# Week 1: Introduction & Foundational Skills (Focus on Project Relevance)

This week, we'll be diving into the exciting world of SQL and databases! We'll explore what SQL is used for, how it benefits web applications, and the building blocks of databases: tables, columns, and data types. But most importantly, we'll get our hands dirty by creating a basic database structure for our upcoming Expense Tracker project!

## Learning Objectives:

Understand the purpose and applications of SQL, particularly for web applications.

Identify the fundamental components of a database: tables, columns, and data types.

Design a basic database schema for our Expense Tracker project.

Instructions

This assignment is designed to be completed in approximately 2 hours.

\*\*What you'll need:\*\*

Access to a computer with internet access

A text editor (Microsoft Word document)

Drawing software (e.g. Draw.io, visual paradigm) for the bonus question.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Submission:

Save your completed assignment as a document (e.g., .docx, pdf)

Submit your document through the designated course platform.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Part 1: Understanding SQL (30 minutes)

\*\*Question 1. Research\*\*

Use online resources like websites or PowerPoint slides.

\*\*1.1.\*\* In a single Word document, summarize your findings in a short paragraph (3-5 sentences).

Web Applications:

Imagine a dynamic website like an online store. How do you think SQL plays a role in managing data behind the scenes? Consider how product information, user accounts, and order details might be stored and accessed.

***Answer:***  In an online store, SQL plays a crucial role in managing data behind the scenes. It is used to store and retrieve data related to products, user accounts, and order details. For example, SQL databases can manage inventory by tracking product quantities, prices, and descriptions. They also handle user information, enabling account creation, login authentication, and maintaining order histories. Efficient data retrieval and organization through SQL ensure smooth and dynamic interactions on the website.

\*\*1.2.\*\* Write a short explanation (3-5 sentences) in your document about the role of SQL in web applications.

***Answer:*** SQL (Structured Query Language) is essential in web applications for managing and manipulating relational databases. It allows developers to create, read, update, and delete data stored in the database, enabling dynamic functionality in web applications. By using SQL, web applications can efficiently manage large amounts of data, perform complex queries, and ensure data integrity and security.

\*\*1.3.\*\* List 3 benefits of using SQL for web applications.

***Answer :*** a) Efficiency

b) Data Organization

c) Data Retrieval Capabilities

\*\*1.4.\*\* Think about efficiency, data organization, and data retrieval capabilities. Briefly explain each benefit in your document (1-2 sentences per benefit).

***Answer :*** **Efficiency:** SQL enables quick and efficient data operations, allowing web applications to handle large volumes of transactions and user queries without significant delays.

**Data Organization:** SQL helps structure data into tables with relationships, making it easier to manage and maintain consistency, which is crucial for the integrity of web applications.

**Data Retrieval Capabilities:** SQL supports complex querying capabilities, allowing developers to extract specific data sets based on various criteria, which enhances the functionality and user experience of web applications.

\*\*1.5.\*\* List any 3 Database Management Systems.

***Answer:*** a) MySQL

b) PostgreSQL

c) Microsoft SQL Server

## Part 2: Database Fundamentals (45 minutes)

\*\*Question 2.1: Tables\*\*

Think about how data is organized in rows and columns.

In your document, define a database table and explain its similarity to a spreadsheet (2-3 sentences).

***Answer:***  A database table is a structured set of data organized in rows and columns. Each row represents a single record, and each column represents a field within that record. This structure is similar to a spreadsheet, where data is stored in a tabular format, making it easy to read, manage, and manipulate.

\*\*Question 2.2: Columns\*\*

Consider different types of data like text, numbers, and dates.

Define "columns" and provide an example with an explanation (2-3 sentences) in your document.

Data Types: Why are data types important in a database? Briefly explain 3 common data types (e.g., Text, Number, Date).

***Answer:***  Columns in a database table define the types of data that can be stored in each field of the table. For example, a column named "Email" in a user accounts table might store text data containing email addresses. Each column in a table corresponds to a specific attribute of the data stored in the rows, such as names, numbers, or dates.

Data Types: Data types are crucial in a database because they define the nature of the data that can be stored in each column, ensuring data integrity and efficient storage. Common data types include:

* **Text:** Used for storing alphanumeric characters, such as names or addresses. Text data types ensure that character-based information is stored and retrieved accurately.
* **Number:** Used for storing numerical values. This data type is essential for performing mathematical operations and aggregations within the database.
* **Date:** Used for storing date and time information. Date data types allow for the precise storage of dates and times, enabling efficient sorting and querying of time-based data.

\*\*Question 2.3: Data Types\*\*

Think about how data types ensure data integrity and efficient storage.

Explain the importance of data types and provide brief explanations of 3 common types (2-3 sentences each) in your document.

