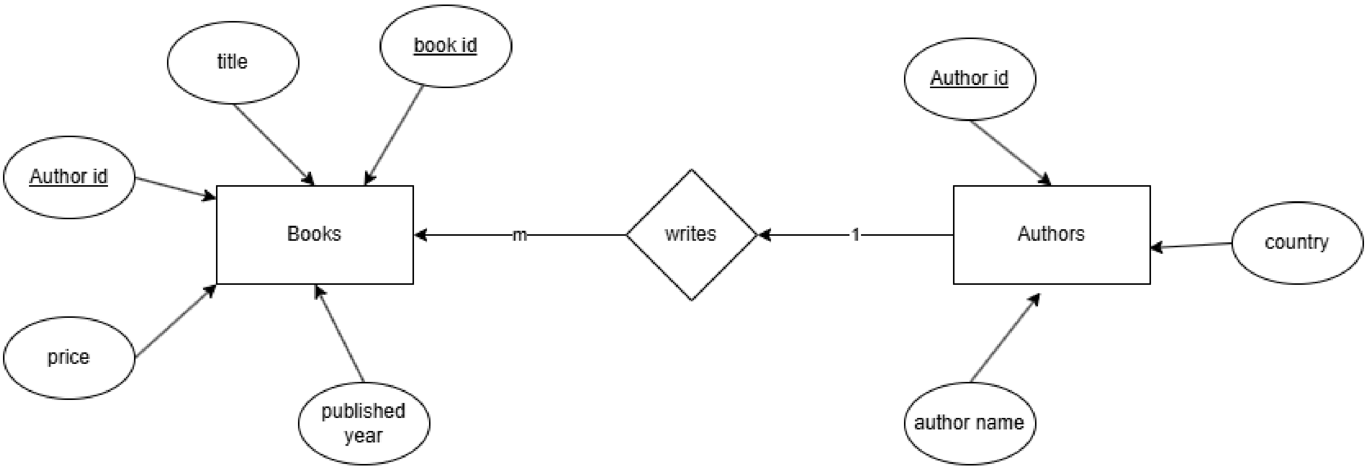
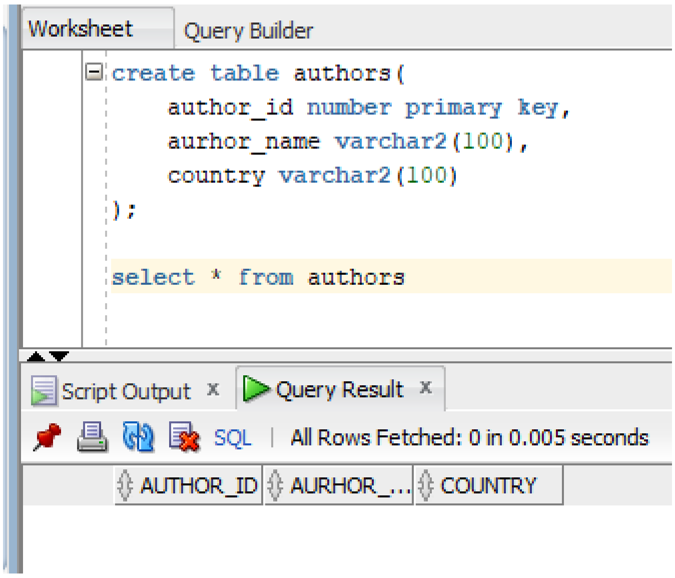
DBMS LAB 2 EXERCISES

Siddharth Karmokar, 123cs0061

Exercise 1:1: ER_DIAGRAM



Excercise 1:2: AUTHORS

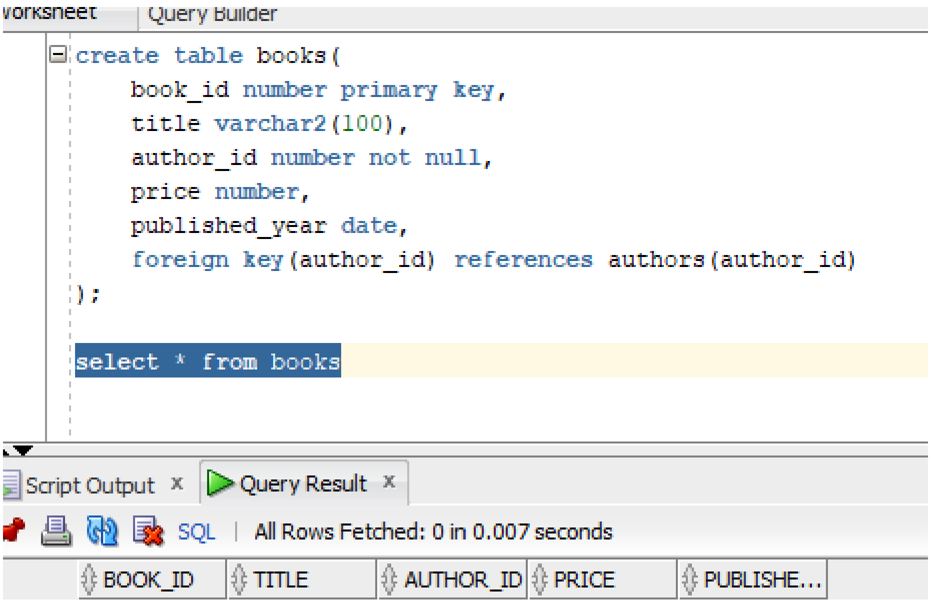


```
create table authors(
    author_id number primary key,
    aurhor_name varchar2(100),
    country varchar2(100)
);

select * from authors
```

All Rows Fetched: 0 in 0.005 seconds

```
AUTHOR_ID  AURHOR_...  COUNTRY
```

Excercise 1:3: BOOKS

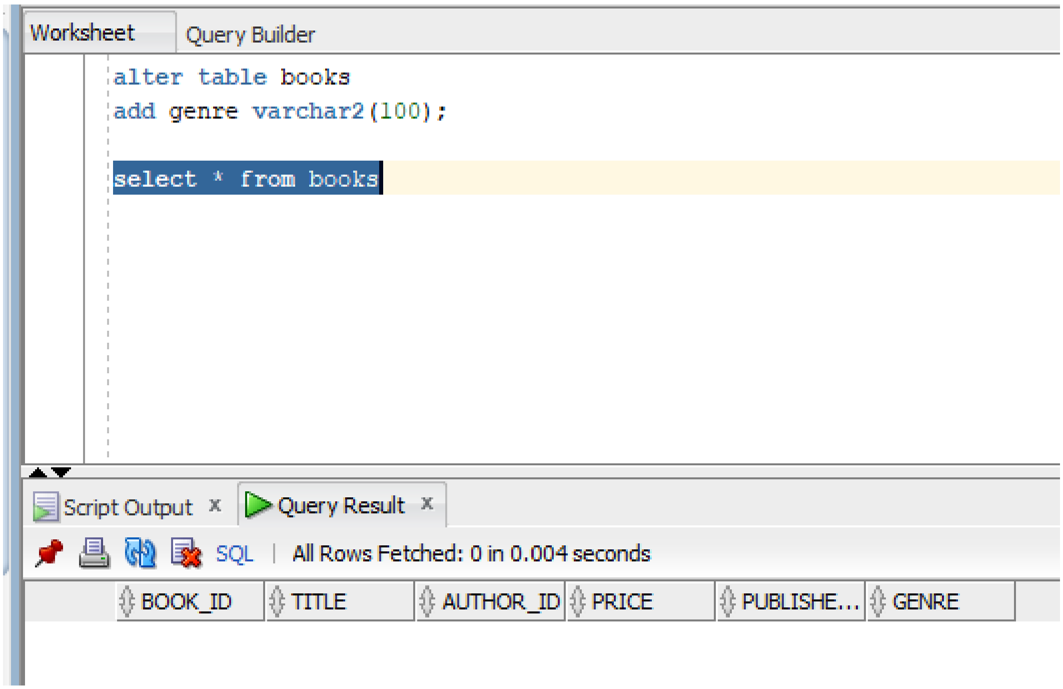


```
create table books(
    book_id number primary key,
    title varchar2(100),
    author_id number not null,
    price number,
    published_year date,
    foreign key(author_id) references authors(author_id)
);

select * from books
```

All Rows Fetched: 0 in 0.007 seconds

BOOK_ID	TITLE	AUTHOR_ID	PRICE	PUBLISHE...

