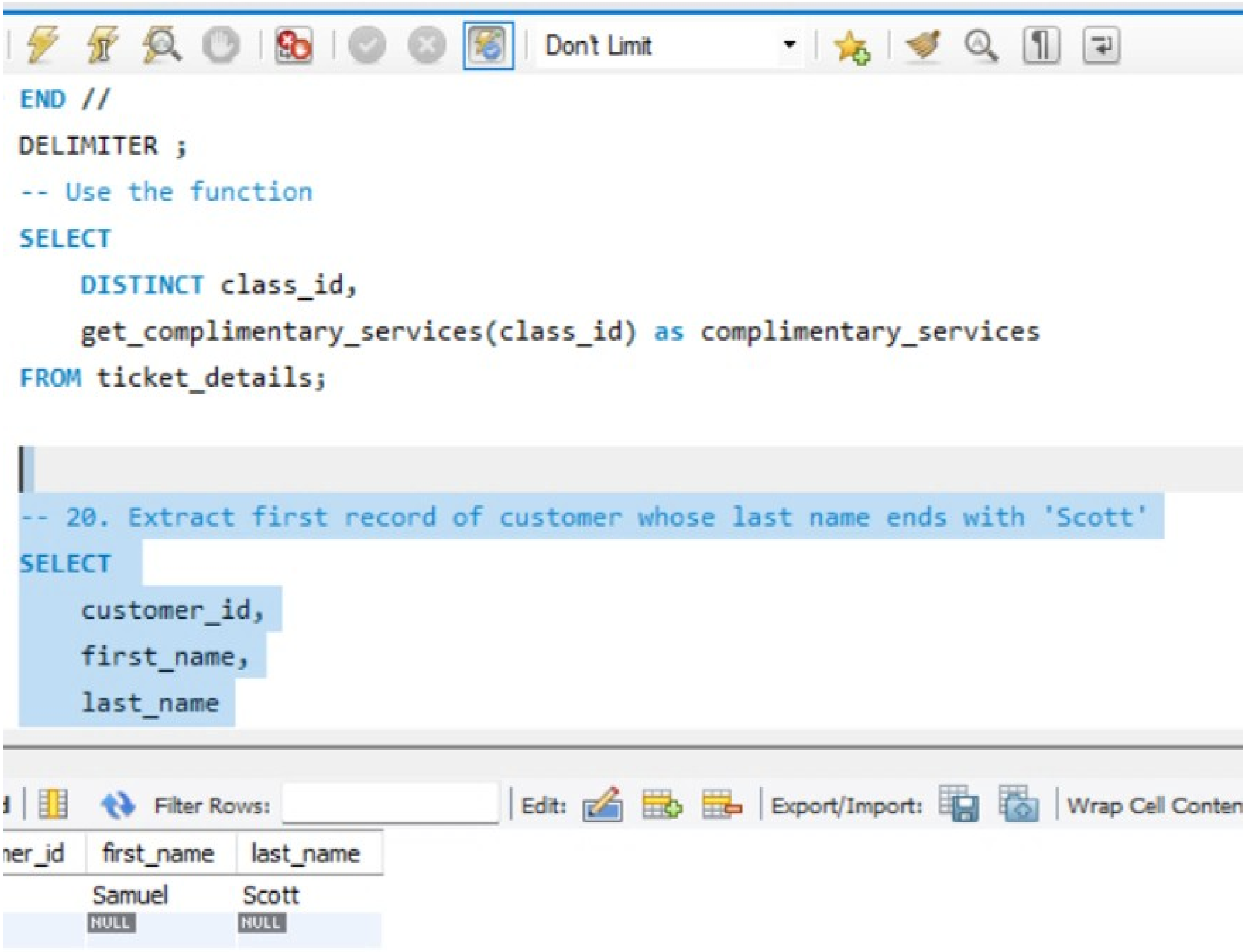
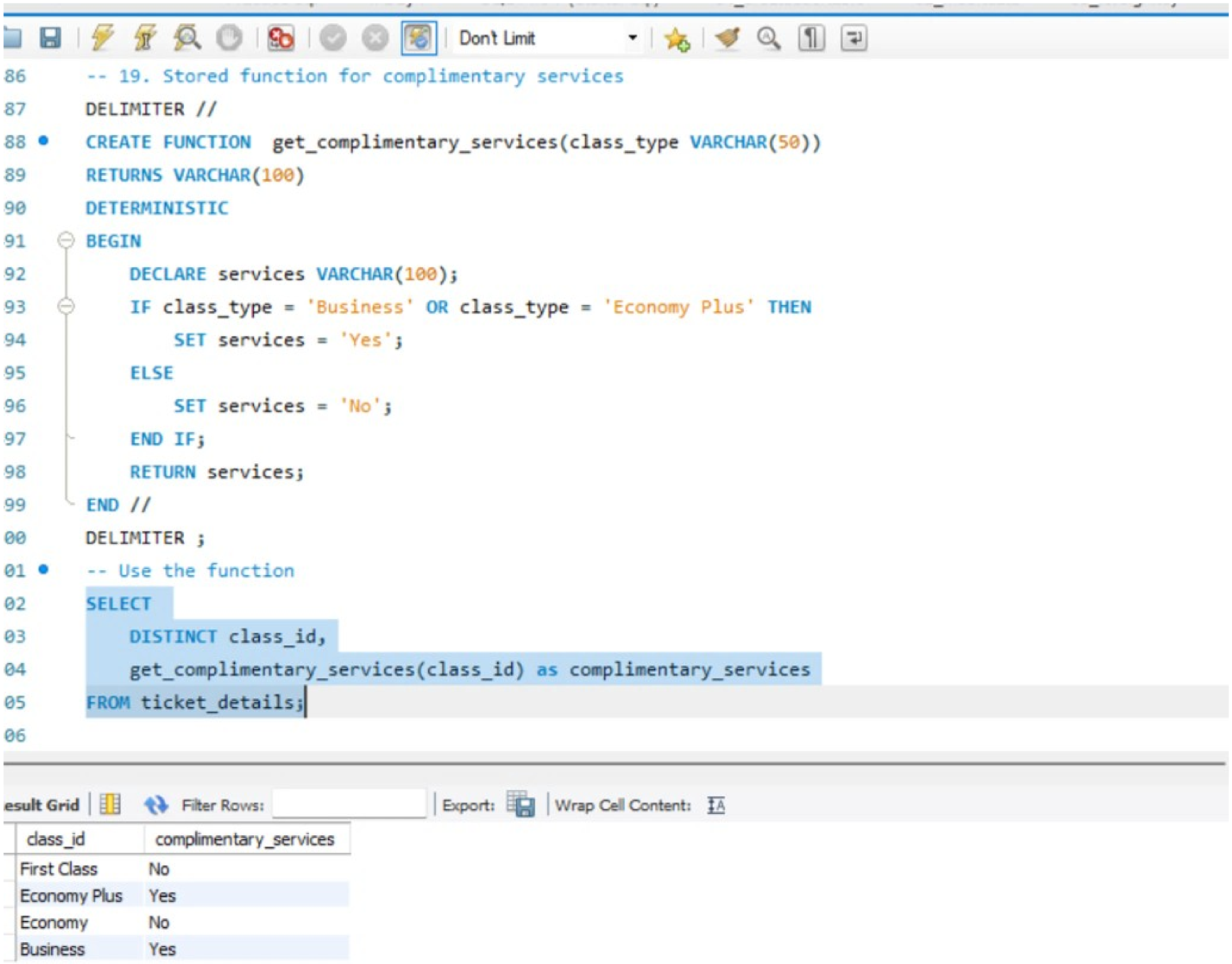


```
END //
DELIMITER ;
-- Use the function
SELECT
    DISTINCT class_id,
    get_complimentary_services(class_id) as complimentary_services
FROM ticket_details;


-- 20. Extract first record of customer whose last name ends with 'Scott'
SELECT
    customer_id,
    first_name,
    last_name
```

```
ner_id  first_name  last_name
        Samuel      Scott
        NULL        NULL
```

