Lesson 1: Introduction to SQL

## Introduction

Structured Query Language (SQL) is a powerful tool for managing and analyzing data stored in relational databases. As a corrections analyst, you likely work with large datasets containing information about individuals, facilities, incidents, and case processing. SQL allows you to efficiently retrieve, filter, and analyze this data, making it an essential skill for data-driven decision-making in the corrections field.

This lesson will introduce SQL, explain how relational databases are structured, and guide you through writing your first SQL queries.

### What is SQL?

SQL (pronounced “sequel” or “S-Q-L”) is the standard programming language used for interacting with relational databases. Unlike general-purpose programming languages like Python or R, SQL is specifically designed for managing structured data. It allows you to:

* Retrieve data from a database
* Filter and sort records based on specific criteria
* Combine data from multiple tables
* Perform calculations and generate summary statistics
* Modify and update records in a dataset

SQL is widely used in government agencies, law enforcement, corrections departments, and research institutions because it provides a reliable and efficient way to store and analyze data.

## Why SQL Matters for Corrections Analysts

As a corrections analyst, you work with large datasets that track case records, time computation, facility populations, and supervision trends. These datasets are often stored in **relational databases**, which are structured collections of data organized into tables.

Rather than manually sorting through spreadsheets, SQL allows you to quickly answer questions like:

* How many individuals were admitted to a correctional facility in the past six months?
* What is the average length of stay for individuals released in a given year?
* How many pending cases are there in a particular jurisdiction?

By mastering SQL, you can retrieve insights more efficiently, ensuring that your reports and analyses are accurate and reproducible.

## Understanding Relational Databases

A **relational database** is a collection of tables that store related data. Each table consists of **rows** (records) and **columns** (fields). A table can be thought of like a spreadsheet, where each row represents an individual record, and each column represents a specific attribute of that record.

### Key Components of a Relational Database

* **Tables** – The fundamental building blocks of a database. Each table stores a specific type of data (e.g., inmate records, facility details, time computation data).
* **Columns (Fields)** – Each column in a table represents a specific type of information, such as name, date of birth, sentence length, or housing assignment.
* **Rows (Records)** – Each row represents an individual entry in a table, such as a specific inmate’s record or a particular facility.
* **Primary Keys** – A unique identifier for each record in a table (e.g., an inmate ID or case number).
* **Foreign Keys** – A column that creates a link between two tables by referencing the primary key of another table.

### Example: Corrections Data Stored in a Relational Database

Consider a simple database that tracks facility populations. It might include tables like these:

In this example:

* The **Individuals Table** records basic details about individuals, including their assigned facility.
* The **Facilities Table** stores information about each facility.
* The **Facility\_ID** in the Individuals Table is a **foreign key** that links each person to a facility in the Facilities Table.

This structure allows SQL queries to pull insights from multiple tables efficiently.

## Writing Your First SQL Query

Now that you understand the basics of relational databases, let’s write a simple SQL query.

The most fundamental SQL command is SELECT, which retrieves data from a table.

### Basic Syntax

SELECT column1, column2, ...  
FROM table\_name;

To retrieve all columns from a table, you can use \*:

SELECT \*  
FROM person;

This query returns all records from the **Individuals Table**, displaying every column.

### Selecting Specific Columns

If you only need certain columns, specify them by name:

SELECT person\_id, last\_name, dob  
FROM person;

## Key Takeaways

* **SQL is the standard language for interacting with relational databases.** It allows analysts to efficiently retrieve and analyze data.
* **Relational databases store data in tables, which are structured collections of rows and columns.**
* **A SELECT statement retrieves data from a table.** You can choose specific columns to display using SELECT column\_name.
* **Understanding SQL is essential for corrections analysts** because it enables efficient reporting, case tracking, and population analysis.

## Additional Resources

This lesson provides a foundation for SQL querying. In the next lesson, you’ll learn how to refine queries further by applying filtering techniques with WHERE, AND, OR, and NOT.