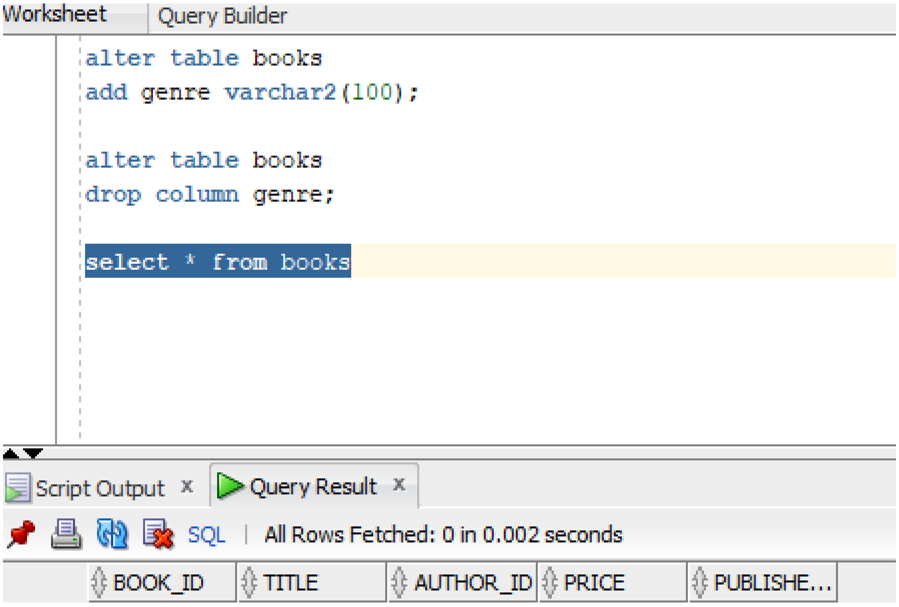
Excercise 1:4: ADD AND DROP GENRE FROM BOOKS



```
alter table books
add genre varchar2(100);

select * from books
```

All Rows Fetched: 0 in 0.004 seconds

BOOK_ID	TITLE	AUTHOR_ID	PRICE	PUBLISHE...	GENRE

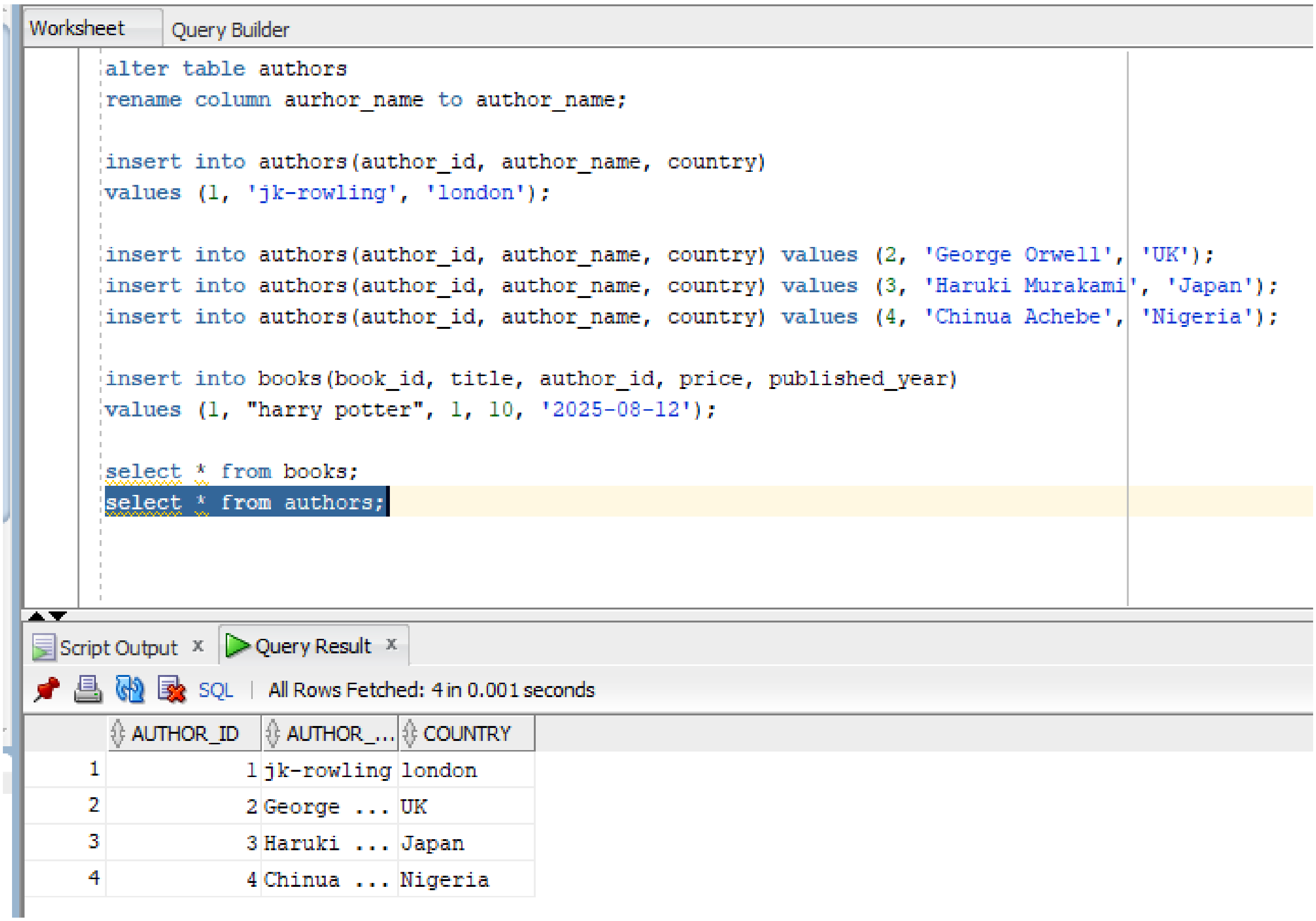


```
alter table books
add genre varchar2(100);

alter table books
drop column genre;

select * from books
```

All Rows Fetched: 0 in 0.002 seconds

BOOK_ID	TITLE	AUTHOR_ID	PRICE	PUBLISHE...

