ORACLE

Develop Modern Applications with Oracle Database

How Oracle Database can help you manage data in the software development life cycle and build scalable, secure applications fast.

September 09, 2020 | Version 1.00

PURPOSE STATEMENT

This document provides an overview of Oracle Database features that help developers build applications. It is intended solely to help you assess the business benefits of using Oracle Database and to plan your development projects.

INTENDED AUDIENCE

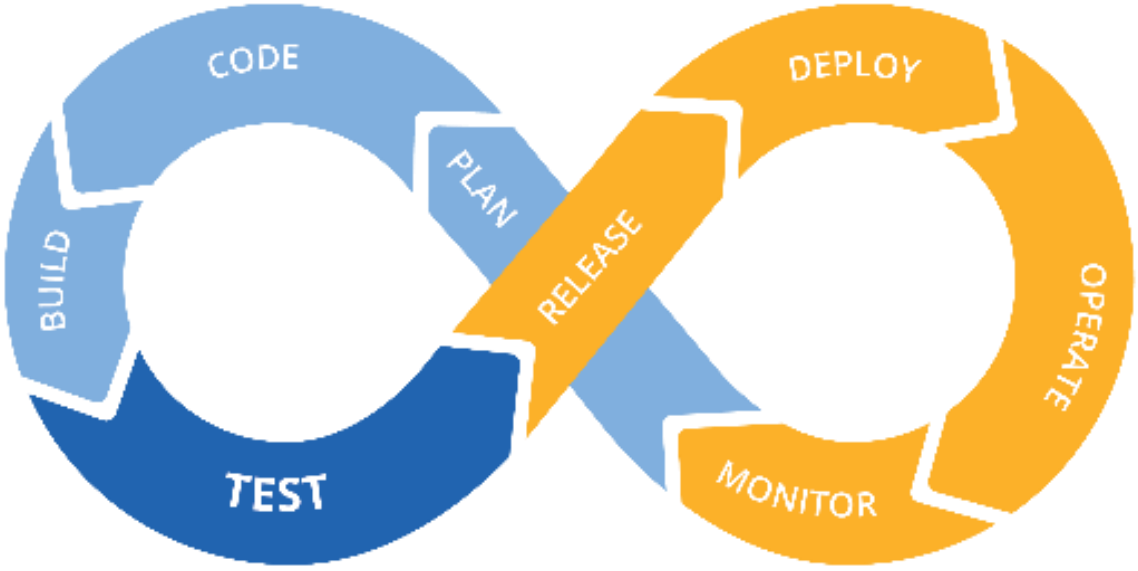
This technical brief is for developers building data-driven applications. It assumes familiarity with basic database terms and the software development life cycle.

DISCLAIMER

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Data-driven applications are at the heart of successful businesses. It's therefore critical that you manage your data effectively throughout the development life cycle. Using the full capabilities of your database makes it easy to get the most out of your data and make your application secure and efficient.



There are many stages in the modern software development process. Databases play a key role at each of these points.

Oracle Database and its supporting tools make it simple to manage data during the software life cycle. We focus on the key features to help application developers create, deploy, and maintain data-driven applications. You'll learn how to:

1. Plan your database schema and use the multimodel capabilities of Oracle Databases to keep your architecture simple and change data formats as needed.
2. Develop applications fast using Oracle Application Express, connect to the database with popular programming languages such as Java, Python, or JavaScript or create database REST APIs to access your data.
3. Build new development and test databases quickly using Oracle Multitenant.
4. Simplify the process of testing data changes using Oracle SQL Developer to create database unit tests and Flashback technologies to revert data to its pre-testing state.
5. Release schema changes safely by automating these deployments with zero application downtime.
6. Provide ongoing support and monitoring of the application. Both in terms of measuring business metrics and monitoring application performance and availability.

As your business grows, Oracle Database has a range of technologies to scale your database and keep SQL fast. You can keep your data secure throughout the process with the many security features of PL/SQL. Further protecting your data is easy with row-level security to control who can see what and powerful controls to manage user access and enforce separation of duties.

When you're ready to get started with Oracle Database, we have a range of free resources to get you up-and-running fast.

OVERVIEW

Writing code is fun, but some phases of the development life cycle can be painful to manage.

Businesses want applications up-and-running as fast as possible, but often give you vague and changing requirements!

This makes defining a data model hard as you're working with incomplete information. As requirements shift, you may need to change how you store data.

You can address this by releasing small changes early and often. This puts features in front of your customers and gives you rapid feedback. To do this, you need smooth processes to push changes from development to production. This requires robust testing procedures and automated deployment processes.

Traditionally, provisioning new test and development databases, validating data changes, and resetting after tests was time consuming and tricky.

You also have the challenge of getting your application into production. Making automated and repeatable deployments can be time consuming. Complex schema changes can take hours to complete, but need to be deployed without service interruption (zero downtime).

After deployment, you must find and fix slow running queries fast. So you need to closely track application performance. As data volumes grow, you must ensure queries perform in compliance with your service-level agreement.

Your company also wants analyze promotions and spot changes in buying trends. So you need to write queries tracking business metrics too.

Through the entire life cycle you need to secure your data. This ensures compliance with data protection laws and avoids reputational damage due to data breaches.

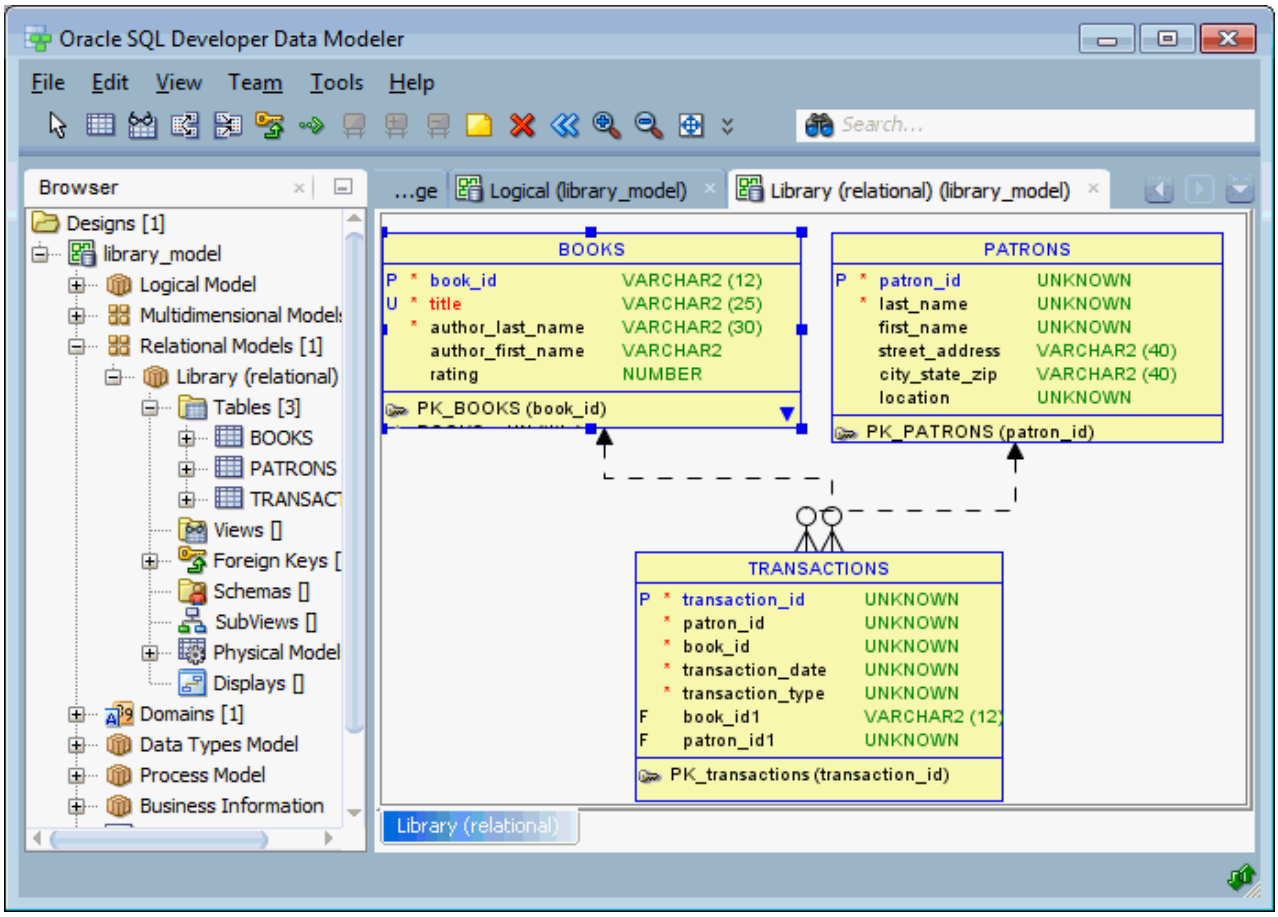
As your business evolves, more applications may need access to your data. You may want to create REST Application Programming Interfaces (APIs) exposing core tables, common queries, and key transactions. Needs may arise to connect to the database using a range of different programming languages.

