

Resumen Cap 2 Mullins

Creando el ambiente de BD:

Una de las tareas principales de un DBA es escoger e instalar una DBMS pero ese no es solo el único trabajo que tienen (missconception), realmente no es lo más difícil, es peor el enviroment.

Definiendo la estrategia de la organización del DBMS:

Choosing a suitable DBMS for enterprise database management is not as difficult as it used to be.

Las compañías grandes tienden a tener varios productos de DBMS (combinados Oracle,MySQL,Windows...) puede ser por esto: a veces se instalan DBMS sin tan siquiera saber si la app pudo haber sido exitosa usando un DBMS que ya existía en la compañía, también que la compañía compró una app que no servía con ninguna de las DBMS que ya existían, solo por seguir modas.

Es difícil borrar un DBMS porque entre las DBMSs son incompatibles y tener que convertir el código de la app, también que cuando se instala un nuevo DBMS, las apps viejas y las BDS no suelen poder migrarse a él, el DBMS debe seguir manteniéndose y dándole soporte.

Básicamente el **único** responsable de tomar las decisiones de cual DBMS comprar debe ser el grupo de DBA.

Escogiendo un DBMS:

Se debe buscar que haya la menor cantidad de DBMS difs

- 1) Obtener un producto de un vendedor de nivel 1, porque por algo tienen la mayor cantidad de productos en el mercado, por ejemplo DB2, Oracle, Microsoft SQL Server, escoger una dif de estas tres debe ser solo bajo circunstancias específicas.
- 2) Después de esas tres viene MySQL,Sybase,Teradata e Informix, son de nivel, tienen buena calidad pero su base instalada es menor, casi siempre las consigues las compañías con menos recursos.
- 3) Open Source: PostgreSQL,EnterpriseDB o MySQL
- 4) Si se busca uno orientado a objetos para un proyecto en específico, se puede considerar ObjectDesign o Versant.
- 5) Los NoSQL pueden ser Hadoop,Cassandra y MongoDB.

Choosing any of the lower-tier candidates involves incurring additional risk. Cuando se quiera escoger un DBMS deben tomarse en cuenta estos factores:

Soporte del SO: Does the DBMS support the operating systems in use at your organization, including the versions that you are currently using and plan on using?

Tipo de Organización: Importante saber la filosofía de la compañía, hay unas que son conservadoras entonces les gustan los ambientes tradicionales de mainframe como las de operaciones gubernamentales, financieras y de insurance and health, las más liberales están más abiertas a cambios, finalmente hay compañías que simplemente no confían en Windows entonces prefieren todo lo que tengan que ver con UNIX.

Benchmarks (punto de referencia para medida, comparación y evaluación) típicamente es un point-in time snapshot measurement (como una captura en un momento determinado) para comparación con otros benchmarks). The Transaction Processing Performance Council (TPC) publica el performance oficial de las BDs, eso puede servir de guía pero no debe ser lo único determinante además estas están en constante update y por eso puede que una tenía un puntaje alto y pasó el tiempo y ya no.

Escalabilidad: la habilidad de incrementar el tamaño o la capacidad efectiva con menor impacto económico o cualquier otro, Lo que se debe preguntar es, este soporta la cantidad de usuarios y el tamaño de las BDs, ¿Es muy difícil mantenerlos o construirlos, si son BDs muy grandes? ¿Hay usuarios independientes que puedan confirmar la scalability que dice que tiene el vendedor?

Disponibilidad de supporting de herramientas de software: si están disponibles las herramientas que uno necesita para ese DBMS todas las del cap 23 del libro.

