

*Capacitación ORACLE*

# Lenguaje SQL y Administración Básica

Gestión de Datos

Docentes: Ing. Juan Zaffaroni  
Ing. María Cristina Chahin

## Capacitación ORACLE. Lenguaje SQL y Administración Básica

### AGENDA GENERAL MÓDULO 1: Introducción y SQL Básico

#### ❖ Día 1

##### 👉 1- Introducción a Modelo Relacional / BD Relacional / Oracle

- Ciclo de vida del desarrollo
- Almacenamiento de Datos
- Sistema de Base de Datos
- Modelo de Relación entre Entidades (DER)
- Modelo Relacional
- RDBMS
- SQL
- PL/SQL
- SQL\*Plus
- Tipos de Usuario
- Tareas del DBA

#### ❖ Día 2

##### 👉 2- Consultas básicas: SELECT

- Consultas sobre una tabla. Alias de columna.
- Selección de filas
- Ordenamiento
- Funciones de una fila

## 1- Introducción a Modelo Relacional / BD Relacional / Oracle

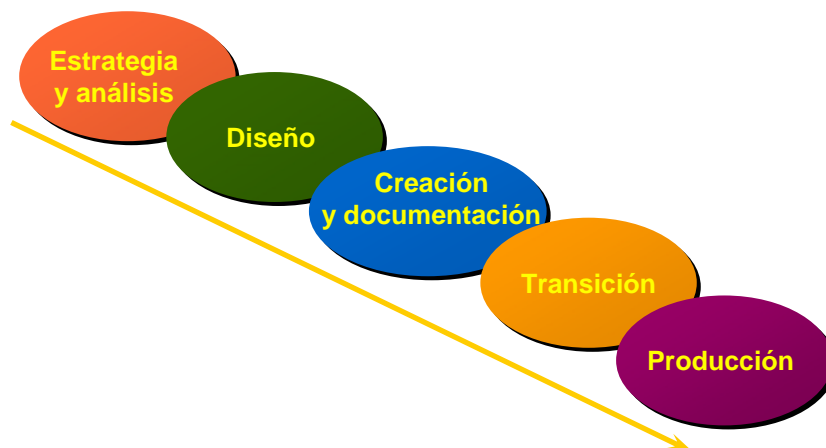
- ✓Ciclo de vida del desarrollo
- ✓Almacenamiento de Datos
- ✓Sistema de Base de Datos
  - Componentes
  - Enfoques
  - Ventajas
- ✓Modelo de Relación entre Entidades (DER)
  - Componentes
  - Convenciones DER
- ✓Modelo Relacional
  - Concepto
  - Componentes
  - Reglas de integridad
  - Operaciones
- ✓Terminología
- ✓RDBMS
- ✓SQL
  - Lenguaje
  - Sentencias
- ✓RDBMS Oracle
- ✓PL/SQL
- ✓SQL\*Plus
- ✓Tipos de Usuario
- ✓Tareas del DBA

