Lookup Function API

`connection = lookup.get_connection("name_of_lookup_table")`

Argument	Name of the lookup table, which is a concatenation of the name of the project or library containing the lookup table and the lookup table name, separated by an underscore.
Return	A connection object used to create delete, reader, range_reader, updater, and writer objects.

`deleter = connection:get_deleter()`

Argument	None.
Return	A deleter object used to delete content from the lookup table referenced by the connection object.

`reader = connection:get_reader("key")`

Argument	The name of a unique or non-unique lookup table key. A key may be composed of one or more attributes.
Return	A reader object used to read content from the lookup table referenced by the connection object.

`reader = connection:get_range_reader("key")`

Argument	The name of a unique or non-unique lookup table key. A key may be composed of one or more attributes.
Return	A reader object used to read content from the lookup table referenced by the connection object.

`updater = connection:get_updater()`

Argument	None.
Return	An updater object used to update content in the lookup table referenced by the connection object.

`writer = connection:get_writer()`

Argument	None.
Return	A writer object used to enter content into the lookup table referenced by the connection object.

`delete:execute(rowid_of_row_to_delete)`

Argument	The rowid or the target row. You must first use reader:execute to obtain the rowid.
Return	None.

`reader:execute({key=val[,…]})`

Argument	A string indexed datascript table in which the index name is the name of the key column (not the key name), and the value is the key value. If the key is comprised of multiple attributes, create a table argument with multiple elements.
Return	None.

`range_reader:execute({key=val[,…]})`

Argument	A string indexed datascript table in which the element name is the name of the key column (not the key name), and the value is the key value. If the key is comprised of multiple attributes, create a table argument with multiple elements.
Return	None.

`row, rowid = reader:next()`

Argument	None.
Return	If the result set returned by the `reader:execute` function is not empty, returns a row from the result set and the `rowid`. If a non-unique key was provided to the `reader:execute` function, the result set may include multiple entries. Invoke `reader:next` to obtain each row.

`rowid, message = updater:execute(rowid, {attribute=val[,…]})`

Arguments	The `rowid` of the row to update. A string indexed datascript table in which the element name is the name of a table attribute and the value is the updated attribute value. This table must include an element for each table attribute in the lookup table.
Return	If successful, the `rowid` of the updated row and an empty message string. If unsuccessful, a nil `rowid` and an error message.

`rowid, message = writer:execute({attribute=val[,…]})`

Argument	A string indexed datascript table in which the element name is the name of a table attribute and the value is the value of that attribute to be written to the table. This table must include an element for each table attribute in the lookup table.
Return	If successful, the `rowid` of the inserted row and an empty message string. If unsuccessful, a nil `rowid` and an error message.