

SQL

```
SELECT object_type, count(object_type)
FROM dba_objects
GROUP BY object_type
ORDER BY object_type;
```

SQL

```
SELECT object_type, count(object_type)
FROM dba_objects
GROUP BY object_type ORDER BY object_type;
```

...

- 01 Tablas
- 02 Vistas
- 03 Sinónimos
- 04 Índices
- 05 Secuencias

SQL

USUARIOS

SYS
SYSTEM

OTROS

...

REGIONS
HR.REGIONS
SYSTEM.REGIONS

SQL

ESQUEMAS

...

PRIVILEGIOS

SQL

El nombre puede tener de 1 a 30 caracteres (excepto los nombres de enlaces de la base de datos que pueden tener hasta 128 Caracteres).

Las palabras reservadas (como SELECT) no pueden utilizarse como nombres de objetos.

Todos los nombres deben comenzar con una letra de la A a la Z.

Los caracteres de un nombre sólo pueden ser letras, números, un guion bajo (_), el signo del dólar (\$), o el símbolo de almohadilla (#).

Las letras minúsculas se convertirán en mayúsculas.

SQL

```
CREATE TABLE lower (c1 date);  
Table created.
```

...

```
CREATE TABLE "lower" (col1 varchar2(2));  
Table created.
```

...

```
SELECT table_name  
FROM user_tables  
WHERE lower(table_name) = 'lower';
```

SQL

ESPACIOS DE NOMBRES

Tablas

Vistas

Secuencias

Sinónimos privados

SQL

TIPOS DE DATOS

Alfanuméricos

VARCHAR2: Longitud variable.

NVARCHAR2: VARCHAR2 - Unicode.

CHAR: Longitud fija. Rellenado con espacios.
De 1 byte a 2000 bytes.

SQL

TIPOS DE DATOS

Binarios:

RAW: Binario de longitud variable.

1 byte hasta 4000 bytes si MAX_STRING_SIZE=STANDARD

32767 bytes si MAX_STRING_SIZE=EXTENDED.

SQL

TIPOS DE DATOS

Numéricos:

- NUMBER: Datos numéricos. Precisión de 1 a 38.
Escala de -84 a 127.
- FLOAT: Tipo de datos ANSI. Coma flotante.
Precisión de 126 binarios
(o 38 decimales).
Oracle también proporciona BINARY_FLOAT y
BINARY_DOUBLE como alternativas.
- INTEGER: Equivalente a NUMBER, con escala cero.

SQL

TIPOS DE DATOS

Fecha y Hora (todos de longitud fija)

DATE: Longitud es 0 si está vacía, o 7 bytes en caso contrario.

Rango 01/01/4712 a.C. al 31/12/9999 d.C.

TIMESTAMP: Longitud cero si está vacía, o hasta 11 bytes,

Similar a DATE, pero con una precisión de hasta 9 decimales para los segundos, 6 posiciones por defecto.

TIPOS DE DATOS

SQL

Fecha y Hora (todos de longitud fija)

TIMESTAMP WITH TIMEZONE:

Como TIMESTAMP, pero con zona horaria.
La longitud puede ser de hasta 13 bytes

TIMESTAMP WITH LOCAL TIMEZONE:

Como TIMESTAMP, pero con zona horaria normalizada.

INTERVAL YEAR TO MONTH:

Período en años y meses entre dos DATEs o TIMESTAMPS.

INTERVAL DAY TO SECOND:

Período en días y segundos entre dos DATEs o TIMESTAMPS.

SQL

TIPOS DE DATOS

Objeto grandes:

CLOB: Datos de caracteres.
Tamaño ilimitado: (4GB -1) x tamaño del bloque

NCLOB: Como CLOB, pero en Unicode.

BLOB: Como CLOB, pero con datos binarios.

BFILE: Un puntero que apunta a un archivo almacenado en el sistema operativo del servidor de base de datos.
El tamaño de los archivos está limitado a 4 GB.

DEPRECATED

LONG: Datos de caracteres en la base de datos, hasta 2 GB.

LONG RAW: Como LONG pero con datos binarios.

