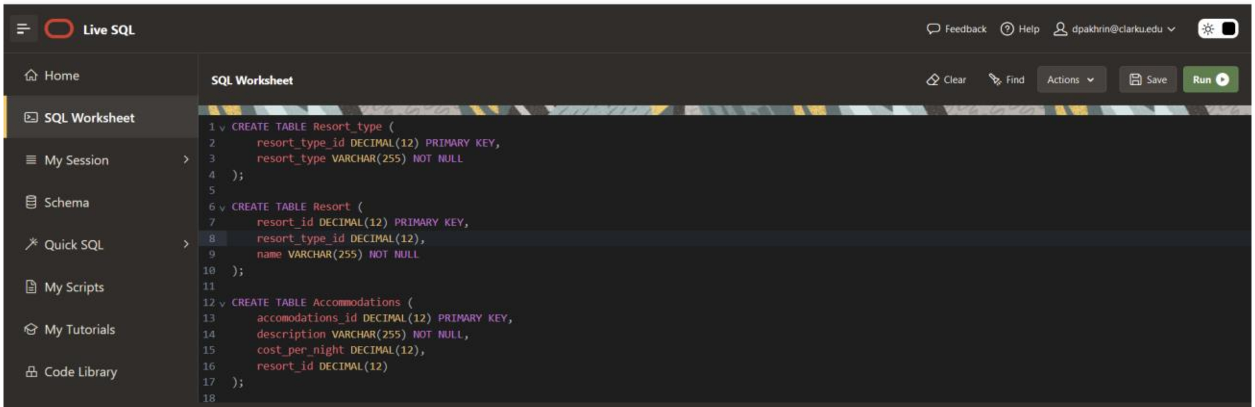
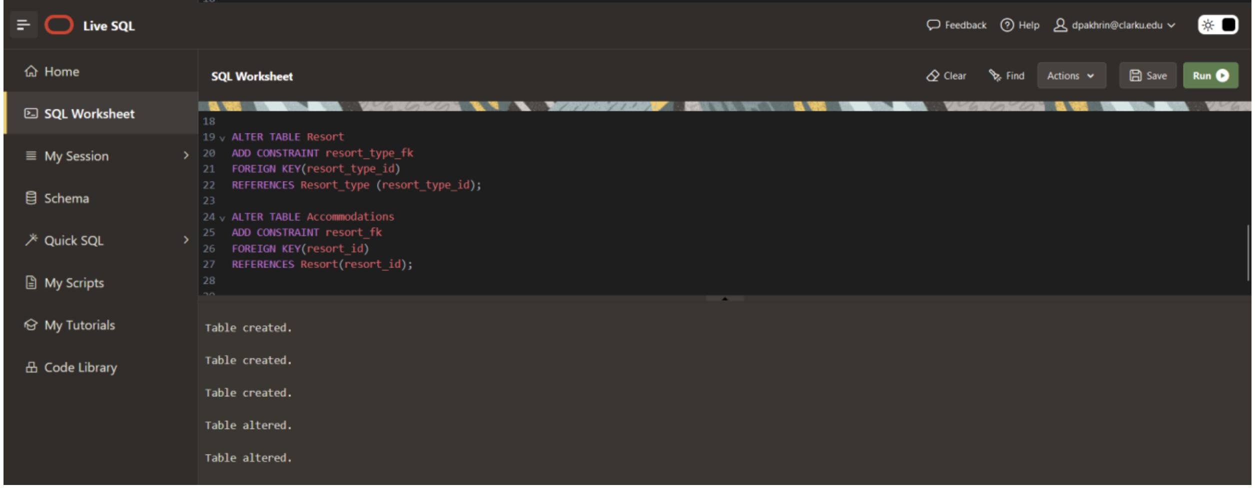
Lab 3 – Submission Sheet

Section One

1. CREATE TABLE & ALTER TABLE (Screen Shots)



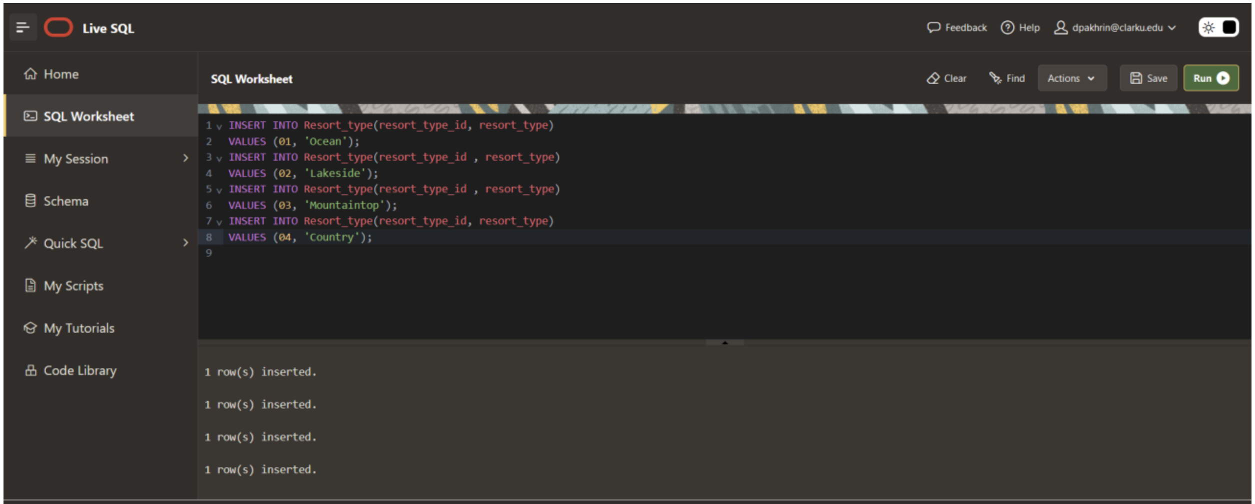
```
CREATE TABLE Resort_type (
    resort_type_id DECIMAL(12) PRIMARY KEY,
    resort_type VARCHAR(255) NOT NULL
);

CREATE TABLE Resort (
    resort_id DECIMAL(12) PRIMARY KEY,
    resort_type_id DECIMAL(12),
    name VARCHAR(255) NOT NULL
);

CREATE TABLE Accommodations (
    accomodations_id DECIMAL(12) PRIMARY KEY,
    description VARCHAR(255) NOT NULL,
    cost_per_night DECIMAL(12),
    resort_id DECIMAL(12)
);
```



```
ALTER TABLE Resort
ADD CONSTRAINT resort_type_fk
FOREIGN KEY(resort_type_id)
REFERENCES Resort_type (resort_type_id);

ALTER TABLE Accommodations
ADD CONSTRAINT resort_fk
FOREIGN KEY(resort_id)
REFERENCES Resort(resort_id);
```