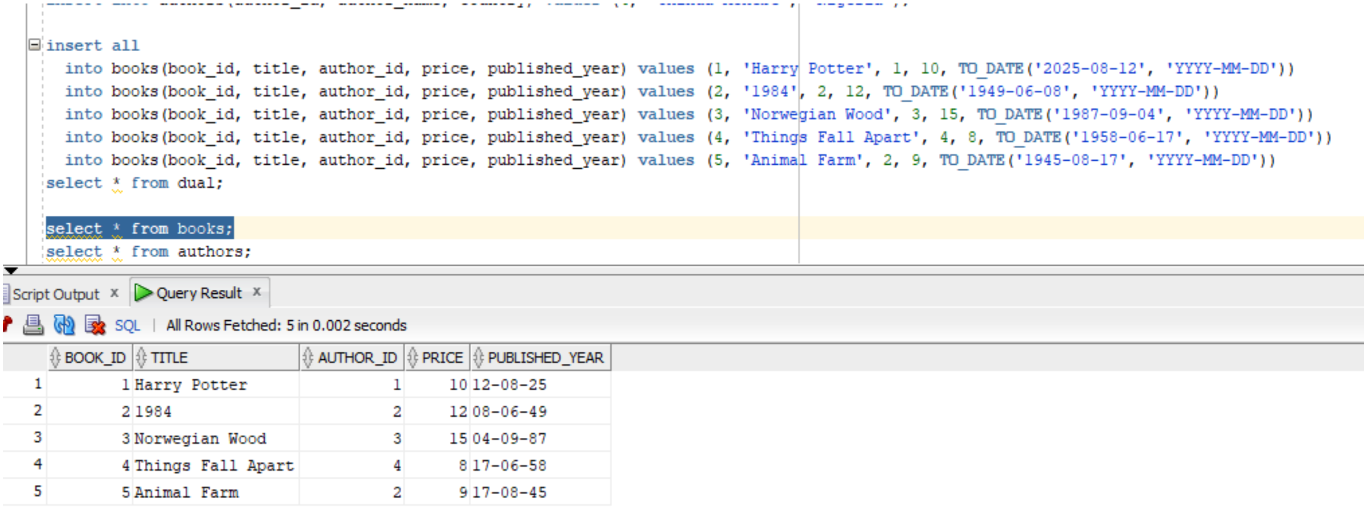
Excercise 1:5: INSERT 5 RECORDS INTO AUTHORS AND BOOKS



```
alter table authors
rename column aurhor_name to author_name;

insert into authors(author_id, author_name, country)
values (1, 'jk-rowling', 'london');

insert into authors(author_id, author_name, country) values (2, 'George Orwell', 'UK');
insert into authors(author_id, author_name, country) values (3, 'Haruki Murakami', 'Japan');
insert into authors(author_id, author_name, country) values (4, 'Chinua Achebe', 'Nigeria');

insert into books(book_id, title, author_id, price, published_year)
values (1, "harry potter", 1, 10, '2025-08-12');

select * from books;
select * from authors;
```

All Rows Fetched: 4 in 0.001 seconds

	AUTHOR_ID	AUTHOR_...	COUNTRY
1	1	jk-rowling	london
2	2	George ...	UK
3	3	Haruki ...	Japan
4	4	Chinua ...	Nigeria

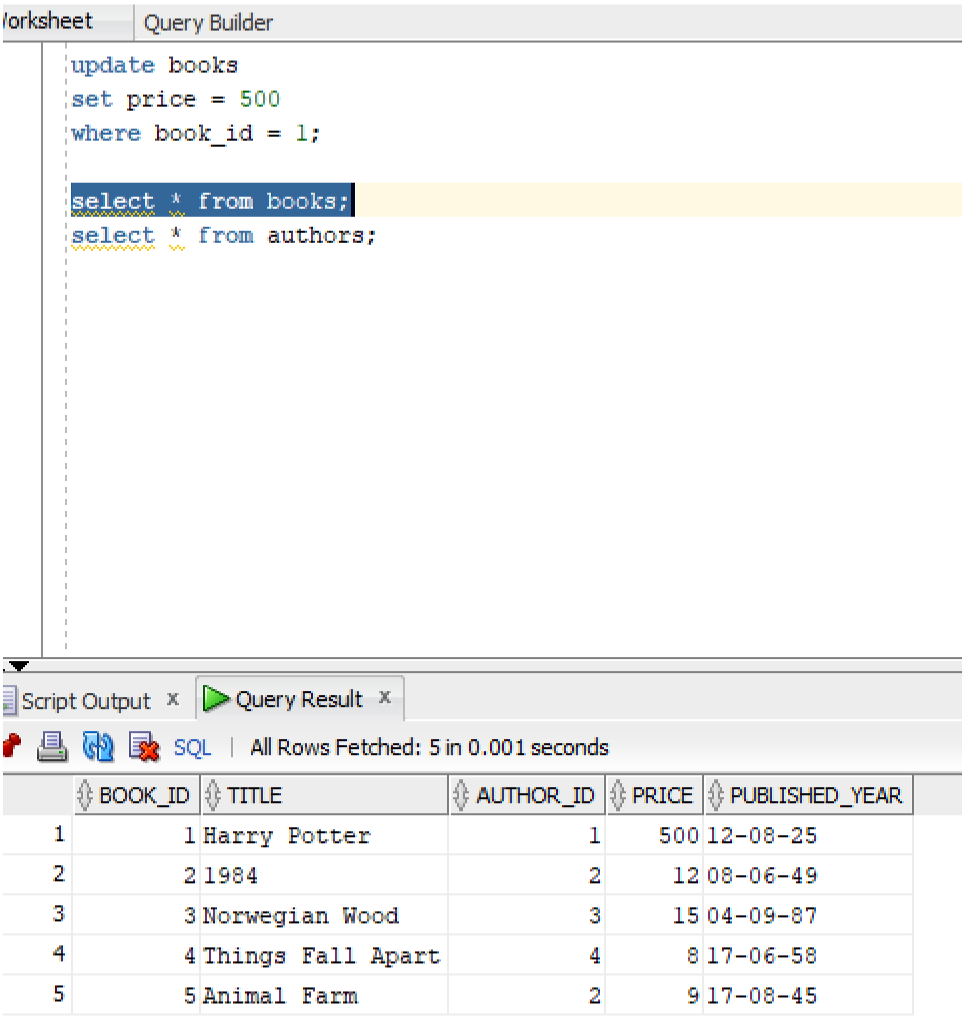


```
insert all
  into books(book_id, title, author_id, price, published_year) values (1, 'Harry Potter', 1, 10, TO_DATE('2025-08-12', 'YYYY-MM-DD'))
  into books(book_id, title, author_id, price, published_year) values (2, '1984', 2, 12, TO_DATE('1949-06-08', 'YYYY-MM-DD'))
  into books(book_id, title, author_id, price, published_year) values (3, 'Norwegian Wood', 3, 15, TO_DATE('1987-09-04', 'YYYY-MM-DD'))
  into books(book_id, title, author_id, price, published_year) values (4, 'Things Fall Apart', 4, 8, TO_DATE('1958-06-17', 'YYYY-MM-DD'))
  into books(book_id, title, author_id, price, published_year) values (5, 'Animal Farm', 2, 9, TO_DATE('1945-08-17', 'YYYY-MM-DD'))
select * from dual;

select * from books;
select * from authors;
```

All Rows Fetched: 5 in 0.002 seconds

	BOOK_ID	TITLE	AUTHOR_ID	PRICE	PUBLISHED_YEAR
1	1	Harry Potter	1	10	12-08-25
2	2	1984	2	12	08-06-49
3	3	Norwegian Wood	3	15	04-09-87
4	4	Things Fall Apart	4	8	17-06-58
5	5	Animal Farm	2	9	17-08-45

