RENEWABLE ENERGY TRACKING SYSTEM

DATABASE MANAGEMENT MINI-PROJECT

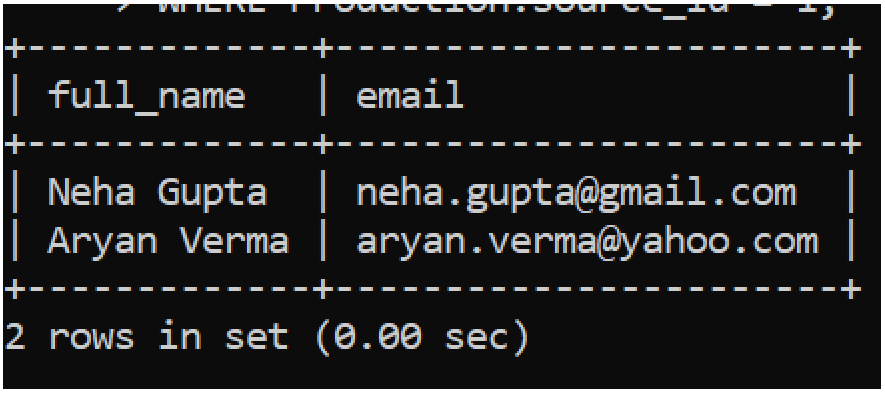
UCE2021422: Niharika Gurav
UCE2021423: Aaditee Iche
UCE2021424: Bhargavi Jadhav
UCE2021434: Hitali Khachane
SY COMP A (A2 Batch)

QUERIES:

1.Retrieve the name and email of all users who have consumed energy from a specific source.

***QUERY*:**

SELECT User.full_name, User.email
FROM User
INNER JOIN Consumption ON User.user_id = Consumption.user_id
INNER JOIN Production ON Consumption.production_id =
Production.production_id
WHERE Production.source_id = 1;

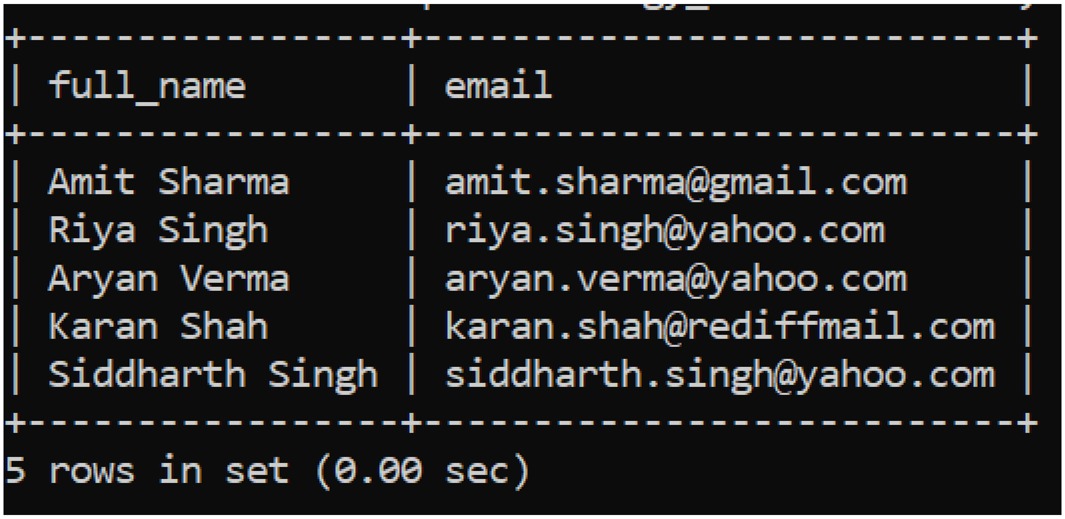


```
    -> WHERE Production.source_id = 1;
+-------------+-----------------------+
| full_name   | email                 |
+-------------+-----------------------+
| Neha Gupta  | neha.gupta@gmail.com  |
| Aryan Verma | aryan.verma@yahoo.com |
+-------------+-----------------------+
2 rows in set (0.00 sec)
```

2.Retrieve the names and emails of all users who have consumed more than 100 units of energy.

