

Permisos del motor de base de datos.

CONTROL

Confiere al receptor del permiso capacidades relacionadas con la propiedad. El receptor del permiso dispone de hecho de todos los permisos definidos para el elemento protegible. Una entidad de seguridad a la que se le haya concedido el permiso CONTROL también puede conceder permisos para el elemento protegible. También puede conceder permisos para el elemento protegible.

ALTER

Confiere la posibilidad de cambiar las propiedades, excepto la propiedad, de un elemento protegible determinado. Cuando se concede para un ámbito, ALTER también confiere la posibilidad de modificar, crear o quitar cualquier elemento protegible que esté contenido en ese ámbito.

ALTER ANY <Server Securable>

Confiere la posibilidad de crear, modificar o quitar instancias individuales del Protegible del servidor.

ALTER ANY <Database Securable>

Confiere la posibilidad de crear (CREATE), modificar (ALTER) o quitar (DROP) instancias individuales del *Protegible de la base de datos*.

TAKE OWNERSHIP

Permite al receptor del permiso tomar propiedad del elemento protegible para el que se concede este permiso.

IMPERSONATE <Login>

Permite al receptor suplantar el inicio de sesión.

IMPERSONATE <User>

Permite al receptor suplantar al usuario.

CREATE <Server Securable>

Confiere al receptor la posibilidad de crear el Protegible del servidor.

CREATE <Database Securable>

Confiere al receptor la posibilidad de crear el Protegible de la base de datos.

CREATE <Schema-contained Securable>

Confiere la posibilidad de crear el elemento protegible contenido en el esquema. No obstante, para crear el elemento protegible en un esquema concreto se requiere el permiso ALTER en el esquema.

VIEW DEFINITION

Permite al receptor obtener acceso a los metadatos.

REFERENCES

El permiso REFERENCES es necesario en una tabla para crear una restricción FOREIGN KEY que hace referencia a esa tabla.

- ❖ Microsoft (30 de octubre de 2020). Permisos (motor de base de datos) [mensaje en un blog]. Recuperado de <https://docs.microsoft.com/es-es/sql/relational-databases/security/permissions-database-engine?view=sql-server-ver15>

Modelos de Datos:

Modelo orientado a objetos:

Las bases de datos relacionales son el estándar en programación y desarrollo web desde hace mucho tiempo. En este modelo, la información se almacena en tablas relacionadas entre sí. Las relaciones también permiten almacenar y consultar información compleja compuesta por varios elementos, al igual que en las bases de datos de objetos. En las últimas, sin embargo, todos los atributos de cada objeto están disponibles de inmediato y, además, los registros pueden ser mucho más complejos. Por otra parte, con las bases de datos relacionales, lo más habitual es intentar simplificar al máximo la información que se introduce. Cuanto más complejo se vuelve el conjunto de los datos, más extensas son las relaciones entre ellos, lo que ralentiza la base de datos.

Modelo objeto/relacional:

El modelo de base de datos objeto-relacional integra los conceptos de la tradicional base de datos relacional y los conceptos de paradigma de objetos que se utiliza en la programación orientada a objetos (POO).

El objetivo de este concepto es poder aplicar la tecnología madura de bases de datos relacionales sobre la organización de los datos complejos es decir datos de texto e imagen, mapas, datos en el rango de audio etc. Las bases de datos Objeto-relacional son compatibles con estos objetos de datos y las operaciones de mayor complejidad.

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- **Encapsulación:** Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- **Herencia:** Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- **Polimorfismo:** Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

Modelos NOSQL:

Modelo y metamodelo conceptual: Permitirá entender la relación y la organización de los posibles conceptos a alto nivel describiéndolos en más de un dominio, por lo tanto, vamos a utilizar el punto de vista de Información de ArchiMate (Information Archimate Viewpoint).

Metamodelo lógico: Permite visualizar todos los campos susceptibles a ser utilizados en los conceptos. Para esto, vamos a utilizar un diagrama de clases de UML (UML Class diagram). A cada concepto le corresponde una clase.

Modelo y metamodelo físico: Este modelo extiende el metamodelo lógico y en él se describen las distintas formas en las que un concepto o esquema puede ser almacenado físicamente. Esta visualización permite verificar los modelos/metamodelos conceptual y lógico para, más tarde, visualizar las bases de datos en funcionamiento. Para esto vamos a utilizar un diagrama de clases de UML (también puede ser un diagrama de objetos UML).

- ❖ Charito. (3 de agosto del 2018). Modelos de datos NoSQL [mensaje en un blog]. Recuperado de <https://eaminds.com/2018/08/03/modelando-nosql-data-bases/>
- ❖ Gómez, E. (2018). BDD Objeto Relacional [mensaje en un blog]. Recuperado de <https://sites.google.com/a/espe.edu.ec/bases-de-datos-ii/introduccion/bdd-objeto-relacional>
- ❖ IONOS, (24 de junio de 2020). Base de datos orientada a objetos: el secreto mejor guardado de los modelos de bases de datos Relacional [mensaje en un blog]. Recuperado de <https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-orientada-a-objetos/>