Solutions

#1) Daily Active Users (DAU)

```
SELECT DATE AS day,
      COUNT(DISTINCT User_id) AS daily_users
   FROM daily_active_users
   GROUP BY day
```

#2) Daily Spenders Conversion: out of total users who came on the day, how many spent in game on that day.

```
SELECT t1.date_t1 AS Date,
(t2.count_t2/t1.count_t1)*100 AS Conversion_percent
FROM
(SELECT COUNT(DISTINCT User_ID) count_t1 ,
STR_TO_DATE(`Date`,'%m/%d/%y') date_t1
FROM daily_active_users
GROUP BY STR_TO_DATE(`Date`,'%m/%d/%y')) t1
 INNER JOIN
 (SELECT COUNT(DISTINCT User_ID) count_t2,
 DATE(`Date_Time`) date_t2
 FROM rev
 GROUP BY DATE(`Date_Time`)) t2
 ON t1.date_t1 = t2.date_t2
```

#3) Revenue per User (DARPU): Total revenue generated on the day/total number of users who came on the day.

```
SELECT t1.date_t1 AS Date,
t2.money/t1.count_t1 AS DARPU
FROM
(SELECT COUNT(DISTINCT User_ID) count_t1 ,
STR_TO_DATE(`Date`,'%m/%d/%y') date_t1
FROM daily_active_users
GROUP BY STR_TO_DATE(`Date`,'%m/%d/%y')) t1
 INNER JOIN
 (SELECT SUM(revenue) money,
 DATE(`Date_Time`) date_t2
 FROM rev
```

```
 GROUP BY DATE(`Date_Time`)) t2
 ON t1.date_t1 = t2.date_t2
```

/*4)Retention funnel
a. D2: (Out of players who were active on D1, how many users came back on the next
day)
b. D7: (Out of players who were active on D1, how many users came back on the 7 the
day)*/

```
SELECT daily_active_users.date,
      SUM( t2.date = daily_active_users.date) AS total_number_of_users,
      SUM( t2.date = daily_active_users.date + interval 1 day ) AS d2,
      SUM( t2.date = daily_active_users.date + interval 6 day ) AS d7,
FROM (SELECT DISTINCT date, user_id
     FROM daily_active_users
    ) t1 LEFT JOIN
    (SELECT DISTINCT date, user_id
     FROM daily_active_users
    ) t2
    ON t1.user_id = t2.user_id AND
       t2.date IN (t1.date, t1.date + interval 1 day,t1.day + interval 6 day)
```

/* 5) 30-day Active Spenders:
a. ex: if you are calculating 30day active spenders for June 2020-07-01, count (Players who spent at least once in the game in last 30days i.e. 2020-06-01 to 2020-07-01).*/

Note: The sample data given is kind of inconsistent, the question asks the count of users who made at least one purchase in the month of June, but in the given data the table rev does not have any info about the transaction for the month of June. So when we run this query it will show the results of July

Also, as per the sample output provide for this question is as follows:

Date	30 day active spenders
01-7-2020	6
02-7-2020	7

The sample out put provided is asking for the count of users who made atleast one transaction each day and probably display the total at the end. So the first Query (the one with union all) provides the same result result as per the sample out put given.

However ,I have also given another query with rollup this returns just one row with month and year and its totals.
This query displays the month and year, total number of user who played the game in a given month and the total number of people who made at least one purchase in a given month

```
SELECT DATE(Date_Time) AS Each_day ,
COUNT(DISTINCT User_ID) AS atleast_one_pruchase
 FROM rev
 GROUP BY DATE(Date_Time)
 UNION ALL
 SELECT '30-day Active Spenders', COUNT(DISTINCT User_ID)
 FROM rev
```

