

UD1. Diseño de Bases de datos

Índice

Diseño de bases de datos.....	2
El modelo de datos.....	3
Fases del diseño.....	5
Análisis de requisitos.....	5
El modelo Entidad-Relación.....	6
El diseño lógico.....	6
El diseño físico.....	7
Los sistemas gestores de bases de datos.....	8
Arquitectura de los SGBD.....	9
Ejemplos de SGBD.....	9
Componentes de los SGBD.....	10
Funciones del SGBD.....	11
El DBA.....	11
Lenguajes para la gestión de bases de datos.....	12
Explotación de bases de datos.....	12
Minería de datos.....	13
Las tecnologías Business Intelligence.....	14
Bases de datos en la nube.....	15
Big data.....	16
El Administrador de la base de datos. El DBA.....	17
Crear una cuenta de usuario nueva con SQL*Plus.....	18
El SQLDeveloper.....	19
El SQL DataModeler.....	20
Conexión a la base de datos con SQLDeveloper.....	20
Crear una cuenta de usuario con el SQLDeveloper.....	22
El diccionario de datos de Oracle.....	23

Diseño de bases de datos

El objetivo del diseño de una base de datos depende de la naturaleza de los datos

y de cómo estos se relacionan entre sí. Puede ocurrir que no exista una gran conexión entre los datos de un almacén de datos porque se trate de demasiadas instancias de muy pocos tipos que se puedan instanciar. De esta manera, no se consigue una estructura clara para almacenar los datos, más bien parece una colección de datos, uno de detrás de otro, sin demasiadas categorías que permitan estructurarlo.

Otras veces, existen tantos tipos de datos que el diseño de las estructuras para registrarlos es tan complejo que se necesitan metodologías muy avanzadas para realizar un diseño óptimo.

Los casos más estándares son aquellos que tienen un conjunto de tipos de datos y millones de instancias de estos tipos. Este tipo de bases de datos casa extraordinariamente bien con una filosofía de trabajo denominada relacional. El elemento que almacena los datos se denomina relación. Estas relaciones se convierten en tablas cuyas columnas son las diversas propiedades de los tipos de datos que se pueden determinar. Las filas son las instancias de estos tipos. Este modelo de diseño se denomina modelo relacional.

Este modelo está compuesto por una serie de fases y cada una tiene un objetivo particular, determinar los conceptos principales de una base de datos, convertirlos a relaciones con tuplas y construir las tablas que almacenan los registros.

Otras veces, no es posible estructurar la información, por lo que se hace necesario usar otros tipos de técnicas de modelado. Por ejemplo, los sistemas Big Data tienen una colección tan gigante de datos que no se pueden estructurar, por eso lo importante no son dichas estructuras, sino el modo en que se accede a los datos que se requiere del modo más rápido posible, tan rápido como el tiempo que se dispone para tomar decisiones en tiempo real. Estas técnicas usan algoritmos matemáticos avanzados llamados inteligencia artificial, porque, de alguna manera, simulan el razonamiento humano.

Las bases de datos relacionales tienen técnicas no muy complicadas que permiten su diseño. Se parte de un conjunto de requisitos o requerimientos que el equipo de análisis ha sabido detectar a través de las reuniones con la organización que solicita la implantación de la base de datos.

Estos requisitos respecto a la base de datos se puede llamar requisitos de datos o requisitos de información. Estos requisitos se transformarán a una representación conceptual que usa un diagrama denominado diagrama conceptual. El más popular es el diagrama entidad-relación, compuesto por una serie de rectángulos relacionados y representados a través de rombos. Otro usa los elementos de un lenguaje de modelado denominado UML. UML está orientado a la ingeniería del software y se convierte en una herramienta para la representación del diseño de diferentes elementos de toda una aplicación informática.

Una vez que se dispone del diagrama conceptual, se puede transformar a otro diagrama compuesto de un grafo, en el que cada nodo representa una relación y las aristas, las claves ajenas que apuntan a las columnas clave de otra relación. Este diagrama se denomina diagrama relacional o esquema relacional.

El esquema relacional ya está cerca del diseño físico, puesto que la transformación es casi directa. Cada relación del esquema relacional se convierte en una tabla. Las propiedades de cada relación son las columnas de la tabla y las restricciones de la relación, las restricciones de la tabla. Cuando una base de datos está diseñada físicamente, comienza su vida y sus usuarios comienzan a introducir registros en las tablas que completan. Las fases se pueden resumir en las siguientes líneas:

1. Modelado de datos: decidir la filosofía estructural que mejor se adapta a la naturaleza de los datos.
2. Análisis de requisitos: determinar cada uno de los requisitos que deben cumplir los datos y sus relaciones.
3. Diseño conceptual: representar a través de diagramas los elementos conceptuales, es decir, determinar sobre qué se quiere registrar información, cuáles son sus características o propiedades y cómo se relacionan.
4. Diseño lógico: transformar el diagrama conceptual con el fin de obtener un diagrama que represente los elementos relacionales. Estos elementos serán relaciones con tuplas formadas por diferentes valores atendiendo a una descripción.
5. Diseño físico: las relaciones del diagrama relacional se convierten en tablas. Se diseñan todos los aspectos físicos de la base de datos que se puedan determinar antes del comienzo de su vida.
6. Explotación: la explotación es la etapa más duradera y la que le da significado a toda la implantación. La explotación también influye en el diseño físico.

El modelo de datos

Los modelos de datos constituyen el medio para representar el conjunto de reglas y símbolos que posee un lenguaje de modelado. De este modo, el modelo lo forma el conjunto de abstractores que representan los elementos principales, sus relaciones y asociaciones y que atienden a un contexto del mundo real. Por ello, describen el universo del discurso concretando la realidad en la que los agentes trabajan. Los elementos principales pueden ser físico, abstractos o hipotéticos.

Todo modelo de datos parte de un **universo del discurso** que se transformará a un modelo de datos para dar soporte al lenguaje de modelado. El universo del discurso se puede definir con el conjunto de necesidades que cumplir. Se llamarán requisitos de información, ya que son las necesidades relacionadas con los datos. El esquema representa un nivel de abstracción del proceso de construcción de una base de datos y hace referencia a la estructura. Sin embargo, cada situación de ese esquema se llama ejemplar del esquema, entendiendo como tal a la colección de datos que se originan al instanciar los elementos del esquema. Por ejemplo, el tipo de entidad Profesor es un

abstractor en el nivel conceptual, compuesto por varias propiedades, dni, nombre, prApellido, fecha de nacimiento, cuenta bancaria, etc. Cuando se dan ejemplares de este tipo de entidad, los atributos toman valores, uno por cada propiedad, y se origina lo que se denomina instancia u ocurrencia de un tipo de entidad. Por ejemplo, Raquel Sánchez Beltrán es profesora si se instancia para este tipo de entidad y existen valores para todas las propiedades obligatorias de este tipo de entidad. Los abstractores serán diferentes para cada nivel de abstracción.

Todo modelo tendrá una parte que no cambia y otra que si. La **parte estática** la forman los símbolos de la notación, el conjunto de reglas que aplicar y las restricciones del modelo. El conjunto de operaciones que hacen que las instancias se originen, cambien y desaparezcan forman la **parte dinámica**. Los elementos estáticos serán los conceptos del universo del discurso, que podrán ser cosas o personas y que, para generalizar, se llamarán tipo de entidad y sus instancias serán las entidades. A estas las define y les da existencia abstracta o concreta una serie de características que se llamarán propiedades de los tipos de entidad. Un tipo de entidad aislado no es parte del modelo de datos, sino que se asocia con otros. Una tabla de históricos de precios o una tabla generada por una consulta podría ser una tabla aislada que se usa con fines de almacenamientos a largo tiempo o cálculos para análisis. Estos elementos no forman parte del modelo estático, sino que, más bien, son parte del modelo dinámico, en particular, de un nivel de abstracción llamado relacional.