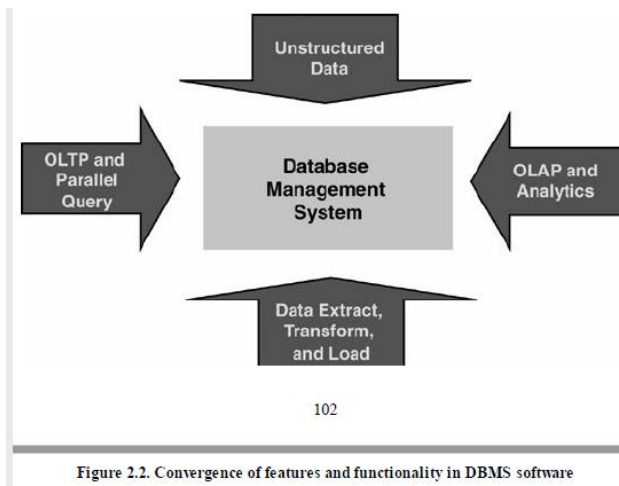
La TPC: organización sin fines de lucro que manejan y administran los tests de benchmarks de las BDs mide el performance en términos de transacciones que se puedan hacer por un tiempo, estas se refieren a tareas como updates para BDs como control de inventario (bienes), reservaciones de aerolíneas (servicios) y bancos (dinero). Define tres benchmarks: TPC-C, para el workload de trabajo de producción planificado en un entorno de transacción, TPC-H, funciona para la toma de decisiones, TPC-E: un workload actualizado de OLTP, basado transacciones financieras

Técnicos: ¿Hay suficientes profesionales expertos para el DBMS?

Costo de propiedad/ownership: Analizar bien los precios en el mercado.

Release Schedule: ¿Cada cuanto el vendedor de DBMS release una nueva versión? Depende hay empresas que les sirve y las que son más conservadoras no tanto.

Reference customers: hablar con customers que utilicen las DBMS para tener referencias de cosas generales como, el support si el vendedor responde bien a los problemas, si hay muchos bugs, como es la calidad de las nuevas releases, etc.



Arquitecturas de DBMS:

The supporting architecture for the DBMS environment is very critical to the success of the database applications.

Hay cuatro niveles:

Enterprise DBMS: está diseñado para la escalabilidad y high performance, es capaz de soportar BDs grandes, muchos usuarios concurrentes y diferentes tipos de apps, brinda además support para multiprocesadores, para queries paralelas etc.

Departmental DBMS: también se le llaman Workgroup DBMS están como en el middle ground, estas dan soporte small to medium-size Workgroups el hacerle upgrades al hardware y al software puede que hagan que estas trabajen de la misma forma que las enterprise, es más barato.

Personal DBMS: para un solo usuario, low to medium-powered PC, versiones Lite, aunque son de bajo costo pero deben ser solo para proyectos de baja escala no debería implementarse en una app multiusuario.

Mobile DBMS: diseñado como una versión especializada de departamental o Enterprise, para usuarios remotos que no suelen estar conectados a la red, puede ser para BDS modificables en celulares o laptops.

DBMS Clustering: es el uso de múltiples sistemas de computación independientes trabajando juntos como uno solo.

Hay dos arquitecturas predominantes para el clustering:

Shared-Nothing: c/d sistema tiene su propio recurso privado como memoria, discos. Los procesadores que están en el cluster se comunican pasándose mensajes mediante una red que interconecta las compus, solo uno de los sistemas pueden tener y acceder a un recurso en particular a la vez. La mayor ventaja es la escalabilidad, porque pueden aumentar los procesadores, no se interrumpen.

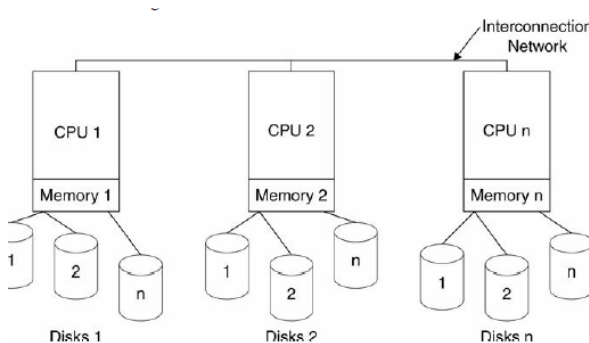


Figure 2.3. Shared-nothing architecture

Shared-Disk: todos los sistemas conectados comparten el mismo device de disco, c/d procesador tiene su memoria privada pero todos los procesadores pueden abordar todos los discos, no son buenos para PCS de baja escala, mejor para empresas grandes con Mainframes (grandes procesadores). *Great benefits can be obtained with only a few clustered mainframes, while many PC and midrange processors would need to be clustered to achieve similar benefits.*

