Background and Significance

Database management systems (DBMSs) have been widely used ever since they first gained popularity in the 1960s. From allowing programmers to efficiently operate databases to easing the way in which consumers access information, they have a wide range of applications. Due to the increased productivity and performance attributed to the introduction of these management systems, DBMSs have become an essential part of many aspects of society. As new implementations are constantly pushed forward, DBMSs have become increasingly complex with endless additions to their features and functionalities, and consequently the demand for man-power to create and maintain the databases has also risen. The combination of growing intricacy within DBMSs alongside the requirement of humans to manage, update and troubleshoot them poses a major dilemma for database administrators. This hurdle can be overcome with the use of autonomous databases. The goal of autonomous databases is to remove the need for human intervention by implementing machine learning to perform all tasks involved in developing, deploying and maintaining a database. As a result, significant factors such as cost, effort, time, and complications associated with these tasks would be significantly decreased.

Although there has been research put into developing full autonomous DMBSs, no database was fully autonomous prior to 2018. Humans were still required to provide input to the DBMS, including making final decisions about the database and fixing any problems. Any research put into automating a DBMS targeted a sole feature of the system, such as deciding index configurations or the physical design. Although these improvements facilitate the use of a DBMS, full automation of the system was not achieved.

In 2018, Oracle released the first completely autonomous database, the Oracle Autonomous Database Warehouse (OADW), which is the software we will be focusing on for this project. There are three main advantages OADW provides, which entices consumers to choose this autonomous DBMS over traditional ones.

1. **OADW is completely self-driving**. Users simply denote the desired output (level of service), and the system handles the rest: provisioning, security, diagnostics, scheduled backups, and database recovery.
2. **OADW is smarter.** The system is able to automatically handle efficient index creation, optimize index configurations, as well as regular database tuning via Machine Learning algorithms. Furthermore, it is able to accept both OLTP & OLAP workloads to maximize flexibility of usage. OADW also provides a more secure environment compared to a manual DBMS. Instead of relying on a database administrator to protect the system against attacks, which can be time-dependent, OADW does so itself via security patches, and always-on encryption. In contrast to a manual DBMS whose security updates are implemented by hand, and therefore rarely, OADW does it automatically every quarter.
3. **OADW is more reliable**. Some of the features that enhance dependability significantly include the ability to automatically restore from physical failures, rewind data to any point in the past, back out user errors, remain online during updates, and run nightly backups. As the demand and preference for autonomous software increases, these functionalities bring OADW ahead of the game against manual databases.

Project Goal and Objectives

OADW provides a number of functionalities to increase user productivity and enhance overall performance. We will be investigating these capabilities by deploying the Palmer Penguin Dataset as our case study, and smaller datasets if necessary to maximize our analysis of the DBMS features. The following features have been selected by our group to be further investigated and evaluated in our case study:

Machine Learning
The OADW provides a number of functions to support machine learning endeavors, including a number of high-performance algorithms, and Oracle machine learning notebooks to increase efficiency of machine learning model creation. Our case study will evaluate the usability and functionality of the interfaces provided by Oracle, as well as assess other provided features such as model tuning and their analytics workflow builder.

Graph Analytics
A complete graph database from Oracle allows users to apply pattern recognition, and data classification without the need to migrate their database. We will explore various features such as the automated graph modeling and visualization, prebuilt algorithms for data relationship exploration, and the sample notebooks together with workflows provided by OADW using our dataset.

Self-Service Data Management
Data exploration is a crucial step when working with large datasets. Users can leverage the various tools provided by OADW to assist in data manipulation, pattern recognition, and identification of unreliable samples. We will evaluate the efficiency and accuracy of these in-database tools in managing our dataset.

Data Protection and Security
OADW prioritizes ensuring data security by providing users with automated security controls, and tools to monitor activity as well as evaluate risks. Our group will investigate the ability of the DBMS to evaluate the database's security, and categorize risks through the provided security assessments.

Methodology & Approach

The following approaches will be used to accomplish the four objectives of our project. (1) Machine Learning: We will build machine learning models and test their ability to effectively classify the samples in our dataset based on the available features. Specifically, we will use the penguin attributes provided in the dataset to train the machine to determine the penguin species. (2) Graph Analytics: We will explore and visualize the relationships between the features of our dataset using the in-house functions provided. (3) Self-Service Data Management: We will compare the tools provided by OADW with manual data exploration done externally (e.g. using provided functions in R). (4) Data Protection and Security: We will run the security assessments on our database following scheduled tasks that mimic real-life use scenarios.Through these approaches, we will be able to see the advantages of Oracle in action.

Resources and Tools

Dataset

Our case study which will be the central focus of our final report and class presentation will be done using the Palmer Penguin dataset (https://github.com/allisonhorst/palmerpenguins)

Reading materials

- Oracle Database Website: https://www.oracle.com/autonomous-database/autonomous-data-warehouse/
- OADW Key Features: https://www.oracle.com/autonomous-database/
- Oracle Autonomous Database Strategy: https://www.oracle.com/a/ocom/docs/database/oracle-autonomous-database-strategy-wp.pdf

Tutorials

- OAWD Tutorials: https://docs.oracle.com/en/cloud/paas/autonomous-data-warehouse-cloud/tutorials.html

Documentation

- Oracle Database Documentation: https://docs.oracle.com/en/database/oracle/oracle-database/

Tools

The investigation and analyses for our case study will predominantly involve the in-database tools and functions provided by OADW. Python v.3.10. and R v.4.2.1 will be used for machine learning (via the Python interface provided by OADW) and data manipulation respectively.

Functionalities and Features

The main feature that sets Oracle Autonomous Database ahead of other database management systems is the fact that it is autonomous, meaning the necessary tasks to maintain the database can be performed without human intervention. The most significant functionalities that are automated include database optimization via machine learning, machine learning for data science, graph analytics, self-service data management, and data protection & security. We decided to explore and test these main features that make the Oracle Autonomous Database autonomous.

DATA OPTIMIZATION BY MACHINE LEARNING

Database optimization is a crucial component of managing a database. Traditionally, database administrators (DBAs) constantly spend time monitoring database usage diagnostics to maintain index performance, manage tablespace storage, create table partitions for performance, and many other tuning-related tasks. This is all done manually by the DBA and takes up much of their available time. DBAs must consistently read logs and identify possible points for improvement by modifying configurations, allocating additional resources, and taking the database down for maintenance to apply required changes. As databases live on, the amount of tuning required increases as stored data grows larger and larger – increasing the work required by the DBA which leads to the importance of an automated solution.

Within the Oracle Autonomous Database, the task of database tuning is largely managed by autonomous means via machine learning. Machine learning is essentially a black box where learning algorithms are provided an input and can optionally be given an output target with examples to replicate or left to derive their own output, where everything in-between is left up to artificial intelligence processes to figure out how to accurately reach the output. Oracle provides several learning algorithms that are applied in the background as scheduled jobs to learn from statistics routinely generated by the database in order to reach a target output of efficient performance. By using database usage statistics as a set for training data, these self-learning algorithms can identify common patterns for improved performance, make fault predictions, and handle errors derived from the database. This allows the autonomous database to massively cut down on manual input from the DBA in areas like provisioning, configuration, scaling, error detection and resolution, as the machine learning algorithms can constantly check for issues and test improvements to keep the database working at its best.

A great example of this process in action is the automated indexing feature provided by Oracle, where a DBA can run a simple SQL command to enable automatic creation of indices for database performance. Once enabled, the database has machine learning algorithms that routinely check the workload of the database to identify areas where new indices could potentially improve performance, as well as detect when old indices should be replaced due to deteriorated efficiency (modified table structure, larger table size, change in query usage, etc.). After identifying tables for improvement, the table statistics are then checked for commonly run SQL statements to provide test cases for new automated index candidates to determine whether they improve or regress performance for the workload.

Machine learning algorithms generate these automated index candidates (by default invisible to the query optimizer) by identifying missing indices from the table, validating that these candidates provide some improvement, and learning from their own mistakes when candidates regress performance. Candidates are then tested against the entire sample of test SQL cases generated by table statistics to ensure they improve overall performance by comparing query plans against the current configuration of the table. If found to improve performance, the automated index candidate is then deployed and fully visible for use by the database as a B-tree index. However, it is important to remember that indices may not improve performance for all query operators – for which a failsafe is included, where candidates who were found to improve performance in one area but regress performance in the other are specifically excluded from being used by those query operators. This is done by creating an Oracle SQL plan baseline, which manages available query execution plans for use by the query optimizer to ensure these automated indices can be used only in areas they improve. The entire process is done entirely automatically and behind the scenes, saving DBAs an immense amount of work and effort from identifying missing indices, creating test cases for performance evaluation, and managing the analysis of evaluating these new indices. Analytic views are provided for logging the creation of automatic indices, reporting their performance, index verification, etc. if the DBA wants to keep track of the automated index processes.

By providing machine learning algorithms with the same data DBAs have access to, the Oracle Autonomous Database can optimize many aspects of the database in ways like the example above. Determining when a database may need additional deployment of servers in times of increased traffic, improving SQL efficiency by automated indices and table partitions, managing optimal configurations for storage and data formats, and error handling for hardware failures are just some of the benefits provided by machine learning for database optimization. As databases grow larger to accommodate more data and demand from users, machine learning techniques will be able to utilize larger amounts of statistics to further fine tune performance and save DBAs from the burden of manual optimization.

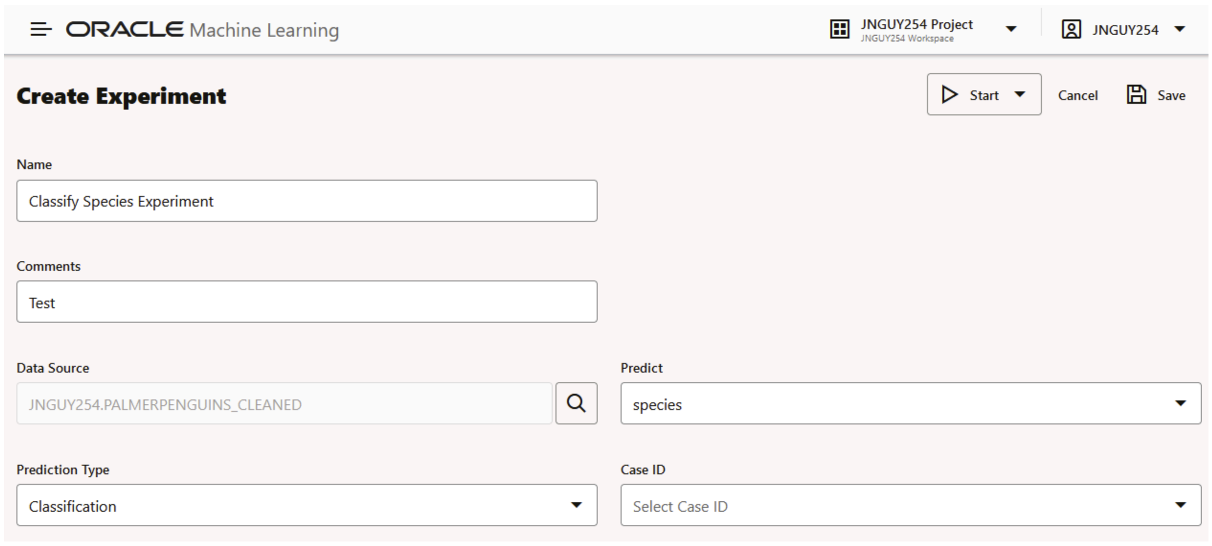
MACHINE LEARNING FOR DATA SCIENCE

Adjacent to the management of the database, another crucial component is the actual use of data within a database to fuel data science focused workflows. Being able to easily extract insights, perform query operations, and identify outliers from the database are integral parts of the data science process, and are typically done manually by data scientists in the traditional database.

