

# SQL



## Reglas de Integridad

## Reglas de Integridad

- Las reglas de integridad obligan a cumplir las restricciones de las tablas.
- Previenen los borrados en una tabla si existen dependencias con otras tablas.
- Reglas de Integridad:
  - NOT NULL
  - UNIQUE KEY
  - PRIMARY KEY
  - FOREIGN KEY

## Reglas de Integridad

```
CREATE TABLE [schema.] table
    (column datatype [DEFAULT expr]
    [column_constraint],
    ...
    [table_constraint]);
```

```
CREATE TABLE emp (
    empno    NUMBER(4) ,
    ename    VARCHAR2(10) ,
    ...
    deptno   NUMBER(7,2) NOT NULL,
    CONSTRAINT emp_empno_pk
              PRIMARY KEY (EMPNO) );
```

## Reglas de Integridad

### – Reglas a nivel columna:

```
column [CONSTRAINT constraint_name] constraint_type,
```

### – Reglas a nivel tabla:

```
column, ...  
[CONSTRAINT constraint_name] constraint_type  
(column, ...),
```

## Reglas de Integridad: NOT NULL

La columna no permite valores nulos

EMP

EMPNO	ENAME	JOB	...	COMM	DEPTNO
7839	KING	PRESIDENT			10
7698	BLAKE	MANAGER			30
7782	CLARK	MANAGER			10
7566	JONES	MANAGER			20
...					

NOT NULL



La columna puede  
contener nulos



NOT NULL



## Reglas de Integridad: NOT NULL

Se define a nivel columna

```
SQL> CREATE TABLE emp (  
2      empno      NUMBER(4) ,  
3      ename      VARCHAR2(10) NOT NULL,  
4      job        VARCHAR2(9) ,  
5      mgr        NUMBER(4) ,  
6      hiredate   DATE ,  
7      sal        NUMBER(7,2) ,  
8      comm       NUMBER(7,2) ,  
9      deptno     NUMBER(7,2) NOT NULL);
```

## Reglas de Integridad: UNIQUE KEY

DEPT

UNIQUE key

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Insert into

50	SALES	DETROIT
60		BOSTON

Error:  
DNAME-SALES ya existe

Permitido

## Reglas de Integridad: UNIQUE KEY

Se define a nivel tabla

```
SQL> CREATE TABLE dept (  
2      deptno      NUMBER(2) ,  
3      dname       VARCHAR2(14) ,  
4      loc         VARCHAR2(13) ,  
5      CONSTRAINT dept_dname_uk UNIQUE(dname) );
```