Capacitación Oracle SQL

3

## 1- Introducción a Modelo Relacional / BD Relacional / Oracle

### Ciclo de Vida del Desarrollo

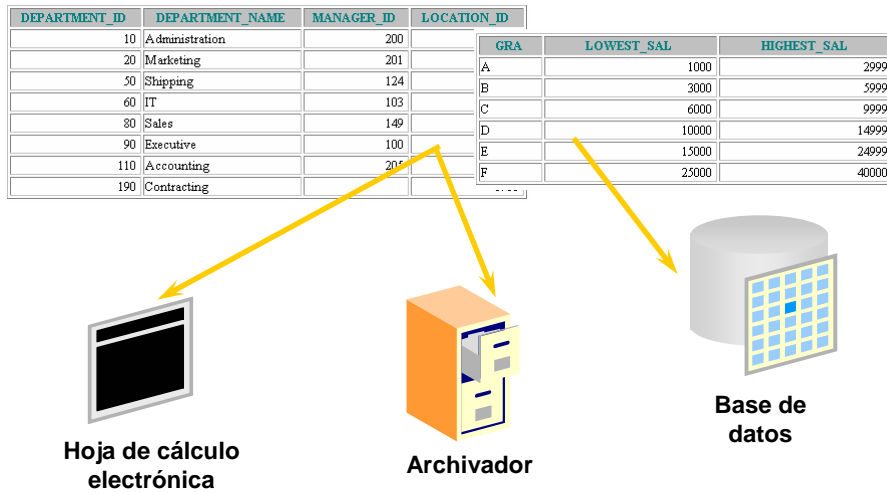


Capacitación Oracle SQL

4

## 1- Introducción a Modelo Relacional / BD Relacional / Oracle

### Almacenamiento de Datos en Diferentes Soportes

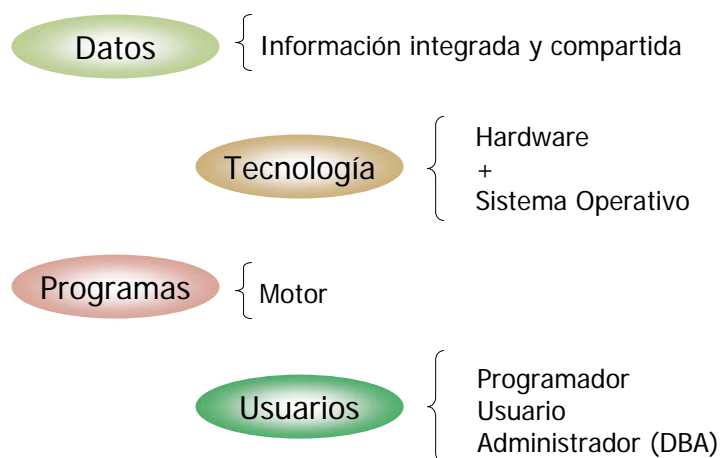


Capacitación Oracle SQL

5

## 1- Introducción a Modelo Relacional / BD Relacional / Oracle

### Componentes de un Sistema de Base de Datos

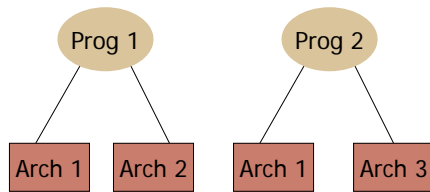


Capacitación Oracle SQL

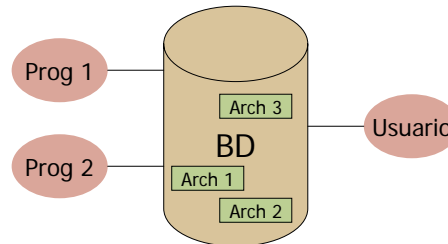
6

## Enfoques de Base de Datos

Orientado a la aplicación



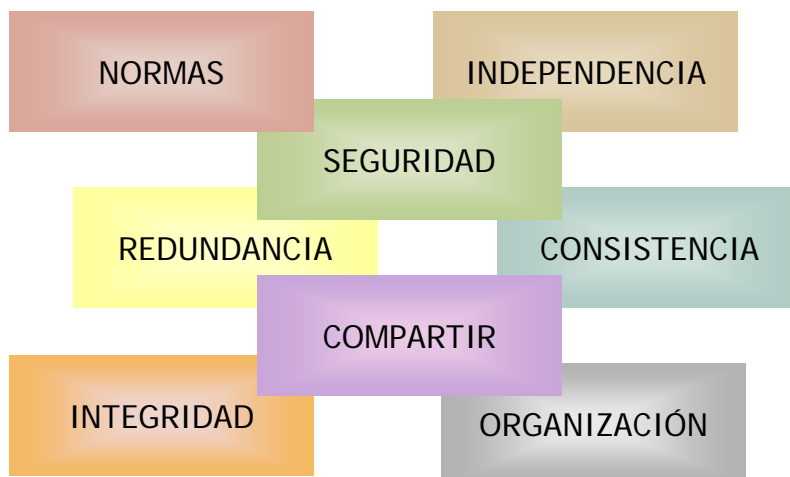
Orientado a los datos



Capacitación Oracle SQL

7

## Ventajas de un Sistema de Base de Datos



Capacitación Oracle SQL

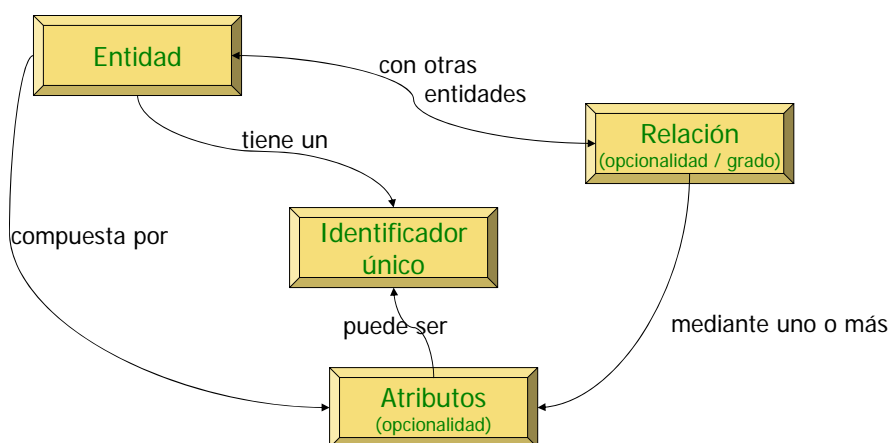
