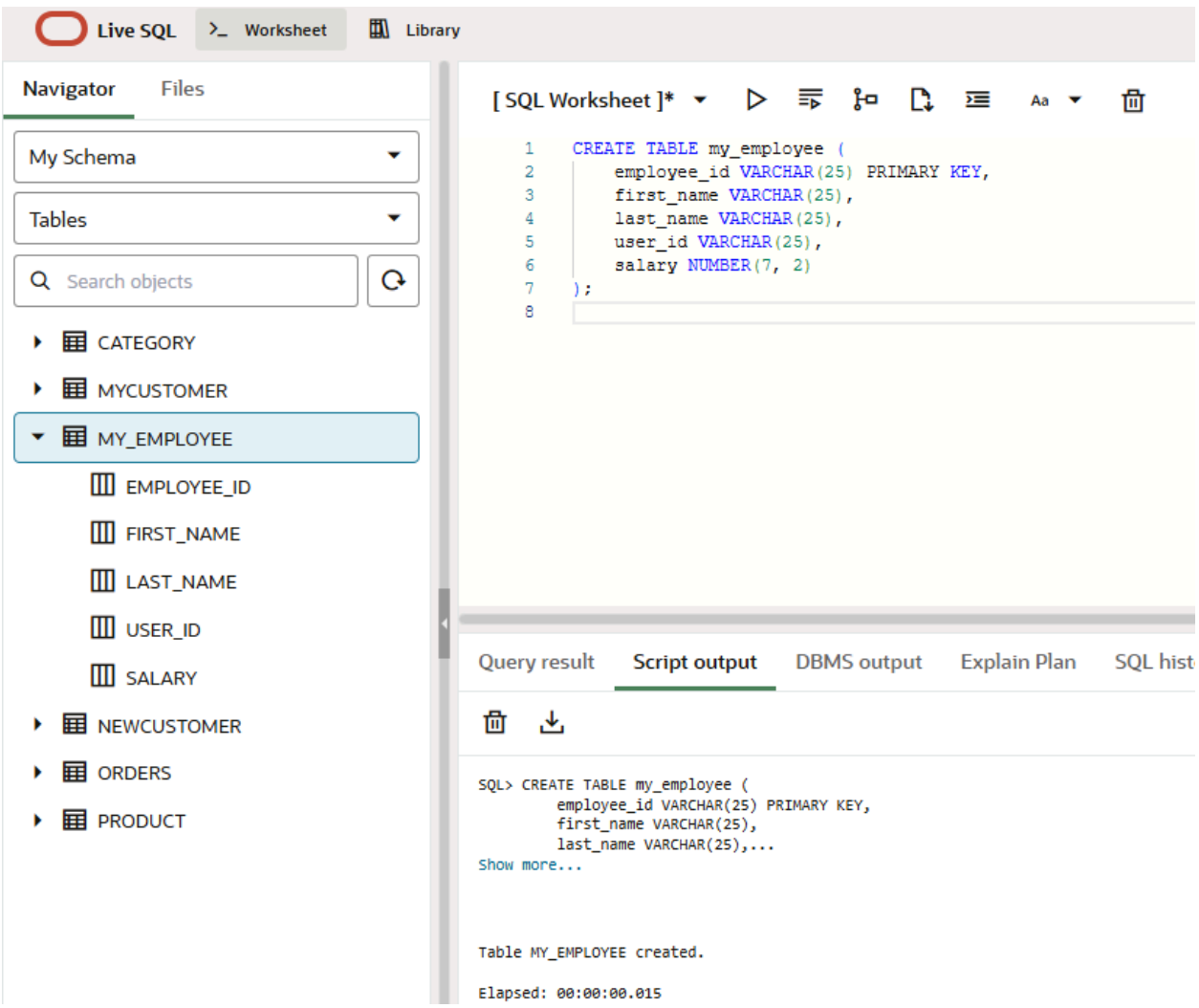
Q1 Create MY_EMPLOYEE table having attributes employee_id, first_name, last_name, userid, salary. Choose appropriate data types for each field. Make employee_id as primary key.

Query

```
CREATE TABLE my_employee (
    employee_id VARCHAR(25) PRIMARY KEY,
    first_name VARCHAR(25),
    last_name VARCHAR(25),
    user_id VARCHAR(25),
    salary NUMBER(7, 2)
);
```