Exercise 1:6: UPDATE PRICE OF ONE BOOK

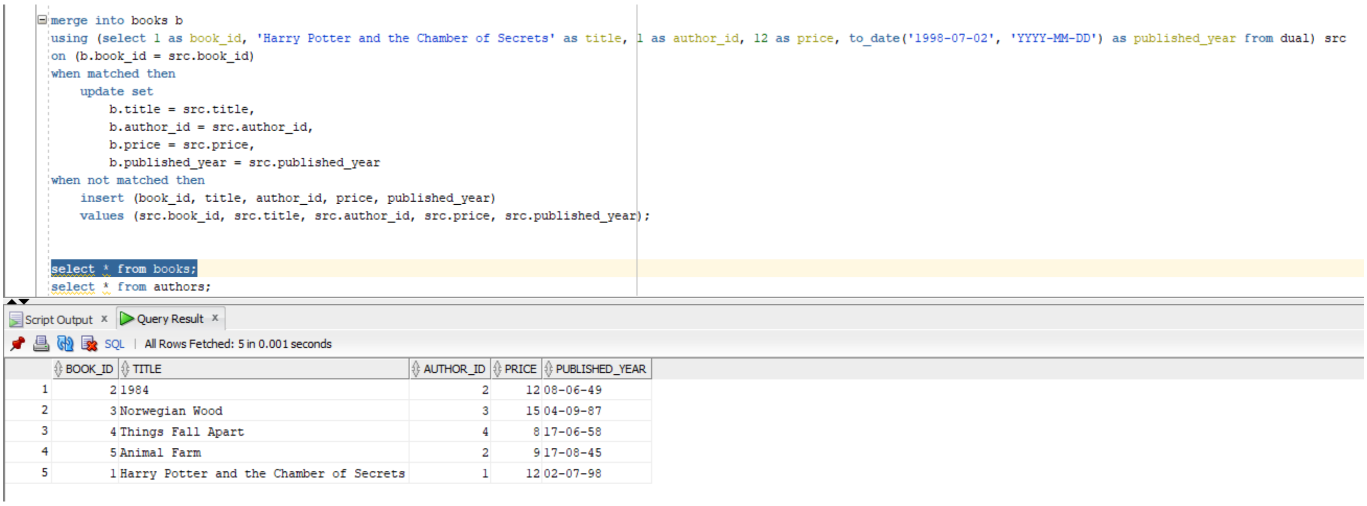


```
update books
set price = 500
where book_id = 1;

select * from books;
select * from authors;
```

All Rows Fetched: 5 in 0.001 seconds

	BOOK_ID	TITLE	AUTHOR_ID	PRICE	PUBLISHED_YEAR
1	1	Harry Potter	1	500	12-08-25
2	2	1984	2	12	08-06-49
3	3	Norwegian Wood	3	15	04-09-87
4	4	Things Fall Apart	4	8	17-06-58
5	5	Animal Farm	2	9	17-08-45

Exercise 1:7: MERGE COMMAND TO UPDATE ONE BOOK RECORD

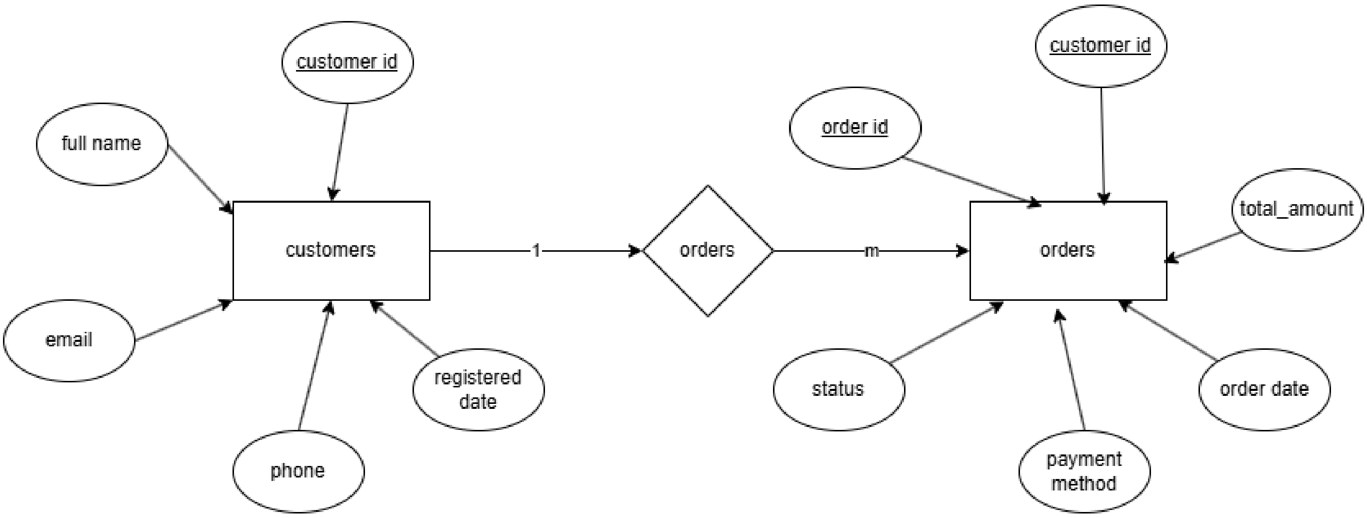


```
merge into books b
using (select 1 as book_id, 'Harry Potter and the Chamber of Secrets' as title, 1 as author_id, 12 as price, to_date('1998-07-02', 'YYYY-MM-DD') as published_year from dual) src
on (b.book_id = src.book_id)
when matched then
    update set
        b.title = src.title,
        b.author_id = src.author_id,
        b.price = src.price,
        b.published_year = src.published_year
when not matched then
    insert (book_id, title, author_id, price, published_year)
    values (src.book_id, src.title, src.author_id, src.price, src.published_year);


select * from books;
select * from authors;
```