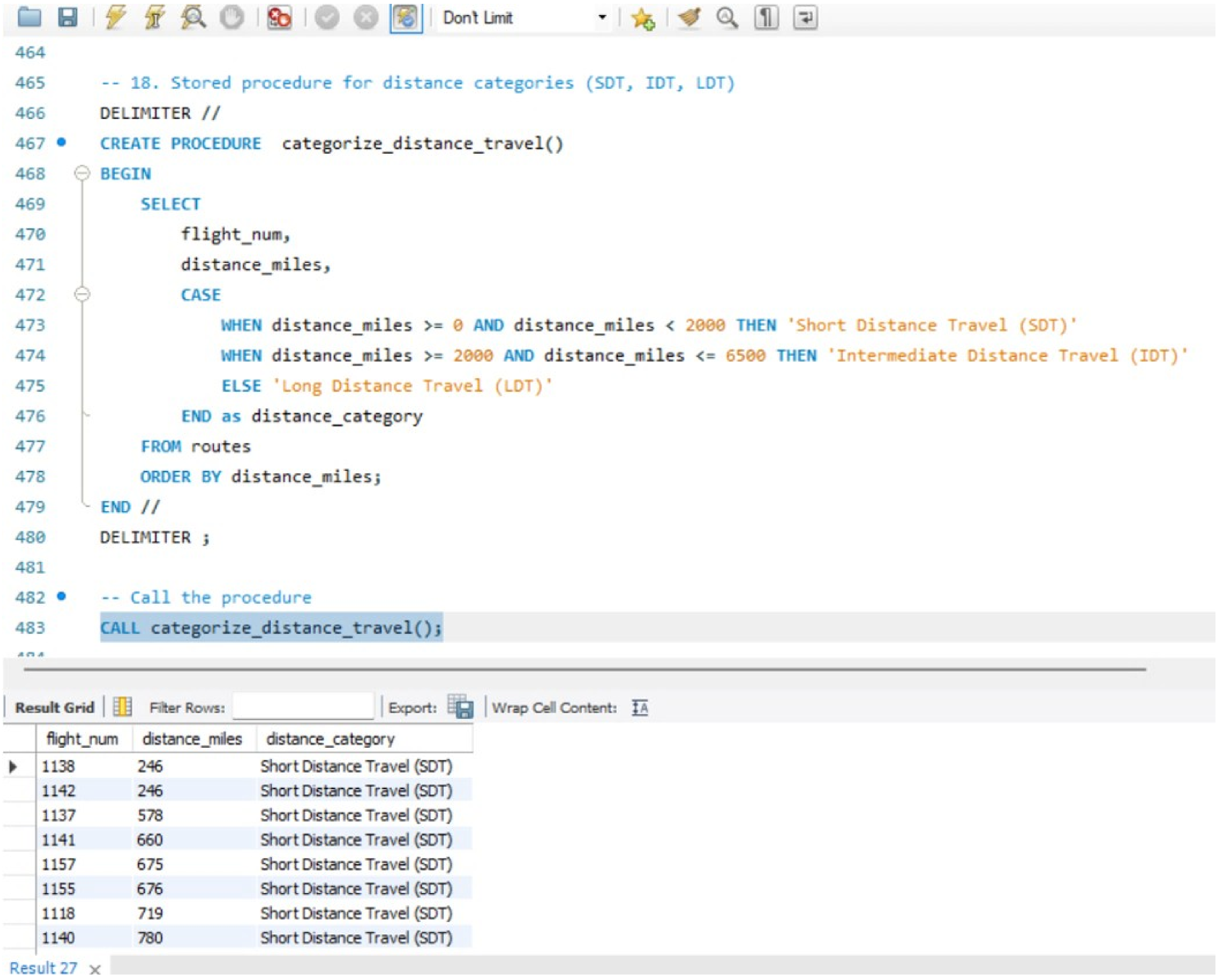


Don't Limit

```
-- 19. Stored function for complimentary services
DELIMITER //
CREATE FUNCTION  get_complimentary_services(class_type VARCHAR(50))
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
    DECLARE services VARCHAR(100);
    IF class_type = 'Business' OR class_type = 'Economy Plus' THEN
        SET services = 'Yes';
    ELSE
        SET services = 'No';
    END IF;
    RETURN services;
END //
DELIMITER ;
-- Use the function
SELECT
    DISTINCT class_id,
    get_complimentary_services(class_id) as complimentary_services
FROM ticket_details;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

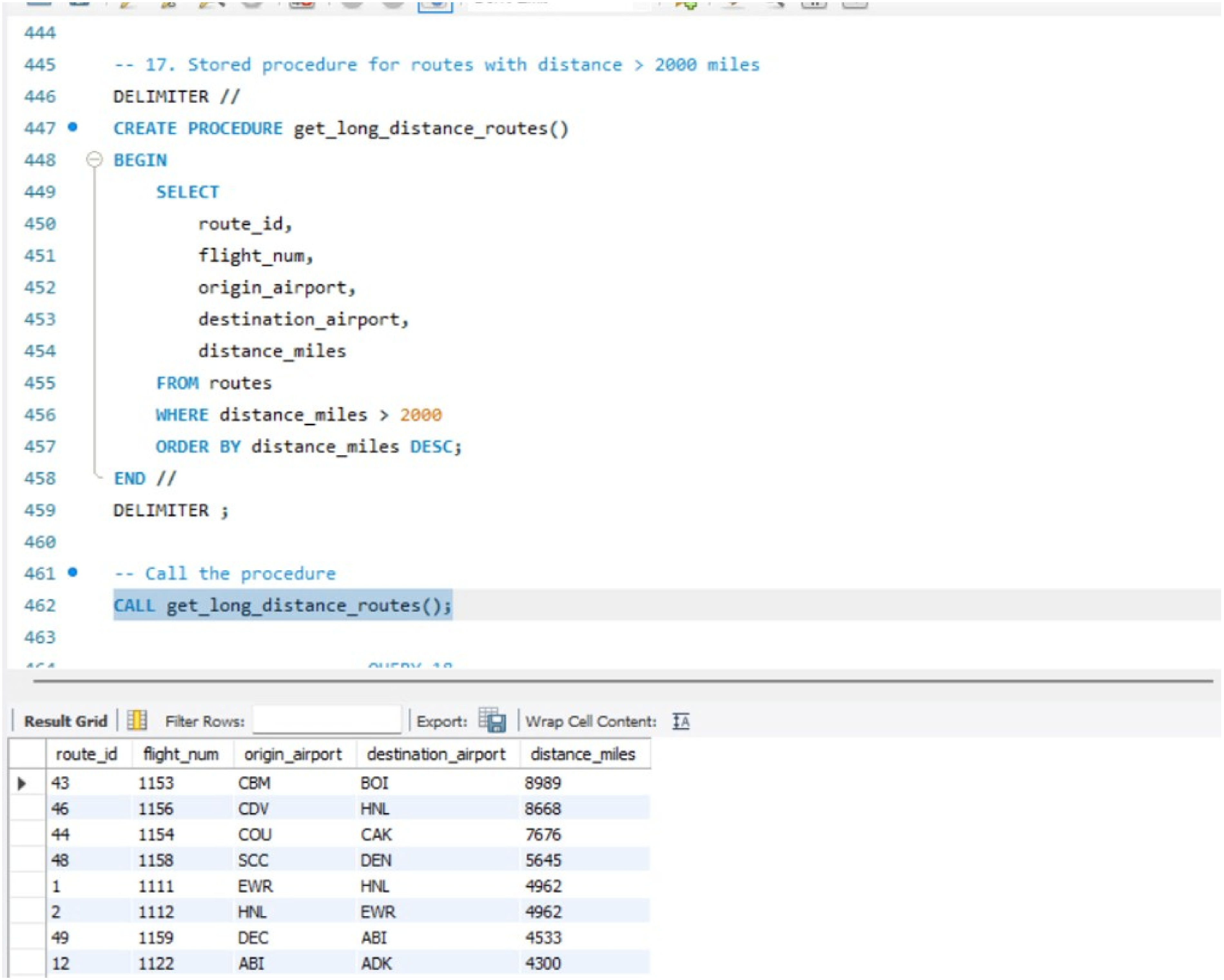
class_id	complimentary_services
First Class	No
Economy Plus	Yes
Economy	No
Business	Yes



```
-- 18. Stored procedure for distance categories (SDT, IDT, LDT)
DELIMITER //
CREATE PROCEDURE  categorize_distance_travel()
BEGIN
    SELECT
        flight_num,
        distance_miles,
        CASE
            WHEN distance_miles >= 0 AND distance_miles < 2000 THEN 'Short Distance Travel (SDT)'
            WHEN distance_miles >= 2000 AND distance_miles <= 6500 THEN 'Intermediate Distance Travel (IDT)'
            ELSE 'Long Distance Travel (LDT)'
        END as distance_category
    FROM routes
    ORDER BY distance_miles;
END //
DELIMITER ;

