

Bases de Datos I

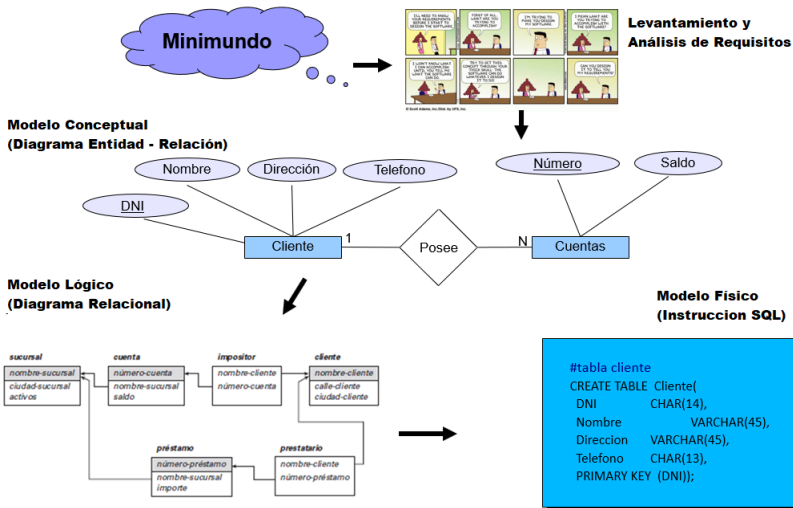
Algebra Relacional



Índice

- 1 Fases del Proyecto de BD
- 2 Lenguaje de Manipulación de Datos en BDR
- 3 Algebra Relacional
- 4 Cálculo Relacional
- 5 SQL - SELECT - DRL

Fases del Proyecto de BD



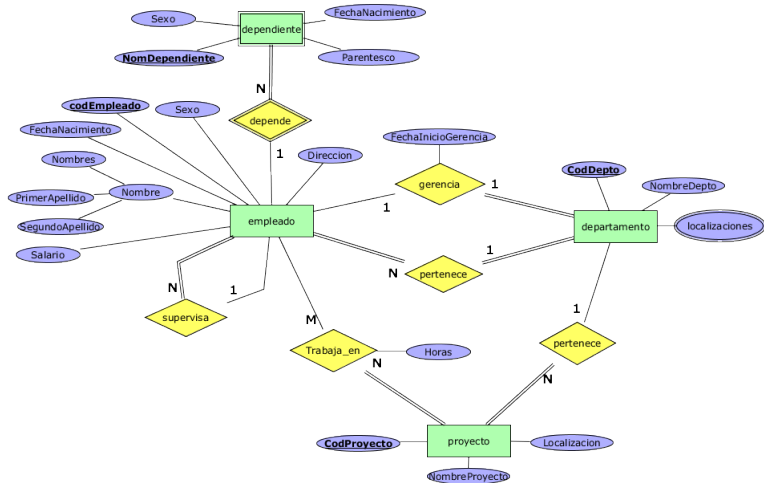
Levantamiento y Análisis de Requisitos

- Una empresa desea mantener control de sus empleados, departamento y proyectos.
- De cada empleado se almacena su código, nombre (nombres, primer apellido y segundo apellido), fecha de nacimiento, dirección, sexo salario.
- Cada empleado es asignado a un departamento.
- Un empleado es supervisado por otro empleado.
- De cada departamento se almacena su código, nombre y localizaciones. Además el código del gerente y su fecha de inicio de gerencia.

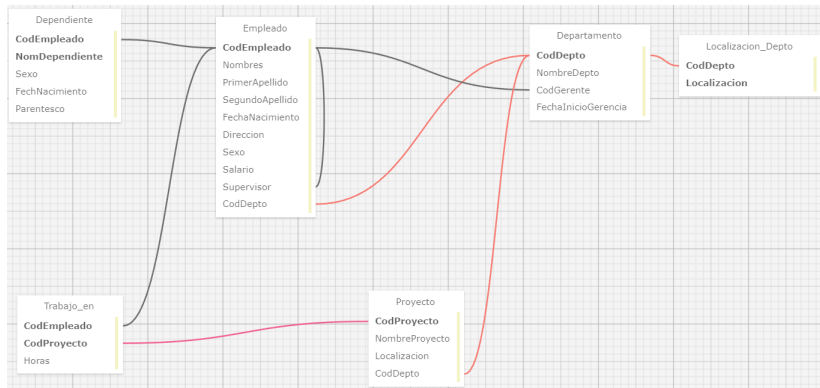
Levantamiento y Análisis de Requisitos

- Un empleado puede trabajar en varios proyectos. Es importante almacenar la horas que cada empleado dedica a cada proyecto.
- En un proyecto trabajar un conjunto de empleados de diferentes departamentos, pero el proyecto pertenece a un departamento. Del proyecto se almacena su nombre del proyecto, localización y el departamento al que pertenece.
- Por último, para asignar el seguro social a los dependientes de cada empleado, se almacena, para cada dependiente se almacena su nombre, sexo, fecha de nacimiento y parentesco.

Modelo Conceptual - Diagrama Entidad Relación



Modelo Lógico - Diagrama Relacional



Modelo Físico - Creación Base de Datos

```
CREATE DATABASE IF NOT EXISTS empresa;
```

```
USE empresa;
```

```
DROP TABLE IF EXISTS Empleado;
```

```
CREATE TABLE Empleado (  
    CodEmpleado INTEGER,  
    Nombres VARCHAR(30),  
    PrimerApellido VARCHAR(30),  
    SegundoApellido VARCHAR(30),  
    Direccion VARCHAR(100),  
    FechaNacimiento DATE,  
    Sexo CHAR(1),  
    Salario DECIMAL(12,2),  
    Supervisor INTEGER,  
    CodDepto INTEGER,  
    PRIMARY KEY (CodEmpleado)  
);
```


Modelo Físico - Creación Base de Datos

```
DROP TABLE IF EXISTS Departamento;
```

```
CREATE TABLE Departamento (  
    CodDepto INTEGER,  
    NombreDepto VARCHAR(30),  
    CodGerente INTEGER,  
    FechaInicioGerencia DATE,  
    PRIMARY KEY (CodDepto)  
);
```

```
DROP TABLE IF EXISTS Localizacion_Depto;
```

```
CREATE TABLE Localizacion_Depto (  
    CodDepto INTEGER,  
    Localizacion VARCHAR(30),  
    PRIMARY KEY (CodDepto, Localizacion)  
);
```