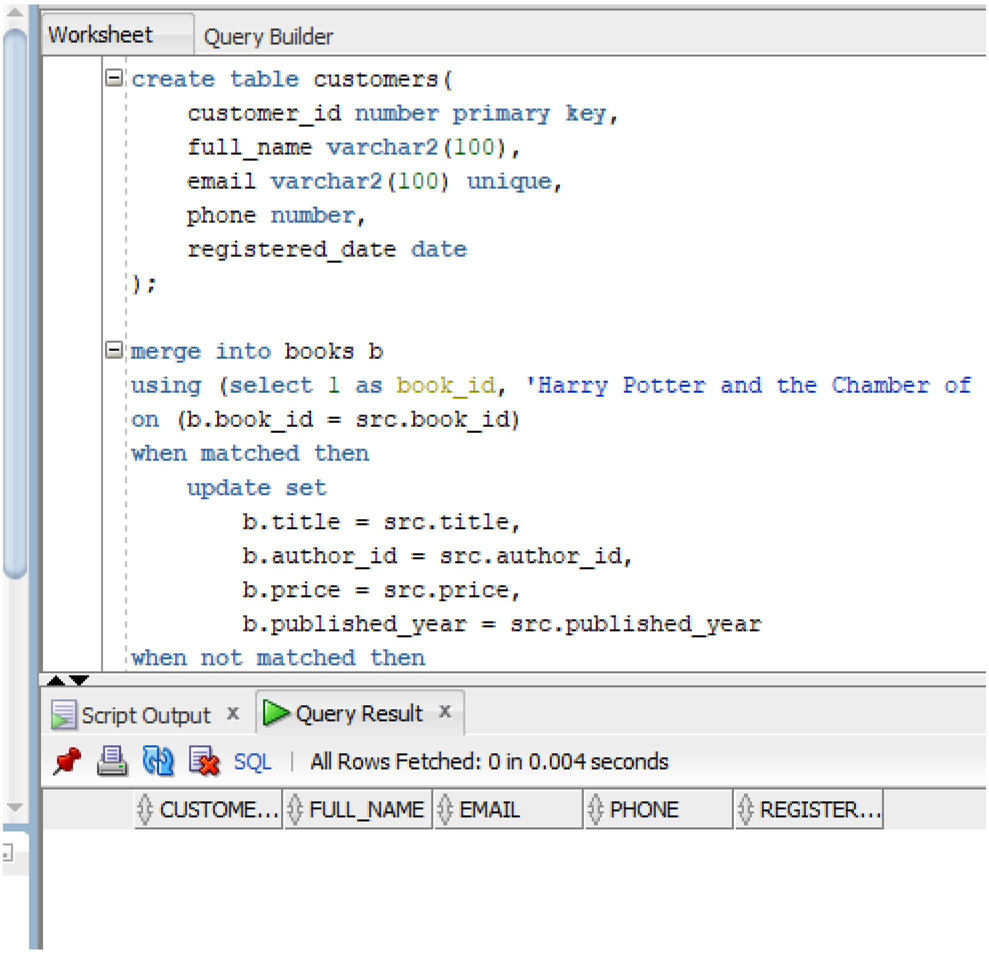
All Rows Fetched: 5 in 0.001 seconds

	BOOK_ID	TITLE	AUTHOR_ID	PRICE	PUBLISHED_YEAR
1	2	1984	2	12	08-06-49
2	3	Norwegian Wood	3	15	04-09-87
3	4	Things Fall Apart	4	8	17-06-58
4	5	Animal Farm	2	9	17-08-45
5	1	Harry Potter and the Chamber of Secrets	1	12	02-07-98

Exercise 2:1: ER_DIAGRAM



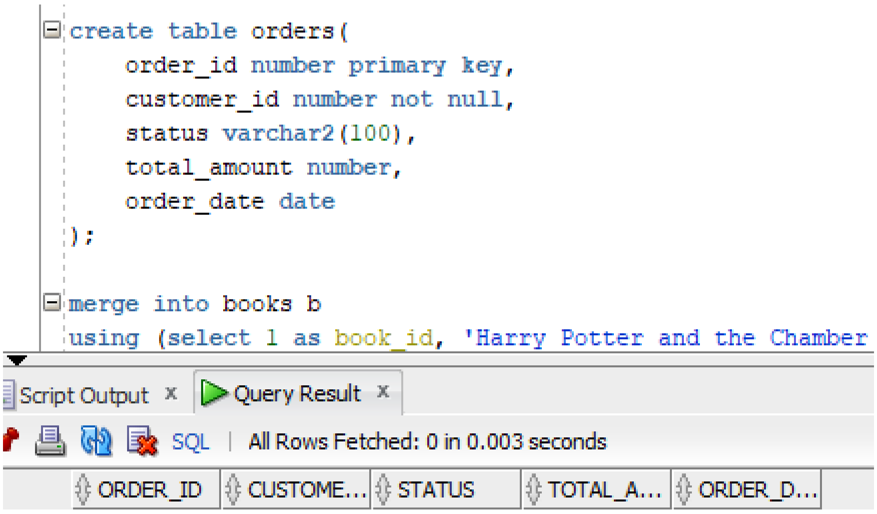
Excercise 2:2: CUSTOMERS



```
create table customers(
    customer_id number primary key,
    full_name varchar2(100),
    email varchar2(100) unique,
    phone number,
    registered_date date
);

merge into books b
using (select 1 as book_id, 'Harry Potter and the Chamber of
on (b.book_id = src.book_id)
when matched then
    update set
        b.title = src.title,
        b.author_id = src.author_id,
        b.price = src.price,
        b.published_year = src.published_year
when not matched then
```

```
All Rows Fetched: 0 in 0.004 seconds

CUSTOME...  FULL_NAME  EMAIL  PHONE  REGISTER...
```

Excercise 2:3: ORDERS

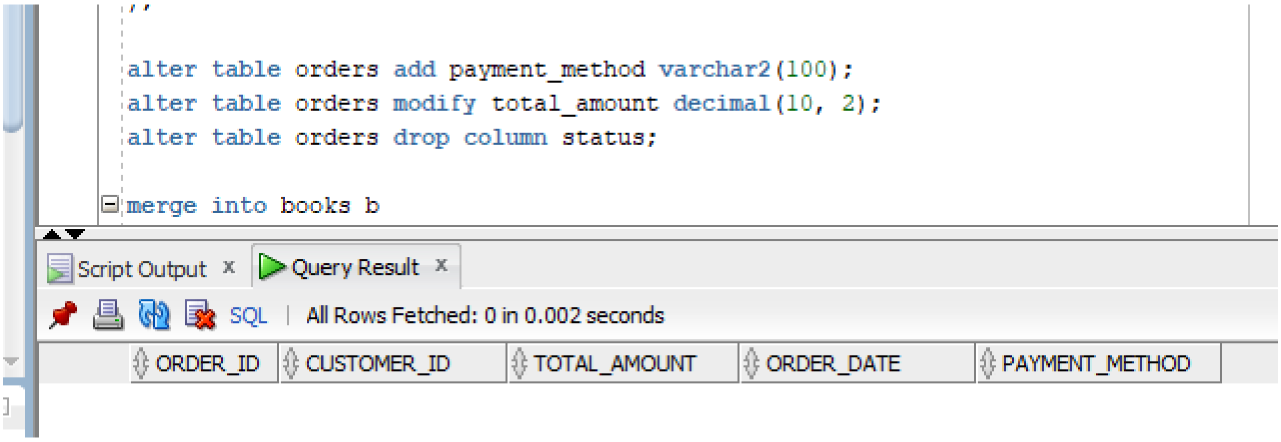


```
create table orders(
    order_id number primary key,
    customer_id number not null,
    status varchar2(100),
    total_amount number,
    order_date date
);

merge into books b
using (select 1 as book_id, 'Harry Potter and the Chamber
```

All Rows Fetched: 0 in 0.003 seconds

ORDER_ID	CUSTOME...	STATUS	TOTAL_A...	ORDER_D...