Whatever your challenges are when building data-driven applications, Oracle Database has a wealth of features to help ease the pains and make application development a delight. It all starts with planning your data model.

PLAN YOUR APPLICATION DATA MODEL

The first step in building any application is planning how to organize your data. A key part of this is identifying what you need to store and creating a data model for these items.

ORACLE SQL DEVELOPER DATA MODELER



Using SQL Developer Data Modeler you can view and edit the tables in a schema along with their relationships and properties.

If you want to simplify data modeling tasks, use Oracle SQL Developer Data Modeler. Using this free graphical tool you can create, browse, and edit models. You can do this with logical, relational, physical, multi-dimensional, and data type models.

The Data Modeler provides forward and reverse engineering capabilities and supports collaborative development through integrated source code control.

You can also use this to generate Data Definition Language (DDL) scripts for your tables. If you have changes to apply to an existing schema, you can compare the design to the schema to create the change scripts.

Those DDL scripts should go beyond simply creating tables with columns. Take full advantage of all the declarative features available to you to define referential integrity and check constraints. The more application intelligence you build *into* the data model, the less code you will have to write later.

For a lightweight way to create DDL scripts, use Quick SQL. This Oracle APEX utility enables you to prototype data models fast using a markdown-like syntax. Quick SQL expands this to Oracle SQL syntax to create tables, indexes, and triggers. It includes options to generate random test data.

STORE DATA IN THE BEST FORMAT FOR THAT BEST FITS YOUR REQUIREMENTS

Early in the development life cycle, requirements are often vague, incomplete, and prone to changing daily. You may also have cases where the columns are not known in advance, or where there is a very high number of varied sparse data.

This makes defining relational tables tricky. Constantly running DDL statements to add, remove or change columns would not be practical. You can extend a relational data store by storing data in a flexible format, such as JSON, to store new attributes without changing the underlying tables.

Or perhaps a shift in requirements means it's more appropriate to model your data as a graph. Or you may need to support XML files.

Using Oracle Database, you can choose the best storage format for each module. This allows you to combine the strength of relational storage with the flexibility of JSON or XML. As requirements evolve, you can change to specialized models such as graph. All the tools you need are readily available in the same database.

Applications may store their data in many different ways. Using Oracle Database you can choose the best model for your requirements.

Using the same database technology to store all types of data keeps the system simple. You can change how you store specific entities and attributes without switching databases. Combining data in different formats is easy – all you need to do is join!

FLEXIBLE AND EFFICIENT CODING

Once you've planned out your data model, it's time to start coding.

DEVELOPMENT ENVIRONMENTS AND TOOLS

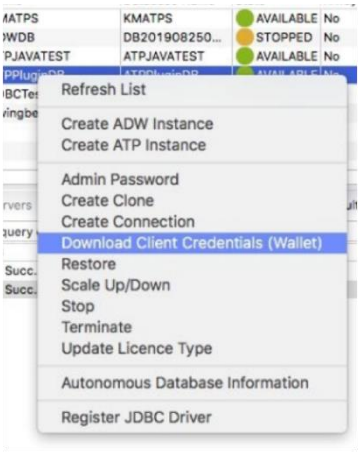
Oracle provides many tools for you to write your database code, including Oracle SQL Developer, SQLcl, and SQL*Plus.

Oracle SQL Developer is a free, integrated development environment. It offers complete end-to-end development of your PL/SQL applications. It also has a worksheet for running queries and scripts, and a reports interface. To make source control easy, it includes Git and Subversion integration.

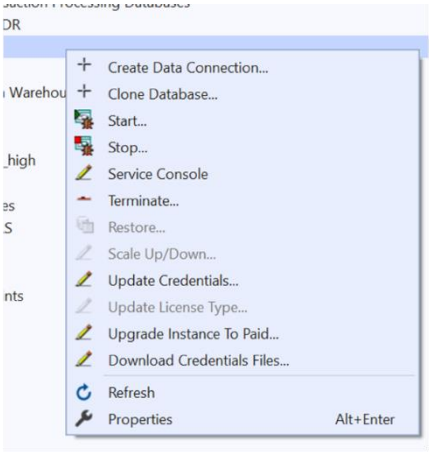
You can use both Oracle SQL Developer and the Data Modeler in traditional and Cloud environments.

If you want command-line access, you can use the free command-line tools SQL*Plus or SQLcl. With these you can run interactively or batch execute SQL and PL/SQL. SQLcl also provides in-line editing, statement completion, and command recall for a feature-rich experience. All while also supporting your previously written SQL*Plus scripts.

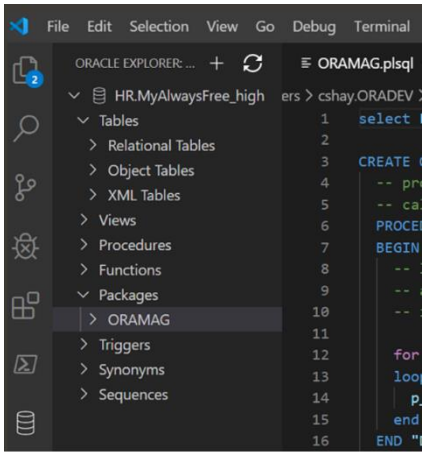
Oracle also provides support for third-party environments. These include Visual Studio Code via the Oracle Developer Tools and the Oracle Enterprise Pack for Eclipse.



