# Function Technical Sheet

## Function Name:

query\_ids\_year\_qtr

## Purpose:

Retrieve records for specific IDs within a year–quarter timeframe using multithreaded Oracle DB queries. Connects to an Oracle database via a session pool, splits input IDs into chunks, queries each chunk in parallel threads, and returns matching rows that meet both the ID and timeframe filters.

## Description:

The function filters data by a list of IDs and a year-quarter range. It uses multithreading to execute multiple queries in parallel, which improves performance for large datasets. Each row in the output contains ID, FIPS code, UI number, run identifier, wage, and a formatted year-quarter string.

## Parameters:

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| ids | list[int] or list[str] | List of IDs to match against the 'id' column. |
| start\_year | int | First year of the timeframe filter (inclusive). |
| start\_qtr | int | First quarter of the timeframe filter (1–4). |
| end\_year | int | Last year of the timeframe filter (inclusive). |
| end\_qtr | int | Last quarter of the timeframe filter (1–4). |
| dsn | str | Oracle DSN string in format 'host:port/service\_name'. |
| user | str | Oracle database username. |
| password | str | Oracle database password. |
| workers | int, default=6 | Number of parallel threads to run. |
| chunk\_size | int, default=1000 | Number of IDs per query chunk; keep ≤1000 for Oracle bind limits. |

## Returns:

list[tuple]: Each tuple contains (id, fips, UI, run, wage, yr\_qtr). yr\_qtr is formatted as 'YYYY-Q'. Example:  
[(101, '12345', 'UI001', 'run1', 55000, '2023-1'),  
 (205, '67890', 'UI002', 'run2', 62000, '2024-2')]

## Notes:

- Filtering is performed at numeric year/qtr level for correct ordering across years.  
- Multithreading is limited by the workers and SessionPool settings.  
- yr\_qtr can be zero-padded (YYYY-01) for better sorting.  
- Adjust chunk\_size based on DB bind limits and performance tests.