Excercise 2:4: ADD PAYMENT_METHOD TO ORDERS, MODIFY PERCISION OF TOTAL_AMOUNT, DROP STATUS

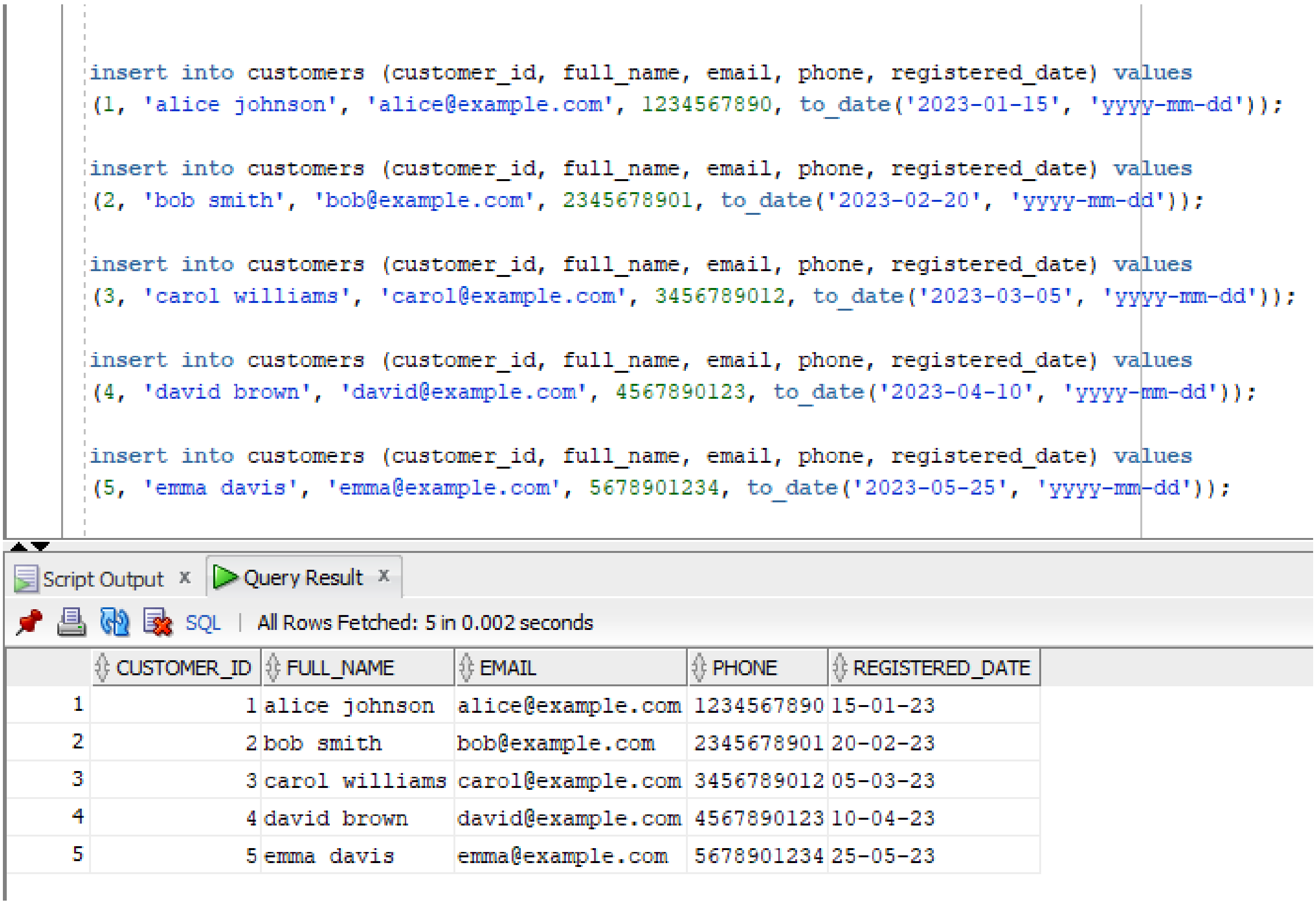


```
alter table orders add payment_method varchar2(100);
alter table orders modify total_amount decimal(10, 2);
alter table orders drop column status;

merge into books b
```

All Rows Fetched: 0 in 0.002 seconds

ORDER_ID	CUSTOMER_ID	TOTAL_AMOUNT	ORDER_DATE	PAYMENT_METHOD

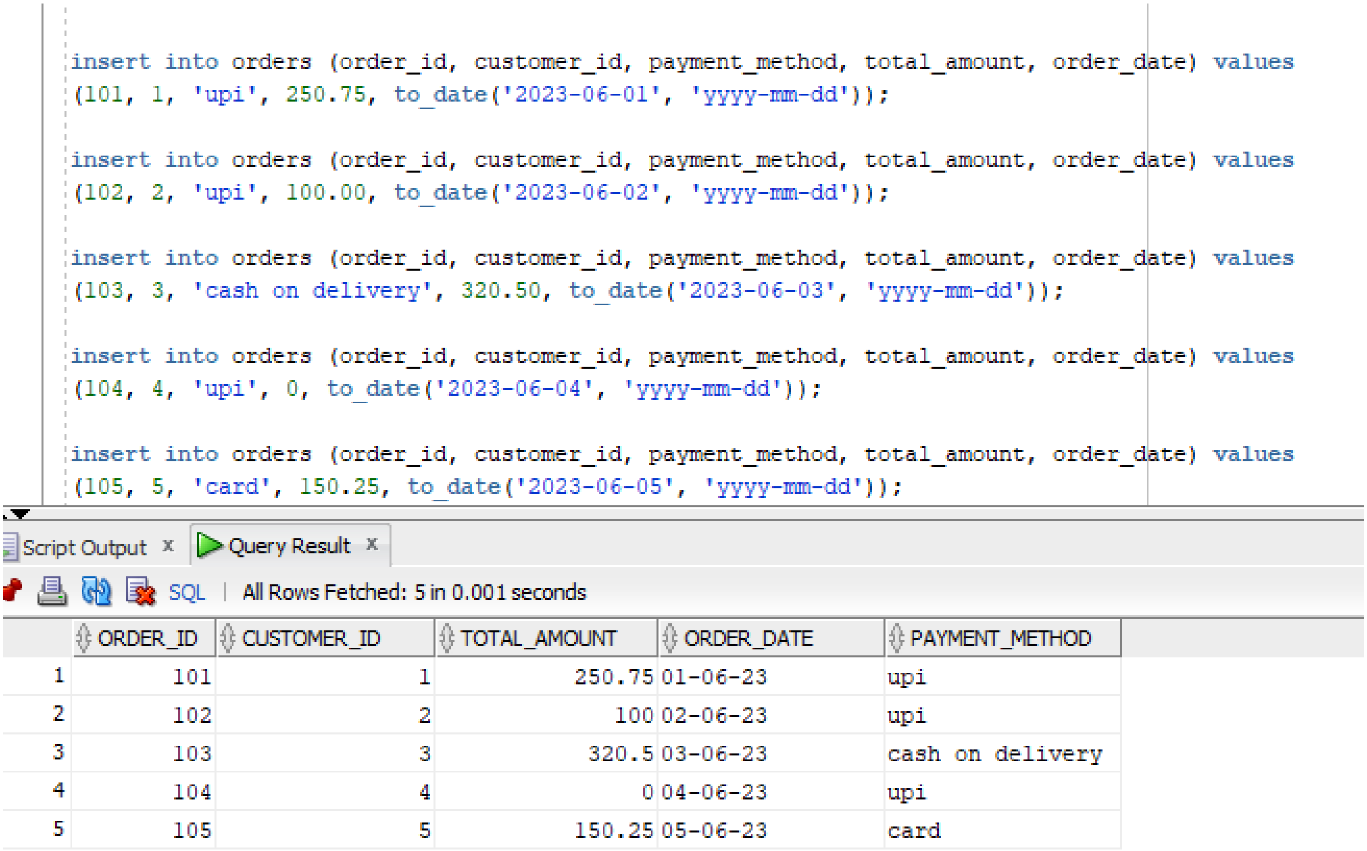
Excercise 2:5: INSERT 5 CUSTOMERS AND LINKED ORDERS



```
insert into customers (customer_id, full_name, email, phone, registered_date) values
(1, 'alice johnson', 'alice@example.com', 1234567890, to_date('2023-01-15', 'yyyy-mm-dd'));

insert into customers (customer_id, full_name, email, phone, registered_date) values
(2, 'bob smith', 'bob@example.com', 2345678901, to_date('2023-02-20', 'yyyy-mm-dd'));

insert into customers (customer_id, full_name, email, phone, registered_date) values
(3, 'carol williams', 'carol@example.com', 3456789012, to_date('2023-03-05', 'yyyy-mm-dd'));

insert into customers (customer_id, full_name, email, phone, registered_date) values
(4, 'david brown', 'david@example.com', 4567890123, to_date('2023-04-10', 'yyyy-mm-dd'));

insert into customers (customer_id, full_name, email, phone, registered_date) values
(5, 'emma davis', 'emma@example.com', 5678901234, to_date('2023-05-25', 'yyyy-mm-dd'));
```

All Rows Fetched: 5 in 0.002 seconds

	CUSTOMER_ID	FULL_NAME	EMAIL	PHONE	REGISTERED_DATE
1	1	alice johnson	alice@example.com	1234567890	15-01-23
2	2	bob smith	bob@example.com	2345678901	20-02-23
3	3	carol williams	carol@example.com	3456789012	05-03-23
4	4	david brown	david@example.com	4567890123	10-04-23
5	5	emma davis	emma@example.com	5678901234	25-05-23