QUERY:

SELECT User.full_name, User.email
FROM User
INNER JOIN Consumption ON User.user_id = Consumption.user_id
INNER JOIN Production ON Consumption.production_id = Production.production_id
WHERE Consumption.energy_consumed > 100;

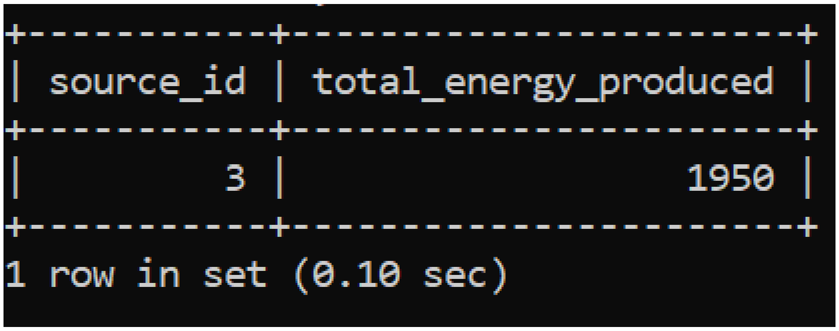


```
+------------------+---------------------------+
| full_name        | email                     |
+------------------+---------------------------+
| Amit Sharma      | amit.sharma@gmail.com     |
| Riya Singh       | riya.singh@yahoo.com      |
| Aryan Verma      | aryan.verma@yahoo.com     |
| Karan Shah       | karan.shah@rediffmail.com |
| Siddharth Singh  | siddharth.singh@yahoo.com |
+------------------+---------------------------+
5 rows in set (0.00 sec)
```

3.Retrieve the top source with the highest amount of energy produced.

QUERY:

SELECT Production.source_id,
    SUM(Production.energy_produced) as total_energy_produced
FROM Production
GROUP BY Production.source_id
ORDER BY total_energy_produced DESC
LIMIT 1;

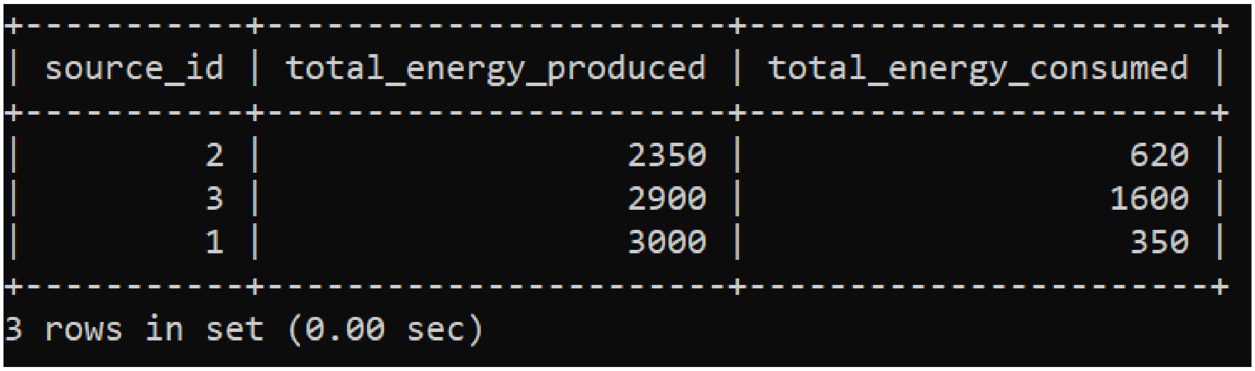


```
+-----------+-----------------------+
| source_id | total_energy_produced |
+-----------+-----------------------+
|         3 |                  1950 |
+-----------+-----------------------+
1 row in set (0.10 sec)
```

4.Retrieve the total amount of energy produced and consumed for each source.