The machine learning algorithms provided by the Oracle Autonomous Database allow for a streamlined approach to these data science processes including data exploration, preparation, and modelling. These algorithms are able to enhance the workflow by performing the heavy lifting in finding connections and patterns within the dataset, which allows data scientists to spend more time with the results from data rather than spending time figuring out how to extract analytics from the database. A common use case for this would be creating a predictive model from a dataset to predict future outcomes on similar sets of data, which could range from classifying data rows based on specific factors or determining an average value for a column based upon historical existing data. The Oracle Machine Learning experiments tool “AutoML” provides exactly this type of functionality, where a user is able to define a table from the database as an input source, denote which column from that table they would like to predict as an output, and through which predictive method they would like between classification and regression (Fig. 1).

In our case, we wanted to take advantage of the machine learning components from the Oracle Autonomous Database to create a model that could classify penguin species correctly based on the samples provided by the Palmer Penguin dataset.

Create Experiment UI

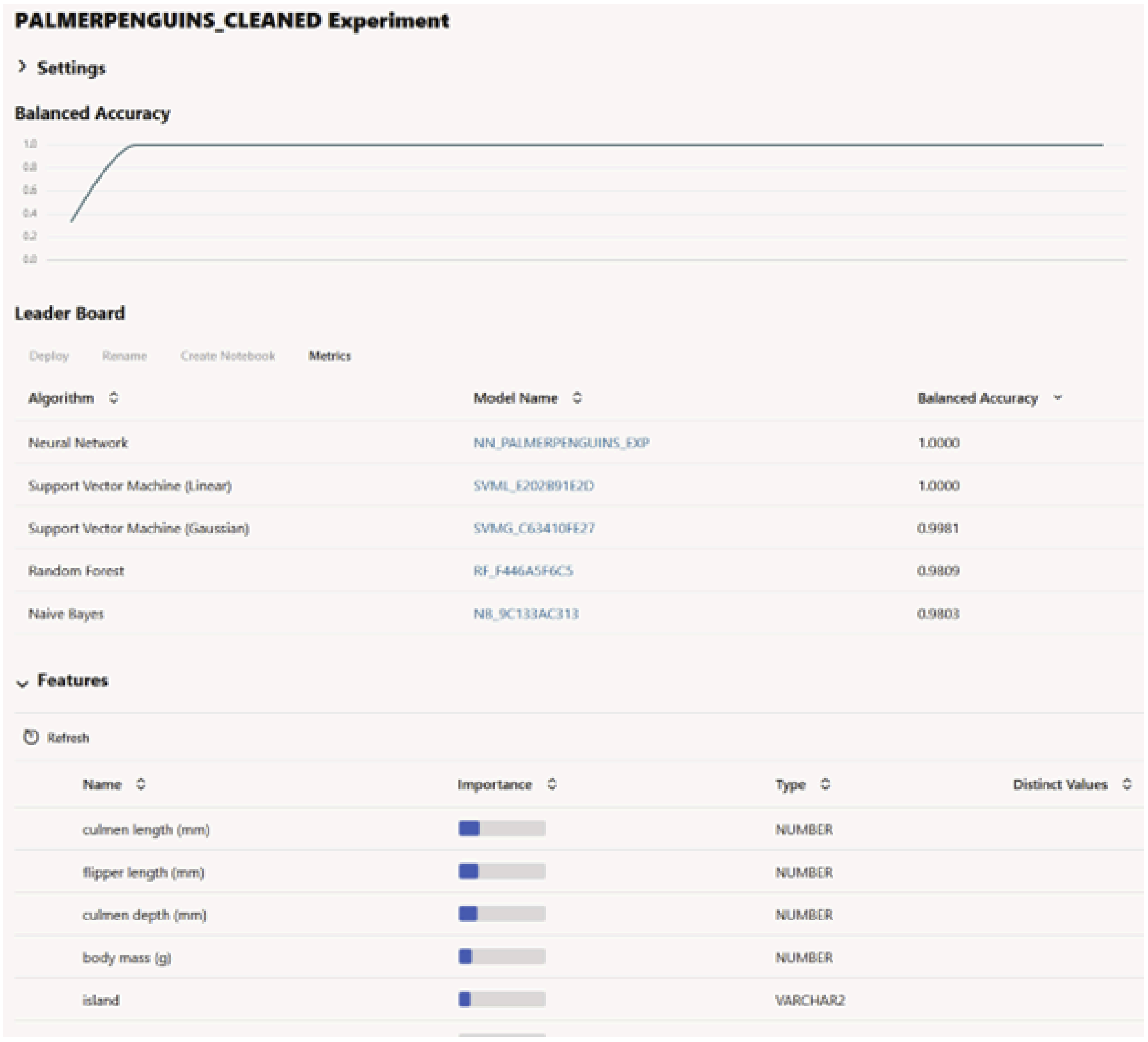


Create Experiment

Name	Classify Species Experiment
Comments	Test
Data Source	JNGUY254.PALMERPENGUINS_CLEANED
Predict	species
Prediction Type	Classification
Case ID	Select Case ID

After successfully running the AutoML experiment, we were shown a leaderboard ranking the various machine learning algorithms (neural networks, random forest, support vector machines, etc.) in how accurate they were in predicting our target column, as well as how important each column from the table was in determining the prediction (Fig. 2). From here, users can also select any model from the leaderboard to see further details on their prediction weighting per column (Fig. 3a) as well as their confusion matrix for the various classifications of the experiment (Fig. 3b). In our case, the neural network algorithm was ranked the highest for accuracy and we were able to save this algorithm as a prediction model for use within our Oracle notebooks that provides an environment to run code in Python, SQL, R, and more.

Experiment Leaderboard



PALMERPENGUINS_CLEANED Experiment

› Settings

Balanced Accuracy

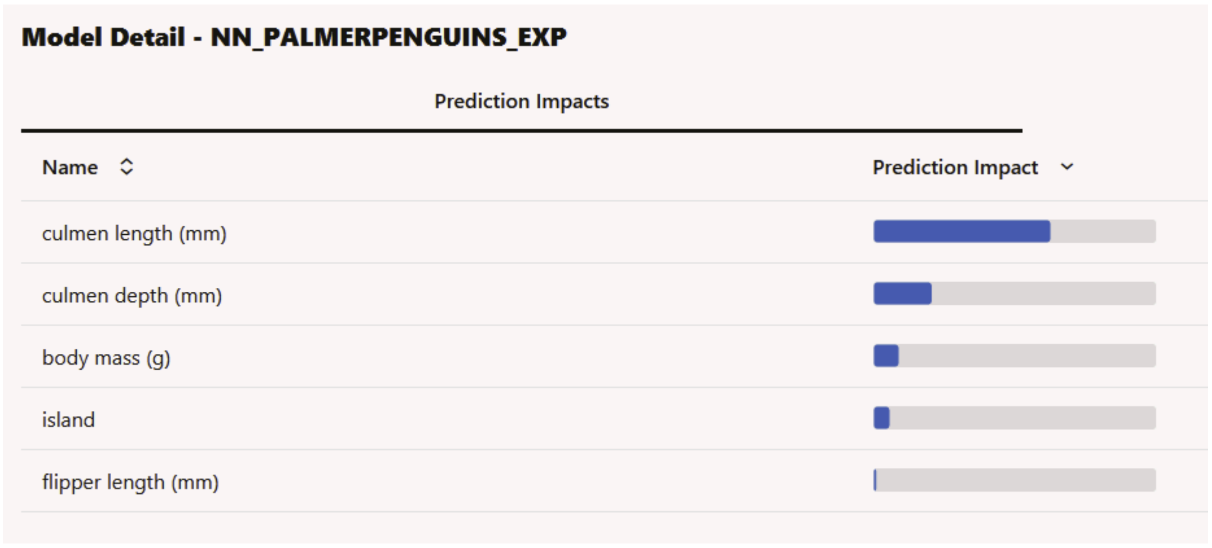
Leader Board

Algorithm	Model Name	Balanced Accuracy
Neural Network	NN_PALMERPENGUINS_EXP	1.0000
Support Vector Machine (Linear)	SVML_E202891E2D	1.0000
Support Vector Machine (Gaussian)	SVMG_C63410FE27	0.9981
Random Forest	RF_F446A5F6C5	0.9809
Naive Bayes	NB_9C133AC313	0.9803

⌄ **Features**

Name	Importance	Type	Distinct Values
culmen length (mm)	▨	NUMBER	
flipper length (mm)	▨	NUMBER	
culmen depth (mm)	▨	NUMBER	
body mass (g)	▨	NUMBER	
island	▨	VARCHAR2	

Model Details - Prediction Impacts of Columns

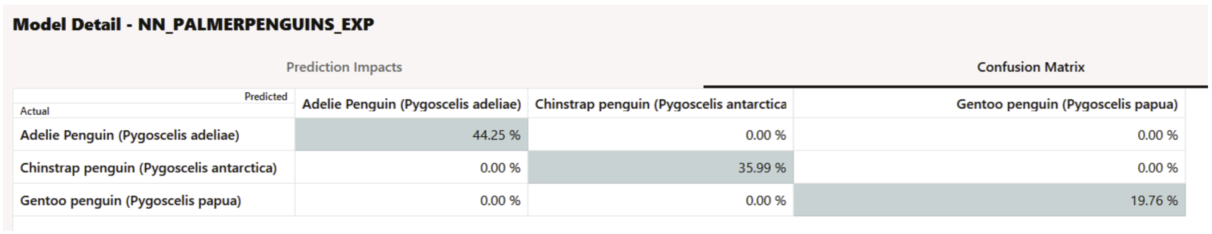


Model Detail - NN_PALMERPENGUINS_EXP

Prediction Impacts

Name	Prediction Impact
culmen length (mm)	
culmen depth (mm)	
body mass (g)	
island	
flipper length (mm)	