-- Call the procedure
CALL categorize_distance_travel();
```

flight_num	distance_miles	distance_category
1138	246	Short Distance Travel (SDT)
1142	246	Short Distance Travel (SDT)
1137	578	Short Distance Travel (SDT)
1141	660	Short Distance Travel (SDT)
1157	675	Short Distance Travel (SDT)
1155	676	Short Distance Travel (SDT)
1118	719	Short Distance Travel (SDT)
1140	780	Short Distance Travel (SDT)

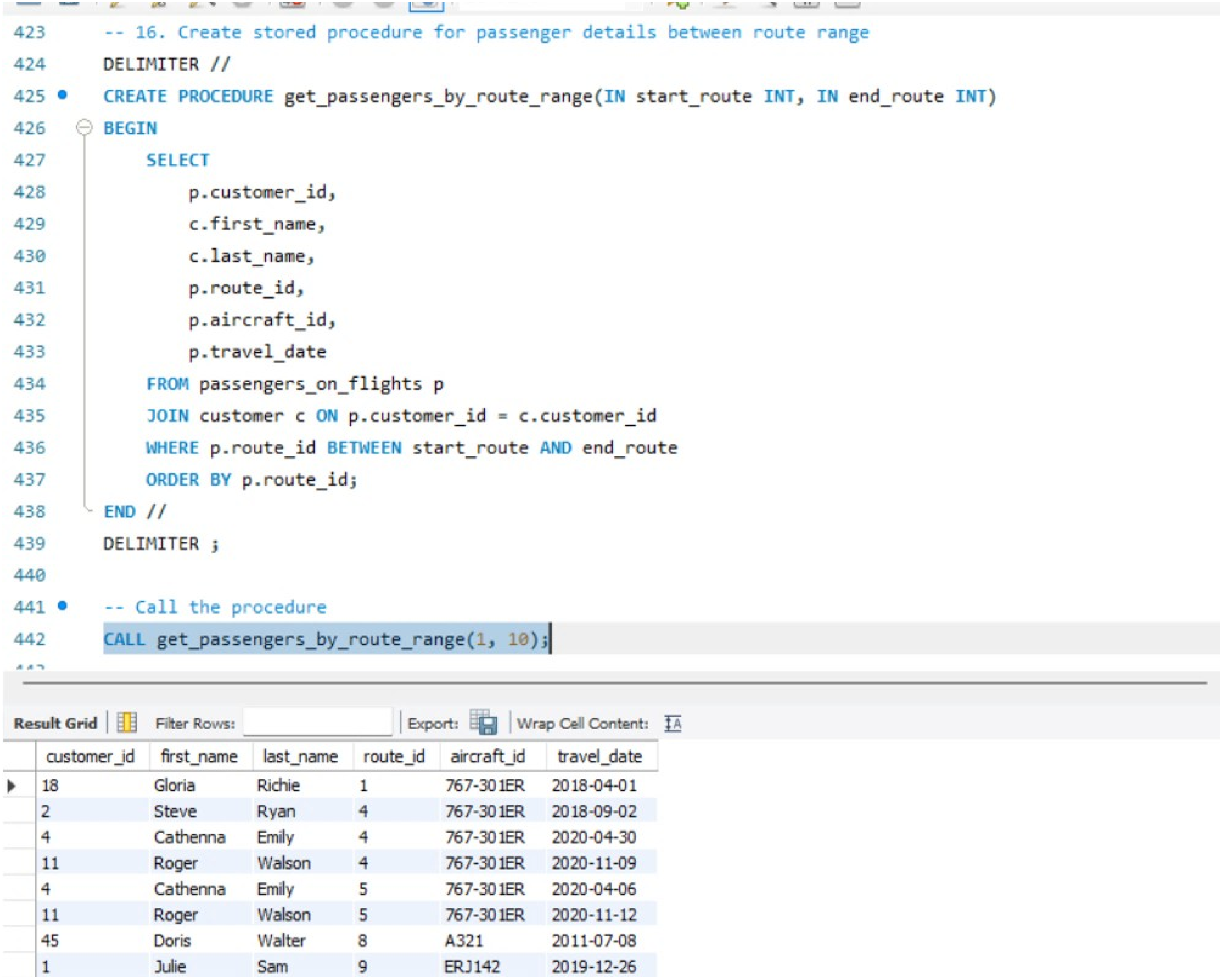


```
-- 17. Stored procedure for routes with distance > 2000 miles
DELIMITER //
CREATE PROCEDURE get_long_distance_routes()
BEGIN
    SELECT
        route_id,
        flight_num,
        origin_airport,
        destination_airport,
        distance_miles
    FROM routes
    WHERE distance_miles > 2000
    ORDER BY distance_miles DESC;
END //
DELIMITER ;

-- Call the procedure
CALL get_long_distance_routes();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

route_id	flight_num	origin_airport	destination_airport	distance_miles
43	1153	CBM	BOI	8989
46	1156	CDV	HNL	8668
44	1154	COU	CAK	7676
48	1158	SCC	DEN	5645
1	1111	EWR	HNL	4962
2	1112	HNL	EWR	4962
49	1159	DEC	ABI	4533
12	1122	ABI	ADK	4300

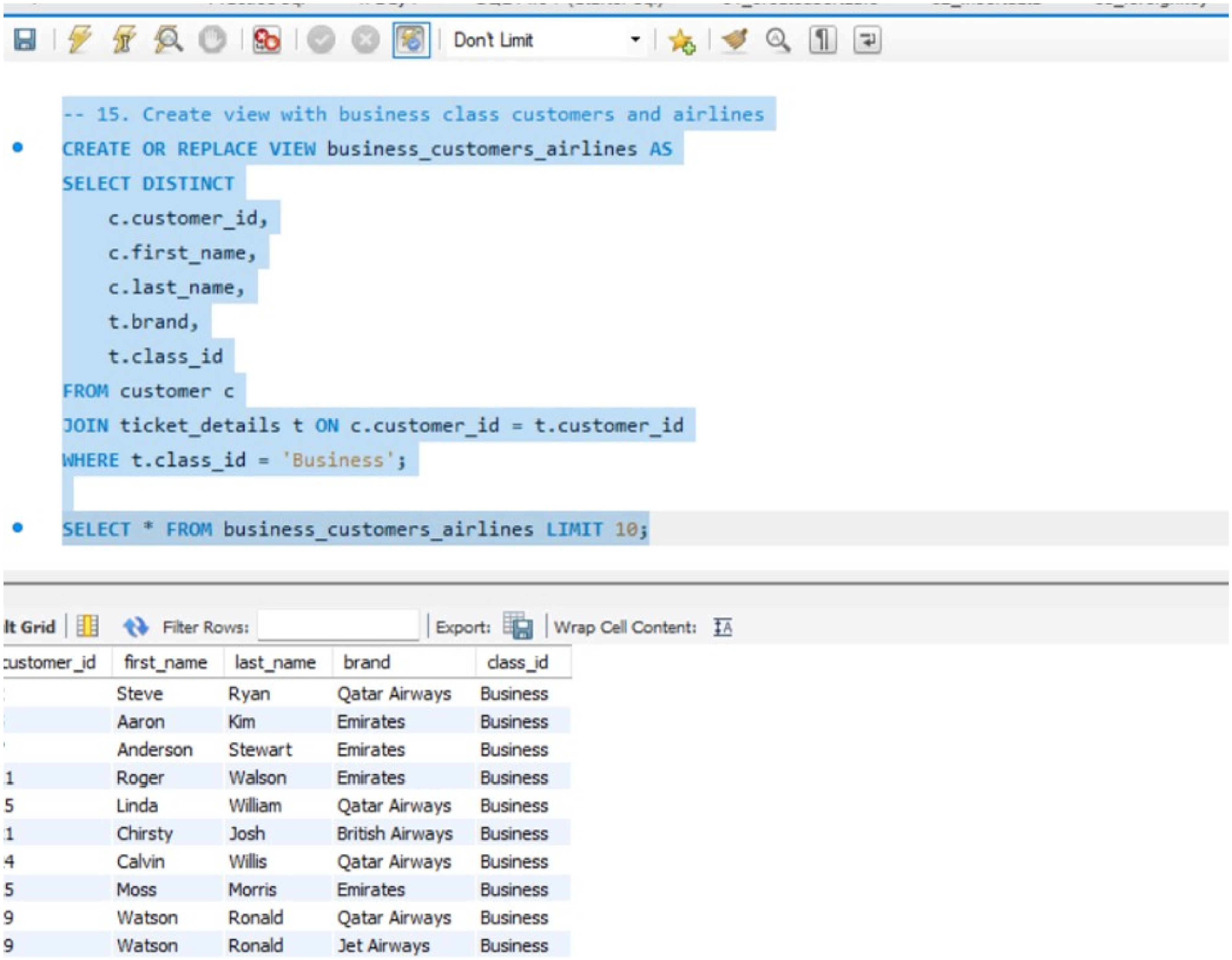


```sql
-- 16. Create stored procedure for passenger details between route range
DELIMITER //
CREATE PROCEDURE get_passengers_by_route_range(IN start_route INT, IN end_route INT)
BEGIN
    SELECT
        p.customer_id,
        c.first_name,
        c.last_name,
        p.route_id,
        p.aircraft_id,
        p.travel_date
    FROM passengers_on_flights p
    JOIN customer c ON p.customer_id = c.customer_id
    WHERE p.route_id BETWEEN start_route AND end_route
    ORDER BY p.route_id;
END //
DELIMITER ;

-- Call the procedure
CALL get_passengers_by_route_range(1, 10);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

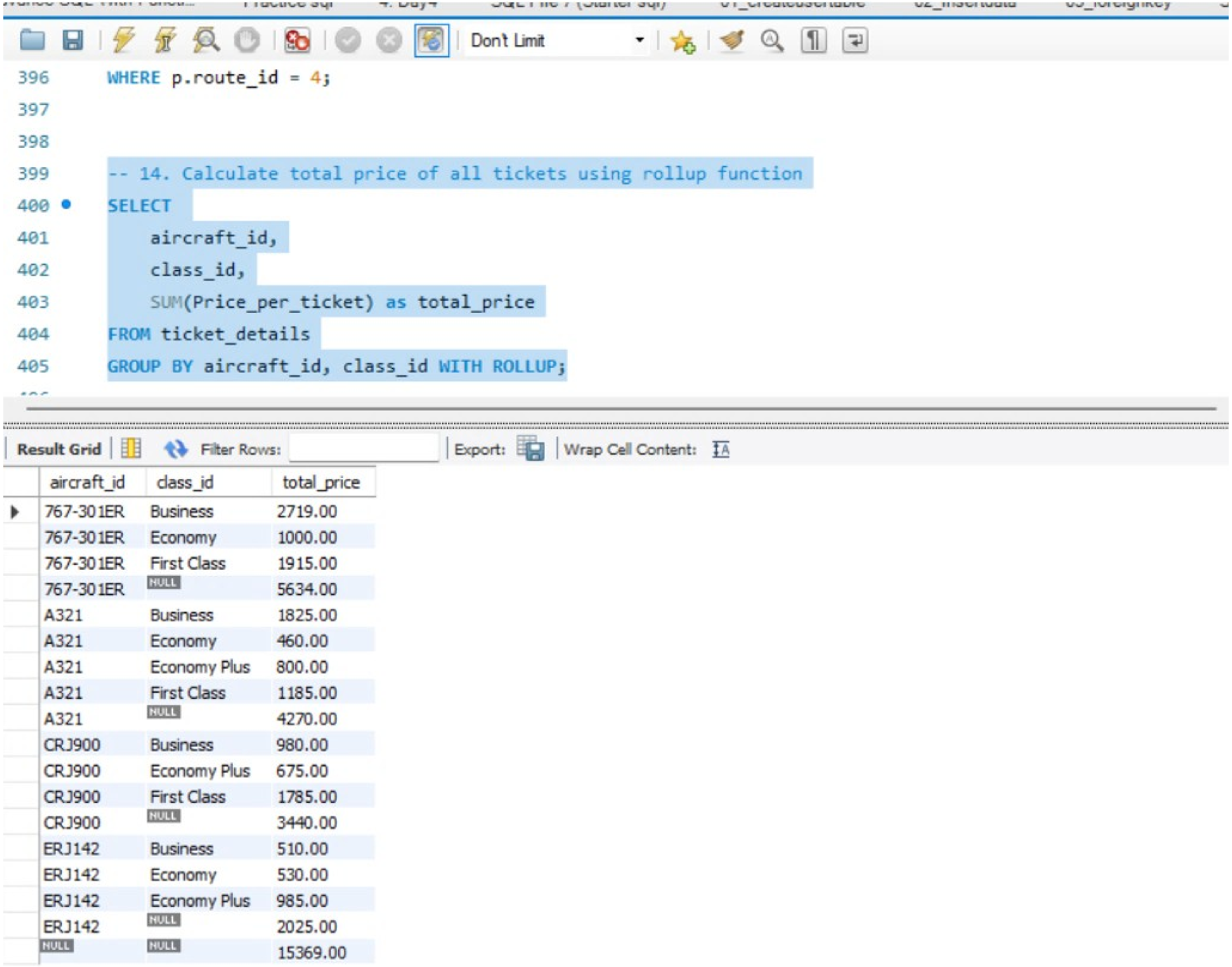
customer_id	first_name	last_name	route_id	aircraft_id	travel_date
18	Gloria	Richie	1	767-301ER	2018-04-01
2	Steve	Ryan	4	767-301ER	2018-09-02
4	Cathenna	Emily	4	767-301ER	2020-04-30
11	Roger	Walson	4	767-301ER	2020-11-09
4	Cathenna	Emily	5	767-301ER	2020-04-06
11	Roger	Walson	5	767-301ER	2020-11-12
45	Doris	Walter	8	A321	2011-07-08
1	Julie	Sam	9	ERJ142	2019-12-26



