

Nombre: Soriano Bonilla Carlos Ivan  
Tarea: 1  
Materia: Base de Datos  
Profesor: Ing. Arreola Franco Fernando  
Semestre: 2021-2

# 1. Privilegios que se le puede otorgar al usuario/roles

## Roles de Base de Datos

Un rol es una colección de privilegios del sistema y de objetos que se otorgan a usuarios y a otras tareas. Los roles están establecidos por niveles en la base de datos que se aplican en lo que respecta a su ámbito de permisos.

Los usuarios más comunes son:

### Tipos de Roles

Nombre de rol de nivel de base de datos	Descripción
db_owner	Los miembros del rol fijo de base de datos <b>db_owner</b> pueden realizar todas las actividades de configuración y mantenimiento en la base de datos y también pueden eliminar la base de datos.
db_securityadmin	Los miembros del rol fijo de base de datos <b>db_securityadmin</b> pueden modificar la pertenencia a roles y administrar permisos. Si se agregan entidades de seguridad a este rol, podría habilitarse un aumento de privilegios no deseado.
db_accessadmin	Los miembros del rol fijo de base de datos <b>db_accessadmin</b> pueden agregar o quitar el acceso a la base de datos para inicios de sesión de Windows, grupos de Windows e inicios de sesión de SQL Server.
db_backupoperator	Los miembros del rol fijo de base de datos <b>db_backupoperator</b> pueden crear copias de seguridad de la base de datos.
db_ddladmin	Los miembros del rol fijo de base de datos <b>db_ddladmin</b> pueden ejecutar cualquier comando del lenguaje de definición de datos (DDL) en una base de datos.
db_datawriter	Los miembros del rol fijo de base de datos <b>db_datawriter</b> pueden agregar, eliminar o cambiar datos en todas las tablas de usuario.
db_datareader	Los miembros del rol fijo de base de datos <b>db_datareader</b> pueden leer todos los datos de todas las tablas de usuario.
db_denydatawriter	Los miembros del rol fijo de base de datos <b>db_denydatawriter</b> no pueden agregar, modificar ni eliminar datos de tablas de usuario de una base de datos.
db_denydatareader	Los miembros del rol fijo de base de datos <b>db_denydatareader</b> no pueden leer datos de las tablas de usuario dentro de una base de datos.

Figura 1: Características de los usuarios

Los roles de base de datos simplifican el proceso de gestión de privilegios, ya que se pueden otorgar privilegios a un rol y luego otorgar el rol a usuarios. Cuando desee revocar privilegios para un usuario, simplemente tiene que revocar la autorización de rol del usuario, en vez de revocar cada privilegio individual.

## Privilegios por Rol de usuario

El rol actúa como plantilla que facilitan la asignación de permisos a un usuario de una base de datos. Se soportan los siguientes roles de usuario de base de datos: **Lectura y escritura** (usado de forma predeterminada), Solo **lectura** y Sólo **escritura**. Cada rol presenta un conjunto de privilegios predeterminados que se conceden al usuario de la base de datos al que se le asigne. Si lo desea, puede editar los conjuntos de privilegios correspondientes a los distintos roles.

Cuando un usuario se le otorgue el permiso de **Solo lectura**, solo puede observar los datos, en cambio si se le otorgo el permiso de **Solo escritura** puede modificar ciertos datos de la base pero la visualización de otros datos puede ser limitada, por último hay usuarios que pueden realizar ambas funciones **Lectura y Escritura**: puede ver datos y modificarlos sin ningún problema, claro que hay limitantes y esto depende al rol que cumpla en la base de datos.

Un usuario puede seleccionar roles para los usuarios de bases de datos, así como añadir o eliminar privilegios de forma independiente. A continuación se enumeran los privilegios asignados por omisión a cada rol.

Privilegio	Lectura y escritura	Sólo lectura	Sólo escritura
Seleccionar	+	+	–
Insertar	+	–	+
Actualizar	+	–	+
Eliminar	+	–	+
Crear	+	–	+
Drop	+	–	+
Alter	+	–	+
Index	+	–	+
Create Temporary Tables	+	–	+
Lock Tables	+	–	+
Create view	+	–	+
Show view	+	+	–

Figura 2: Privilegios que pueden tener los usuarios

Los usuarios pueden seleccionar roles (Lectura y escritura, Sólo lectura, Sólo escritura), pero no pueden añadir o eliminar permisos de forma independiente. A continuación se enumeran los permisos predeterminados para cada rol.