8

## Modelo de Relación entre Entidades (DER)

*Ilustración de diversas entidades en un negocio, y la relación entre ellas*

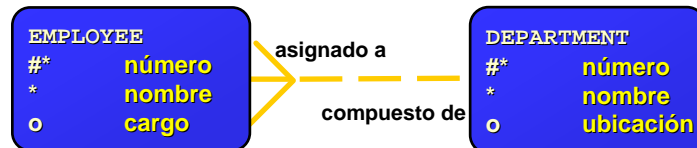
- ➡ Documentar
- ➡ Fácil comprensión
- ➡ Marco para integrar aplicaciones

## Componentes de un DER



## Convenciones DER (I)

- Cree un diagrama de relación entre entidades a partir de descripciones o especificaciones de negocios



- Supuesto
  - "... Asigne uno o más empleados a un departamento ..."
  - "... Algunos departamentos aún no tienen empleados asignados. ..."

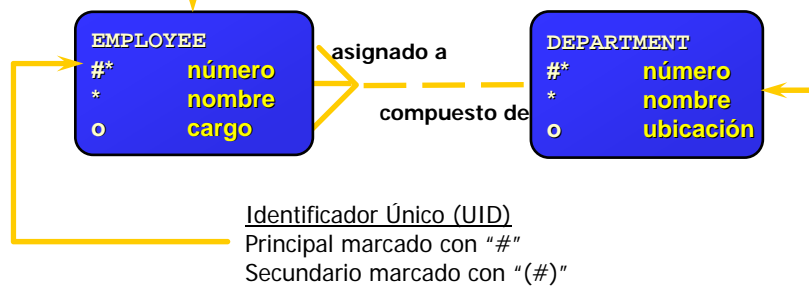
## Convenciones DER (II)

### Entidad

Recuadro flexible  
Nombre singular y único  
Mayúsculas  
Sinónimo entre paréntesis

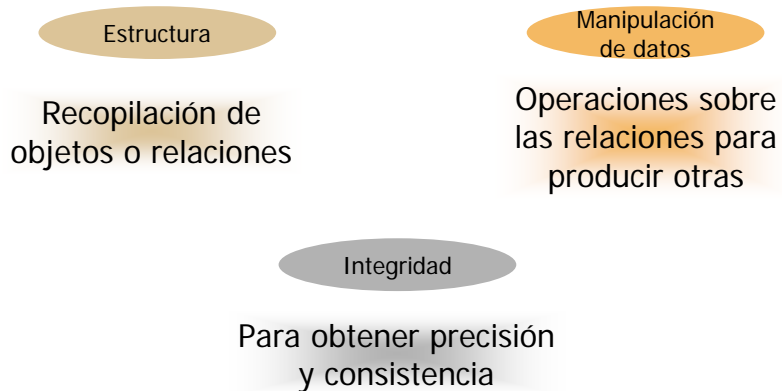
### Atributo

Nombre en singular  
Minúsculas  
Obligatorio marcado con "\*"   
Opcional marcado con "o"