```sql
-- 15. Create view with business class customers and airlines
CREATE OR REPLACE VIEW business_customers_airlines AS
SELECT DISTINCT
    c.customer_id,
    c.first_name,
    c.last_name,
    t.brand,
    t.class_id
FROM customer c
JOIN ticket_details t ON c.customer_id = t.customer_id
WHERE t.class_id = 'Business';

SELECT * FROM business_customers_airlines LIMIT 10;
```

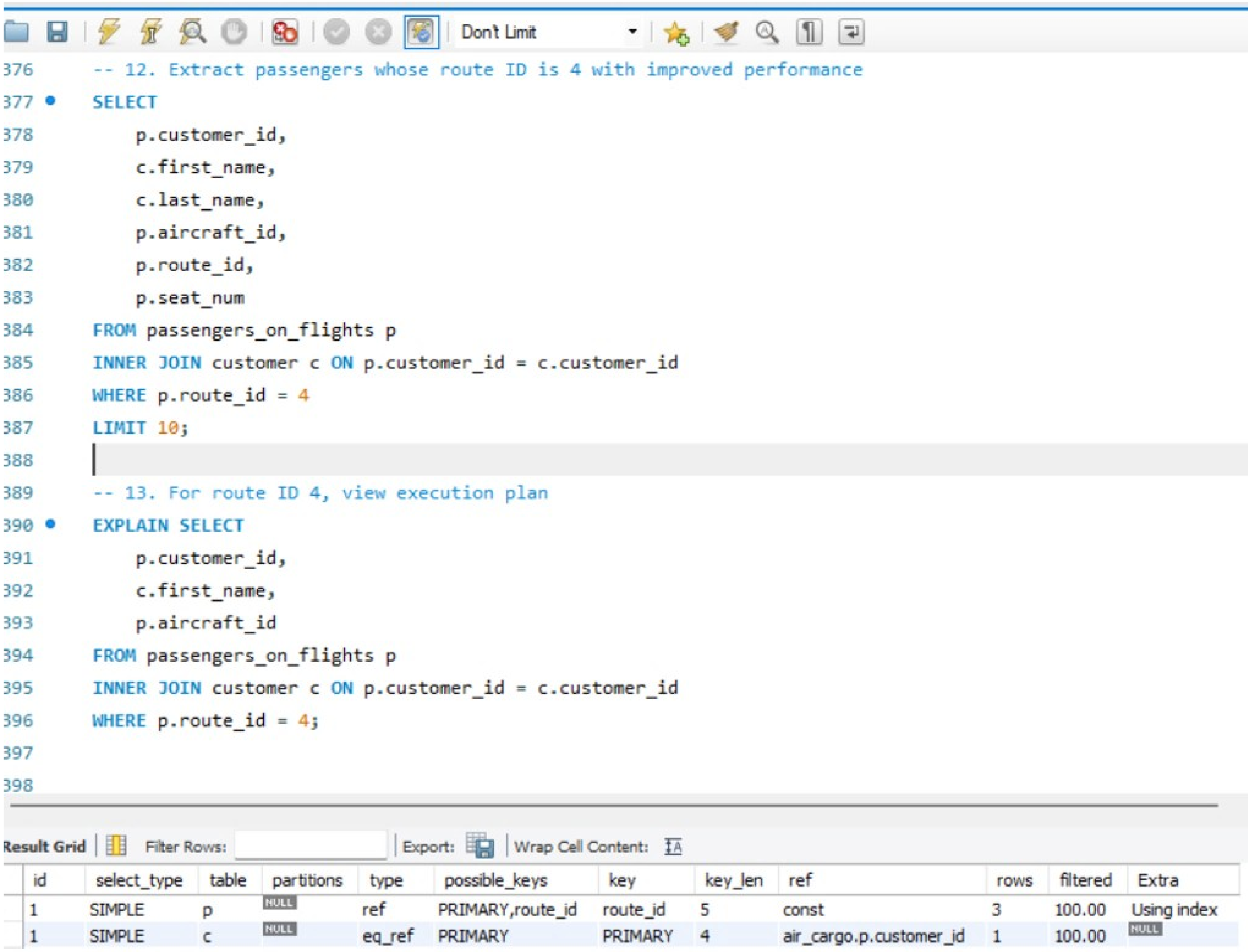
ustomer_id	first_name	last_name	brand	class_id
	Steve	Ryan	Qatar Airways	Business
	Aaron	Kim	Emirates	Business
	Anderson	Stewart	Emirates	Business
1	Roger	Walson	Emirates	Business
5	Linda	William	Qatar Airways	Business
1	Chirsty	Josh	British Airways	Business
4	Calvin	Willis	Qatar Airways	Business
5	Moss	Morris	Emirates	Business
9	Watson	Ronald	Qatar Airways	Business
9	Watson	Ronald	Jet Airways	Business