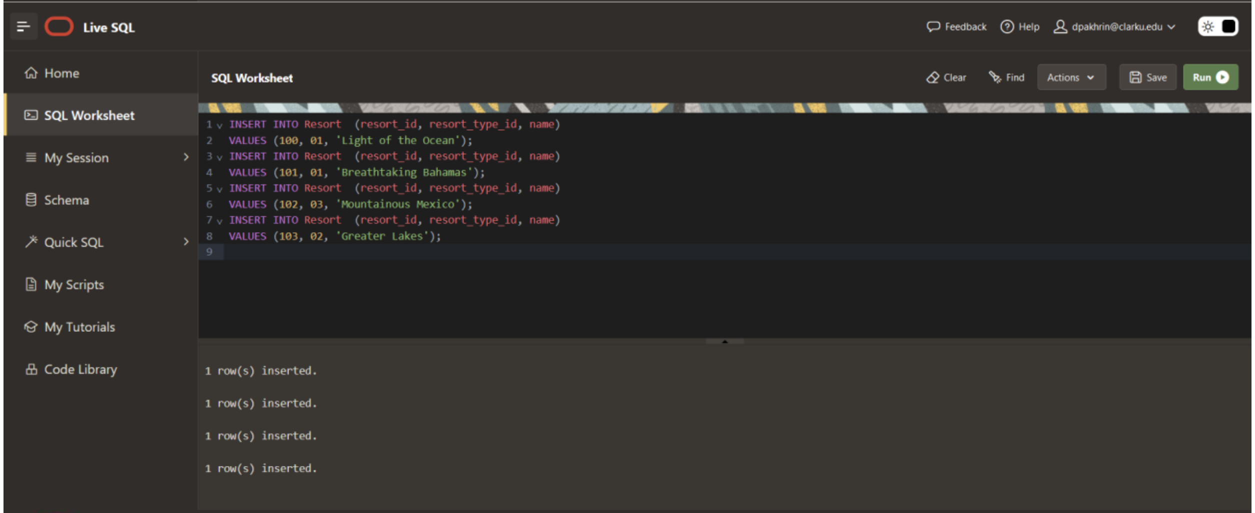
```
Table created.

Table created.

Table created.

Table altered.

Table altered.
```

2. INSERT (Screen Shots)



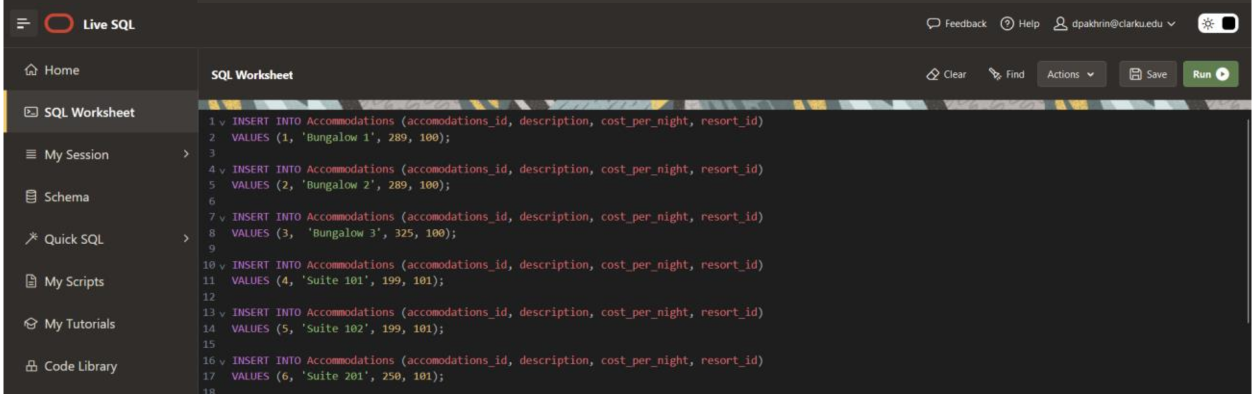
```sql
INSERT INTO Resort_type(resort_type_id, resort_type)
VALUES (01, 'Ocean');
INSERT INTO Resort_type(resort_type_id , resort_type)
VALUES (02, 'Lakeside');
INSERT INTO Resort_type(resort_type_id , resort_type)
VALUES (03, 'Mountaintop');
INSERT INTO Resort_type(resort_type_id, resort_type)
VALUES (04, 'Country');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

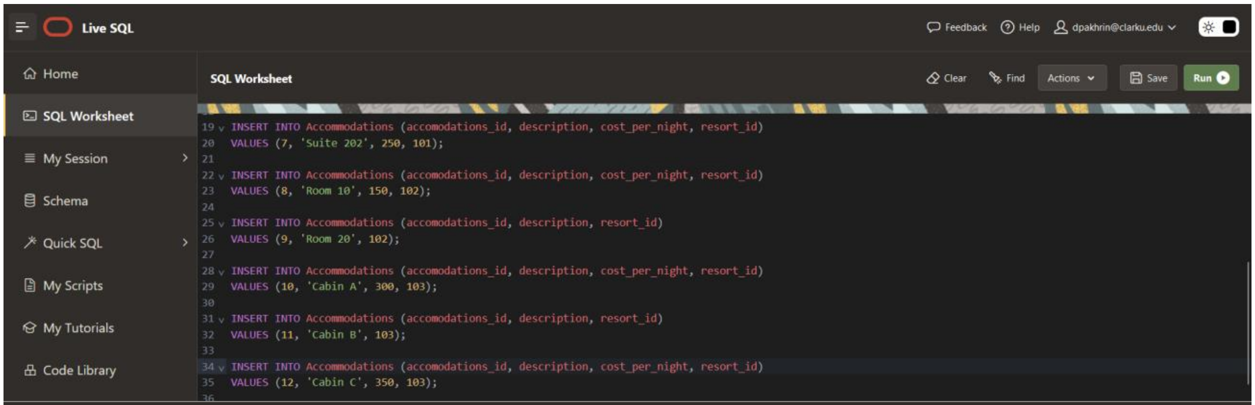


```sql
INSERT INTO Resort  (resort_id, resort_type_id, name)
VALUES (100, 01, 'Light of the Ocean');
INSERT INTO Resort  (resort_id, resort_type_id, name)
VALUES (101, 01, 'Breathtaking Bahamas');
INSERT INTO Resort  (resort_id, resort_type_id, name)
VALUES (102, 03, 'Mountainous Mexico');
INSERT INTO Resort  (resort_id, resort_type_id, name)
VALUES (103, 02, 'Greater Lakes');
```

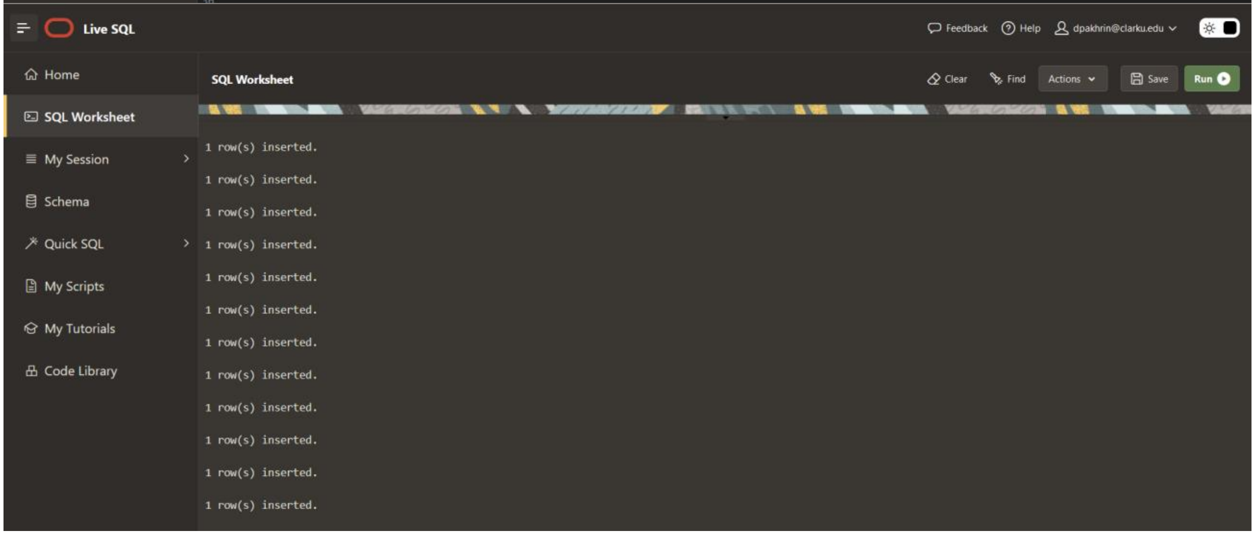
```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```



```sql
INSERT INTO Accommodations (accomodations_id, description, cost_per_night, resort_id)
VALUES (1, 'Bungalow 1', 289, 100);

INSERT INTO Accommodations (accomodations_id, description, cost_per_night, resort_id)
VALUES (2, 'Bungalow 2', 289, 100);

INSERT INTO Accommodations (accomodations_id, description, cost_per_night, resort_id)
VALUES (3,  'Bungalow 3', 325, 100);

INSERT INTO Accommodations (accomodations_id, description, cost_per_night, resort_id)
VALUES (4, 'Suite 101', 199, 101);

INSERT INTO Accommodations (accomodations_id, description, cost_per_night, resort_id)
VALUES (5, 'Suite 102', 199, 101);

INSERT INTO Accommodations (accomodations_id, description, cost_per_night, resort_id)
VALUES (6, 'Suite 201', 250, 101);
```