Model Details - Confusion Matrix of Classifications



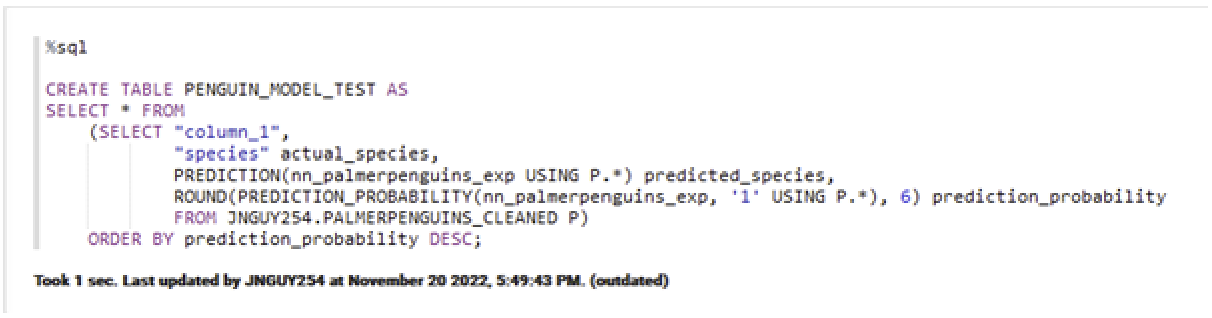
Model Detail - NN_PALMERPENGUINS_EXP

Prediction Impacts   Confusion Matrix

Actual \ Predicted	Adelie Penguin (Pygoscelis adeliae)	Chinstrap penguin (Pygoscelis antarctica	Gentoo penguin (Pygoscelis papua)
Adelie Penguin (Pygoscelis adeliae)	44.25 %	0.00 %	0.00 %
Chinstrap penguin (Pygoscelis antarctica)	0.00 %	35.99 %	0.00 %
Gentoo penguin (Pygoscelis papua)	0.00 %	0.00 %	19.76 %

Using this model within notebooks, we were able to run an SQL create table statement that utilized the predictive capabilities of the model to generate its own predictions for a species column given an input dataset (Fig. 4). Finally, we were successful in using our predictive model to generate a table that accurately predicted the species column for the Palmer Penguin dataset while only using the culmen length, flipper length, column depth, body mass, and island columns (Fig. 5).

SQL to Create Table via Predictive Model

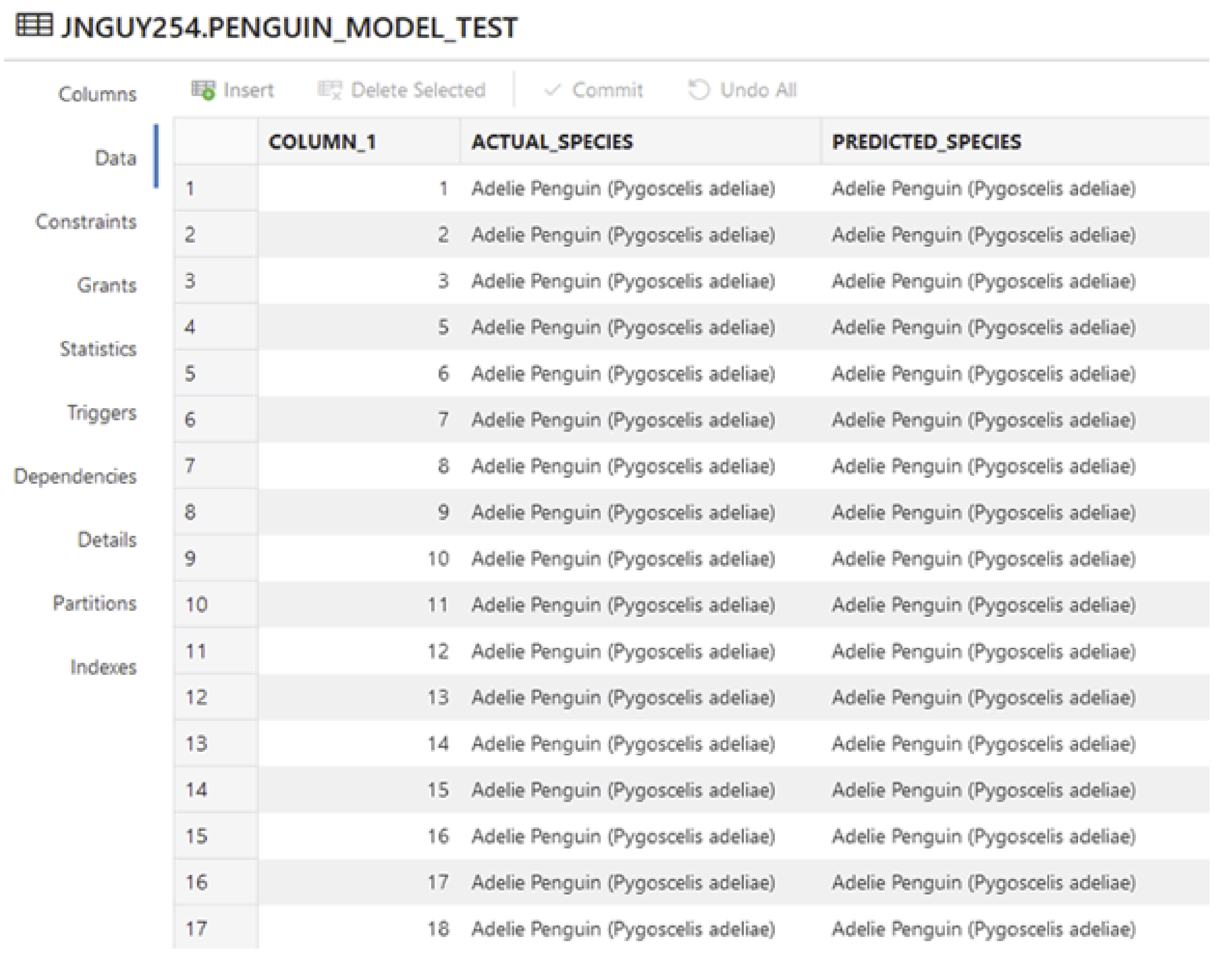


```
%sql

CREATE TABLE PENGUIN_MODEL_TEST AS
SELECT * FROM
    (SELECT "column_1",
            "species" actual_species,
            PREDICTION(nn_palmerpenguins_exp USING P.*) predicted_species,
            ROUND(PREDICTION_PROBABILITY(nn_palmerpenguins_exp, '1' USING P.*), 6) prediction_probability
            FROM JNGUY254.PALMERPENGUINS_CLEANED P)
    ORDER BY prediction_probability DESC;
```

Took 1 sec. Last updated by JNGUY254 at November 20 2022, 5:49:43 PM. (outdated)

Results



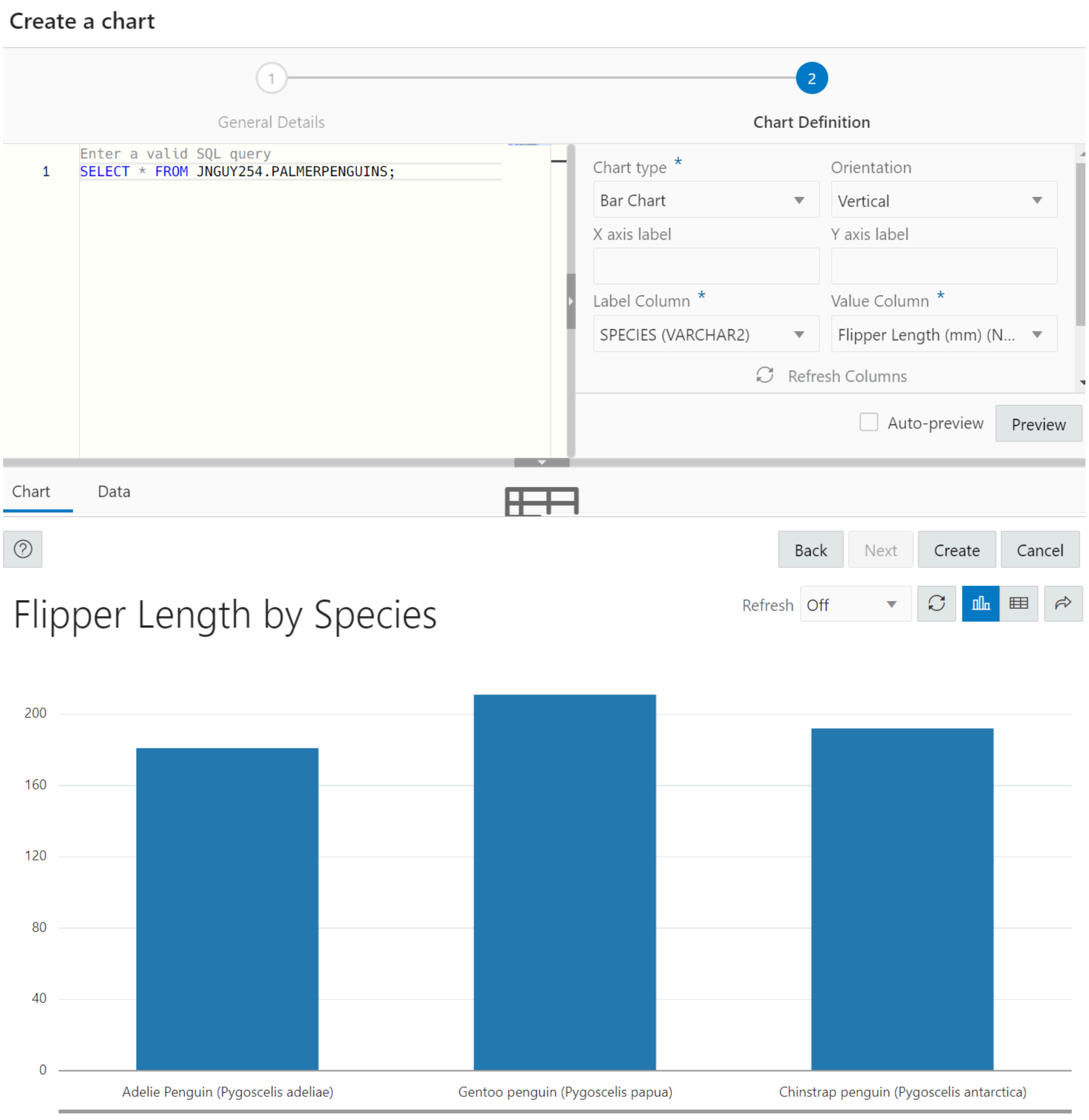
JNGUY254.PENGUIN_MODEL_TEST

	COLUMN_1	ACTUAL_SPECIES	PREDICTED_SPECIES
1	1	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
2	2	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
3	3	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
4	5	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
5	6	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
6	7	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
7	8	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
8	9	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
9	10	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
10	11	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
11	12	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
12	13	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
13	14	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
14	15	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
15	16	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
16	17	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)
17	18	Adelie Penguin (Pygoscelis adeliae)	Adelie Penguin (Pygoscelis adeliae)

GRAPH ANALYTICS

As more and more data is being generated, analyzed, and maintained, there is an increasing demand for convenient yet effective methods to capture important details of the data visually. In-house visualization tools are provided by Oracle Autonomous Database to facilitate the relationship-based analysis of data without transferring data entities to external platforms. A user-friendly interface enables graph features for ranking, clustering, and mapping out patterns in the data without the need of extensive coding or direct manipulation from the database user. The applications of these quick visuals include the identification of anomalies and trends, classifying groups, and discovering associations in the data. They are also beneficial for professional charts to facilitate the dissemination of information.

