|  |  |  |
| --- | --- | --- |
|  | | |
|  | **Hurlingham**, Agosto del 2021  Buenos Aires, Argentina | |
| **ESTRATEGIAS DE PERSISTENCIA (759\_45)**  Profesor: Lic. Pablo Marcelli  **TP\_CLASE1\_TEORICA**  Alumno: Raul A. Gutierrez  correo electronico: [correo@raulgutierrez.com.ar](mailto:correo@raulgutierrez.com.ar)  [raul.gutierrez@estudiantes.unahur.edu.ar](mailto:raul.gutierrez@estudiantes.unahur.edu.ar)  Entrega: Lunes 16 de Agosto 2021 | |  |

Cuestionario:

1.- Que son las condiciones de integridad

Son las reglas con las que se restringen acciones de usuarios, por ejemplo, con condiciones especificas sobre una base de datos concreta o tambien, las condiciones propias de integridad basadas en un modelo de datos, el cual se debe cumplir para toda base de datos que pertenezca a dicho modelo de datos.

Las condiciones de integridad para usuarios de una base de datos se basan en restricciones, que una vez definidas, el servicio de gestion de base de datos es el encargado de hacer cumplir dichas restricciones.

En las reglas de integridad de un modelo, en cambio, no se define ninguna restriccion para una base de datos en particular, porque ya se consideran preestablecidas por todas las bases de datos que pertenezcan a un modelo. Aqui el servicio de gestion de base de datos, debe velar por el cumplimiento de las reglas y restricciones, ya preestablecidas por su modelo de datos.

2.- Enumere las reglas de integridad y explique cada una.

*Regla de integridad de unicidad de la clave primaria*

Esta regla esta basada en la definicion de una clave primaria, que se establece que para toda relacion que se elija para la clave primaria, no puede haber valores repetidos. Entonces, si en una relacion de R no puede haber en ningun momento dos tuplas con la misma combinacion de valores para los atributos de la clave primaria.

*Regla de integridad de entidad de la clave primaria*

La regla de entidad dispone que los atributos de la clave primaria de una relacion NO pueden tener valores nulos.

Esta regla es necesaria para que los valores de las claves primarias puedas identificar las tuplas individuales de las relaciones.

De haber valores nulos, es posible que las tuplas no se puedan distinguir.

Un servicio de gestion de base de datos, tendra que garantizar el cumplimiento de esta regla de integridad en todas las inserciones y modificaciones que afecten a atributos de la clave primaria.

*Regla de integridad referencial*

Esta regla esta vinculada con el concepto de clave foranea, lo que determina que todo valor que toma una clave foranea, debe ser valores nulos o valores existen en la clave primaria de referencia.

Restriccion:

La restriccion en caso de borrado, consiste en no permitir borrar una tupla, si tiene una clave primaria de referencia por una clave foranea. En caso de modificacion, no se permite modificar ningun atributo de la clave primaria, si esta referenciada por alguna clave foranea.

Actualizacion en cascada:

Consiste en permitir la operacion de actualizacion de la tupla, y efectuar operaciones que propagen en cascada la actualizacion a las tuplas que hagan referencia a esta.

Anulacion:

La anulacion, consiste en permitir la actualizacion de la tupla y en efectuar operaciones que propagen valores nulos a los atributos de la clave foranea, que la referencian. Solo se puede aplicar la anulacion si los atributos de la clave foranea si lo admiten.

*Regla de integridad de dominio*

Esta regla de integridad establece dos condiciones:

condicion 1.-: implica que todos los valores no nulos que contiene la base de datos para un determinado atributo, debe ser del dominio declarado para dicho atributo.

Condicion 2.-: un operador determinado solo se puede aplicar sobre valores que tengan dominios que le sean adecuados.

3.- Que tipo de datos conoce, y para que sirve cada uno?.

* *Datos numericos*
* *Datos alfanumericos*
* *Datos fecha y hora*

*Datos numericos*

Enteros: TINYINT, SMALLINT, MEDIUMINT,INT o INTEGER, BIGINT.

Se usan para almacenar datos numericos enteros, como: edades, cantidades, magnitudes sin decimales, etc.

Decimales: FLOAT, DOUBLE, DECIMAL.

Almacenan valores con numeros decimales, como ser: precios salarios, importes, etc.

En la estructura de un campo FLOAT, se lo puede definir con dos valores, longitud total del numero (incluyendo decimal y coma), y cuantos digitos solo para el decimal, ej.:

FLOAT (6,2) → -999.99 al 999.99

El tipo de dato DOUBLE, es de mayor presicion, solo se permite que la cantidad de decimales se defina entre 25 y 53.

*Datos alfanumericos*

Permite almacenar combinaciones de letras, numeros y hasta caracteres especiales. Hay varios tipos, los cuales dependen de la extension que se requiera.

Tipos: CHAR, VARCHAR, BINARY, VARBINARY, TINYBLOB, TINYTEXT, BLOB, TEXT, MEDIUMBLOB, MEDIUMTEXT, LONGBLOB, LONGTEX, ENUM, SET.

*Datos fecha y hora*

Almacenan valores basados en el transcurso de tiempo, dependiendo el tipo, puede calcular transcurso de horas, minutos y segundo, como asi tambien dias, meses y años.

Tipos: DATETIME, DATE, TIME, TIMESTAMP.

4.- ¿Que entiende por ORM?

Es un objeto de mapeo-relacional, permite convertir datos de nuestros objetos, en el formato correcto, para poder guarda esa informacion en una base de datos (mapeo), creandose una base de datos virtual donde los datos que se encuentran en nuestra aplicacion, quedan vinculados a la base de datos (persistencia).

Gracias a los ORM, nos abstraemos del motor de base de datos que estamos utilizando. Es decir, si mas adelante nos toca cambiar de base de datos, nos resultara mas sencillo, dado el nivel de abstraccion que manejamos respecto al motor que hace funcionar a la base de datos.