

I used an INNER JOIN which selects only matching left and right table rows. This produced a table of first and last names for each of the four departments based on the condition:

* **Employee.Department\_ID = Department.Department\_ID**.

Both the Employee and Department table share the Department\_ID column. This was the cardinality between the two tables, which allowed access to all matching rows.

My CSV file contained eleven records.

Data Extraction refers to the process of retrieving data from one format into a more “useful” format for further processing (Perez, 2022). This was demonstrated in the query below, where custom matching information was extracted from 2 different tables.

Produce a table with the columns first name, last name, and Department\_Name on the Department table

**SELECT**

**First\_Name,**

**Last\_Name,**

**Department.Department\_Name**

from the matching rows of the Employee and Department table

**FROM**

**Employee**

**INNER JOIN**

**Department**

wherein the Department\_ID for both tables is either 2 or 3

**WHERE**

**Employee.Department\_ID = 3**

**OR**

**Employee.Department\_ID = 2**

Perez, M. (2022, February 2). *What is data extraction and what is it used for?* ParseHub. Retrieved November 11, 2022, from [What is Data Extraction and What is it Used For? | ParseHub](https://www.parsehub.com/blog/data-extraction/)