## METHODOLOGY
## BRIGHT TV

The dataset contained two columns labeled 'userid'. To ensure data consistency and avoid redundancy, one of the duplicate 'userid' columns was removed

```
alter table viewership
drop column "userid";
```

To determine the total number of unique viewers, user_profile and viewership table were joined using the common key userid. The inner join ensured that only records present in both tables were considered, thereby including only valid user–viewership relationships. The function count (distinct userid) was then applied to eliminate duplicate entries and return the total count of distinct users who engaged in viewership activities.

```
select count (distinct userid) as total_viewers
from user_profile as a
inner join viewership as b
on a.userid = b."UserID";
```

| | TOTAL_VIEWERS |
|---|---|
| 1 | 4386 |

The query joins the user_profile and viewership tables using the userid field to include only users with viewership records, groups the data by province and counts the number of viewers in each province. The results are arranged in descending order to show which provinces have the highest number of viewers.

```
select province,
       count (userid) as total_viewers
from user_profile as a
inner join viewership as b
on a.userid = b."UserID"
group by province
order by total_viewers desc;
```

| | PROVINCE | TOTAL_VIEWERS |
|---|---|---|
| 1 | Gauteng | 3654 |
| 2 | Western Cape | 1845 |
| 3 | Kwazulu Natal | 1001 |
| 4 | Mpumalanga | 918 |
| 5 | Limpopo | 763 |
| 6 | Eastern Cape | 690 |
| 7 | North West | 344 |
| 8 | Free State | 292 |
| 9 | None | 263 |
| 10 | Northern Cape | 230 |

The query joins the user_profile and viewership tables using the userid field to include only users with valid viewership records. It then groups the data by channel2 and counts the number of viewers for each

channel. Finally, the results are sorted in descending order to highlight the channels with the highest number of viewers.

```
select channel2,
       count (userid) as total_viewers
from user_profile as a
inner join viewership as b
on a.userid = b."UserID"
group by channel2
order by total_viewers desc;
```

| | CHANNEL2 | TOTAL_VIEWERS |
|---|---|---|
| 1 | Supersport Live Events | 1638 |
| 2 | ICC Cricket World Cup 2011 | 1465 |
| 3 | Channel O | 1050 |
| 4 | Trace TV | 952 |
| 5 | SuperSport Blitz | 896 |
| 6 | Africa Magic | 859 |
| 7 | Cartoon Network | 793 |
| 8 | Boomerang | 714 |
| 9 | CNN | 505 |
| 10 | E! Entertainment | 367 |
| 11 | SawSee | 251 |
| 12 | M-Net | 116 |
| 13 | Vuzu | 111 |
| 14 | DStv Events 1 | 107 |
| 15 | Break in transmission | 66 |
| 16 | kykNET | 45 |
| 17 | MK | 32 |

The query joins the user_profile and viewership tables using the userid field to include only users with viewership records. It then groups the data by race and counts the total number of viewers in each racial group. The results are sorted in descending order to show which racial groups have the highest number of viewers.

| | RACE | TOTAL_VIEWERS |
|---|---|---|
| 1 | black | 4331 |
| 2 | coloured | 1633 |
| 3 | indian_asian | 1575 |
| 4 | white | 1292 |
| 5 | None | 1057 |
| 6 | other | 102 |

Joined the user_profile and viewership tables using the userid field to include only users with viewership records. Then grouped the data by race and gender and counts the total number of viewers for each group. The results are sorted in descending order to show which race and gender groups have the highest number of viewers

```sql
select race,
       gender,
       count (*) as total_viewers
from user_profile as a
inner join viewership as b
on a.userid = b."UserID"
group by race ,
         gender
order by total_viewers desc;
```

| | RACE | GENDER | TOTAL_VIEWERS |
|---|---|---|---|
| 1 | black | male | 3830 |
| 2 | coloured | male | 1498 |
| 3 | indian_asian | male | 1483 |
| 4 | white | male | 1177 |
| 5 | None | male | 670 |
| 6 | black | female | 501 |
| 7 | None | None | 262 |
| 8 | coloured | female | 135 |
| 9 | None | female | 125 |
| 10 | white | female | 115 |
| 11 | other | male | 95 |
| 12 | indian_asian | female | 92 |
| 13 | null | male | 8 |
| 14 | other | female | 7 |
| 15 | null | female | 2 |

Joined user_profile and viewership tables using the userid field to include only users with valid viewership records. It then groups users into defined age buckets (e.g., kids, teenager, youth, adult) using a case statement. For each age group, the total number of viewers is counted with count (userid). The results are sorted in descending order to show which age group has the most viewers.

```sql
select count (userid) as total_viewers,
       case
         when age = 0 then 'not applicable'
         when age between 1 and 12 then 'kids'
         when age between 13 and 19 then 'teenager'
         when age between 20 and 35 then 'youth'
         when age between 36 and 50 then 'adult'
         when age between 51 and 65 then 'Mature adult'
         else 'retired'
        end as age_bucket
from user_profile as a
inner join viewership as b
on a.userid = b."UserID"
group by age_bucket
order by total_viewers desc;
```

| | TOTAL_VIEWERS | AGE_BUCKET |
|---|---|---|
| 1 | 5733 | youth |
| 2 | 2972 | adult |
| 3 | 452 | Mature adult |
| 4 | 436 | teenager |
| 5 | 260 | not applicable |
| 6 | 99 | kids |
| 7 | 48 | retired |

Joined the viewership and user_profile tables using the userid field. Then extract the day of the week from the recorddate2 field and groups the data by each day. For every day, it calculates the percentage of total viewers by dividing the daily count by the overall number of viewership records and rounding the result to two decimal places. The days are ordered in descending order to show which days of the week had the highest percentage of viewers.

```
select dayname (a.recorddate2) as days_of_the_week,
     round( count (*) * 100.0 / (select count(*) from viewership),2) as percentage_of_viewers
from viewership a
join user_profile b
on a."UserID" = b.userid
group by days_of_the_week
order by percentage_of_viewers desc;
```

| | DAYS_OF_THE_WEEK | PERCENTAGE_OF_VIEWERS |
|---|---|---|
| 1 | Sat | 16.55 |
| 2 | Fri | 16.42 |
| 3 | Wed | 15.39 |
| 4 | Thu | 14.71 |
| 5 | Sun | 13.98 |
| 6 | Tue | 13.21 |
| 7 | Mon | 9.74 |

The query converts timestamps from UTC to South African Time and groups viewers into time-of-day buckets: Night, Morning, Afternoon, and Evening. Then counts the total viewers in each bucket and orders the results from most to least viewers.

```
select
    count(userid) as total_viewers,
    case
        when extract(hour from RecordDate2 + interval '2 hours') between 0 and 5 then 'Night'
        when extract(hour from RecordDate2 + interval '2 hours') between 6 and 11 then 'Morning'
        when extract(hour from RecordDate2 + interval '2 hours') between 12 and 17 then 'Afternoon'
        else 'Evening'
    end as time_bucket
from user_profile AS a
inner join viewership as b
on a.userid = b."UserID"
group by time_bucket
order by total_viewers desc;
```

| | TOTAL_VIEWERS | TIME_BUCKET |
|---|---|---|
| 1 | 3734 | Afternoon |
| 2 | 3344 | Evening |
| 3 | 2361 | Morning |
| 4 | 561 | Night |