Permiso	Lectura y escritura	Sólo lectura	Sólo escritura
db_backupoperator	+	+	+
db_datareader	+	+	–
db_datawriter	+	–	+
db_ddladmin	+	–	+

Figura 3: Privilegios por rol de usuario

## 2. Modelo Orientado a Objetos

La orientación a objetos representa el mundo real y resuelve problemas a través de objetos, ya sean tangibles o digitales. Este paradigma tecnológico considera un sistema como una entidad dinámica formada de componentes. Un sistema sólo se define por sus componentes y la manera en que éstos interactúan



Figura 4: Esquema con las principales características del modelo orientado a objetos

En una base de datos orientada a objetos, los componentes se almacenan como objetos y no como datos, tal y como hace una base relacional, cuya representación son las tablas. Utilizando sobre todo el Paradigma Orientado a Objetos

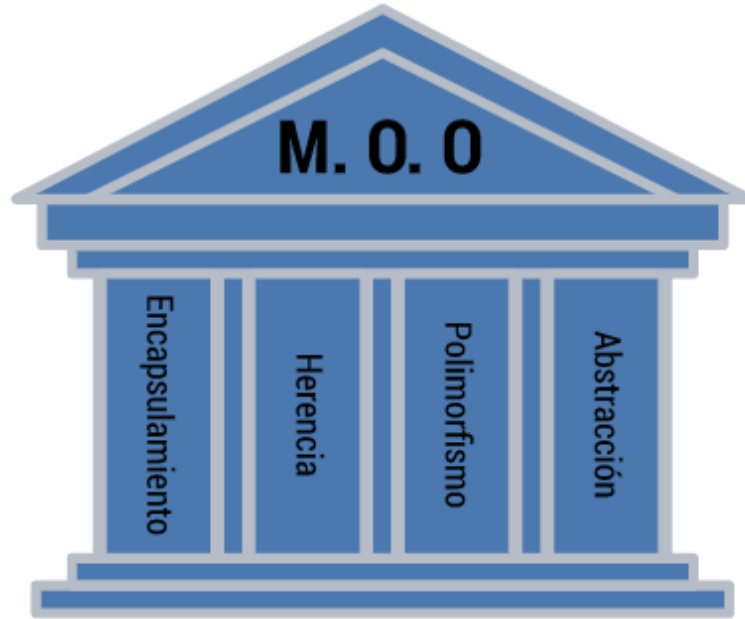


Figura 5: Pilares del modelo orientado a objetos

Los manejadores de bases de datos orientados a objetos deben tomar en cuenta las siguientes operaciones:

- Ser capaces de definir sus propios tipos de datos.
- El tamaño de los datos puede ser muy grande.
- La duración de las transacciones puede ser muy larga.
- Recuperar rápidamente objetos complejos.
- Lenguajes de consulta de objetos, un ejemplo es OQL (Object Query Language).
- Mecanismos de seguridad basados en la noción de objeto.
- Funciones para definir reglas deductivas

La persistencia es una característica necesaria de los datos en un sistema de bases de datos. Recordemos que consiste en la posibilidad de recuperar datos en el futuro. Esto implica que los datos se almacenan a pesar del término del programa de aplicación. En resumen, todo administrador de base de datos brinda persistencia a sus datos.

En el caso de los sistemas de gestión de base de datos orientada a objetos (OODBMS por sus siglas en inglés), la persistencia implica almacenar los valores de atributos de un

objeto con la transparencia necesaria para que el desarrollador de aplicaciones no tenga que implementar ningún mecanismo distinto al mismo lenguaje de programación orientado a objetos.



Figura 6: Persistencia en una base de datos orientada a objetos

Los sistemas de bases de datos orientados a objetos parecen ser la tecnología más prometedora para los próximos años, aunque carecen de un modelo de datos común y de fundamentos formales, además de que su comportamiento en seguridad y manejo de transacciones no están a la altura de los programas actuales de administradores de bases de datos.