Los tipos de entidad y sus relaciones están sujetos a múltiples restricciones derivadas del propio universo del discurso que los limitan y les dan integridad.

Una estrategia de modelos de datos definida por instituciones y comités sobre normas, requisitos y planificación de estándares es la conocida con cuatro niveles de abstracción denominados nivel externo, nivel conceptual, nivel interno y nivel físico. El primero, el más abstracto para la máquina, pero el menos abstracto para los agentes, describe el modo en el que estos ven a la base de datos y está formado por las interfaces de aplicaciones y los lenguajes de gestión de datos.

El siguiente nivel menos abstracto es el conceptual y representa cada elemento del universo del discurso. Se usan los modelos de representación conceptual para establecer sus componentes, relaciones, restricciones, integridad y seguridad. Un ejemplo de modelo conceptual es el llamado entidad-relación.

En un nivel más inferior, se encuentra el nivel interno, en el que se expresa cómo los datos serán representados, estructurados y almacenados físicamente y accesibles por un software denominado sistemas gestores de bases de datos. Un ejemplo es el esquema relacional.

Tanto el nivel externo como el conceptual definen el modelo lógico y el nivel físico describe el modelo físico.

El modelo entidad—relación (MER) permite crear una representación básica de una base de datos en términos de entidades, atributos y propiedades, relaciones entre entidades y restricciones o reglas que permiten la integridad de los datos. Este modelo es independiente de las herramientas que se usan para la implementación de bases de datos.

Fases del diseño

Los pasos generales en el diseño de una base de datos se pueden resumir con los siguientes:

1. Describir en lenguaje natural los requisitos que debe cumplir la base de datos.
2. Diseñar el diagrama conceptual. El diagrama entidad-relación o diagrama de Chen modeliza el problema mediante entidades asociadas por relaciones.
3. Elegir el modelo de datos, por ejemplo, relacional.
4. Transformar el diseño conceptual a un modelo relacional, obteniendo así un conjunto de tablas relacionadas entre si.
5. Normalizar aquellos elementos que tienen defectos de diseño atendiendo a diferentes reglas.
6. Optimizar la solución obtenida en fases anteriores según criterios de almacenamiento físicos, espacio de disco, tiempo de acceso a datos, etcétera.

Análisis de requisitos

Cuando se diseñan bases de datos, se hace imprescindible acotar el contexto o universo del discurso en el que se integran. Para que los datos sean fieles, se debe conceptualizar a través de ideas y definiciones que den soporte a este microuniverso. Esta representación se denomina modelo conceptual. Partiendo de este modelo, se obtendrá una descripción de los datos compuesta por un conjunto de relaciones llamado esquema relacional de datos. Por último, este esquema se traducirá a estructuras físicas de almacenamiento.

Para llevar a cabo este procedimiento, se hace necesario emplear un conjunto de herramientas para diseñar los datos y sus relaciones y otras que permitan llevar las acciones con las bases de datos. La primera es el modelo de datos y la segunda, el sistema gestor de la base de datos. Por tanto, es el modelo de datos el que permite el diseño de una base de datos.

La tarea más importante es la detección del conjunto de requisitos de información que el sistema debe cumplir. De hecho, el conjunto de requisitos supone el primer diseño de la base de datos. El detalle de la información y la claridad con la que se exponen afecta al desarrollo completo de una base de datos. Pulir los requisitos de información es una tarea bastante importante. Se deben analizar al detalle, con el objetivo de detectar si falta información, en cuyo caso, serán necesarias nuevas entrevistas para obtenerlas o herramientas para deducirlas. Esta información que no está en el conjunto de requisitos de información se puede denominar conjunto de consideraciones semánticas externas.

El modelo Entidad-Relación

A través de esta técnica, se representan y se definen todos los datos que se introducen, almacenan, transforman y producen independientemente de la tecnología que se pretenda usar. El modelo de datos es el medio para comunicar el significado de los datos, las relaciones entre ellos y las reglas de negocio. Las ventajas de realizar un modelo de datos son:

- Control de los posibles errores.
- Elaboración de estructuras de datos independientes de la arquitectura física.
- Entendimiento de los datos.
- Perfeccionamiento del mantenimiento.

La técnica persigue la representación a través de un diagrama que se centra en los datos y que no depende de cómo se transforma ni se detiene en aspectos sobre la eficiencia.

El modelo entidad-relación con un alto nivel de abstracción describe cómo se distribuyen los datos. Los elementos principales de este modelo son las entidades y las propiedades. En la literatura sobre el modelo entidad-relación, existen unos elementos añadidos que dan nombre al modelo entidad—relación extendido. Las extensiones al modelo básico añaden elementos como la jerarquía entre entidades.

Existen muchas herramientas que ayudan a desarrollar esquemas conceptuales en bases de datos relacionales. Una de las herramientas más avanzadas es DataModeler, un producto de Oracle que puede usarse sin necesidad de emplear ningún sistema gestor de bases de datos ni su producto SQL Developer, que integra junto con este en su entorno de desarrollo. Es decir, en un ordenador, se puede tener instalado DataModeler para obtener los esquemas relacionales y otras potentes funciones, que se verán en un futuro.

El diseño lógico

Una vez modelado conceptualmente el universo del discurso, se obtiene un diagrama conceptual que representa los elementos de los que se quiere almacenar información, sus propiedades y cómo se relacionan entre sí. En la siguiente fase de diseño, la etapa de diseño lógico, se deben transformar estos elementos en otros que estén más cerca del paradigma en el que interesa trabajar, por ejemplo, en un paradigma relacional. Por tanto, cada uno de esos elementos deben ser transformados para conseguir un conjunto de tablas compuestas por columnas y que se relacionan entre sí.

Para desarrollar estas transformaciones, se utilizara un conjunto de heurísticas que se convierten en reglas automáticas para cada uno de los elementos del diagrama entidad-relación extendido. Por tanto, en esta fase, se llevan a cabo, con el conocimiento de cada una de las reglas de transformación, todos los cambios posibles.

De este modo, el diseño lógico es la etapa del proceso de diseño de una base de datos en la que se obtiene la representación de su estructura entérminos de almacenamiento. El modelo entidad-relación se usa para el modelado conceptual y el relacional para el lógico

El diseño físico

En esta etapa, se pretende conseguir un esquema interno creado a través del sistema gestor de bases de datos elegido. Este esquema interno debe funcionar según los requerimientos que los usuarios hayan planteado en la etapa de análisis. Además de cumplir con estos requisitos, el diseño físico contemplará los mecanismos que permitan disminuir tanto el tiempo de respuesta como el espacio de almacenamiento usado. Los procesos asociados con la seguridad también son aspectos asociados al diseño físico. En esta fase, se hace necesario conocer los recursos hardware y software con los que se cuenta, además del esquema lógico obtenido en la etapa anterior. Se estudiarán las políticas de seguridad de los datos, el conjunto de aplicaciones que interactuará con la base de datos y las transacciones que esta interacción generará.

En este proceso, se hace necesario usar un conjunto de heurísticas que permitan optimizar el uso de los recursos y los procesos de accesos a los datos y su seguridad. Estas heurísticas o reglas de aproximación son dependientes del SGBD, con lo que se buscará la definición de las estructuras de archivos, la indexación de tablas, el uso del espacio de memoria principal, los roles de seguridad y los objetos de gestión.

Estos pasos serán testeados y puestos a evaluación hasta su refinación. Este proceso de mejora se conoce como ajuste de la base de datos o proceso tuning. El tuning o afinamiento es el proceso sistemático de búsqueda de origen, naturaleza y corrección de los problemas de rendimiento de un sistema de bases de datos. Se usarán diferentes comandos del SGBD para llevar estos objetivos.