Output



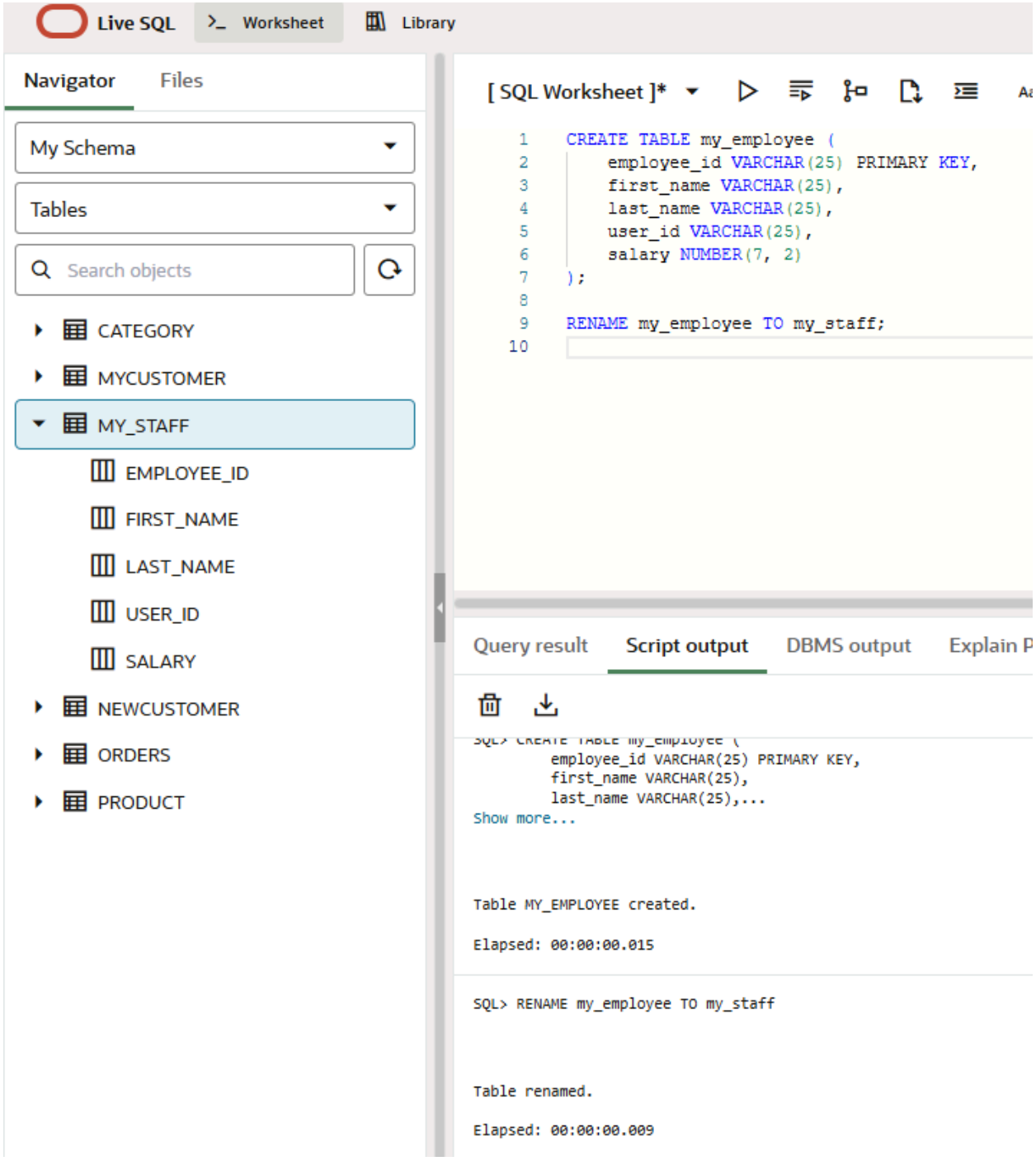
Q2

Rename MY_EMPLOYEE table as MY_STAFF.

Query

```
RENAME my_employee TO my_staff;
```

Output



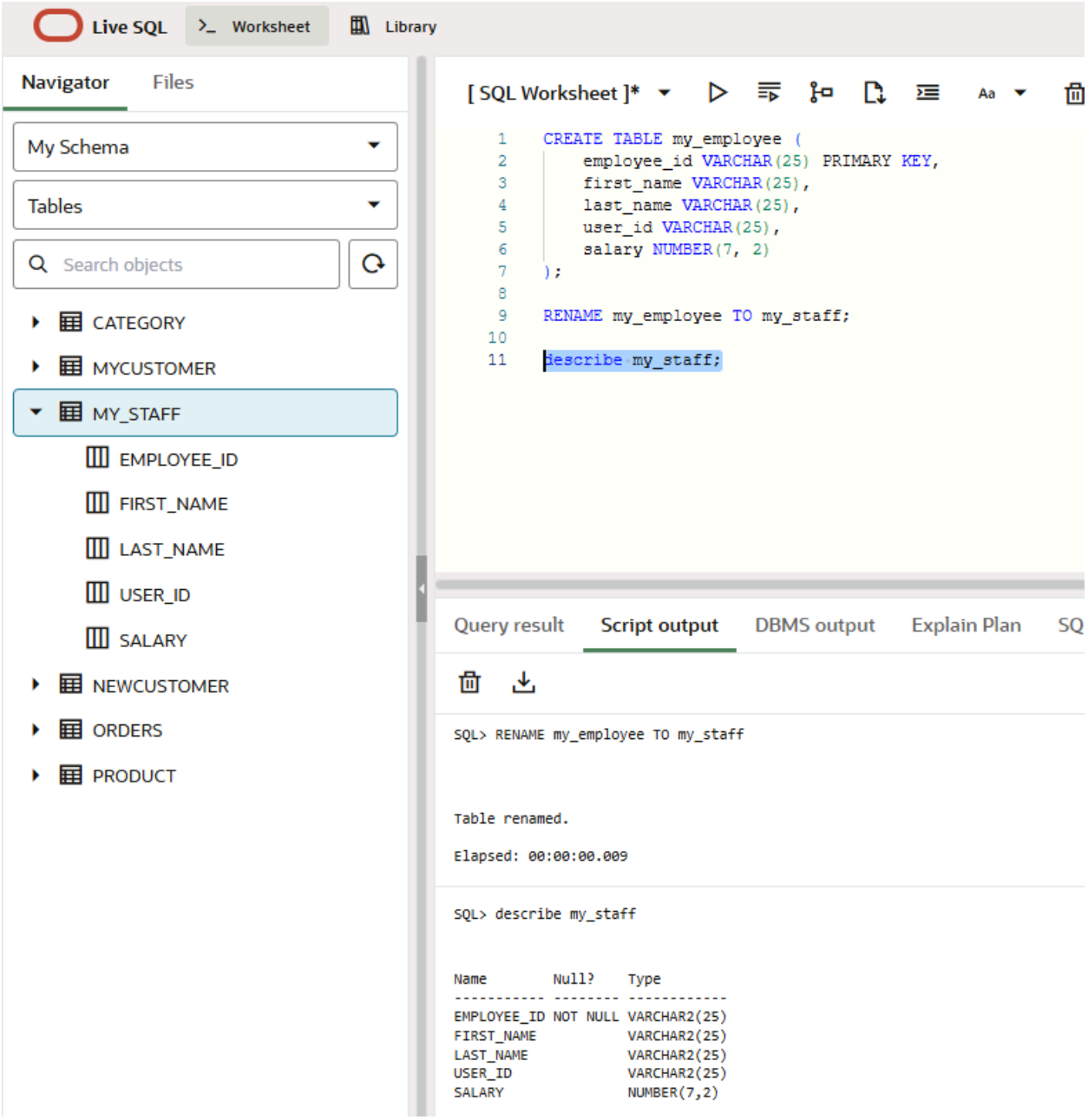
Q3

Describe the structure of the MY_STAFF table to identify the column names.