Eclipse – Java Developers



Visual Studio - .NET Developers

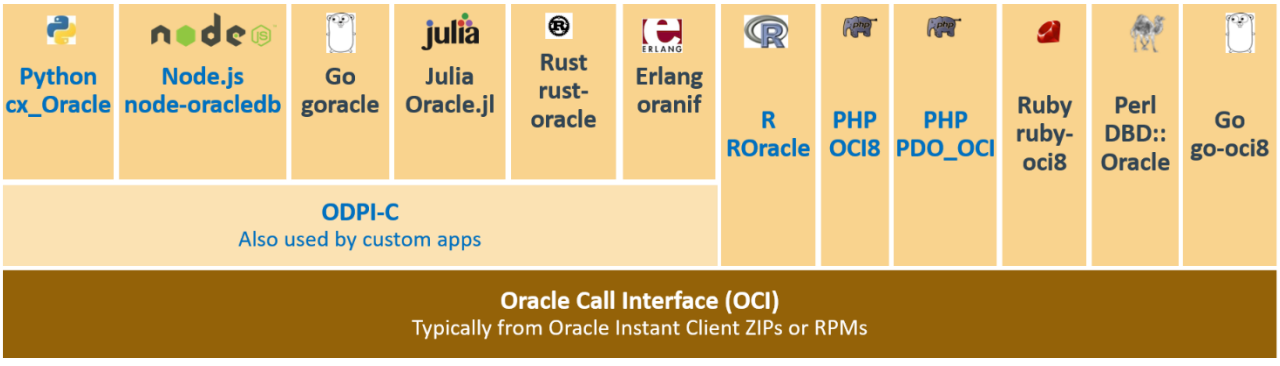


VS Code - Code Editor

You can use Eclipse, Visual Studio or VS code to build applications using Oracle Database

PROGRAM WITH ANY LANGUAGE

Oracle Database has a wide range of drivers to connect to the database. These include Java, .NET, Python, Node.js, PHP, C/C++, and Ruby on Rails. These drivers are optimized to provide high performance and advanced security features, allowing developers to focus more of their time on meeting end-user requirements.



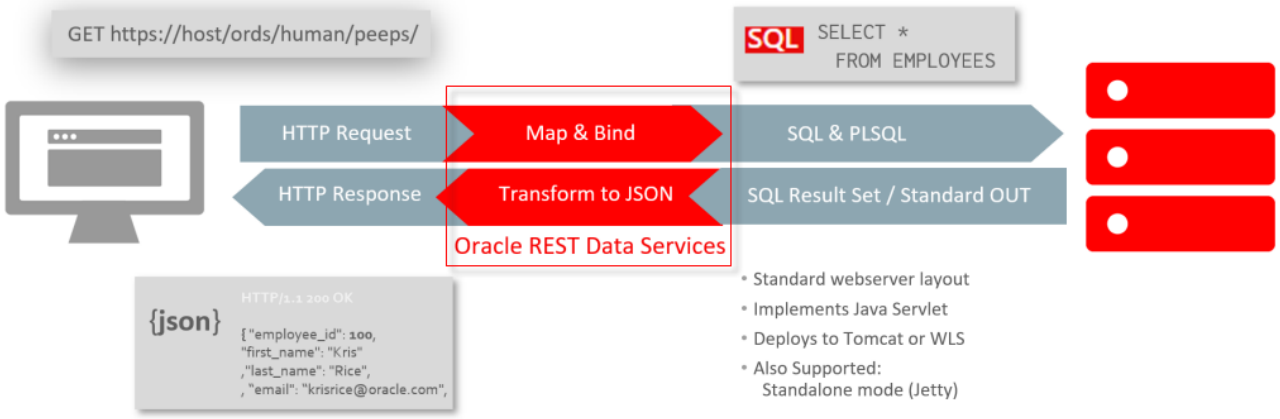
There are many drivers available for you to connect to Oracle Database from your chosen programming language

Can't find your favorite language in the list?

The Oracle Database Programming Interface for C is an open-source library you can use to create language specific drivers. And with the ODBC driver, you can connect ODBC applications to the database. Or you can create a language independent access by defining REST APIs to expose common business transactions, queries or tables.

CREATE DATABASE REST APIS

REST has become the de facto standard for data services. Building a REST API layer on top of your database makes it easy for new applications to access your data.



Manage ORDS from your terminal, with its PL/SQL API, or our GUI in SQL Developer.

Using Oracle REST Data Services (ORDS), you can expose PL/SQL procedures, SQL queries or individual tables as REST endpoints. This enables you to build database APIs exposing business functionality.

With AutoREST you can enable or disable ORDS for specified tables or views in a schema. This makes it easy for you to quickly expose data. You can also auto enable PL/SQL packages, procedures and functions. If you want to allow clients to send their own SQL statements you can make REST-Enabled SQL services.

Whichever method you use to expose your data, SQL Developer includes modules to help you develop and manage REST services.

You can also use ORDS to expose backend functions, such as managing pluggable databases, exporting data, and reviewing database performance.

RAPID APPLICATION PROTOTYPING AND DEVELOPMENT WITH ORACLE APPLICATION EXPRESS (APEX)

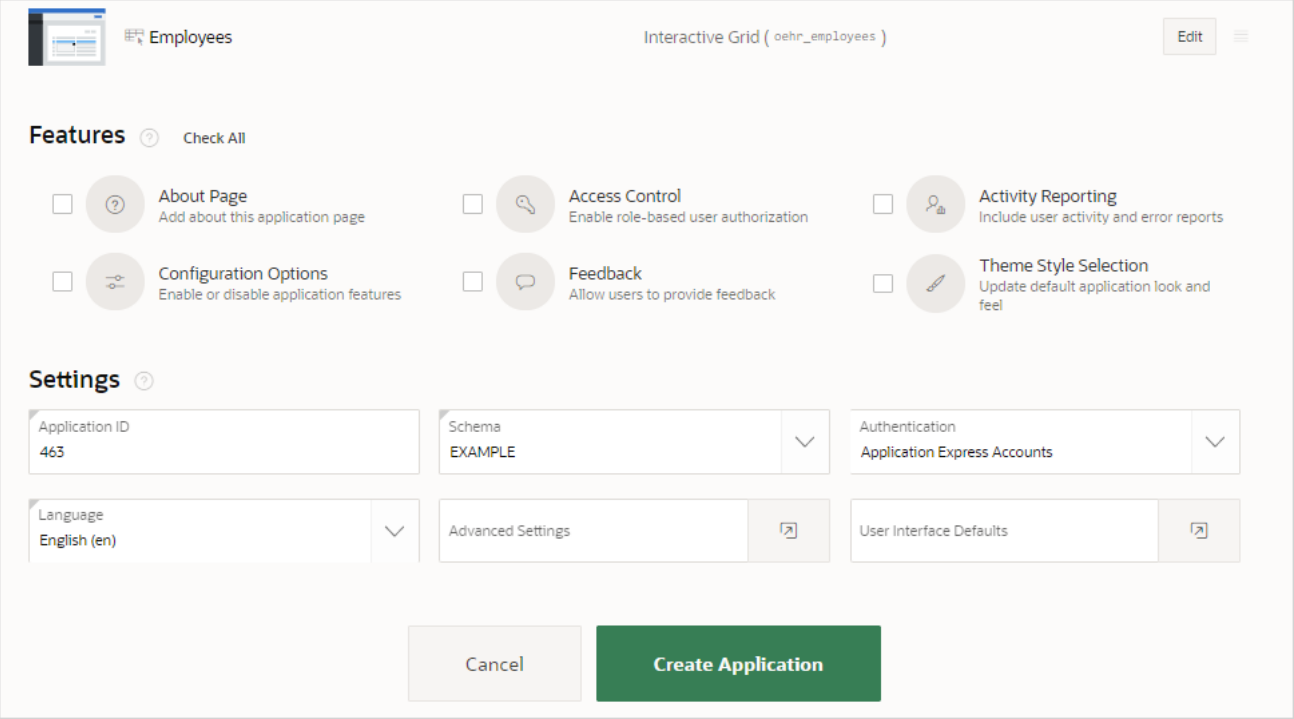
When developing applications, you're under pressure to complete changes fast. With many technologies, you can spend huge amounts of time getting frameworks working before you start coding the business requirements. Time your company wants you to spend implementing their features.

