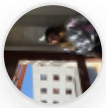
WUOLAH



beacc
www.wuolah.com/student/beacc

84

SQL_Practicas.pdf

Prácticas SQL 1-8

2º Fundamentos de Bases de Datos

Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
UGR - Universidad de Granada

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

PRÁCTICA 1 SQL

DDL -> data definition languaje
Tiene tres instrucciones: CREATE, DROP, ALTER

CREATE table | user Nombre
DROP table | user Nombre
ALTER table | user Nombre

Como cambiar la contraseña:
alter user x4537524 IDENTIFIED BY ugr0506 ;

Creamos las tablas:

TABLA PROVEEDOR

```
create table proveedor (
codpro  char(3) constraint Codpro_es_clave primary key ,
nompro varchar2(30) constraint nompro_no_nulo not null,
status number(2) constraint status_entre_1_y_10 check (status >=1 and status <=10),
ciudad varchar2(15) );
```

*constraint Codpro_es_clave sirve para cuando haya un error que nos diga el mensaje codpro_es_clave y sepamos de que se trata.

T**ABLA PIEZA**

```
create table pieza (
codpie char(3) constraint Codpie_es_clave primary key,
nompie varchar2(10) constraint nompie_no_nulo not null,
color varchar2(10),
peso number(5,2) constraint peso_entre_1_y_100 check( peso>0 and peso <=100),
ciudad  varchar2(15)  );
```

TABLA PROYECTO

```
create table proyecto(
codpj char(3) constraint codpj_ES_clave primary key,
nompj varchar2(20) constraint nompj_no_nulo not null,
ciudad varchar2(15));
```

*Si quisieramos restringir por ejemplo las ciudades check(ciudad IN ('Malaga','Granada','Almeria'))
Si quisieramos restringir un rango check(status between 1 and 10)

TABLA VENTAS

```
create table ventas (
codpro constraint codpro_clave_externa_proveedor references proveedor(codpro),
codpie constraint codpro_clave_externa_pieza references pieza(codpie) ,
codpj  constraint codpro_clave_externa_proyecto references proyecto(codpj),
cantidad  number(4),
constraint clave_primaria primary key (codpro,codpie,codpj) );
```

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

NEWS

WUOLAH

LUN 3 JUN

Noticias para el mundo universitario.

nº 39. Semana del 3 al 9

Destinos más económicos para irte de viaje.

Se acercan las vacaciones de verano y es el mejor momento para pensar en irte de viajes con amigos. Eso sí, a un destino barato pero encantador.

1. Malta. Una combinación entre cultura y paraíso. Malta es uno de los destinos favoritos entre los viajeros gracias a sus tantos lugares de visita, como La Valeta y sus murallas, los Jardines de Hastings o la Catedral de San Juan de La Valeta; y a su mar azul y su perfecta temperatura para disfrutar de un buen cóctel. Un destino cercano y económico.

2. Oporto (Portugal). Además de ser una impresionante ciudad arquitectónica, Oporto cuenta con unas playas celestiales dignas de admirar. A todo ello se suma una exquisita gastronomía de lo más barata y los mejores vinos, no podrás irte de Oporto si probar una Francesinhas junto a un buen vinho do Porto.

3. Tánger (Marruecos). Hemos probado con destinos europeos, ¿Que tal uno de África? De la mano del estrecho de Gibraltar, descubre un nuevo universo de colores, olores y sabores a través de las pequeñas callejuelas de la Medina y los zocos de Tánger. Toma un té en el Hafa, visita el Parque de la Mendubia o pasea por Rue des Siaghins.

4. Canaria (España). Más cercano que nunca. Las islas Canarias son el destino perfecto si no quieres moverte de España. Clima veraniego todo el año, playas paradisíacas en cualquiera de sus islas, zonas boscosas para hacer senderismo, reservas y parques naturales, cuevas y dunas. Todo un mundo idílico a la vuelta de la esquina.

Esto es un Verano Wuolah.

¿Por qué os estamos contando todo esto? Desde Wuolah hemos puesto en marcha un concurso que premiará a los ganadores con el **Verano Wuolah**. Durante el curso os ayudamos a aprobar y ahora nos toca ayudaros a desconectar y descansar y que os quitéis las ralladas estudiantes.