The graphing platform supports a wide variety of chart types, including area charts, bar charts, funnels, line graphs, pie charts, and pyramid charts. The user must enter an SQL statement to indicate the table containing the attributes of interest. Other parameters to specify include the type of chart to create, the attribute(s) to graph, and a few customizable details such as labels and chart orientation (Fig. 6). A tabular view of the data is made readily available with the option to export alongside the graphic (Fig. 7).


Create a chart

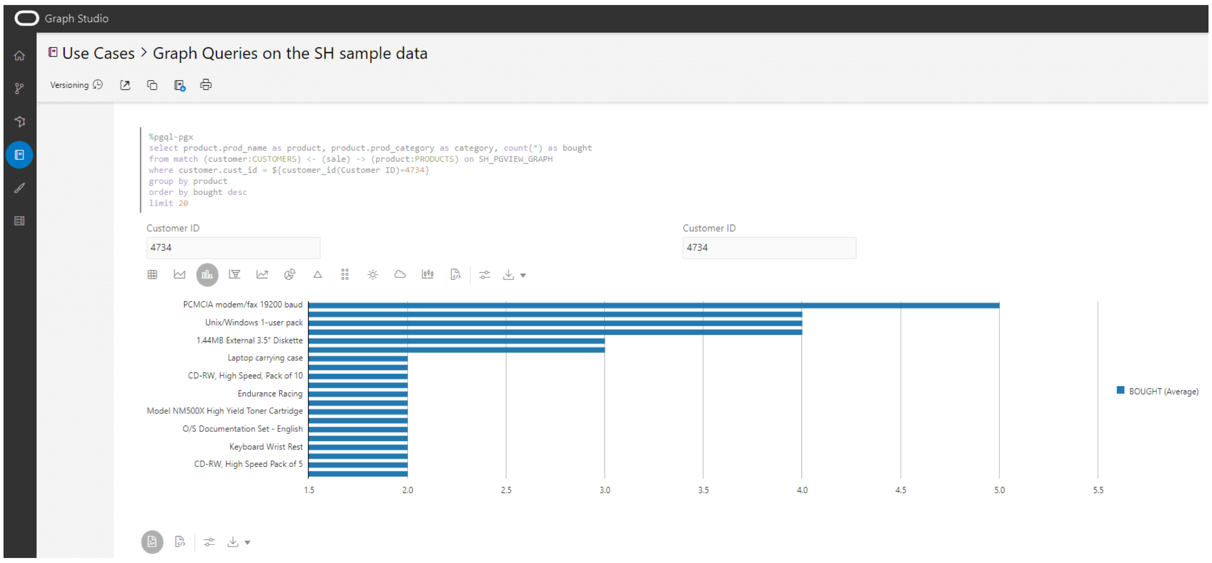
1 General Details — 2 Chart Definition

```
Enter a valid SQL query
SELECT * FROM JNGUY254.PALMERPENGUINS;
```

Chart type *	Bar Chart
Orientation	Vertical
X axis label	
Y axis label	
Label Column *	SPECIES (VARCHAR2)
Value Column *	Flipper Length (mm) (N...

Refresh Columns
Auto-preview
Preview
Chart
Data
Back
Next
Create
Cancel
Flipper Length by Species
Refresh
Off
200
160
120
80
40
0
Adelie Penguin (Pygoscelis adeliae)
Gentoo penguin (Pygoscelis papua)
Chinstrap penguin (Pygoscelis antarctica)

Oracle Graph Studio is another platform included in the Oracle Autonomous Database that enables further customization of created visuals. Users are required to use relational query language-based commands to draw graphs using their data. From there, users are provided a range of graph types to select from, examples include area charts, bar plots, funnels, line graphs, pie charts, etc (Fig. 8a). The option to view the data in tabular and json formats are also available for viewing and export (Fig. 8b).

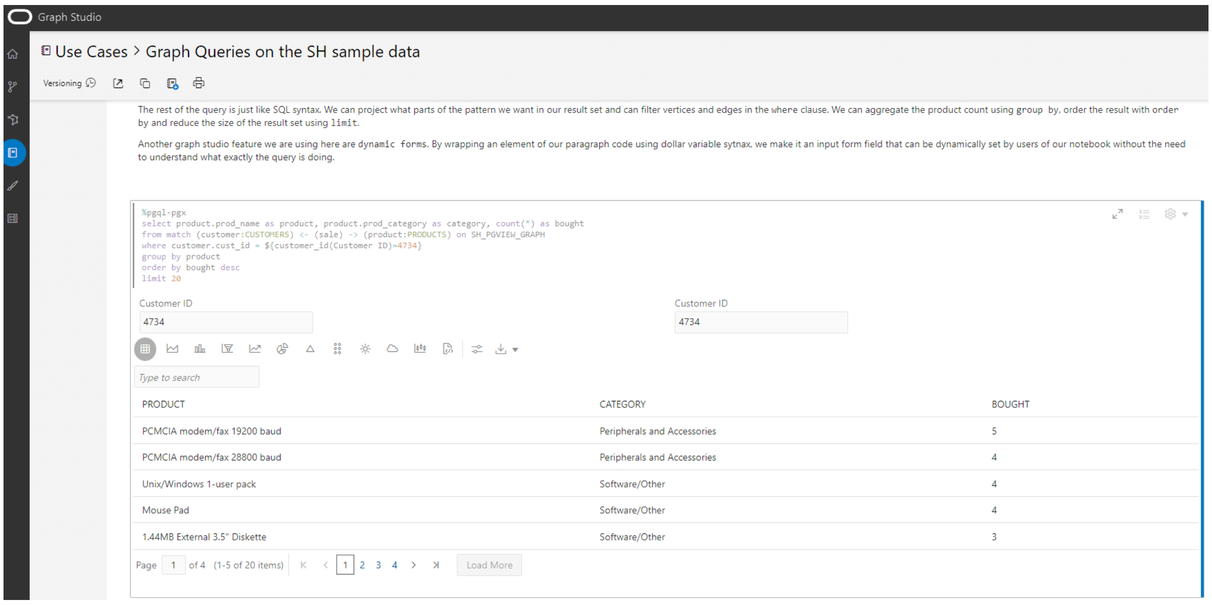

Graph Studio
Use Cases > Graph Queries on the SH sample data
Versioning

```
%pgql-pgx
select product.prod_name as product, product.prod_category as category, count(*) as bought
from match (customer:CUSTOMERS) <- (sale) -> (product:PRODUCTS) on SH_PGVIEW_GRAPH
where customer.cust_id = ${customer_id(Customer ID)=4734}
group by product
order by bought desc
limit 20
```

Customer ID: 4734

Customer ID: 4734

PCMCIA modem/fax 19200 baud
Unix/Windows 1-user pack
1.44MB External 3.5" Diskette
Laptop carrying case
CD-RW, High Speed, Pack of 10
Endurance Racing
Model NM500X High Yield Toner Cartridge
O/S Documentation Set - English
Keyboard Wrist Rest
CD-RW, High Speed Pack of 5
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
BOUGHT (Average)



The rest of the query is just like SQL syntax. We can project what parts of the pattern we want in our result set and can filter vertices and edges in the `where` clause. We can aggregate the product count using `group by`, order the result with `order by` and reduce the size of the result set using `limit`.

Another graph studio feature we are using here are `dynamic forms`. By wrapping an element of our paragraph code using dollar variable sytnax, we make it an input form field that can be dynamically set by users of our notebook without the need to understand what exactly the query is doing.

```
%pgql-pgx
select product.prod_name as product, product.prod_category as category, count(*) as bought
from match (customer:CUSTOMERS) <- (sale) -> (product:PRODUCTS) on SH_PGVIEW_GRAPH
where customer.cust_id = ${customer_id(Customer ID)=4734}
group by product
order by bought desc
limit 20
```

Customer ID: 4734

Customer ID: 4734

PRODUCT	CATEGORY	BOUGHT
PCMCIA modem/fax 19200 baud	Peripherals and Accessories	5
PCMCIA modem/fax 28800 baud	Peripherals and Accessories	4
Unix/Windows 1-user pack	Software/Other	4
Mouse Pad	Software/Other	4
1.44MB External 3.5" Diskette	Software/Other	3

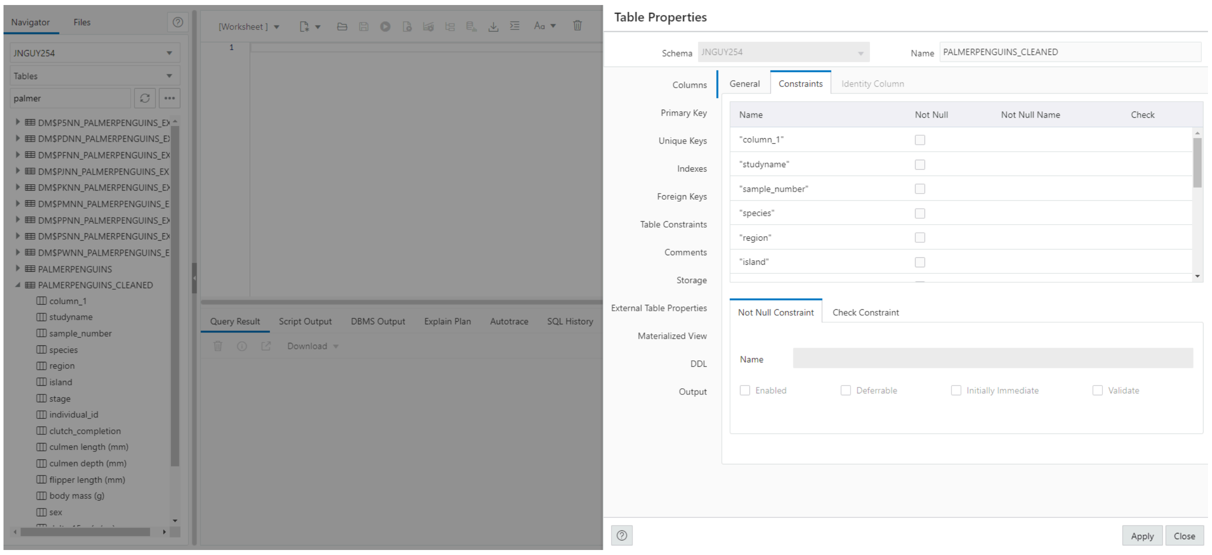
SELF-SERVICE DATA MANAGEMENT

Data exploration is a fundamental and crucial early step before any further data usage. The conventional procedures to convert data across various formats and/or identify and manage anomalies often require extensive user inputs and involvement. These processes are often time-consuming and vulnerable to human error. Oracle Autonomous Database supports direct manipulation and investigation of data from the database without transferring to external platforms, granting both increased convenience and security for database users. Investigations can be submitted as jobs to run automatically without human interference. This effectively improves productivity and efficiency.