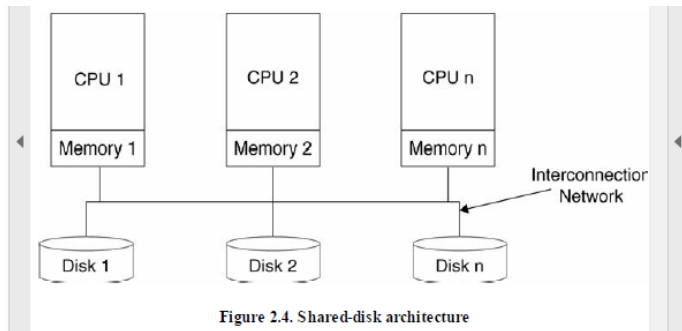


Figure 2.4. Shared-disk architecture

Table 2.3. Comparison of Shared-Disk and Shared-Nothing Architectures

106

Shared-Disk	Shared-Nothing
Quick adaptability to changing workloads	Can exploit simpler, cheaper hardware
High availability	Almost unlimited scalability
Performs best in a heavy read environment	Works well in a high-volume, read-write environment
Data need not be partitioned	Data is partitioned across the cluster

DBMS Proliferation: La proliferación de diferentes productos de DBMS son difíciles de dar soporte, aunque como DBA se esté siendo bombardeado con marketing es mejor solamente acceder si se tienen razones de peso para obtener alguno y aún así se debe hacer un check dos veces.

Hardware Issues / Problemas: tanto el hardware como el SO en que correrá el DBMS afectan: Escalabilidad, fiabilidad, y disponibilidad.

Cloud Database Systems: mueve la data y su administración del ambiente local computacional y brinda los servicios de DBMS por Internet, representa menos costos en software y hardware, sin embargo se debe confiar plenamente en el proveedor del sistema de cloud, Ej: Microsoft Azure
At a high level, cloud computing is the delivery of computing as a service. The term *cloud* is used as a metaphor for the Internet.

Instalando el DBMS: A DBMS is a complex piece of software that

requires up-front planning for installation to be successful.

Lo que primero se tiene que tomar en cuenta son los prerequisites, por ejemplo en Hardware (suficiente memoria) y Software (versión del S.O.)

Después de eso, se debe leer la guía de instalación y seguir sus instrucciones explícitas, además de trabajar en conjunto con el vendedor durante este proceso.

Requerimientos de Hardware:

Cada DBMS tiene requerimientos básicos de CPU como la versión y la velocidad mínima de los procesadores, adicionalmente algunos DBMS tienen modelos de hardware que son requeridos o no son soportados. Furthermore, each DBMS offers different “flavors” of its **software** for specific needs.

Requerimientos de Storage/Almacenamiento:

Los DBMS usarán almacenamiento de disco para guardar los índices de la BD, también los siguientes ítems:

- El catálogo del Sistema o el diccionario de data es usado por el DBMS para administrar y track BDS e información relacionada a las mismas, entre más objetos de DBS se vayan a crear, más será la cantidad de storage requerido por el catálogo.
- Cualquier otra base de datos del sistema requerida por el DBMS, por ejemplo, para soportar conexiones distribuidas o herramientas de gestión.
- Archivos de registro que almacenan todos los cambios realizados en cada base de datos, como registros activos, registros de archivo, segmentos de reversión y cualquier otro tipo de registro de cambios requerido por el DBMS.
- Archivos de arranque o de control, se accesan a ellos cuando el DBMS se inicializa.
- Archivos de trabajo que se utilizan para ordenar la data o cualquier otro requerimiento de procesamiento.
- DBS por default para estructuras del sistema o para nuevos objetos de BDS mientras sean creados
- Estructuras temporales de DBS para data que no es necesario que sea persistente pero necesita tener un espacio almacenado durante las operaciones, como reconstruir índices de clusters.
- Archivos de errores de procesamiento y desechos del sistema.
- BDS de DBA usadas para administrar monitorear y tuning, como para tests de nuevos lanzamientos, migración de scripts, etc.

Es importante que si se van a guardar muchas BDS, tener varios dispositivos de almacenamiento aún si estos no se llenan, entre mejor esté organizado, el trabajo de los DBMS serán mas eficientes y no serán perjudicadas por el disco físico cuando se accede a la data. También que es importante para los backups y los log off-loading.