¿En qué consiste este premio? El ganador obtendrá una recompensa de **1000€** para pagar su próxima matrícula, aunque sabemos que 1000€ dan para mucho y podrás invertirlo en muchas cosas. Pero esto no es todo, el premio incluye **un viaje a elegir para el ganador y 3 amigos** en el que costeamos los vuelos al destino.

Para participar tendrás que invitar a nuevos usuarios a que se registren en Wuolah.

> *"Invita a tus amigos, consigue 1000€ y un viaje a elegir para ti y 3 colegas."*

El participante que más amigos invite tendrá **mayor probabilidad** de ganar. Esto es porque cada usuario con cuenta confirmada que tú hayas invitado contará como 1 participación para ti. Para confirmar la cuenta los nuevos usuarios recibirán un mail ¡A invitar!

¿Cómo puedes invitar a tus amigos? Muy fácil, desde https://www.wuolah.com/invite, donde cada usuario tendrá un enlace personalizado que podrá copiar para pasarlo a sus amigos por Whatsapp o publicarlo directamente en los medios y redes que quieras y que se enteren todos tus seguidores. Esto no es todo, los nuevos usuarios que se registren mediante tu invitación recibirán a cambio una participación en el concurso.

¡¡Invita a tus colegas y piensa donde ir!!

Wuolah Giveaway

Heladera TAURUS Tasty Ncream. ¡Día Mundial del Helado de Chocolate! Sorteamos esta heladera para que te refesques con tus sabores favirtos.

Wuolah Giveaway

Set de snorkel. Explora los océanos, contempla los peces más bonitos del mar, corales o estrellas de mar. Participa y llévate este Set de snorkel.

* Clave primaria conjunta se pone al final primary key (codpro,codpie,codpj)
 Clave externa conjunta se pone al final foreign key (...)
 Check más de un atributo (at1+at2 <100)

Para consultar una tabla sin utilizar la interfaz gráfica:
DESCRIBE Nombre_tabla

Añadimos la fecha a la tabla de ventas:
ALTER table ventas **ADD** (fecha date DEFAULT sysdate);

PRÁCTICA 2 SQL

Insertar filas:
Las filas se insertan de una en una con el comando **INSERT INTO** ... **VALUES** (...)
COMMIT para guardar los datos en disco (asegurarse que no se quedan en el buffer)
ROLLBACK deshace la última operación (la borra del buffer)

```
insert into proveedor values('S1','JOSE FERNANDEZ',2,'MADRID') ;
insert into proveedor values('S2','MANUEL VIDAL',1,'LONDRES') ;
insert into proveedor values('S3','LUISA GOMEZ',3,'LISBOA') ;
insert into proveedor values('S4','PEDRO SANCHEZ',4,'PARIS') ;
insert into proveedor values('S5','MARIA REYES',5,'ROMA') ;

commit ;

insert into pieza values ('P1',  'TUERCA'  , 'GRIS'  , 2.5  , 'MADRID');
insert into pieza values ('P2',  'TORNILLO'  , 'ROJO'  , 1.25  , 'PARIS');
insert into pieza values ('P3',  'ARANDELA'  , 'BLANCO'  , 3  , 'LONDRES');
insert into pieza values ('P4',  'CLAVO'  , 'GRIS'  , 5.5  , 'LISBOA');
insert into pieza values ('P5',  'ALCAYATA'  , 'BLANCO'  , 10  , 'ROMA');

commit ;

insert into proyecto values ('J1'  , 'PROYECTO1'  , 'LONDRES'  );
insert into proyecto values ('J2'  , 'PROYECTO2'  , 'LONDRES'  );
insert into proyecto values ('J3'  , 'PROYECTO3'  , 'PARIS'  );
insert into proyecto values ('J4'  , 'PROYECTO4'  , 'ROMA'  );

commit ;
```

Eliminar el contenido de la tabla:
DELETE nombre_tabla;

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Consultar el contenido de la tabla:
SELECT * FROM nombre_tabla ;
(El asterisco significa que quieres que te muestre todas las columnas)
({listra de atributos})
SELECT * | **{lista atributos}** **FROM** nombre_tabla **WHERE** (condicion) ;

select * from pieza ;
select * from proyecto ;
select * from proveedor;