Query

```
describe my_staff;
```

Output



Q4

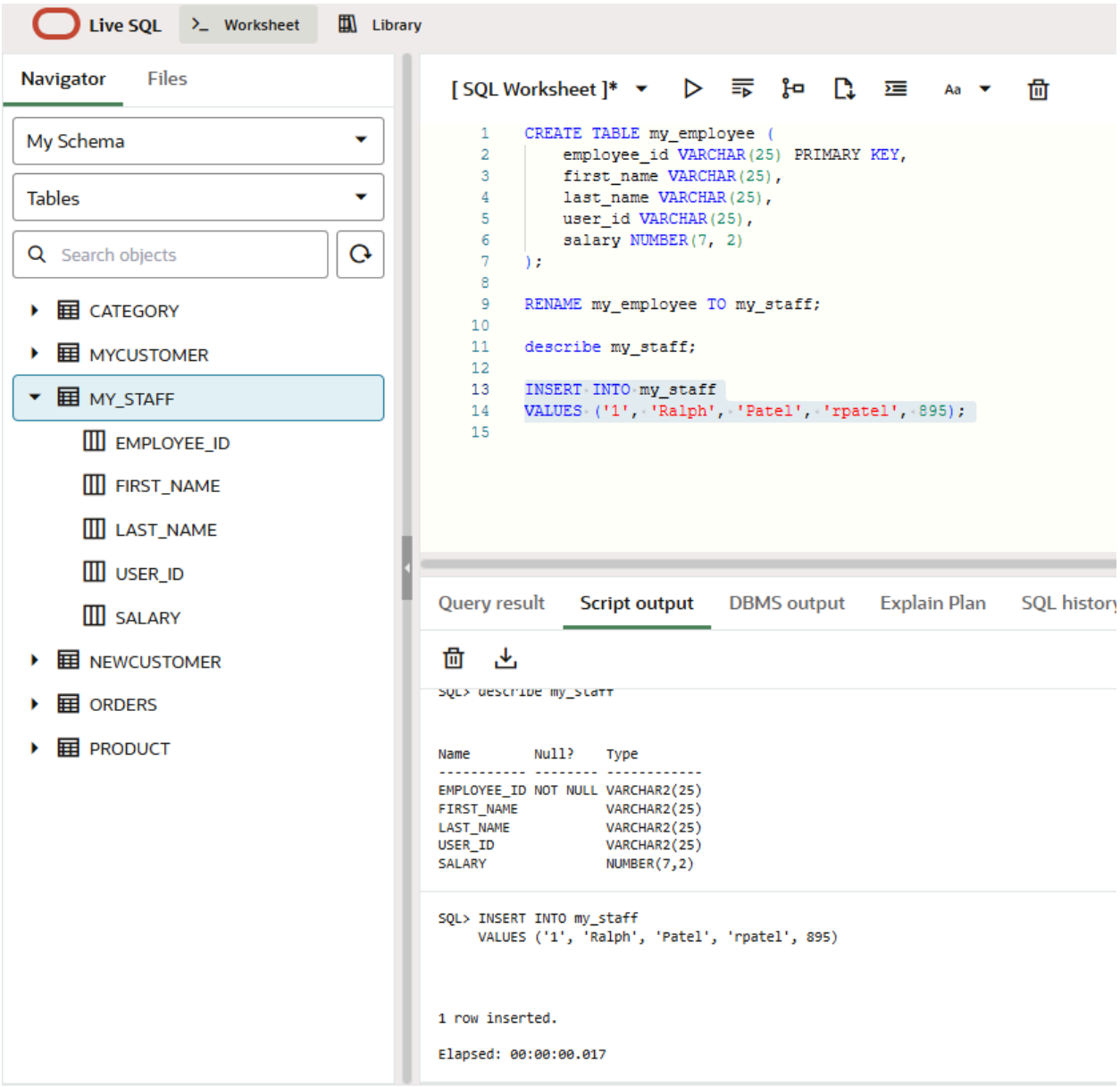
Add the first row of data to the MY_STAFF table from the following sample data. Do not list

the columns in the INSERT clause.

Query

```
INSERT INTO my_staff
VALUES ('1', 'Ralph', 'Patel', 'rpatel', 895);
```

Output



Q5

Populate the MY_STAFF table with the second row of sample data from the preceding list.

This time, list the columns explicitly in the INSERT clause.

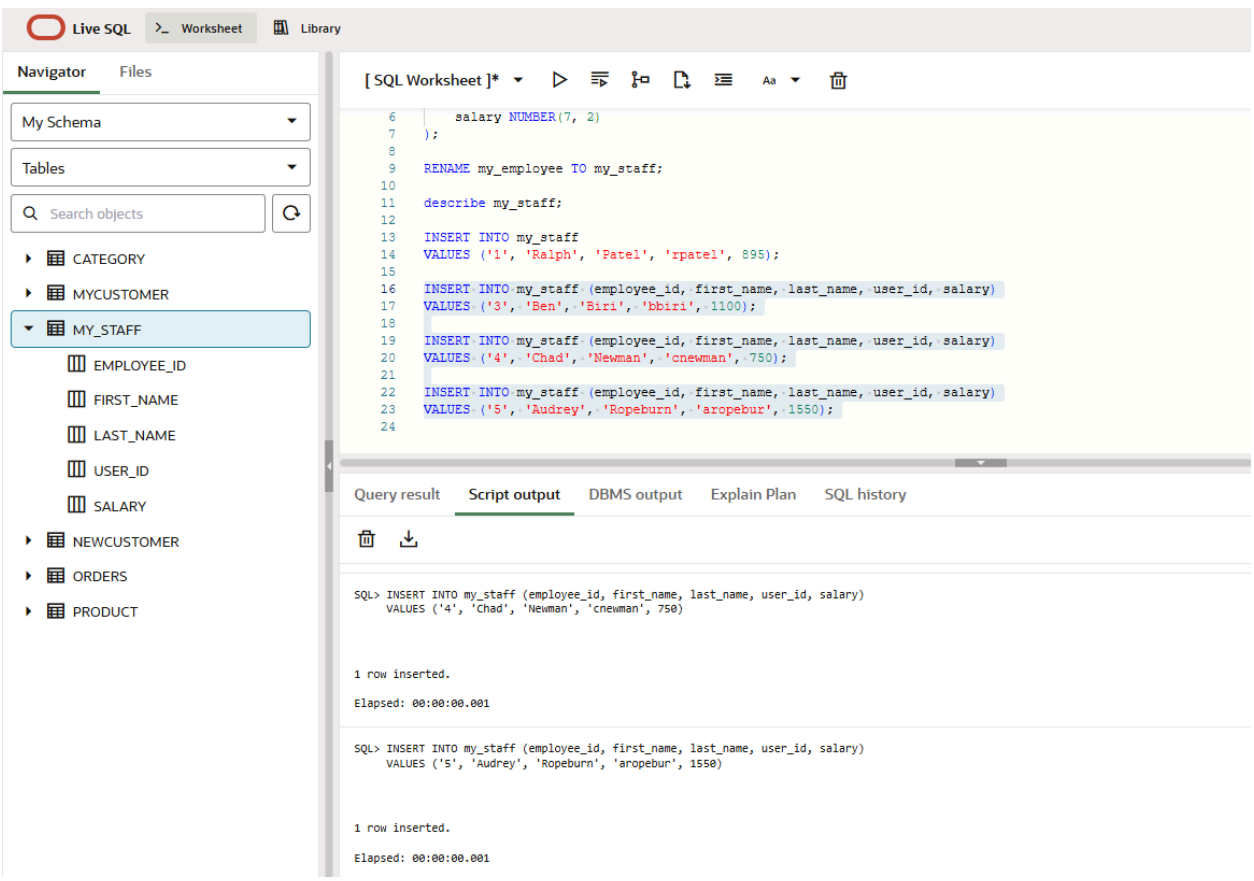
Query

```
INSERT INTO my_staff (employee_id, first_name, last_name, user_id, salary)
VALUES ('3', 'Ben', 'Biri', 'bbiri', 1100);

INSERT INTO my_staff (employee_id, first_name, last_name, user_id, salary)
VALUES ('4', 'Chad', 'Newman', 'cnewman', 750);

INSERT INTO my_staff (employee_id, first_name, last_name, user_id, salary)
VALUES ('5', 'Audrey', 'Ropeburn', 'aropebur', 1550);
```