Data Manipulation

Oftentimes, data is not immediately usable for queries and other investigations without data manipulation. Certain configuration parameters are necessary for performing a variety of actions on the data, some of which are either not set previously or lost during data transfer and population into the database.

Oracle Autonomous Database provides an SQL workbench for manipulation of tables and data values. The interface provides an SQL query box for those with familiarity who may prefer to work from a coding environment. There is also a less complex coding window available for users without SQL knowledge (Fig. 8). This view provides a more accessible option to manipulate data tables. Features available for customization include setting up attribute details (e.g. NOT NULL), identifying primary keys, indexes, foreign keys, constraints, etc.



JNGUY254
Tables
palmer

- DM$P5NN_PALMERPENGUINS_EX
- DM$PDNN_PALMERPENGUINS_EX
- DM$PFNN_PALMERPENGUINS_EX
- DM$PJNN_PALMERPENGUINS_EX
- DM$PKNN_PALMERPENGUINS_EX
- DM$PMNN_PALMERPENGUINS_E
- DM$PPNN_PALMERPENGUINS_EX
- DM$PSNN_PALMERPENGUINS_EX
- DM$PWNN_PALMERPENGUINS_E
- PALMERPENGUINS
- PALMERPENGUINS_CLEANED
  - column_1
  - studyname
  - sample_number
  - species
  - region
  - island
  - stage
  - individual_id
  - clutch_completion
  - culmen length (mm)
  - culmen depth (mm)
  - flipper length (mm)
  - body mass (g)
  - sex

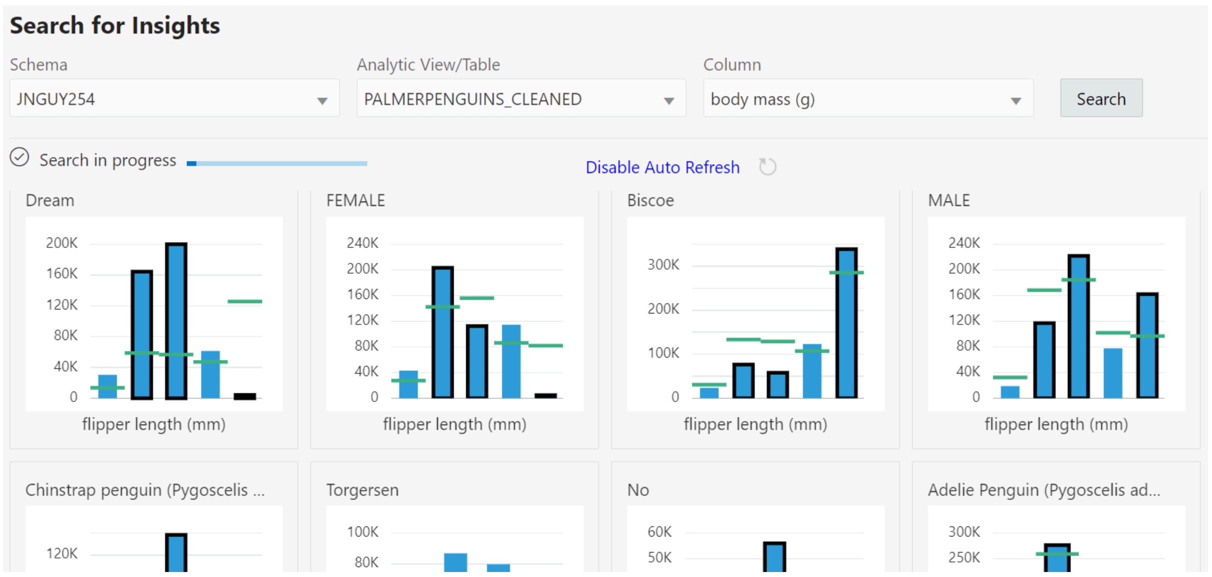
Schema: JNGUY254 Name: PALMERPENGUINS_CLEANED

Name	Not Null	Not Null Name	Check
"column_1"	☐		
"studyname"	☐		
"sample_number"	☐		
"species"	☐		
"region"	☐		
"island"	☐		

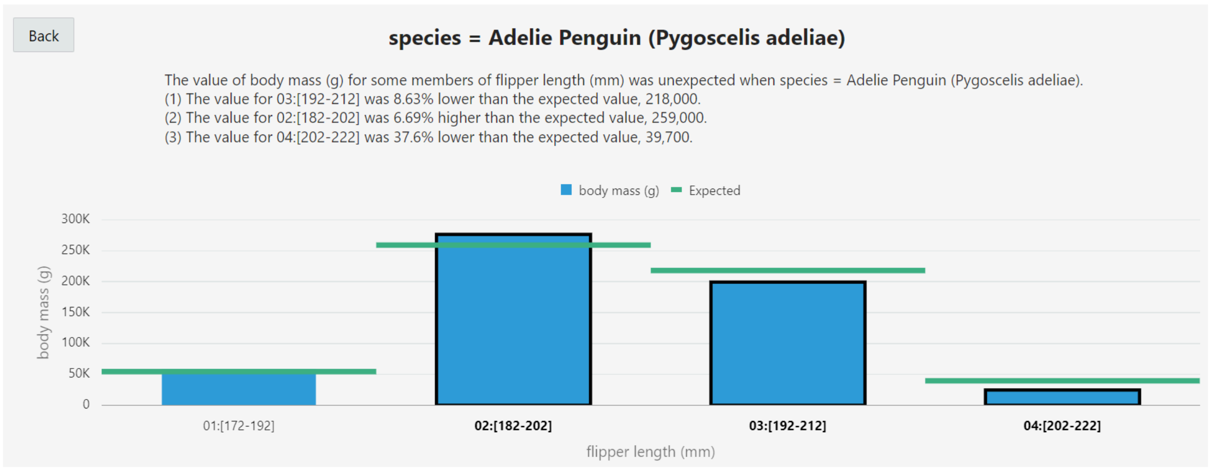
Name:
☐ Enabled ☐ Deferrable ☐ Initially Immediate ☐ Validate

Data Insights

This feature makes use of a variety of analytic functions built into the Oracle Autonomous Database to capture important details from the data. The page will highlight anomalies and unexpected patterns in the pairs of attributes. Users are able to select their column of interest, and the function will automatically generate a series of bar charts that displays values of the selected measure across all pairs of hierarchy or attributes (Fig. 9a). The page will also generate estimates of expected values for the measure to use as benchmarks (Fig. 9b).



Schema: JNGUY254
Analytic View/Table: PALMERPENGUINS_CLEANED
Column: body mass (g)



The value of body mass (g) for some members of flipper length (mm) was unexpected when species = Adelie Penguin (Pygoscelis adeliae).
(1) The value for 03:[192-212] was 8.63% lower than the expected value, 218,000.
(2) The value for 02:[182-202] was 6.69% higher than the expected value, 259,000.
(3) The value for 04:[202-222] was 37.6% lower than the expected value, 39,700.

In comparison to external platforms for data exploration, such as RStudio, this Oracle Autonomous Database function is much more user-friendly and efficient. To achieve similar results using RStudio, a linear modelling loop would have to be created with results subsequently graphed. This required greater amounts of memory and was not comparably efficient in terms of time costs.

<u>DATA PROTECTION & SECURITY</u>

Security

Database security refers to the preservation of data confidentiality and integrity. It is important for a database to have a collection of tools and measures in place to enforce database security as well as prevent data breaches. Enterprises must ensure their database maintains data confidentiality as breaches to data can compromise brand-related information, property, and reputation. Threats to data security can come from direct human effort or malware, and it is essential for databases to be able to prevent malicious attacks from all entry points.

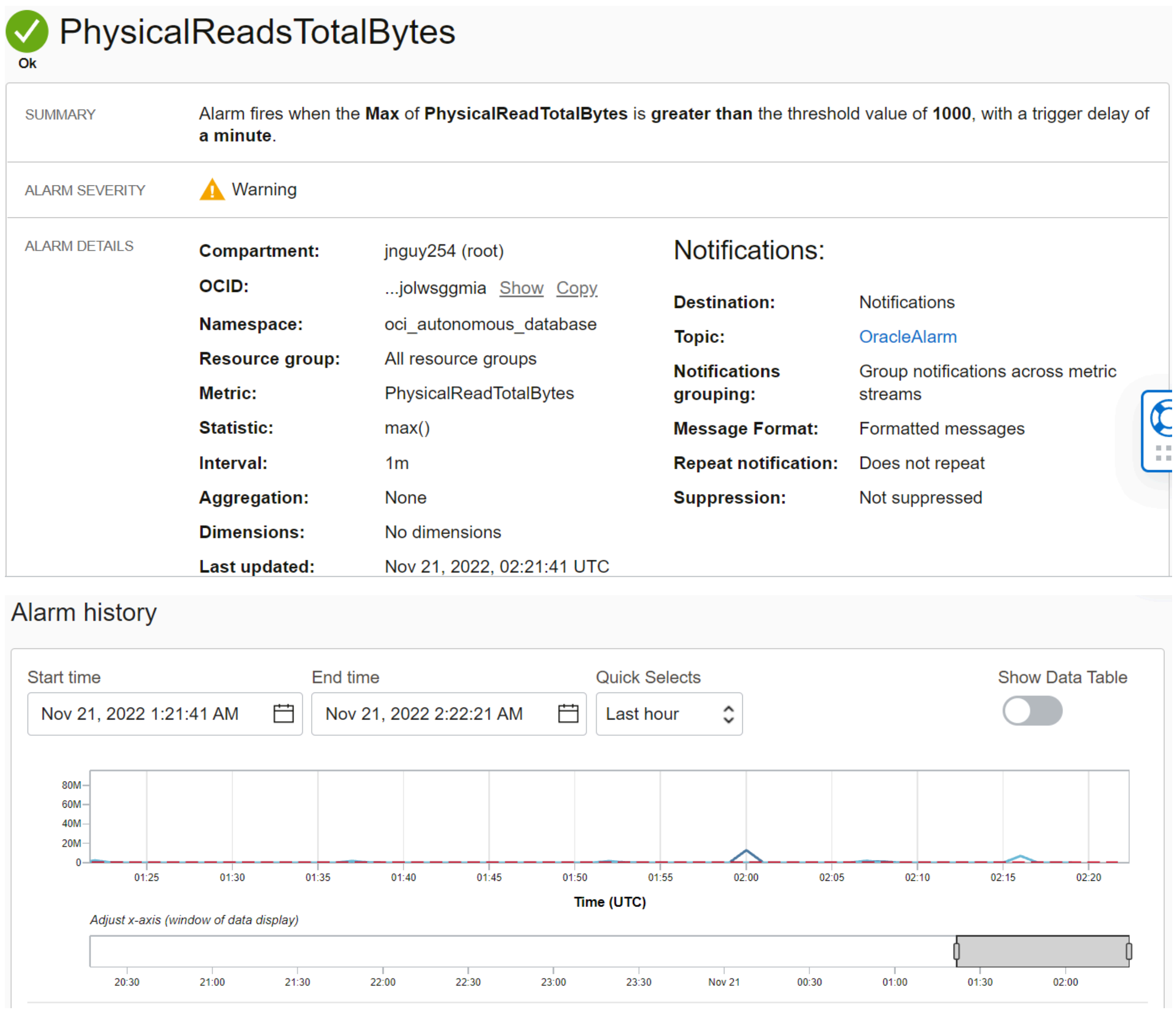
Oracle Autonomous Database provides a number of data protection tools:

Encryption. Oracle Autonomous Database applies automatic always-on encryption to the entire database, backups, and network connections. This ensures that data is protected whether it is at rest or in transit (i.e., connected to protocols such as SQL). Unique encryption keys are created and managed by Oracle Autonomous Database for each database and any backups or cloned databases. These keys are stored in a secure keystore on the same systems that support the database. Should a company wish to use their own keys, Oracle Autonomous Database will support key rotation to meet security needs.

Data access controls. Oracle allows specific controls for specific users, ensuring that data is only accessible to and edited by the appropriate designated individuals. There are three types of access controls: client access, database user and oracle cloud user. The parameters and controls for each user can be set up and edited as needed. This is crucial for data security as it prevents certain users (such as clients) from accessing confidential information.

Security assessments. Oracle provides an integrated platform known as Oracle Data Safe to assist in identifying and protecting sensitive data. Risk evaluation and security controls are supported by this platform to facilitate comprehensive assessment of database vulnerability while minimising direct user involvement.

Data activity monitoring. Users have the option to set alarms to receive direct email alerts when suspicious activities are detected in real-time. Flexibility is granted in terms of the parameters for the alarms. Activity metrics that the DBMS monitor can monitor include user activity (e.g., commits, rollbacks, failed logons), utilisation (CPU and storage), statements (e.g., number of queued statements), etc. Users can then specify the statistic to quantify, for example the rate, sum, or maximum value of a certain attribute. Other fields axvailable for customization include the threshold value and the delay before the alarm is triggered (Fig. 10a). Reports of the specified metric will be generated and available for users to monitor (Fig. 10b). This allows for evaluation of the alarm design and identification of potential modifications that may be necessary.



PhysicalReadsTotalBytes

Ok

SUMMARY	Alarm fires when the **Max** of **PhysicalReadTotalBytes** is **greater than** the threshold value of **1000**, with a trigger delay of **a minute**.
ALARM SEVERITY	⚠ Warning

ALARM DETAILS

Compartment:	jnguy254 (root)
OCID:	...jolwsggmia Show Copy
Namespace:	oci_autonomous_database
Resource group:	All resource groups
Metric:	PhysicalReadTotalBytes
Statistic:	max()
Interval:	1m
Aggregation:	None
Dimensions:	No dimensions
Last updated:	Nov 21, 2022, 02:21:41 UTC

Notifications:

Destination:	Notifications
Topic:	OracleAlarm
Notifications grouping:	Group notifications across metric streams
Message Format:	Formatted messages
Repeat notification:	Does not repeat
Suppression:	Not suppressed

Alarm history

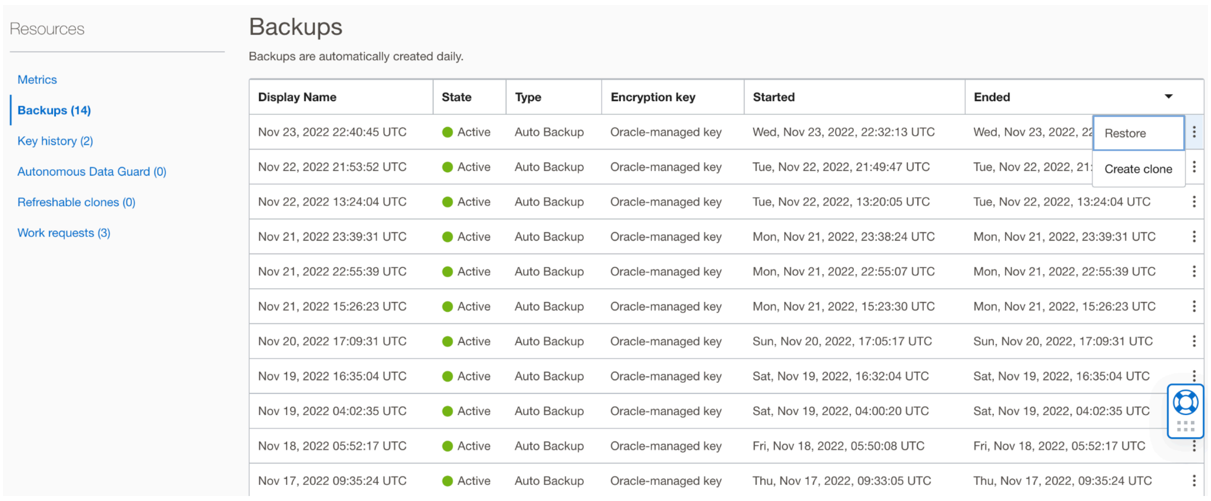
Start time	End time	Quick Selects	Show Data Table
Nov 21, 2022 1:21:41 AM	Nov 21, 2022 2:22:21 AM	Last hour	

Backups

In databases, backups are a critical component to maintaining the reliability of the data. A backup is the process of backing up the database software's operational state, architecture and data so that the database can be recreated in case of corruption or loss. This is critical to the protection of the database info.

In a traditional database, backups must be requested manually by a user. Oracle Autonomous Database removes the need for direct backup requests by performing automatic backups that are kept for 60 days. Daily incremental backups, weekly incremental backups, and bi-monthly full backups occur in addition to any manual backups the user can request. These backups can automatically be recovered and restored at any time once the user specifies the point in time or chooses the backup.

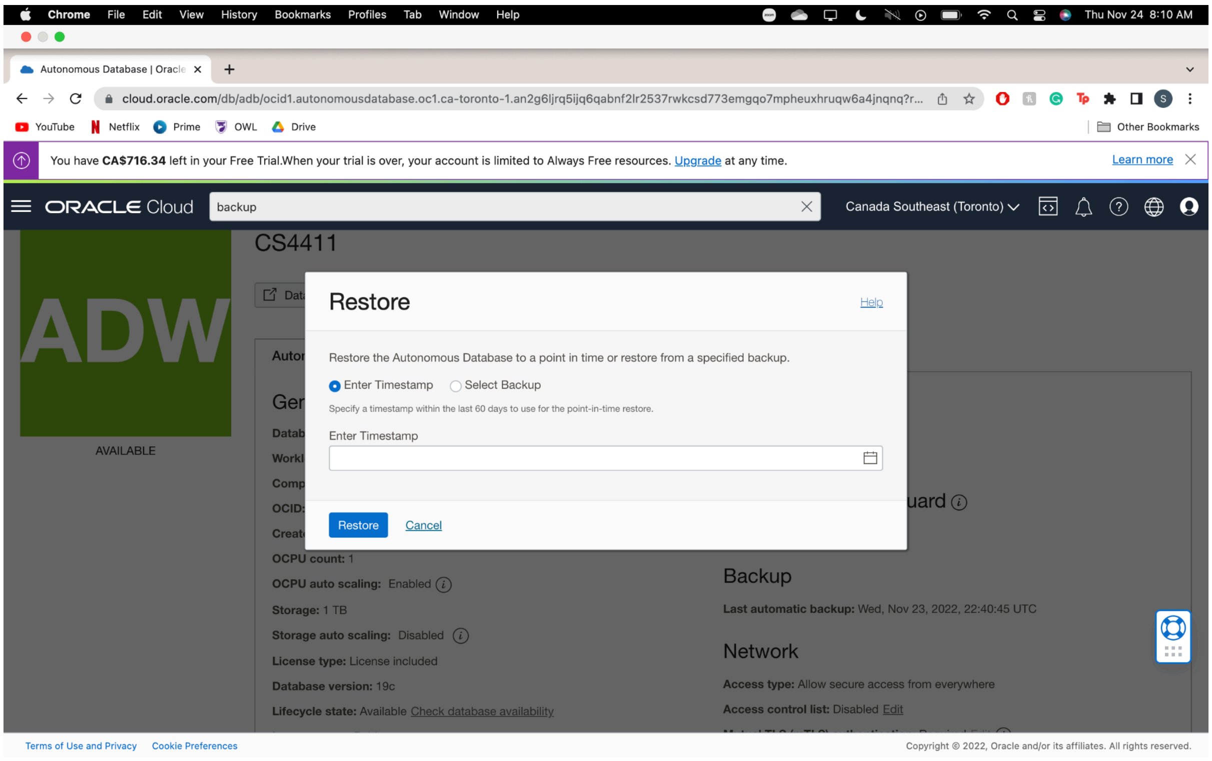
With the creation of a database in Oracle Autonomous Database Warehouse, the most recent backups, including all details, can be easily viewed under Autonomous Database details (Fig. 11). For each listed backup, users are provided the options to either clone the backup or restore the database back to this point. Alternatively, users can specify which specific backup or timestamp they would like to restore the database to (Fig. 12).



Backups

Backups are automatically created daily.

