

# Lenguaje SQL (Structured Query Language)

Prof. Sergio Sánchez.  
Primer sesión



# Modelo Relacional de Codd

El modelo relacional consiste en lo siguiente:

- Colección de objetos o relaciones.
- Conjunto de operadores para actuar en las relaciones.
- Integridad de los datos de precisión y coherencia.

# SQL Structured query language

- Lenguaje estándar ANSI de BDR.
- Eficiente, fácil de aprender y usar.
- Completamente funcional(puede definir, recuperar y manipular datos en las tablas).

```
SELECT department_name  
FROM departments;
```

*department\_name*

Contabilidad

TI

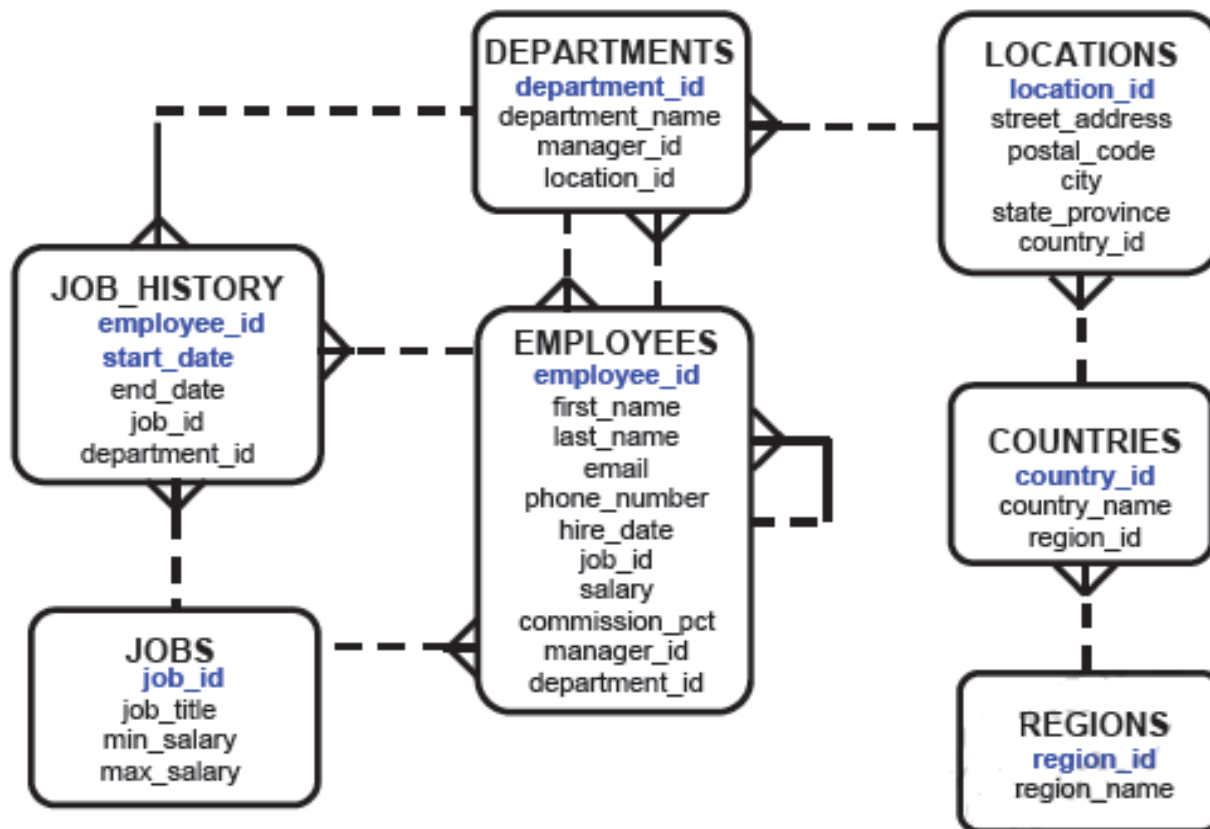
Auditoria

Servicio al cliente

Tesorería

# Esquema de ejemplo

## Human Resources (HR) Schema



# Data definition language (DDL)

- CREATE
- ALTER
- DROP
- RENAME
- TRUNCATE
- COMMENT

# Objetos de la Base de Datos

## Data Defination Language (DDL)

Objeto	Descripción
TABLE	Unidad básica de almacenamiento, compuesta por registros
VIEW	Representación lógica de sub consultas de datos de una o mas tablas
SEQUENCE	Generador de valores numéricos consecutivos
INDEX	Mejora el rendimiento de algunas consultas
SYNONYM	Da nombre alternativo algún objeto

