

Herramientas de trabajo (4)

Miguel Angel Piña Avelino

Ingeniería de Software,
Facultad de Ciencias, UNAM

17 de septiembre de 2018

Índice

① PostgreSQL

② Java Persistence API

PostgreSQL

PostgreSQL

¿Qué es PostgreSQL?

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos y libre.

PostgreSQL

PostgreSQL se define como:

- Una poderosa base de datos open-source objeto-relacional.

PostgreSQL

PostgreSQL se define como:

- Una poderosa base de datos open-source objeto-relacional.
- Es totalmente compatible con características ACID (atomicidad, consistencia, aislamiento y durabilidad).

PostgreSQL

PostgreSQL se define como:

- Una poderosa base de datos open-source objeto-relacional.
- Es totalmente compatible con características ACID (atomicidad, consistencia, isolación y durabilidad).
- Con soporte para *llaves foráneas, uniones, vistas, triggers y procedimientos almacenados*.

PostgreSQL

PostgreSQL se define como:

- Una poderosa base de datos open-source objeto-relacional.
- Es totalmente compatible con características ACID (atomicidad, consistencia, isolación y durabilidad).
- Con soporte para *llaves foráneas*, *uniones*, *vistas*, *triggers* y *procedimientos almacenados*.
- Soporte de la mayoría de los tipos de datos del estándar ANSI SQL 2008.

PostgreSQL

PostgreSQL se define como:

- Una poderosa base de datos open-source objeto-relacional.
- Es totalmente compatible con características ACID (atomicidad, consistencia, isolación y durabilidad).
- Con soporte para *llaves foráneas, uniones, vistas, triggers y procedimientos almacenados*.
- Soporte de la mayoría de los tipos de datos del estándar ANSI SQL 2008.
- Interfaces para trabajar con varios lenguajes de programación (C/C++, Java, Perl, .Net, Python, etc.)

PostgreSQL

También tiene las siguientes características:

Ver más

- Ser una base de datos de clase empresarial

PostgreSQL

También tiene las siguientes características:

Ver más

- Ser una base de datos de clase empresarial
- Control de concurrencia multiversión

PostgreSQL

También tiene las siguientes características:

Ver más

- Ser una base de datos de clase empresarial
- Control de concurrencia multiversión
- Replicación asíncrona

PostgreSQL

También tiene las siguientes características:

Ver más

- Ser una base de datos de clase empresarial
- Control de concurrencia multiversión
- Replicación asíncrona
- Transacciones anidadas

PostgreSQL

También tiene las siguientes características:

Ver más

- Ser una base de datos de clase empresarial
- Control de concurrencia multiversión
- Replicación asíncrona
- Transacciones anidadas
- Altamente escalable

Límites de PostgreSQL

Algunos límites y características generales que se incluyen en PostgreSQL.

| | |
|-----------------------------------|--|
| Tamaño máximo de la Base de datos | Ilimitado |
| Tamaño máximo de la tablas | 32 TB |
| Tamaño máximo de la fila | 1.6 TB |
| Tamaño máximo para cada campo | 1 GB |
| Máximo de filas por tabla | Ilimitado |
| Maximo de columnas por tabla | 250-1600 dependiendo del tipo de columna |
| Máximo de índices por tabla | Ilimitado |

¿Dónde conseguir información?

`https://www.postgresql.org/docs/9.5/static/index.html`

Java Persistence API

JPA

- JPA (Java Persistence API) nos permite establecer una correlación entre una base de datos relacional y un sistema orientado a objetos.
- Esta correlación es llamada ORM (Object Relational Mapping), la cual genera anotaciones sobre Entidades. JPA establece una interface común que es implementada por un proveedor de persistencia de nuestra elección (TopLink, EclipseLink, Hibernate, entre otros).

JPA

Una entidad es una clase POJO que debe proporcionar un método constructor por defecto, no debe ser final, y debe implementar Serializable para accesos remotos, por ejemplo:

```
@Entity
public class Persona implements Serializable {
    public Persona() {}
}
```

JPA

Todas las entidades tiene que poseer una identidad única (conocido como Primary Key en bases de datos relacionales) por lo que deben tener una propiedad marcada con la anotación `@Id` y `@GeneratedValue` (para la generación de la primary key por el proveedor de persistencia).

```
@Entity
public class Persona implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue
    private int id;

    public Persona() {}
}
```

JPA

Las anotaciones sobre las entidades y sobre las propiedades pueden tener las configuraciones `@Table` y `@Column`. La anotación `@Table` permite definir el nombre de la tabla en la base de datos relacional (si el nombre de la entidad es distinto al nombre de la base de datos) y `@Column` permite definir las restricciones sobre las columnas (como el nombre).

```
@Entity
@Table(name = "Persona")
public class Persona implements Serializable {
    private static final long serialVersionUID =
        1L;

    @Id
    @GeneratedValue
    @Column(name = "id_persona")
    private int id;

    public Persona() {}
}
```

JPA

Cuando una entidad es creada en memoria, todos los atributos (valores de las columnas) son obtenidos al mismo instante de la creación. En algunos casos, no necesitamos obtener un valor de un atributo dado que nunca se utilizará en la implementación. Para modificar el tipo de acceso sobre los atributos, utilizaremos el `FetchType` en la anotación `@Basic` sobre los atributos:

```
@Column(length = 20)
@Basic(fetch = FetchType.EAGER)
private String nombre;

@Column(length = 20)
@Basic(fetch = FetchType.LAZY)
private String direccion;
```

1

¹<http://oraclejuniors.blogspot.com/2014/11/que-es-jpa-java-persistence-api.html>

JPA

Ejemplo práctico.