El diseño físico de las consultas es una tarea importante en este proceso. Las consultas acceden a un conjunto de tablas para hacer operaciones de interés. Una consulta está compuesta por la selección de columnas que se quiere proyectar, cláusula SELECT; un conjunto de tablas combinadas por algunas columnas, cláusula FROM, y unas condiciones que cumplir sujetas a las posibles columnas de las tablas seleccionadas.

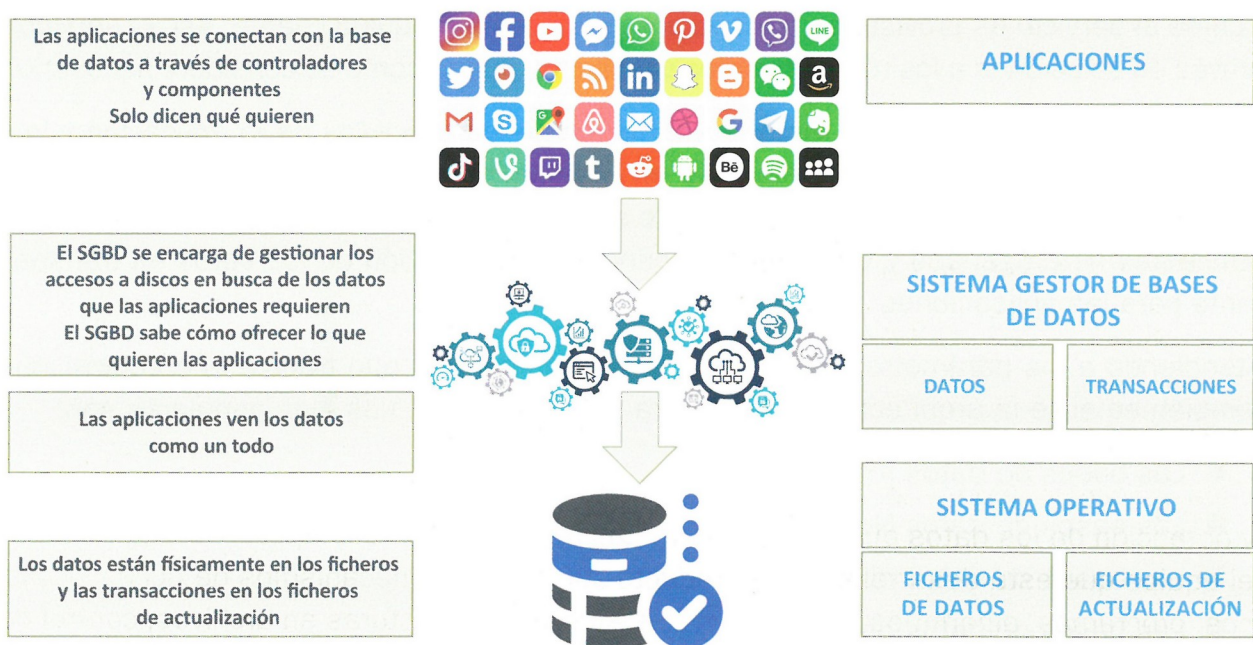
Por motivos de implementación, en la etapa física, se pueden tomar decisiones sobre la redundancia de datos con el fin de aumentar la eficiencia temporal en las acciones de accesos a datos. La redundancia de datos evita buscar información extra en tablas gigantes, lo que aumentaría el tiempo en las operaciones asociadas a estas tablas. Para evitar esos accesos, los datos que se requieren pueden estar incluidos en las tablas iniciales. Este proceso se puede llamar desnormalización de tablas. La teoría de la normalización busca tablas eficientes espacialmente, pero, en ocasiones, se pueden hacer excepciones en busca de esta eficiencia temporal. Un ejemplo podría ser almacenar el precio de coste y venta actual de un producto, aunque este se encuentre incluido en la tabla Precio que almacena los diferentes valores que toma un producto en cuestión.

Esta técnica hace que las actualizaciones de los datos sean más complicadas, ya que el mismo valor está en más de un lugar. Se emplearán técnicas de programación para llevar el control de estos valores. Esto ralentiza dichas acciones y hace que el esquema lógico pierda flexibilidad. Por ello, es importante poner en una balanza los aspectos que se desean mejorar.

Los sistemas gestores de bases de datos

Los sistemas gestores de bases de datos se convierten en la herramienta informática que permite definir, crear y dar mantenimiento a las bases de datos y, luego, su administración, con el fin de proporcionar acceso controlado y gestión eficiente del espacio de almacenamiento y de la explotación de los recursos, entre otros objetivos principales. Esta gestión completa actúa como interfaz intermedia entre las aplicaciones que requieren, crean y usan los datos y el almacén masivo de datos.

Desde este punto de vista, la base de datos es la colección de todos los datos creados y manipulados por los usuarios a través de las aplicaciones, mientras que las aplicaciones son los agentes que usan el sistema gestor de bases de datos. El propósito de cada aplicación puede variar, desde permitir el alta, baja y modificación de los datos almacenados en repositorios hasta aplicaciones destinadas a la administración del sistema gestor.



Cuando no se usan sistemas gestores, se hace necesario gestionar los elementos que almacenan físicamente los datos en los soportes de almacenamiento masivos. Estos elementos se denominan archivos o ficheros. Así, una de las tareas fundamentales de un SGBD es hacer transparente la gestión de sistemas de ficheros. Por tanto, si no se dispone de un SGBD entre las aplicaciones y los datos, son las propias aplicaciones las que deben implementar su propio sistema de gestión de ficheros. Esto conllevaba una alta redundancia de los datos, ya que cada aplicación usaría sus ficheros de almacenamiento de datos, desaprovechando el espacio físico en disco. Además, los tiempos de proceso aumentarían, ya que se repiten las mismas acciones implementadas en cada aplicación. La posibilidad de información inconsistente es muy alta en este tipo de sistemas, ya que, al repetirse la información en innumerables ficheros, existe mucha posibilidad de volverse corruptos. Como cada aplicación está implementada de forma aislada con otras, los datos también lo están, por lo que aumentaría enormemente la redundancia.

Arquitectura de los SGBD

Una arquitectura es el conjunto de componentes funcionales desarrollados bajo convenciones, estándares, procesos y reglas con el fin de integrar una serie de productos y servicios para su explotación en una organización. Los modelos de arquitectura en bases de datos más usados son cliente-servidor, distribuidos, orientados a la documentación, bases de datos noSQL y Big Data.

En la arquitectura cliente-servidor, el cliente es la máquina que necesita un servicio y busca en la red otra máquina que lo provea de dicho servicio, el servidor. En cierto modo, dónde y cómo el servicio es provisto es transparente para dicha máquina cliente. Así, el servidor provee de soluciones a los requerimientos del cliente, tantos como se considere necesario.

En la arquitectura distribuida, tanto la interfaz como los procesos de la aplicación y los datos locales necesarios están alojados en la máquina cliente, el resto de datos está alojado en la máquina servidor. Esto permite accesos a datos que están almacenados en ambientes heterogéneos y, como ventaja principal, la ubicación de los datos es transparente para las aplicaciones.

Atendiendo a los parámetros técnicos de los datos con los que trabaja la organización, también se elige la arquitectura óptima para su tratamiento. Los más populares son:

- Las bases de datos relacionales.
- Las bases de datos NoSQL.
- Las bases de datos objeto-relacionales.
- Las bases de datos objeto.

Ejemplos de SGBD

A continuación, se ejemplifican sistemas gestores de bases de datos actuales y sus características más interesantes:

Access de Microsoft: es de manejo muy sencillo y está diseñado para bases de datos muy pequeñas. Está orientado a usuarios individuales o pequeños equipos.