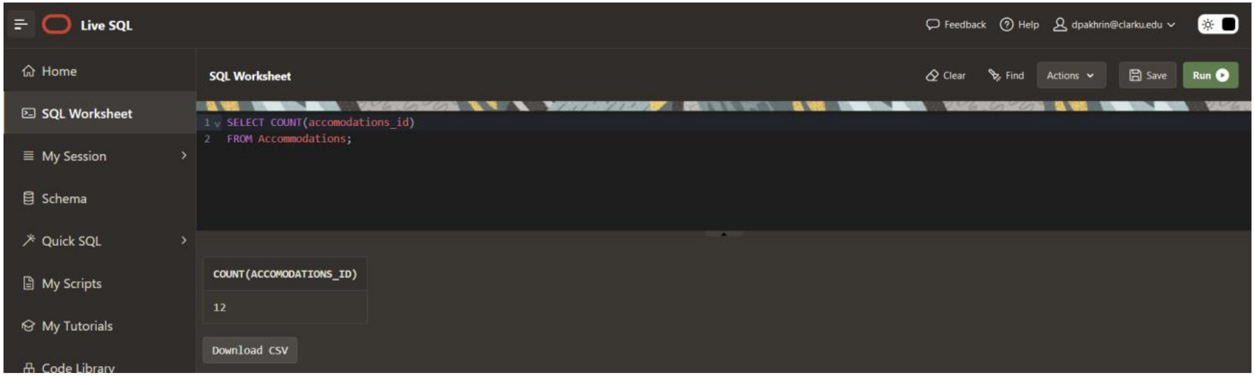


```sql
INSERT INTO Accommodations (accomodations_id, description, cost_per_night, resort_id)
VALUES (7, 'Suite 202', 250, 101);

INSERT INTO Accommodations (accomodations_id, description, cost_per_night, resort_id)
VALUES (8, 'Room 10', 150, 102);

INSERT INTO Accommodations (accomodations_id, description, resort_id)
VALUES (9, 'Room 20', 102);

INSERT INTO Accommodations (accomodations_id, description, cost_per_night, resort_id)
VALUES (10, 'Cabin A', 300, 103);

INSERT INTO Accommodations (accomodations_id, description, resort_id)
VALUES (11, 'Cabin B', 103);

INSERT INTO Accommodations (accomodations_id, description, cost_per_night, resort_id)
VALUES (12, 'Cabin C', 350, 103);
```



```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

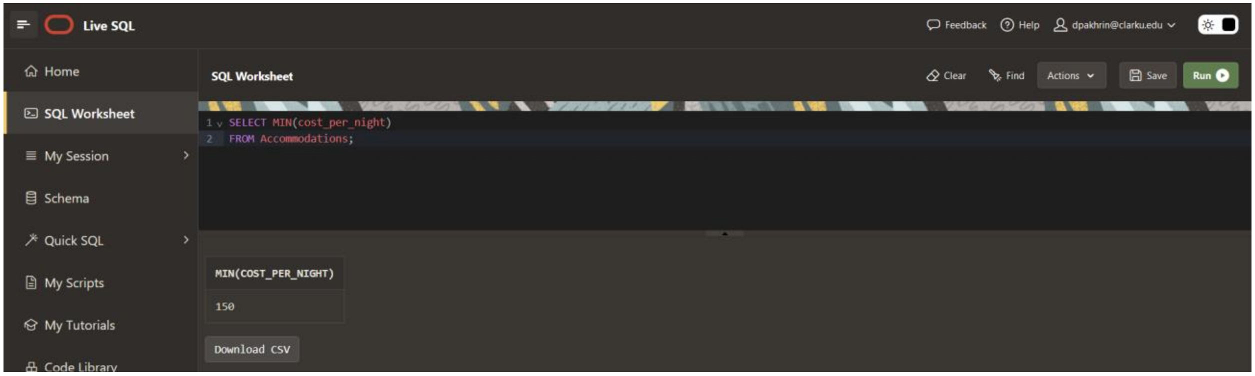
3. SELECT (Screen Shot)



```sql
SELECT COUNT(accomodations_id)
FROM Accommodations;
```

COUNT(ACCOMODATIONS_ID)
12

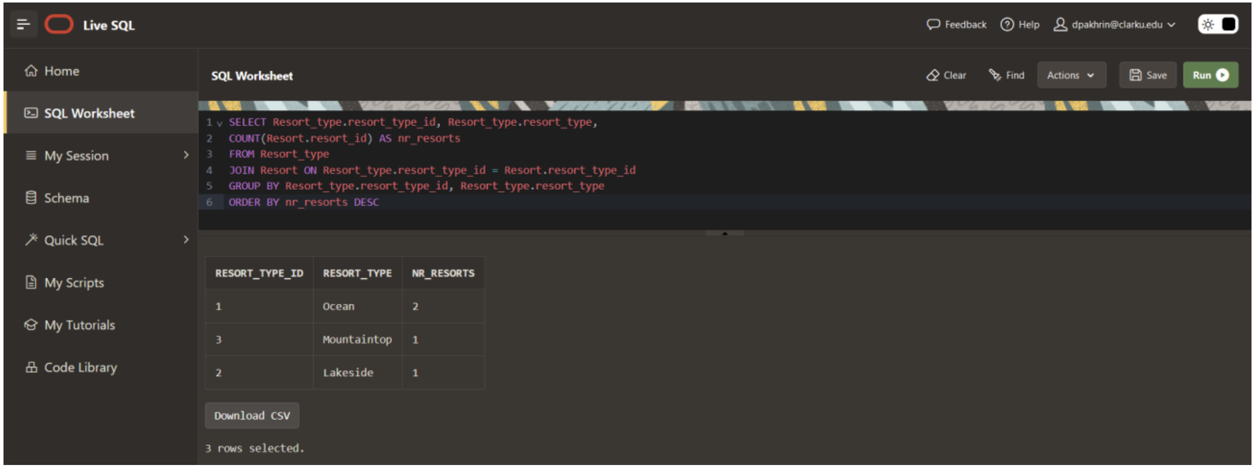
4. SELECT (Screen Shot)