Using Oracle APEX, you can avoid all this effort and get up-and-running fast.

APEX is a low-code development platform. It enables you to build data-driven, scalable, secure applications fast.

Using APEX, you can quickly develop and deploy compelling apps that solve real problems and provide immediate value. You won't need to be an expert in a vast array of technologies to deliver sophisticated solutions. You can focus on solving the problem and let APEX take care of the rest.

Oracle APEX strives to make it as easy as possible for you to display, manipulate, chart, and process data as easily and efficiently as possible. No matter where your data comes from, whether it is from a local database, remote database, or a web service, Oracle APEX features state-of-the-art functionality to help you turn data into information.



With the APEX create application wizard, you can build a new app in minutes from many data sources, including spreadsheets and DDL scripts.

With the Interactive Report component, you can empower your users to easily customize the data they see in a way that uniquely satisfies their needs. From simple changes like determining which columns to show, to more sophisticated customizations such as pivoting data, it is all within reach.

Using the Interactive Grid component, it's easy to rapidly edit multiple rows of data — as simple as clicking on a cell and editing its value. If your users are familiar with spreadsheets, they'll feel right at home, with features like row and column selection, copy down, fill, and much more. Unlock the full potential of your data in an easy to use and highly customizable way with Interactive Grids.

Enjoy powerful chart and visualization capabilities in Application Express with, powered by Oracle JET. You can add highly customizable, accessible, and extremely versatile charts to your applications with ease.

Using APEX, you can build beautiful applications fast.

Whatever your chosen programming language, you can connect and create applications using Oracle Database.

After defining your schema and coding database changes, you need to start the process of pushing these to production. So begins the continuous integration (CI) and continuous delivery (CD) pipeline. Applying the changes to staging environments, testing them, and deploying to production.

Standardizing the data platform means you only have one technology to get up-and-running. This makes it easier to provision new development and test environments. But populating non-production databases with test data can still be a challenge.

RAPID APPLICATION TESTING

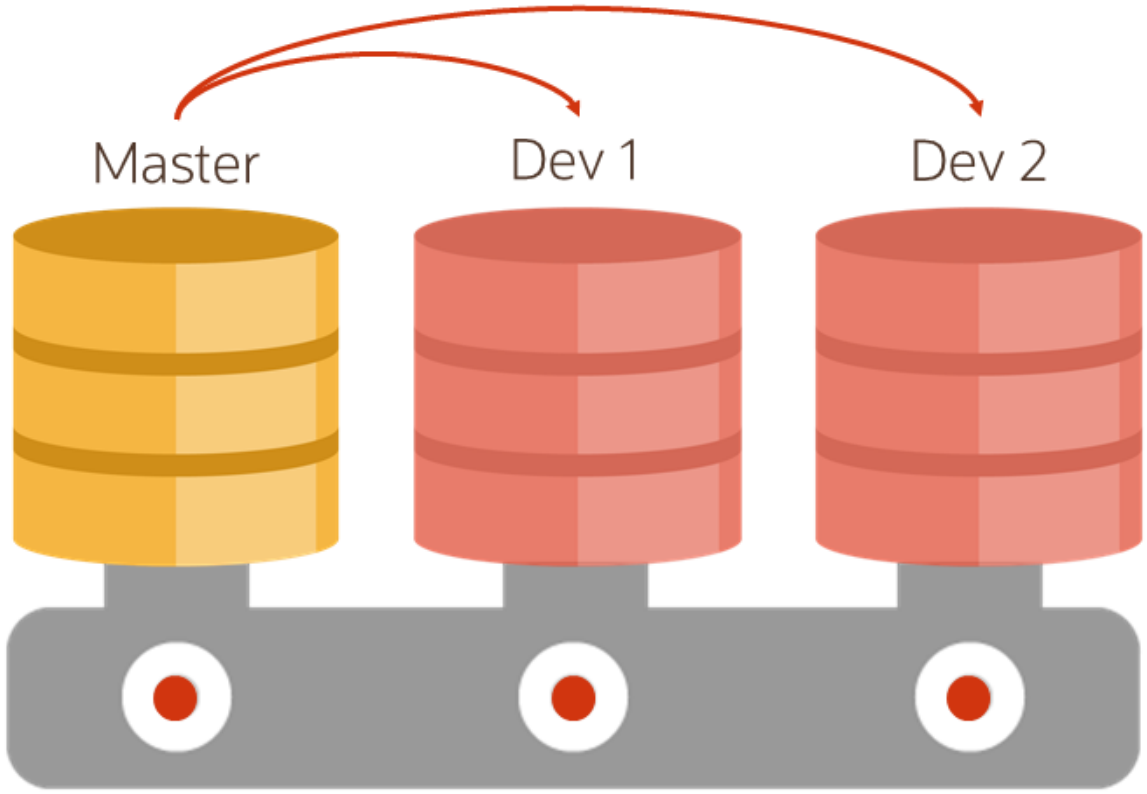
CREATE AND REFRESH DEVELOPMENT AND TEST DATABASES WITH EASE

Provisioning a development or test environment is simplified in a Multitenant environment.

Using Oracle Multitenant, you can create a Pluggable Database (PDB) for each developer and test environment. You can initialize these to the latest version of the schema by cloning an existing PDB.

Refreshing data by cloning the production database is appealing, but often impractical. Either because the production database is too large to refresh on a regular basis, or it contains sensitive data, which developers should not see.

You can avoid this risk by creating a gold master PDB containing the latest schema and test data. When you apply changes to production, you also update this master PDB.



You can use PDB cloning to create new development environments. You define one PDB to be the gold master, holding the latest schema and data needed run the application. You can then copy this to provision new databases fast.

You can clone this master database, giving you a workable schema in minutes. Or for more experimental development, you can copy the schema without data.

Once you've got the environment ready and applied your changes, you need to test them.

SIMPLE DATA TESTING

Testing database changes is challenging. Even simple transactions may need specific reference data in place to work. To repeat the tests, you often need to back out the effects of previous runs.

Details | Results

hr_orcl

Test Run	Status	Duration	Message
AWARD_BONUS			
Run On - 2009-09-08 19:15:23.012	SUCCESS	266	
Implementation - Test Implementation 1	SUCCESS	109	
Startup	SUCCESS	31	
Operation Call	SUCCESS	0	
IN Parameter #1 - EMP_ID			Value: [177]
IN Parameter #2 - SALES_AMT			Value: [5000]
Validation #1 - Private (Query returning row(s))	SUCCESS	0	
Teardown	SUCCESS	78	
Implementation - empty_comm_pct	SUCCESS	157	Expected exception: [6510], Received: [6510: ORA-06510: PL/SQL: unhandled
Startup	SUCCESS	63	
Operation Call	SUCCESS	0	Expected exception: [6510], Received: [6510: ORA-06510: PL/SQL: unhandled
IN Parameter #1 - EMP_ID			Value: [101]
IN Parameter #2 - SALES_AMT			Value: [5000]
Teardown	SUCCESS	94	

After completing a run using SQL Developer Unit Testing, you get a report showing you the outcome of each test

To help you manage this, Oracle SQL Developer includes an extensive unit testing suite. This allows you to define setup routines to initialize data and clean-up processes to revert them. This helps you check each component works as expected. When you run integration and regression tests, you may wish to back out the changes for specific transactions, or restore tables to a specific point part-way through the testing suite. Or you may wish to revert the entire database back to the start of the test run.

