**SQL\_HEIST ASSIGNMENT CLASS 6**

**select \* from youtube;**

**--Q1**

**select channeltype, sum(revenuelastmonth) from youtube group by channeltype**

**having channeltype = 'Gaming';**

**--Q2**

**select country , sum(revenuelastmonth) as total from youtube group by country ;**

**--Q3**

**select country ,channeltype, sum(revenuelastmonth) from youtube**

**group by country, channeltype having channeltype = 'Gaming' ;**

**--Q4**

**select country, sum(revenuelastmonth) from youtube group by country having sum(revenuelastmonth) > 200000;**

**--Q4**

**select country , sum(revenuelastmonth) as total from youtube**

**where channeltype = 'Gaming' and subscribers > 1000000 group by country;**

**--Q5**

**select country , sum(revenuelastmonth)**

**from youtube**

**where channeltype = 'Gaming'**

**group by country**

**having sum(revenuelastmonth) > 30000;**

**--Q6**

**select country , sum(revenuelastmonth) , count(influencerid) , avg(revenuelastmonth)**

**from youtube group by country;**

**--Q7**

**SELECT Country, SUM(revenuelastmonth) AS TotalRevenue, AVG(revenuelastmonth) AS AvgRevenuePerInfluencer**

**FROM youtube**

**GROUP BY country**

**HAVING SUM(revenuelastmonth) > 150000 AND AVG(revenuelastmonth) > 20000;**

**--Q8**

**select channeltype , sum(revenuelastmonth) as total from youtube**

**group by channeltype order by total desc limit 1;**

**--Q9 Wrote a sub-query to find the 50% of max revenue by an influencer the used its output in having clause**

**-- to comapre it with avg revenue of ecah influencer .**

**SELECT channeltype, AVG(revenuelastmonth) AS AvgRevenuePerInfluencer**

**FROM youtube**

**GROUP BY channeltype**

**having avg(revenuelastmonth) > (select sum(revenuelastmonth)/2 as total from youtube**

**group by influencername order by total DESC limit 1);**

**-- Q10 Wrote a subquery to find 10% of totalviews then wrote an outer query to compare the totalviews of**

**-- each channeltype with the 10% of totalviews .**

**select channeltype , sum(avgviewspervideo) from youtube group by channeltype**

**having sum(avgviewspervideo) > (select (sum(avgviewspervideo)/100)\*10 from youtube);**

**-- query to find 10% of total views**

**select (sum(avgviewspervideo)/100)\*10 from youtube;**

**-- Query to find totalviews of each channeltype**

**select channeltype , sum(avgviewspervideo) from youtube group by channeltype;**

**DATASENSE PRACTICE QUESTIONS**

**Q1 . select festival\_name , country , ticket\_type from festivaldata ;**

**Q2. select Festival\_Name , city , date  from festivaldata**

**where country = 'USA' ;**

**Q3 . select festival\_name , ticket\_type , price from festivaldata**

**where country = 'India' and Price >100 ;**

**Q4. select festival\_name , country , ticket\_type from festivaldata**

**where country = 'India' or country = 'USA' ;**

**Q5. select ticket\_type , sum(price) from festivaldata**

**group by ticket\_type ;**

**Q6. select festival\_name , price from festivaldata**

**order by Price DESC LIMIT 5 ;**

**Q7. select festival\_name , price from festivaldata order by**

**price desc limit 5 offset 5 ;**

**Q8. select festival\_name , sum(organizer\_profit) as Total\_profit**

**from festivaldata**

**where country = 'India'**

**group by festival\_name**

**having  avg(price) > 100 ;**

**Q9 . select country , max(organizer\_profit) from festivaldata**

**group by country ;**

**Q10. select ticket\_type , sum(organizer\_profit) as total\_profit**

**from festivaldata where country = 'India' or country = 'USA'**

**group by ticket\_type**

**having avg(price) > 50 order by total\_profit DESC limit 3;**

**Q11. select movie\_name , avg(rating) as avg\_rating**

**from user\_watch\_activity**

**group by movie\_name**

**order by avg\_rating DESC limit 5 offset 3 ;**

**Q12. select user\_name, watch\_time\_minutes**

**from user\_watch\_activity**

**where watch\_time\_minutes BETWEEN 100 AND 150**

**order by watch\_time\_minutes ASC ;**

**Q13. select user\_name , watch\_time\_minutes from user\_watch\_activity**

**where genre = 'Sci-Fi' order by watch\_time\_minutes DESC limit 1**

**offset 1 ;**

**Q14. select genre , watch\_time\_minutes as total\_watch\_time**

**from user\_watch\_activity**

**where country = 'Japan' order by watch\_time\_minutes DESC limit 1;**

**Q15. select user\_name , total\_movies\_watched from user\_watch\_activity**

**where subscription\_type = 'Premium' order by total\_movies\_watched**

**DESC Limit 5 ;**

**Q16. select transaction\_id , customer\_name , transaction\_value**

**from payment\_transactions**

**order by  transaction\_value DESC limit 5;**

**Q17. select product\_category ,**

**sum(price\_per\_unit \* Quantity\_purchased) as before\_discount ,**

**sum(transaction\_value) as after\_discount**

**from payment\_transactions**

**group by product\_category ;**

**Q18. select payment\_method , avg(transaction\_value) as avg\_transaction\_value**

**from payment\_transactions**

**group by payment\_method order by avg\_transaction\_value DESC limit 1 ;**

**Q19. SELECT**

**customer\_name,**

**SUM(transaction\_value) AS total\_spending,**

**CASE**

**WHEN SUM(transaction\_value) > 2000 THEN 'High Spend'**

**WHEN SUM(transaction\_value) BETWEEN 1000 AND 2000 THEN 'Medium Spend'**

**ELSE 'Low Spend'**

**END AS spending\_segment**

**FROM**

**payment\_transactions**

**GROUP BY**

**customer\_name;**

**Q20. select customer\_name , avg(customer\_rating)**

**from payment\_transactions**

**group by customer\_name**

**having avg(customer\_rating) < 4;**