```sql
    WHERE p.route_id = 4;


-- 14. Calculate total price of all tickets using rollup function
SELECT
    aircraft_id,
    class_id,
    SUM(Price_per_ticket) as total_price
FROM ticket_details
GROUP BY aircraft_id, class_id WITH ROLLUP;
```

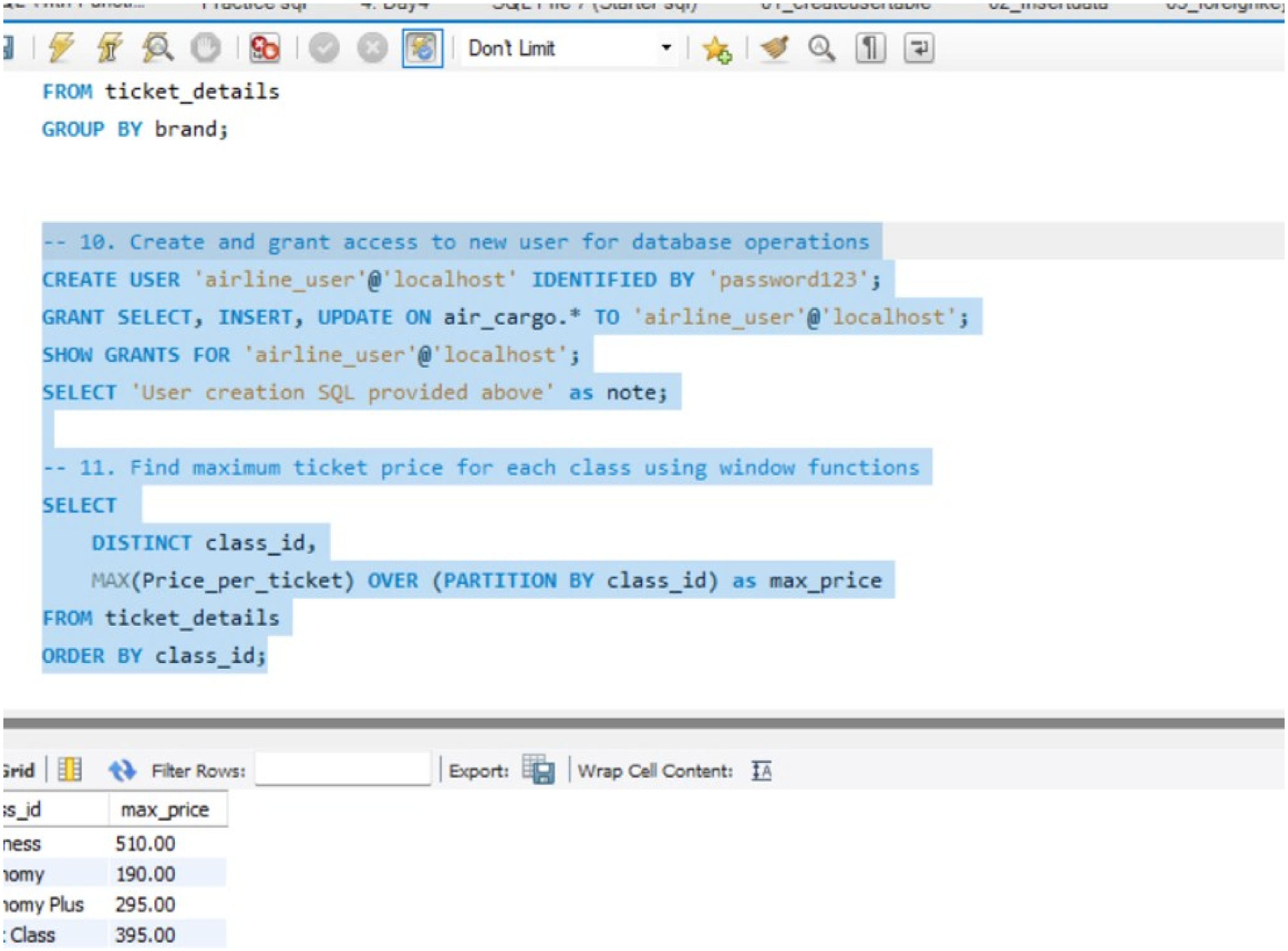
aircraft_id	class_id	total_price
767-301ER	Business	2719.00
767-301ER	Economy	1000.00
767-301ER	First Class	1915.00
767-301ER	NULL	5634.00
A321	Business	1825.00
A321	Economy	460.00
A321	Economy Plus	800.00
A321	First Class	1185.00
A321	NULL	4270.00
CRJ900	Business	980.00
CRJ900	Economy Plus	675.00
CRJ900	First Class	1785.00
CRJ900	NULL	3440.00
ERJ142	Business	510.00
ERJ142	Economy	530.00
ERJ142	Economy Plus	985.00
ERJ142	NULL	2025.00
NULL	NULL	15369.00



```
-- 12. Extract passengers whose route ID is 4 with improved performance
SELECT
    p.customer_id,
    c.first_name,
    c.last_name,
    p.aircraft_id,
    p.route_id,
    p.seat_num
FROM passengers_on_flights p
INNER JOIN customer c ON p.customer_id = c.customer_id
WHERE p.route_id = 4
LIMIT 10;

-- 13. For route ID 4, view execution plan
EXPLAIN SELECT
    p.customer_id,
    c.first_name,
    p.aircraft_id
FROM passengers_on_flights p
INNER JOIN customer c ON p.customer_id = c.customer_id
WHERE p.route_id = 4;
```

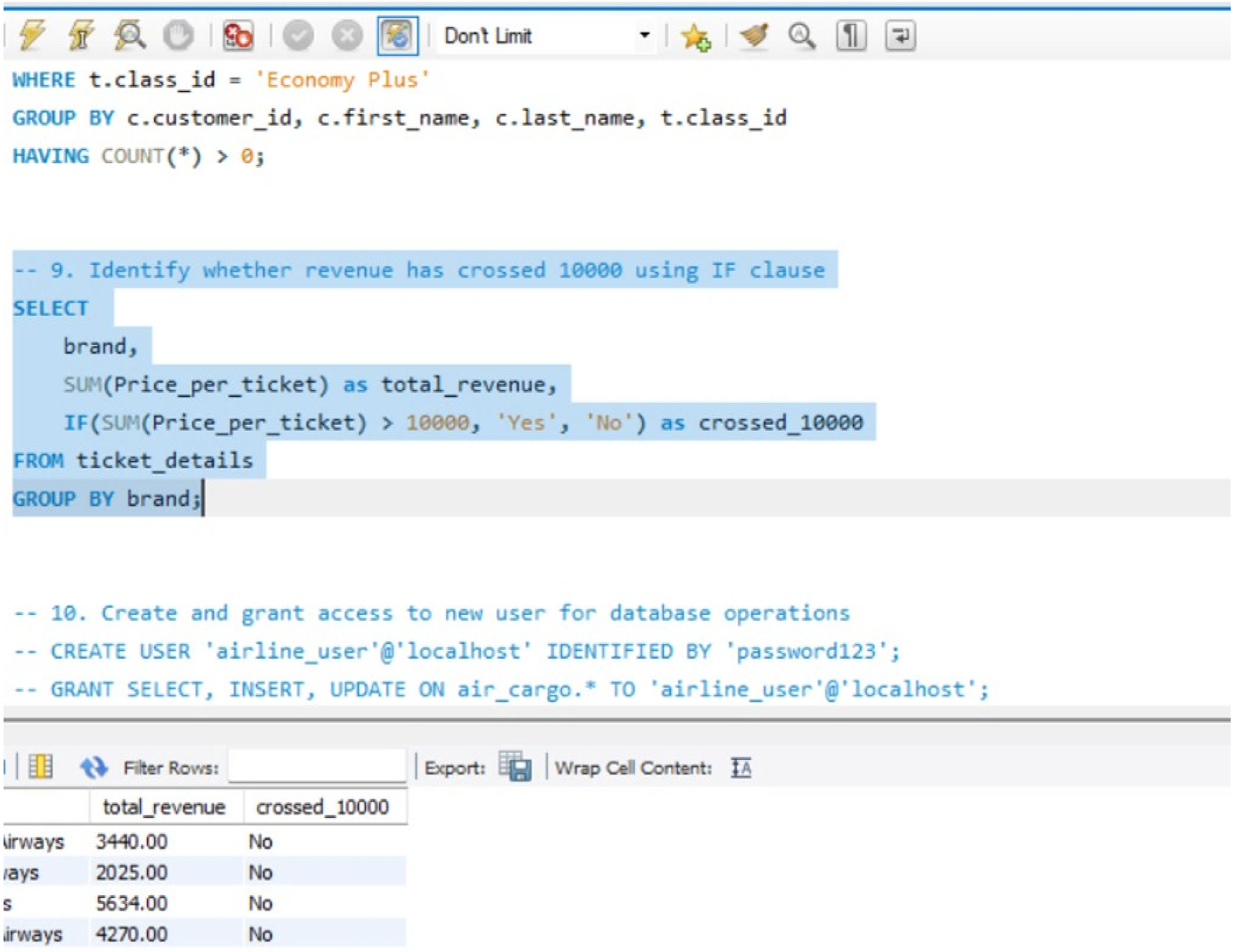
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	p	NULL	ref	PRIMARY,route_id	route_id	5	const	3	100.00	Using index
1	SIMPLE	c	NULL	eq_ref	PRIMARY	PRIMARY	4	air_cargo.p.customer_id	1	100.00	NULL


