Section – A: Theory Questions

1. **list functions used to display the current date and time in SQL.**
   now(), curdate(), curtime(), current_timestamp() are some of the functions used to display the current date and time in sql

2. **What are the date functions in SQL and explain with examples.**
   Functions:
   day, dayname, month, monthname, year, dayofyear, dayofweek, weekend, weekday, hour, minute, seconds
   Description:
   - The "**DAY**" function in SQL returns the day of the month for a given date value. The syntax for the DAY function is "DAY(date)". For example, "DAY('2023-02-08')" would return "8".
   - The "**DAYNAME**" function in SQL returns the name of the day of the week for a given date value. The syntax for the DAYNAME function is "DAYNAME(date)". For example, "DAYNAME('2023-02-08')" would return "Wednesday".
   - The "**MONTH**" function in SQL returns the number of months for a given date value. The syntax for the MONTH function is "MONTH(date)". For example, "MONTH('2023-02-08')" would return "2".
   - The "**MONTHNAME**" function in SQL returns the name of the month for a given date value. The syntax for the MONTHNAME function is "MONTHNAME(date)". For example, "MONTHNAME('2023-02-08')" would return "February".
   - The "**YEAR**" function in SQL returns the year for a given date value. The syntax for the YEAR function is "YEAR(date)". For example, "YEAR('2023-02-08')" would return "2023".
   - The "**DAYOFYEAR**" function in SQL returns the day of the year for a given date value. The syntax for the DAYOFYEAR function is "DAYOFYEAR(date)". For example, "DAYOFYEAR('2023-02-08')" would return "39".
   - The "**DAYOFWEEK**" function in SQL returns the number of the day of the week for a given date value, where Sunday is 1 and Saturday is 7. The syntax for the DAYOFWEEK function is "DAYOFWEEK(date)". For example, "DAYOFWEEK('2023-02-08')" would return "3".
   - The "**WEEKEND**" and "WEEKDAY" functions in SQL return a 1 if the day of the week is a weekend or weekday, respectively, for a given date value. The syntax for these functions is "WEEKEND(date)" or "WEEKDAY(date)". For example, "WEEKEND('2023-02-08')" would return "0" (weekday), while "WEEKDAY('2023-02-05')" would return "1" (weekend).
   - The "**HOUR**" function in SQL returns the hour of the day for a given time value. The syntax for the HOUR function is "HOUR(time)". For example, "HOUR('14:30:00')" would return "14".

- The "**MINUTE**" function in SQL returns the minute of the hour for a given time value. The syntax for the MINUTE function is "MINUTE(time)". For example, "MINUTE('14:30:00')" would return "30".
- The "**SECONDS**" function in SQL returns the second of the minute for a given time value. The syntax for the SECONDS function is "SECONDS(time)". For example, "SECONDS('14:30:15')" would return "15".

3. **What are the date difference functions in SQL and explain with examples.**

Functions:

datediff, timediff, extract, date_add, date_sub, dateformat, format, str_to_date

Description:

- The "**DATEDIFF**" function in SQL returns the number of days between two date values. It takes two date expressions as arguments and returns the number of days between them. Ex: SELECT DATEDIFF(date1, date2) FROM table_name;
- The "**TIMEDIFF**" function in SQL returns the difference between two-time values. It takes two-time expressions as arguments and returns the difference between them. Ex: SELECT TIMEDIFF(time1, time2) FROM table_name;
- The "**EXTRACT**" function in SQL is used to extract a portion of a date or time value. It takes a date or time value and an argument indicating the portion to extract (e.g. year, month, day, hour, minute, etc.) and returns that portion of the value. Ex: SELECT EXTRACT(YEAR FROM date_column) FROM table_name;
- The "**DATE_ADD**" function in SQL adds a specified interval to a date or time value. It takes a date or time value and an interval expression and returns a new date or time value that is the original value plus the interval. Ex: SELECT DATE_ADD(date_column, INTERVAL 10 DAY) FROM table_name;
- The "**DATE_SUB**" function in SQL subtracts a specified interval from a date or time value. It takes a date or time value and an interval expression and returns a new date or time value that is the original value minus the interval. Ex: SELECT DATE_SUB(date_column, INTERVAL 5 DAY) FROM table_name;
- The "**DATEFORMAT**" function in SQL is used to format a date or time value in a specific way. It takes a date or time value and a format string and returns the value formatted according to the format string. Ex: SELECT DATE_FORMAT(date_column, '%Y-%m-%d') FROM table_name;
- The "**FORMAT**" function in SQL is similar to the "DATEFORMAT" function and is used to format a value in a specific way. It takes a value and a format string and returns the value formatted according to the format string. Ex: SELECT FORMAT(value, 2) FROM table_name;
- The "**STR_TO_DATE**" function in SQL is used to convert a string value into a date or time value. It takes a string value and a format string and returns the string value converted into a date or time value. Ex: SELECT STR_TO_DATE(string_column, '%Y-%m-%d') FROM table_name;

4. **What is the difference between datetime and timestamp?**

   The main difference between datetime and timestamp is storage

   datetime : 1000 year - 9999 year

   timestamp : 1970 year - 2038 year

   Format for datetime and timestamp are as follows:

   datetime : yyyy-mm-dd hh:mm:ss

   timestamp : it is calculated from 1970, Jan,1

5. **What is a join in SQL and why is it used?**

   A join in SQL is a way to combine data from two or more tables based on a common column between them. It allows you to query data from multiple tables as if they were one table. Joins are used to avoid data redundancy, save storage space, and make database queries more efficient.

Section – B: Practice Questions

-- ---

Database: Seattle bike share

-- ---

1. **How many unique bikes were used for trips between October 1, 2014, and November 1, 2014?**

```
SELECT COUNT(DISTINCT bikeid) AS num_bikes
FROM trips
WHERE starttime BETWEEN '2014-10-01 00:00:00' AND '2014-11-01 23:59:59';
```

2. **What was the average trip duration in minutes for trips taken by users who were born in 1990?**

```
SELECT AVG(tripduration/60) AS avg_trip_duration
FROM trips
WHERE birthyear = 1990;
```

3. **Which station had the most bike rentals in the month of September 2014?**

```
SELECT from_station_name, COUNT(trip_id) AS num_rentals
FROM trips
WHERE starttime BETWEEN '2014-09-01 00:00:00' AND '2014-09-30 23:59:59'
GROUP BY from_station_name
ORDER BY num_rentals DESC
LIMIT 1;
```

4. **How many trips were taken in the month of July 2014? (not applicable using code eval)**

```
SELECT COUNT(trip_id) AS num_trips
FROM trips
WHERE cast(starttime as datetime) BETWEEN '2014-07-01 00:00:00' AND '2014-07-31 23:59:59';
```

5. **What was the average trip duration in minutes for only male riders versus female riders in the month of June 2015?**

```
SELECT gender, AVG(tripduration/60) AS avg_trip_duration
FROM trips
where starttime BETWEEN '2015-06-01 00:00:00' and '2015-06-31 00:00:00'
group by gender
having gender is not null;
```

6. **How many trips started and ended at the same station?**

```
SELECT COUNT(trip_id) AS num_trips
FROM trips
WHERE from_station_id = to_station_id;
```

7. **What was the average trip duration in minutes for trips taken on weekdays versus trips taken on weekends? (not applicable using code eval)**

```
SELECT WEEKDAY(t.starttime) AS day_of_week,
    CASE WHEN WEEKDAY(t.starttime) IN (5, 6) THEN 'weekend'
        ELSE 'weekday' END AS day_type,
    AVG(t.tripduration/60) AS avg_duration_minutes
FROM trips AS t
GROUP BY day_of_week, day_type;
```

8. **For each day in the month of December 2014, what was the average trip duration in hours?**

```
SELECT DATE(starttime) AS trip_date, AVG(tripduration/(60*12)) AS avg_trip_duration
FROM trips
WHERE starttime BETWEEN '2014-12-01 00:00:00' AND '2014-12-31 23:59:59'
GROUP BY trip_date;
```

9. **What was the total precipitation in inches for each trip taken in October 2014, along with the corresponding start time and weather data for that day? (not applicable using code eval)**

```
SELECT t.starttime, w.Precipitation_In
FROM trips t
JOIN weather w ON DATE(t.starttime) = str_to_date(w.Date, "%m/%d/%Y")
WHERE t.starttime BETWEEN '2014-10-01' AND '2014-10-31';
```

10. **What is the maximum temperature in Fahrenheit recorded in that month? What number of trips are taken in each month of the year? (not applicable using code eval)**

```
select year(trips.starttime), monthname(trips.starttime), max(weather.Max_Temperature_F)
as max_temp, count(*)
from trips inner join weather ON trips.starttime = str_to_date(weather.Date, "%m/%d/%Y")
group by 1, 2
order by 1,2;
```