Output



Q6

Confirm your addition to the table.

Query

```
SELECT * FROM my_staff;
```

Output

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	USER_ID	SALARY
1	1	Ralph	Patel	rpatel	895
2	3	Ben	Biri	bbiri	1100
3	4	Chad	Newman	cnewman	750
4	5	Audrey	Ropeburn	aropebur	1550

Question 7

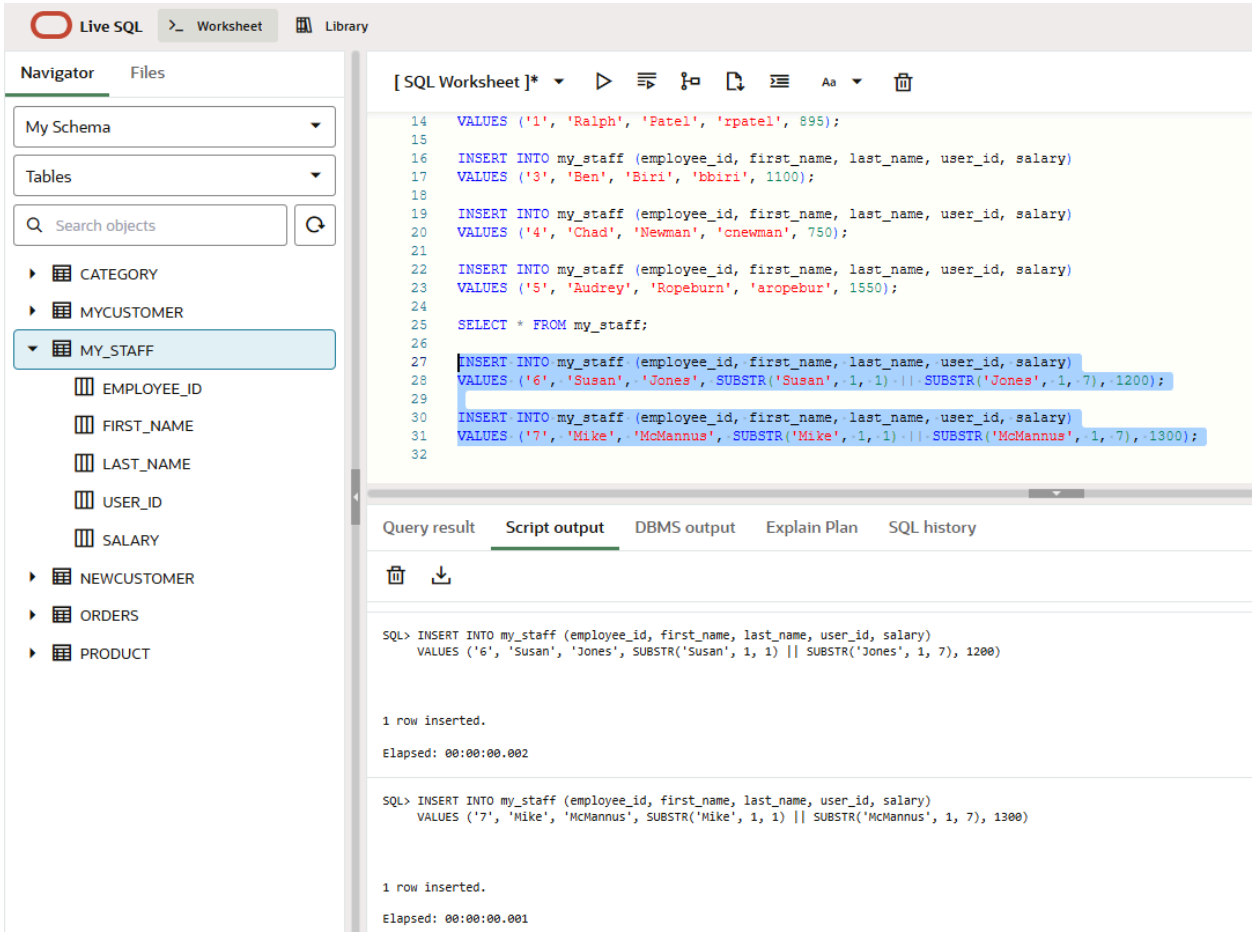
Populate the table with the next two rows of sample data by running the INSERT statement in such a way that you concatenate the first letter of the first name and the first seven characters of the last name to produce the user ID, instead of hardcoding it.