Copiar la tabla de ventas de Olga:
select * from opc.ventas;
insert into ventas (select * from opc.ventas) ;
commit ;

select * from proveedor where (status >2) ;
select * from proveedor where ciudad in ('MADRID' , 'PARIS') ;
select * from pieza where (peso between 10 and 15) ;

Listar los proveedores con status mayor a 4 y que no sean de londres
select * from proveedor where (status >4 and ciudad <> 'LONDRES');

Listar los nombres de piezas grises de mas de 4g
select NOMPIE from pieza where (peso >4);

Listar los pedidos de cantidades superiores a 200 unidades y listando solo los cod de proveedor
select * from ventas where (cantidad > 200);

select CODPRO from ventas where (cantidad > 200);
select distinct CODPRO from ventas where (cantidad > 200);
select NOMPIE from pieza where (upper(color)='GRIS' and peso >4);
select NOMPIE from pieza where (lower(color)='gris' and peso >4);
select * from proveedor where (ciudad is (not) null) ;
select * from proveedor where (NOMPRO like '%M%');
select * from pieza where (NOMPIE like 'A%');
select * from ventas where (fecha <= '01/01/14') ;

Listar piezas que han sido vendidas en un mismo pedido en cantidad mayor de 500 unidades despues del 17 de marzo de 2015
select distinct * from ventas where (cantidad > 500 and fecha > '17/03/15') ;

Listar nombre de piezas grises o blancas que pesen menos de 5g (usando upper)
select NOMPIE from pieza where (upper(color) in ('GRIS','BLANCO') and peso <5);

Listar encuentros jugados tras el 1 de octubre de 2017 en los que haya ganado el visitante (usar la tabla de olga)
select * from opc.encuentros where (fecha > '01/10/17' and RES2>RES1);

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

PRACTICA 3 SQL

Consultas

comando **select (distinct) *** | {lista atributos} | (cantidad/365) * 0,01 **order by** cantidad ;

order by siempre se pone al final de la sentencia, pegado al punto y coma. Muestra los atributos/tabla que queremos consultar ordenada.
Se puede ordenar por dos campos: order by codpro, codpie ;
Ejemplo: como ordenar a los alumnos por primer apellido y luego segundo apellido.

select * from proveedor , ventas ;
La coma realiza el **producto cartesiano** de proveedor x ventas. Mezcla las dos tablas a lo bestia, a cada proveedor añade todas las ventas.

select * from proveedor , ventas where proveedor.codpro = ventas.codpro --> **Reunión natural de tablas**

select proveedor.codpro,nompro,ciudad,codpie from proveedor , ventas where proveedor.codpro = ventas.codpro ;

1)Nombre de proveedores que han vendido la pieza p2
select nompro from proveedor,ventas where proveedor.codpro = ventas.codpro and codpie = 'P2' order by nompro;

2)Datos completos de las piezas grises o rojas vendidas al proyecto j3 (color,peso,ciudad)
select * from pieza,ventas where(ventas.codpie = pieza.codpie and lower(color) in ('rojo', 'gris') and codpj='J3') ;

3)Nombres de proveedor, pieza y proyecto que se encuentren en la misma ciudad (proveedor,pieza,proyecto)
select nompro,nompie,nompj from proveedor,pieza,proyecto where (proveedor.ciudad = pieza.ciudad and pieza.ciudad = proyecto.ciudad) order by nompro ;

4)Ventas hechas antes del 01/01/14 cuya cantidad de unidades supera las 200 y que hayan sido realizadas por un proveedor con status >2
select * from ventas, proveedor where (fecha < '01/01/14' and cantidad >200 and status>2) ;

5)Nombres de proveedor, pieza y proyecto que se encuentren en la misma ciudad (proveedor,pieza,proyecto) y que hayan tenido una venta, una relacion comercial conjunta
select nompro,nompie,nompj from proveedor,pieza,proyecto,ventas where (proveedor.ciudad = pieza.ciudad and pieza.ciudad = proyecto.ciudad and proveredor.codpro= ventas.codpro and pieza.codpie=ventas.codpie and proyecto.codpj=ventas.codpj) order by nompro ;