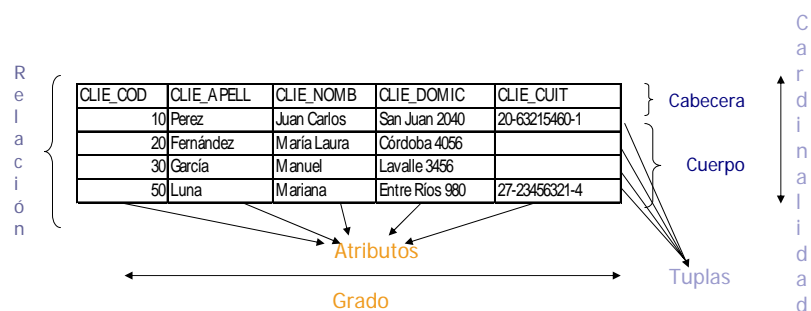


## Modelo Relacional: Concepto

- Propuesto por el Dr. E.F. Codd en 1970, consta de:



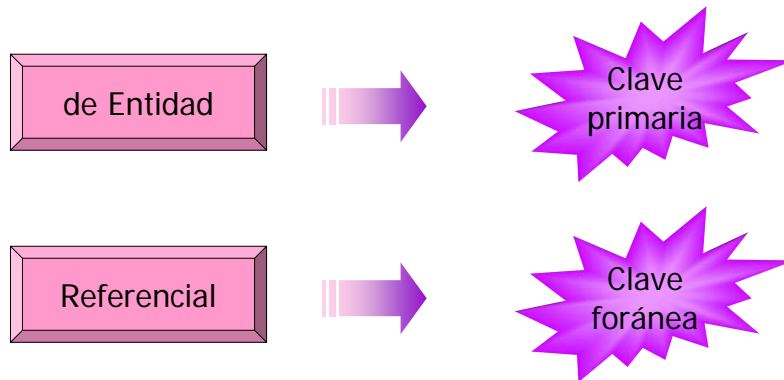
## Modelo Relacional: Componentes



**TABLA:** Representación de una Relación.

**Base de datos Relacional:** Colección de relaciones normalizadas de diversos grados, que varía con el tiempo.

## Modelo Relacional: Reglas de Integridad

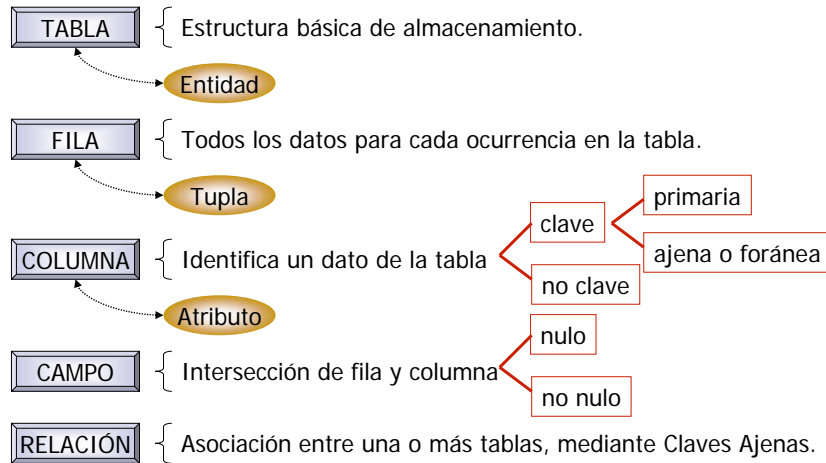


## Modelo Relacional: Operaciones

- Proyección
- Selección
- Producto Cartesiano
- Unión
- Intersección
- Diferencia
- Join (o Reunión o Junta)