Query

```
INSERT INTO my_staff (employee_id, first_name, last_name, user_id, salary)
VALUES ('6', 'Susan', 'Jones', SUBSTR('Susan', 1, 1) || SUBSTR('Jones', 1, 7), 1200);

INSERT INTO my_staff (employee_id, first_name, last_name, user_id, salary)
VALUES ('7', 'Mike', 'McMannus', SUBSTR('Mike', 1, 1) || SUBSTR('McMannus', 1, 7),
1300);
```

Output



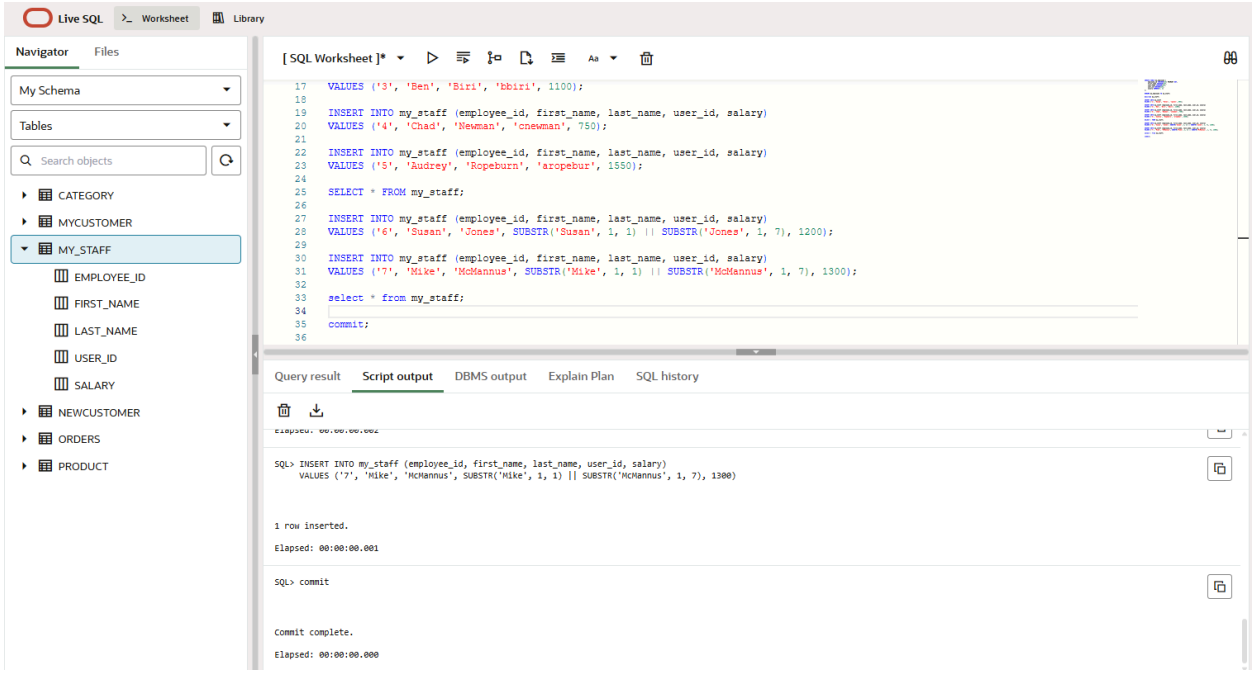
Question 8

Confirm your additions to the table and make the data additions permanent.

Query

```
select * from my_staff;
commit;
```

Output



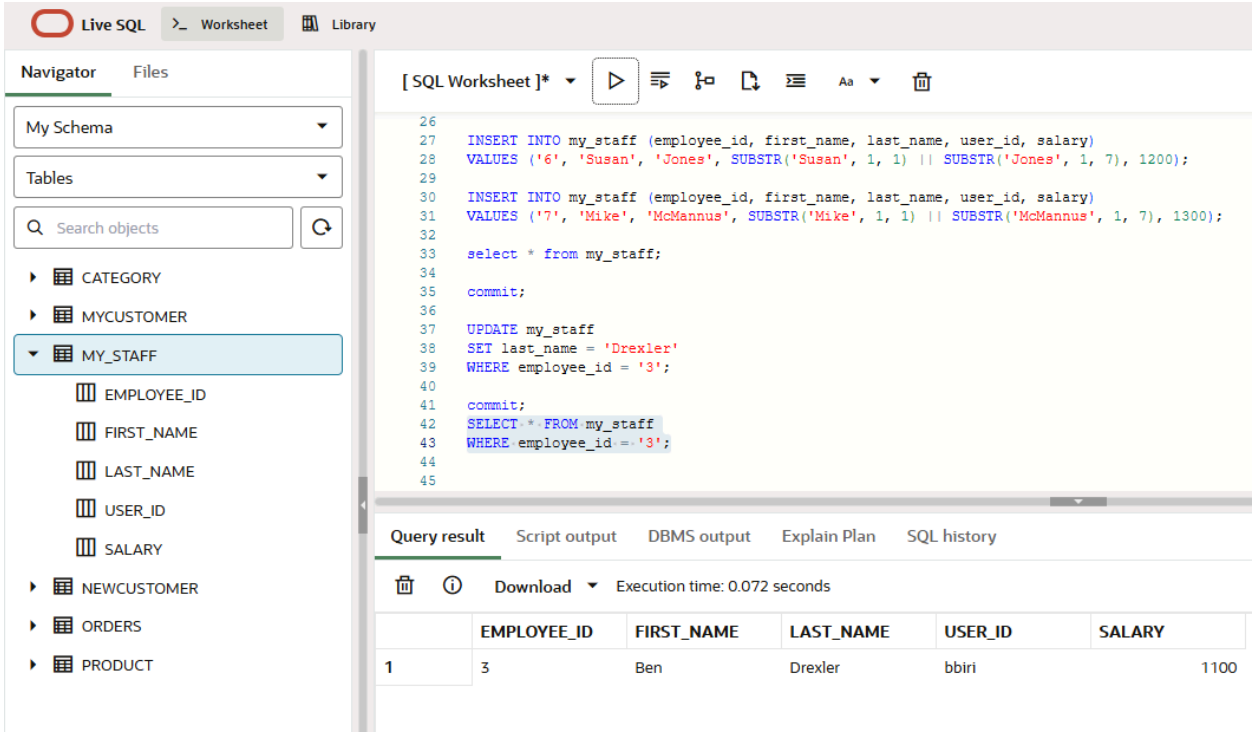
Q9

Change the last name of employee 3 to Drexler.

