

Advance Views

View Name: STATES_HEALTH_INFO

This view calculates a state residents' average steps counts, average distance traveled, and calories burned, and compares it with standard recommendations to categorize the state as a healthy state or unhealthy state.

SQL Code:

```
USE fitbit_db;
```

```
CREATE OR REPLACE VIEW states_health_info AS
SELECT
  AVG(TotalSteps) AS Avgerage_steps,
  AVG(TotalDistance) AS Average_distance,
  AVG(Calories) AS Average_calories,
  IF ((AVG(TotalSteps) > 5000 && AVG(TotalDistance) > 5 && AVG(Calories) > 2000) ,
  "Healthy" , "Not Healthy") AS Health_status
FROM dailyactivity
WHERE Id IN
( SELECT Id
  FROM State_User_Info
  WHERE state = "Utah"
);

SELECT
  *
FROM
  states_health_info;
```

Results:

```

1 • USE fitbit_db;
2
3 • CREATE OR REPLACE VIEW states_health_info AS
4     SELECT
5         AVG(TotalSteps) AS Average_steps,
6         AVG(TotalDistance) AS Average_distance,
7         AVG(Calories) AS Average_calories,
8         IF ((AVG(TotalSteps) > 5000 && AVG(TotalDistance) > 5 && AVG(Calories) > 2
9         FROM dailyactivity
10        WHERE Id IN
11        ( SELECT Id
12          FROM State_User_Info
13           WHERE state = "Utah"
14        );
15
16 • SELECT
17     *
18 FROM
19     states_health_info;
20

```

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

	Average_steps	Average_distance	Average_calories	Health_status
▶	6574.8869	4.466812326365042	1952.7943	Not Healthy

states_health_info 25 ×

Output



Action Output

	#	Time	Action	Message
✔	1	15:34:24	USE fitbit_db	0 row(s) affected
⚠	2	15:34:24	CREATE OR REPLACE VIEW states_health_info AS SELECT AVG...	0 row(s) affected, 2 warning(s)
✔	3	15:34:24	SELECT * FROM states_health_info LIMIT 0, 400	1 row(s) returned

View Name: STATES_HEALTH_INFO

This view returns state wise userId, average weight and average age.

SQL Code:

```
use fitbit_db;
```

```
CREATE OR REPLACE VIEW State_User_Info AS
SELECT
    Id AS UserId,
    state AS State,
    AVG(Weight) OVER (PARTITION BY state) AS Avgrage_weight,
    AVG(age) OVER (PARTITION BY state) AS Average_age
FROM
    userdetails
    left JOIN
    addressinfo ON userdetails.AddressID = addressinfo.AddressID
order by state, Id;

SELECT
    *
FROM
    States_Users_Info;
```

Results:

States_Health_Info_View

State_User_Info_View

State_Health_Rank

Limit to 400 rows

```

1 • use fitbit_db;
2
3 CREATE OR REPLACE VIEW State_User_Info AS
4 SELECT
5     Id AS UserId,
6     state AS State,
7     AVG(Weight) OVER (PARTITION BY state) AS Avgrage_weight,
8     AVG(age) OVER (PARTITION BY state) AS Average_age
9 FROM
10     userdetails
11     left JOIN
12     addressinfo ON userdetails.AddressID = addressinfo.AddressID
13     order by state, Id;

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	UserId	state	Avgrage_weight	Average_age
▶	1503960366	Alabama	65.0000	40.0000
	4319703636	Alabama	65.0000	40.0000
	4319703592	Alaska	65.1667	35.8333
	4319703619	Alaska	65.1667	35.8333
	4319703674	Alaska	65.1667	35.8333
	4319703690	Alaska	65.1667	35.8333
	4319703707	Alaska	65.1667	35.8333
	4319703724	Alaska	65.1667	35.8333
	4319703602	Arizona	48.2500	46.7500
	4319703643	Arizona	48.2500	46.7500
	4319703646	Arizona	48.2500	46.7500
	4319703661	Arizona	48.2500	46.7500
	4319703687	Arkansas	73.3333	36.0000
	4319703715	Arkansas	73.3333	36.0000
	4319703732	Arkansas	73.3333	36.0000

States_Users_Info 16 x

Output

Action Output

#	Time	Action	Message
✓ 1	20:37:06	use fitbit_db	0 row(s) affected
✓ 2	20:37:06	CREATE OR REPLACE VIEW State_User_Info AS SELECT I...	0 row(s) affected
✓ 3	20:37:06	SELECT * FROM States_Users_Info LIMIT 0, 400	177 row(s) returned

View Name: USER_HEALTH_RANK

This view assigns the rank to the user based on the avg calories burn.

SQL Code:

```
use fitbit_db;
```

```
CREATE OR REPLACE VIEW USER_HEALTH_RANK AS  
WITH USER_STATES as
```

```
( SELECT  
    Id AS UserId,  
    state AS State  
FROM userdetails  
    left JOIN  
    addressinfo ON userdetails.AddressID = addressinfo.AddressID ),
```

```
USER_HEALTH_INFO as
```

```
( Select DENSE_RANK() OVER (ORDER BY avg(Calories)) AS 'health_rank',  
    Id,  
        round(avg(Calories), 2) as Average_Calories_Burn,  
    TotalMinutesAsleep,  
    Count(*) as No_Of_Records  
from dailyactivity  
left join sleeptime  
on dailyactivity.SleepId = sleeptime.SleepId  
group by Id)
```

```
SELECT
```

```
*
```

```
FROM USER_HEALTH_INFO join
```

```
    USER_STATES
```

```
    ON USER_STATES.UserId = USER_HEALTH_INFO.Id
```

```
ORDER BY health_rank;
```

```
SELECT health_rank, Id, Average_Calories_Burn, state  
FROM USER_HEALTH_RANK;
```

Results:

States_Health_Info_View

State_User_Info_View

State_Health_Rank

SQL File 42*

Limit to 400 rows

```

14         Id,
15         round(avg(Calories), 2) as Average_Calories_Burn,
16         TotalMinutesAsleep,
17         Count(*) as No_Of_Records
18     from dailyactivity
19     left join sleeptime
20     on dailyactivity.SleepId = sleeptime.SleepId
21     group by Id)
22
23     SELECT
24         *
25     FROM USER_HEALTH_INFO join
26         USER_STATES
27         ON USER_STATES.UserId = USER_HEALTH_INFO.Id
28     ORDER BY health_rank;
29
30 • SELECT health_rank, Id, Average_Calories_Burn, state
31     FROM USER_HEALTH_RANK;

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content: IA

	health_rank	Id	Average_Calories_Burn	State
▶	1	1624580081	1483.35	Maryland
	2	3977333714	1513.67	New Hampshire
	3	2026352035	1540.65	Utah
	4	1844505072	1573.48	Montana
	5	2320127002	1724.16	Rhode Island
	6	1503960366	1816.42	Alabama
	7	2873212765	1916.97	Delaware
	8	3372868164	1933.10	California
	9	4319703577	1964.63	South Dakota
	10	4057192912	1973.75	Kansas
	11	2347167796	2043.44	Ohio
	12	1927972279	2172.81	New Jersey
	13	4020332650	2385.81	Louisiana

USER_HEALTH_RANK 14

Output

Action Output

#	Time	Action	Message
✓ 1	20:51:41	SELECT health_rank, Id, Average_Calories_Burn, state FROM USER...	15 row(s) returned