**PUT YOUR NAME HERE:**

**Introduction to SQL (Part 2): INSERT, UPDATE, and DELETE**

This second part of the SQL exercise builds on the BlandCo database you created in Part 1. In this activity you will practice making changes to the data using the INSERT, UPDATE, and DELETE statements. Each task below includes an example SQL command.

Try running the example, then modify it to explore on your own. Always double-check your work with a SELECT query to see the results.

**1. Adding new records (INSERT)**

A. Add a new employee. For example, to add an employee named Adam Wilson who is a Consultant hired on April 1, 2024:

**INSERT INTO employees (first\_name, last\_name, role, hire\_date)
VALUES ('Adam', 'Wilson', 'Consultant', '2024-04-01');**

|  |
| --- |
| DISCUSS: Why was it not necessary to specify the **id** column in the last statement? |

|  |
| --- |
| TAKE NOTE: In SQL, dates are typically expressed as Year – Month – Day. So, Valentines Day, 2026 would be '2026-02-14' |

|  |
| --- |
| Now, can you list all employees in roster order? |

B) Add a new customer, "Abel Industries" as an "Agriculture" company.

|  |
| --- |
| Now, can you generate a list of all customers, sorted first by industry name, and then by company name? |

C) Add a new engagement linking the employee Isabella Davis to the company Cedar Solutions. It is an Active “IT Security Assessment” engagement with zero hours billed.

D) Add a new engagement linking the employee Adam Wilson to the company Abel Industries. It is an Active “Cloud Migration” engagement with zero hours billed.

**2. Updating existing records (UPDATE)**

A) Promote an employee. Suppose you want to change Alice Smith’s role to 'Senior Consultant':

 **UPDATE employees
 SET role = 'Senior Consultant'
 WHERE id =1;**

|  |
| --- |
| Verify the change with a SELECT query. |

|  |
| --- |
| DISCUSS: We used id instead of **first\_name = 'Alice' and last\_name = 'Smith'** in the where clause. Would this be any better or worse than using the employee id? Why? |

B) Correct an engagement’s hours. If you discover that the 'Network Upgrade Plan' engagement actually took 35 hours instead of 30. Use UPDATE to make this correction

C) Mark the 'Orion Gate' engagement as completed:

**3. Removing records (DELETE)**

A) Before we begin, write this SELECT statement:

**SELECT \* FROM engagements WHERE status = 'Active'
ORDER BY id DESC;**

B) Remove the Cloud Migration project engagement you created for Abel Industries. You decide no longer need this engagement entry, so delete it:

**DELETE FROM engagements WHERE id = 31;**

|  |
| --- |
| After deleting, verify that it is gone using SELECT. |

|  |
| --- |
| DISCUSS: How is this different from marking the engagement complete? |

C) Delete the customer you added, Abel Industries.

D) Delete the employee you added, Adam Wilson.

|  |
| --- |
| DISCUSS: Why is it important to include a WHERE clause with UPDATE and DELETE statements? |