Query

```
UPDATE my_staff
SET last_name = 'Drexler'
WHERE employee_id = '3';
```

Output



Q10

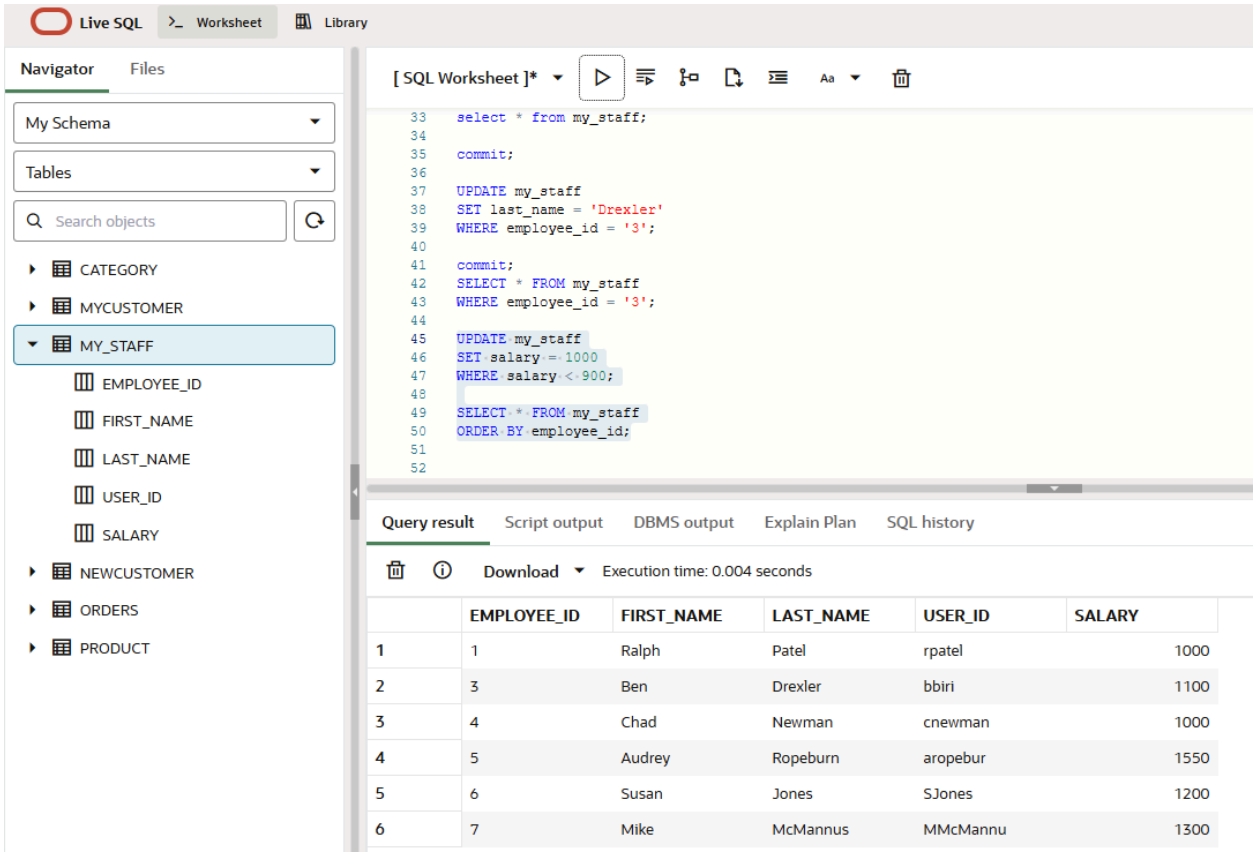
Change the salary to 1000 for all employees with a salary less than 900. Verify your changes to the table.

Query

```
UPDATE my_staff
SET salary = 1000
WHERE salary < 900;

SELECT * FROM my_staff
ORDER BY employee_id;
```