## Base de Datos Relacional: Terminología



## BD Relacional: RDBMS

### Sistema de Gestión de Bases de Datos Relacionales

Procesos

Estructuras de memoria

### Funciones

- \* Almacenamiento y recuperación de información relacionada (organización física de los datos)
- \* Manejo de la concurrencia de acceso
- \* Seguridad
- \* Diccionario de datos
- \* Conectividad
- \* Integridad / Consistencia
- \* Optimización del acceso
- \* Recuperación de fallos (backup / recovery)
- \* Acceso a los datos (SQL, PL/SQL, Java)
- \* Manejo de transacciones
- \* Administración y chequeo de recursos

## BD Relacional: SQL

Lenguaje Estructurado de Consulta



Acceso a la base ↔ Manipulación de datos

- ✓ Eficiente
- ✓ Sencillo
- ✓ Completo

## BD Relacional: Sentencias SQL

SELECT } Recuperar datos

INSERT  
UPDATE  
DELETE  
MERGE } DML

COMMIT  
ROLLBACK  
SAVEPOINT } Transacciones lógicas

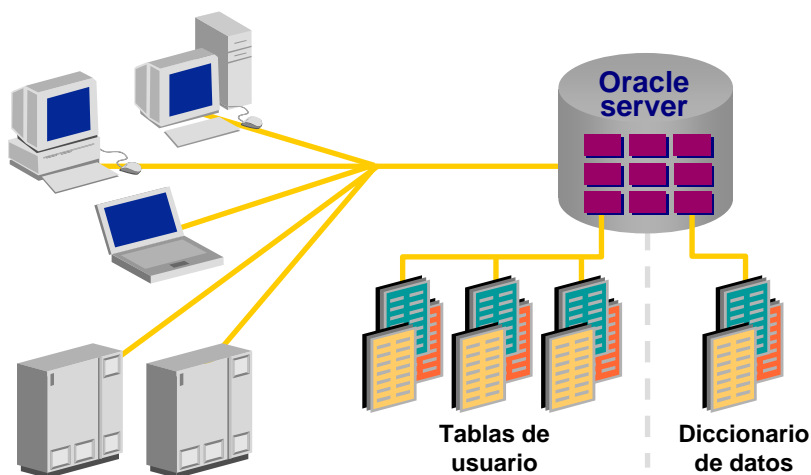
CREATE  
ALTER  
DROP  
RENAME  
TRUNCATE } DDL

GRANT  
REVOKE } DCL

## Sistema de Gestión de Bases de Datos Relacionales y Relacionales de Objetos

- Modelo relacional y modelo relacional de objetos
- Tipos de dato y objetos definidos por el usuario
- Totalmente compatible con base de datos relacional
- Soporte de multimedia y objetos grandes
- Funciones de servidor de base de datos de alta calidad

## Sistema de Gestión de BD Relacional



## PL/SQL

### Lenguaje procedural, extensión a SQL.

- Almacenado en forma centralizada en la Base de Datos
- Reduce el tráfico de red
- Acceso a los datos
- Operaciones complejas

Unidades de Programas	{	Procedimientos Funciones Paquetes Triggers Bloques autónomos
-----------------------------	---	--

Capacitación Oracle SQL

23

## SQL \*Plus

### Herramienta para ejecución de sentencias SQL y bloques PL/SQL

- Ingresar, editar, almacenar y recuperar SQL y PL/SQL
- Formatear, calcular, almacenar e imprimir resultados de consultas
- Listar definiciones de tablas
- Administrar la Base de Datos

Capacitación Oracle SQL

24

## Tipos de Usuarios

- ❖ Administradores de Base de Datos
- ❖ Encargados de Seguridad
- ❖ Administradores de red
- ❖ Desarrolladores de aplicaciones
- ❖ Administradores de aplicaciones
- ❖ Usuarios de la Base de Datos