```
SELECT MIN(cost_per_night)
FROM Accommodations;
```

MIN(COST_PER_NIGHT)
150

5. SELECT (Screen Shot)

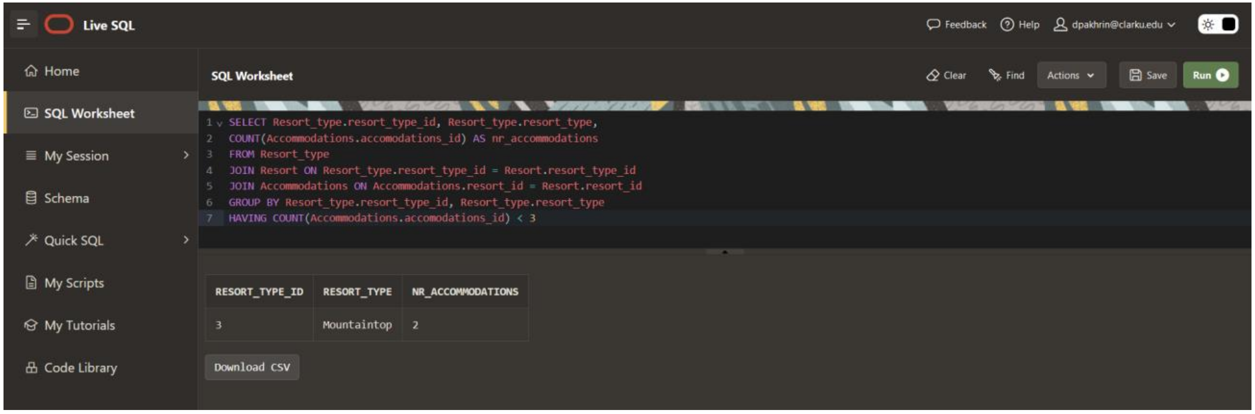


```
SELECT Resort_type.resort_type_id, Resort_type.resort_type,
COUNT(Resort.resort_id) AS nr_resorts
FROM Resort_type
JOIN Resort ON Resort_type.resort_type_id = Resort.resort_type_id
GROUP BY Resort_type.resort_type_id, Resort_type.resort_type
ORDER BY nr_resorts DESC
```

RESORT_TYPE_ID	RESORT_TYPE	NR_RESORTS
1	Ocean	2
3	Mountaintop	1
2	Lakeside	1

3 rows selected.

6. SELECT (Screen Shot)

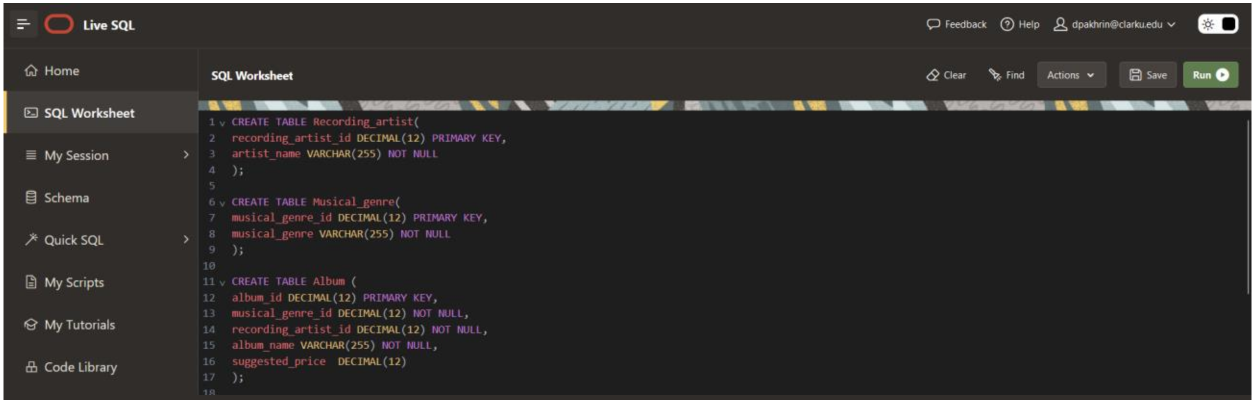


```
SELECT Resort_type.resort_type_id, Resort_type.resort_type,
COUNT(Accommodations.accomodations_id) AS nr_accommodations
FROM Resort_type
JOIN Resort ON Resort_type.resort_type_id = Resort.resort_type_id
JOIN Accommodations ON Accommodations.resort_id = Resort.resort_id
GROUP BY Resort_type.resort_type_id, Resort_type.resort_type
HAVING COUNT(Accommodations.accomodations_id) < 3
```

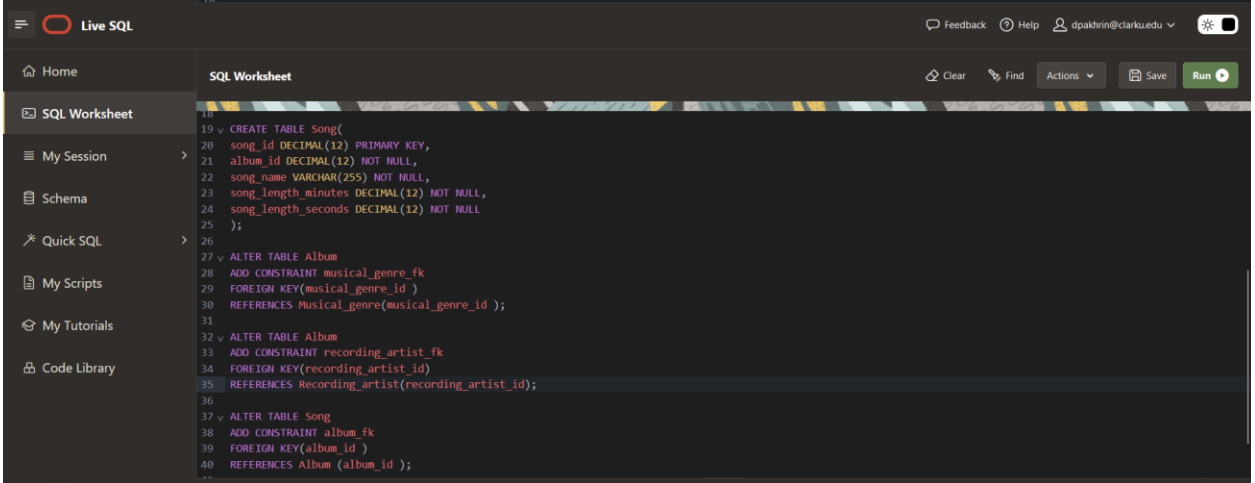
RESORT_TYPE_ID	RESORT_TYPE	NR_ACCOMMODATIONS
3	Mountaintop	2

Section Two

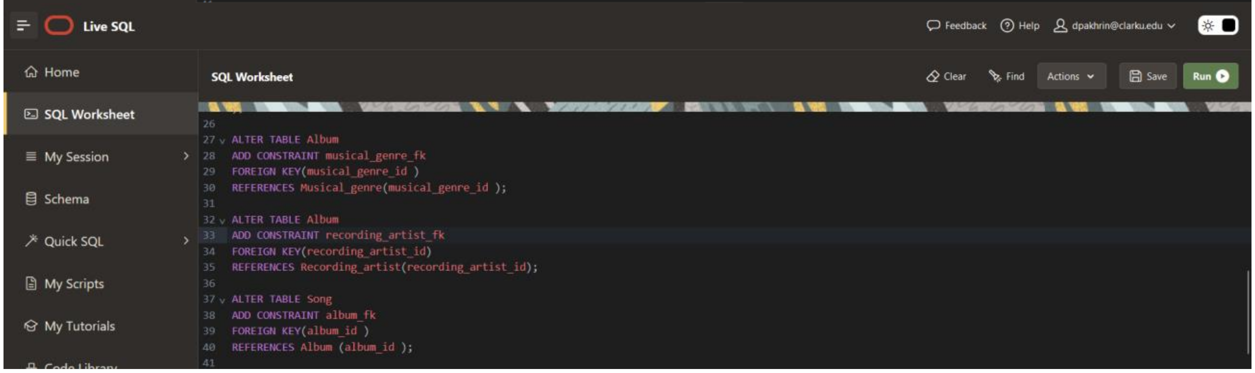
7. CREATE TABLE & ALTER TABLE (Screen Shots)



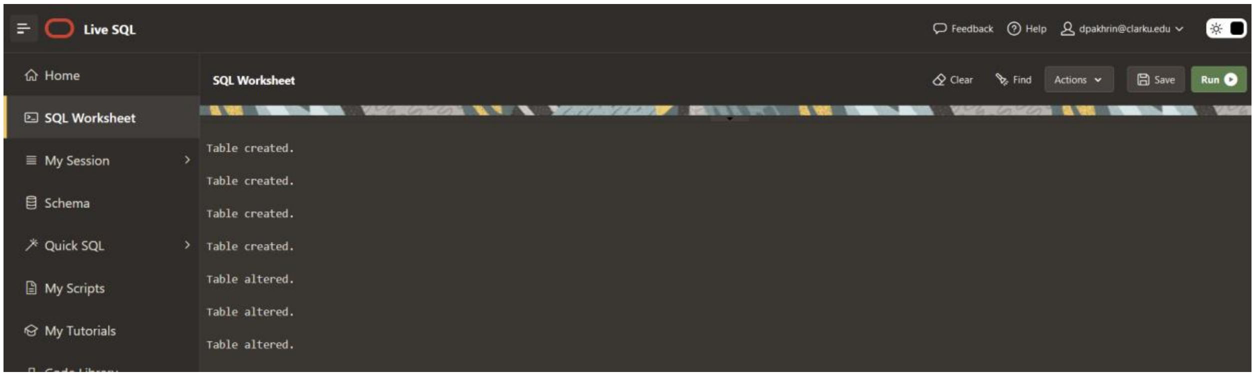
```
CREATE TABLE Recording_artist(
recording_artist_id DECIMAL(12) PRIMARY KEY,
artist_name VARCHAR(255) NOT NULL
);