```
FROM ticket_details
GROUP BY brand;


-- 10. Create and grant access to new user for database operations
CREATE USER 'airline_user'@'localhost' IDENTIFIED BY 'password123';
GRANT SELECT, INSERT, UPDATE ON air_cargo.* TO 'airline_user'@'localhost';
SHOW GRANTS FOR 'airline_user'@'localhost';
SELECT 'User creation SQL provided above' as note;

-- 11. Find maximum ticket price for each class using window functions
SELECT
    DISTINCT class_id,
    MAX(Price_per_ticket) OVER (PARTITION BY class_id) as max_price
FROM ticket_details
ORDER BY class_id;
```

```
ss_id        max_price
ness         510.00
nomy         190.00
nomy Plus    295.00
: Class      395.00
```

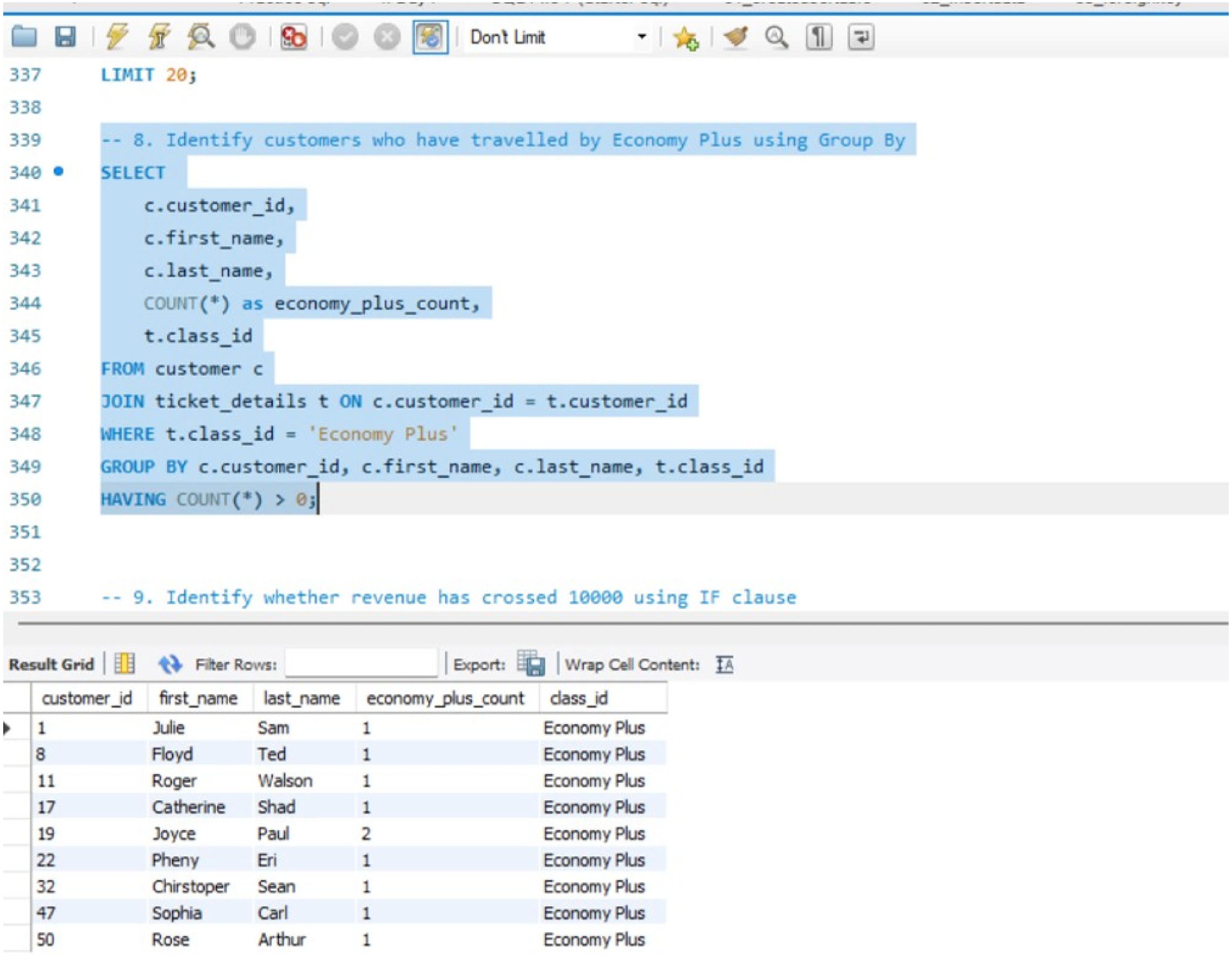


```sql
WHERE t.class_id = 'Economy Plus'
GROUP BY c.customer_id, c.first_name, c.last_name, t.class_id
HAVING COUNT(*) > 0;


-- 9. Identify whether revenue has crossed 10000 using IF clause
SELECT
    brand,
    SUM(Price_per_ticket) as total_revenue,
    IF(SUM(Price_per_ticket) > 10000, 'Yes', 'No') as crossed_10000
FROM ticket_details
GROUP BY brand;


-- 10. Create and grant access to new user for database operations
-- CREATE USER 'airline_user'@'localhost' IDENTIFIED BY 'password123';
-- GRANT SELECT, INSERT, UPDATE ON air_cargo.* TO 'airline_user'@'localhost';
```

```
         total_revenue   crossed_10000
irways   3440.00         No
ays      2025.00         No
s        5634.00         No
irways   4270.00         No
```

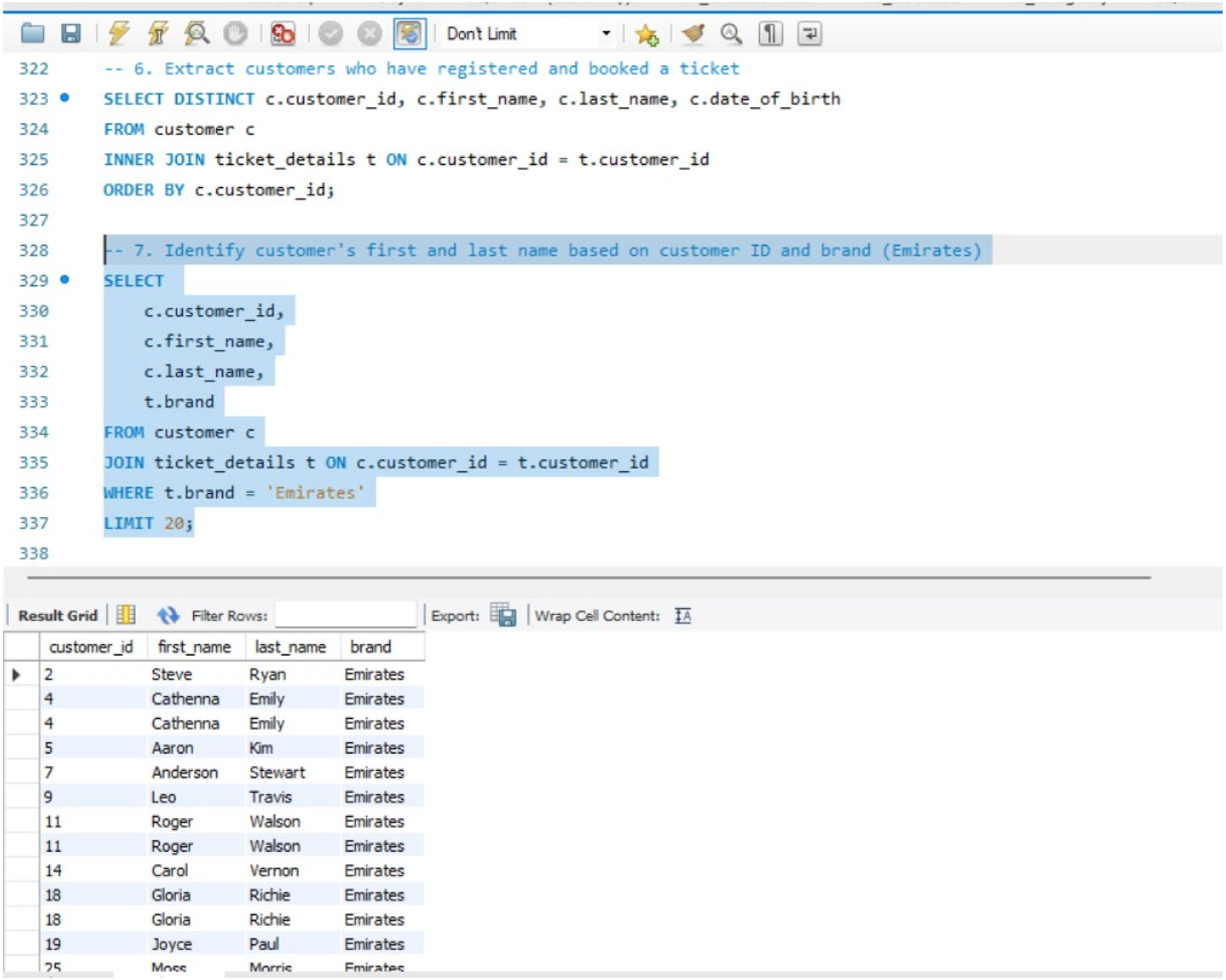