Modelo Físico - Creación Base de Datos

```
DROP TABLE IF EXISTS Proyecto;
```

```
CREATE TABLE Proyecto (  
  CodProyecto INTEGER,  
  NombreProyecto VARCHAR(30),  
  Localizacion VARCHAR(30),  
  CodDepto INTEGER,  
  PRIMARY KEY (CodProyecto)  
);
```

```
DROP TABLE IF EXISTS Trabaja_en;
```

```
CREATE TABLE Trabaja_en (  
  CodEmpleado INTEGER,  
  CodProyecto INTEGER,  
  Horas INTEGER,  
  PRIMARY KEY (CodEmpleado, CodProyecto)  
);
```

Modelo Físico - Creación Base de Datos

```
DROP TABLE IF EXISTS Dependiente;
```

```
CREATE TABLE Dependiente (  
    CodEmpleado INTEGER,  
    NomDependiente VARCHAR(30),  
    Sexo CHAR(1),  
    FechNacimiento DATE,  
    Parentesco VARCHAR(30),  
    PRIMARY KEY (CodEmpleado, NomDependiente)  
);
```

```
ALTER TABLE Trabaja_en ADD FOREIGN KEY (CodEmpleado) REFERENCES Empleado (CodEmpleado);
```

```
ALTER TABLE Trabaja_en ADD FOREIGN KEY (CodProyecto) REFERENCES Proyecto (CodProyecto);
```

```
ALTER TABLE Empleado ADD FOREIGN KEY (Supervisor) REFERENCES Empleado (CodEmpleado);
```

```
ALTER TABLE Empleado ADD FOREIGN KEY (CodDepto) REFERENCES Departamento (CodDepto);
```

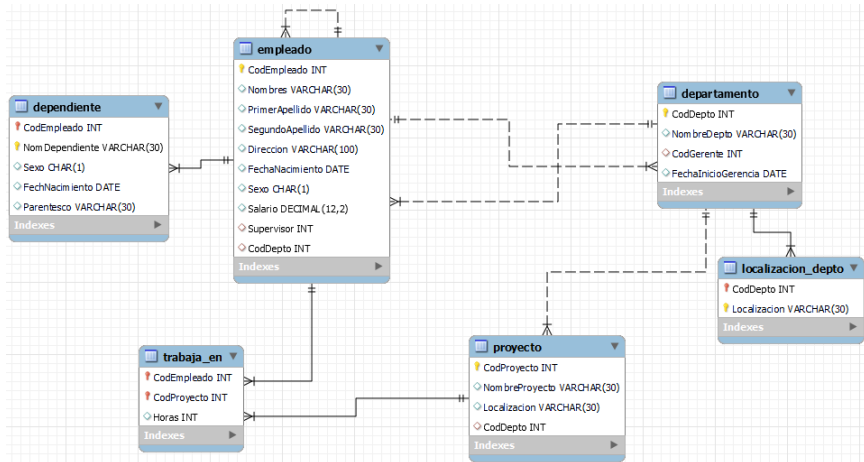
```
ALTER TABLE Departamento ADD FOREIGN KEY (CodGerente) REFERENCES Empleado (CodEmpleado);
```

```
ALTER TABLE Localizacion_Depto ADD FOREIGN KEY (CodDepto) REFERENCES Departamento (CodDepto);
```

```
ALTER TABLE Proyecto ADD FOREIGN KEY (CodDepto) REFERENCES Departamento (CodDepto);
```

```
ALTER TABLE Dependiente ADD FOREIGN KEY (CodEmpleado) REFERENCES Empleado (CodEmpleado);
```

Modelo Físico - Creación Base de Datos



Modelo Físico - Creación Base de Datos

```
INSERT INTO Departamento VALUES (1,'Direccion',NULL,'2001-06-19');
INSERT INTO Departamento VALUES (4,'Administracion',NULL,'1995-01-01');
INSERT INTO Departamento VALUES (5,'Investigacion',NULL,'1998-05-22');
```

```
INSERT INTO Empleado VALUES (123456789,'Juan','Perez','Rodriguez','Calle Numero A 1','1965-01-09','M',300.00,NULL,5);
INSERT INTO Empleado VALUES (333445555,'Frank','Velazquez','Flores','Calle Numero B 2','1955-12-08','M',4000.00,NULL,5);
INSERT INTO Empleado VALUES (453453453,'Daniela','Acco','Olvarez','Calle Numero F 6','1962-07-31','F',2500.00,NULL,5);
INSERT INTO Empleado VALUES (666884444,'Pedro','Lima','Maldonado','Calle Numero E 5','1952-09-15','M',1200.00,NULL,5);
INSERT INTO Empleado VALUES (888665555,'Francisco','Linares','Gomez','Calle Numero H 8','1957-11-10','M',5500.00,NULL,1);
INSERT INTO Empleado VALUES (987654321,'Luisa','Santos','Ferrel','Calle Numero D 4','1951-06-20','F',430.00,NULL,4);
INSERT INTO Empleado VALUES (987987987,'Mateo','Vela','Marruecos','Calle Numero G 7','1979-03-29','M',2500.00,NULL,4);
INSERT INTO Empleado VALUES (999887777,'Alice','Jimenez','Portugal','Calle Numero C 3','1968-07-19','F',2500.00,NULL,4);
```

Modelo Físico - Creación Base de Datos

```
UPDATE Departamento SET CodGerente=888665555 WHERE CodDepto=1;
```

```
UPDATE Departamento SET CodGerente=987654321 WHERE CodDepto=4;
```

```
UPDATE Departamento SET CodGerente=333445555 WHERE CodDepto=5;
```

```
UPDATE Empleado SET Supervisor=333445555, CodDepto=5 WHERE CodEmpleado=123456789;
```

```
UPDATE Empleado SET Supervisor=888665555, CodDepto=5 WHERE CodEmpleado=333445555;
```

```
UPDATE Empleado SET Supervisor=333445555, CodDepto=5 WHERE CodEmpleado=453453453;
```

```
UPDATE Empleado SET Supervisor=333445555, CodDepto=5 WHERE CodEmpleado=666884444;
```

```
UPDATE Empleado SET CodDepto=1 WHERE CodEmpleado=888665555;
```

```
UPDATE Empleado SET Supervisor=888665555, CodDepto=4 WHERE CodEmpleado=987654321;
```

```
UPDATE Empleado SET Supervisor=987654321, CodDepto=4 WHERE CodEmpleado=987987987;
```

```
UPDATE Empleado SET Supervisor=987654321, CodDepto=4 WHERE CodEmpleado=999887777;
```