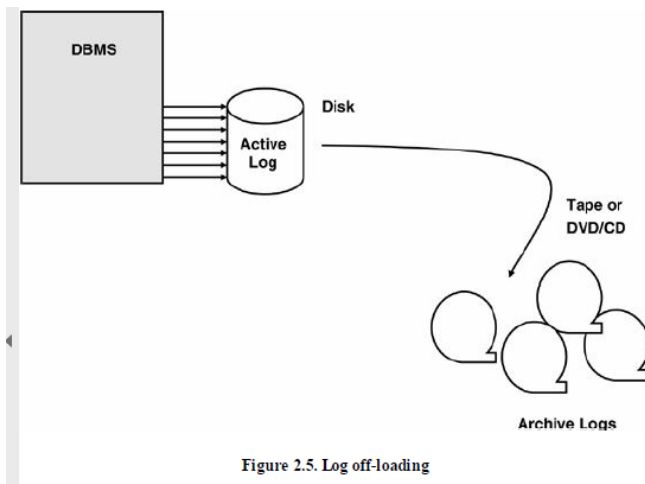


Figure 2.5. Log off-loading

Requerimientos de memoria:

Un DBMS tanto como sus DBS necesitan la memoria para funcionalidades básicas y además la usan para procesos internos como mantener el área global del sistema y muchas tareas más, se necesita una cantidad significativa de memoria para data de caché para evitar I/O, leer data de un disco es más caro y lento que mover la data a la memoria, la imagen muestra como el DBMS usa una estructura de memoria llamada buffer pool (An area of memory set aside and used to avoid I/O operations when actual data is being read from the database. Also referred to as a *data cache*.) para reducir las requests físicas de I/O, en general, entre más grande sea el buffer pool, más tiempo la data se queda en la memoria y el procesamiento de las BDS es mayor.

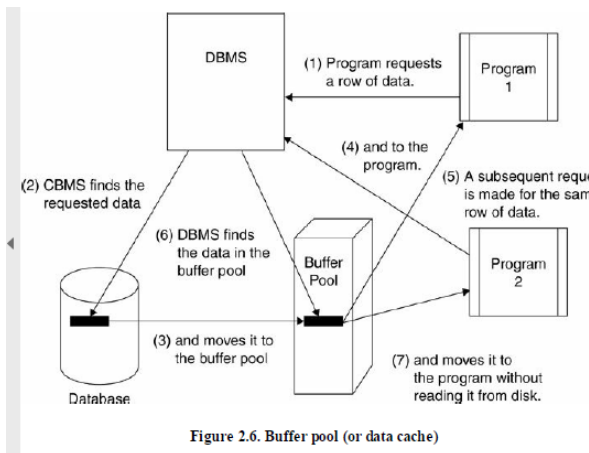


Figure 2.6. Buffer pool (or data cache)

Los caché del programa almacena cosas como sentencias SQL "compiladas", autorizaciones de bases de datos y bloques de estructura de base de datos que son utilizados por los programas a medida que se ejecutan.

El DBMS generalmente requiere memoria para admitir otras características como

el bloqueo de manejo solicitudes, facilitar solicitudes de datos distribuidos, ordenando data, optimizando procesos y procesamiento de SQL.

Configurando el DBMS:

La forma en que se configuren los parámetros del sistema controlan cuáles funciones del DBMS y recursos estarán disponibles, durante la instalación, las entradas que se escojan en el script de instalación se usarán para establecer los settings iniciales de los parámetros del sistema.

También cada sistema brinda un método para cambiar estos parámetros una vez que el DBMS esté operando, como cambiarlos por medio de comandos del DBMS, editar un archivo que los contenga, pero se debe ser cuidadoso si es de esta forma. ¿Qué controlan los parámetros del sistema? La autorización del DBA al DBMS, el número de BDS loggeadas, la cantidad de memoria usada para la data y el caché, además enciende o apaga features del DBMS. Se debe ser cuidadoso cuando se usan valores por default, ya que aunque sean más rápidos para la instalación pueden causar problemas subsecuentes como de performance, integridad de la data o fallas del DBMS total.

Conectando el DBMS a Infraestructura del Sistema De Soporte (Supporting System Infrastructure):

Parte de instalar el DBMS también es conectar el DBMS con otros componentes de software que interactuaran con él, ejemplo de estas infraestructuras son: redes, monitores de procesamiento de transacciones, listas de mensajes, otros tipos de middleware, lenguajes de programación, software de management, operation y trabajo de control de software, servidores web y servidores de apps.