Apache **Derby**: también es de tamaño muy reducido, ocupando 3,5 MB para el motor y el controlador JDBC que viene integrado. Es relacional y de código abierto implementada en Java. Está disponible bajo licencia Apache. Permite multiplataforma, es muy portable y se puede configurar en arquitectura embebida o cliente—servidor.

Apache **CouchDB**: no es relacional y está orientado a documentos. Utiliza lenguaje NoSQL, como JSON, y está bajo licencia Apache. Usa el Couch Replication Protocol explotable en entornos de clústeres de servidores distribuidos. También se puede usar MapReduce para la recuperación de datos.

DB2 de IBM: este SGBD es muy potente. Está diseñado para uso en multiplataformas. Integra el lenguaje XML de forma nativa, pureXML. Su versión gratuita se llama DB2 Express. Permite la búsqueda jerárquica en los documentos XML integrándolo con búsquedas relacionales

FireBird: es relacional, multiplataforma, de bajo consumo y la gestión de la concurrencia es óptima. Es de código abierto.

Informix de IBM: es un SGBD sencillo, confiable, relacional y multiplataforma. También usa principalmente el lenguaje XML. Es un SGBD que consume menos recursos que Oracle.

MariaDB: es un SGBD multiplataforma derivado de MySQL. Tiene licencia GPL.

MongoDB: es un SGBD multiplataforma no relacional. Usa BSON con esquema dinámico.

MySQL: es un SGBD relacional, multihilo, multiusuario y multiplataforma. Está diseñado para su explotación en entornos web y también funciona bien en bases de datos de tamaño medio. Hay una versión comercial y otra libre.

SQL Server de Microsoft: esta es la propuesta de Microsoft. Es de arquitectura principalmente cliente—servidor, relacional y mayor competidor de Oracle.

Oracle: es un SGBD relacional multiplataforma, confiable y seguro. Su arquitectura principal es cliente—servidor y está orientándose a otros tipos de modelos. Usa también el lenguaje JSON. Ofrece una versión gratuita, Express Edition. Es un producto muy competitivo.

OrientDB: orientado a documentos con licencia Apache.

PostgreSQL: es un SGBD multiplataforma y permite la gestión de objetos, por lo que se convierte en el mejor sistema relacional-objetos. Es el SGBD libre más avanzado del mundo.

SQLite: no es propiamente un SGBD, pero tiene funciones suficientes que permiten la gestión de datos. Trabaja con tablas y permite accesos muy rápidos. Es multiplataforma.

SyBase: es un SGBD relacional, altamente escalable y de bajo costo.

Componentes de los SGBD

Las bases de datos principalmente están compuestas por la colección masiva de datos correspondiente a los procesos de gestión de la información. Además de la base principal, existen otros elementos en un entorno de una base de datos:

Datos: la colección de datos, tanto los propios del usuario como aquellos que permiten realizar las operaciones de consultas sobre ellos. Los datos pueden estar encriptados para aumentar la seguridad del sistema.

Metadatos: son datos que registran el comportamiento de la información almacenada en la base de datos. Se dice que componen la información de la información. Toda esta información sobre cualquier objeto que existe en la base de datos se almacena en un espacio lógico especial denominado diccionario de datos.

Sistemas gestores de bases de datos: conjunto de aplicaciones que permiten la gestión completa de los datos almacenados en la colección de datos. Su núcleo se denomina motor y es el encargado de gestionar los accesos a los datos físicos y ofrecer un

mecanismo de abstracción a los niveles superiores que requieren de información demandada por los usuarios de la base de datos.

Usuarios: cada uno de los modos de acceso a los datos se denomina usuario de la base de datos. Generalmente, se asocia a aquellas personas que explotan la base de datos, pero una persona también puede tener diferentes modos de acceder a las bases de datos. Cada uno de ellos es un usuario. Una aplicación informática también puede ser un usuario de la base de datos. Las cuentas de usuario identifican al usuario y emplean un identificador y una contraseña que lo autentica. Un mismo usuario puede tener diferentes modos de acceso y de explotación y utilizar los roles y perfiles para este cometido.

Funciones del SBBB

El sistema gestor de bases de datos (SGBD) actúa como interfaz entre los datos y las aplicaciones que requieren esos datos y proporciona mecanismos para definirlos, almacenarlos de forma óptima, mantenerlos y manipularlos. Además, ofrece potentes herramientas para la administración orientada, principalmente, a la gestión de los usuarios y el almacenamiento de los datos, gestión de instancias, control de transacciones, recuperación y copias de seguridad.

Las funciones principales que destacar de un sistema gestor de bases de datos son:

Definición y control centralizado: descripción de los datos y las interrelaciones entre las estructuras que los almacenan. Mantiene toda la descripción en el diccionario de datos.

Manipulación de los datos: mecanismos para consultar y transformar los datos almacenados y para añadir nuevos datos. La evolución del esquema de la base de datos debe interferir lo menos posible en las aplicaciones que interactúa con la base de datos.

Seguridad e integridad: mecanismos para la gestión de accesos autorizados, el control de vistas externas de usuario y el control de las restricciones, garantizando integridad y consistencia de los datos.

Tareas de administración: copias de seguridad, recuperación ante fallos, explotación del sistema mediante roles, control de privilegios, etcétera.

Control de la concurrencia: se debe garantizar el acceso simultáneo a los datos y la veracidad de estos.

El DBA

El administrador es el responsable final del mantenimiento de toda la base de datos. Cuando el sistema es muy complejo, este perfil puede dividirse en otros administradores con funciones más concretas. De cualquier forma, siempre existe un administrador principal que tiene el control de toda la base de datos. Aparte de este usuario administrador, pueden existir diferentes administradores con diferentes perfiles administrativos.

Lenguajes para la gestión de bases de datos

La interacción del usuario con las bases de datos debe efectuarse a través de alguna técnica que haga fácil la comunicación y que permita al usuario centrarse en el problema que desea solucionar, más que en la forma de expresarlo con las técnicas que se le suministran. La mejor forma de alcanzar este objetivo es darle un lenguaje parecido al lenguaje natural, que le permita expresar de forma sencilla los requerimientos.

Los lenguajes en los sistemas gestores de bases de datos permiten la comunicación a la base de datos a través de comandos. Se persigue que estos comandos posean el mayor nivel de abstracción posible y se diseñan con el fin de que se parezcan al lenguaje natural con los elementos léxicos, sintácticos y semánticos necesarios y suficientes para aportar la máxima flexibilidad posible.

Dependiendo de la finalidad del conjunto de comandos que pertenece al lenguaje, se establecen varios tipos:

Lenguaje de definición de datos (Data Definition Language o DDL): son los comandos que usa el administrador de la base de datos para la construcción de sus elementos. Ejemplos de comandos DDL son CREATE, ALTER y DROP.

Lenguaje de manipulación de datos (Data Manipulation Language o DML): son los comandos que usan los usuarios avanzados que permiten efectuar las operaciones sobre las bases de datos. Estas operaciones son el alta de nuevos datos, la supresión de datos registrados, la modificación de valores que cambian, las consultas de datos y la gestión de vistas. Estos comandos pueden embeberse en otros lenguajes de programación que los programadores usan para desarrollar las aplicaciones. Ejemplo de comandos DML son INSERT, DELETE y UPDATE.

Lenguaje de control de datos (Data Control Language o DCL): son los comandos que, generalmente, usan los administradores para controlar el acceso a los datos. Ejemplos de comandos DCL son GRANT y REVOKE.

Lenguaje de control de transacciones (Transaction Control Language o TCL): son los comandos que permiten gestionar las transacciones. Una transacción es una operación atómica que se hace sobre un conjunto de datos y no puede dividirse, por lo que se debe garantizar que comienza y termina de forma correcta. Ejemplos de comandos TCL son COMMIT, ROLLBACK y SET TRANSACTION.