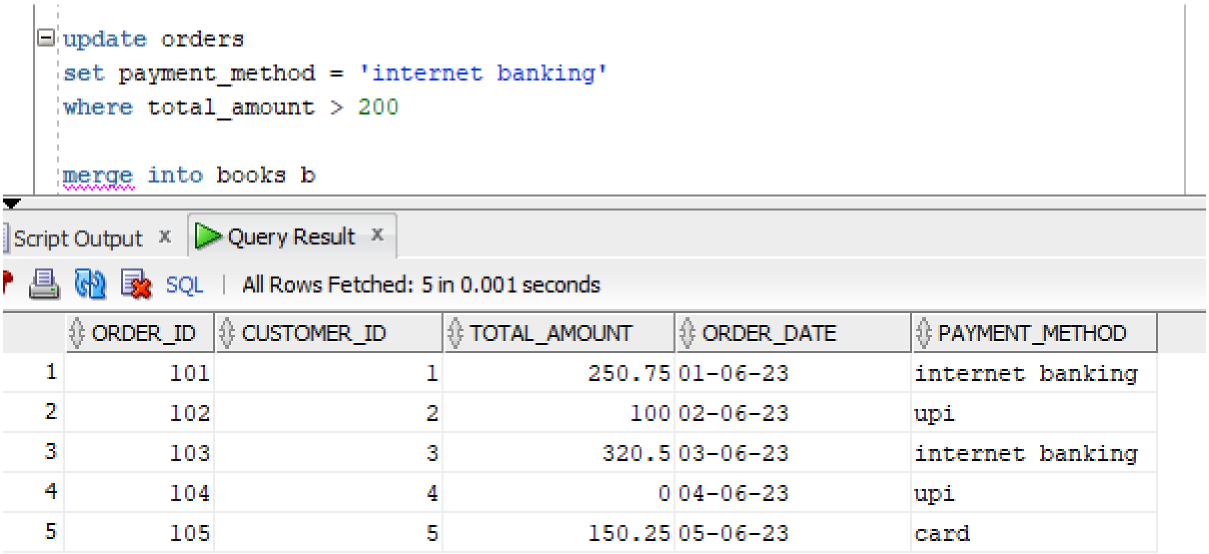


```
insert into orders (order_id, customer_id, payment_method, total_amount, order_date) values
(101, 1, 'upi', 250.75, to_date('2023-06-01', 'yyyy-mm-dd'));

insert into orders (order_id, customer_id, payment_method, total_amount, order_date) values
(102, 2, 'upi', 100.00, to_date('2023-06-02', 'yyyy-mm-dd'));

insert into orders (order_id, customer_id, payment_method, total_amount, order_date) values
(103, 3, 'cash on delivery', 320.50, to_date('2023-06-03', 'yyyy-mm-dd'));

insert into orders (order_id, customer_id, payment_method, total_amount, order_date) values
(104, 4, 'upi', 0, to_date('2023-06-04', 'yyyy-mm-dd'));

insert into orders (order_id, customer_id, payment_method, total_amount, order_date) values
(105, 5, 'card', 150.25, to_date('2023-06-05', 'yyyy-mm-dd'));
```

All Rows Fetched: 5 in 0.001 seconds

	ORDER_ID	CUSTOMER_ID	TOTAL_AMOUNT	ORDER_DATE	PAYMENT_METHOD
1	101	1	250.75	01-06-23	upi
2	102	2	100	02-06-23	upi
3	103	3	320.5	03-06-23	cash on delivery
4	104	4	0	04-06-23	upi
5	105	5	150.25	05-06-23	card

Exercise 2:6: UPDATE PAYMENT METHODS FOR ORDERS ABOVE A CERTAIN VALUE

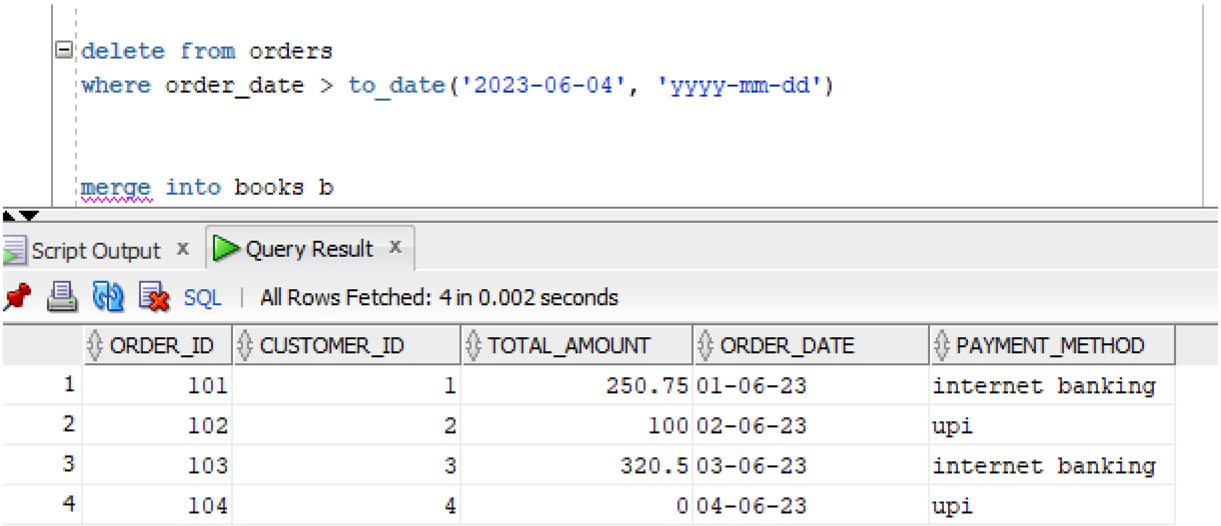


```
update orders
set payment_method = 'internet banking'
where total_amount > 200

merge into books b
```

All Rows Fetched: 5 in 0.001 seconds

	ORDER_ID	CUSTOMER_ID	TOTAL_AMOUNT	ORDER_DATE	PAYMENT_METHOD
1	101	1	250.75	01-06-23	internet banking
2	102	2	100	02-06-23	upi
3	103	3	320.5	03-06-23	internet banking
4	104	4	0	04-06-23	upi
5	105	5	150.25	05-06-23	card

Exercise 2:7: DELETE ORDERS FROM YETERDAY

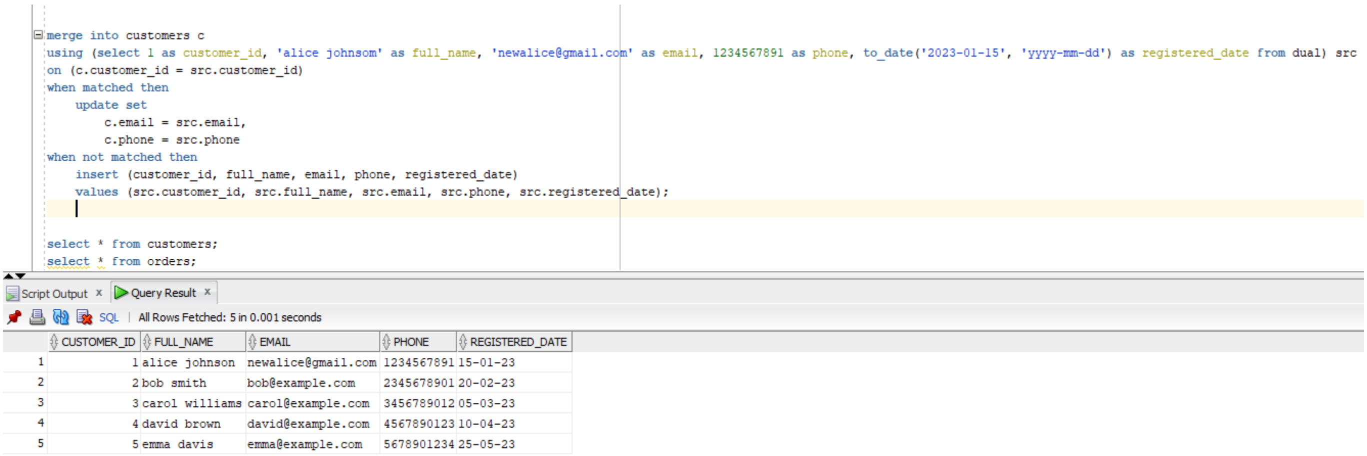


```
delete from orders
where order_date > to_date('2023-06-04', 'yyyy-mm-dd')


merge into books b
```