# CREATE table

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

```
CREATE TABLE dept  
      (deptno      NUMBER(2),  
       dname       VARCHAR2(14),  
       loc         VARCHAR2(13),  
       create_date DATE DEFAULT SYSDATE);
```

```
CREATE TABLE succeeded.
```

```
CREATE TABLE table  
      [(column, column...)]  
AS subquery;
```

# Tipo de datos

TIPO DE DATO	DESCRIPCION
VARCHAR2(size)	Carácter de variable tamaño
CHAR(size)	Carácter de tamaño fijo
NUMBER(p,s)	Numérico de tamaño variable
DATE	Valor de fecha y hora
LONG	Carácter de variable tamaño(mas de 2GB)
CLOB	Carácter (mas de 4GB)
RAW and LONG RAW	Datos binarios
BLOB	Datos binarios (mas de 4GB)
BFILE	Datos binarios en archivos externos(mas de 4GB)
ROWID	Sistema de números basados en 64-bit representado una dirección única de registro en la tabla
TIMESTAMP	Fecha con segundos
INTERVAL YEAR TO MONTH	Intervalos de años y meses
INTERVAL DAY TO SECOND	Intervalos de días, horas, minutos y segundos



# CONSTRAINT

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

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

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

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

# Alter table

- Agregar columnas

```
ALTER TABLE table
ADD          (column datatype [DEFAULT expr]
             [, column datatype]...);
```

- Modificar columnas

```
ALTER TABLE table
MODIFY       (column datatype [DEFAULT expr]
             [, column datatype]...);
```

- Borrar columnas

```
ALTER TABLE table
DROP (column [, column] ...);
```

# Alter Table

```
ALTER TABLE dept80  
ADD          (job_id VARCHAR2(9));
```

```
ALTER TABLE dept80 succeeded.
```

```
ALTER TABLE dept80  
MODIFY      (last_name VARCHAR2(30));
```

```
ALTER TABLE dept80 succeeded.
```

```
ALTER TABLE  dept80  
DROP COLUMN   job_id;
```

```
ALTER TABLE dept80 succeeded.
```

# CONSTRAINT

```
CREATE TABLE employees(  
    employee_id NUMBER(6)  
    CONSTRAINT emp_emp_id_pk PRIMARY KEY,  
    first_name VARCHAR2(20),  
    ...);
```

1

```
CREATE TABLE employees(  
    employee_id NUMBER(6),  
    first_name VARCHAR2(20),  
    ...  
    job_id VARCHAR2(10) NOT NULL,  
    CONSTRAINT emp_emp_id_pk  
    PRIMARY KEY (EMPLOYEE_ID));
```

2

```
ALTER TABLE employees  
    ADD CONSTRAINT pk_emp_id PRIMARY KEY (EMPLOYEE_ID);
```

# CONSTRAINT

```
CREATE TABLE employees(  
    employee_id      NUMBER(6),  
    last_name        VARCHAR2(25) NOT NULL,  
    email            VARCHAR2(25),  
    salary            NUMBER(8,2),  
    commission_pct   NUMBER(2,2),  
    hire_date        DATE NOT NULL,  
    ...  
    CONSTRAINT emp_email_uk UNIQUE(email));
```

```
CREATE TABLE employees(  
    employee_id      NUMBER(6),  
    last_name        VARCHAR2(25) NOT NULL,  
    email            VARCHAR2(25),  
    salary            NUMBER(8,2),  
    commission_pct   NUMBER(2,2),  
    hire_date        DATE NOT NULL,  
    ...  
    department_id    NUMBER(4),  
    CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)  
        REFERENCES departments(department_id),  
    CONSTRAINT emp_email_uk UNIQUE(email));
```

# Constraint

```
ALTER TABLE <table_name>  
ADD [CONSTRAINT <constraint_name>]  
type (<column_name>);
```

```
ALTER TABLE emp2  
MODIFY employee_id PRIMARY KEY;
```

ALTER TABLE emp2 succeeded.

```
ALTER TABLE emp2  
ADD CONSTRAINT emp_mgr_fk  
FOREIGN KEY(manager_id)  
REFERENCES emp2(employee_id);
```

ALTER TABLE succeeded.