QUERY:

```
SELECT Production.source_id,
      SUM(Production.energy_produced) as total_energy_produced,
      SUM(Consumption.energy_consumed) as total_energy_consumed
FROM Production
LEFT JOIN Consumption ON Production.production_id =
Consumption.production_id
GROUP BY Production.source_id;
```

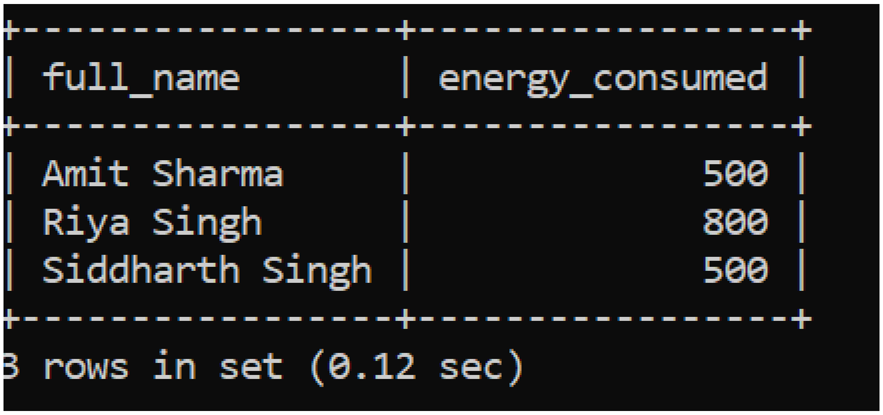


```
+-----------+----------------------+-----------------------+
| source_id | total_energy_produced | total_energy_consumed |
+-----------+----------------------+-----------------------+
|         2 |                  2350 |                   620 |
|         3 |                  2900 |                  1600 |
|         1 |                  3000 |                   350 |
+-----------+----------------------+-----------------------+
3 rows in set (0.00 sec)
```

5.Retrieve all users who have consumed more energy than the average energy consumption:

QUERY:

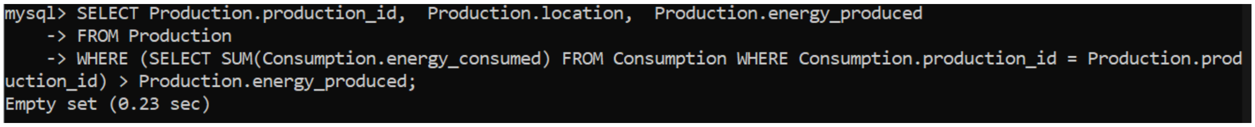
```
SELECT User.full_name, Consumption.energy_consumed FROM User
INNER JOIN Consumption ON User.user_id = Consumption.user_id
WHERE Consumption.energy_consumed > (SELECT AVG(energy_consumed)
FROM Consumption );
```



```
+-----------------+-----------------+
| full_name       | energy_consumed |
+-----------------+-----------------+
| Amit Sharma     |             500 |
| Riya Singh      |             800 |
| Siddharth Singh |             500 |
+-----------------+-----------------+
3 rows in set (0.12 sec)
```

6.Retrieve all productions where the energy consumed by users is higher than the energy produced.

QUERY:

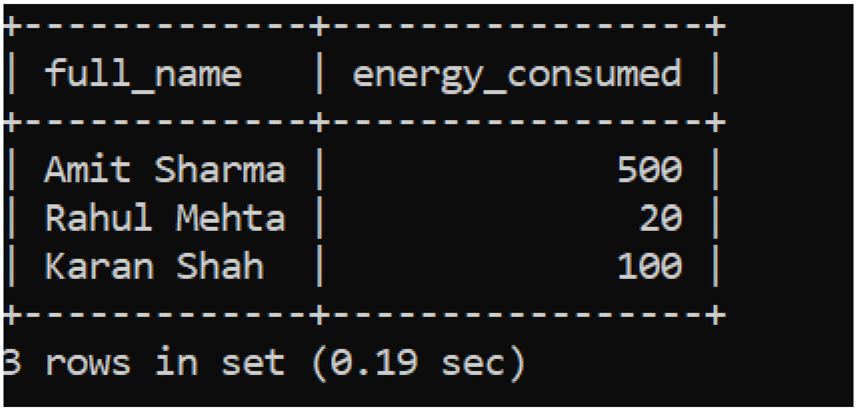


```
mysql> SELECT Production.production_id,  Production.location,  Production.energy_produced
    -> FROM Production
    -> WHERE (SELECT SUM(Consumption.energy_consumed) FROM Consumption WHERE Consumption.production_id = Production.prod
uction_id) > Production.energy_produced;
Empty set (0.23 sec)
```

7.Retrieve all users who have consumed energy from a specific source type:

QUERY:

SELECT User.full__name, Consumption.energy__consumed
FROM User INNER JOIN Consumption ON User.user__id = Consumption.user__id
WHERE Consumption.production__id IN (SELECT production__id FROM Production WHERE source__id IN (SELECT source__id FROM Source WHERE type = ‘Solar’));

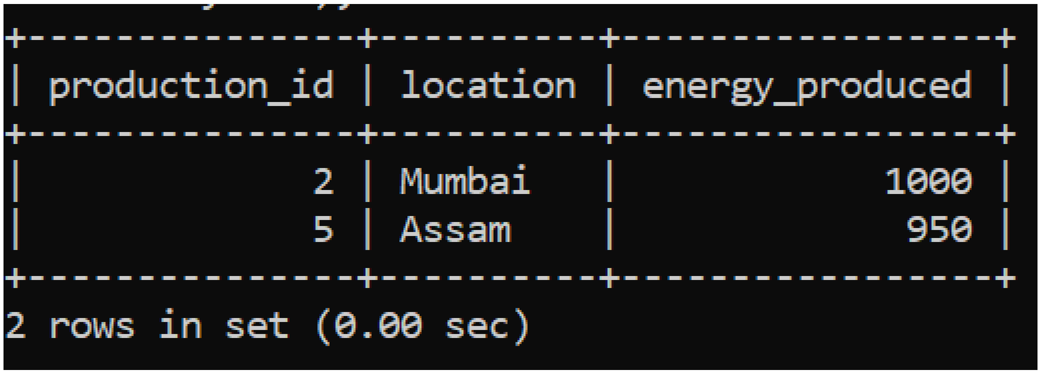


```
+-------------+-----------------+
| full_name   | energy_consumed |
+-------------+-----------------+
| Amit Sharma |             500 |
| Rahul Mehta |              20 |
| Karan Shah  |             100 |
+-------------+-----------------+
3 rows in set (0.19 sec)
```

8.Retrieve all productions for a specific source type:

QUERY:

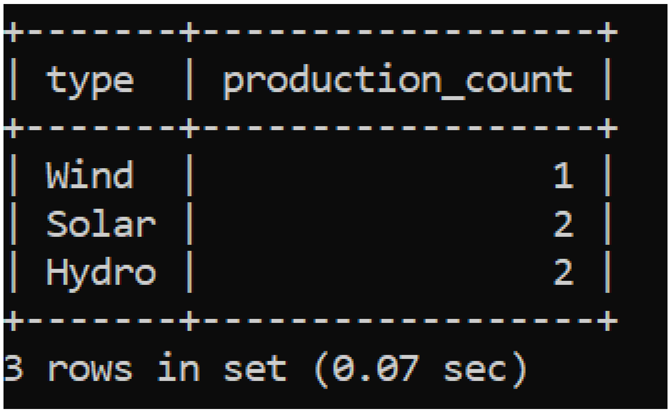
SELECT Production.production__id, Production.location, Production.energy__produced
FROM Production
WHERE Production.source__id IN (SELECT source__id FROM Source WHERE type = 'Hydro’');



```
+---------------+----------+-----------------+
| production_id | location | energy_produced |
+---------------+----------+-----------------+
|             2 | Mumbai   |            1000 |
|             5 | Assam    |             950 |
+---------------+----------+-----------------+
2 rows in set (0.00 sec)
```

9.Retrieve the source type and the corresponding count of productions for each source type.