Explotación de bases de datos

El análisis de requisitos, el diseño conceptual y el diseño lógico no son útiles si no se tiene claro cuál es el uso que se le va a dar a la colección de datos y cuáles serán los resultados que obtener. La finalidad de cualquier conjunto de datos es su análisis para tomar decisiones que mejore la situación de la organización. De nada sirve definir rúbricas altamente bien diseñadas si luego no se toman decisiones con la información obtenida.

La tendencia sobre la explotación de las bases de datos es cada vez más global, desde un punto de vista mayor que el simple mantenimiento de miles de yottas de datos en cintas magnéticas. La inteligencia artificial y las técnicas de Big Data están

revolucionando los objetivos de la explotación de los almacenes de datos. La aparición de robots con inteligencia suficiente para tomar decisiones, a través de todos estos nuevos mecanismos, harán que los almacenes de datos tomen una importancia antes no tenida en cuenta. Los almacenes de datos y los algoritmos que los maneje comportarán la base de las aplicaciones del mañana y de la nueva revolución industrial. De este modo, los humanos irán de la mano de las máquinas en todo aquello que tengan que hacer.

Minería de datos

La minería de datos, Data Mining, es la técnica que recopila datos en la búsqueda de reglas y estructuras a partir de los datos localizados en un almacén de datos. El objeto es la búsqueda del conocimiento en las bases de datos. La plataforma con la que trabaja la minería de datos es el almacén de datos y ofrece una serie de técnicas que automatizan, en la medida de lo posible, la extracción de información oculta en ella, trabajando bajo las siguientes líneas de actuación:

- Se pretende construir patrones o categorías.
- Se incluye el análisis de información aplicado en los sistemas de soporte a la decisión.
- Se automatizan los procesos de planteamiento y descubrimiento de hechos e hipótesis.
- Se pretende predecir futuras tendencias y comportamiento.
- Se buscan relaciones y patrones globales ocultos en los almacenes de datos.

El procedimiento para resolver un problema a través de la minería de datos se divide en dos grandes etapas: la preparación de los datos y la minería de datos propiamente dicha.

1. Preparación de los datos: se parte del planteamiento del problema definiéndolo y determinando los recursos, las fuentes de información y su disponibilidad. Se seleccionan las fuentes válidas, realizando un tratamiento y estructuración de la información, eliminando la información no útil y detectando otra que falta. Se unifican criterios de presentación, se eliminan la redundancia y los duplicados.

2. Minería de datos: se seleccionan las técnicas, los algoritmos en búsqueda de patrones, los modelos y parámetros apropiados. Se interpretan los resultados en busca de modelos útiles en la toma de decisión y se evalúan usando diferentes casos de aplicación.

Las técnicas más usadas en minería de datos son:

Análisis multidimensional: se cruzan datos de múltiples formas y con distintos niveles de agregación. Se basan en las bases de datos multidimensionales y en las técnicas OLAP. (El área de procesamiento analítico en línea (Online Analytical Processing u OLAP) trata de las herramientas y las técnicas para el análisis de los datos que pueden dar respuestas casi instantáneas a las consultas que soliciten datos resumidos, aunque la base de datos sea extremadamente grande.

Agentes inteligentes: análisis para detectar patrones y relaciones. Se usan técnicas con metodologías basadas en sistemas expertos y el aprendizaje automático. Los

algoritmos usados son muy avanzados y emplean estructuras complejas —como pueden ser las redes neuronales y los árboles de decisión— y avanzadas técnicas —tales como el agrupamiento y la lógica difusa—. Estas técnicas se usan en gran medida para la detección de patrones.

Consultas e informes: son las metodologías tradicionales a través de consultas y usan la estadística para el análisis multidimensional y agentes inteligentes. Son metodologías muy útiles para bases de datos relacionales.

Detección de alarmas: se ejecutan agentes para desencadenar acciones extraordinarias.

Las tecnologías Business Intelligence

La inteligencia de negocios o Business Intelligence (BI) es un conjunto de arquitecturas técnicas, aplicaciones, datos, estrategias, productos y tecnologías enfocadas a la creación y administración de conocimiento, a través del análisis de los datos almacenados en grandes almacenes de datos y que facilita la toma de decisiones. Por ejemplo, los tableros o Dashboards son interfaces gráficas que permiten leer los datos recopilados de una forma cómoda y flexible a través de gráficas y tablas. Muestran resúmenes, tendencias clave, comparaciones y excepciones en históricos, entre muchos otros, con el fin de tomar decisiones óptimas. Otra herramienta, los informes ad hoc, dan respuestas a preguntas comerciales de interés. Se basan en los esquemas DSS estudiados con anterioridad y se enmarca como producto comercial. También se usa comercialmente la herramienta denominada descubrimiento de datos. Esta consiste en la búsqueda de patrones en bases de datos gigantes.

Las tendencias en Business Intelligence actuales son:

Tecnología de **Analytics**: software de minería de datos e inteligencia artificial (Data Mining, Deep Learning y Machine Learning). Ejemplos: Apache Spark, Azure, IBM Watson Analytics y Oracle Analytics

Tecnología de **Big Data**: software para manejar grandes volúmenes de datos provenientes de diversas fuentes y que se generan a gran velocidad, siendo necesario su tratamiento en tiempo real. Ejemplos: Apache Hadoop, AdoreBoard, CrownEditlon, Domo, Panoply, Text Emotions, Teradata y Tibco

Tecnología del **lenguaje**: software del procesamiento del lenguaje natural (Natural Processing Language o NPL). Ejemplos: Apache Lucene and Soir, Apache OpenNLP, Apache UIMA, Azure, Affectiva, DeepFase, GATE, Natural Language Toolkit, Kairos, SightCorp y SkyBiometry.

Tecnologías Integradoras: integran más de un tipo de tecnología. Ejemplos: SAP BI y Power BI

Bases de datos en la nube

Las bases de datos en la nube representan el conjunto de servidores y Sistemas de almacenamiento masivos que otras organizaciones alquilan a las organizaciones que explotan las máquinas remotas. Las bases de datos están alojadas en cualquier lugar del mundo, a través de redes de ordenadores complejas que permiten la ejecución de aplicaciones y accesos remotos como si estuvieran en la máquina local. Son los proveedores de estos servicios los encargados de mantener el sistema, siempre fiable las 24 horas del día y los 7 días de la semana. El cliente no requiere complejos sistemas, ya que toda la arquitectura es explotada en remoto.

Los modelos de implantación de bases de datos en la nube son principalmente dos:

1. **Bases de datos en la nube:** la gestión es análoga a la que se haría si la base de datos estuviera alojada físicamente en la intranet de la organización, pero el sistema está alojado en máquinas virtuales y soportes de almacenamiento que se alquila a un proveedor de servicios en la nube. El modelo que usan los desarrolladores se llama DevOps. La organización supervisa y gestiona la base de datos.

2. **Base de datos como servicio (DBaaS):** la base de datos se ejecuta en la nube y el proveedor se encarga de las tareas operativas, de mantenimiento, de gestión y de administración, y todo en tiempo real, con la mayor seguridad y alta disponibilidad y con procesos de automatización de software.

Esta nueva tecnología incluye una serie de ventajas que la hace muy potente:

1. Comercialización más rápida: debido a que el despliegue tecnológico local no es necesario, el producto en la nube entra en desarrollo en tiempos mucho más cortos.
2. Costes más bajos: debido al contrato por uso y la escalabilidad dinámica, el cliente paga en función de los momentos de alta o baja funcionalidad de su producto en la nube según la demanda. Incluso los servicios pueden ser desactivados en los momentos necesarios. Esto conlleva ahorros muy elevados.
3. Despliegue más rápido: la configuración en la nube es mucho más rápida, simplificando las pruebas y las nuevas implantaciones.
4. Riesgos más bajos: los errores humanos y los esquemas de alta seguridad son responsabilidad del proveedor del servicio, que automatiza, en la medida de lo posible, estos procesos.