Oracle's Flashback technologies make all these tasks simple. With Flashback Query you can view data in a table as it existed a few minutes ago. This makes it easy to compare old and new versions of the data, so you can ensure the tests changed the right rows.

You can also use Flashback Query see all the changes made to rows in a period of time. This helps you verify that your code changes the correct rows. After completing a test, often you want to reset data back to its pre-testing state. You can do this table-by-table with Flashback Table.

Or you can create guaranteed restore points (GRPs) for a database at appropriate points in a test run. Using Flashback Database you can revert the whole database back to these points. This is faster and easier than rebuilding the whole database to get it back to its pre-test state.

Once your testing is complete, it's time for the riskiest part of the process: pushing your changes to production.

RELIABLE DATABASE PRODUCTION DEPLOYMENTS

Many things can go wrong when you change the production database. Bugs can slip through testing and queries may get slower, making customers frustrated. The release itself may fail, leaving the application in an unworkable state.

Releasing small changes often minimizes these risks. The fewer changes you make, the less likely it is something will go wrong. And if any cause application errors or slow SQL the root cause is easier to find and fix.

But to release database changes regularly, you need two things in place:

- Automated database deployments
- Zero application downtime when making database changes

AUTOMATE SCHEMA DEPLOYMENTS

Manual deployments are error-prone and time-consuming. These problems can be a barrier to frequent database releases. Automating the push to production minimizes these issues. This gives you more time to spend on improving functionality.

To help you do this, Oracle SQLcl has direct integration with Liquibase. This is an open-source database-independent library for tracking, managing and applying database schema changes.

Using SQLcl, you can capture database changes in Liquibase changesets and changelogs. You can apply these changes to test and production databases using the SQLcl or Liquibase interfaces. This gives you the flexibility to manage releases how you want.

Once you've packaged up your changes, the next challenge is deploying them to production with little – or no – application downtime.

ZERO-DOWNTIME DATABASE CHANGES

When dealing with petabytes of data, many schema changes can take hours to complete. So you can do these while customers use your application, many DDL statements in Oracle Database are non-blocking. This allows application DML to run while the release DDL executes.

But some schema changes have several steps, such as adding a column and updating its value. These could be almost impossible to complete while the application is live.

Using Edition-based Redefinition (EBR), you can upgrade your application online. This feature introduces the concept of an edition and editionable objects (such as editioning views and cross edition triggers). Sessions connect to a specific edition and can only view editioned objects changes in that edition.

This allows you to create a private edition you release changes to, while customers continue to use the previous version of the schema. Once you've completed the release, you can allow customers to use the latest schema.

You can then seamlessly migrate customer sessions to the latest version of your application. If anything goes wrong in this process, rollback is easy. Since your online application upgrade is designed to run two editions concurrently, you simply point customers back to the previous edition.

When data changes are involved, cross edition triggers may be left enabled or not based on your assessment on the impact. Since the users are not impacted by downtime, changes can be made during peak hours and time can be taken to diagnose any unforeseen issues during a deployment. This reduces the stress caused by mission critical systems outages and its downtime associated cost.

Combined with automated and tested release scripts, you can deploy schema changes with little human effort.

But problems can arise hours after a release. When these happen, you need to know fast. So you need comprehensive monitoring in place. This must identify slow SQL and track your business' Key Performance Indicators (KPIs).

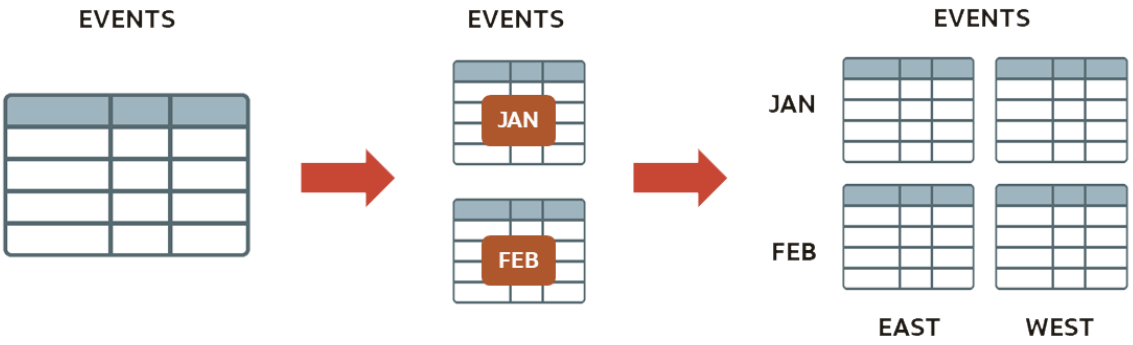
MAINTAIN OPERATIONAL EXCELLENCE

KEEP THE APPLICATION FAST

In the early days of an application, tuning queries is a matter of having the right indexes in place.

But as data volumes grow to terabytes, petabytes, and beyond, proper indexing alone may not be enough to give acceptable performance.

You can mitigate this using Oracle Partitioning. This effectively carves up one table into many smaller tables, while still accessing it as one. Writing your queries to access one or two partitions limits the maximum amount of data a query will read. With the ability to enable or disable indexes on each partition, this helps you tune your SQL queries.



A large table can be hard to manage. Partitioning splits the data into smaller, more manageable pieces.

Partitioning also makes the data easier to manage. With date partitioned table, it's easy for you to copy and remove the oldest partitions from the table. This makes partitioning a key component in information life cycle management processes.

But reporting queries often span many partitions. Often these read millions of rows, but return a handful.

You can make these aggregation queries faster by storing their results in a summary table. This can give huge performance gains, but is a lot of work. You need to refresh the summary when the underlying data change and point queries to the new tables.

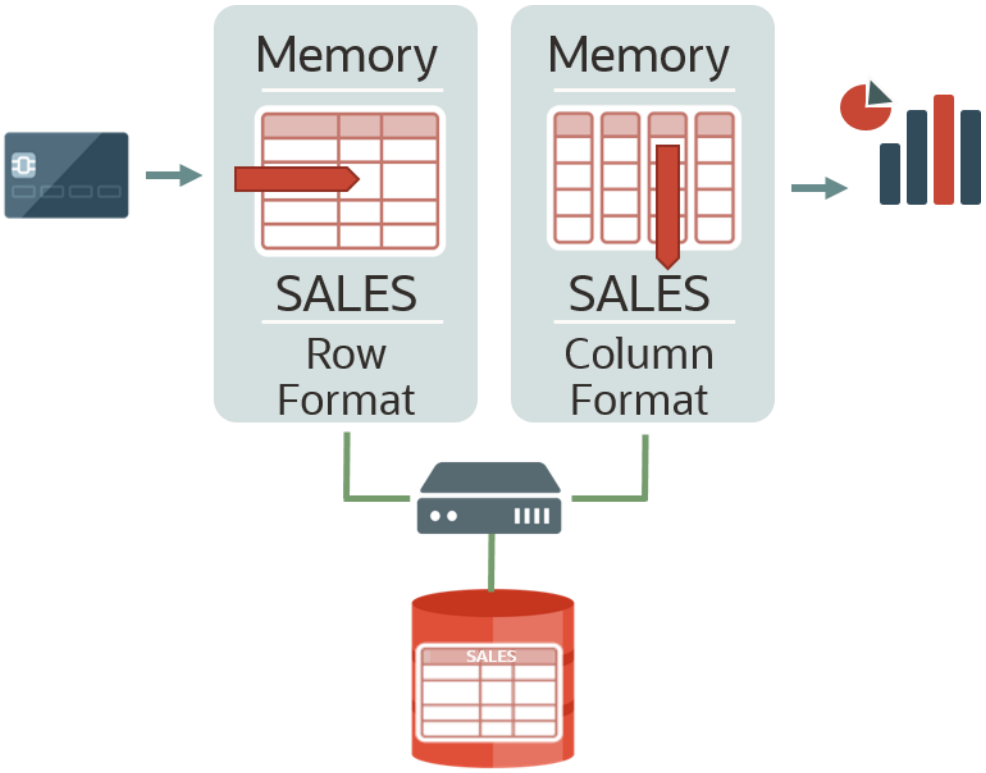
Using Oracle Database, you can avoid this effort using materialized views or Database In-Memory.