Display Name	State	Type	Encryption key	Started	Ended
Nov 23, 2022 22:40:45 UTC	● Active	Auto Backup	Oracle-managed key	Wed, Nov 23, 2022, 22:32:13 UTC	Wed, Nov 23, 2022, 22
Nov 22, 2022 21:53:52 UTC	● Active	Auto Backup	Oracle-managed key	Tue, Nov 22, 2022, 21:49:47 UTC	Tue, Nov 22, 2022, 21:
Nov 22, 2022 13:24:04 UTC	● Active	Auto Backup	Oracle-managed key	Tue, Nov 22, 2022, 13:20:05 UTC	Tue, Nov 22, 2022, 13:24:04 UTC
Nov 21, 2022 23:39:31 UTC	● Active	Auto Backup	Oracle-managed key	Mon, Nov 21, 2022, 23:38:24 UTC	Mon, Nov 21, 2022, 23:39:31 UTC
Nov 21, 2022 22:55:39 UTC	● Active	Auto Backup	Oracle-managed key	Mon, Nov 21, 2022, 22:55:07 UTC	Mon, Nov 21, 2022, 22:55:39 UTC
Nov 21, 2022 15:26:23 UTC	● Active	Auto Backup	Oracle-managed key	Mon, Nov 21, 2022, 15:23:30 UTC	Mon, Nov 21, 2022, 15:26:23 UTC
Nov 20, 2022 17:09:31 UTC	● Active	Auto Backup	Oracle-managed key	Sun, Nov 20, 2022, 17:05:17 UTC	Sun, Nov 20, 2022, 17:09:31 UTC
Nov 19, 2022 16:35:04 UTC	● Active	Auto Backup	Oracle-managed key	Sat, Nov 19, 2022, 16:32:04 UTC	Sat, Nov 19, 2022, 16:35:04 UTC
Nov 19, 2022 04:02:35 UTC	● Active	Auto Backup	Oracle-managed key	Sat, Nov 19, 2022, 04:00:20 UTC	Sat, Nov 19, 2022, 04:02:35 UTC
Nov 18, 2022 05:52:17 UTC	● Active	Auto Backup	Oracle-managed key	Fri, Nov 18, 2022, 05:50:08 UTC	Fri, Nov 18, 2022, 05:52:17 UTC
Nov 17, 2022 09:35:24 UTC	● Active	Auto Backup	Oracle-managed key	Thu, Nov 17, 2022, 09:33:05 UTC	Thu, Nov 17, 2022, 09:35:24 UTC


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu Nov 24 8:10 AM
Autonomous Database | Oracle
cloud.oracle.com/db/adb/ocid1.autonomousdatabase.oc1.ca-toronto-1.an2g6ljrq5ijq6qabnf2lr2537rwkcsd773emgqo7mpheuxhruqw6a4jnqnq?r...
YouTube Netflix Prime OWL Drive
Other Bookmarks
ORACLE Cloud
backup
Canada Southeast (Toronto)
CS4411
ADW
AVAILABLE

You have **CA$716.34** left in your Free Trial.When your trial is over, your account is limited to Always Free resources. Upgrade at any time. Learn more

Restore

Help

Restore the Autonomous Database to a point in time or restore from a specified backup.

[✓ Enter Timestamp] Select Backup

Specify a timestamp within the last 60 days to use for the point-in-time restore.

Enter Timestamp

Restore Cancel

OCPU count: 1
OCPU auto scaling: Enabled
Storage: 1 TB
Storage auto scaling: Disabled
License type: License included
Database version: 19c
Lifecycle state: Available Check database availability

Backup
Last automatic backup: Wed, Nov 23, 2022, 22:40:45 UTC

Network
Access type: Allow secure access from everywhere
Access control list: Disabled Edit

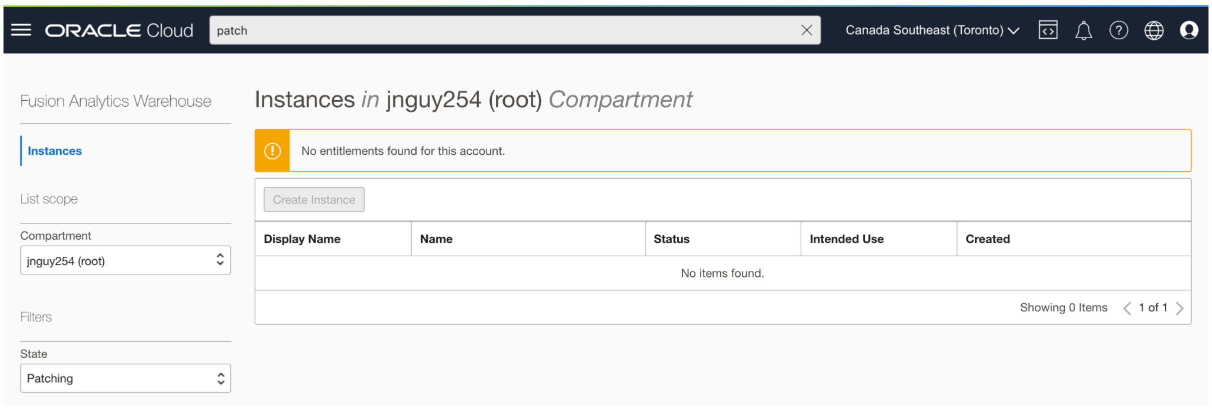
Terms of Use and Privacy Cookie Preferences
Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Patching

A patch is a set of changes to the database or its data with the goal of updating, fixing, or improving the system performance, which includes fixing security vulnerabilities & bugs. Patching is the process of implementing these changes so the database remains secure and updated.

In a traditional database, patches are to be installed manually by a user using an editing tool or a debugger. This is quite a time-consuming and tedious task in deciding which patches to apply, the effects on the database, and the execution of these installations. It also usually requires downtime in using the database or files while patching occurs.

In Oracle Autonomous Database, patches are done automatically with no downtime and the database can be used while patching occurs as it runs in the background. It can do so by using predefined maintenance windows to automatically patch the database. The patching history of the DBMS can be viewed through the patching filter (Fig. 13).



Instances *in* jnguy254 (root) *Compartment*

No entitlements found for this account.

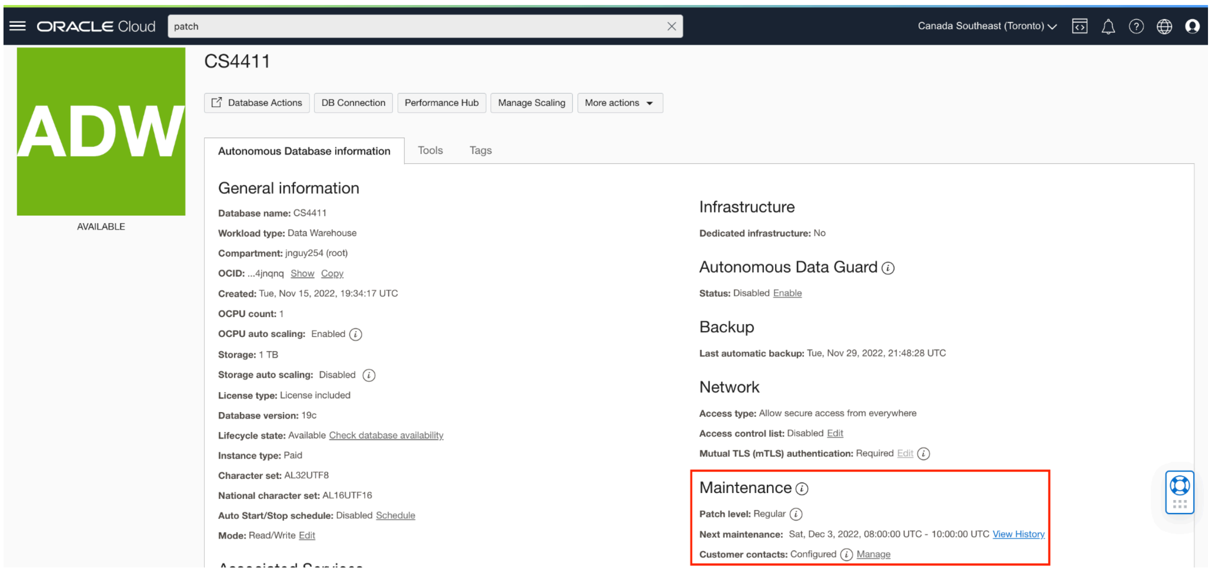
Compartment: jnguy254 (root)

State: Patching

Display Name	Name	Status	Intended Use	Created
No items found.				

Showing 0 Items 1 of 1

Database users are able to monitor upcoming scheduled maintenance, including patching, via the database details (Fig. 14). The patch level can be set to regular, in which it occurs at the same time as the maintenance or early, in which case it occurs one week before regularly scheduled patch.



CS4411

General information

Database name: CS4411
Workload type: Data Warehouse
Compartment: jnguy254 (root)
OCID: ...4jnqnq Show Copy
Created: Tue, Nov 15, 2022, 19:34:17 UTC
OCPU count: 1
OCPU auto scaling: Enabled
Storage: 1 TB
Storage auto scaling: Disabled
License type: License included
Database version: 19c
Lifecycle state: Available Check database availability
Instance type: Paid
Character set: AL32UTF8
National character set: AL16UTF16
Auto Start/Stop schedule: Disabled Schedule
Mode: Read/Write Edit

Infrastructure

Dedicated infrastructure: No

Autonomous Data Guard

Status: Disabled Enable

Backup

Last automatic backup: Tue, Nov 29, 2022, 21:48:28 UTC

Network

Access type: Allow secure access from everywhere
Access control list: Disabled Edit
Mutual TLS (mTLS) authentication: Required Edit

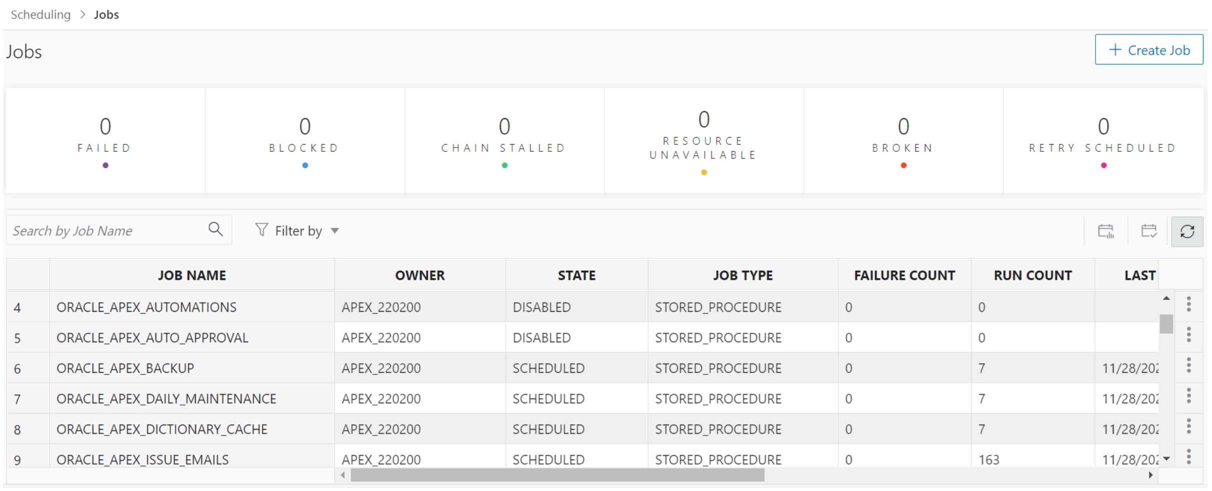
Maintenance

Patch level: Regular
Next maintenance: Sat, Dec 3, 2022, 08:00:00 UTC - 10:00:00 UTC View History
Customer contacts: Configured Manage

DATABASE PERFORMANCE EVALUATIONS

Job Scheduling

Functions such as backups, database maintenance, alert emails, etc are automatically set up as jobs by Oracle Autonomous Database. Statistics of these jobs are available through a dashboard on the platform for surveillance (Fig. 15). Users are able to set their own jobs for a variety of features, including some of the aforementioned functions explored in this project. The autonomous database will evaluate scheduled jobs and implement an effective schedule for accomplishing the required actions.