CREATE TABLE Musical_genre(
musical_genre_id DECIMAL(12) PRIMARY KEY,
musical_genre VARCHAR(255) NOT NULL
);

CREATE TABLE Album (
album_id DECIMAL(12) PRIMARY KEY,
musical_genre_id DECIMAL(12) NOT NULL,
recording_artist_id DECIMAL(12) NOT NULL,
album_name VARCHAR(255) NOT NULL,
suggested_price  DECIMAL(12)
);
```



```
CREATE TABLE Song(
song_id DECIMAL(12) PRIMARY KEY,
album_id DECIMAL(12) NOT NULL,
song_name VARCHAR(255) NOT NULL,
song_length_minutes DECIMAL(12) NOT NULL,
song_length_seconds DECIMAL(12) NOT NULL
);

ALTER TABLE Album
ADD CONSTRAINT musical_genre_fk
FOREIGN KEY(musical_genre_id )
REFERENCES Musical_genre(musical_genre_id );

ALTER TABLE Album
ADD CONSTRAINT recording_artist_fk
FOREIGN KEY(recording_artist_id)
REFERENCES Recording_artist(recording_artist_id);

ALTER TABLE Song
ADD CONSTRAINT album_fk
FOREIGN KEY(album_id )
REFERENCES Album (album_id );
```

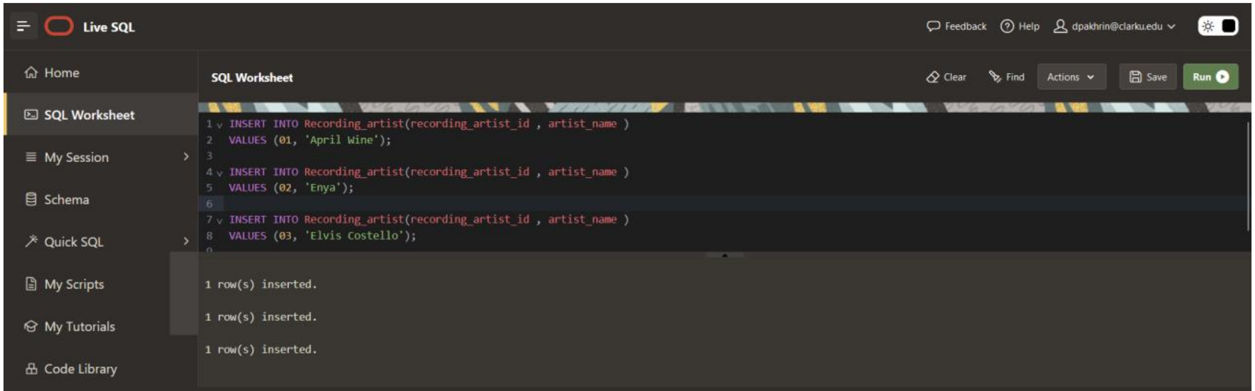


```
ALTER TABLE Album
ADD CONSTRAINT musical_genre_fk
FOREIGN KEY(musical_genre_id )
REFERENCES Musical_genre(musical_genre_id );

ALTER TABLE Album
ADD CONSTRAINT recording_artist_fk
FOREIGN KEY(recording_artist_id)
REFERENCES Recording_artist(recording_artist_id);

ALTER TABLE Song
ADD CONSTRAINT album_fk
FOREIGN KEY(album_id )
REFERENCES Album (album_id );
```



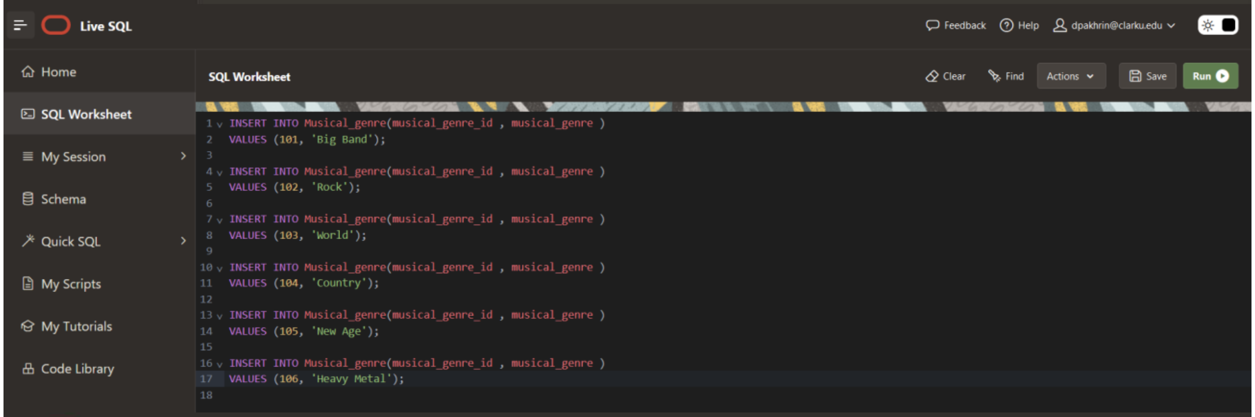
```
Table created.