AGGREGATE MILLIONS OF ROWS FAST

Using materialized views (MVs), you can pre-compute the results of queries. Oracle Database can keep these in sync with the base tables when you commit changes and direct existing queries to use MVs with no code changes.

This makes them perfect for giving business summaries, such as sales totals by day, month or year. For cases where MVs are unsuitable, you can make reporting queries much faster using Database In-Memory.

This stores data in a columnar format, optimized for analytic queries. The database writes data to the row and column stores at the same time. Because both formats store the same data, the optimizer can choose which to use when you run your queries.



When you commit, the database writes the change to the table and In-Memory column store at the same time. The optimizer chooses whether to access the data in row or column format at run time.

So your existing SQL can use In-Memory data with zero code changes. All you need to do is decide which tables you want to place in the column store. Generally, tables that change infrequently, but are queried intensively are the highest priority candidates for this powerful feature.

SCALE YOUR APPLICATION GLOBALLY

As your business grows around the globe the performance challenges increase. If you have customers in Australia and servers in Europe, it may take longer to send and receive the request than process it in your application!

Branching into new regions may also give you regulatory challenges. Some countries may require that you only store data for its citizens within its borders.

You can address these problems with Oracle Sharding. Using it, you can build global-scale applications. This enables you to have servers around the world, yet view them as one database.

You choose a sharding key. This maps values to a set of servers. When you include this key in your requests, the database routes them to the correct location. This helps reduce the physical distance between your customers and your hardware. It also allows you to follow data sovereignty rules.

A sharded database has servers around the globe. This reduces the distance between your customers and your servers, improves fault tolerance, and may be necessary to comply with data sovereignty laws.

Even after locating servers close to your customers, you may need to reduce response times further. Placing all your data in memory ensures queries are as fast as possible.

For ultrafast processing of transactional data, Oracle TimesTen is a pure In-Memory database. This can handle millions of transactions per second. You can use this as a standalone database, or use it as a cache for a traditional Oracle Database.

By combining these technologies, you can scale your application to serve the most demanding workloads.

But performance is only one concern. Throughout the development process you need to keep data safe. Oracle Database has many tools to help you build your defenses.

KEEP SENSITIVE DATA SECURE

Hackers are constantly trying to access confidential company data. The fallout from a data breach can cost your company millions. So it's imperative you secure sensitive data throughout the development process.

The number one Open Web Application Security Project (OWASP) security risk is injection attacks. This includes SQL injection. This enables hackers to add their own code to SQL statements executed on your database.

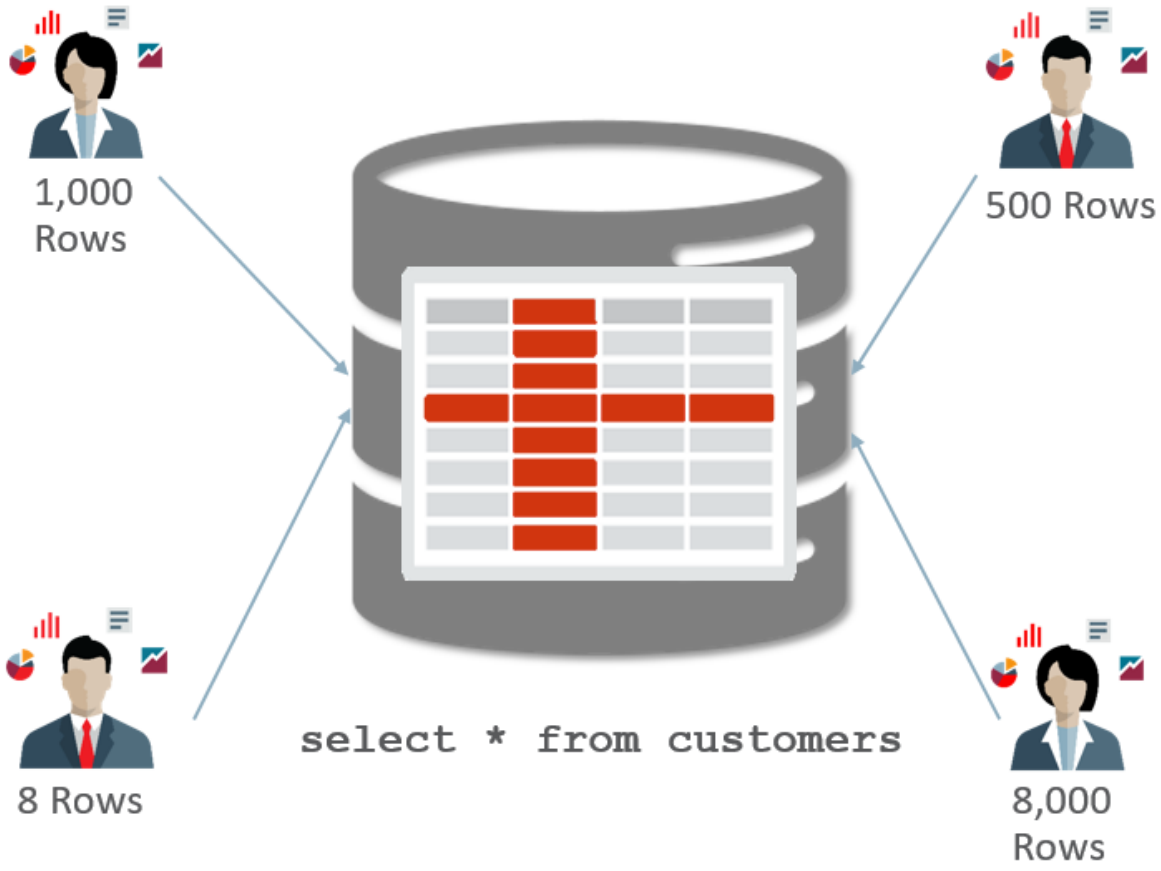
In most programming languages, it's easier to write SQL vulnerable to injection than safe statements.

PL/SQL flips this on its head. The natural way to write SQL in PL/SQL – static SQL – is secure from injection. This reduces the chance of vulnerabilities sneaking in by mistake. If your logic requires dynamic SQL, PL/SQL provides further protections to help you guard against SQL injection. These include bind variables and validation routines to check user input.

But even when you've protected against SQL injection attacks, there are still rules you need to enforce. General support staff may need to execute some high privilege tasks. Customer support teams may need to view part of customer details to verify their identities. And customers must to be able to view their personal data, but only their data.

