### TUNING VISTAS

Una vista es un objeto a través de la cual vemos subconjuntos de datos de tablas ya existentes. Los datos que muestra cualquier vista son producto de tablas.

* Los "datos" de las vistas no se almacenan físicamente, se obtienen de las tablas que se están especificando.
* Las vistas se definen por medio de sentencias SQL, que se almacenan en la vista del diccionario de datos **VIEWS.**
* La información se recupera a través de las vistas con la misma sintaxis que hemos utilizado para las tablas poniendo en vez del nombre de la tabla el nombre de la vista.
* Una vista puede especificar un subconjunto de datos vertical (subconjunto de filas) y/o horizontal (subconjunto de columnas) de una tabla.
* Los datos se pueden obtener a partir de múltiples tablas, unión de tablas y combinación de tablas.
* Una vista puede llamar a su vez a otra vista, pero el producto final siempre vendrá de una tabla.

Razones para la utilización de vistas

1.- **Simplificar consultas**

Algunas aplicaciones requieren consultas muy complejas, almacenando consultas intermedias se permite a los usuarios simplificar sus operaciones.

2.- **Seguridad**

Utilizando los mecanismos de seguridad sobre tablas y vistas (lenguaje de control de datos DCL) y las vistas se tiene un sistema potente y flexible de control de acceso a los datos.

3.- **Almacenamiento**

Las vistas son proposiciones SQL que se almacenan en el diccionario de datos de forma compilada, por lo que el costo de análisis y compilación sólo se produce al crearlas. Las vistas son consultas AD HOC, por lo que, al ejecutarlas, ya están en memoria y son más rápidas.

Las vistas contienen sólo datos virtuales, los datos reales se almacenan en los bloques de datos de las tablas, por lo tanto sólo se almacena la definición compilada, por lo que el gasto de almacenamiento es mínimo.

**CREACION DE VISTAS**

Para crear vistas se utiliza el comando del lenguaje de definición de datos CREATE VIEW, que sirve para darle un nombre a un comando SQL válido.

**CREATE OR REPLACE VIEW** <nombre\_de\_la\_vista>

[<lista\_de\_columnas>]

**AS**  <sentencia\_SQL\_válida>;

Se utiliza la **lista\_de\_columnas** para cambiarle el nombre a las columnas que se obtienen con la sentencia SQL. Los alias de las columnas son opcionales.

Queremos crear una vista que contenga información sobre los empleados.

CREATE OR REPLACE VIEW VISTA\_EMPLEADOS

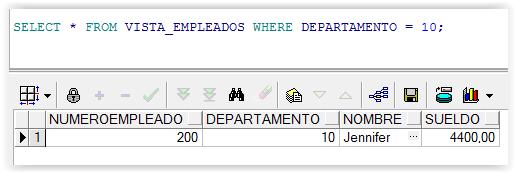
(NUMEROEMPLEADO, DEPARTAMENTO, NOMBRE, SUELDO)

AS SELECT EMPLOYEE\_ID, DEPARTMENT\_ID, FIRST\_NAME, SALARY

FROM EMPLOYEES;

Como comentamos anteriormente las vistas se tratan como las tablas para recuperar datos.

Vamos a seleccionar los datos de empleados del departamento 10.



También podemos crear alias para las columnas utilizando los alias típicos de cualquier consulta SQL.

CREATE OR REPLACE VIEW VISTA\_EMPLEADOS

AS SELECT EMPLOYEE\_ID AS NUMEROEMPLEADO

, DEPARTMENT\_ID AS DEPARTAMENTO

, FIRST\_NAME AS NOMBRE

, SALARY AS SUELDO

FROM EMPLOYEES;

El departamento financiero necesita información sobre los sueldos y números de empleado, su comisión si existe y su número de departamento, para realizar la nómina. Se omitirán todas las otras columnas que no se utilicen para confeccionar la nómina por motivos de seguridad.

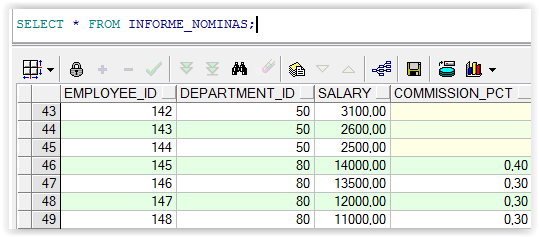
CREATE OR REPLACE VIEW INFORME\_NOMINAS

(EMP\_NO, DEPT\_NO, SALARIO, COMISION)

AS

SELECT EMPLOYEE\_ID, DEPARTMENT\_ID, SALARY, COMMISSION\_PCT

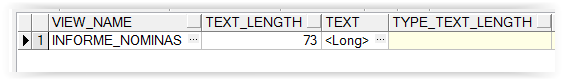
FROM EMPLOYEES;



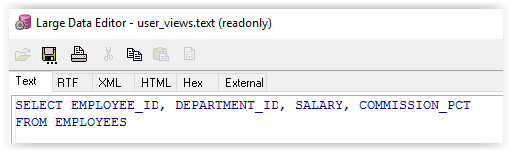
Cada vez que se crea una vista se actualizan las tablas del diccionario de datos para incluir el nombre y la definición de este nuevo objeto.