Table created.

Table created.

Table created.

Table altered.

Table altered.

Table altered.
```

8. INSERT (Screen Shots)

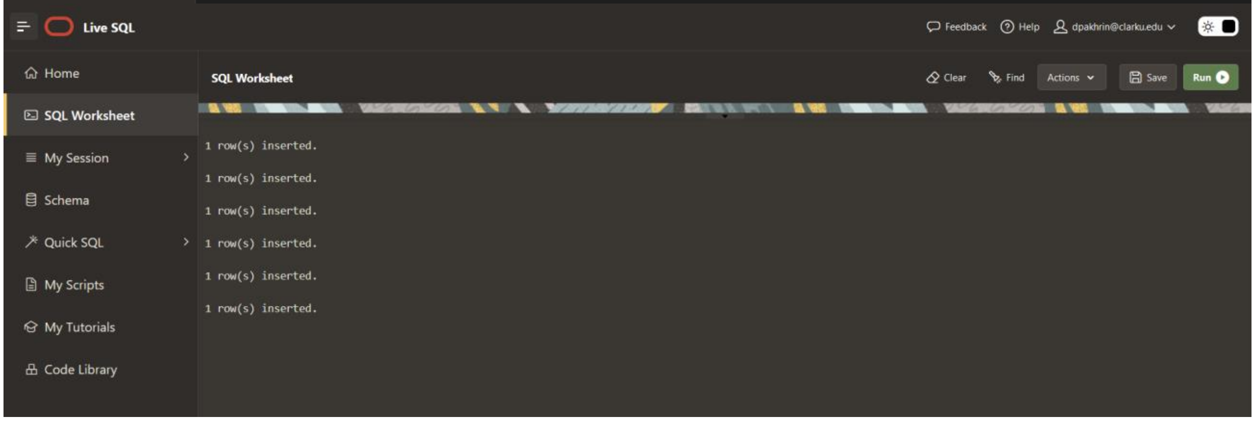


```
INSERT INTO Recording_artist(recording_artist_id , artist_name )
VALUES (01, 'April Wine');

INSERT INTO Recording_artist(recording_artist_id , artist_name )
VALUES (02, 'Enya');

INSERT INTO Recording_artist(recording_artist_id , artist_name )
VALUES (03, 'Elvis Costello');
```

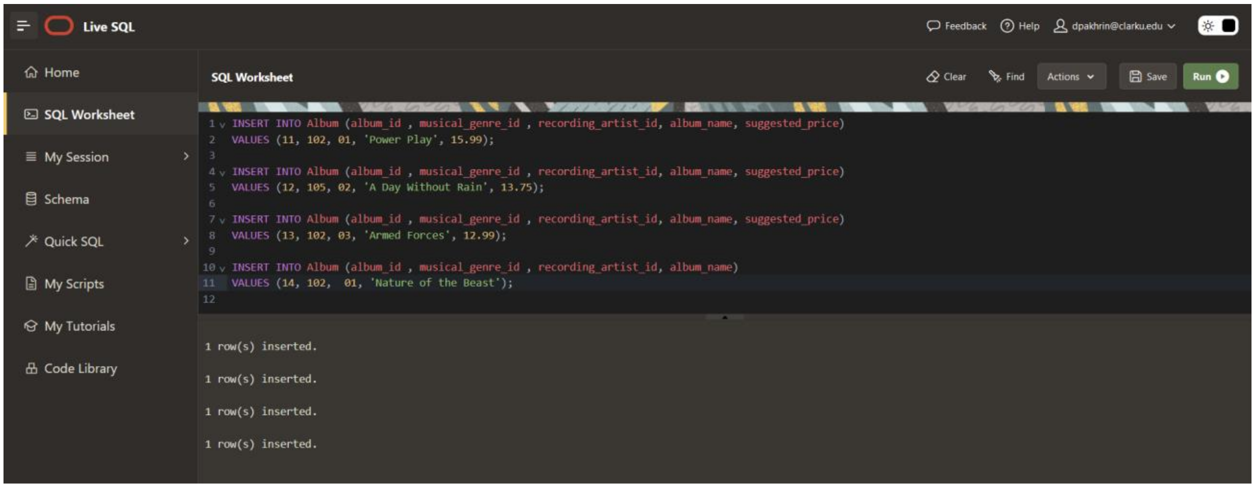
```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```



```
INSERT INTO Musical_genre(musical_genre_id , musical_genre )
VALUES (101, 'Big Band');

INSERT INTO Musical_genre(musical_genre_id , musical_genre )
VALUES (102, 'Rock');

INSERT INTO Musical_genre(musical_genre_id , musical_genre )
VALUES (103, 'World');

INSERT INTO Musical_genre(musical_genre_id , musical_genre )
VALUES (104, 'Country');

INSERT INTO Musical_genre(musical_genre_id , musical_genre )
VALUES (105, 'New Age');

INSERT INTO Musical_genre(musical_genre_id , musical_genre )
VALUES (106, 'Heavy Metal');
```

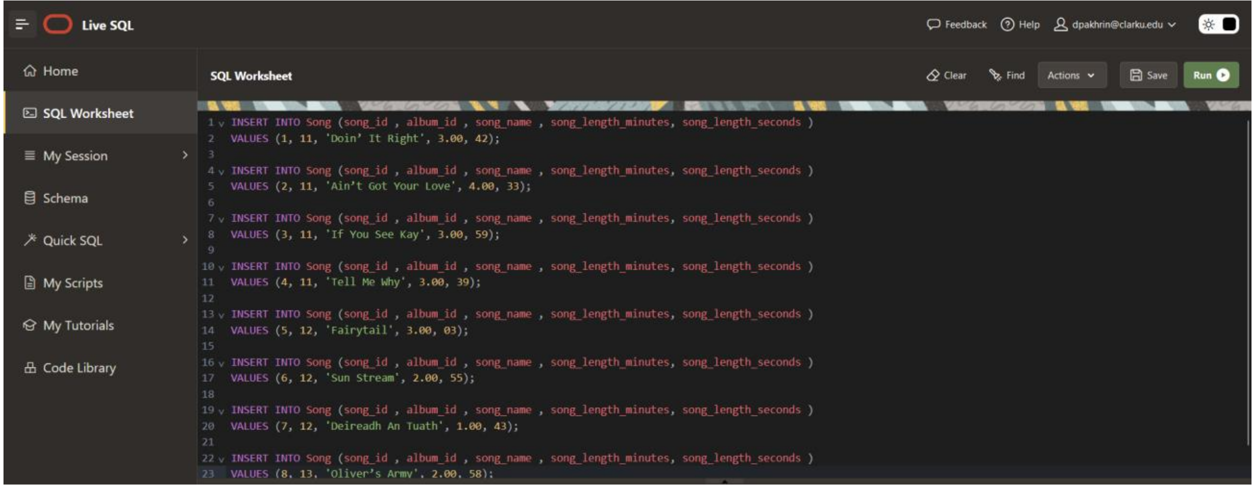


```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

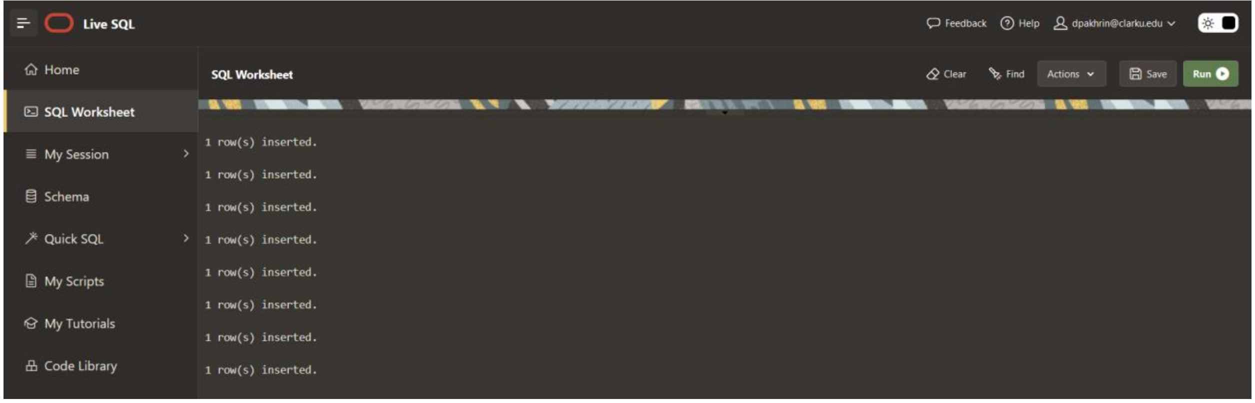