Using PL/SQL Code Based Access Control you can grant roles to specific procedures. This enables you to restrict powerful actions to a few points in your code. Low-clearance staff can execute these procedures with no extra privileges. You can also define accessor lists on PL/SQL units to restrict which other procedures can call that unit.

You can further secure your data with Oracle Advanced Security. Using data redaction policies you can show parts of sensitive data. For example, the last 4-digits of credit card numbers or the domain of email addresses.



With row-level security you can configure policies controlling who can see which rows and columns

And with Virtual Private Database (VPD) and Real Application Security (RAS), you can apply row-level security policies. This ensures that customers and staff can only view rows and columns they have permission to see.

You can control who has access with Oracle Database Vault. This prevents unauthorized privileged user access to sensitive data and unauthorized changes to the database through strong controls and enforced separation of duties. Database Vault uses trusted paths as additional controls to protect the database from unauthorized access. Database Vault provides a powerful and transparent security solution that helps you comply with regulations, deploy systems in a cost efficient manner, and prevent unauthorized access to sensitive data.

Implementing these policies is key as your customer base grows. As the business evolves, new applications will need access to the data. Having robust security policies reduces the chance of data breaches.

MONITOR APPLICATION AND BUSINESS PERFORMANCE

Keeping an application running is a challenge. Services can fail with no warning. Releases may introduce bugs. Queries may get slower over time. And as the business grows, you need more hardware resources to run the application.

To keep on top of this, you need monitoring in place. This must track system performance and availability and watch customer behavior.

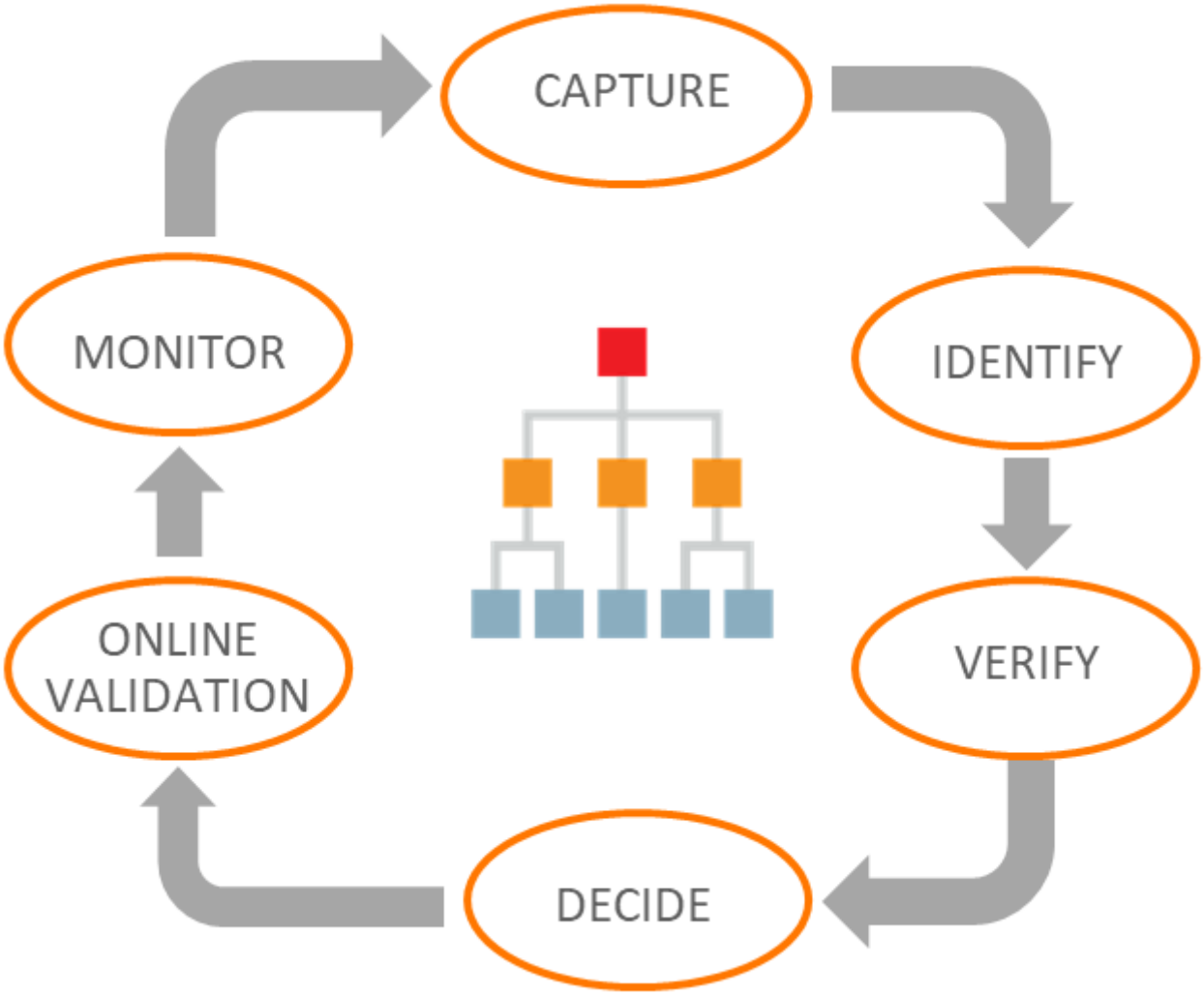
APPLICATION PERFORMANCE MONITORING

Queries getting slower is a major cause of database performance issues. Oracle's Diagnostics and Tuning Packs help you find these queries and apply fixes to make them fast again.

Using the Automatic Workload Repository, you can capture snapshots of database activity. By comparing two periods before and after a change, you can identify the SQLs that have degraded and any new SQL that requires tuning. .

The Automatic Database Diagnostic Monitor (ADDM) automatically detects and reports on performance problems with the database. This creates a report of problems and recommendations to resolve these. This includes the SQL Tuning Advisor.

This can create SQL profiles to help the optimizer find better execution plans. And SQL Plan Baselines which lock statements to specific plans, giving better stability.



Automatic Indexing monitors your SQL, identifies candidate indexes, and validates them before exposing these to the application.

When you use Oracle Autonomous Database, it gets even better. With Automatic Indexing the system proactively analyses your workload. This expert system reviews SQL statements and tests candidate indexes. It only exposes indexes that make queries faster to the rest of the application.

POWERFUL SQL ANALYTICS

There's more to application monitoring than system metrics. Your company wants to know how new changes affect customer behavior. Sales and marketing teams want to know instantly if customers hate new features, leading to a drop in orders.

```
select *
from   daily_weblogs match_recognize (
  partition by user_id
  order by access_date
  measures
    first ( access_date ) as start_date,
    count (*) as consecutive_days
  pattern ( init consecutive* )
  define consecutive as
    access_date = ( prev ( access_date ) + 1 )
);
```

With a few lines of SQL you can answer complex questions, such as how many consecutive days each customer visited your website

Oracle's advanced SQL engine enables you to answer key business questions with a few lines of SQL. This gives you access to wide range of analytic functions. These enable you to calculate running totals, do advanced statistical analysis, and split rows into groups. This makes it easy to build KPI dashboards showing business activity.

You also need to find outliers and anomalies. These can indicate hacking attempts, fraudulent transactions, or high-value customers. Machine learning helps fill the gap when you need to catch the unknown and you are not sure what you are looking for.