SQL

TIPOS DE DATOS

ROWID: Un valor codificado en base 64 que es el puntero a la ubicación de un registro en una tabla. Encriptado dentro está la dirección física exacta.

Es un tipo de datos propietario de Oracle, no visible a menos que se seleccione específicamente.

SQL

TABLAS

HEAP	Contiene registros de longitud variable en orden aleatorio.
Indexadas	Almacenan registros en orden según una clave índice.

...

Bloques indexados	Pueden desnormalizar las relaciones padre-hijo de las tablas de manera que se pueden agrupar registros de diferentes tablas.
Bloques referenciados	Fuerzan una distribución aleatoria de registros, que descompondrá cualquier orden basado en la secuencia de entrada.
Tablas particionadas	Almacenan registros en estructuras físicas separadas, las particiones, asignando registros según el valor de una columna.

SQL

TABLAS

```
CREATE TABLE [schema.]table  
  [ORGANIZATION HEAP] (  
    column datatype [DEFAULT expression]  
    [,column datatype [DEFAULT expression]]...  
  );
```

...

```
CREATE TABLE SCOTT.EMP (  
  EMPNO NUMBER(4)  
  , ENAME VARCHAR2(10)  
  , HIREDATE DATE DEFAULT TRUNC(SYSDATE)  
  , SAL NUMBER(7,2)  
  , COMM NUMBER(7,2) DEFAULT 0.03);
```

TABLAS

SQL

```
CREATE TABLE [schema.]table AS subquery;
```

...

```
CREATE TABLE employees_copy AS  
    SELECT * FROM employees;
```

...

```
CREATE TABLE emp_dept AS  
    SELECT last_name ename, department_name dname, round(sysdate – hire_date)  
service  
    FROM employees NATURAL  
        JOIN departments  
    ORDER BY dname, ename;
```

...

```
CREATE TABLE no_emps AS  
    SELECT *  
    FROM employees  
    WHERE 1=2;
```

SQL

TABLAS

```
ALTER TABLE emp ADD (job_id number);  
ALTER TABLE emp DROP COLUMN comm;  
ALTER TABLE tablename DROP UNUSED COLUMNS;
```

...

```
ALTER TABLE emp MODIFY (comm number(4,2) DEFAULT 0.05);
```

...

```
ALTER TABLE emp SET UNUSED COLUMN job_id;  
ALTER TABLE emp RENAME COLUMN hiredate TO recruited;  
ALTER TABLE emp READ ONLY;
```

SQL

TABLAS

```
DROP TABLE [schema].tablename;
```

SQL

RESTRICCIONES DE COLUMNA

UNIQUE

NOT NULL

PRIMARY KEY

FOREIGN KEY

CHECK

SQL

RESTRICCIONES DE COLUMNA

```
CREATE TABLE dept(  
    deptno NUMBER(2,0) CONSTRAINT dept_deptno_pk PRIMARYKEY  
    CONSTRAINT dept_deptno_ck CHECK  
        (deptno BETWEEN 10 AND 90),  
    dname VARCHAR2(20) CONSTRAINT dept_dname_nn NOT NULL);
```

SQL

RESTRICCIONES DE COLUMNA

```
CREATE TABLE emp(  
    empno NUMBER(4,0)      CONSTRAINT emp_empno_pk PRIMARY KEY,  
    ename VARCHAR2(20)     CONSTRAINT emp_ename_nn NOT NULL,  
    mgr NUMBER(4,0)        CONSTRAINT emp_mgr_fk  
                            REFERENCES emp (empno),  
  
    dob DATE,  
    hiredate DATE,  
    deptno NUMBER(2,0)     CONSTRAINT emp_deptno_fk  
                            REFERENCES dept(deptno)  
                            ON DELETE SET NULL,  
    email VARCHAR2(30)     CONSTRAINT emp_email_uk UNIQUE,  
  
    CONSTRAINT emp_hiredate_ck CHECK (hiredate >= dob + 365*16),  
    CONSTRAINT emp_email_ck  
        CHECK ((INSTR (email, '@') > 0) AND (INSTR (email, '.') > 0)));
```