```sql
INSERT INTO Album (album_id , musical_genre_id , recording_artist_id, album_name, suggested_price)
VALUES (11, 102, 01, 'Power Play', 15.99);

INSERT INTO Album (album_id , musical_genre_id , recording_artist_id, album_name, suggested_price)
VALUES (12, 105, 02, 'A Day Without Rain', 13.75);

INSERT INTO Album (album_id , musical_genre_id , recording_artist_id, album_name, suggested_price)
VALUES (13, 102, 03, 'Armed Forces', 12.99);

INSERT INTO Album (album_id , musical_genre_id , recording_artist_id, album_name)
VALUES (14, 102,  01, 'Nature of the Beast');

```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

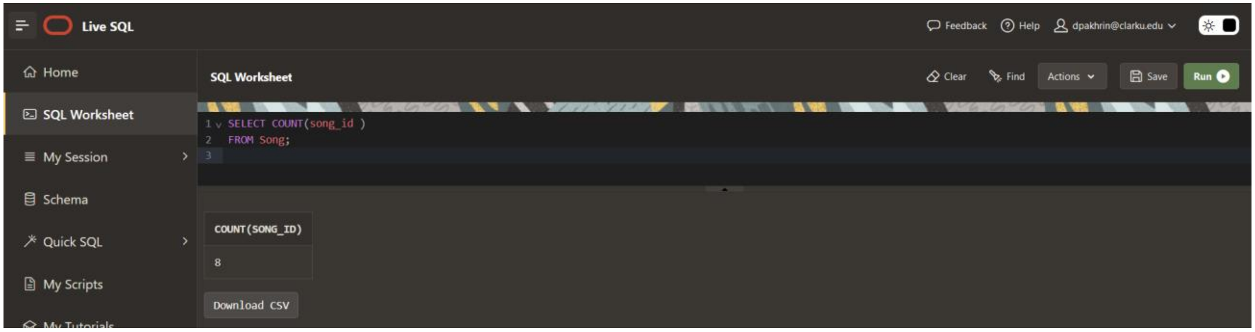


```sql
INSERT INTO Song (song_id , album_id , song_name , song_length_minutes, song_length_seconds )
VALUES (1, 11, 'Doin' It Right', 3.00, 42);

INSERT INTO Song (song_id , album_id , song_name , song_length_minutes, song_length_seconds )
VALUES (2, 11, 'Ain't Got Your Love', 4.00, 33);

INSERT INTO Song (song_id , album_id , song_name , song_length_minutes, song_length_seconds )
VALUES (3, 11, 'If You See Kay', 3.00, 59);

INSERT INTO Song (song_id , album_id , song_name , song_length_minutes, song_length_seconds )
VALUES (4, 11, 'Tell Me Why', 3.00, 39);

INSERT INTO Song (song_id , album_id , song_name , song_length_minutes, song_length_seconds )
VALUES (5, 12, 'Fairytail', 3.00, 03);

INSERT INTO Song (song_id , album_id , song_name , song_length_minutes, song_length_seconds )
VALUES (6, 12, 'Sun Stream', 2.00, 55);

INSERT INTO Song (song_id , album_id , song_name , song_length_minutes, song_length_seconds )
VALUES (7, 12, 'Deireadh An Tuath', 1.00, 43);

INSERT INTO Song (song_id , album_id , song_name , song_length_minutes, song_length_seconds )
VALUES (8, 13, 'Oliver's Army', 2.00, 58);
```



```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

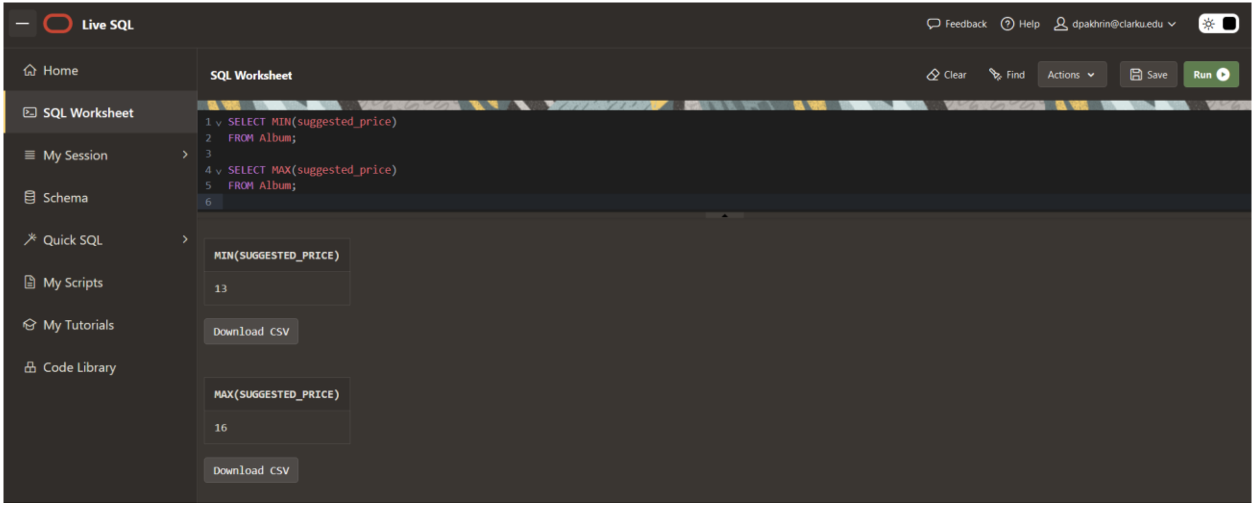
9. SELECT (Screen Shot)



```
SELECT COUNT(song_id )
FROM Song;
```

```
COUNT(SONG_ID)
8
```