INTEGRATED MACHINE LEARNING

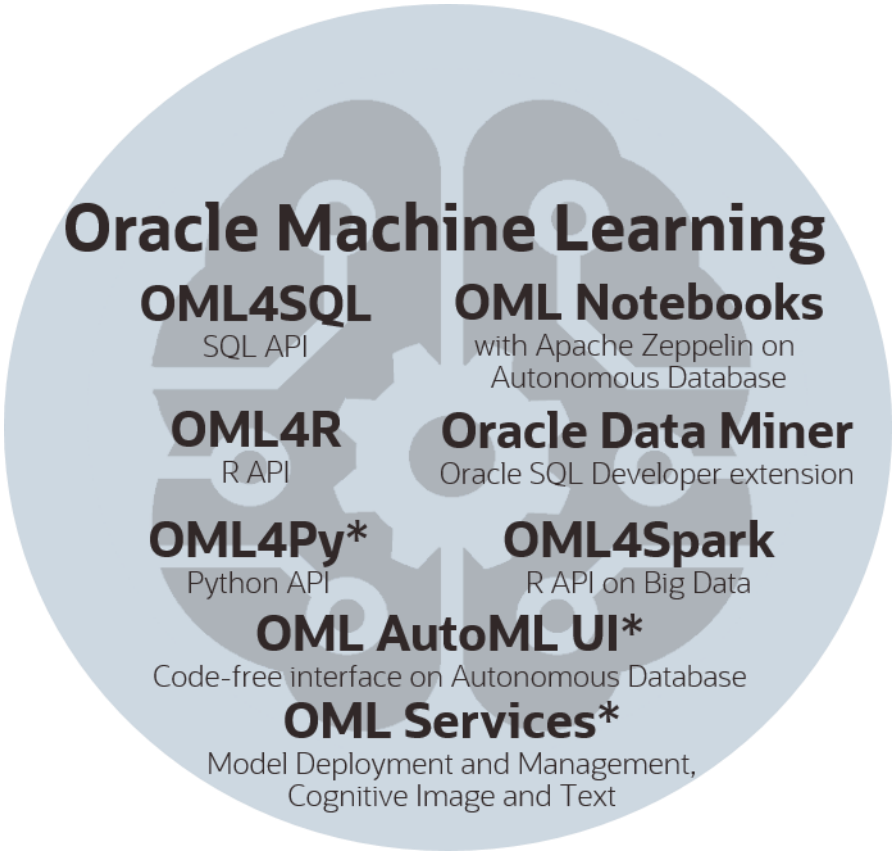
Machine learning uses a wide range of algorithms to analyze data and spot trends. This helps your business target customers with effective offers and identify fraudulent activity.

Using Oracle Machine Learning, you move the algorithms to the data. This eliminates time-consuming data movement, helping you get answers faster. You can access these algorithms using SQL, R, or Oracle SQL Developer.

Building and applying machine learning models in Oracle Database gives you the full power of the database engine. It also brings the security, scalability, reliability, and backup features of Oracle Database for managing the data. And you always have immediate access to current data.

Oracle Autonomous Database has integrated Zeppelin Notebooks which you can use to build and deploy your algorithms then visualize their results. This makes it easy to collaborate with other developers, data scientists and business users analyzing your data.

Oracle Machine Learning also supports a "drag and drop" graphical user interface, Oracle Data Miner that is integrated with Oracle SQL Developer. This is capable of generating SQL scripts from user-created analytics workflows.



*OML supports a wide range of languages and interfaces, so you can build your models with the tools you're most comfortable with. *Coming soon.*

Combining SQL and machine learning, you can build a comprehensive monitoring platform. This helps your business spot a wide range of problems. Using these insights, you can work other departments to make changes the application to increase sales, customer retention, or other business metrics.

CONCLUSION

Oracle Database makes it easy for you to handle data throughout the development process. Using Oracle Database gives you the freedom to store data how you want, the power to access it efficiently, and the strength to keep it secure:

1. You can store your data relationally, as JSON, XML, or more specialized formats such as a Graph in Oracle Database. Whichever format you choose, SQL Developer Data Modeler makes it easy to plan and view your schema, manage changes, and generate DDL scripts.
2. With Oracle APEX, you have a low-code platform backed by the power of Oracle Database to help you create beautiful applications fast. But whatever your chosen programming language, Oracle is here to support you. We have drivers for a wide range or languages or you can use ORDS to create REST APIs for technology agnostic endpoints.
3. Building new development and test databases populated with the necessary data is easy with Oracle Multitenant. Cloning PDBs enables you to get new environments up-and-running in minutes.
4. Managing test data is simple. You can define unit testing suites with Oracle SQL Developer and reset data after test runs using Flashback technologies.
5. Integration with Liquibase makes it simple to create repeatable, automated release scripts. Combined with EBR, you can deploy schema changes while the application is online with minimal human interaction.
6. Oracle's SQL monitoring tools make it easy to spot slow statements. With SQL Plan Management and Automatic Indexing the database can self-correct and resolve many performance issues.
7. You can measure business performance using Oracle's advanced SQL engine and spot anomalies with Oracle Machine Learning.
8. You can use the many performance features to scale your database and keep SQL fast as the size of your data and number of transactions grow. Partitioning splits tables into smaller, more manageable chunks, materialized views pre-compute the result of queries and Database In-Memory boosts analytic SQL performance. For truly global scale

applications you can use Oracle Sharding. This enables you to store data around the globe, yet view it as one database.

9. Throughout the process you can keep your data secure by using PL/SQL to protect against SQL injection, row-level security to control who can see what, and Oracle Database Vault to control user access and enforce separation of duties.

Using the unique capabilities of Oracle Database you can unlock the power of your data and build data-driven applications that help your business grow and meet the needs of the most demanding workloads.

GET STARTED

It's easy to start developing with Oracle Database. There are many free resources you can use to learn SQL, prototype applications or use in your development environment.

The free tier on Oracle Cloud gives you unlimited access to two Oracle Autonomous Databases and two Oracle Cloud Infrastructure Compute VMs. Combined with Block, Object, and Archive Storage you can get up and running, building applications with Oracle APEX and Oracle SQL Developer.

If you want to work locally, you can download and install Oracle Database 18c XE. This fully-featured edition of Oracle Database allows you experiment with features like Partitioning, In-Memory, and Advanced Security on your laptop. All for free!

Oracle Database 18c XE is available in many formats. You can download and install it as a standalone application, use rpm install, get a Docker image or VirtualBox VM.

If you want to play with Oracle APEX, you can request a free workspace on apex.oracle.com. This hosted environment allows you to try out APEX.

Or if you're looking for a sandbox to learn Oracle SQL, head to Live SQL. This free, browser-based tool gives you an interactive client to run SQL. You can save scripts in a private repository or share them with others. The code library includes a wide range of tutorials teaching you how to use Oracle SQL. This makes Live SQL the perfect place to start learning how to write SQL.

Head to https://cloud.oracle.com/free and get started today!

FOR FURTHER READING

- Oracle Free Cloud Tier
- Database Development Guide
- Oracle Application Express App Builder User's Guide
- JSON Developer's Guide
- XML DB Developer's Guide
- Spatial and Graph Developer's Guide
- Oracle Machine Learning for SQL User's Guide
- Database Data Warehousing Guide
- Database Performance Tuning Guide
- Using Oracle Sharding
- Database Security Guide
- Oracle SQL Developer
- Using Liquibase with SQLcl
- Oracle Database XE
- Oracle Live SQL
- Get Started with Oracle Database

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Modern Applciation Development with Oracle Database
September, 2020
Author: Chris Saxon
Contributing Authors: Louise Morin, Henry Byorum