Big data

La recolección masiva de datos a la que antes se ha hecho mención en la minería de datos forma parte de lo que se conoce como Big Data. El tamaño de estos almacenes de datos es tan grande que los procesos de análisis en busca de patrones ocultos requieren técnicas muy avanzadas en su análisis. El objetivo es más desafiante, la obtención de conocimiento en tiempo real, permitido por la propia expansión de internet, avances en las tecnologías de procesamiento de información y el almacenamiento en la nube.

Esta unión de almacenes de datos gigantes y algoritmos de análisis en tiempo real se denomina Big Data.

Los Big Data tienen las siguientes características, las cinco uves:

1. **Valor.** Aportar valor a la sociedad ayudando a impulsar el desarrollo, la innovación y la competitividad, pero sin olvidar la calidad de vida de las personas.
2. **Variedad.** Conjuguar y combinar cada tipo de información y su tratamiento específico para alcanzar un todo homogéneo.
3. **Velocidad.** El análisis de la información disponible debe ser en un tiempo lo más real posible, de lo contrario, su evolución estaría siempre por encima de su tratamiento, haciendo de su explotación tecnologías improductivas, pues se alejan de estados reales.
4. **Veracidad.** Los datos deben ser de calidad y estar siempre disponibles.
5. **Volumen.** La información debe ser lo más completa posible para que la toma de decisión sea óptima.

Un ejemplo de producto comercial para la gestión completa de almacenes de datos gigantes es **Apache Hadoop** y se ha convertido en uno de los mejores entornos de trabajo para programar aplicaciones distribuidas de acceso a tecnologías Big Data.

La inteligencia artificial es una parte de la ciencia de la computación que se encarga de crear mecanismos para dotar a las máquinas capacidades del ser humano, como el razonamiento, el aprendizaje y la creatividad. A través de los sensores, estos autómatas autoprogramables reciben información que digitaliza, los procesa y les sirve para tomar decisiones. La tecnología actual está influyendo en el avance de estos procesos, ya que requieren una arquitectura física y lógica más compleja. Los sistemas de inteligencia artificial software pueden ser los automóviles sin conductor, los asistentes virtuales, los motores de búsquedas, los sistemas de reconocimiento de voz, el software de análisis de imágenes con diferentes propósitos, etcétera.

Para que un sistema informático pueda considerarse con capacidad de inteligencia artificial, este debe predecir y realizar acciones basándose en los patrones de datos de los que dispone y pueda corregir los errores tomados con anterioridad. La minería del conocimiento permite crear y entrenar modelos de aprendizaje automático, agregar inteligencia a las aplicaciones y convertir la información no estructurada en datos donde se pueden hacer búsquedas.

PRÁCTICA

INSTALACIÓN DE ORACLE DATABASE EXPRESS EDITION

Pasos:

1 . Baja la última versión

Selecciona [la última versión de Oracle Express Edition](#) y procede a su instalación. Es recomendable usar una máquina virtual con al menos 2 GB de RAM e 11GB de disco duro libre. Fíjate que se puede instalar en un Oracle Linux (tendríamos que crear una máquina virtual con el) o en Windows. Vamos a instalarlo en una máquina virtual con Windows 10.

2. Para instalarlo seguiremos [este documento de Oracle](#).

Para la mayor parte, sigue las instrucciones del documento anterior. Salvo:

En lugar de doble-clic en el fichero setup.exe file, haz click con el botón derecho y selecciona **Run as Administrator**.

Acepta todo por defecto.

Cuando te salga el mensaje del bloqueo firewall de windows defender permite acceso.

3. Ten muy en cuenta el paso correspondiente a la introducción de la contraseña que usarán las cuentas de usuario SYS, SYSTEM y PDBADMIN. Esta contraseña sera la que se emplee para conectarse a la base de datos por primera vez a través de la cuenta SYS AS SYSDBA.

4. Ten paciencia, le lleva mucho tiempo acabar. Apunta en un documento la info que aparece: contenedor, conexión y url del Enterprise Manager.

Este producto crea una base de datos que ya está preparada para su explotación. Su nombre es XE. Dentro del contenedor principal, existe una base de datos conectable, llamada XEPDB1, con la que también se trabajará.

El Administrador de la base de datos. El DBA

En la instalación de Oracle Express Edition, el programa solicita la contraseña para las cuentas de usuarios SYS, SYSTEM y **PDBADMIN**. En la máquina donde se ha instalado el sistema gestor de bases de datos, están corriendo los servicios necesarios para dar respuesta a los clientes Oracle que quieran conectarse a la base de datos. Se requiere, entonces, de una aplicación que permita la conexión al motor del sistema gestor. Esto se hace a través de los procesos que están atentos a cualquier solicitud de un programa cliente. La conexión requiere una cuenta de usuario y su contraseña. Por ello, debe existir como minimo una cuenta de usuario que permita entrar en el servidor. Toda instalación Oracle cuenta con las cuentas SYS, SYSTEM y PDBADMIN con la contraseña que se ha dado en la instalación.

La cuenta **SYS** se usa para la gestión del diccionario de datos. La cuenta SYS tiene el espacio de tablas **SYSTEM** como espacio de tablas por defecto y contiene todas las tablas y vistas del diccionario de datos.

La instalación de Oracle Express Edition provee de un programa cliente que permite comunicarse con el servidor a través de una consola de comandos. Este programa se llama sqlplus. A continuación, se describe el uso de este programa ejecutable para hacer una conexión al servidor a través de la cuenta SYS con el rol **SYSDBA**. Cuando una cuenta de usuario tiene más de un rol y se quiere indicar cuál de ellos usar en el proceso de autenticación, se usa la palabra AS. De este modo, el usuario que se va a usar es SYS AS SYSDBA, es decir, el usuario SYS con el rol SYSDBA.

Primero, se debe abrir una consola de comandos. En Windows, se puede usar el comando cmd. Para ello, se puede pulsar la tecla Windows y sin soltarla se pulsa la tecla R. A continuación, se escribe cmd. Se abrirá una consola de comandos. Para cambiar el color de fondo y el de los caracteres, se puede usar el comando COLOR F0. En esta consola, se escribirá el nombre del programa ejecutable, es decir, **sqlplus**.

En este instante, solicitará el nombre del usuario con el texto Introduzca el nombre del usuario. Se escribe SYS AS SYSDBA y se pulsa la tecla Intro. A continuación, sqlplus solicitará la contraseña de este usuario, que será la que se introdujo en el proceso de instalación. Una vez que Oracle valida el usuario y su contraseña, el programa crea un entorno de comunicación con el servidor. El prompt del sistema operativo cambia a SQL> con el fin de indicar que los comandos que puede ejecutar ahora son los que el programa sqlplus permite. Para salir de una conexión, se usa DISCONNECT y, para una nueva conexión, se usa CONNECT. La palabra EXIT permite salir de la consola de comandos de sqlplus.

Crear una cuenta de usuario nueva con SQL*Plus

Para crear una cuenta de usuario, el administrador debe autenticarse usando, por ejemplo, SYS o SYSTEM. Una vez autenticado, se debe ejecutar una serie de comandos para la creación de un usuario en entorno Multitenant. La cuenta de usuario que se procede a crear es local para la base de datos conectable XEPDB1. Todas las bases de datos conectables cuelgan de un contenedor principal. En la instalación de Oracle Express Edition, la base de datos XE se encuentra en este contenedor principal y solo admite usuarios globales. Para crear un usuario local en una base de datos conectable o PDB, se debe indicar la base de datos XEPDB1. Para ello, se indica el contenedor donde trabajar con el comando

ALTER SESSION SET CONTAINER=XEPDB1;

El comando para crear un usuario es **CREATE USER** y se deben indicar algunos parámetros necesarios. A continuación, se le concederá unos privilegios que son necesarios para que el usuario pueda conectarse y trabajar en un entorno Multitenant.