10. SELECT (Screen Shot)



```
SELECT MIN(suggested_price)
FROM Album;

SELECT MAX(suggested_price)
FROM Album;
```

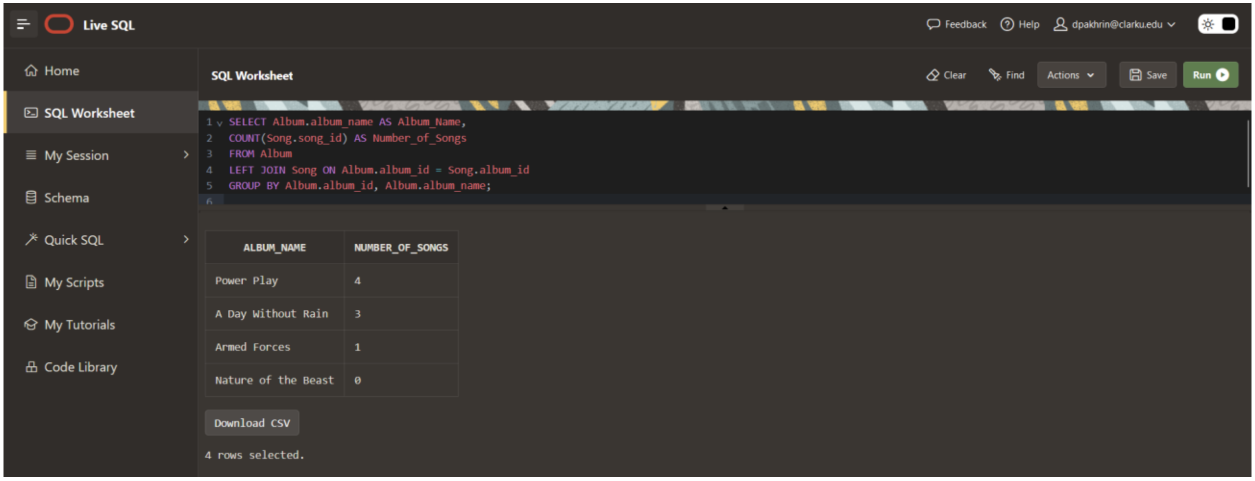
```
MIN(SUGGESTED_PRICE)
13
```

```
MAX(SUGGESTED_PRICE)
16
```

11. EXPLANATION

The album_name “Nature of the Beast” does not have a suggested_price in the table album. Since, the MIN an MAX function is calculated based on the suggested_price in the query 10, it excludes the entire row that doesn’t have the suggested price which is “Nature of the Beast” in our case.

12. SELECT (Screen Shot)

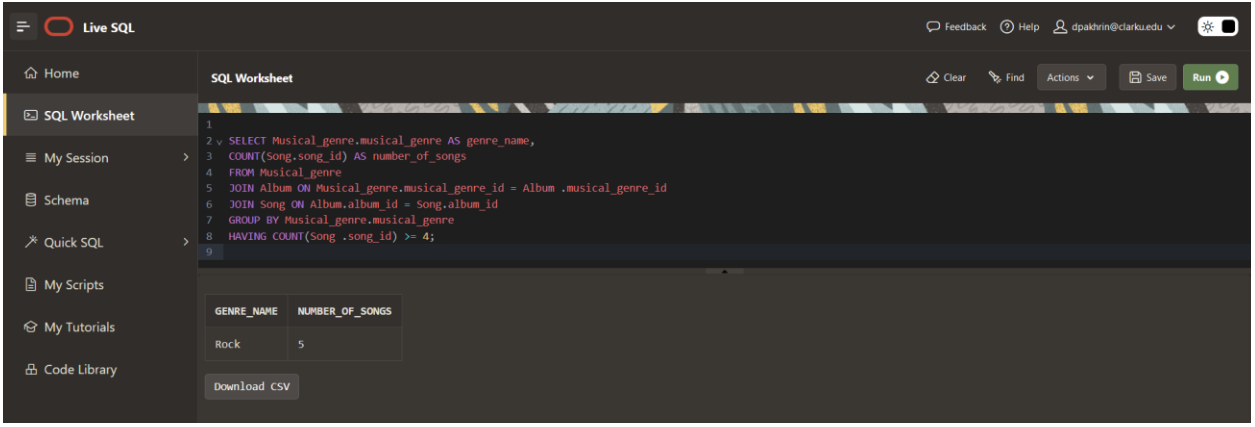


```sql
SELECT Album.album_name AS Album_Name,
COUNT(Song.song_id) AS Number_of_Songs
FROM Album
LEFT JOIN Song ON Album.album_id = Song.album_id
GROUP BY Album.album_id, Album.album_name;
```

ALBUM_NAME	NUMBER_OF_SONGS
Power Play	4
A Day Without Rain	3
Armed Forces	1
Nature of the Beast	0

4 rows selected.

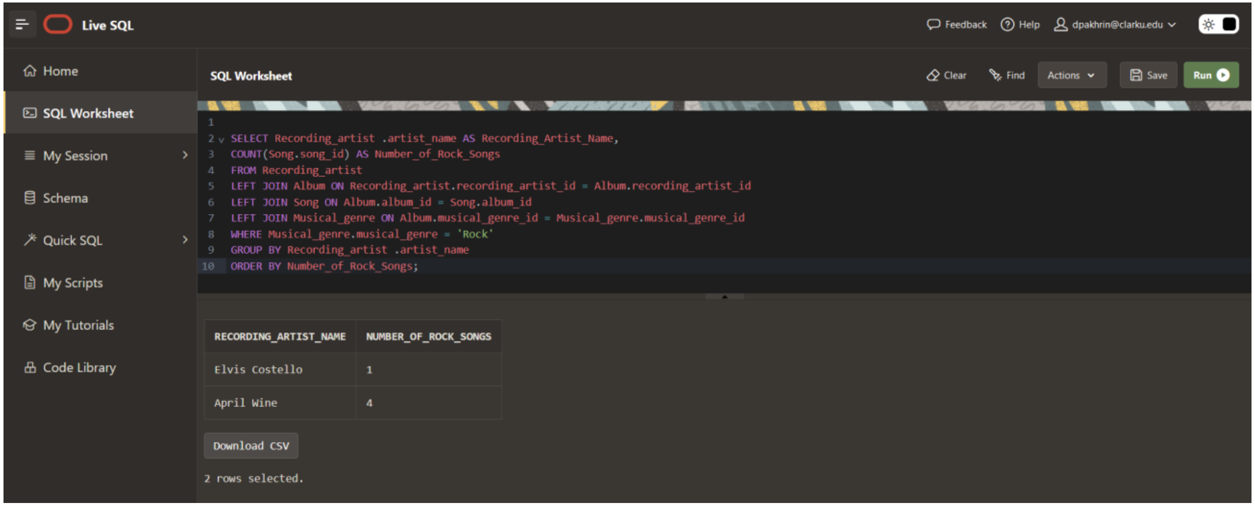
13. SELECT (Screen Shot)



```sql

SELECT Musical_genre.musical_genre AS genre_name,
COUNT(Song.song_id) AS number_of_songs
FROM Musical_genre
JOIN Album ON Musical_genre.musical_genre_id = Album .musical_genre_id
JOIN Song ON Album.album_id = Song.album_id
GROUP BY Musical_genre.musical_genre
HAVING COUNT(Song .song_id) >= 4;
```

GENRE_NAME	NUMBER_OF_SONGS
Rock	5

14. SELECT (Screen Shot)



```sql

SELECT Recording_artist .artist_name AS Recording_Artist_Name,
COUNT(Song.song_id) AS Number_of_Rock_Songs
FROM Recording_artist
LEFT JOIN Album ON Recording_artist.recording_artist_id = Album.recording_artist_id
LEFT JOIN Song ON Album.album_id = Song.album_id
LEFT JOIN Musical_genre ON Album.musical_genre_id = Musical_genre.musical_genre_id
WHERE Musical_genre.musical_genre = 'Rock'
GROUP BY Recording_artist .artist_name
ORDER BY Number_of_Rock_Songs;
```

RECORDING_ARTIST_NAME	NUMBER_OF_ROCK_SONGS
Elvis Costello	1
April Wine	4

2 rows selected.