



Programación orientado a objetos

Sesión 6

Docentes

Nestor Audante

Rony Hancoco



- Base de datos
- Consultas SQL
- Maven



clave primaria

campo (propiedad)

registro

ID	Título	Año	Director
1	The player	1992	Robert Altman
2	Cookie's fortune	1999	Robert Altman
3	The man who shot Liberty balance	1992	John Ford

tabla de base de datos



```
create table department  
  (dept_name varchar (20),  
   building   varchar (15),  
   budget     numeric (12,2),  
   primary key (dept_name));
```



```
insert into course (course_id, title, dept_name, credits)  
  values ('CS-437', 'Database Systems', 'Comp. Sci.', 4);
```



```
update instructor  
set salary = salary * 1.03  
where salary > 100000;
```



```
delete from instructor  
where salary < (select avg (salary)  
from instructor);
```



Proyección

Las consultas deben especificar los campos a mostrar de la tabla

```
SELECT campo1, campo2, campo3, campo4, campo5  
FROM TABLA
```

```
SELECT location_id, department_id  
FROM departments 7
```

	A 2	LOCATION_ID	A 2	DEPARTMENT_ID
1		1700		10
2		1800		20
3		1500		50
4		1400		60

```
SELECT last_name, salary, 12*salary+100  
FROM employees;
```

1



Eliminación de datos duplicados completos

Las consultas pueden devolver datos únicos usando la palabra “distinct”.

SELECT **distinct** *campo1, campo2, campo3, campo4, campo5*
FROM TABLA

1

```
SELECT department_id  
FROM employees;
```

	DEPARTMENT_ID
1	90
2	90
3	90
4	60
5	60
6	60
7	50
8	50

...

2

```
SELECT DISTINCT department_id  
FROM employees;
```

	DEPARTMENT_ID
1	(null)
2	90
3	20
4	110
5	50
6	80
7	60
8	10



Restricción

Las restricciones son puestas como condiciones de valor lógico en la clausula “Where” usando los operadores lógicos (AND, OR , NOT).

SELECT campo1, campo2, campo3, campo4, campo5
FROM TABLA

WHERE ***campo1 < 100 or (campo3 = 500 and campo5 between 1 and 40)***

	EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
1	100 King		AD_PRES	90
2	101 Kochhar		AD_VP	90
3	102 De Haan		AD_VP	90
4	103 Huneold		IT_PROG	60
5	104 Ernst		IT_PROG	60
6	107 Lorentz		IT_PROG	60

	EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
1	100 King		AD_PRES	90
2	101 Kochhar		AD_VP	90
3	102 De Haan		AD_VP	90



Restricción

Para tipos de datos varchar se puede usar el operador like para búsqueda por patrones.

% **Cero, Uno o más caracteres**

. **Un carácter.**

```
SELECT campo1, campo2, campo3, campo4, campo5  
FROM TABLA  
WHERE campo1 like '%HA%'
```



Ordenamiento

Las consultas pueden ser ordenadas asignados mediante la clausula ORDER BY.

```
SELECT campo1, campo2, campo3, campo4, campo5  
FROM TABLA  
ORDER BY campo1 asc, campo2 desc
```



Alias

Las consultas pueden contener alias para las columnas y tablas, sólo se adiciona el alias al lado del elemento

```
SELECT campo1 miAlias1, campo2 miAlias3, campo3 miAlias3  
FROM TABLA miTabla1
```



Funciones agrupadas

Existen funciones que pueden ser calculadas en base a una condición de grupo en la clausula GROUP BY, las cuales son:

Promedio	AVG
Máximo	MAX
Mínimo	MIN
Suma	SUM
Cuenta	COUNT

```
SELECT campo1, sum(campo1)  
-> Sólo pueden ir elementos de  
   agrupación o funciones agrupadas  
FROM TABLA  
GROUP BY campo1
```