Modelo Físico - Creación Base de Datos

```
INSERT INTO Localizacion_Depto VALUES (1,'Lima');  
INSERT INTO Localizacion_Depto VALUES (4,'Arequipa');  
INSERT INTO Localizacion_Depto VALUES (5,'Cusco');  
INSERT INTO Localizacion_Depto VALUES (5,'Puno');  
INSERT INTO Localizacion_Depto VALUES (5,'Trujillo');
```

```
INSERT INTO Dependiente VALUES (123456789,'Anabel','F','1998-12-31','Hija');  
INSERT INTO Dependiente VALUES (123456789,'Elizabeth','F','1957-05-05','Conyugue');  
INSERT INTO Dependiente VALUES (123456789,'Michael','M','1988-01-21','Hijo');  
INSERT INTO Dependiente VALUES (333445555,'Ana','F','1967-04-03','Hija');  
INSERT INTO Dependiente VALUES (333445555,'Juana','F','1958-05-05','Conyugue');  
INSERT INTO Dependiente VALUES (333445555,'Victor','M','1973-10-25','Hijo');  
INSERT INTO Dependiente VALUES (987654321,'Igor','M','1952-02-29','Conyugue');
```

Modelo Físico - Creación Base de Datos

```
INSERT INTO Proyecto VALUES (1,'Proyecto X','Oficina 317',5);
INSERT INTO Proyecto VALUES (2,'Proyecto Y','Oficina 22',5);
INSERT INTO Proyecto VALUES (3,'Proyecto Z','Oficina 544',5);
INSERT INTO Proyecto VALUES (10,'Proyecto A','Oficina 51',4);
INSERT INTO Proyecto VALUES (20,'Proyecto B','Oficina 67',1);
INSERT INTO Proyecto VALUES (30,'Proyecto C','Oficina 385',4);
```

```
INSERT INTO Trabaja_en VALUES (123456789,1,32);
INSERT INTO Trabaja_en VALUES (123456789,2,7);
INSERT INTO Trabaja_en VALUES (333445555,2,10);
INSERT INTO Trabaja_en VALUES (333445555,3,10);
INSERT INTO Trabaja_en VALUES (333445555,10,10);
INSERT INTO Trabaja_en VALUES (333445555,20,10);
INSERT INTO Trabaja_en VALUES (453453453,1,20);
INSERT INTO Trabaja_en VALUES (453453453,2,20);
INSERT INTO Trabaja_en VALUES (666884444,3,40);
INSERT INTO Trabaja_en VALUES (888665555,20,7);
INSERT INTO Trabaja_en VALUES (987654321,20,15);
INSERT INTO Trabaja_en VALUES (987654321,30,20);
INSERT INTO Trabaja_en VALUES (987987987,10,35);
INSERT INTO Trabaja_en VALUES (987987987,30,5);
INSERT INTO Trabaja_en VALUES (999887777,10,10);
INSERT INTO Trabaja_en VALUES (999887777,30,30);
```


Modelo Físico - Contenido de las Tablas

• Tabla empleados

| CodEmpleado | Nombres | PrimerApellido | SegundoApellido | Direccion | FechaNacimiento | Sexo | Salario | Supervisor | CodDepto |
|-------------|-----------|----------------|-----------------|------------------|-----------------|------|---------|------------|----------|
| 123456789 | Juan | Perez | Rodriguez | Calle Numero A 1 | 1965-01-09 | M | 300.00 | 333445555 | 5 |
| 333445555 | Frank | Velazquez | Flores | Calle Numero B 2 | 1955-12-08 | M | 4000.00 | 888665555 | 5 |
| 453453453 | Daniela | Acco | Olvarez | Calle Numero F 6 | 1962-07-31 | F | 2500.00 | 333445555 | 5 |
| 666884444 | Pedro | Lima | Maldonado | Calle Numero E 5 | 1952-09-15 | M | 1200.00 | 333445555 | 5 |
| 888665555 | Francisco | Linares | Gomez | Calle Numero H 8 | 1957-11-10 | M | 5500.00 | 888665555 | 1 |
| 987654321 | Luisa | Santos | Ferrel | Calle Numero D 4 | 1951-06-20 | F | 430.00 | 888665555 | 4 |
| 987987987 | Mateo | Vela | Marruecos | Calle Numero G 7 | 1979-03-29 | M | 2500.00 | 987654321 | 4 |
| 999887777 | Alice | Jimenez | Portugal | Calle Numero C 3 | 1968-07-19 | F | 2500.00 | 987654321 | 4 |

Modelo Físico - Contenido de las Tablas

- Tabla departamentos

| CodDepto | NombreDepto | CodGerente | FechaInicioGerencia |
|----------|----------------|------------|---------------------|
| 1 | Direccion | 888665555 | 2001-06-19 |
| 4 | Administracion | 987654321 | 1995-01-01 |
| 5 | Investigacion | 333445555 | 1998-05-22 |

Modelo Físico - Contenido de las Tablas

- Tabla dependientes

| CodEmpleado | NomDependiente | Sexo | FechNacimiento | Parentesco |
|-------------|----------------|------|----------------|------------|
| 123456789 | Anabel | F | 1998-12-31 | Hija |
| 123456789 | Elizabeth | F | 1957-05-05 | Conyugue |
| 123456789 | Michael | M | 1988-01-21 | Hijo |
| 333445555 | Ana | F | 1967-04-03 | Hija |
| 333445555 | Juana | F | 1958-05-05 | Conyugue |
| 333445555 | Victor | M | 1973-10-25 | Hijo |
| 987654321 | Igor | M | 1952-02-29 | Conyugue |

Modelo Físico - Contenido de las Tablas

- Tabla localizacion_depto

| CodDepto | Localizacion |
|----------|--------------|
| 1 | Lima |
| 4 | Arequipa |
| 5 | Cusco |
| 5 | Puno |
| 5 | Trujillo |