```
LIMIT 20;

-- 8. Identify customers who have travelled by Economy Plus using Group By
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    COUNT(*) as economy_plus_count,
    t.class_id
FROM customer c
JOIN ticket_details t ON c.customer_id = t.customer_id
WHERE t.class_id = 'Economy Plus'
GROUP BY c.customer_id, c.first_name, c.last_name, t.class_id
HAVING COUNT(*) > 0;


-- 9. Identify whether revenue has crossed 10000 using IF clause
```

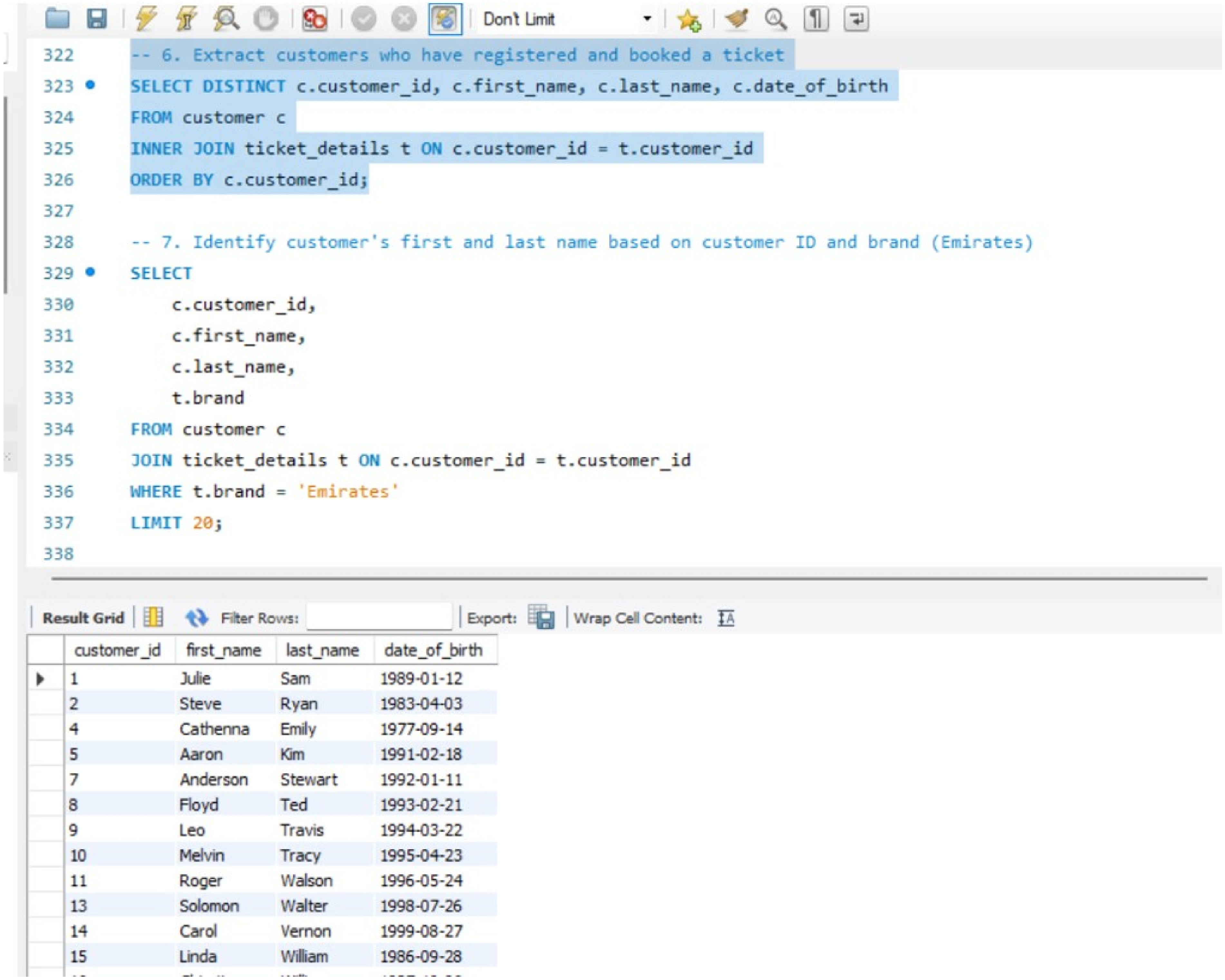
customer_id	first_name	last_name	economy_plus_count	class_id
1	Julie	Sam	1	Economy Plus
8	Floyd	Ted	1	Economy Plus
11	Roger	Walson	1	Economy Plus
17	Catherine	Shad	1	Economy Plus
19	Joyce	Paul	2	Economy Plus
22	Pheny	Eri	1	Economy Plus
32	Chirstoper	Sean	1	Economy Plus
47	Sophia	Carl	1	Economy Plus
50	Rose	Arthur	1	Economy Plus



```sql
-- 6. Extract customers who have registered and booked a ticket
SELECT DISTINCT c.customer_id, c.first_name, c.last_name, c.date_of_birth
FROM customer c
INNER JOIN ticket_details t ON c.customer_id = t.customer_id
ORDER BY c.customer_id;

-- 7. Identify customer's first and last name based on customer ID and brand (Emirates)
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    t.brand
FROM customer c
JOIN ticket_details t ON c.customer_id = t.customer_id
WHERE t.brand = 'Emirates'
LIMIT 20;
```

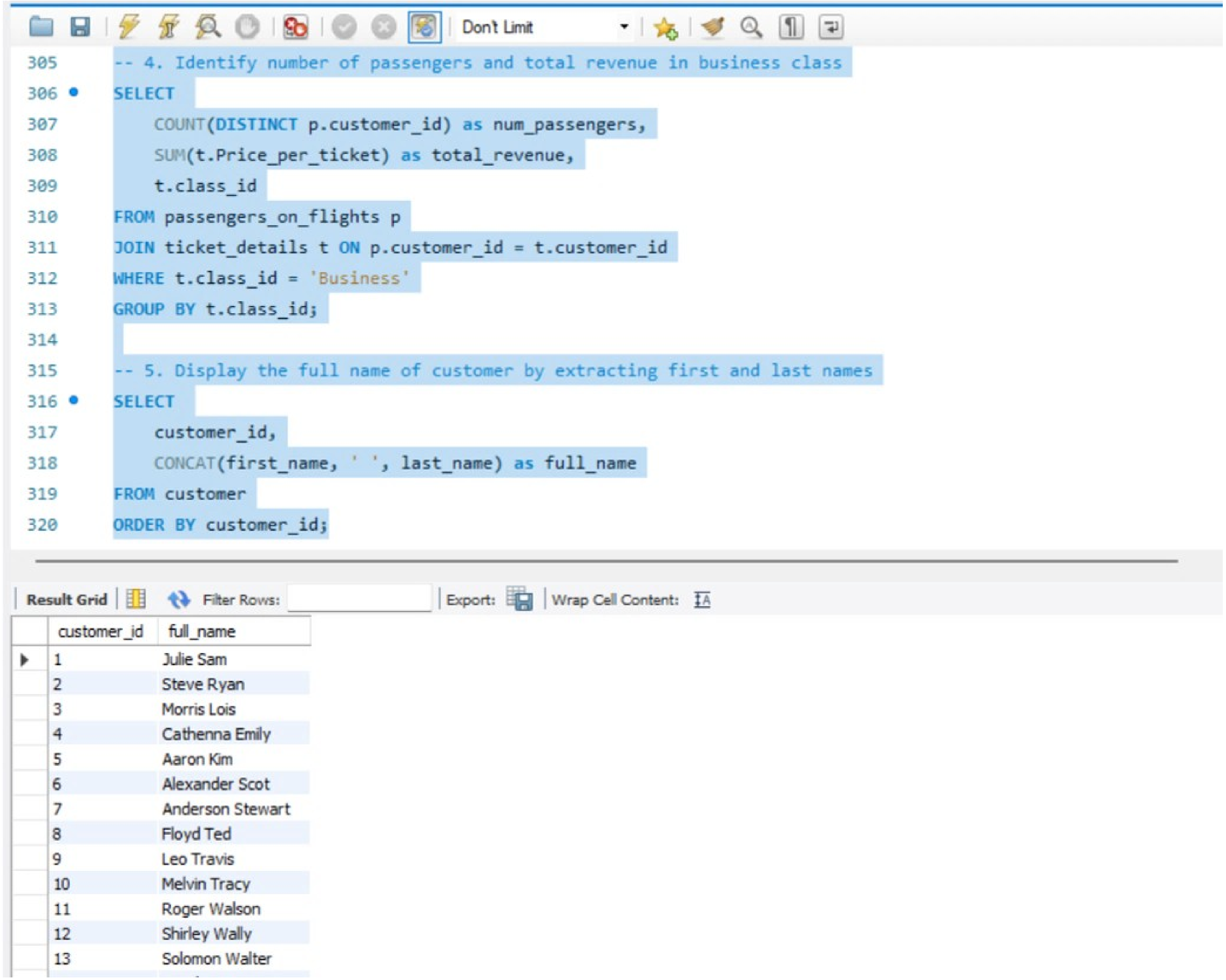
customer_id	first_name	last_name	brand
2	Steve	Ryan	Emirates
4	Cathenna	Emily	Emirates
4	Cathenna	Emily	Emirates
5	Aaron	Kim	Emirates
7	Anderson	Stewart	Emirates
9	Leo	Travis	Emirates
11	Roger	Walson	Emirates
11	Roger	Walson	Emirates
14	Carol	Vernon	Emirates
18	Gloria	Richie	Emirates
18	Gloria	Richie	Emirates
19	Joyce	Paul	Emirates
25	Moss	Morris	Emirates