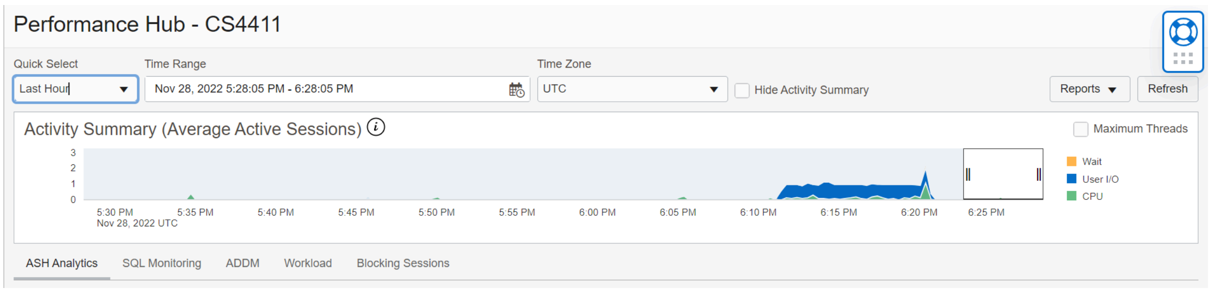


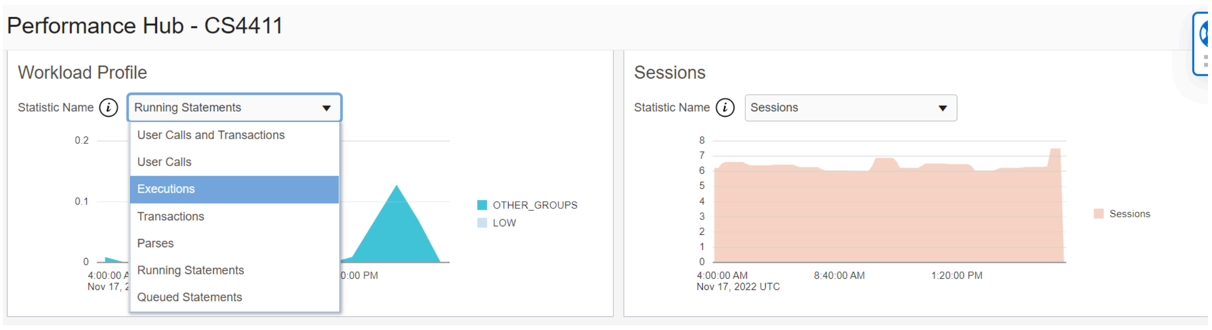
0 FAILED	0 BLOCKED	0 CHAIN STALLED	0 RESOURCE UNAVAILABLE	0 BROKEN	0 RETRY SCHEDULED

	JOB NAME	OWNER	STATE	JOB TYPE	FAILURE COUNT	RUN COUNT	LAST
4	ORACLE_APEX_AUTOMATIONS	APEX_220200	DISABLED	STORED_PROCEDURE	0	0	
5	ORACLE_APEX_AUTO_APPROVAL	APEX_220200	DISABLED	STORED_PROCEDURE	0	0	
6	ORACLE_APEX_BACKUP	APEX_220200	SCHEDULED	STORED_PROCEDURE	0	7	11/28/202
7	ORACLE_APEX_DAILY_MAINTENANCE	APEX_220200	SCHEDULED	STORED_PROCEDURE	0	7	11/28/202
8	ORACLE_APEX_DICTIONARY_CACHE	APEX_220200	SCHEDULED	STORED_PROCEDURE	0	7	11/28/202
9	ORACLE_APEX_ISSUE_EMAILS	APEX_220200	SCHEDULED	STORED_PROCEDURE	0	163	11/28/202

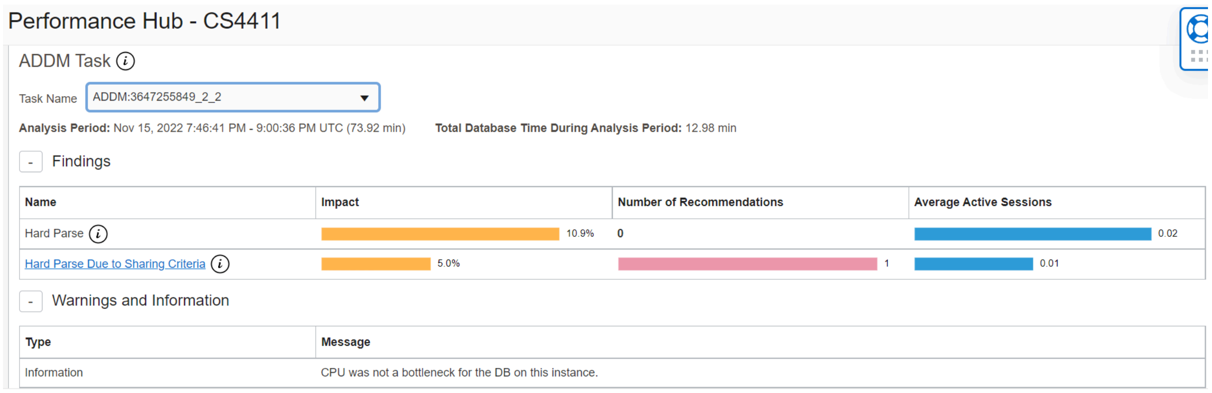
Performance Hub

An overview of the autonomous database's performance is also available through the Performance Hub to get a quick glance or in-depth report of various evaluations of the database's performance (Fig. 16a). Users are able to get summaries of measures such as user calls, transactions, and statements to better understand the workload from any given week (Fig. 16b). The dashboard also monitors the active sessions from the week.





The Performance Hub also displays ADDM tasks, which are reports from the autonomous database of the analyzed data captured by the Automatic Workload Repository. These examinations are used to determine possible performance problems in the Oracle Autonomous Database and aid in further modifications to optimize database performance (Fig. 17).



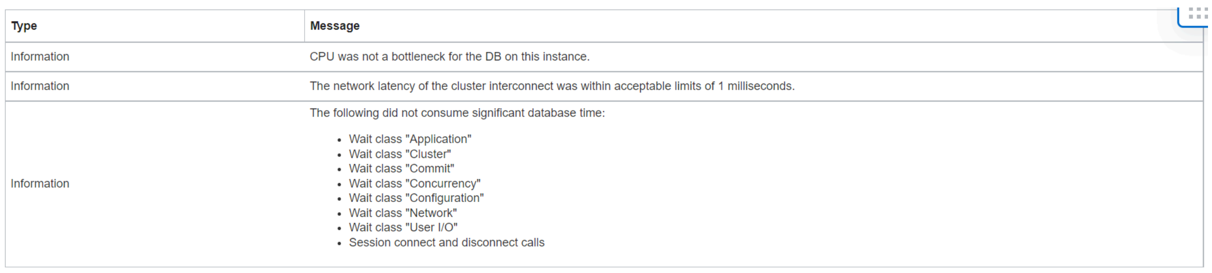
Analysis Period: Nov 15, 2022 7:46:41 PM - 9:00:36 PM UTC (73.92 min) **Total Database Time During Analysis Period:** 12.98 min

Findings

Name	Impact	Number of Recommendations	Average Active Sessions
Hard Parse	10.9%	0	0.02
Hard Parse Due to Sharing Criteria	5.0%	1	0.01

Warnings and Information

Type	Message
Information	CPU was not a bottleneck for the DB on this instance.



Type	Message
Information	CPU was not a bottleneck for the DB on this instance.
Information	The network latency of the cluster interconnect was within acceptable limits of 1 milliseconds.
Information	The following did not consume significant database time: • Wait class "Application" • Wait class "Cluster" • Wait class "Commit" • Wait class "Concurrency" • Wait class "Configuration" • Wait class "Network" • Wait class "User I/O" • Session connect and disconnect calls

CONCLUSIONS

Through this system study, the importance and effectiveness of autonomous databases were made extremely evident. Even without having looked into all the functionalities and features of Oracle Autonomous Database Warehouse, it was extremely clear why autonomous databases are gaining so much popularity in today's society. From the involvement of machine learning to customizable graphing, protection & data management, the advantages of an autonomous database in comparison to a traditional one are undeniable.

An extension of this study would be to do a more in-depth analysis testing the quality and implementation of all of Oracle Autonomous Database Warehouse's features. Given the limited time and scope of this system study, it was not possible to test all functionalities but this would provide more insight into autonomous warehouses. The use of a larger dataset to create a database would also be a future step that can be taken to allow the analysis of how this autonomous database works with larger data which is a more accurate reflection of how it will be used in everyday life.

Oracle's ease of access, limited coding knowledge required, and vast functionalities is what makes it one of the leading software in the world and from this analysis, it is clear that it is optimal to use.

REFERENCES

https://docs.oracle.com/en-us/iaas/autonomous-database-shared/doc/autonomous-auto-index.html?Highlight=index

https://docs.oracle.com/en/database/oracle/oracle-database/19/admin/managing-indexes.html#GUID-D1285CD5-95C0-4E74-8F26-A02018EA7999

https://docs.oracle.com/en/database/oracle/oracle-database/19/admin/managing-indexes.html#GUID-082972AD-1866-411A-8250-9B23D4088582

https://www.oracle.com/a/ocom/docs/database/idc-oracles-autonomous-database.pdf

https://www.oracle.com/ca-en/autonomous-database/what-is-autonomous-database/

https://www.oracle.com/a/ocom/docs/database/atp-accelerate-innovation-ipaper.pdf

https://www.oracle.com/ca-en/database/graph/

https://www.ibm.com/cloud/learn/database-security

https://docs.oracle.com/en/cloud/paas/autonomous-database/adbsa/gs-security-and-authentation-autonomous-database.html#GUID-9B98BC3F-AD86-4098-925C-6F141D719BAE

https://docs.oracle.com/en/cloud/paas/autonomous-database/adbsa/data-insights.html