Output



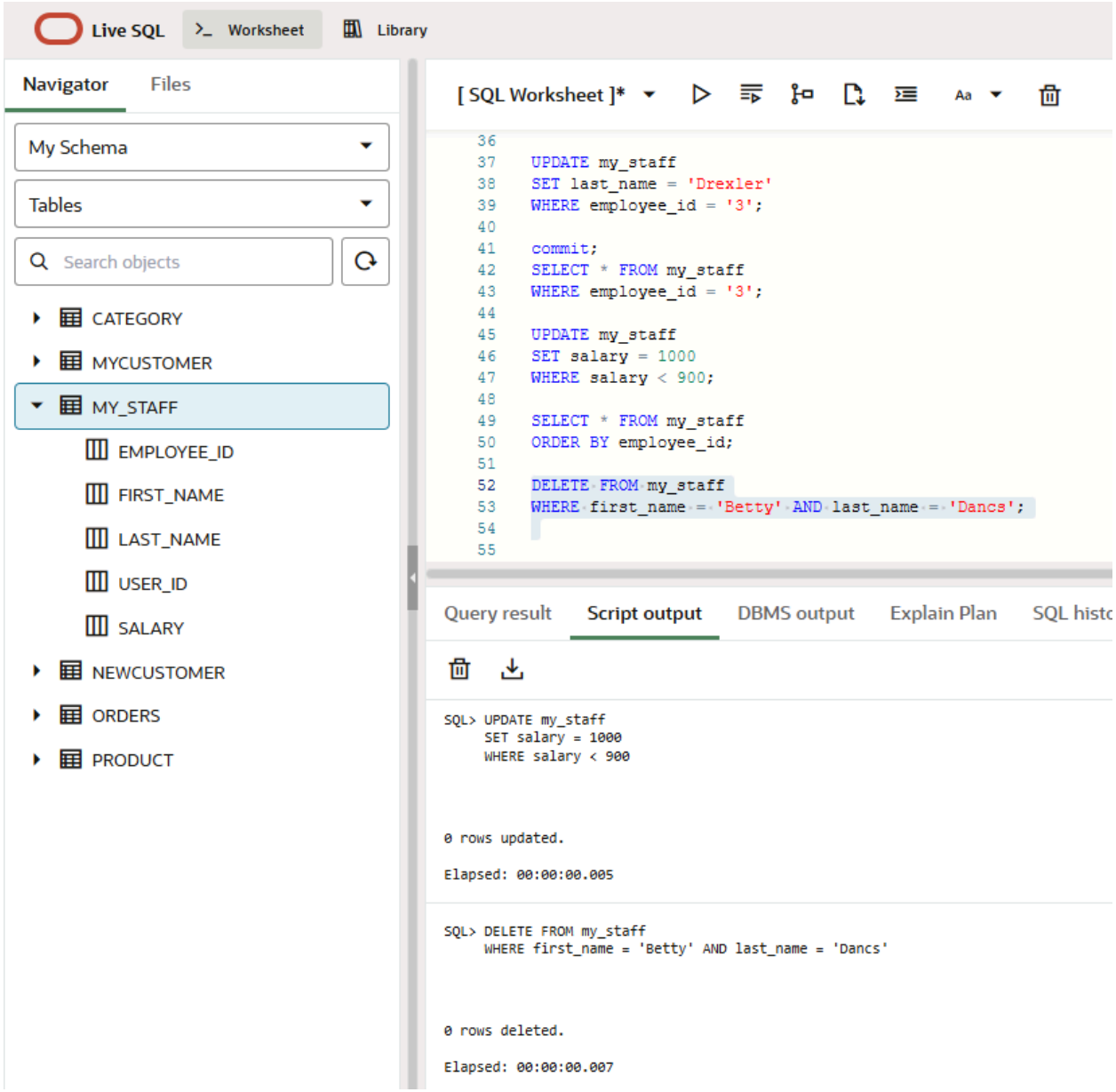
Q11

Delete Betty Dancs from the MY_STAFF table.

Query

```
DELETE FROM my_staff
WHERE first_name = 'Betty' AND last_name = 'Dancs';
```

Output



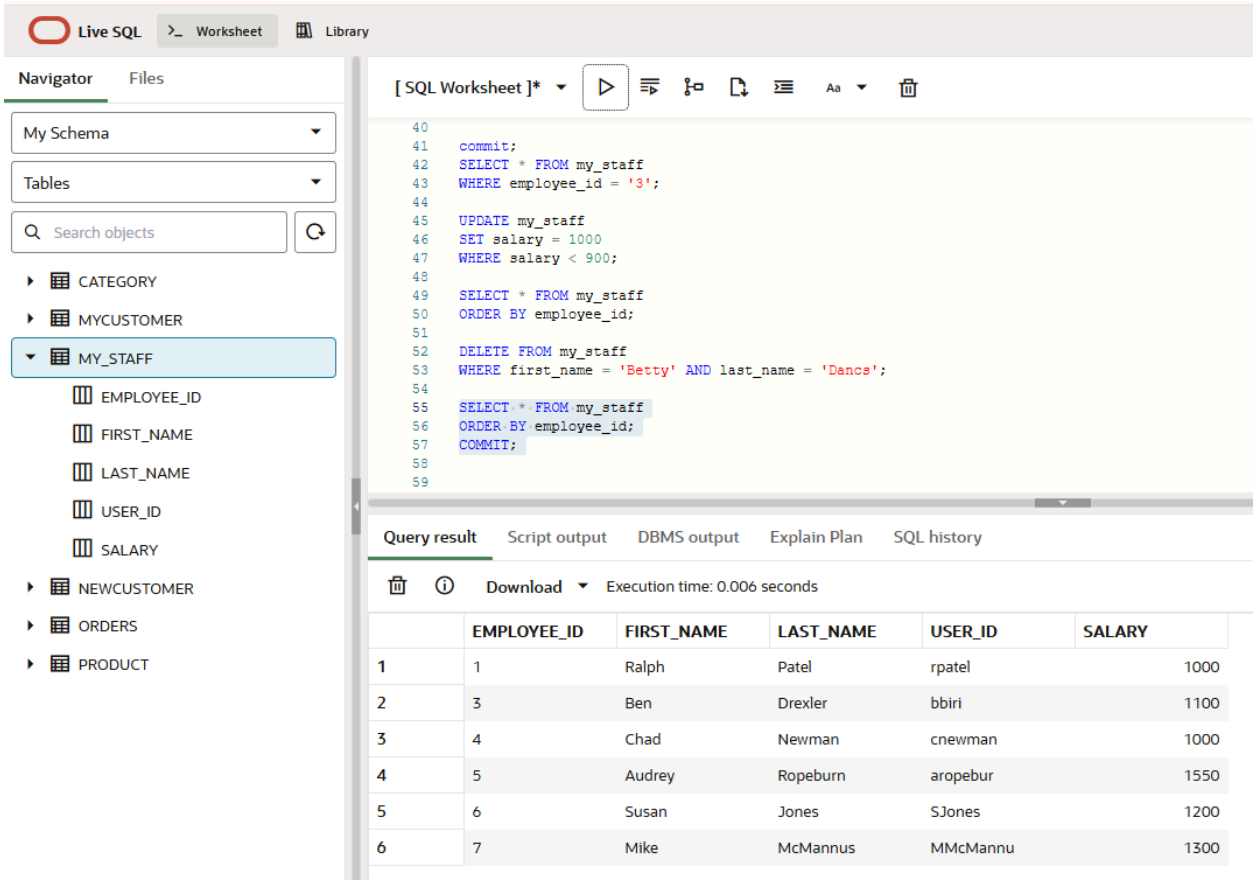
Q12

Confirm your changes to the table and Commit all pending changes.

Query

```
SELECT * FROM my_staff
ORDER BY employee_id;
COMMIT;
```

Output



	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	USER_ID	SALARY
1	1	Ralph	Patel	rpatel	1000
2	3	Ben	Drexler	bbiri	1100
3	4	Chad	Newman	cnewman	1000
4	5	Audrey	Ropeburn	aropebur	1550
5	6	Susan	Jones	SJones	1200
6	7	Mike	McMannus	MMcMannu	1300