A continuación, se procede a crear una cuenta de usuario llamada Tienda. Esta cuenta tendrá la contraseña tienda y su espacio de tablas por defecto será USERS. La secuencia de comandos para crear a este usuario es la siguiente:

CREATE USER TIENDA IDENTIFIED BY oracle
DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP;
GRANT CONNECT, RESOURCE TO TIENDA;

Se debe prestar especial atención al modo en el que se escribe la contraseña, ya que este valor sí es sensible a las mayúsculas.

Para conectarnos podemos escribir

sqlplus tienda/oracle@localhost:1521/XEPDB1

El SQLDeveloper

SQL Developer es un producto desarrollado por Oracle cuyo objetivo es ofrecer una plataforma única para la gestión de bases de datos Oracle. Las funcionalidades de este producto son muchas y posee herramientas muy útiles para la minería de datos y para las técnicas OLPT. Esta herramienta puede usarse desde el desarrollador de programas PL/SQL hasta el administrador, ya que, con su entorno simple y eficaz, permite ejecutar cualquier comando y explotar su interfaz gráfica para minimizar los procesos de creación de los objetos de una base de datos.

Toda gestión gráfica ofrece el comando sql necesario para posteriores ejecuciones y recreación. De este modo, se pueden construir los ficheros de comandos necesarios para posteriores reacciones. El entorno también admite código Java y XML. Además, cuenta con nuevos componentes que admiten trabajar con bases de datos no relacionales a través del lenguaje BSON y JSON.

SQL Developer cuenta con varias herramientas que permiten generar planes de ejecución, exportar los datos a otros formatos muy usados por otras arquitecturas como son Excel, XML, HTML, PDF, etc. El área de trabajo dispone de controles para depurar el código, probar y ejecutar los programas de bases de datos.

Además, SQL Developer cuenta con un panel para el administrador de bases de datos. Este panel tiene acceso a la herramienta de recuperación RMAN, a procesos de auditorías, a la gestión de roles y usuarios y a la gestión del almacenamiento.

PRÁCTICA

INSTALACIÓN DE ORACLE SQL Developer

SQL Developer es una herramienta como TOAD que permite conectarte a una base de datos y hacer tareas de administración o de desarrollo, tanto en SQL como en PL/SQL
Pasos:

1. Mira que versión tienes de JAVA. Si no tienes java SE Development Kit 8 (Java Version 1.8.x), 11, o 17 instalado (por ejemplo porque estamos usando una máquina virtual), si es así podemos elegir la versión con **JDK 8 incluido**

Para ver tu versión de java, abre cmd y teclea: `java -version`

Si tienes JDK 8, 11, o 17 instalado, elige el **instalador que requiere el JDK**.

2. Baja el [programa de esta página](#). Recuerda bajar el instalador con JDK o sin JDK incluido dependiendo de si tienes el JAVA instalado o no. Tendrás que tener un usuario en la web de Oracle. Si no estás dado/a de alta, date.

2. Sigue las [instrucciones de esta página](#). Te recomiendo que extraigas el sqldeveloper al directorio C:\Program Files\Oracle\ o C:

El SQL DataModeler

SQL Developer cuenta con una solución muy completa para el arquitecto de aplicaciones y modelador de datos. Esta solución se denomina Oracle SQL Data Modeler. Se puede ejecutar junto con SQL Developer o como herramienta separada. Esta herramienta permite construir el diseño lógico de una base de datos y obtener los modelos relacionales y físico, además del script DDL que construye todos los objetos de la base de datos. Es una herramienta muy potente para los diseñadores, ya que, con tan solo poseer el diagrama entidad-relación y su equivalente diagrama lógico, el resto de productos son generados automáticamente.

Entre otras opciones muy interesantes, cabe destacar la importación desde diccionarios de datos, scripts, repositorios de Oracle Designer y ERwin, el control de versiones de los diseños, comparación de modelos con generación de scripts ALTER, creación de informes, etcétera.

PRÁCTICA

INSTALACIÓN DE ORACLE SQL DataModeler

Conéctate a la página web de Oracle e instala el DataModeler en tu máquina virtual

Conexión a la base de datos con SQLDeveloper

Una vez abierta la aplicación SQL Developer, se debe crear una conexión a la base de datos o utilizar una ya creada. Para crear una nueva, se hace clic sobre la cruz verde (1) de la ventana principal, se obtiene otra para indicar una serie de parámetros de conexión. El proceso de acceso a un sistema gestor de bases de datos siempre usa una cuenta de usuario creada en él. Si aún no se dispone de cuentas de usuario, se debe usar SYS o SYSTEM con la contraseña que se indicó en la instalación de Oracle y proceder a crear una cuenta de usuario nueva. Supóngase que se está ejecutando SQL Developer en la misma máquina donde se tiene alojado el sistema gestor de bases de datos. Los parámetros que se deben definir son los siguientes:

Nombre de conexión: cada conexión creada debe tener un nombre único, por ejemplo, Mi primera conexión.

Usuario: cuenta de usuario que ya ha sido creada en el SGBD y que se quiere usar para la conexión. Si es la primera vez o no se dispone de cuentas de usuarios, se debe usar la cuenta SYS o SYSTEM.

Password: contraseña para dicha cuenta. La que se haya dado en la instalación.

Rol: se selecciona SYSDBA para la cuenta de usuario SYS. Para SYSTEM no se selecciona ningún rol.

Nombre del host: el nombre de host o dirección IP que tiene la máquina. En el caso de conectarse a la misma máquina, se escribe localhost. Si es otra máquina, se escribirá la dirección IP. También admite otros valores como el FQDN.

Puerto: número de puerto que usar, generalmente 1521.

SID: aquí se indica el nombre de la base de datos. Si se está usando la versión Express Edition, se escribirá XE. En caso contrario, se escribirá el nombre exacto de la base de datos.

Nombre del servicio: se cumplimentará para conexiones remotas o para conexiones contenedores de bases de datos conectables. Si se utiliza tecnología multitenant, tecnología que se usa desde la versión 12c, se debe escribir XEPDB1, si se está usando Express Edition. Si no, se escribirá el nombre de la base de datos conectable contenida en un contenedor.

Nueva / Seleccionar Conexión a Base de Datos

Nombre de Co... Detalles de Co... Nombre de Conexión

cxpostigo sys@//localhos... Usuario

Contraseña

☐ Guardar Contraseña ☐ Color de Conexión

Oracle

Tipo de Conexión Básico Rol valor por defecto

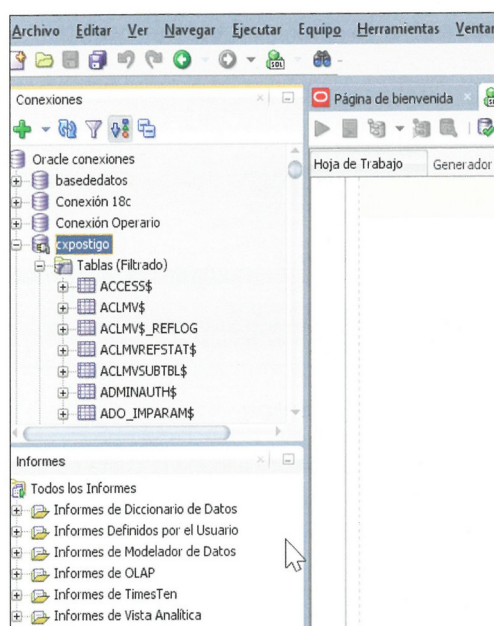
Nombre del Host localhost

Puerto 1521

☒ SID xe

☐ Nombre del Servicio

☐ Autenticación del Sistema Operativo ☐ Autenticación Kerberos Avanzadas...