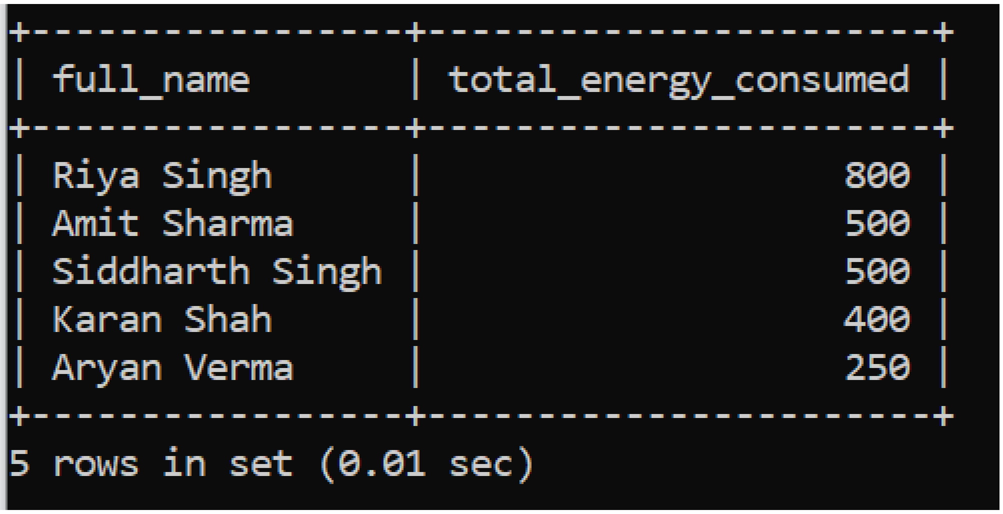
QUERY:
SELECT Source.type, COUNT(Production.production__id) AS production__count
FROM Source
LEFT JOIN Production ON Source.source__id = Production.source__id
GROUP BY Source.type;



```
+-------+------------------+
| type  | production_count |
+-------+------------------+
| Wind  |                1 |
| Solar |                2 |
| Hydro |                2 |
+-------+------------------+
3 rows in set (0.07 sec)
```

10. Retrieve the top 5 users who have consumed the highest amount of energy.

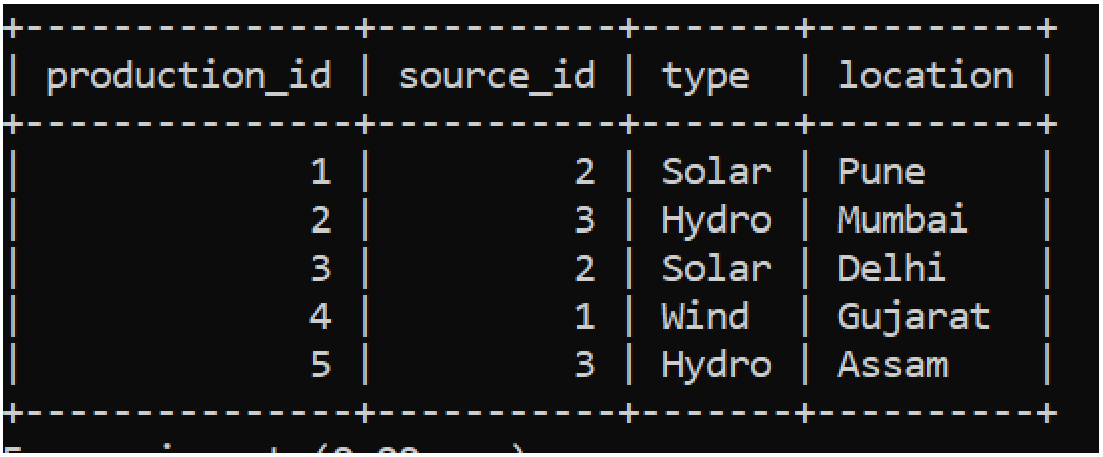
QUERY:
SELECT User.full__name, SUM(Consumption.energy__consumed) AS total__energy__consumed
FROM User INNER JOIN Consumption ON User.user__id = Consumption.user__id
GROUP BY User.full__name
ORDER BY total__energy__consumed DESC
LIMIT 5;



```
+------------------+------------------------+
| full_name        | total_energy_consumed |
+------------------+------------------------+
| Riya Singh       |                    800 |
| Amit Sharma      |                    500 |
| Siddharth Singh |                    500 |
| Karan Shah       |                    400 |
| Aryan Verma      |                    250 |
+------------------+------------------------+
5 rows in set (0.01 sec)
```