Verificación de la Instalación:

Después de instalarlo, se deben hacer tests que comprueben que está bien instalado y configurado, hay vendedores que ya incliyen programas para esto o se puede hacer testeando interfaces estándar que son interactivables de SQL para hacer statements al DBMS como código pequeño para consultas simples como SELECT, INSERT, etc.

Ambientes de DBMS: el DBA debe crear varios environments para el support como, testing, garantía de calidad, integración y trabajo de producción, el hecho de que el DBMS pueda aguantar varios ambientes para una sola BD minimiza problemas de migración y que no se necesitarán nombres compljos para las convenciones de support.

Versiones mejoradas del DBMS y lanzamientos: un ciclo común de releases de software de DBMS va e 18 a 24 meses, con constantes arreglos de bugs y updates de mantenimiento entre cada release.

Diferencia entre versión upgrades y releases:

Una upgrade de versión puede verse como un caso especial de una nueva instalación, todos los procedimientos requeridos en una instalación se aplican en un upgrade, debe ser planeada para causar la menos cantidad de problemas con los

usuarios existentes antes de ella, así como que las herramientas, apps y demás no se vean afectadas.

Una reléase implica menos cambios y no tantas features, usualmente las funcionalidades significativas son añadidas en las upgrades y no en los releases.

Beneficios de pasarse a un nuevo lanzamiento:

- Los desarrolladores pueden aprovechar las nuevas características y funcionalidades entregadas solo en el nuevo lanzamiento.
- Para las aplicaciones compradas, el proveedor de la aplicación puede requerir un DBMS específico sea versión o lanzamiento.
- Generalmente ofrecen características mejoradas de rendimiento y disponibilidad que puede optimizar las aplicaciones existentes.
- Los proveedores de DBMS suelen brindar un mejor soporte y responden a los problemas más rápido para un nuevo lanzamiento de su software.
- Los ahorros de costos pueden acumularse.
- Alineará la prueba y la producción de entornos de bases de datos, esto porque si se ejecuta una nueva versión en el entorno de prueba durante demasiado tiempo, las tareas de administración de bases de datos y desarrollo de aplicaciones se vuelven más difíciles porque las BDS de prueba funcionarán de manera diferente a las de producción.

Riesgos:

- Implica cierto nivel de interrupción a las operaciones del negocio, como mínimo, las BDS no estarán disponibles mientras se esté utilizando el DBMS actualizado, eso puede causar tiempo de inactividad y oportunidades comerciales perdidas si se hace durante el horario comercial normal (o si no hay un downtime planificado).
- Tener que convertir las estructuras de las DBS o darse cuenta que features anteriores de soporte fueron eliminadas en el nuevo lanzamiento también retrasos en implementación de apps.
- El costo de hacer un upgrade puede ser una barrera significativa para una migración de DBMS, porque debe estar dentro del Budget (son más caras y requieren comprar hardware distinto), también en costos de instalación, testing y deploying no solo para el DBMS sino también para las apps que lo usan.
- Los proveedores tienden a asegurar que se van a lograr mejoras en el performance con una nueva versión, pero las técnicas de optimización de sql cambian, esta versión provocará que las rutas de acceso de SQL sean peor que antes, cuando el performance sufre, el código de las apps se tendrá que cambiar.

- Puede causar que características y sintaxis que se están utilizando en aplicaciones se vuelvan obsoletas.
- El DBA tendrá que hacer cambios invasivos, como incrementar el tamaño máximo de un objeto en una BD, o sea que deberá borrarlo y volver a crearlo con uno nuevo.
- Los productos de software de soporte pueden carecer de soporte inmediato para una nueva versión de DBMS.

Se debe contar con una estrategia apropiada para los upgrades de los DBMS, se debe tomar en cuenta si serán mayores la cantidad de beneficios con respecto a los riesgos, si hay más riesgos, es posible que las empresas esperen un tiempo antes de hacer upgrades, como saltarse versiones, a esto se le **llama Multiple-release upgrade**, las siguientes secciones mostrarán los problemas que deben tomarse en cuenta para una estrategia de estas:

Features and Complexity: quizás lo más importante para saber cuando y como hacer un upgrade del DBMS es la funcionalidad que soporta el lanzamiento. Si la funcionalidad del DBMS puede minimizar el costo y esfuerzo para los desarrolladores es momento de migrar a uno nuevo, el grupo de DBA debe asegurarse que los estándares actuales están modificados para nuevas features, educar a los desarrolladores y usuarios para cómo estas funcionan y cómo deben ser usadas así como preparar la infraestructura que soporte la nueva funcionalidad.

Complejidad del ambiente o entorno del DBMS: entre más complejo sea el actual, más difícil será hacer un upgrade, el primer problema es el tamaño del entorno, entre más BDS. Instancias, apps y usuarios más complejidad tiene, también es menos complejo si solo apps simples u orientadas a parches están involucradas, la locación en que se encuentran los servidores de BDS también afectan un posible upgrade, si hay servidores esparcidos en diferentes locaciones, diferentes zonas y horarios y para diferentes aplicaciones es difícil, también no se debería soportar diferentes versiones en producción.

Complexity issues include the following:

- Uso de los stored procedures y UDFS (user-defined functions)
- Complejidad del SQL, entre más tablas complejas, es más difícil asegurarse que los cambios de la ruta de acceso no impacten en el performance.
- Procesamiento Cliente/Servidor: el uso de la red y de múltiples nodos.
- Apps que estén diseñadas, codificadas y generadas por un framework o un IDE pueden tener componentes adicionales que se necesitan testear.
- Integración con otras infraestructuras de software como colas de mensajes, transacción de procedimientos pueden complicarlo.
- El lenguaje usado por los programas por que se debe hacer el support para las versiones compiladas, cambios en los APIs, etc.

Reputación del Proveedor del DBMS: se debe tomar en cuenta para soporte técnico, cómo son para arreglar bugs y para responder problemas, siendo importante las referencias de los customers, entre mejor tenga, es más probable que la organización se adapte.

Políticas de soporte del DBMS: no se debe estar usando un lanzamiento de DBMS en producción que ya no cuente con soporte del proveedor.

Estilo de la organización: los analíticos de industria de Gartner Inc han hecho un ranking de las organizaciones: **Tipo A** son las que la empresa está enfocada en tecnología y es mas probable que se arrisgen a nuevas releases sin ser probadas para ganar una ventaja competitiva, **las B:** menos probable que tomen riesgos pero sí adoptaran nuevas tecnologías en el momento que se hayan desecho de los bugs, y **la C** son muy consientes del costo y del riesgo, se quedarán de últimos en migrar a una nueva tecnología.

DBA Staff Skill Set: es más fácil si cuentan con altas skills y con mucha experiencia. The risk of an upgrade increases as the skills of the DBA staff decrease.

Soporte de plataformas: casi siempre los proveedores brindan soporte a las plataformas y SO que tienen más customers con licencia, por eso se deben tomar en cuenta estos factores.

Software de Soporte: estos incluyen aplicaciones compradas, herramientas de DBA y de reporte y análisis además para colas. Diferencia entre supporting y exploiting un lanzamiento por parte de los proveedores:
Los Software que dan soporte a un lanzamiento continuara funcionando igual que antes pero sin nuevas capacidades, por eso si una herramienta de DBA sporta por ejp una nueva versopn de Oracle, da todos los servicios que daba para la última versión mientras ninguna de las nuevas features se usen. Por el contrario, una herramienta que explote un lanzamiento proveerá la funcionalidad que se requiere para operar en nuevas features.

Fallback Planning o planificación de reserva: todas las versiones deberían venir con un manual que explique las nuevas features y que describa los procedimientos de reserva para poder volver a una versión anterior.

Verificación de Migración: el DBA debe implementar procedures que verifiquen que el upgrade del lanzamiento sea satisfactorio.

Estándares y Procedures de la BD: deben estar dentro de un documento en la ubicación central de la empresa. Los estándares son practicas comunes que aseguran las consistencia y efectividad del ambiente o entorno de la BD por ejemplo convenciones de nomenclatura. Los procedimientos son instrucciones definidos paso por paso para dirigir los procesos que requieren manejar eventos específicos como por ejemplo un plan de recuperación en caso de desastre.

Convenciones de nomenclatura de las BDS: sin estas será difícil identificar los objetos de las BDS correctamente y poder realizar tareas de administración, estas convenciones deben ser desarrolladas en conjunto con los estándares de nomenclatura del IT, también con el departamento de administración de la data si existe, importantes para minimizar cambios de nombres en los entornos. En general no se deben hacer restricciones innecesarias para los nombres de los objetos accedidos por los usuarios, ya que las BDS relacionales deben ser amigables, deben ser lo más descriptivos posibles y se debe evitar nombrarlos de formas que solo sean para hacerlos más pequeños, además recordar que deben ser consistentes en toda la BD. Se deben hacer abreviaciones SOLO cuando el texto es muy largo por el contrario no es necesario.

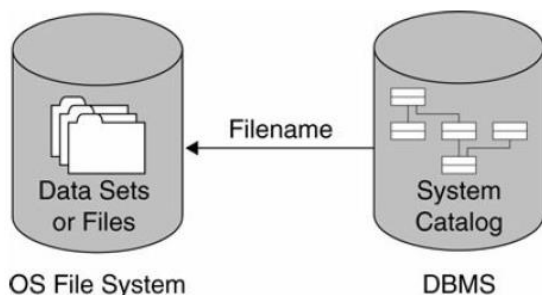


Figure 2.7. Database objects map to filenames

Otros estándares y procedimientos:

Independientemente de si se compran de un proveedor, se escriben desde cero o se adoptan de un grupo de usuarios o comité, cada una de las siguientes áreas debe estar cubierta:

Roles y Responsabilidad: Una matriz de gestión y administración de BDS debe desarrollarse para que documente cada tarea de soporte y quién lo hace

Table 2.4. Roles y responsabilidades de soporte de bases de datos

Task	DBA	DA	SA	Management	Operations	Applications	End Users
DBMS budget	X		X	P		X	X
DBMS installation	P		X		X	X	X
DBMS upgrade	P		X	X	X	X	X
Database usage policy	P	X		X			
Capacity planning	X		P	X	X	X	
Data modeling and analysis	X	P					X
Metadata policy	X	P		X			X
Governance and compliance	X	X	X	X			P
Database design	P	X				X	
Database creation	P						
System performance	X		P				
Database performance	P		X			X	
Application performance	X		X			P	
Backup and recovery	P		X		X	X	
Disaster recovery	P		X		X		
Database security	P		X		X		
Stored procedures	X					P	
Triggers	P					X	
User-defined functions	X					P	
Application design	X					P	
Application turnover	X				X	P	X
Application design reviews	X	X	X	X	X	P	X

Por supuesto, puede crear las tareas que considere necesarias en la matriz de roles y de responsabilidades, pero asegúrese de mantenerla precisa y actualizada con nuevas funciones y tareas de DBMS

Estándares de comunicación

Se puede optar por desarrollar estándares específicos para la comunicación entre grupos y personal específico.

El desarrollo de estándares de comunicación sólidos puede simplificar el trabajo de un DBA durante el tiempo inevitable de inactividad que se produce debido a errores de sistema, aplicación o incluso de hardware

Adoptar un estándar por el cual el DBA se comunica solo con el gerente durante resolución de problemas y remediación de emergencia. El gerente puede comunicar el estado hacia afuera mientras que el DBA se enfoca exclusivamente en solución de problemas.

Estándares de administración de datos

Si existe un grupo DA dentro de su organización, deben desarrollar una administración básica de datos guía de estándares para delinear el alcance de su rol laboral. Si no existe asegúrese de incluir estándares DA en los estándares DBA según corresponda. Deben incluir

- Una declaración clara de la política general de la organización con respecto a los datos, incluyendo su importancia para la empresa

- Directrices para establecer la propiedad y la administración de los datos.
- Reglas para la creación de datos, propiedad de datos y administración de datos.
- Política de gestión de metadatos.
- Directrices de modelado de datos conceptuales y lógicos.
- Los objetivos de la organización con respecto a la creación de un modelo de datos empresariales.
- Responsabilidad de crear y mantener modelos de datos lógicos.
- Pautas para el uso de herramientas e instrucciones sobre cómo se deben crear, almacenar modelos de datos
- Políticas de intercambio de datos organizacionales.
- Instrucciones sobre cómo documentar cuando las bases de datos físicas se desvían del modelo lógico de datos.
- Directrices sobre comunicación entre la administración de datos y la administración de bases de datos. para garantizar la creación y el uso efectivo de la base de datos