```sql
-- 6. Extract customers who have registered and booked a ticket
SELECT DISTINCT c.customer_id, c.first_name, c.last_name, c.date_of_birth
FROM customer c
INNER JOIN ticket_details t ON c.customer_id = t.customer_id
ORDER BY c.customer_id;

-- 7. Identify customer's first and last name based on customer ID and brand (Emirates)
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    t.brand
FROM customer c
JOIN ticket_details t ON c.customer_id = t.customer_id
WHERE t.brand = 'Emirates'
LIMIT 20;
```

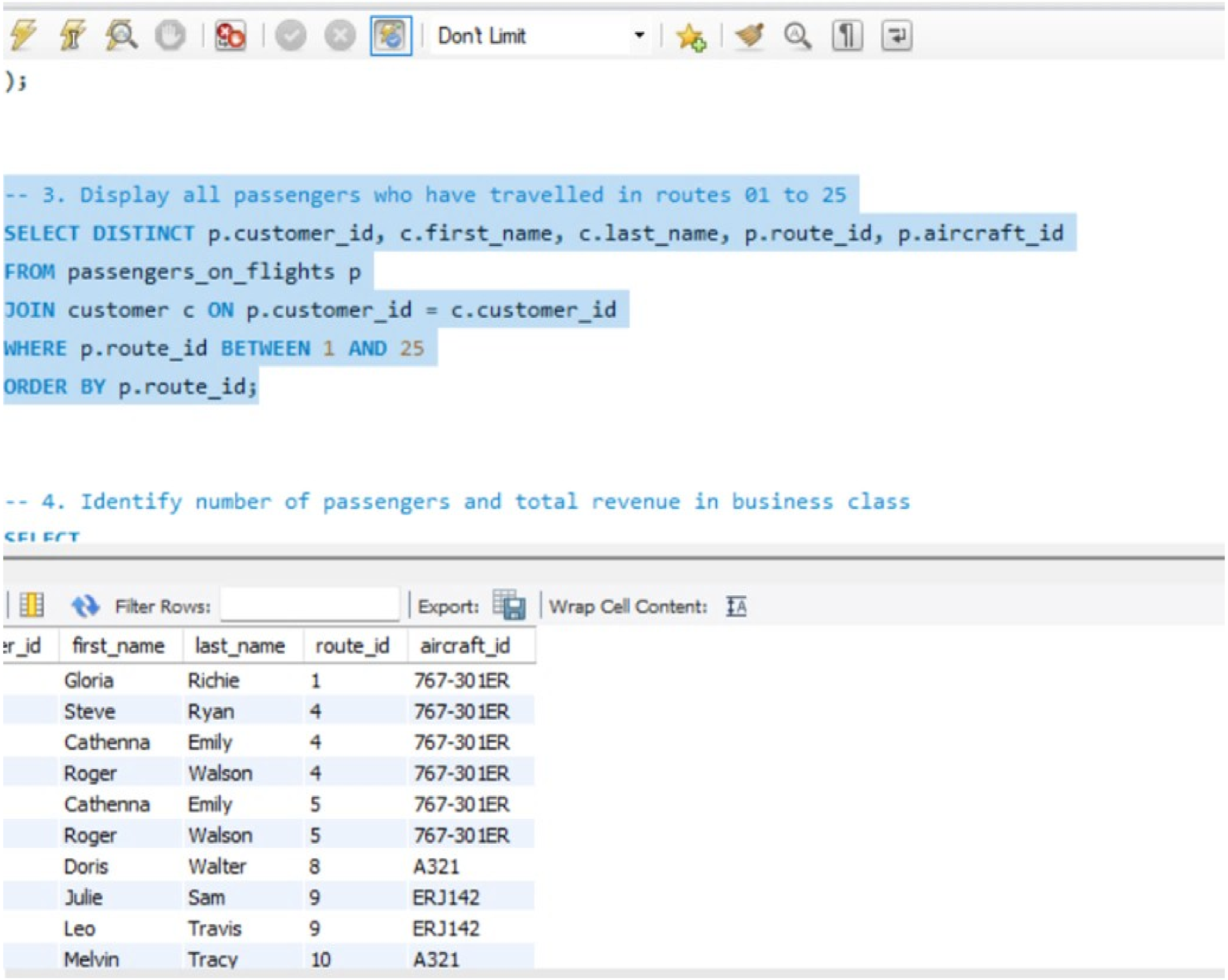
customer_id	first_name	last_name	date_of_birth
1	Julie	Sam	1989-01-12
2	Steve	Ryan	1983-04-03
4	Cathenna	Emily	1977-09-14
5	Aaron	Kim	1991-02-18
7	Anderson	Stewart	1992-01-11
8	Floyd	Ted	1993-02-21
9	Leo	Travis	1994-03-22
10	Melvin	Tracy	1995-04-23
11	Roger	Walson	1996-05-24
13	Solomon	Walter	1998-07-26
14	Carol	Vernon	1999-08-27
15	Linda	William	1986-09-28



```sql
-- 4. Identify number of passengers and total revenue in business class
SELECT
    COUNT(DISTINCT p.customer_id) as num_passengers,
    SUM(t.Price_per_ticket) as total_revenue,
    t.class_id
FROM passengers_on_flights p
JOIN ticket_details t ON p.customer_id = t.customer_id
WHERE t.class_id = 'Business'
GROUP BY t.class_id;

-- 5. Display the full name of customer by extracting first and last names
SELECT
    customer_id,
    CONCAT(first_name, ' ', last_name) as full_name
FROM customer
ORDER BY customer_id;
```

customer_id	full_name
1	Julie Sam
2	Steve Ryan
3	Morris Lois
4	Cathenna Emily
5	Aaron Kim
6	Alexander Scot
7	Anderson Stewart
8	Floyd Ted
9	Leo Travis
10	Melvin Tracy
11	Roger Walson
12	Shirley Wally
13	Solomon Walter



```
);


-- 3. Display all passengers who have travelled in routes 01 to 25
SELECT DISTINCT p.customer_id, c.first_name, c.last_name, p.route_id, p.aircraft_id
FROM passengers_on_flights p
JOIN customer c ON p.customer_id = c.customer_id
WHERE p.route_id BETWEEN 1 AND 25
ORDER BY p.route_id;


-- 4. Identify number of passengers and total revenue in business class
SELECT
```

r_id	first_name	last_name	route_id	aircraft_id
	Gloria	Richie	1	767-301ER
	Steve	Ryan	4	767-301ER
	Cathenna	Emily	4	767-301ER
	Roger	Walson	4	767-301ER
	Cathenna	Emily	5	767-301ER
	Roger	Walson	5	767-301ER
	Doris	Walter	8	A321
	Julie	Sam	9	ERJ142
	Leo	Travis	9	ERJ142
	Melvin	Tracy	10	A321