11. Retrieve all productions along with their corresponding source and location.

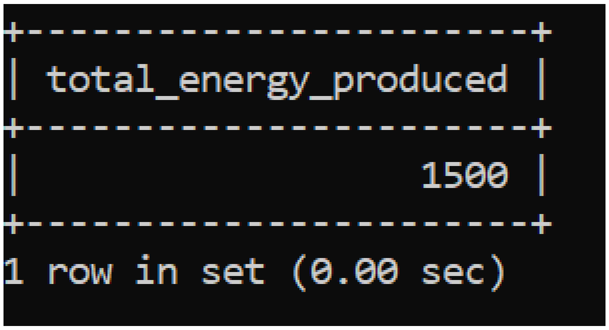
QUERY:
SELECT Production.production_id, Source.source_id, Source.type,
Production.location
FROM Production INNER
JOIN Source ON Production.source_id = Source.source_id;



```
+---------------+-----------+-------+----------+
| production_id | source_id | type  | location |
+---------------+-----------+-------+----------+
|             1 |         2 | Solar | Pune     |
|             2 |         3 | Hydro | Mumbai   |
|             3 |         2 | Solar | Delhi    |
|             4 |         1 | Wind  | Gujarat  |
|             5 |         3 | Hydro | Assam    |
+---------------+-----------+-------+----------+
5 rows in set (0.00 sec)
```

12. Retrieve the total energy produced by a specific source.

QUERY:
SELECT SUM(energy_produced) AS total_energy_produced
FROM Production
WHERE type=’ Wind’;



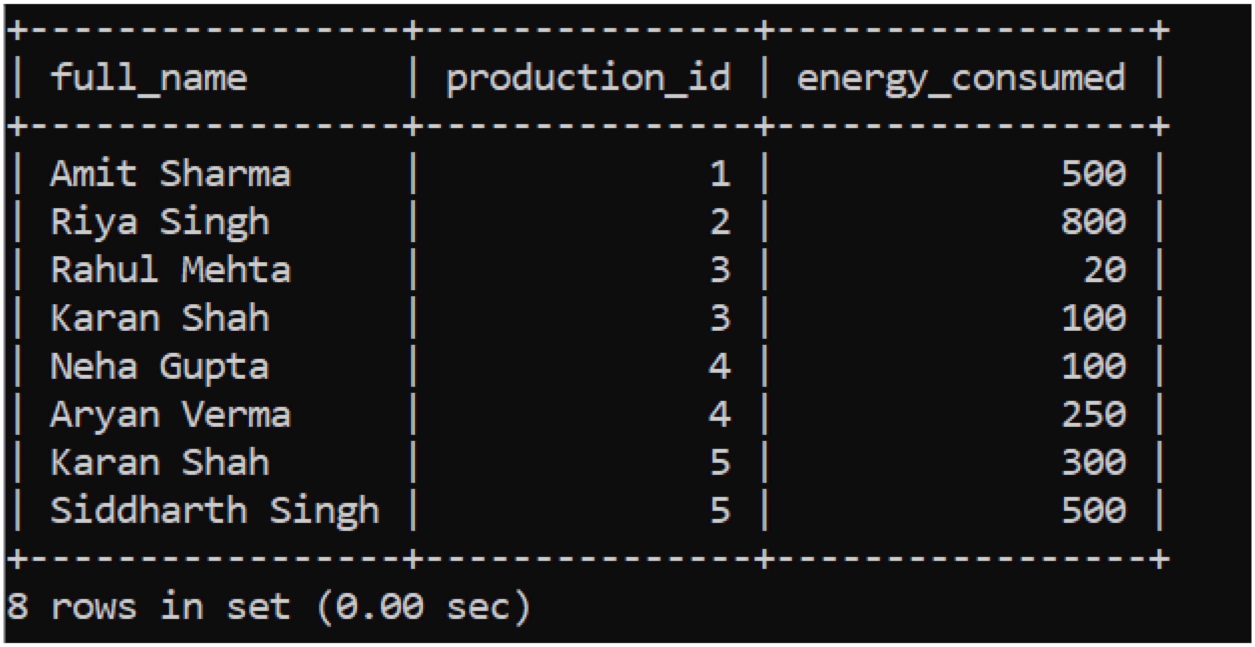
```
+----------------------+
| total_energy_produced |
+----------------------+
|                  1500 |
+----------------------+
1 row in set (0.00 sec)
```

13.Retrieve all users with their corresponding billing information:

14.Retrieve all users and their corresponding productions along with energy consumed.