## Tareas del DBA

- ✓ Evaluar el Hardware del servidor de Base de Datos
- ✓ Garantizar la compatibilidad con el licenciamiento Oracle
- ✓ Instalar / actualizar el Software de Oracle (servidor y herramientas de aplicaciones)
- ✓ Planificar la Base de Datos
  - Estructura lógica
  - Diseño general
  - Estrategia de Backup
- ✓ Crear y abrir la Base de Datos
- ✓ Resguardar la Base de Datos (inicial / cambios)
- ✓ Enrolar usuarios y mantener la seguridad del sistema
- ✓ Implementar el diseño de la base de datos
  - Crear estructuras de almacenamiento
  - Crear objetos
  - Modificar la estructura de la base
- ✓ Optimizar el rendimiento de la base

## 2- Consultas básicas: SELECT

- ✓ Consultas sobre una tabla. Alias de columna.
- ✓ Selección de filas
- ✓ Ordenamiento
- ✓ Funciones de una fila

## 2- Consultas básicas: SELECT

### Introducción

- El lenguaje SQL provee la sentencia **SELECT** para extraer información de una Base de Datos.
- Consultas básicas: **SELECT** explica la forma de realizar una consulta simple para obtener información, con las siguientes características:
  - Consulta a una sola tabla
  - Uso de alias de columnas
  - Uso de operadores
  - Tratamiento de duplicados
  - Restricción de filas
  - Ordenamiento del resultado de consultas
  - Uso de funciones de una sola fila

## 2- Consultas básicas: SELECT – Una tabla. Alias de columna.

### Objetivo

- Recuperar información de una sola tabla de la Base de Datos.
  - Correspondiente a
    - *Todas las columnas* de la tabla.
    - *Una o más columnas* específicas de la tabla.
    - *Una o más expresiones*
  - Las columnas recuperadas pueden o no tomar valor en cada una de las filas
- Asignar un nombre distinto a alguna de las columnas
- Concatenar valores de columna
- Mostrar una sola vez los valores duplicados

### Ejemplos

- Obtener de la tabla EMPLOYEES un listado de todos los empleados de la empresa. Mostrar las columnas LAST\_NAME y FIRST\_NAME como "<LAST\_NAME>, <FIRST\_NAME>" denominada "EMPLEADO". Incrementar el salario en 200.
- Mostrar únicamente los nombres, asegurándose de que si hay duplicados, no se muestren.

Capacitación Oracle SQL

29

## 2- Consultas básicas: SELECT – Una tabla. Alias de columna.

### Sintaxis

```
SELECT * | {[DISTINCT] columna | expresión [alias], ...}  
FROM      tabla;
```

Operadores aritméticos: +, -, \*, /

Operador de concatenación: ||

Capacitación Oracle SQL

30

### Aplicación

- ```
SELECT employee_id,  
       last_name || ', ' || first_name AS empleado,  
       email,  
       phone_number,  
       hire_date,  
       job_id,  
       salary + 200,  
       commission_pct,  
       manager_id,  
       department_id  
FROM   employees;
```
- ```
SELECT DISTINCT last_name, first_name  
FROM   employees;
```

### Objetivo

- Recuperar información de [una sola tabla](#) de la Base de Datos.
  - Correspondiente a *algunas filas* de la tabla (según cierto [criterio](#)).

### Ejemplo

- Obtener de la tabla EMPLOYEES un listado con código de empleado, apellido, nombre, código de puesto, código de departamento y fecha de contratación de todos los empleados de la empresa que fueron contratados durante el 2007, pertenecientes a los departamentos 10, 20 ó 90, o bien aquellos con código de puesto (JOB\_ID) comenzado en "AD".



## Sintaxis

```
SELECT * | {[DISTINCT] columna | expresión [alias], ...}  
FROM   tabla  
[WHERE condición(es)];
```

Operadores de comparación: =, >, >=, <, <=, <>, !=

Otros: BETWEEN ... AND ..., IN, LIKE, IS NULL

Condiciones lógicas: NOT, AND, OR

## Aplicación

```
➤ SELECT employee_id, last_name, first_name,  
        job_id, department_id, hire_date  
FROM   employees  
WHERE  (TO_CHAR(hire_date, 'yyyy') = '2007'  
        AND department_id IN (10, 20, 90))  
OR     job_id LIKE 'AD%';
```

## 2- Consultas básicas: SELECT – Ordenamiento

### Objetivo

- Recuperar información de una sola tabla de la Base de Datos.
  - Ordenar las filas resultante en base a *una o más columnas* de la tabla.