## Reglas de Integridad: PRIMARY KEY

**DEPT** PRIMARY KEY

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Insert into

20	MARKETING	DALLAS
	FINANCE	NEW YORK

Error: DEPTNO-20 ya existe

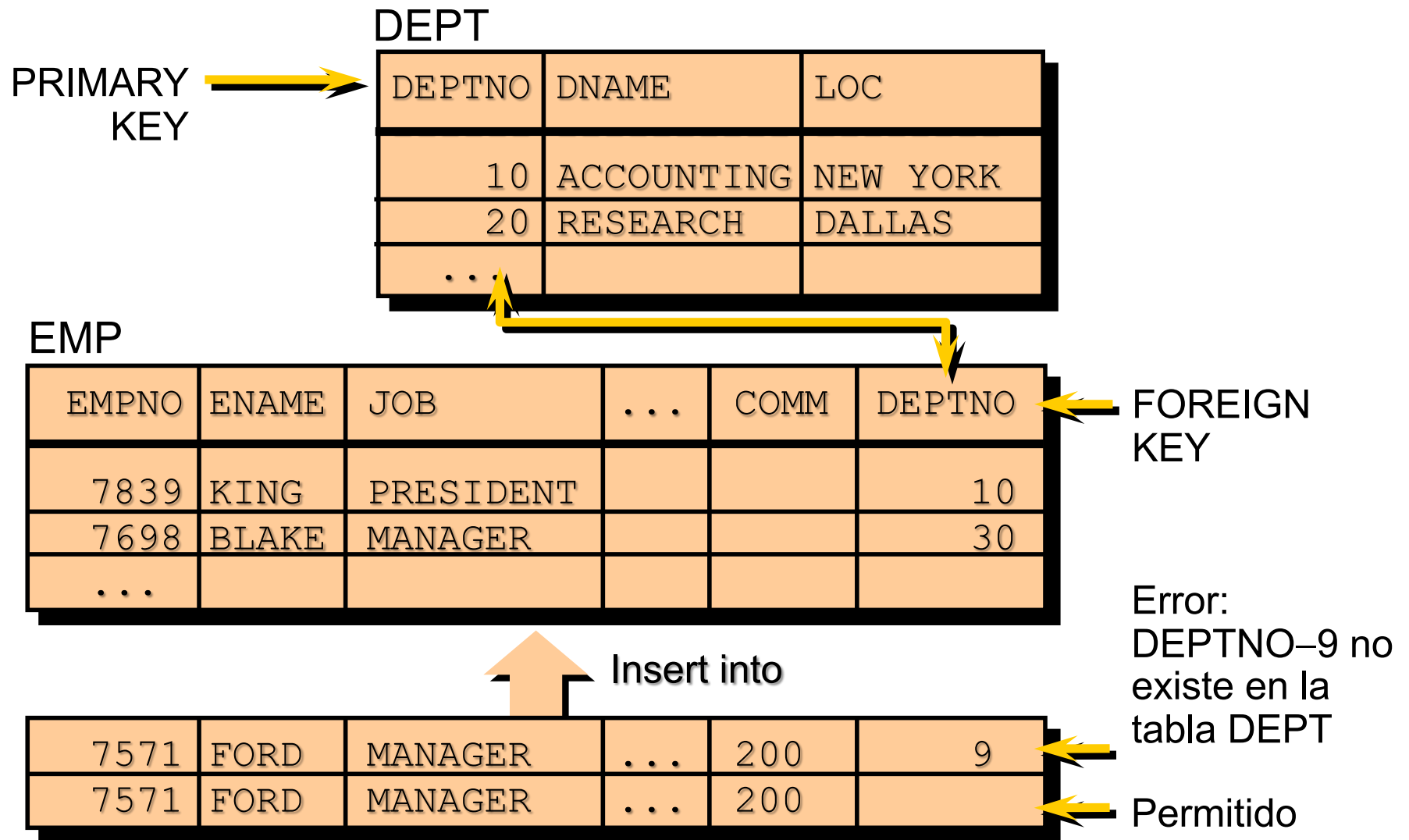
Error:DEPTNO es un valor nulo

## Reglas de Integridad: PRIMARY KEY

Se define a nivel columna

```
SQL> CREATE TABLE dept (  
2      deptno      NUMBER(2) ,  
3      dname       VARCHAR2(14) ,  
4      loc         VARCHAR2(13) ,  
5      CONSTRAINT dept_dname_uk UNIQUE (dname) ,  
6      CONSTRAINT dept_deptno_pk PRIMARY KEY(deptno)) ;
```

## Reglas de Integridad: FOREIGN KEY



## Reglas de Integridad: FOREIGN KEY

### Se define a nivel columna

```
SQL> CREATE TABLE emp (  
  2      empno      NUMBER(4) ,  
  3      ename      VARCHAR2(10) NOT NULL,  
  4      job        VARCHAR2(9) ,  
  5      mgr        NUMBER(4) ,  
  6      hiredate   DATE,  
  7      sal        NUMBER(7,2) ,  
  8      comm       NUMBER(7,2) ,  
  9      deptno     NUMBER(7,2) NOT NULL,  
 10      CONSTRAINT emp_deptno_fk FOREIGN KEY (deptno)  
 11                      REFERENCES dept (deptno));
```

## Reglas de Integridad: FOREIGN KEY

- FOREIGN KEY  
Columna con referencia a una PK en otra tabla
- REFERENCIAS  
Identifica la tabla y columna(s) de la PK
- ON DELETE CASCADE  
Permite eliminar el renglón de la llave primaria y los renglones dependientes de las tablas secundarias(FK)

## Reglas de Integridad: CHECK

**Son condiciones que se deben satisfacer.**

No está permitido:

1. Las referencias a pseudocolumnas CURRVAL, NEXTVAL, LEVEL, y ROWNUM
2. Las llamadas a SYSDATE, UID, USER
3. Funciones de usuario
4. Consultas que hacen referencia a valores de otros renglones

```
..., deptno  NUMBER(2),  
        CONSTRAINT emp_deptno_ck  
        CHECK (DEPTNO BETWEEN 10 AND 99), ...
```

## Reglas de Integridad: Anexando Reglas

```
ALTER TABLE table  
ADD [CONSTRAINT constraint] type (column);
```

Esta permitido:

- Anexar o borrar pero no modificar
- Activar o desactivar
- Adicionar la regla de NOT NULL usando la clausula MODIFY

Agregar una restricción FOREIGN KEY a la tabla EMP, el gerente debe existir como empleado tabla EMP.

```
SQL> ALTER TABLE      emp
      2  ADD CONSTRAINT  emp_mgr_fk
      3                FOREIGN KEY (mgr) REFERENCES emp (empno) ;
Table altered.
```



## Eliminando una regla

```
SQL> ALTER TABLE      emp
      2  DROP CONSTRAINT  emp_mgr_fk;
Table altered.
```

Elimina la PK de la tabla DEPT y borra la restricción de FK en la columna DEPTNO.

```
SQL> ALTER TABLE      dept
      2  DROP PRIMARY KEY CASCADE;
Table altered.
```

# Deshabilitando reglas

- Cláusula DISABLE de la sentencia ALTER TABLE desactiva la restricción de integridad.
- Opción CASCADE desactiva las restricciones que dependen de la integridad.

```
SQL> ALTER TABLE emp  
2  DISABLE CONSTRAINT emp_empno_pk CASCADE;  
Table altered.
```

# Habilitando Reglas

- La clausula ENABLE activa la regla de integridad.
- El índice de la PK se crea en automático cuándo se activa esta regla de integridad.

```
SQL> ALTER TABLE          emp  
      2  ENABLE CONSTRAINT  emp_empno_pk;  
Table altered.
```

## Reglas de Integridad: Mostrando las reglas de Integridad

En la tabla USER\_CONSTRAINTS se almacenan todas las reglas de integridad

```
SQL> SELECT constraint_name, constraint_type,  
2         search_condition  
3 FROM   user_constraints  
4 WHERE  table_name = 'EMP';
```

CONSTRAINT_NAME	C SEARCH_CONDITION
-----	- - - - -
SYS_C00674	C EMPNO IS NOT NULL
SYS_C00675	C DEPTNO IS NOT NULL
EMP_EMPNO_PK	P
...	

## Reglas de Integridad

La vista USER\_CONS\_COLUMNS muestra todas las columnas asociadas a una regla de integridad

```
SQL> SELECT  constraint_name, column_name
2  FROM      user_cons_columns
3  WHERE      table_name = 'EMP';
```

CONSTRAINT_NAME	COLUMN_NAME
-----	-----
EMP_DEPTNO_FK	DEPTNO
EMP_EMPNO_PK	EMPNO
EMP_MGR_FK	MGR
SYS_C00674	EMPNO
SYS_C00675	DEPTNO