TEMA 1

Cambiar password > alter user x-login identified by password;
nueva

Salir > commit ; exit;

Las palabras reservadas pueden ir en mayúscula o minúscula indistintamente.

Creación tablas

```
> create table prueba1 (
    cad char(3),
    n int,
    x float);
```

> describe prueba1
indica los valores y restricciones básicas de la tabla.

En la creación de tablas, cada atributo puede ser:

cad char(3) [NOT NULL] PRIMARY KEY clave primaria (puede omitirse)
UNIQUE clave externa
REFERENCES nombre_tabla2 (atributo);

```
estadocivil varchar2(10)
  Check(estadocivil IN ('soltero','casado')
```

indica que debe tomar uno de esos valores

Tipos datos

int, float, char, varchar(n), varchar2(n), number(p,s), long, date

varchar(n): ilimitado
varchar2(n): max 4000
number(p,s): precisión p, escala s
date: fecha

Tablas

> drop table nombre_tabla — Elimina la tabla.

```
>alter table nombre_tabla add(
    atributo [tipo] ... );
```

Añade un atributo.

```
> alter table nombre_tabla add
    constraint nombre_restriccion | UNIQUE | PRIMARY KEY lista_columnas
                                  | foreign key lista_columna references tabla()
                                  | check (condicion)
```

Eliminar restricción > drop constraint nombre_restricción

Constraint le da nombre a las restricciones.

TEMA 2

> insert into nombre_tabla [(colum1, colum2, ...)]
values (v1, v2, ...) — Inserta Valores en la tabla

> insert into nombre_tabla [(colum1, colum2 ...)]
(select colum1, colum2, ...
from nombre_tabla2); — Inserta valores de la tabla 2 en tabla 1.

SYSDATE indica la fecha y hora del sistema.

Mostrar tablas.

> select * from nombre_tabla; — Proyecta todos los atributos de la tabla.

> select campos from tabla1;

> select table_name from user_tables — muestra todas las tablas creadas.

Modificar tablas

> update nombre_tabla
set estadocivil = 'divorciado'
where nombre = 'Juan'; — Modifica el estado de tabla

> delete from prueba2; — Borra todas las tuplas de prueba2

> delete from prueba
where <condicion> — Elimina las tuplas de prueba donde condicion es cierta.

SYSDATE+1, mañana, SYSDATE+7, semana siguiente.

To_date ('22/10/2000', 'dd/mm/yyyy') permite poner las fechas así.

To_char (fechaalta, 'dd-mon-yyyy') muestra las fechas en el formato pero son de tipo char.

TEMA 3

> select [distinct | all]
<expresion>

from [esquema] tabla|vista

[where <condicion>]

> select ciudad from proyecto; proyecta la columna ciudad de la tabla proyecto.

> select distinct ciudad from proyecto; lo mismo eliminando repetidos

$\Pi_{codpro}(\sigma_{codpj='J1'}(ventas))$ = select codpro from ventas where codpj = 'J1';

like compara cadenas caracteres mediante patrones.
% se sustituye por cualquier cadena
_ se instituye por cualquier caracter.

> select * from proveedor where ciudad like 'L%';

redondear round(n,d)
truncar trunc(n,d)
n es número, d es el decimal elegido.

aproximar por el entero inferior ( ) floor( )
aproximar por el entero superior ( ) ceil( )

IS [NOT] NULL es para comparar con el valor nulo.

> select * from proveedor where status IS NOT NULL.

<instrucción> UNION ∪ <instrucción>
UNION ALL
INTERSECT ∩
MINUS −

Unión all devuelve repetidos, los demás no

Para el prod. cartesiano, solo hay que incluir más tablas después del from.

La reunion natural se hace con NATURAL JOIN en la clausula from.

proveedor s, (select * from ventas where cant >800) v
where s.codpro = v.codpro

> select cantidad, nompro
from proveedor NATURAL JOIN (select * from ventas where cantidad>800)

Si queremos reunir a partir de campos con distinto nombre

- Prod. cart. con dación reunión en clausula where
- clausula JOIN... ON que es la equi-reunion

Mismo ejemplo

> select nompro, cantidad con JOIN... ON
from proveedor s JOIN (select * from ventas where cantidad>800) v
ON (s.codpro = v.codpro)

ORDER BY permite ordenar los resultados (orden ascendente)

>select nompro from proveedor order by nompro ASC;

Podemos indicar con ASC o DESC el orden usado.