### Ejemplo

- Obtener de la tabla EMPLOYEES un listado con código de empleado, apellido, nombre, código de puesto, salario multiplicado por 1,1 (newsal), código de departamento y fecha de contratación de todos los empleados de la empresa que fueron contratados durante el 2007, pertenecientes a los departamentos 10, 20 ó 90, ordenando por apellido, nombre, newsal y fecha de contratación (esta última, descendente).

## 2- Consultas básicas: SELECT – Ordenamiento

### Sintaxis

```
SELECT * | {[DISTINCT] columna | expresión [alias], ...}  
FROM   tabla  
[WHERE condición(es)]  
[ORDER BY {columna|expresión|alias} [ASC|DESC]];
```

### Aplicación

```
➤ SELECT  employee_id, last_name, first_name, job_id,
          salary * 1.1 newsal, department_id, hire_date
FROM      employees
WHERE     TO_CHAR(hire_date, 'yyyy') = '2007'
AND       department_id IN (10, 20, 90)
ORDER BY  last_name, first_name, newsal, hire_date DESC;
```

### Objetivo

- Utilizar [funciones de una sola fila](#) en:
  - Selección de columnas *en cláusula SELECT*
  - Restricción de filas *en cláusula WHERE*
- Diferenciar funciones de:
  - *Carácter*
  - *Número*
  - *Fecha*
  - *Conversión*
- [Anidar](#) funciones de una sola fila

### Ejemplo

- Obtener de la tabla EMPLOYEES un listado con código de empleado, apellido (en mayúsculas), longitud en caracteres del email, nombre, código de puesto, código de departamento y año y mes de contratación, cantidad de meses de antigüedad, y ind\_comisión, que indica "SI" o "NO" según si el porcentaje de comisión tiene un valor o es nulo, de todos los empleados de la empresa, pertenecientes a los departamentos 10, 20 ó 90.

## 2- Consultas básicas: SELECT – Funciones de una fila

### Sintaxis

```
SELECT * | {[DISTINCT] columna | expresión [alias], ...}  
FROM   tabla  
[WHERE condición(es)]  
[ORDER BY {columna|expresión|alias} [ASC|DESC]];
```

Funciones de caracteres: lower, upper, initcap, concat, substr, length, instr, lpad, rpad, trim

Funciones numéricas: round, trunc, mod, sign, ceil, floor

Funciones de fecha: sysdate, add\_months, last\_day, round, trunc

Funciones de conversión: to\_number, to\_date, to\_char

Funciones generales: nvl, nvl2, nullif, coalesce

Expresiones condicionales: case, decode

## 2- Consultas básicas: SELECT – Funciones de una fila

### Aplicación

```
➤ SELECT employee_id, upper(last_name), length(email), first_name,  
        job_id, department_id, to_char(hire_date, 'yyyy/mm'),  
        (sysdate-hire_date)/30,  
        nvl2(commission_pct, 'SI', 'NO') ind_comision  
FROM   employees  
WHERE  department_id IN (10, 20, 90);
```

### Resumen

- Mediante la sentencia **SELECT** podemos escribir una consulta simple que permita:

- Recuperar información de una sola tabla
- Renombrar encabezados de columna mediante el uso de *alias de columnas*
- Flexibilizar y potenciar la recuperación, filtro y conversión de datos mediante el uso de *operadores*
- Tratar datos *duplicados*
- Restringir filas del resultado mediante *condiciones*
- Establecer un *criterio de ordenamiento* para el resultado, basado en una o más columnas
- Flexibilizar y potenciar la recuperación de datos mediante el uso de *funciones de una sola fila*