Estándares de administración de bases de datos

Se debe establecer un conjunto básico de estándares de administración de bases de datos para garantizar el éxito continuo de la función DBA. **Los estándares sirven como una guía para los servicios DBA ofrecidos y para enfoques específicos para apoyar el entorno de la base de datos.** Además, los estándares se deben de compartirse con los programadores ya que cuanto más entiendan los programadores de aplicaciones los matices del DBMS y el papel del DBA, mejor será la relación de trabajo entre DBA y el desarrollo

Estándares de administración del sistema: Se necesitan por las mismas razones que los estándares DBA. Estándares para SA pueden incluir:

- Instalación de DBMS y procedimientos de prueba
- Actualizar políticas y procedimientos
- Corrección de errores y prácticas de mantenimiento.
- Una lista de verificación de departamentos para notificar cambios inminentes
- Consideraciones de interfaz
- Almacenamiento, uso y procedimientos de monitoreo de DBMS

Normas de desarrollo de aplicaciones de bases de datos: El DBA debe documentar las consideraciones especiales de desarrollo necesarias al escribir programas que acceden a bases de datos. Debe incluir:

- Una descripción de cómo el acceso a la base de datos difiere del acceso a archivos planos
- estándares de codificación SQL
- Consejos y técnicas de rendimiento de SQL
- Procedimientos de preparación del programa y orientación sobre cómo incrustar SQL en un programa de aplicación
- Interpretaciones de SQLSTATE y códigos de error.
- Referencias a otros materiales de programación útiles para monitores de teleprocesamiento, lenguajes de programación y estándares generales de desarrollo de aplicaciones

Estándares de seguridad de bases de datos: A menudo las aplica y administra el DBA. Algunas tiendas de la unidad de seguridad de datos corporativos manejan la seguridad DBMS. Contienen la siguiente información:

- Detalles sobre qué autoridad otorgar para tipos específicos de situaciones;
- Documentación específica de cualquier procedimiento especial o documentación requerida para solicitudes relacionadas con el gobierno y el cumplimiento
- Una lista definitiva de quién puede aprobar qué tipos de solicitudes de autorización de bases de datos
- Información sobre las interfaces que se utilizan para conectar la seguridad DBMS con el funcionamiento productos de seguridad del sistema
- Políticas sobre el uso de la cláusula WITH GRANT OPTION de la instrucción SQL GRANT y cómo se deben manejar los REVOKE en cascada
- Procedimientos para notificar al solicitante que se le ha otorgado seguridad a la base de datos.
- Procedimientos para eliminar la seguridad de los empleados retirados, reubicados y despedidos.

Procedimientos de migración de aplicaciones y rotación

El número mínimo de entornos para admitir aplicaciones de bases de datos son dos: prueba y producción, algunas organizaciones usan múltiples entornos para apoyar fases del ciclo de vida del desarrollo, incluidas:

- Pruebas unitarias: para desarrollar y probar programas individuales

- Pruebas de integración: para probar cómo interoperan los programas individuales
- Pruebas de aceptación del usuario: para pruebas de usuario final antes del estado de producción
- Garantía de calidad: para eliminar los errores del programa
- Educación: para capacitar a los usuarios finales sobre cómo trabajar el sistema de aplicación

Cuando existen múltiples entornos, se requieren procedimientos para migrar objetos de base de datos y programas de medio ambiente a medio ambiente. Se necesitan pautas específicas para lograr migración de una manera propicia para el uso de cada entorno. Se debe documentarse antes de la migración.

Pautas de revisión de diseño: importante para garantizar un diseño, construcción, aplicación adecuados y rendimiento.

Estándares de soporte operativo: se define como la parte de la organización de TI que supervisa la base de datos entorno y asegura que las aplicaciones se ejecuten de acuerdo con la programación

Educación DBMS: Las organizaciones que usan DBMS deben dar clases de educación técnica continua para DBA, programadores y administradores de sistemas. Como mínimo deben dar:

- Descripción general de DBMS
- Modelado de datos y diseño de bases de datos
- Administración de bases de datos: clases técnicas en profundidad para DBA, SA y sistemas programadores
- Introducción a SQL
- SQL avanzado
- Programación de bases de datos.