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

6) Códigos de proveedores que han vendido alguna pieza gris o blanca
select codpro from ventas v, pieza pi where (v.codpie = pi.codpie and upper(color) in ('BLANCO' , 'GRIS')) ;

si sabemos que las piezas blancas o grises son: p1,p4,p12,p26,p32
select codpro from ventas where codpie in ('P1','P4','P12','P26','P32') ;

normalmente deberemos prreguntar al sistema que piezas cumplen esa condicion
select codpie from piezas where color in ('GRIS','BLANCO')

select codpro from ventas where codpie in (select codpie from pieza where color in ('GRIS','BLANCO')) ; --> Subconsulta (más eficiente)

Consultas Baloncesto

1)Listado de los jugadores que participaron en el encuentro entre los equipos Barcelona, Unicaja
select * from opc.jugadores,opc.faltas where (opc.faltas.eq1='FCB' and opc.faltas.eq2='UNI' and opc.faltas.codj = opc.jugadores.codj) ;

2)Nombre del equipo del jugador Felipe Reyes
select nombree from opc.equipos where((select opc.jugadores.code from opc.jugadores where nombrej = 'Felipe Reyes') = opc.equipos.code);

3)Jugadores del Valencia que han cometido más de 3 faltas personales jugando en casa
select * from opc.jugadores,opc.faltas where opc.jugadores.codj = opc.faltas.codj and opc.jugadores.code= 'RMA' and num>3 and eq1='RMA' ;

PRÁCTICA 4 SQL

DELETE ventas **where** fecha < '01/01/14'
Si solo se quiere borrar una fila con delete después del where tendrá que ponerse la CP de dicha fila

UPDATE ventas **set** ciudad = 'Lisboa' **where** ciudad = 'Madrid'

Adelantar todas las fechas un mes a las ventas de S1
update ventas **set** fecha = fecha + 30 where codpro = 'S1' ;

select (distinct) * | <lista atributos> | operaciones + - * / from tabla(s) alias where **condiciones {< , > , in, not in, between …}**

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

OPERADORES DE CONJUNTO

UNION es el operador union de conjuntos como en algebra. Como en los conjuntos no existen elementos repetidos, en este caso es el unico en el que sql los elimina. Si se desea mostrar los repetidos se debera poner **union all. Se puede formar con el producto cartesiano y or**

INTERSECT es el operador interseccion, se queda solo con aquellas filas que sean iguales en ambas. **Se puede formar con el producto cartesiano y and.**

MINUS es el operador diferencia, devuelve objetos del primer conjunto . Al primer conjunto le resta el segundo conjunto y devuelve lo que queda. Es minimal, no se puede formar con otros operadores.

Consultas

Encotrar Codpie que han sido vendidas por S1 y por S4

select codpie from ventas where codpro = 'S1' **intersect** select codpie from ventas where codpro = 'S4' ;
select v1.codpie from ventas v1, ventas v2 where v1.codpie = v2.codpie and v1.codpro = 'S1' and v2.codpro = 'S4';

Nombres de proyectos que han usado P1 y P3

select nompj from proyecto where codpj **in** (select codpj from ventas where codpie= 'P1' **intersect** select codpj from ventas where codpie='P3') ; mas eficiente

select nompj from (select codpj from ventas where codpie= 'P1' **intersect** select codpj from ventas where codpie='P3' t), pryectos p where t.codpj = p.codpj

Nombres de proyectos que han usado P1 pero no P3

select nompj from proyecto where codpj in (select codpj from ventas where codpie = 'P1' **minus** select codpj from ventas where codpie = 'P3') ;

select (nomp)(*) j from proyecto where codpj in (select codpj from ventas where codpie = 'P1' **and** codpj **not in** (select codpj from ventas where codpie = 'P3')) ;

3.7 Ciudades donde viven proveedores con status mayor que 2 en las que no se fabrica 'P1' usando intersect
select ciudad from proveedor where status >2 intersect select ciudad from pieza where codpie <> 'P1' ;

3.8 Codpj a los que solo abastece 'S1'
select codpj from ventas where codpro='S1' ;

Ejercicios 3.7, 3.8, 3.9, 3.10, 3.12, 3.13 , 3.15, 3.16, 3.17 , 3.18 , 3.19 , 3.20, 3.21

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

PRACTICA 5 SQL

Funciones de agregación:

Sum(n_columna)
max()
min()
avg()
count()

select sum(cantidad) from ventas ;
select * from ventas where cantidad = (select **max**(cantidad) from ventas) ;

Ventas que superan a la media
select * from ventas where cantidad > (select **avg(**cantidad) from ventas) ;

select **count (*)** from ventas ; -->Devuelve el num de filas (incluye los valores nulos)

Piezas diferentes que se han vendido
select **count (distinct** codpie **)** from ventas ; -->Devuelve los valores no nulos y diferentes entre si

select **count (**ciudad**)** from ... -> Devuelve el num de valores no nulos en esa columna

Generacion de grupos -> group by

palabras mágicas: para cada

Para cada proveedor dime los pedidos realizados (codpro y cantidad de pedidos)
select codpro,count(*) from ventas **group by** codpro ;
select codpro,count(*) from ventas **group by** codpro **order by** count(*) ;

Codpro y nombre de los proveedores junto con las cantidades de pedidos de cada uno
select * from proveedor **p**, (select codpro,**count(*)** from ventas **group by** codpro) **n_ped**
where (p.codpro = n_ped.codpro) ;

La cantidad de piezas que ha vendido cada proveedor
select codpie,codpro ,count(*) from ventas group by codpro,codpie ;

EJERCICIO 3.32
Encontrar la cantidad media de unidades de la pieza p1 realizada por cada proveedor
select codpro, avg (cantidad) from ventas where(codpie = 'P1') group by codpro

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

-Sobre la consulta anterior, mostrar solo aquellos proveedores cuya cantidad de piezas p1 vendidas supere las 300 unidades

Filtrado despues de una tabla agrupada having siempre va a ir detrás de group by
select codpro, avg (cantidad) from ventas where(codpie = 'P1') **group by** codpro
having avg (cantidad) >300 ;

select * from (select codpro, avg (cantidad) media from ventas where(codpie = 'P1') group by codpro) where (media >300);

-Sobre la consulta anterior, mostrar solo aquellos proveedores que sean de Londres
select codpro, avg (cantidad) from ventas where(codpie = 'P1') **group by** codpro and **having codpro in** (select codpro from proveedor where ciudad = 'Londres') ; no funciona

EJERCICIO 3.33
Encontrar la cantidad total (sum) de cada pieza enviada a cada proyecto
select codpj,codpie, sum(cantidad) from ventas group by codpj, codpie ;

EJERCICIO 3.35
Mostrar los nombres de los proveedores tales que el total de sus ventas supere las 1000 unidades
select nompro from proveedor p, (select codpro, sum(cantidad) cant from ventas group by codpro) num_v where (p.codpro = num_v.codpro and cant >1000) ;

EJERCICIO 3.36
Mostrar para cada pieza la maxima cantidad vendida
select codpie, max(cantidad) from ventas group by codpie ;

Codigo de la pieza más vendida
select codpie, sum(cantidad) from ventas group by codpie having sum(cantidad) **> = all** (select **sum(**cantidad**)** from ventas group by codpie) ;
select codpie, sum(cantidad) from ventas group by codpie having sum(cantidad) **=** (select **max(sum(**cantidad**))** from ventas group by codpie) ;

valor >= ALL | ANY
> ALL | ANY
<= ALL | ANY
<> ALL |ANY

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

EJERCICIO 3.42
Mostrar los codigos de aquellos proveedores que hayan superado a las ventas totales del proveedor S1

EJERCICIO 3.43
Mostrar los mejores proveedores, que hayan vendido mas piezas

EJERCICIO 3.45
Encontrar aquellos proveedores que hayan hecho al menos 10 pedidos

EJERCICIO 3.49
Encontrar la cantidad media de piezas suministrada por aquellos proveedores que venden la pieza p3

PRÁCTICA 6 SQL

División:
Las columnas comunes entre dividendo/divisor desaparecen.