QUERY:

SELECT User.full_name, Production.production_id, Consumption.energy_consumed
FROM User INNER JOIN Consumption ON User.user_id = Consumption.user_id
INNER JOIN Production ON Consumption.production_id = Production.production_id;

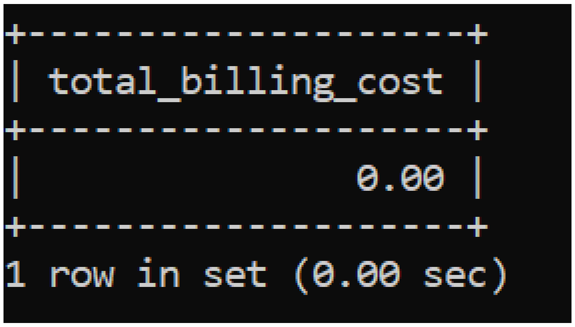


```
+-----------------+---------------+-----------------+
| full_name       | production_id | energy_consumed |
+-----------------+---------------+-----------------+
| Amit Sharma     |             1 |             500 |
| Riya Singh      |             2 |             800 |
| Rahul Mehta     |             3 |              20 |
| Karan Shah      |             3 |             100 |
| Neha Gupta      |             4 |             100 |
| Aryan Verma     |             4 |             250 |
| Karan Shah      |             5 |             300 |
| Siddharth Singh |             5 |             500 |
+-----------------+---------------+-----------------+
8 rows in set (0.00 sec)
```

15. Retrieve the total cost of billing for a specific user

QUERY:

```
SELECT SUM(cost) AS total_billing_cost
FROM billing
WHERE user_id =2;
```



```
+--------------------+
| total_billing_cost |
+--------------------+
|               0.00 |
+--------------------+
1 row in set (0.00 sec)
```

16. Trigger to automatically calculate the cost for a billing record based on the energy consumed and insert it into the billing table.

TRIGGER:

```
 DELIMITER //
mysql> CREATE TRIGGER calculate_billing_cost
   -> BEFORE INSERT ON Billing
   -> FOR EACH ROW
   -> BEGIN
   ->     DECLARE energy_price DECIMAL(10, 2);
   ->     SET energy_price = 0.10;
   ->     SET NEW.cost = NEW.Consumption.energy_consumed * energy_price;
   -> END;
   -> //
```

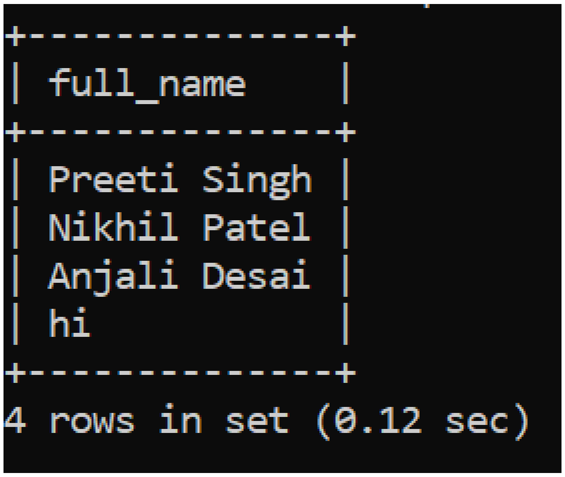
17.

```
INSERT INTO Billing (user_id, production_id) VALUES (7, 3);
```

18. Retrieve the names of all users who have not consumed any energy.

QUERY:

```
SELECT User.full_name
FROM User
LEFT JOIN Consumption ON User.user_id = Consumption.user_id
WHERE Consumption.user_id IS NULL;
```



```
+--------------+
| full_name    |
+--------------+
| Preeti Singh |
| Nikhil Patel |
| Anjali Desai |
| hi           |
+--------------+
4 rows in set (0.12 sec)
```

---

19. Create an index on the production table for faster queries.

INDEX:

```
CREATE INDEX production_source_id ON Production (source_id);
```

20.A trigger that automatically deletes a row from the Billing table when the corresponding row is deleted from the Consumption table.

TRIGGER:

```
DELIMITER //
CREATE TRIGGER delete_billing
AFTER DELETE ON Consumption
FOR EACH ROW
BEGIN
  DELETE FROM Billing WHERE user_id = OLD.user_id AND production_id = 
OLD.production_id;
END; //
```