#or a second senario

```
 SELECT
   t1.date_t1 AS Date,
   total_users_on_a_day,
   atleast_one_pruchase
FROM
   (SELECT
      COUNT(DISTINCT User_ID) total_users_on_a_day,
         DATE_FORMAT(STR_TO_DATE(Date, '%m/%d/%y'),'%Y-%m') date_t1
   FROM
      daily_active_users
   GROUP BY DATE_FORMAT(STR_TO_DATE(Date, '%m/%d/%y'),'%Y-%m')) t1
      INNER JOIN
```

```
   (SELECT
      COUNT(DISTINCT User_ID) atleast_one_pruchase,
         DATE_FORMAT(Date_Time,'%Y-%m') date_t2
   FROM
      rev
   GROUP BY DATE_FORMAT(Date_Time,'%Y-%m') WITH ROLLUP) t2 ON t1.date_t1 =
t2.date_t2
```

/* 6) Cumulative Spend:
a. Ex: if a user spent 10$ on 2020-07-01 ,5$ on 2020-07-02 and 0$ on 2020-07-03, his
cumm spend for 10$ for 2020-07-01, 15$ on 2020-07-02 and 15$ on 2020-07-03 */

```
SELECT date(Date_time) AS Day,
      user_id,
      ROUND(SUM(Revenue),2) AS amt_spent,
      SUM(ROUND(SUM(Revenue),2)) OVER (PARTITION BY user_id ORDER BY
date(Date_time)) AS cumulative_amount
FROM rev
GROUP BY Day, user_id;
```

Stats Q3: Assume that a typical computer manufactured by HP lasts 10 Months and that the standard deviation is 50 days. Computer life follows a normal distribution. What is the probability that a computer made by this company will last at most 1 Year? *(Assume 1 months has approx. 30 days).*

Solution:

Given that mean is 10 months the mean in the number of days will be 10X 30= 300 days (*Assuming 1 month has approx. 30 days)*

and a standard deviation of 50 days,

we want to find the cumulative probability that computer's life is less than or equal to 365 days (1 year).
Thus, we know the following:

- The value of the normal random variable is 365 days.

- The mean is equal to 300 days.
- The standard deviation is equal to 50 days.

We enter these values into the Normal Distribution Calculator and compute the cumulative probability.
The answer will be P(X $\leq$ 365) = 0.90.
The probability that the HP computer will last at most 365 days is 90%

Stats Q4: Assume we have divided our player base into two groups A and B. A has 20,000 players and B has 40,000 players.
Games played per Player for Group A = 20
Games Played Per Player for Group B = 22

Standard Deviation for Group A = 5 Standard Deviation for Group B = 19

What is the confidence that Group B is better than Group A? Explain the methodology used.

Note: I have done this calculation in python. Since the aim of the experiment/ back story of the scenario is not mentioned in the problem I am Computing the Confidence Interval for a Difference Between Two Means

First we need to formulate the hypothesis

Then, we need to compute Sp, the pooled estimate of the common standard deviation.
Aka std_N1N2 #average standard deviations between groups as mentioned in the calculation

$$S_p = \sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$$

We get value for SP

We are going with Table because the sample size is > 30
Next we substitute the Z score for 95% confidence

$$(\bar{x}_1 - \bar{x}_2) \pm z\, S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

Difference between means will be calculated.

So possible interpretation would be that You reject/accept null hypothesis based on the results.

```
N1 = 20000 #numbers of observations
N2 = 40000

df = (N1 + N2 - 2) #deegrees of freedom

std1 = 5 #standard deviations
std2 = 19

std_N1N2 = sqrt( ((N1 - 1)*(std1)**2 + (N2 - 1)*(std2)**2) / df) #average standard deviations between groups.

diff_mean = 22- 20 #difference between means
MoE = z.ppf(1.96, df) * std_N1N2 * sqrt(1/N1 + 1/N2) # margin of error
```

The **Z value** for **95**% **confidence** is **Z**=1.96.

```
print(f"Difference between means is: {diff_mean:.2f} ( {diff_mean-MoE:.2f}, {diff_mean + MoE:.2f} )")
```