Para comprobarlo los usuarios pueden consultar la vista del diccionario de datos **VIEWS**, que se encuentra dentro de la tabla USER\_VIEWS.

SELECT \* FROM USER\_VIEWS WHERE VIEW\_NAME = 'INFORME\_NOMINAS';



Y, en la columna TEXT, podremos visualizar el contenido de la vista:



**VISTAS CON COLUMNAS VIRTUALES**

Las columnas que se crean con una vista no tienen por qué ser columnas de la tabla de la que se obtiene la vista, pueden ser también columnas virtuales o campos calculados en las que se agrupa la información de las columnas de la tabla.

Ejemplo: El departamento financiero necesita una vista que le proporcione información estadística sobre los sueldos de los empleados agrupados por departamento.

CREATE OR REPLACE VIEW ESTADISTICA

(DEPARTAMENTO, SALARIO\_MINIMO, SALARIO\_MAXIMO)

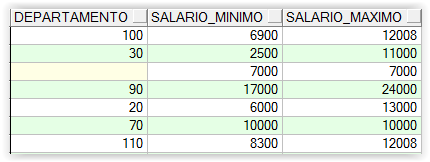
AS

SELECT DEPARTMENT\_ID, MIN(SALARY), MAX(SALARY)

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID;

SELECT \* FROM ESTADISTICA;



**COMBINACIONES PARA DEFINIR VISTAS**

Se pueden definir vistas que obtengan los datos de la combinación de varias tablas.

Queremos que en la vista anterior se recupere el nombre del departamento, en vez de su número.

CREATE OR REPLACE VIEW ESTADISTICA\_NOMBRE

(DEPARTAMENTO, SALARIO\_MINIMO, SALARIO\_MAXIMO)

AS

SELECT DEPARTMENTS.DEPARTMENT\_NAME

, MIN(EMPLOYEES.SALARY), MAX(EMPLOYEES.SALARY)

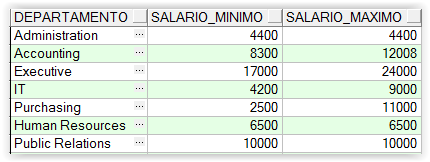
FROM EMPLOYEES

INNER JOIN DEPARTMENTS

ON EMPLOYEES.DEPARTMENT\_ID = DEPARTMENTS.DEPARTMENT\_ID

GROUP BY DEPARTMENTS.DEPARTMENT\_NAME;

SELECT \* FROM ESTADISTICA\_NOMBRE;



**MANIPULACION DE DATOS A TRAVES DE LAS VISTAS**

Actualización de las tablas

Como las vistas son “capas” a través de las que se ven los datos de las tablas, cualquier actualización (inserción, borrado, modificación) de una fila quedará automáticamente reflejada en la vista.

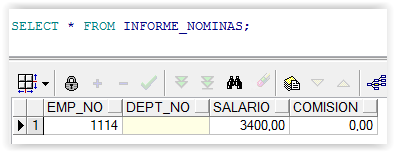
Si insertamos un nuevo empleado en la tabla employees se reflejará en las vistas nómina, estadística\_1 y estadística\_2.

INSERT INTO EMPLOYEES

(EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, COMMISSION\_PCT)

VALUES

(1114, 'JOSE MARIA', 'GUTIERREZ', 3400, 0)



ACTUALIZACION DE LAS VISTAS

También se puede utilizar el lenguaje de manipulación de datos con las vistas, y los cambios quedarán automáticamente reflejados en los bloques de datos de las tablas.

Al igual que con las tablas se pueden conceder privilegios (comando GRANT) sobre las vistas a otros usuarios.

Al insertar filas a través de una vista hay que tener en cuenta que las columnas no especificadas en la definición de la vista tomarán valores nulos, por lo que si una está definida como no nula, no se podrán insertar filas a través de la vista.