Piezas que han sido compradas por todos los proyectos
Π codpie,codpj (ventas) ÷ Πcodpj (proyecto) --> La división en algebra

Where exists (select * ....)
not exists (select * ...)

exists vale verdadero si la select que hay dentro devuelve alguna fila
not exists vale verdadero si la select que hay dentro no devuelve nada

Todos los datos de los proveedores que no han hecho ninguna venta todavia
select codpro from proveedor ***minus*** *select codpro from ventas ;*
select * from proveedor where not exists (select * from ventas where (ventas.codpro = proveedor.codpro)) ;

Piezas que han sido compradas por todos los proyectos
dividendo : codpie, codpj divisor: codpj de todos los proyectos guardados en la BD

select **distinct** codpie from ventas v1 **where not exists**
((select codpj from proyecto) **minus** (select codpj from ventas v2 **where** v1.codpie = v2.codpie)) ;

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

select codpie from pieza p1 **where not exists**
((select codpj from proyecto) **minus** (select codpj from ventas v2 **where** p1.codpie = v2.codpie)) ;

Proveedores que suministran todas las piezas de color blanco
select * from proveedor p **where not exists**
((select codpie from pieza where color='BLANCO') **minus**
(select codpie from ventas v **where** p.codpro= v.codpro)) ;

Proyectos que han sido abastecidos por todos los proveedores con status mayor que 2
select * from proyecto j **where not exists**
((select codpro from proveedor where status>2) **minus**
(select codpro from ventas v **where** v.codpj = j.codp j)) ;

proveedor |><| (Πcodpj,codpro (ventas) ÷ Πcodpro(σ(status>2) proveedor)) --> La división en algebra

Piezas que han sido vendidas por el mismo proveedor a todos los proyectos de Londres
select distinct codpie,codpro from ventas v1 **where not exists**
((select codpj from proyecto p where p.ciudad='LONDRES') **minus**
(select codpj from ventas v2 **where** v1.codpro=v2.codpro and v1.codpie=v2.codpie)) ;

EJERCICIO 3.47
Encontrar la cantidad total de piezas que ha vendido cada proveedor que cumple la condición de vender todas las piezas suministradas por el proveedor S2

Proveedores que han vendido todas las piezas suministradas por S2:
select codpro,cantidad from ventas p where codpro<>'S2' and not exists
((select codpie from ventas where codpro = 'S2') minus
(select codpie from ventas where ventas.codpro = p.codpro)) ;

Cantidad total de piezas vendidas por cada proveedor anterior
select codpro, sum (cantidad) from ..... group by codpro ;

select codpro, sum (cantidad) from (select codpro,cantidad from ventas p where codpro<>'S2' and not exists
((select codpie from ventas where codpro = 'S2') minus
(select codpie from ventas where ventas.codpro = p.codpro))) group by codpro ;

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

PRÁCTICA 7 SQL

Fecha : **DATE**
El tipo de dato fecha se puede representar de muchas maneras: 'dd/mm/yy' es el formato que utiliza la escuela pero tambien hay otros como 'dd/mmm/yyyy'
Fecha de hoy: **sysdate**

Formatos de las fechas:

d -> dias numerados del 1-7 de lunes a domingo
dd -> dias numerados del 1 al 31
ddd -> dias numerados del 1 al 365
day/DAY/Day -> escribe con letras el dia de hoy en mayuscula o en minuscula

mm -> meses con dos digitos del 1 al 12
month -> meses con letras
mon -> escribe solo las 3 primeras letras del mes

yy -> años mostrando solo los dos ultimos digitos
yyyy -> años con sus 4 digitos
year -> año escrito con letra

Introducir fechas:

TO_DATE : Convierte una cadena en una fecha

insert into cuadros values ('C28 ' , ... , **to_date** ('18 - 04 - 1720 ', 'dd-mm-yyyy')

insert into ventas values ('S5' , 'P2', 'J1' ,100, to_date(' 05/06/1998' , 'dd/mm/yyyy')) ;

Visualizar fechas:

TO_CHAR : Convierte una fecha en un char

select codpro, codpie, codpj, **to_char** (fecha, 'Day, dd "de" Month "de" yyyy') from ventas ;

Todos los pedidos del año 2016
select * from ventas where **to_char**(fecha, 'yyyy') = '2016' ;

Pedidos que se hacen en los primeros 15 dias de cada mes
select * from ventas where **to_char**(fecha, 'dd') <= '15' ;