***Answer:***  Data types are crucial in a database because they define the nature of the data that can be stored in each column, ensuring data integrity and efficient storage. Common data types include:

* **Text:** Used for storing alphanumeric characters, such as names or addresses. Text data types ensure that character-based information is stored and retrieved accurately.
* **Number:** Used for storing numerical values. This data type is essential for performing mathematical operations and aggregations within the database.
* **Date:** Used for storing date and time information. Date data types allow for the precise storage of dates and times, enabling efficient sorting and querying of time-based data.

## Part 3: Expense Tracker Database Design (45 minutes)

\*\*3.1. Planning:\*\*

We'll be building an Expense Tracker application. What kind of data do you think we'll need to track? List at least 5 data points relevant to our project.

\* Consider information like expense amount, date, and category.

\* List your identified data points in your document.

***Answer:*** For an Expense Tracker application, the data points that need to be tracked include:

1. Expense Amount
2. Expense Date
3. Expense Category
4. Description of Expense
5. Payment Method
6. Vendor/Payee

\*\*3.2. Tables:\*\*

Considering the data points you listed, design a basic database schema with one main table (likely named "Expenses").

\* Define the columns needed for this table.

\* Assign appropriate data types to each column based on the kind of data it will hold. (e.g., amount: number, date: date, category: text)

In your document, create a table structure that includes:

\* Table name (e.g., Expenses)

\* Column names (e.g., expense\_id, amount, date, category)

\* Data type for each column (e.g., INT, DECIMAL, DATE, TEXT)

***Answer:*** Based on the identified data points, the main table "Expenses" will be structured as follows:

**Table Name:** Expenses

| Column Name | Data type | Description |
| --- | --- | --- |
| expense\_id | INT | Unique identifier for each expense |
| amount | DECIMAL | Amount of the expense |
| date | DATE | Date of the expense |
| category | TEXT | Category of the expense |
| description | TEXT | Description or notes about the expense |
| payment\_method | TEXT | Payment method used for the expense |
| vendor | TEXT | Vendor or payee for the expense |

**Column Definitions:**

* **expense\_id:** INT - A unique identifier for each expense entry, typically auto-incremented.
* **amount:** DECIMAL - The monetary amount of the expense, allowing for decimal values to handle cents.
* **date:** DATE - The date when the expense was incurred.
* **category:** TEXT - The category of the expense (e.g., groceries, utilities, entertainment).
* **description:** TEXT - A brief description or notes about the expense.
* **payment\_method:** TEXT - The method of payment used (e.g., credit card, cash, bank transfer).
* **vendor:** TEXT - The name of the vendor or payee for the expense.

This table structure will allow for efficient tracking and querying of expense data in the Expense Tracker application.

## Bonus:

Sketch a simple Entity Relational Diagram (ERD) of your table structure, including column names and data types.

Use drawing software or a simple table format to visually represent your schema.

\*\* Remember: There might be multiple ways to design your database schema. The goal is to understand the concepts and create a logical structure to store our expense tracking data.

***Answer:***  Here's a simple Entity-Relationship Diagram (ERD) and a corresponding table format for the Expense Tracker database schema

Entity-Relationship Diagram (ERD):

+-------------------+ +-------------------+

| Expenses | | |

+-------------------+ | Category |

| | +-------------------+

| +---------------+ | 1 \* | |

| | expense\_id |<----------| category\_id |

| | INT (PK) | | INT (PK) |

| +---------------+ | category\_name |

| | amount | +-------------------+

| | DECIMAL | \* 1

| +---------------+

| | date |

| | DATE |

| +---------------+

| | category\_id |

| | INT (FK) |

| +---------------+

| | description |

| | TEXT |

| +---------------+

| | payment\_method|

| | TEXT |

| +---------------+

| | vendor |

| | TEXT |

| +---------------+

### Table Format:

**Table Name: Expenses**

| Column Name | Data type | Constraints | Description |
| --- | --- | --- | --- |
| expense\_id | INT | PRIMARY KEY | Unique identifier for each expense |
| amount | DECIMAL |  | Amount of the expense |
| date | DATE |  | Date of the expense |
| category | TEXT | FOREIGN KEY | Category of the expense |
| description | TEXT |  | Description or notes about the expense |
| payment\_method | TEXT |  | Payment method used for the expense |
| vendor | TEXT |  | Vendor or payee for the expense |

Table Name: Category

| Column Name | Data Type | Constraints | Description |
| --- | --- | --- | --- |
| category\_id | INT | PRIMARY KEY | Unique identifier |
| category\_name | TEXT |  | Name of the expense category |

### Explanation:

* **Expenses Table:** Contains detailed information about each expense, including its amount, date, category (referenced by category\_id), description, payment method, and vendor.
* **Category Table:** Stores unique categories for expenses, identified by category\_id, and includes the name of each category.

This structure allows for efficient organization and retrieval of expense data based on categories while maintaining data integrity through foreign key constraints.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_WELLDONE\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_