La manipulación de datos a través de las vistas se podrá realizar sólo si:

- la vista está basada en una única tabla

- su definición no contiene ni funciones ni expresiones

- su definición no contiene cláusulas GROUP BY

Ejemplo:

CREATE OR REPLACE VIEW INFORME\_NOMINAS

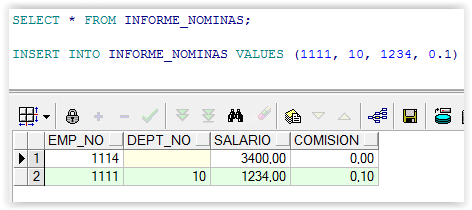
(EMP\_NO, DEPT\_NO, SALARIO, COMISION)

AS

SELECT EMPLOYEE\_ID, DEPARTMENT\_ID, SALARY, COMMISSION\_PCT

FROM EMPLOYEES;

INSERT INTO INFORME\_NOMINAS VALUES (1111, 10, 1234, 0.1)



CLAUSULA **"with check option"**

Esta opción fuerza a que las inserciones y modificaciones hechas a través de la vista afecten sólo a filas que cumplan las condiciones de creación de la vista.

**CREATE OR REPLACE VIEW** <nombre\_de\_la\_vista>

[<lista\_de\_columnas>]

**AS** <sentencia\_SQL\_válida>

**WITH CHECK OPTION**;

Ejemplo:

CREATE OR REPLACE VIEW VISTA\_VENDEDORES

AS

SELECT APELLIDO, SALARIO, OFICIO FROM EMP

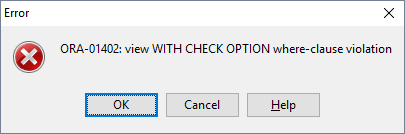
WHERE UPPER(OFICIO) = 'VENDEDOR'

WITH CHECK OPTION;

Ahora vamos a modificar los vendedores mediante la vista. Vamos a cambiar su oficio a COMERCIAL.

UPDATE VISTA\_VENDEDORES SET OFICIO = 'COMERCIAL'

WHERE UPPER(OFICIO) = 'VENDEDOR';



Nos muestra error ya que no se permiten modificaciones en aquellos campos que afecten a los registros que retorna la vista. NO podemos modificar el campo oficio porque al hacerlo, ese registro ya no aparecería en la vista.

Si que nos dejará modificar cualquier otro campo.

Lo mismo sucedería al insertar un registro nuevo:

CREATE OR REPLACE VIEW VISTA\_VENDEDORES

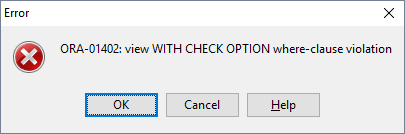
AS

SELECT EMP\_NO, APELLIDO, SALARIO, OFICIO FROM EMP

WHERE UPPER(OFICIO) = 'VENDEDOR'

WITH CHECK OPTION;

INSERT INTO VISTA\_VENDEDORES values(222,'Benito',12345,'ASTRONAUTA')



Solo nos dejaría insertar un registro con valor VENDEDOR en el campo oficio.

INSERT INTO VISTA\_VENDEDORES values(222,'Benito',12345,'VENDEDOR')

VISTAS DE SOLO LECTURA

También podemos crear vistas que solo vayan a ser de lectura, de forma que no podamos modificar el contenido de la tabla a partir de ellas. Es decir, no podremos realizer modificaciones ni inserciones.

CREATE OR REPLACE VIEW VISTA\_VENDEDORES

AS

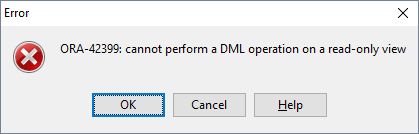
SELECT APELLIDO, SALARIO, OFICIO FROM EMP

WHERE UPPER(OFICIO) = 'VENDEDOR'

WITH READ ONLY;

UPDATE VISTA\_VENDEDORES SET OFICIO = 'COMERCIAL'

WHERE UPPER(OFICIO) = 'VENDEDOR';



**BORRADO DE VISTAS**

El comando DROP VIEW borra la definición de las vistas del diccionario de datos, por lo que no se podrán volver a utilizar.

Las tablas referenciadas en la vista borrada no se verán afectadas.

**DROP VIEW** <nombre\_de\_la\_vista>;

No existe un comando para modificar la definición de una vista, si se desea hacerlo habrá que borrarla y volver a crear la vista.

**RESUMEN**

* Los nombres por omisión de las columnas de las vistas coinciden con los nombres en las tablas de las que se han obtenido.
* Cuando se seleccionen columnas virtuales para crear vistas se les deberá dar un nuevo nombre obligatoriamente.
* Cuando se seleccionen columnas de varias tablas habrá que cambiar el nombre a las columnas que resulten ambiguas.
* Para definir vistas se pueden utilizar otras vistas o tablas.
* Se pueden utilizar funciones, expresiones, la cláusula "group by" para definir nuevas vistas.
* **No se debe utilizar la cláusula "order by" (no tiene sentido ya que perdemos tiempo al consultar)** para definir nuevas vistas, se podrá utilizar cuando se recuperen los datos de las vistas.
* Dentro de las vistas no podemos utilizar lenguaje PL/SQL ni sentencias que no sean SELECT.