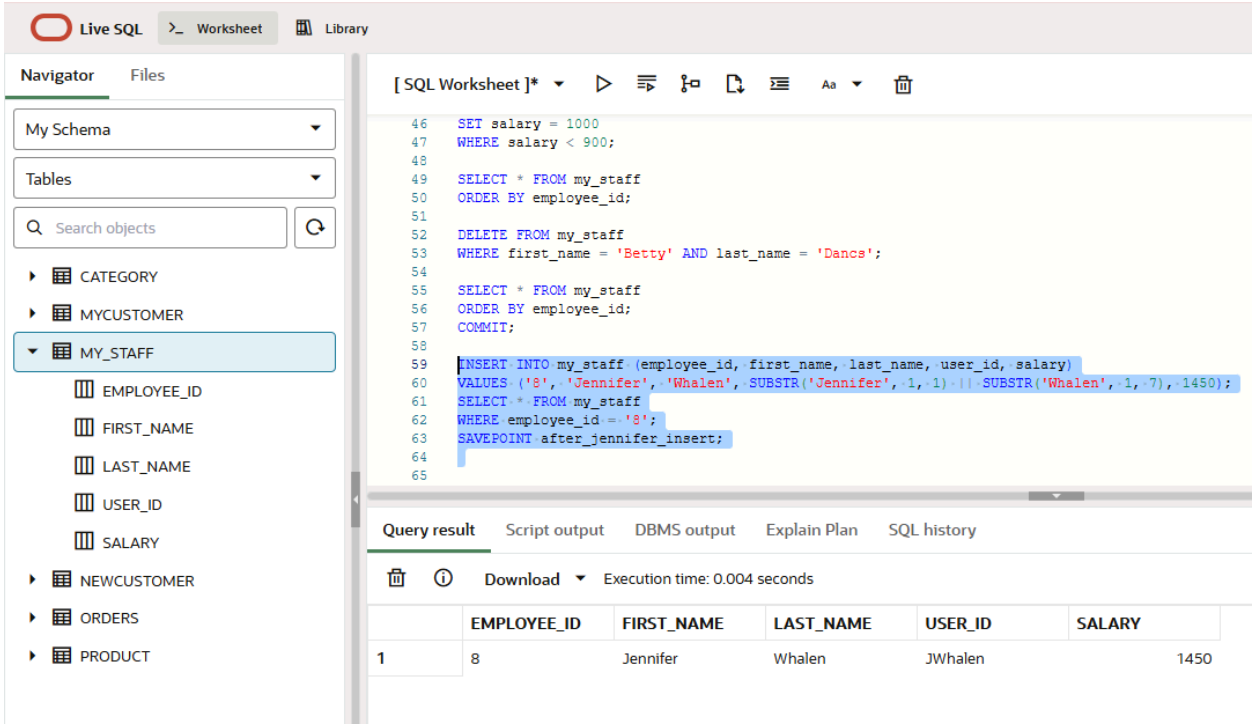
Q13

Populate the table with the last row of sample data by modifying the insert statement that you created in step 7. Confirm your addition to the table. Create savepoint.

Query

```
INSERT INTO my_staff (employee_id, first_name, last_name, user_id, salary)
VALUES ('8', 'Jennifer', 'Whalen', SUBSTR('Jennifer', 1, 1) || SUBSTR('Whalen', 1, 7),
1450);
SELECT * FROM my_staff
WHERE employee_id = '8';
SAVEPOINT after_jennifer_insert;
```

Output



Q14

Empty the entire table and confirm that the table is empty.

Query

```
DELETE FROM my_staff;
SELECT * FROM my_staff;
COMMIT;
```

Output

[SQL Worksheet]*

```
ORDER BY employee_id;

DELETE FROM my_staff
WHERE first_name = 'Betty' AND last_name = 'Dancs';

SELECT * FROM my_staff
ORDER BY employee_id;
COMMIT;

INSERT INTO my_staff (employee_id, first_name, last_name, user_id, salary)
VALUES ('8', 'Jennifer', 'Whalen', SUBSTR('Jennifer', 1, 1) || SUBSTR('Whalen', 1, 7), 1450);
SELECT * FROM my_staff
WHERE employee_id = '8';
SAVEPOINT after_jennifer_insert;

DELETE FROM my_staff;
SELECT * FROM my_staff;
COMMIT;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

Download Execution time: 0.004 seconds

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	USER_ID	SALARY

No items to display.

Q15

Discard the most recent DELETE operation without discarding the earlier INSERT operation.

Confirm that the new row is still intact. Make the data addition now permanent.

Query

```
ROLLBACK TO after_jennifer_insert;
SELECT * FROM my_staff
WHERE employee_id = '8';
COMMIT;
```

Output

Q16

Create MY_DEPART table with necessary attributes. Choose appropriate data types for each

field. Make depart_id as primary key.

Query

```
CREATE TABLE my_depart (
    depart_id VARCHAR2(10) PRIMARY KEY,
    depart_name VARCHAR2(50)
);
```

Output

[SQL Worksheet]*

```
CREATE TABLE my_depart (
    depart_id VARCHAR2(10) PRIMARY KEY,
    depart_name VARCHAR2(50)
);
```

Query result | Script output | DBMS output | Explain Plan | SQL history

```
SQL> COMMIT


Commit complete.

Elapsed: 00:00:00.000
```

```
SQL> CREATE TABLE my_depart (
         depart_id VARCHAR2(10) PRIMARY KEY,
         depart_name VARCHAR2(50)
     )



Table MY_DEPART created.

Elapsed: 00:00:00.021
```

Q17

Modify MY_STAFF table to include depart_id as foreign key. What happens to previous 5

rows with this change in the MY_STAFF table?

Query

```
ALTER TABLE my_staff
ADD depart_id VARCHAR2(10);

ALTER TABLE my_staff
ADD CONSTRAINT fk_depart
FOREIGN KEY (depart_id)
REFERENCES my_depart(depart_id);
```

Output

Live SQL | Worksheet | Library

Navigator | Files

My Schema

Tables

Search objects

- CATEGORY
- MYCUSTOMER
- MY_DEPART
- MY_STAFF
- NEWCUSTOMER
- ORDERS
- PRODUCT

[SQL Worksheet]*

```sql
CREATE TABLE my_depart (
    depart_id VARCHAR2(10) PRIMARY KEY,
    depart_name VARCHAR2(50)
);

ALTER TABLE my_staff
ADD depart_id VARCHAR2(10);

ALTER TABLE my_staff
ADD CONSTRAINT fk_depart
FOREIGN KEY (depart_id)
REFERENCES my_depart(depart_id);
```

Query result | Script output | DBMS output | Explain Plan | SQL histo

```
SQL> ALTER TABLE my_staff
     ADD depart_id VARCHAR2(10)


Table MY_STAFF altered.

Elapsed: 00:00:00.026

SQL> ALTER TABLE my_staff
     ADD CONSTRAINT fk_depart
     FOREIGN KEY (depart_id)
     REFERENCES my_depart(depart_id)


Table MY_STAFF altered.

Elapsed: 00:00:00.013
```