# Constraint

```
ALTER TABLE emp2 ADD CONSTRAINT emp_dt_fk  
FOREIGN KEY (Department_id)  
REFERENCES departments(department_id) ON DELETE CASCADE;
```

```
ALTER TABLE Emp2 succeeded.
```

```
ALTER TABLE emp2 ADD CONSTRAINT emp_dt_fk  
FOREIGN KEY (Department_id)  
REFERENCES departments(department_id) ON DELETE SET NULL;
```

```
ALTER TABLE Emp2 succeeded.
```

# Constraint

```
ALTER TABLE emp2  
DROP CONSTRAINT emp_mgr_fk;
```

```
ALTER TABLE Emp2 succeeded.
```

```
ALTER TABLE dept2  
DROP PRIMARY KEY CASCADE;
```

```
ALTER TABLE dept2 succeeded.
```

```
ALTER TABLE emp2  
DISABLE CONSTRAINT emp_dt_fk;
```

```
ALTER TABLE Emp2 succeeded.
```

```
ALTER TABLE emp2  
DROP COLUMN employee_id CASCADE CONSTRAINTS;
```

```
ALTER TABLE Emp2 succeeded.
```



# Renombrar columnas

a

```
ALTER TABLE marketing RENAME COLUMN team_id  
TO id;
```

```
ALTER TABLE marketing succeeded.
```

b

```
ALTER TABLE marketing RENAME CONSTRAINT mktg_pk  
TO new_mktg_pk;
```

```
ALTER TABLE marketing succeeded.
```

# Crear un índice con create table

```
CREATE TABLE NEW_EMP  
(employee_id NUMBER(6)  
PRIMARY KEY USING INDEX  
(CREATE INDEX emp_id_idx ON  
NEW_EMP(employee_id)),  
first_name VARCHAR2(20),  
last_name VARCHAR2(25));
```

CREATE TABLE succeeded.

Crear un índice simple

```
CREATE INDEX upper_dept_name_idx  
ON dept2(UPPER(department_name));
```

CREATE INDEX succeeded.

# Borrar un índice

```
DROP INDEX index;
```

```
DROP INDEX upper_dept_name_idx;
```

```
DROP INDEX upper_dept_name_idx succeeded.
```

# DROP table

```
DROP TABLE dept80;
```

```
DROP TABLE dept80 succeeded.
```

# CREATE VIEW

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view
  [(alias[, alias]...)]
  AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY [CONSTRAINT constraint]];
```

```
CREATE VIEW empvu80
  AS SELECT employee_id, last_name, salary
  FROM employees
  WHERE department_id = 80;
```

CREATE VIEW succeeded.

```
CREATE VIEW salvu50
  AS SELECT employee_id ID_NUMBER, last_name NAME,
           salary*12 ANN_SALARY
  FROM employees
  WHERE department_id = 50;
```

CREATE VIEW succeeded.

```
SELECT *
FROM salvu50;
```

	ID_NUMBER	NAME	ANN_SALARY
1	124	Mourgos	69600
2	141	Rajs	42000
3	142	Davies	37200
4	143	Matos	31200
5	144	Vargas	30000

# DROP VIEW

```
DROP VIEW view;
```

```
DROP VIEW empvu80;
```

```
DROP VIEW empvu80 succeeded.
```

# SEQUENCE

```
CREATE SEQUENCE sequence
  [INCREMENT BY n]
  [START WITH n]
  [{MAXVALUE n | NOMAXVALUE}]
  [{MINVALUE n | NOMINVALUE}]
  [{CYCLE | NOCYCLE}]
  [{CACHE n | NOCACHE}];
```

```
CREATE SEQUENCE dept_deptid_seq
  INCREMENT BY 10
  START WITH 120
  MAXVALUE 9999
  NOCACHE
  NOCYCLE;
```

```
CREATE SEQUENCE succeeded.
```

# SEQUENCE

```
INSERT INTO departments(department_id,  
                        department_name, location_id)  
VALUES                (dept_deptid_seq.NEXTVAL,  
                      'Support', 2500);
```

1 rows inserted

```
ALTER SEQUENCE dept_deptid_seq  
          INCREMENT BY 20  
          MAXVALUE 999999  
          NOCACHE  
          NOCYCLE;
```

ALTER SEQUENCE dept\_deptid\_seq succeeded.



# INDEX

```
CREATE [UNIQUE] [BITMAP] INDEX index  
ON table (column[, column]...);
```

```
CREATE INDEX emp_last_name_idx  
ON          employees(last_name);
```

```
CREATE INDEX succeeded.
```

```
DROP INDEX index;
```

# SYNONYM

```
CREATE [PUBLIC] SYNONYM synonym  
FOR    object;
```

```
CREATE SYNONYM d_sum  
FOR dept_sum_vu;
```

```
CREATE SYNONYM succeeded.
```



GRACIAS