Clase	Objetos	Atributos/datos
Empleado	Juan Pérez	Edad: 25
		Puesto: Psicóloga social
		Salario: 8000
	María Suárez	Edad: 23
		Puesto: Pedagoga
		Salario: 15 000

Figura 7: Almacenamiento de datos en un OODBMS

Algo que apoya esta tendencia es que a pesar de que la ingeniería de software orientada a objetos requiere mucho tiempo de análisis, la mayoría de los proyectos de desarrollo son más cortos y requieren menos personas, además de que la cantidad de código es menor.

### 3. Modelo Objeto/Relacional

El **modelo relacional** de bases de datos es llamado así porque se fundamenta en estructuras de almacenamiento de datos llamadas *relaciones*.

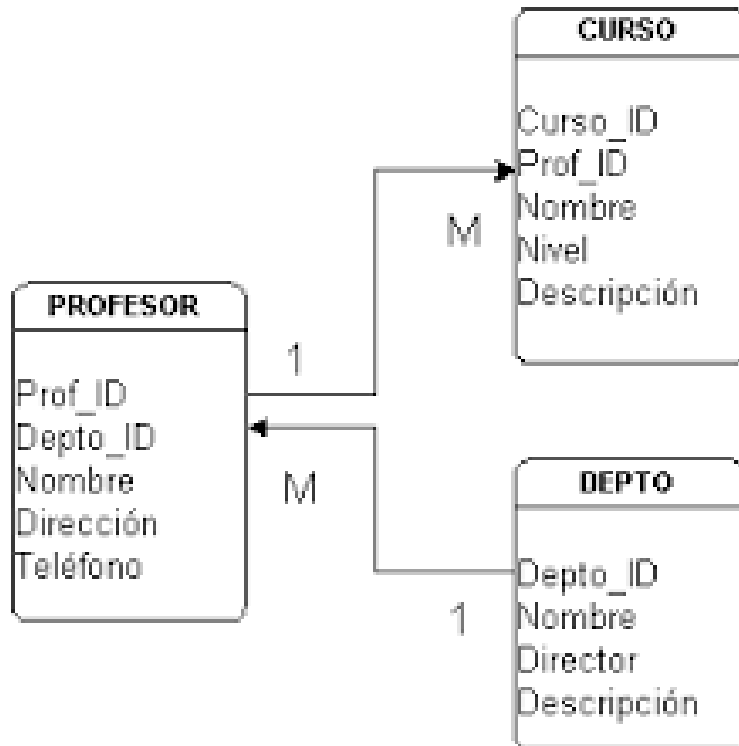


Figura 8: Ejemplo de un Modelo Relacional

#### *Características*

Las dos características fundamentales del modelo relacional de bases de datos son

1. Los datos son percibidos por el usuario como relaciones y nada más que relaciones.
2. Para consultar los datos, el usuario cuenta con operadores que generan nuevas relaciones a partir de otras —álgebra relacional—.

#### *Partes*

El modelo relacional se fundamenta en una teoría abstracta de datos que toma ciertos aspectos de las matemáticas —teoría de conjuntos y lógica de predicados—. De esta manera, podemos definir una relación como un conjunto de tuplas y atributos que se compone de dos partes: encabezado y cuerpo. El encabezado de una relación denota un cierto predicado o función valuada como verdadera. Por su parte, cada fila en el cuerpo denota una cierta proposición verdadera obtenida del predicado por medio de la sustitución de ciertos valores, del tipo apropiado, en los indicadores de posición o parámetros de ese predicado. En otras

palabras, podemos decir que el cuerpo es un conjunto de instancias o ejemplos del predicado.

### *Propiedades*

- Cada relación tiene un nombre distinto
- Los valores de los atributos son atómicos (relaciones normalizadas)
- Cada atributo tiene un nombre distinto
- Los atributos no están ordenados
- No hay tuplas duplicadas.

En una relación, es posible encontrar tres tipos de claves:

Candidata	Principal	Foránea
Es un atributo que identifica de manera única a cada tupla de una relación. Una relación puede tener varias claves candidatas y éstas pueden ser simples, formadas por un atributo o compuestas, cuando se forman por varios. A las últimas se les llama superclaves.	En este caso, a diferencia de las claves candidatas, sólo existe una por relación. También pueden ser superclaves.	Una clave foránea es un atributo que hace referencia a una clave principal en otra relación. Esta referencia implica que no pueda existir un valor en la clave foránea que no exista primero en la clave principal.

Figura 9: Tipos de claves

Comparando el Modelo Relacional y el Modelo Orientado a Objetos podemos diferenciar en una almacena datos y los relaciona, en la otra almacena objetos.

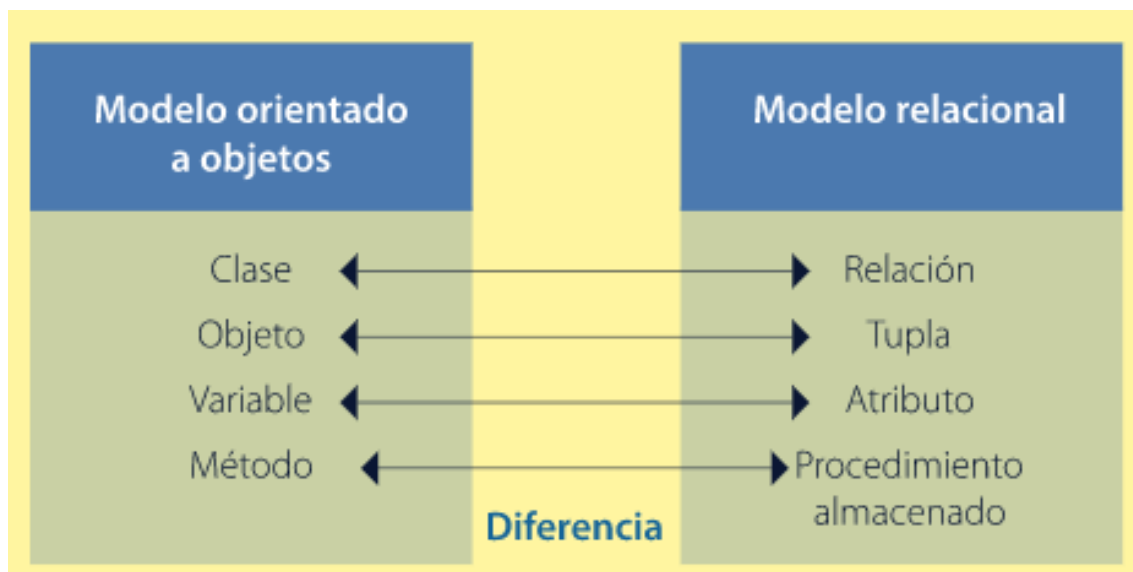


Figura 10: Comparacion del Modelo Relacional vs el Modelo Orientado

## 4. Modelos NoSQL

El termino NoSQL se refiere a la denominación en inglés Not Only SQL. Plantea modelos de datos específicos de esquemas flexibles que se adaptan a los requisitos de las aplicaciones más modernas. Tienen un conjunto increíble de características y varios modelos que descubriremos más adelante.

### *Tipos de Bases de Datos NoSQL*

Existe una clasificación que debemos conocer para entender cómo estas BBDD están construidas y nos pueden ser de utilidad. Veamos cuales son las principales clasificaciones:

#### 1. Bases de datos documentales

Una base de datos documental u orientada a documentos, es un tipo de BBDD NoSQL que orienta su funcionamiento a datos alojados en documentos o tiendas de documentos. Estas bases de datos se utilizan para almacenar, recuperar y administrar datos semiestructurados.

Las bases de datos documentales almacenan cada uno de los registros y los datos asociados en un solo documento. Cada documento contiene datos semiestructurados que pueden ser consultados con diferentes herramientas de análisis. Estas bases de datos ofrecen mucha flexibilidad, escritura rápida, y consultas rápidas gracias a su gran capacidad de indexación. Entre las BBDD documentales más reconocidas encontramos la ya conocida por nosotros a MongoDB, de 10gen, y CouchDB, de Apache.

También gozan de un gran reconocimiento en el mundo de las Bases de datos NoSQL la RavenDB, de Hibernating Rhinos, BaseX, djondb, eXist, SimpleDB, de Amazon, IBM Lotus Domino y Terrastore

#### 2. Bases de datos en orientadas a grafos

Las bases de datos orientadas a grafos son excelentes herramientas para trabajar datos complejos. Estas BBDD nos permite darle representar los datos en estructuras de grafos. Esto es de gran utilidad cuando los datos que vamos a procesar tienen altos niveles de interrelación. Estas versátiles bases de datos, nos permite ejecutar consultas y almacenamiento de datos de cualquier característica sin siquiera preocuparnos por el volumen de los datos.

Gozan de un excepcional rendimiento para responder de forma eficiente al análisis y consulta de volúmenes gigantescos de datos. Ofrece también una diversidad de métodos analíticos y de consulta que la convierten en una de las opciones más flexibles en NoSQL. Es bastante frecuente conseguir la implementación de este modelo de BBDD en estructuras web de blogs. Twitter es uno de los casos más relevantes donde una



BBDD de Grafos está relacionada.

Las Bases de Datos orientadas a grafos también tiene otras funcionalidades que no te puedes perder. Entre las marcas más reconocidas está nuestra ya conocida Neo4j, pero esta no es la única. También están las conocidas HyperGraphDB, InfoGrid, Allegro-Graph, InfiniteGraph, Sones y DEX/Sparksee.

### 3. Bases de datos clave/valor

Las bases de datos clave valor son modelos no relacionales que utilizan un método simple de almacenamiento de datos. Este tipo de BBDD toma los datos como un conjunto de pares “clave-valor” en los que las claves cumplen funciones de identificadores únicos.

Las claves o lo valores pueden ser cualquier tipo de datos. Pueden ser objetos simples o complejos. Estas BBDD son altamente divisibles y brindan grandes capacidades de escalabilidad horizontal.

Estos modelos de BBDD son uno de los preferidos de los clientes NoSQL, porque son simples en cuanto a funcionalidad y brindan alto rendimiento a la hora de ejecutar las lecturas y escrituras de datos. Ya hemos dado algunos ejemplos de Bases de Datos Clave Valor como Cassandra y Redis, pero es obligatorio expandir la lista cuando tenemos importantes BBDD como BigTable de Google, Dynamo de Amazon, Project Voldemort de LinkedIn, OracleNoSQL y Riak.

### 4. Bases de datos multivalor

5. Las bases de datos multivalor son sistemas interesantes que incorporan diferentes características multidimensionales y NoSQL para la clasificación y manejo de los datos. Estas BBDD comparten significativas similitudes con los modelos relacionales tradicionales. Ambos esquemas contienen tablas. Pero que esto no te engañe, las BBDD multivalor proporcionan un esquema de trabajo menos rígido.

Además de proporcionar mayor flexibilidad, los datos almacenados acá pueden contener listas de valores. Esto quiere decir que cualquier dato almacenado puede tener diversos valores asignados.

Estas bases de datos tienen un nivel de complejidad un poco más elevado debido a que incorporan reglas de normalización para su diseño. Entre las BBDD multivalor más conocidas podemos destacar Rocket D3 DBMS, Rocket mvBase DBMS, Rocket U2 Universe, Rocket U2 Unidata, OpenQM, Caché InterSystems, Reality, Jbase, OpenInsight, Extensible storage engine.

## 6. Bases de datos orientadas a objetos

Como bien lo indica su nombre, las BBDD de este tipo están conformadas por objetos. Estos objetos pueden ser de diferentes tipos, sobre los que se definen unas operaciones que determinan sus propiedades de interacción. Las Bases de datos orientadas a objetos han revivido el interés de los usuarios gracias a sus características principales.

Estas BBDD son altamente flexibles y admiten lenguajes de programación como JAVA, Visual Basic o C++. Entre las más Bases de Datos orientadas a objetos más conocidas tenemos a ObjectDB, Zope Object Database, ZooDB, GemStone, Objectivity y Realm.io

## 7. Bases de datos tabulares

Una BBDD tabular no es más que la estructuración de una BBDD en forma de tabla. Incorpora elementos en columnas y líneas. Cada una de las celdas genera intersecciones entre las columnas y las líneas. A estas intersecciones se le asignan una numeración única para establecer un orden eficiente de los datos. Están pensadas para grandes volúmenes de datos.

Estas tienen la capacidad de almacenar gran cantidad de datos dispersos. Entre las principales DDBB de este estilo podemos conseguir a HBase de Apache que es utilizada para soportar el servicio de mensajería de Facebook, también a BigTable de Google y la versión abierta llamada LevelDB y a Hypertable.

## 8. Bases de datos de Arrays

Las Bases de datos arrays sirven para trabar colecciones de datos conocidas como raster data. Sitúan los datos en una cuadrícula regular con más de dos dimensiones. Estas bases de datos se utilizan para representar simulaciones, sensores y datos estadísticos. Son capaces de manejar volúmenes de datos importantes ofreciendo una flexibilidad y escalabilidad.

Estas bases de datos son consideradas una generación tecnológica emergente. De las bases de datos más destacadas que trabajan este modelo podemos mencionar a Oracle que ha profundizado sus trabajos en ella y SciDB, de Paradigm4

Ventajas de Desventajas de utilizar el Modelos de NoSQL

### **Ventajas**

- Alto rendimiento y baja exigencia: un punto común de las BBDD NoSQL es que pueden ejecutarse en máquinas con recursos modestos. A diferencia de los sistemas SQL, la inversión que necesitamos hacer en estos casos es significativamente baja.

- Escalabilidad horizontal: para mejorar el rendimiento de estos sistemas no debemos inventar un mundo nuevo o una revolución total. Podemos incrementar o añadir nodos de forma horizontal para pisar a fondo el acelerador de rendimiento.
- El volumen de datos no es problema: se pueden manejar gran cantidad de datos sin dudarlo. No se generan cuellos de botella ni se corre el riesgo de ralentizar el sistema.
- Flexibilidad: como ya pudimos detallar en cada uno de sus tipos, estas BBDD NoSQL son extremadamente flexibles y podemos hacer que se ajusten a casi todo lo que necesitamos.

## Desventajas

- Funciones de Fiabilidad: esta si se quiere es una de las debilidades más grandes de los modelos NoSQL. Estas bases al no admitir funciones de fiabilidad lleva a los desarrolladores a implementar su propio código, lo que agrega mayor complejidad a los sistemas.
- Aplicabilidad: Esta desventaja está íntimamente relacionada con la anterior. La falta de inclusión de funciones de fiabilidad limita la aplicabilidad de estas BBDD a funciones delicadas como las del sector bancario, limitando mucho su potencial.
- Incompatibilidad en consultas SQL: en la mayoría de los casos las BBDD NoSQL son incompatibles a consultas SQL, lo que agrega una dificultad mayor ya que en caso de ser necesario se debe incluir una consulta manual, que puede hacer los procesos más lentos y complejos.

## Referencias

- S.A. "Modelo Orientado a Objetos de Base de Datos". 26/02/201, de FACULTAD DE CONTADURIA Y ADMINISTRACION SSISTEMA UNIVERSIDAD ABIERTA Y EDUCACION A DISTANCIA DE LA UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO Sitio web:  
[https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod\\_resource/content/8/contenido/index.html](https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/782/mod_resource/content/8/contenido/index.html)
- S.A. "Modelo Relacional de Base de Datos". 26/02/201, de FACULTAD DE CONTADURIA Y ADMINISTRACION SSISTEMA UNIVERSIDAD ABIERTA Y EDUCACION A DISTANCIA DE LA UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO Sitio web:  
[https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/857/mod\\_resource/content/8/contenido/index.html#:~:text=El%20modelo%20relacional%20se%20fundamenta,dos%20partes%3A%20encabezado%20y%20cuerpo.](https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/857/mod_resource/content/8/contenido/index.html#:~:text=El%20modelo%20relacional%20se%20fundamenta,dos%20partes%3A%20encabezado%20y%20cuerpo.)
- S.A. Bases de Datos NoSQL | Qué son, marcas, tipos y ventajas". 26/02/201, de Graph Everywhere Sitio web:  
<https://www.grapheverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/>