All Rows Fetched: 4 in 0.002 seconds

	ORDER_ID	CUSTOMER_ID	TOTAL_AMOUNT	ORDER_DATE	PAYMENT_METHOD
1	101	1	250.75	01-06-23	internet banking
2	102	2	100	02-06-23	upi
3	103	3	320.5	03-06-23	internet banking
4	104	4	0	04-06-23	upi

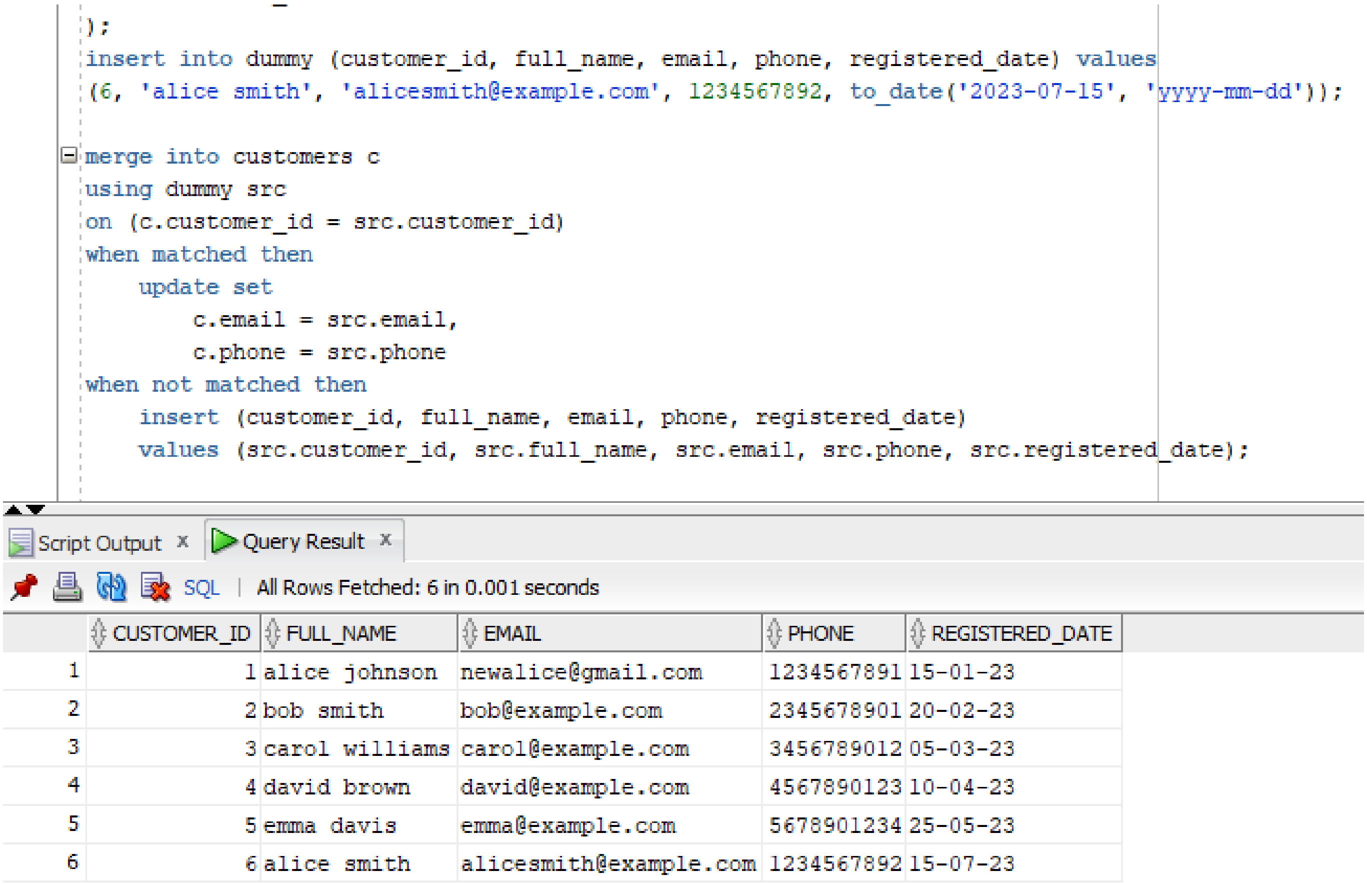
Exercise 2:7: MERGE TO UPDATE CUSTOMER DATA



```sql
merge into customers c
using (select 1 as customer_id, 'alice johnsom' as full_name, 'newalice@gmail.com' as email, 1234567891 as phone, to_date('2023-01-15', 'yyyy-mm-dd') as registered_date from dual) src
on (c.customer_id = src.customer_id)
when matched then
    update set
        c.email = src.email,
        c.phone = src.phone
when not matched then
    insert (customer_id, full_name, email, phone, registered_date)
    values (src.customer_id, src.full_name, src.email, src.phone, src.registered_date);


select * from customers;
select * from orders;
```

All Rows Fetched: 5 in 0.001 seconds

	CUSTOMER_ID	FULL_NAME	EMAIL	PHONE	REGISTERED_DATE
1	1	alice johnson	newalice@gmail.com	1234567891	15-01-23
2	2	bob smith	bob@example.com	2345678901	20-02-23
3	3	carol williams	carol@example.com	3456789012	05-03-23
4	4	david brown	david@example.com	4567890123	10-04-23
5	5	emma davis	emma@example.com	5678901234	25-05-23



```sql
);
insert into dummy (customer_id, full_name, email, phone, registered_date) values
(6, 'alice smith', 'alicesmith@example.com', 1234567892, to_date('2023-07-15', 'yyyy-mm-dd'));

merge into customers c
using dummy src
on (c.customer_id = src.customer_id)
when matched then
    update set
        c.email = src.email,
        c.phone = src.phone
when not matched then
    insert (customer_id, full_name, email, phone, registered_date)
    values (src.customer_id, src.full_name, src.email, src.phone, src.registered_date);
```

All Rows Fetched: 6 in 0.001 seconds

	CUSTOMER_ID	FULL_NAME	EMAIL	PHONE	REGISTERED_DATE
1	1	alice johnson	newalice@gmail.com	1234567891	15-01-23
2	2	bob smith	bob@example.com	2345678901	20-02-23
3	3	carol williams	carol@example.com	3456789012	05-03-23
4	4	david brown	david@example.com	4567890123	10-04-23
5	5	emma davis	emma@example.com	5678901234	25-05-23
6	6	alice smith	alicesmith@example.com	1234567892	15-07-23