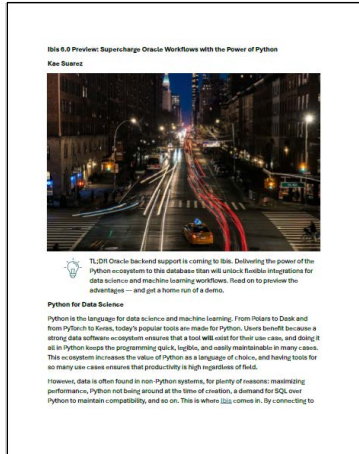




Ibis 6.0 Preview:

Supercharge Oracle Workflows with the Power of Python

Thank you for downloading this Voltron Data resource. Carahsoft is the distributor for Voltron Data's AI and ML solutions available via NASA SEWP V, ITES-SW2, NASPO ValuePoint, and many more contract vehicles.

To learn how to take the next step toward acquiring Voltron Data's solutions, please check out the following resources and information:



For Voltron Data overview:
carah.io/Voltron-Data



For upcoming events:
carah.io/Voltron-Events



For additional resources:
carah.io/Voltron-Resources



For additional Artificial Intelligence:
carah.io/ai-solutions



To set up a meeting:
VoltronData@carahsoft.com



To purchase, check out the contract vehicles available for procurement:
carah.io/procurement

Ibis 6.0 Preview: Supercharge Oracle Workflows with the Power of Python

Kae Suarez



TL;DR Oracle backend support is coming to Ibis. Delivering the power of the Python ecosystem to this database titan will unlock flexible integrations for data science and machine learning workflows. Read on to preview the advantages — and get a home run of a demo.

Python for Data Science

Python is the language for data science and machine learning. From Polars to Dask and from PyTorch to Keras, today's popular tools are made for Python. Users benefit because a strong data software ecosystem ensures that a tool **will** exist for their use case, and doing it all in Python keeps the programming quick, legible, and easily maintainable in many cases. This ecosystem increases the value of Python as a language of choice, and having tools for so many use cases ensures that productivity is high regardless of field.

However, data is often found in non-Python systems, for plenty of reasons: maximizing performance, Python not being around at the time of creation, a demand for SQL over Python to maintain compatibility, and so on. This is where [Ibis](#) comes in. By connecting to

18+ engines, ranging from MS SQL to Snowflake, Ibis offers the Python ecosystem — and its own powerful API for analytics — to a variety of applications. Soon, it'll support another one, and you may recognize it.

Oracle Database Support in Ibis

Oracle powers countless enterprises, from [FedEx to AT&T and GE](#). Presumably, billions of dollars are managed through these systems, and for good reason: it's powerful, well-maintained, and has a long history. It keeps up with the times, too — it's easier than ever to deploy Oracle Databases in the cloud.

With Ibis's 6.0 release, you can now leverage all that power using Python — no SQL required. Oracle users will be able to access the power of Ibis's API for data analytics, paired with the Python data ecosystem to enable quick and easy creation of new workflows and use cases. Not only does this bring the raw power of the ecosystem, but it also ensures that Python developers can access the power of Oracle Database while staying in their language of choice — enabling them to focus on logic instead of syntax, optimizing their productivity.



TL;DR The Oracle backend support is experimental, and uses the `oracledb` thin client driver, tested against Oracle Database 23. Other versions are not presently tested in CI, and your mileage may vary.

Since Oracle support in Ibis is an upcoming feature, the build is not on PyPi or conda-forge. If you want to get a sneak peek on your own, install a development build using the instructions in the [Ibis documentation](#). If you just want to see how it looks in practice, read on!

For this example, we'll use a [container hosted locally developed by Gerald Venzl](#). This is a fantastic resource for testing out Oracle Database and is even used in the Ibis CI for the Oracle backend. Furthermore, we'll use the database preloaded with the baseball data used in the Ibis CI — reading data into the database via Ibis is not currently recommended, as Oracle Database has constraints on how data ingest works that make preloading preferable.

First, as with every backend, we import Ibis, set it to interactive mode, then connect with our credentials:

PYTHON

```
import ibis
ibis.options.interactive = True
con = ibis.oracle.connect(<credentials>)
```

Then, it's just like any other use of Ibis, this time powered by Oracle Database. We'll find which tables we have:

PYTHON

```
con.list_tables()

OUT:
['functional_alltypes', 'win', 'diamonds', 'batting', 'awards_players']
```

Let's go ahead and get the awards_players table, and see what there are.

PYTHON

```
t = con.tables["awards_players"]
t
```

OUT:

playerID	awardID	yearID	lgID	tie	notes
string	string	int64	string	string	string
sislege01	Most Valuable Player	1922	AL	NULL	NULL
hornsro01	Triple Crown	1922	NL	NULL	NULL
pippwa01	Baseball Magazine All-Star	1923	AL	NULL	1B
collied01	Baseball Magazine All-Star	1923	AL	NULL	2B
duganjo01	Baseball Magazine All-Star	1923	AL	NULL	3B
perkicy01	Baseball Magazine All-Star	1923	AL	NULL	C
heilhma01	Baseball Magazine All-Star	1923	AL	NULL	OF
ruthba01	Baseball Magazine All-Star	1923	AL	NULL	OF
speaktr01	Baseball Magazine All-Star	1923	AL	NULL	OF
uhlege01	Baseball Magazine All-Star	1923	AL	NULL	P1
...	...	...	...	...	...

Now, just to do something with the data, let's see how many times the "Baseball Magazine All-Star" award appears.

PYTHON

```
t.count(t.awardID == "Baseball Magazine All-Star")
```

OUT:

1520

It's simple, definitely, but it's also quick and Pythonic, which is the point. Ibis is the entire interface — once you're connected, you get Ibis's interface with Oracle Database's capabilities.

Onwards to Ibis 6.0

This is just one of many exciting additions to look forward to in Ibis 6.0, and opens new horizons for countless users. We look forward to showing off what you can do with Oracle and Ibis as soon as 6.0 is released. If you want to keep track of what's coming next with Ibis, be sure to watch their Github at <https://github.com/ibis-project/ibis>, and check out the 6.0 milestones!

Ibis is just one of the always-growing open source projects that we support at Voltron Data. To get more resources, check out our [Ibis page](#) or explore our [Product](#) page to see how we help enterprises design and build composable data systems.