Una vez que la conexión con la base de datos ha tenido éxito, el nombre de la conexión se convierte en una opción maximizable que se representa con un signo + a la izquierda del nombre.

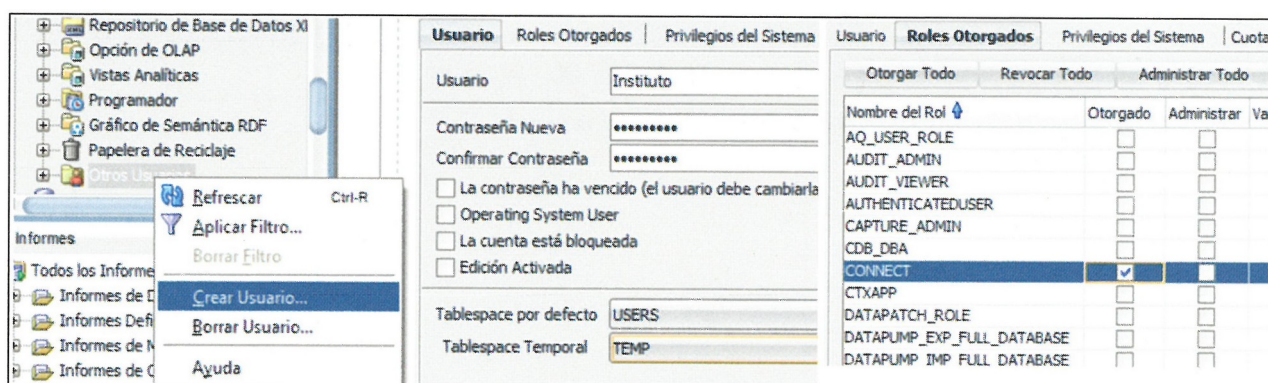
Pulsando en el, se obtienen todos los objetos de la base de datos clasificados por su tipo. Si el usuario se ha autenticado como SYS o como SYSTEM, obtendrá todos los objetos del diccionario de datos.

Crear una cuenta de usuario con el SQLDeveloper

El esquema de la base de datos será el conjunto de tablas de las que el usuario dispone, ya que este es su propietario.

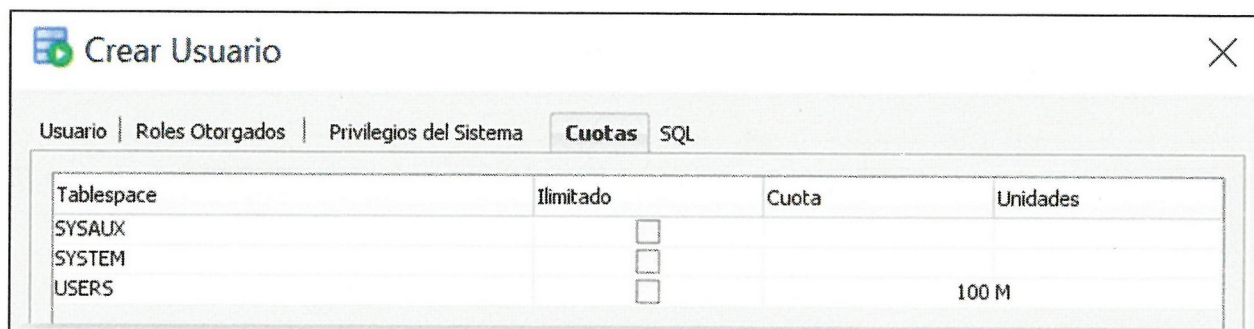
Una vez autenticado como SYS, en el menú contextual de Otros usuarios, que se obtiene haciendo clic con el botón derecho del ratón sobre el texto Otros usuarios, existe la opción Crear Usuario. Se obtiene una ventana cuyas pestañas son Usuario, Roles Otorgados, Privilegios del Sistema, Cuotas y SQL.

En la pestaña Usuario, se debe indicar el nombre del usuario, por ejemplo tu nombre; la contraseña de dicha cuenta; el espacio de tabla por defecto de este usuario, USERS, y su espacio temporal, TEMP. El espacio de tabla es un elemento lógico de Oracle para la gestión del almacenamiento. Cuando una tabla se crea, se almacena en el espacio necesario en un fichero de datos de un espacio de tablas. De este modo, el administrador no tiene que indicar en que fichero físico se almacena, sino que lo hace a través de su unidad de gestión lógica denominada espacio de tabla o tablespace. Cuando se crea un usuario nuevo, es conveniente indicar cuál es su espacio de tabla por defecto. De este modo, cuando crea una tabla y no indica el espacio de tablas que usar, lo crea dentro del denominado espacio de tablas por defecto o default tablespace.



En la pestaña Roles Otorgados se selecciona CONNECT, RESOURCE

Por último se debe indicar para este usuario la cuota de uso del espacio de tablas USERS. En la pestaña Cuotas, se indica el tamaño con un número y, luego, la unidad, K, M, G o T o podemos darle cuota UNLIMITED en el tablespace USERS para que pueda crear objetos.



En la pestaña SQL Se muestra el comando que el motor interpreta para poder llevar a cabo la creación del usuario. Si te fijas es igual que el código que usamos antes para crear nuestro usuario.

El diccionario de datos de Oracle

El diccionario de datos está compuesto por un conjunto de tablas y de vistas que registran datos de cada uno de los objetos que se crean en una base de datos. Así, cuando se crea una tabla, la información de esta, número de columnas, nombre de la tabla, número de restricciones, propietario de las tablas, etc., se registra en una tabla con nombre DBA_TABLES.

Si se crea un usuario, la información de este se almacena en una tabla del diccionario de datos con nombre DBA_USERS. Para los índices, se hace en DBA_INDEXES; para los espacios de tablas, se hace en DBA_TABLESPACES; para los ficheros de datos, se hace en DBA_DATA_FILES; para las restricciones, se hace en DBA_CONSTRAINTS, y así sucesivamente para cada tipo de objeto que se puede crear en la base de datos.

El conocimiento del nombre de cada tabla que compone el diccionario de datos provee a los administradores de una capacidad completa de obtener cualquier tipo de información de esta. El usuario SYS puede acceder a la totalidad de todos los objetos que crea cualquier usuario. Un usuario que no sea SYS solo verá los objetos que él haya creado o que otros usuarios permitan que vea. De este modo, en vez de usarla palabra DBA_ que precede a las tablas que antes se han mencionado, se usa USER_ cuando un usuario no SYS quiere acceder a la información de sus objetos y ALL_ cuando quiere acceder a los suyos y a los objetos que los otros usuarios permiten que vea. De este modo, existen para las tablas las vistas DBA_TABLES para el administrador, USER_TABLES para el usuario que ejecuta el comando y ALL_TABLES para los objetos suyos y de los demás que pueda ver.

La siguiente consulta solo mostraría las tablas que haya creado el usuario *rodeira*:

```
SQL>conn rodeira/lapassword@localhost:1521/XEPDB1;  
SQL>select * from user_tables;
```

PRÁCTICA

INSTALACIÓN DE ORACLE SQL Developer y DataModeler

Baja la última versión de SQLDeveloper y DataModeler a tu ordenador del aula.
Conéctate al Servidor Oracle del Instituto sabiendo que la cadena de conexión de SQL*Plus sería
tunombre/oracle@ies.iesrodeira.com:64000:1521/orclpdb.iesrodeira.com