Modelo Físico - Contenido de las Tablas

- Tabla proyecto

| CodProyecto | NombreProyecto | Localizacion | CodDepto |
|-------------|----------------|--------------|----------|
| 1 | Proyecto X | Oficina 317 | 5 |
| 2 | Proyecto Y | Oficina 22 | 5 |
| 3 | Proyecto Z | Oficina 544 | 5 |
| 10 | Proyecto A | Oficina 51 | 4 |
| 20 | Proyecto B | Oficina 67 | 1 |
| 30 | Proyecto C | Oficina 385 | 4 |

Modelo Físico - Contenido de las Tablas

- Tabla trabaja_en

| CodEmpleado | CodProyecto | Horas |
|-------------|-------------|-------|
| 123456789 | 1 | 32 |
| 123456789 | 2 | 7 |
| 333445555 | 2 | 10 |
| 333445555 | 3 | 10 |
| 333445555 | 10 | 10 |
| 333445555 | 20 | 10 |
| 453453453 | 1 | 20 |
| 453453453 | 2 | 20 |
| 666884444 | 3 | 40 |
| 888665555 | 20 | 7 |
| 987654321 | 20 | 15 |
| 987654321 | 30 | 20 |
| 987987987 | 10 | 35 |
| 987987987 | 30 | 5 |
| 999887777 | 10 | 10 |
| 999887777 | 30 | 30 |

Lenguajes de consulta en BDR

- Los lenguajes de consulta son los lenguajes en el que los usuarios solicitan información de la base de datos.
- Estos lenguajes son generalmente de más alto nivel que los lenguajes de programación.
- Los lenguajes de consulta pueden clasificarse como procedimentales y no procedimentales.

Lenguajes de consulta en BDR

- Vamos a estudiar tres lenguajes de consulta en bases de datos relacionales:
 - Álgebra Relacional.
 - Cálculo Relacional.
 - SQL (Structured Query Language).

Álgebra Relacional y Cálculo Relacional

- Álgebra y cálculo relacional son dos formalismos lógico-matemáticos para escribir consultas.
- Álgebra y cálculo relacional fueron propuestos por Codd como lenguajes de consulta en BDR. Ambos son equivalentes.
- SQL se diseñó basándose en álgebra y cálculo relacional.
- Hasta cierto punto equivalentes a SQL. Permiten asegurar la consistencia matemática de SQL.

Algebra Relacional y Cálculo Relacional

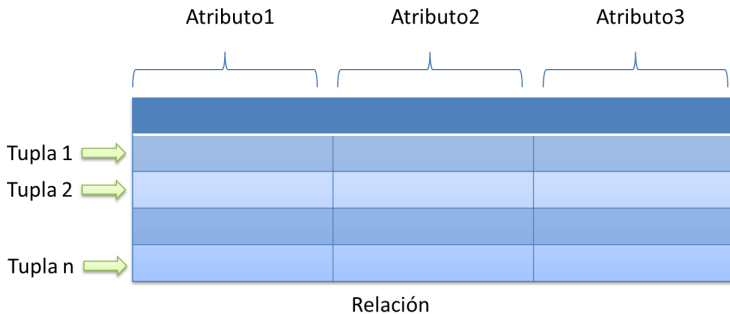
- Los lenguajes de consulta procedimentales requieren que el usuario especifique que datos se necesitan y cómo obtenerlos.
- El álgebra relacional es un lenguaje de consulta procedimental.
- El álgebra relacional define operadores que funcionan sobre las tablas, por ejemplo, unión, intersección, etc, para llegar al resultado deseado o información.

Algebra Relacional y Cálculo Relacional

- Los lenguajes de consulta no procedimentales requieren que el usuario especifique que datos se necesitan sin especificar cómo obtenerlos.
- El cálculo relacional es un lenguaje de consulta no procedimental.
- El cálculo relacional define el resultado o información a ser obtenida sin dar un procedimiento específico para obtener dicha información.

Algebra Relacional

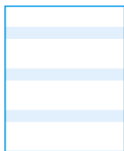
- Álgebra Relacional se define como un conjunto de operaciones que se ejecutan sobre las relaciones (tablas) para obtener un resultado, el cual es otra relación.



Algebra Relacional

- Las operaciones se pueden dividir en dos:
 - Operaciones tradicionales: unión, intersección, diferencia y producto cartesiano.
 - Operaciones especiales: selección, proyección, entre otras.

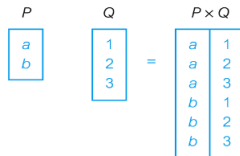
Algebra Relacional



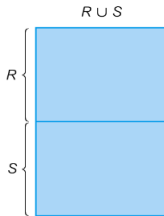
(a) Selection



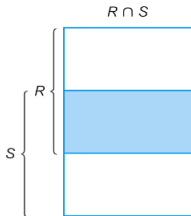
(b) Projection



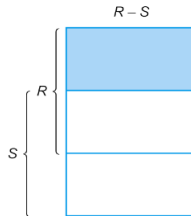
(c) Cartesian product



(d) Union



(e) Intersection



(f) Set difference

Algebra Relacional: Selección

- La operación de Selección es utilizada para seleccionar un conjunto de tuplas de una relación:

$$\sigma_{\langle cond \rangle}(\langle R \rangle)$$

- Donde $\langle cond \rangle$ es una condición de selección y $\langle R \rangle$ es el nombre de una relación.

Algebra Relacional: Selección

- Ejemplo: Seleccionar todos los empleados que trabajan en el departamento 5.

$$\sigma_{CodDepto=5}(Empleado)$$

| CodEmpleado | Nombres | PrimerApellido | SegundoApellido | Direccion | FechaNacimiento | Sexo | Salario | Supervisor | CodDepto |
|-------------|---------|----------------|-----------------|------------------|-----------------|------|---------|------------|----------|
| 123456789 | Juan | Perez | Rodriguez | Calle Numero A 1 | 1965-01-09 | M | 300.00 | 333445555 | 5 |
| 333445555 | Frank | Velazquez | Flores | Calle Numero B 2 | 1955-12-08 | M | 4000.00 | 888665555 | 5 |
| 453453453 | Daniela | Acco | Olvarez | Calle Numero F 6 | 1962-07-31 | F | 2500.00 | 333445555 | 5 |
| 666884444 | Pedro | Lima | Maldonado | Calle Numero E 5 | 1952-09-15 | M | 1200.00 | 333445555 | 5 |

Algebra Relacional: Selección

- Es una operación unitaria (realizada en una única relación).
- El grado (número de atributos) de la relación resultante es el mismo de la relación original.
- Se puede combinar un conjunto de operaciones de Selección en una única operación de Selección.
- La operación de selección es conmutativa:

$$\sigma_{\langle cond1 \rangle}(\sigma_{\langle cond2 \rangle}(\langle R \rangle)) = \sigma_{\langle cond2 \rangle}(\sigma_{\langle cond1 \rangle}(\langle R \rangle))$$

Algebra Relacional: Proyección

- La operación de Proyección es utilizada para seleccionar un conjunto de atributos de una relación:

$$\pi_{\langle atributos \rangle}(\langle R \rangle)$$

- Donde $\langle atributos \rangle$ es una lista de atributos dentro de los atributos de la relación R y $\langle R \rangle$ es el nombre de un relación.

Algebra Relacional: Proyección

- Ejemplo: Listar el nombre, primer apellido y el salario de todos los empleados.

$\pi_{Nombres, PrimerApellido, Salario}(Empleado)$

| Nombres | PrimerApellido | Salario |
|-----------|----------------|---------|
| Juan | Perez | 300.00 |
| Frank | Velazquez | 4000.00 |
| Daniela | Acco | 2500.00 |
| Pedro | Lima | 1200.00 |
| Francisco | Linares | 5500.00 |
| Luisa | Santos | 430.00 |
| Mateo | Vela | 2500.00 |
| Alice | Jimenez | 2500.00 |

Algebra Relacional: Proyección

- Es una operación unitaria (realizada en una única relación).
- Caso la lista de atributos incluya solamente atributos que no sean claves de R , es posible que ocurran tuplas duplicadas:
 - La operación Proyección elimina tuplas duplicadas de tal forma que el resultado sea una relación válida.
 - Con ello, el número de tuplas en la relación resultante es siempre menor o igual al número de tuplas de la relación R .
- La operación de proyección no es conmutativa.

Algebra Relacional: Secuencia de Operaciones

- Es común aplicar diversas operaciones del álgebra relacional, una después de la otra (secuencia de operaciones).
- Se puede escribir las operaciones en la forma de una única expresión o aplicar una operación secuencialmente, creando relaciones de resultado intermedio, en ese último caso, se debe nombrar las relaciones implicadas.

Algebra Relacional: Secuencia de Operaciones

- Ejemplo: Listar el nombre, primer apellido y el salario de todos los empleados que trabajan en el Departamento número 5.

$$\pi_{Nombres, PrimerApellido, Salario}(\sigma_{CodDepa=5}(Empleado))$$

Equivalente a:

$$Dep5Empleado \leftarrow \sigma_{CodDepa=5}(Empleado)$$

$$Resultado \leftarrow \pi_{Nombres, PrimerApellido, Salario}(Dep5Empleado)$$

| Nombres | PrimerApellido | Salario |
|---------|----------------|---------|
| Juan | Perez | 300.00 |
| Frank | Velazquez | 4000.00 |
| Daniela | Acco | 2500.00 |
| Pedro | Lima | 1200.00 |

Algebra Relacional: Secuencia de Operaciones

- Se puede utilizar la técnica de secuencia de operaciones para renombrar los atributos en la relaciones intermedias y del resultado: debemos listar los nombres de los nuevos atributos entre paréntesis juntamente con los nombres de las nuevas relaciones:

$$Dep5Empleado \leftarrow \sigma_{CodDepa=5}(Empleado)$$

$$Resultado(Nom, ApePaterno, Sueldo) \leftarrow$$

$$\pi_{Nombres, PrimerApellido, Salario}(Dep5Empleado)$$

Algebra Relacional: Teoría de Conjuntos

- El álgebra relacional posee un grupo patrón de operaciones matemáticas sobre conjuntos:
 - Las operaciones son binarias, es decir, envuelven dos relaciones.
 - Para algunas operaciones, las relaciones deben poseer el mismo tipo de tuplas, siendo consideradas compatibles para la unión, intersección y diferencia.
- Dos relaciones $R(A_1, A_2, \dots, A_n)$ y $S(B_1, B_2, \dots, B_n)$ son compatibles para la unión, intersección y diferencia si poseen el mismo grado "n" y si $dom(A_i) = dom(B_i)$ para $1 \leq i \leq n$.

Algebra Relacional: Teoría de Conjuntos

- Las operaciones de teoría de conjuntos que exigen relaciones compatibles son:
 - Unión: Denotada por $R \cup S$, genera una relación que incluye todas las tuplas que están en R o en S o en ambas.
 - Intersección: Denotada por $R \cap S$, genera una relación que incluye todas las tuplas que están en R y en S .
 - Diferencia: Denotada por $R - S$, genera una relación que incluye todas las tuplas que están en R , pero no en S .

Algebra Relacional: Teoría de Conjuntos

- La relación resultante de las operaciones posee los mismos nombres de atributos de la primera relación (R) envueltas en las operaciones.
- Las operaciones de unión e intersección son conmutativas y asociativas.

$$R \cup S = S \cup R \quad R \cap S = S \cap R$$

$$R \cup (S \cap T) = (R \cup S) \cap T \quad R \cap (S \cup T) = (R \cap S) \cup T$$

Algebra Relacional: Ejemplo - Unión

- Listar el código de todos los empleados que trabajan en el departamento 5 o que supervisan a algún empleado que trabaje en el departamento 5.

$$Dep5Empleados \leftarrow \sigma_{codDepto=5}(Empleado)$$

$$Resultado1 \leftarrow \pi_{CodEmpleado}(Dep5Empleados)$$

$$Resultado2(CodEmpleado) \leftarrow \pi_{supervisor}(Dep5Empleados)$$

$$Resultado(CodEmpleado) \leftarrow Resultado1 \cup Resultado2$$

| |
|-----------|
| |
| 123456789 |
| 333445555 |
| 453453453 |
| 666884444 |
| 333445555 |
| 888665555 |
| 333445555 |
| 333445555 |

Algebra Relacional: Teoría de Conjuntos

- La operación de conjunto binaria Producto Cartesiano, representada por \times , es utilizada para combinar tuplas de dos relaciones de forma combinatoria.
- Las relaciones no necesitan ser compatibles como para la unión.
- El resultado $R(A_1, A_2, \dots, A_n) \times S(B_1, B_2, \dots, B_m)$ es una relación $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ con $n + m$ atributos.
- La relación Q posee una tupla para cada combinación de tuplas de las relaciones implicadas: Si R posee n_R tuplas y S posee n_S tuplas, entonces Q poseerán $n_R * n_S$ tuplas.

Algebra Relacional: Teoría de Conjuntos

- No es una operación muy usual porque genera tuplas que no hacen sentido. Siempre se usa seguida por una Selección que combina valores de atributos en las relaciones implicadas.
- Ejemplo: Mostrar el Nombre, Apellido Paterno y Código de todos los empleados de sexo femenino que tengan dependientes:

Algebra Relacional: Teoría de Conjuntos

$$EmpMujeres \leftarrow \sigma_{sexo='F'}(Empleado)$$

$$NomEmpleado(Nom, ApePaterno, CodigoEmp)$$

$$\leftarrow \pi_{Nombres, PrimerApellido, CodEmpleado}(EmpMujeres)$$

$$DepenEmple \leftarrow NomEmpleado \times Dependiente$$

$$DepenCorrectos \leftarrow \sigma_{codigoEmp=CodEmpleado}(DepenEmple)$$

$$Resultado \leftarrow \pi_{Nomb, ApePaterno, CodigoEmp}(DepenCorrectos)$$

| Nomb | ApePaterno | CodigoEmp |
|-------|------------|-----------|
| Luisa | Santos | 987654321 |

Algebra Relacional: Teoría de Conjuntos

- Una vez que la operación Producto Cartesiano, seguida de la operación Selección, es usada con frecuencia, fue definida una operación especial, denominada Yunción, para especificar tal secuencia como una única operación.
- La operación Yunción es utilizada para combinar tuplas relacionadas de dos relaciones en un única tupla:

$$R \bowtie_{<cond>} S$$

- Donde R y S son relaciones y $< cond >$ es una condición de yunción entre las relaciones.

Algebra Relacional: Teoría de Conjuntos

- $R(A_1, A_2, \dots, A_n) \bowtie_{\langle cond \rangle} S(B_1, B_2, \dots, B_m)$ es una relación $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ con $n + m$ atributos.
- La relación Q posee una tupla para cada combinación de tuplas de las relaciones implicadas, siempre que la combinación satisfaga la condición de yunción.
- Una condición general de yunción es: $\langle cond_1 \rangle$ y $\langle cond_2 \rangle$ y ... y ... $\langle cond_N \rangle$, donde cada condición es de la forma $A_i \theta B_j$:
- A_i es el atributo de R , B_j es el atributo de S del mismo dominio de A_i , y θ es un operador de comparación $=, <, >, \leq, \geq, \neq$.

Algebra Relacional: Teoría de Conjuntos

- Ejemplo: Mostrar el Nombre, Apellido Paterno y Código de todos los empleados de sexo femenino que tengan dependientes:

$$EmpMujeres \leftarrow \sigma_{sexo='F'}(Empleado)$$

$$NomEmpleado(Nom, ApePaterno, CodigoEmp)$$

$$\leftarrow \pi_{Nombres, PrimerApellido, CodEmpleado}(EmpMujeres)$$

$$DepenCorrectos$$

$$\leftarrow NomEmpleado \bowtie_{codigoEmp=CodEmpleado} (Dependiente)$$

$$Resultado \leftarrow \pi_{Nomb, ApePaterno, CodigoEmp}(DepenCorrectos)$$

Algebra Relacional: Teoría de Conjuntos

- La operación Yunción más común, denominada Equiyunción, envuelve apenas condiciones de yunción con comparaciones de igualdad.
- Una equiyunción donde dos atributos de la comparación tienen el mismo nombre es llamada de Yunción Natural, siendo definida por $*$; en ese caso, apenas uno de los atributos de la comparación aparece en la relación resultante y la condición de yunción no es especificada.
- Ejemplo: Listar, para cada empleado del sexo femenino, los nombres de sus dependientes:

Algebra Relacional: Teoría de Conjuntos

$$EmpMujeres \leftarrow \sigma_{sexo='F'}(Empleado)$$
$$DepenCorrectos \leftarrow EmpMujeres * Dependiente$$
$$Resultado \leftarrow \pi_{Nomb, ApePaterno, CodigoEmp}(DepenCorrectos)$$

| Nomb | ApePaterno | CodigoEmp |
|-------|------------|-----------|
| Luisa | Santos | 987654321 |

Cálculo Relacional

- Al igual que el Álgebra Relacional (AR), el Cálculo Relacional de Tuplas (CRT) es un lenguaje de consulta asociado al Modelo Relacional.
- CRT es un lenguaje declarativo o no procedimental: Describe cuáles tuplas se deben devolver pero no como se calculan.
- Cualquier consulta escrita en AR puede ser expresada en CRT y viceversa, es decir, ambos tienen la misma expresividad.

Cálculo Relacional

- El CRT posee una base firme en la lógica matemática.
- El CRT se basa sobre la especificación de variables tupla.
- Cada variable tupla se extiende a lo largo de una relación y puede tomar como valor cualquier tupla de esa relación.
- El lenguaje de consulta estándar (SQL) tiene varios de sus fundamentos en el CRT. Veremos ejemplos de selección, proyección y yunción (también podemos usar unión, intersección y diferencia pero no veremos ejemplos de ello).

Cálculo Relacional

- La Fórmula General de Expresión puede ser definida de la siguiente manera:

$$\{t|F(t)\}$$

Conjunto de tuplas t tal que $F(t)$ es verdadera.

Cálculo Relacional: Relación de Intervalo (Range)

- Ejemplo: Listar todas las tuplas de la tabla empleados:

$$\{t | t \in \text{empleado}\}$$

| CodEmpleado | Nombres | PrimerApellido | SegundoApellido | Direccion | FechaNacimiento | Sexo | Salario | Supervisor | CodDepto |
|-------------|-----------|----------------|-----------------|------------------|-----------------|------|---------|------------|----------|
| 123456789 | Juan | Perez | Rodriguez | Calle Numero A 1 | 1965-01-09 | M | 300.00 | 333445555 | 5 |
| 333445555 | Frank | Velazquez | Flores | Calle Numero B 2 | 1955-12-08 | M | 4000.00 | 888665555 | 5 |
| 453453453 | Daniela | Acco | Olvarez | Calle Numero F 6 | 1962-07-31 | F | 2500.00 | 333445555 | 5 |
| 666884444 | Pedro | Lima | Maldonado | Calle Numero E 5 | 1952-09-15 | M | 1200.00 | 333445555 | 5 |
| 888665555 | Francisco | Linares | Gomez | Calle Numero H 8 | 1957-11-10 | M | 5500.00 | 888665555 | 1 |
| 987654321 | Luisa | Santos | Ferrel | Calle Numero D 4 | 1951-06-20 | F | 430.00 | 888665555 | 4 |
| 987987987 | Mateo | Vela | Marruecos | Calle Numero G 7 | 1979-03-29 | M | 2500.00 | 987654321 | 4 |
| 999887777 | Alice | Jimenez | Portugal | Calle Numero C 3 | 1968-07-19 | F | 2500.00 | 987654321 | 4 |

Cálculo Relacional: Selección

- Ejemplo: Listar las tuplas de la tabla empleados que sean hombres.

$$\{t \mid t \in \text{empleado} \wedge t.\text{Sexo} = 'M'\}$$

| CodEmpleado | Nombres | PrimerApellido | SegundoApellido | Direccion | FechaNacimiento | Sexo | Salario | Supervisor | CodDepto |
|-------------|-----------|----------------|-----------------|------------------|-----------------|------|---------|------------|----------|
| 123456789 | Juan | Perez | Rodriguez | Calle Numero A 1 | 1965-01-09 | M | 300.00 | 333445555 | 5 |
| 333445555 | Frank | Velazquez | Flores | Calle Numero B 2 | 1955-12-08 | M | 4000.00 | 888665555 | 5 |
| 666884444 | Pedro | Lima | Maldonado | Calle Numero E 5 | 1952-09-15 | M | 1200.00 | 333445555 | 5 |
| 888665555 | Francisco | Linares | Gomez | Calle Numero H 8 | 1957-11-10 | M | 5500.00 | NULL | 1 |
| 987987987 | Mateo | Vela | Marruecos | Calle Numero G 7 | 1979-03-29 | M | 2500.00 | 987654321 | 4 |

Cálculo Relacional: Proyección

- Ejemplo: Mostrar el nombre y Primer Apellido de la tabla empleados.

$$\{t | \exists s \in \text{empleado} (t.Nombres = s.Nombres \\ \wedge t.PrimerApellido = s.PrimerApellido) \}$$

| Nombres | PrimerApellido |
|-----------|----------------|
| Juan | Perez |
| Frank | Velazquez |
| Daniela | Acco |
| Pedro | Lima |
| Francisco | Linares |
| Luisa | Santos |
| Mateo | Vela |
| Alice | Jimenez |

Cálculo Relacional: Yunción

- Ejemplo: Mostrar el Nombre, Primer Apellido y Código de todos los empleados de sexo masculino que tengan dependientes:

Cálculo Relacional: Yunción

$$\{t | \exists e \in \text{empleado} \exists d \in \text{dependientes}$$

$$(t.Nombres = e.Nombres \wedge t.ApePaterno = e.ApePaterno$$

$$\wedge t.CodEmpleado = e.CodEmpleado$$

$$\wedge t.Sexo = 'M'$$

$$\wedge e.CodEmpleado = d.CodEmpleado) \}$$

| Nombres | PrimerApellido | CodEmpleado |
|---------|----------------|-------------|
| Juan | Perez | 123456789 |
| Juan | Perez | 123456789 |
| Juan | Perez | 123456789 |
| Frank | Velazquez | 333445555 |
| Frank | Velazquez | 333445555 |
| Frank | Velazquez | 333445555 |

SQL - Data Retrieval Language (DRL) - SELECT

- Para obtener información de una BDR podemos utilizar el comando SELECT

```

SELECT
[ALL | DISTINCT | DISTINCTROW ]
[HIGH_PRIORITY]
[STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
[SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr] ...
[into_option]
[FROM table_references
  [PARTITION partition_list]]
[WHERE where_condition]
[GROUP BY {col_name | expr | position}, ... [WITH ROLLUP]]
[HAVING where_condition]
[WINDOW window_name AS (window_spec)
  [, window_name AS (window_spec)] ...]
[ORDER BY {col_name | expr | position}
  [ASC | DESC], ... [WITH ROLLUP]]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[into_option]
[FOR {UPDATE | SHARE}
  [OF tbl_name [, tbl_name] ...]
  [NOWAIT | SKIP LOCKED]
  | LOCK IN SHARE MODE]
[into_option]

into_option: {
  INTO OUTFILE 'file_name'
    [CHARACTER SET charset_name]
    export_options
  | INTO DUMPFILE 'file_name'
  | INTO var_name [, var_name] ...
}

```

SQL - SELECT - Selección

- Seleccione la tabla empleados

- 1 • **USE** empresa;
- 2 • **SELECT * FROM** empleado;

| result Grid | | | | | | | | | | |
|--------------|-----------|----------------|-----------------|------------------|-----------------|--------------------|---------|------------|----------|--|
| Filter Rows: | | Edit: | | Export/Import: | | Wrap Cell Content: | | | | |
| CodEmpleado | Nombres | PrimerApellido | SegundoApellido | Direccion | FechaNacimiento | Sexo | Salario | Supervisor | CodDepto | |
| 123456789 | Juan | Perez | Rodriguez | Calle Numero A 1 | 1965-01-09 | M | 300.00 | 333445555 | 5 | |
| 333445555 | Frank | Velazquez | Flores | Calle Numero B 2 | 1955-12-08 | M | 4000.00 | 888665555 | 5 | |
| 453453453 | Daniela | Acco | Olvarez | Calle Numero F 6 | 1962-07-31 | F | 2500.00 | 333445555 | 5 | |
| 666884444 | Pedro | Lima | Maldonado | Calle Numero E 5 | 1952-09-15 | M | 1200.00 | 333445555 | 5 | |
| 888665555 | Francisco | Linares | Gomez | Calle Numero H 8 | 1957-11-10 | M | 5500.00 | 888665555 | 1 | |
| 987654321 | Luisa | Santos | Ferrel | Calle Numero D 4 | 1951-06-20 | F | 430.00 | 888665555 | 4 | |
| 987987987 | Mateo | Vela | Marruecos | Calle Numero G 7 | 1979-03-29 | M | 2500.00 | 987654321 | 4 | |
| 999887777 | Alice | Jimenez | Portugal | Calle Numero C 3 | 1968-07-19 | F | 2500.00 | 987654321 | 4 | |

SQL - SELECT - Selección

- Seleccione los empleados que sean hombres.

- 1 • **USE** empresa;
- 2 • **SELECT** * **FROM** empleado **WHERE** Sexo = 'M';

| result Grid | | | | | | | | | | |
|--------------|-----------|----------------|-----------------|------------------|-----------------|--------------------|---------|------------|----------|--|
| Filter Rows: | | Edit: | | Export/Import: | | Wrap Cell Content: | | | | |
| CodEmpleado | Nombres | PrimerApellido | SegundoApellido | Direccion | FechaNacimiento | Sexo | Salario | Supervisor | CodDepto | |
| 123456789 | Juan | Perez | Rodriguez | Calle Numero A 1 | 1965-01-09 | M | 300.00 | 333445555 | 5 | |
| 333445555 | Frank | Velazquez | Flores | Calle Numero B 2 | 1955-12-08 | M | 4000.00 | 888665555 | 5 | |
| 666884444 | Pedro | Lima | Maldonado | Calle Numero E 5 | 1952-09-15 | M | 1200.00 | 333445555 | 5 | |
| 888665555 | Francisco | Linares | Gomez | Calle Numero H 8 | 1957-11-10 | M | 5500.00 | NULL | 1 | |
| 987987987 | Mateo | Vela | Marruecos | Calle Numero G 7 | 1979-03-29 | M | 2500.00 | 987654321 | 4 | |

SQL - SELECT - Selección

- Como hemos visto, el comando WHERE puede utilizarse para seleccionar datos condicionalmente de una tabla.
- Esta condición puede ser una condición simple (como la vimos anteriormente), o puede ser una condición compuesta. Las condiciones compuestas están formadas por múltiples condiciones simples conectadas por AND u OR.
- No existe límites en el número de condiciones simples que pueden presentarse en una sola instrucción SQL.

SQL - SELECT - Selección

- Muestre los empleados con nombre Juan o Pedro

- 1 • **USE** empresa;
- 2 • **SELECT** * **FROM** empleado
- 3 • **WHERE** Nombres = 'Juan' **OR** Nombres = 'Pedro';

| result Grid | | | | | | | | | |
|--------------|---------|----------------|-----------------|------------------|-----------------|--------------------|---------|------------|----------|
| Filter Rows: | | Edit: | | Export/Import: | | Wrap Cell Content: | | | |
| CodEmpleado | Nombres | PrimerApellido | SegundoApellido | Direccion | FechaNacimiento | Sexo | Salario | Supervisor | CodDepto |
| 123456789 | Juan | Perez | Rodriguez | Calle Numero A 1 | 1965-01-09 | M | 300.00 | 333445555 | 5 |
| 666884444 | Pedro | Lima | Maldonado | Calle Numero E 5 | 1952-09-15 | M | 1200.00 | 333445555 | 5 |

SQL - SELECT - Selección

- Muestre los empleados que tengan un sueldo entre 100 y 1500 soles.
- 1 • **USE** empresa;
 - 2 • **SELECT** * **FROM** empleado
 - 3 • **WHERE** Salario >= 100 **AND** Salario <= 1500;

| result Grid | | | | | | | | | |
|--------------|---------|----------------|-----------------|------------------|-----------------|--------------------|---------|------------|----------|
| Filter Rows: | | Edit: | | Export/Import: | | Wrap Cell Content: | | | |
| CodEmpleado | Nombres | PrimerApellido | SegundoApellido | Direccion | FechaNacimiento | Sexo | Salario | Supervisor | CodDepto |
| 123456789 | Juan | Perez | Rodriguez | Calle Numero A 1 | 1965-01-09 | M | 300.00 | 333445555 | 5 |
| 666884444 | Pedro | Lima | Maldonado | Calle Numero E 5 | 1952-09-15 | M | 1200.00 | 333445555 | 5 |
| 987654321 | Luisa | Santos | Ferrel | Calle Numero D 4 | 1951-06-20 | F | 430.00 | 888665555 | 4 |

SQL - SELECT - Proyección

- Muestre el código y nombre de cada departamento.

- 1 • **USE** empresa;
- 2 • **SELECT** CodDepto **AS** 'Código', NombreDepto Nombre **FROM** departamento;

| Result Grid | | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|--------|----------------|---------|--------------------|
| | Código | Nombre | | |
| 1 | | Direccion | | |
| 4 | | Administracion | | |
| 5 | | Investigacion | | |

SQL - SELECT - Proyección

- LIKE es otro comando que se utiliza en la cláusula WHERE.
- Básicamente, LIKE nos permite hacer una búsqueda basada en un patrón en vez de especificar exactamente lo que se desea.
- Tarea: Estudiar las forma de búsqueda de textos.





SQL - SELECT - Proyección

- patrón generalmente consiste en comodines. Por ejemplo:
- 'A_Z': Toda línea que comience con 'A', otro carácter y termine con 'Z'. Por ejemplo, 'ABZ' y 'A2Z' deberían satisfacer la condición, mientras 'AKKZ' no debería (debido a que hay dos caracteres entre A y Z en vez de uno).
- 'ABC %': Todas las líneas que comienzan con 'ABC'. Por ejemplo, 'ABCD' y 'ABCABC' ambas deberían satisfacer la condición.
- ' %XYZ': Todas las líneas que terminan con 'XYZ'. Por ejemplo, 'WXYZ' y 'ZZXYZ' ambas deberían satisfacer la condición.
- ' %AN %': Todas las líneas que contienen el patrón 'AN' en cualquier posición. Por ejemplo, 'LOS ANGELES' y 'SAN FRANCISCO' ambos deberían satisfacer la condición.

SQL - SELECT - Proyección y Selección

- Muestre el código, Nombres, Primer Apellido y Segundo Apellido de los empleados cuyo apellido comience en 'L'.

```
1 • USE empresa;  
2 • SELECT CodEmpleado, Nombres, PrimerApellido, SegundoApellido  
3 FROM empleado  
4 WHERE PrimerApellido LIKE 'L%';
```

| Result Grid | | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|----------------|-----------------|
| Filter Rows: <input type="text"/> | | | |
| Edit:  | | | |
| Export/Import:   | | | |
| Wrap Cell Content:  | | | |
| CodEmpleado | Nombres | PrimerApellido | SegundoApellido |
| 666884444 | Pedro | Lima | Maldonado |
| 888665555 | Francisco | Linares | Gomez |