Restricción en funciones agrupadas

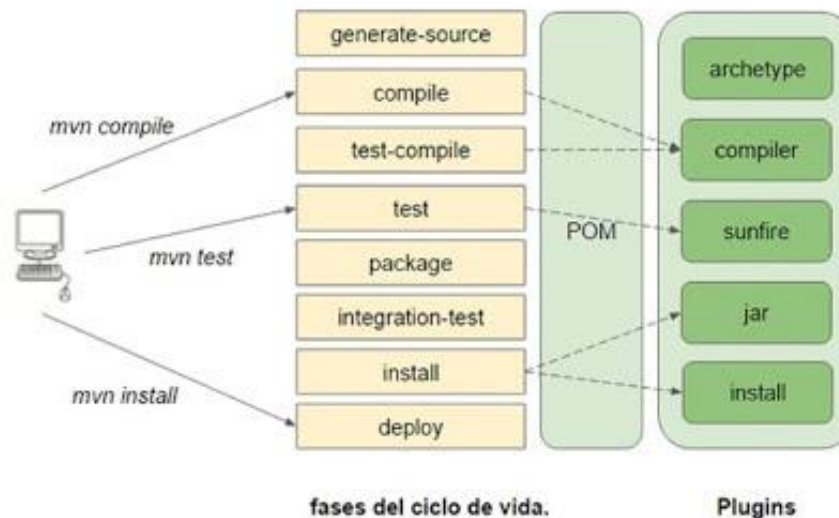
Las restricciones de funciones agrupadas se realizan con la clausula HAVING.

```
SELECT campo1, sum(campo1)  
FROM TABLA  
GROUP BY campo1  
HAVING sum(campo1) > 500
```



¿Para qué sirve?

Permite gestionar el software en sus diferentes ciclos como validar, compilar, probar, empaquetar, desplegar.



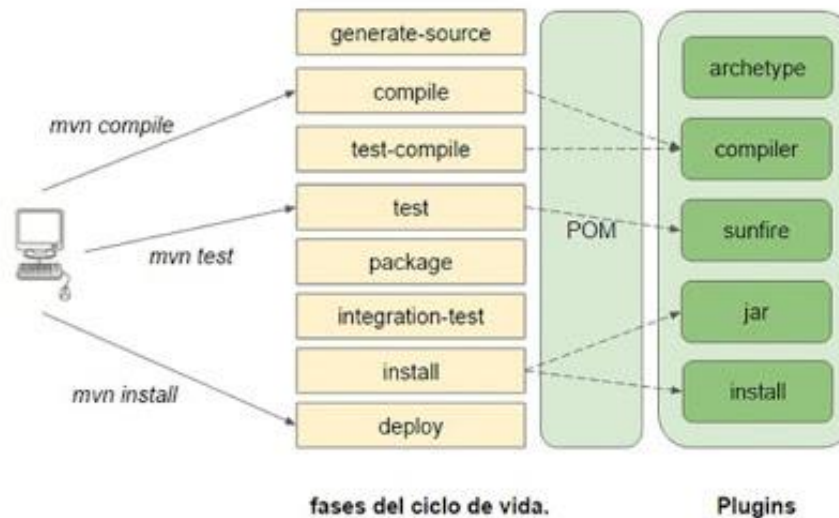


Instalación

<https://maven.apache.org/download.cgi>

Descargar el zip(windows) o tar.gz(linux)

Si dispones de IntelliJ Idea no realizar ninguna instalación.





Dependencias

Centraliza las librerías y sus dependencias declarándolos en el archivo **pom.xml**.

Un Modelo de Objeto de Proyecto o POM es la unidad fundamental de trabajo en Maven.

Es un archivo XML que contiene información sobre el proyecto y detalles de configuración utilizados por Maven para construir el proyecto. Contiene los siguientes valores por defecto para la mayoría de los proyectos.

```
<project>
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1</version>
</project>
```



Declaración de dependencias

Las dependencias se declaran en el segmento dependencias del archivo **pom.xml**.

```
<dependencies>  
  <dependency>  
    <groupId>group-a</groupId>  
    <artifactId>artifact-a</artifactId>  
    <version>1.0</version>  
  </dependency>  
</dependencies>
```



Thanks for listening

Merci pour l'écoute