Pedidos que se hayan hecho en Enero
select * from ventas where **to_char**(fecha, 'mm') = '01' ;
Cantidades totales vendidas cada año
select **to_char**(fecha, 'yyyy') , sum(cantidad) from ventas group by **to_char**(fecha, 'yyyy') order by **to_char**(fecha, 'yyyy') ;

Agrupar por meses el numero de pedidos hechos en 2015
select to_char(fecha, 'mm') , count(*) from ventas where to_char(fecha, 'yyyy') = '2015' group by to_char(fecha, 'mm') order by to_char(fecha, 'mm') ;

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

Para cada año las ventas que se han hecho mensuales
select to_char(fecha, 'yyyy'), to_char(fecha, 'mm'), sum(cantidad) from ventas
group by **to_char**(fecha, 'yyyy'), to_char(fecha, 'mm') order by **to_char**(fecha, 'yyyy'),
to_char(fecha, 'mm') ;

Pedidos de hoy
select * from ventas where fecha= sysdate ;

Vistas y Catálogo de la Base de Datos

VISTAS de una Base de Datos **:**son tablas virtuales que muestran la ejecución de comandos **NIVEL EXTERNO**

CREATE VIEW ventas_mes_2015 AS **select to_char**(fecha, 'mm') Mes, **count(*)**
NPedidos from ventas **where to_char**(fecha, 'yyyy') = '2015'
group by to_char(fecha, 'mm') **order by to_char**(fecha, 'mm') ;

CATALOGOS

User_tables : contiene todas las tablas de cada usuario, la informacion de donde se almacena, cuanto ocupa
describe user_tables ;
select table_name from user_tables ;

user_tables es una vista de la tabla original dba_tables
user_tab_columns
describe user_tab_columns

Describe es una macro, se forma con esta consulta
select table_name, column_name, data_type,data_length, nullable from user_tab_columns;
select table_name, column_name, data_type,data_length, nullable from user_tab_columns where table_name = 'VENTAS' ;

Para consultar las restricciones de mis tables :
select * from user_constraints;

PRÁCTICA 8 SQL

Cluster: Almacena fisicamente la reunion natural (algebra) de dos tablas. (Página mixta)

Un ejemplo seria un cluster para proveedor y ventas:
En la primera página almacena el primer proveedor: S1 Juan 3 Londres y justo a continuación añade todas las ventas que ha realizado S1.
Para el segundo proveedor, S2, utiliza otra página distinta.

WUOLAH

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Hay que informar al sistema que se va a hacer un cluster antes de crear las tablas. Los clusteres tienen utilidad si se reunen mucho ambas tablas y llega a afectar al rendimiento del sistema.

Si en el cluster de proveedor y ventas hacemos select*from proveedor tiene un rendimiento malísimo ya que ahora por cada pagina hay un unico proveedor, habría que leer 100 paginas (por ejemplo) mientras que si no estuviese en un cluster con 2 o 3 páginas ya tendriamos todos los proveedores. Sin embargo, en selec* from ventas apenas se nota la bajada de rendimiento porque ocupan toda la pagina excepto un bloque, el del proveedor, y no hay apenas diferencia.

Oracle tiene un pequeño indice para los clusteres ya que no está ordenados, con punteros a las páginas de comienzo . Este indici hay que crearlo manualmente : CREATE INDEX ....

Cómo crear un cluster

CREATE CLUSTER prov_ventas (codpro char(3)) ; --> create cluster nombre_cluster (atributo comun ambas tablas y tipo)

TABLA PROVEEDOR

```
create table proveedor2 ( codpro  char(3) ,
nompro varchar2(30),
status number(2) ,
ciudad varchar2(15) ) cluster prov_ventas (codpro);
```

TABLA VENTAS

```
create table ventas2 (
codpro char(3),
codpie char(3),
codpj char(3),
cantidad  number(4))  cluster prov_ventas (codpro);
```

Cómo consultar un cluster

select * from **user_clusters;**

Cómo crear un indice de un cluster

create index indx_prov_ventas **on cluster** prov_ventas **;**

Rellenamos las tablas

```
insert into proveedor2 (select * from proveedor) ;
insert into ventas2 ( select codpro,codpie,codpj,cantidad  from ventas) ;
commit ;
```

Cómo cargarme todo un cluster

drop cluster prov_ventas **including tables** (cascade constraints) **;**

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH