**1.1**

SQL plays a crucial role in managing data for dynamic websites like online stores. It handles the storage, retrieval, and manipulation of product information, user accounts, and order details within relational databases. SQL allows efficient querying of product catalogs, ensures secure handling of user credentials, and supports the processing of transactions, enabling real-time updates and interactions on the website.

**1.2**

Although SQL makes it possible to store, retrieve, and manipulate data within relational databases, it is essential to web applications. It oversees vital functions like transaction processing, product information, and user identification, guaranteeing safe and effective data handling. Because SQL is structured, programmers may execute intricate queries and changes, enabling dynamic and interactive web functions.

**1.3**

Data Integrity: To guarantee data integrity, SQL databases offer features like triggers, constraints, and foreign key connections.

Scalability: As online applications expand, SQL databases can manage massive data volumes and scale successfully.  
Flexibility of Query: SQL provides a strong and adaptable query language that enables developers to effectively access, modify, and analyze data.

**1.4**

Efficiency: Because SQL databases can process complicated queries and transactions quickly, they are effective at managing massive volumes of data.  
Data Organization: SQL makes it possible to organize structured data using tables, relationships, and constraints, guaranteeing that the data is readily available and properly arranged.  
Data Retrieval Capabilities: SQL has strong querying features that let users quickly retrieve particular subsets of data, enabling rapid and adaptable data retrieval for applications.

**1.5**

MySQL: Because of its scalability and ease of use, this open-source relational database management system is frequently utilized for online applications.  
A powerful relational database management system (DBMS) created by Microsoft, SQL Server is renowned for its scalability, security features, and compatibility with other Microsoft products.  
MongoDB is a well-known NoSQL database management system that can handle massive amounts of unstructured data since it stores data in flexible documents that resemble JSON.

**DATABASE FUNDAMENTALS**

**2.1**

A database table is an ordered set of data that is arranged in rows and columns. A record is represented by a row, and a field or attribute by a column. A spreadsheet, on the other hand, likewise arranges data in rows and columns, with distinct data contained in each cell. This tabular structure is used by database tables and spreadsheets to effectively store, alter, and retrieve data.

**2.2**

In a database table, columns denote the vertical structures containing particular kinds of data. The table is structured and consistent since each column is intended to contain a specific type of data. A "Customers" table's "Customer Name" column, for instance, might hold textual information like "John Doe" or "Jane Smith."  
Why do data types matter in relation to databases?  
Text: Text strings and alphanumeric characters are stored in this data type. It guarantees correct text data storage and retrieval without information loss or truncation.  
Number: Numeric values, which can be either integers or floating-point numbers, are stored in the numbers data type. It guarantees accurate execution of mathematical operations and comparisons.

Date: The date data type keeps track of calendar dates, making sure that date-related tasks like filtering, sorting, and date arithmetic are performed accurately.

**2.3**

In order to guarantee data integrity and effective database storage, data types are essential:  
Text (VARCHAR): Variable-length alphanumeric characters are stored in this data type. By allotting space only as required for each entry, it maximizes storage efficiency and guarantees effective storage.  
Whole numbers are stored in integer data types (INTs), which do not include decimal points. By imposing restrictions on numerical values and prohibiting the storage of incorrect data types like text or dates, they are crucial for preserving data integrity.  
Date (DATE): Date data types accept only valid date entries, guaranteeing data integrity by storing calendar dates. They provide effective querying and reporting and are essential for date arithmetic and comparisons.

**Expense Tracker Database Design**

**3.1**

To efficiently monitor and evaluate spending, an Expense Tracker application must track multiple data points. Among them are:

Expense Amount: The total amount paid or incurred.  
Date of item: The precise day the item was incurred, which is important for keeping track of expenses over time.  
Classification of the Cost: For the purposes of budgeting and analysis, categorization aids in the organization of expenses (such as food and transportation).  
Description/Notes: Added details that shed light on the cost; helpful for future reference and comprehension.  
Payment Method: The method used to pay for the expense (cash, credit card, etc.), which helps with financial tracking and reconciliation.

**3.2**

Essential columns like expense id for unique identification, amount for the expense amount, date for the expense's occurrence, category for classifying the expense type, description for any further notes, and payment method for tracking the expense's mode of payment are all included in this schema for the "Expenses" table. The purpose of these columns is to effectively organize and save data for an expense tracker application.

**Table Name: EXPENSES**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| Expense id | INT | Primary key, auto-incremented ID for each expense |
| amount | DECIMAL | Stores the amount spent on the expense |
| date | DATE | Date when the expense occurred |
| category | VARCHAR | Category of the expense (e.g., groceries, utilities) |
| description | TEXT | Optional field for additional notes or description |
| payment method | VARCHAR | Payment method used for the expense (e.g., credit card, cash) |