Subconsultas

select <expresion> from <tabla>
where <expresion> OPERADOR < select instruction>

operador < IN
EXISTS (devuelve true si ∃ alguna tupla sobre lo que se aplica)

Ej. IN
> select codpie from ventas
where codpro IN (select codpro from proveedor where ciudad=londres)

Ej EXISTS
> select codpro from proveedor
where exists (select * from ventas where
ventas.codpro = proveedor.codpro and ventas.codpie='P1');

otros operadores: < | >=| <=| > | ALL | ANY

> select codpie from pieza
where peso > ANY (select peso from pieza where nompie LIKE 'Tornillo

Operador División

$\Pi_{codpro, codpie}(SP) \div \Pi_{codpie}(P)$

(proveedores que suministran todas las piezas)

AR

Relacion1 ÷ Relacion2 =

$\Pi_A(R1) - \Pi_A((\Pi_A(R1) \times R2) - R1)$, $A = \{Atrib1 - Atrib2\}$

$\Pi_A(R1) - \Pi_A[\Pi_B(\Pi_A(R1) \times \Pi_C(R2)) - \Pi_{BA}(R1)]$

C = Atrib 2
B = Atrib 1

CR

Otra forma de verlo es la siguiente
"Seleccionar proveedores tal que ∄ una pieza para la que ∄ un suministro de ese proveedor".

MIXTA

3ª alternativa:
"Seleccionar proveedores tal que (todas las piezas) menos (piezas suministradas por ese proveedor) sea vacio."

Funciones Agregación

SUM(), suma | MIN(), mínimo | MAX(), máximo | AVG(), media

COUNT(), cardinal | STDDEV(), desviación típica.

Podemos usar distinct.

>Select Max(cantidad) from ventas;

Formando grupos

GROUP BY , HAVING

> select codpro , avg(cantidad) from ventas
group by (codpro) having count(*) > 3;

En having podemos usar todas las funciones de agregación

TEMA 4

Una vista es una presentación de datos procedentes de tablas consiste en asignar un alias a la salida de una consulta y utilizarla. Nivel más alto dentro del nivel lógico.

El catálogo de la BD es una porción de la BD que usa las vistas para mostrar a cada usuario la información que le concierne

> select * from all_views; nos da todas las vistas del catálogo.

las vistas No contienen → ni group by ni fagregación.
- → ni distinct
- → solo sobre 1 tabla
- → atributos nulos o prum key deben insertarse en la def.

> create (or replace) view gemplo as(...);

TEMA 5

user_tables tiene infor de las tablas de las que el usuario es propietario.

table_name, tablespace_name, num_rows, avg_space
(nombre tabla, espacio donde está, información extra)

Privilegios
- Alto nivel (create session, create tablespace)
- gestión objetos esquema usuario (create table)
- gestión objetos esquema * (create any table)

comandos que controlan privilegios: GRANT y REVOKE

TEMA 6

No necesario ni conveniente crear índices sobre la clave primaria

> Create index nombre ON tabla

Mejor rellenar la tabla y luego añadirle el índice.

> Select * from user_indexes where index_name like 'indice_prov'

> create index indicelibros ON libros (genero, titulo, editorial)

> drop index nombre.

Al borrar una tabla se borran los índices asociados

> create index nombre ON tabla (...) Reverse

Bitmap (Baja cardinalidad, ↑ en consultas)

Tablas IOT : > Create index nombre ON tabla (...) ORGANIZATION INDEX

Cluster

> Create cluster nombre (campo tipo_dato)

> Create table ___(

:

Cluster nombre-cluster(clave-cluster))

> Create index indice ON Cluster cluster;

cláusulas
- Hash is (clave con único atributo tipo number y enteros)
- SIZE (estimación del tamaño del cluster
- Hash keys (nº valores distintos

> drop cluster nombre;

> drop index nombre;

Privilegios sobre el sistema

- grant
  - Public: todos los usuarios
  - With admin option: permite que el usuario pueda otorgar privilegio a otros
- Revoke
  - Deroga privilegios creados por grant.
  - Vigilar los efectos al derogar un privilegio.
  - No hay efecto cascada.

Privilegios sobre objetos

(grant)

	Tabla	Vista
Delete		
Insert	"	"
Referenas	"	"
Update	"	"
Execute	Procedimiento	
Index	Tabla	
Select	Tabla	Vista Secuencia

All hace ref a todos los privilegios concedidos con With grant option, que autoriza al usuario a conceder el privilegio.

(Revoke) Cascade constraints propaga la derogación hacia todas las tablas.

El usuario solo puede deroga aquello que ha concebido

Siempre cascade con respecto a la congestión con With grant option.

> grant select ON nombre_tabla TO usuario (with grant option)

> revoke select ON nombre_tabla FROM usuario