

FUNCIONES DE AGRUPACIÓN

```
SELECT  department_id, MAX(salary)
FROM    employees
WHERE   department_id <> 100
GROUP BY department_id
HAVING  MAX(salary)>10000 ;
```

CONCEPTO DE RELACIÓN

- Al inicio del curso, se comentaba que una base de datos se compone de entidades relevantes para un sistema/empresa en particular y que se representan como tablas.
- También se compone de las relaciones entre las tablas, esto tiene que ver con relaciones/asociaciones conceptuales por ejemplo:
 - La tabla ALUMNO estaría relacionada con una tabla ASIGNATURA porque un alumno puede tomar una o más asignaturas durante un semestre universitario.
 - La tabla CLIENTE estaría relacionada con una tabla PÓLIZA porque un cliente puede adquirir una o más pólizas.

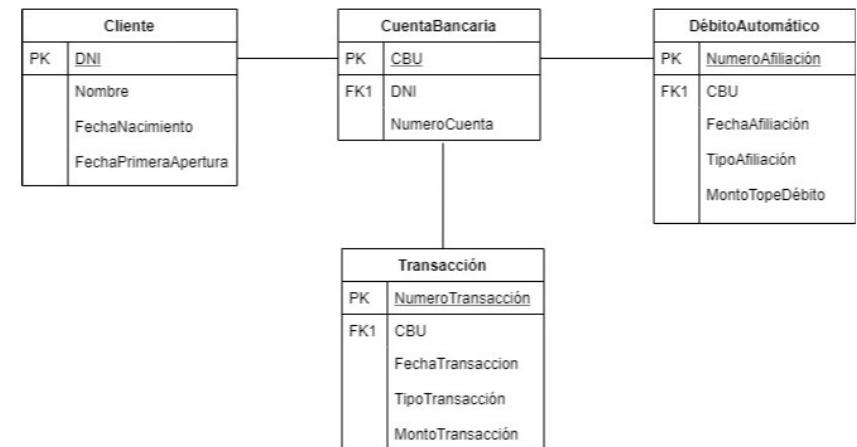
CARDINALIDAD EN LA RELACIÓN

Cada relación puede tener una cardinalidad específica.

La cardinalidad se refiere a cómo es la relación en función a cuántos registros de una tabla pueden estar relacionadas con registros de la segunda tabla y viceversa.

Puede ser:

- Uno a uno
- Uno a muchos
- Muchos a muchos

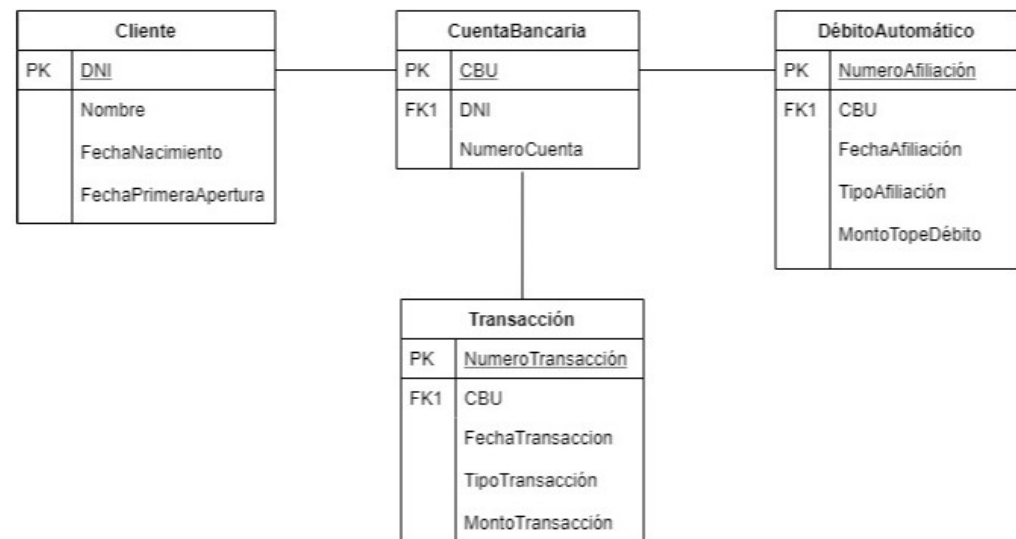


CÓMO GUARDAR LA RELACIÓN ENTRE LOS DATOS

- Estas relaciones también deben almacenarse en la base de datos según sea el caso:
 - Cuando se debe guardar información sobre la relación entre las tablas porque puede cambiar en el tiempo, se recomienda crear una tercera tabla llamada tabla-relación. Por ejemplo en la relación CLIENTE y SERVICIO se puede crear una tercera tabla SERVICIO_CLIENTE con la siguiente información:
 - ID_CLIENTE
 - ID_SERVICIO
 - TIPO_SERVICIO
 - FECHA_AFILIACION
 - FECHA_DESAFILIACION

CÓMO GUARDAR LA RELACIÓN ENTRE LOS DATOS

- Cuando la relación sea simple, agregando en una de las tablas el campo o campos que identifican al registro de la otra tabla, es decir el valor de su clave primaria (Primary Key). A este campo que será agregado se le llamará clave foránea (Foreign Key) porque es una clave pero que referencia a otra tabla.



RESTRICCIÓN FOREIGN KEY

```
CREATE TABLE CUSTOMERS(  
  ID INT NOT NULL,  
  NAME VARCHAR (20) NOT NULL,  
  AGE INT NOT NULL,  
  ADDRESS CHAR (25) ,  
  SALARY DECIMAL (18,2),  
  PRIMARY KEY (ID)  
);
```

```
CREATE TABLE ORDERS (  
  ID INT NOT NULL,  
  DATE DATETIME,  
  CUSTOMER_ID INT references CUSTOMERS(ID),  
  AMOUNT double,  
  PRIMARY KEY (ID)  
);
```

```
ALTER TABLE ORDERS  
  ADD FOREIGN KEY (Customer_ID) REFERENCES CUSTOMERS (ID);
```

CONCEPTO DE INTEGRIDAD REFERENCIAL



Es una propiedad de la base de datos.



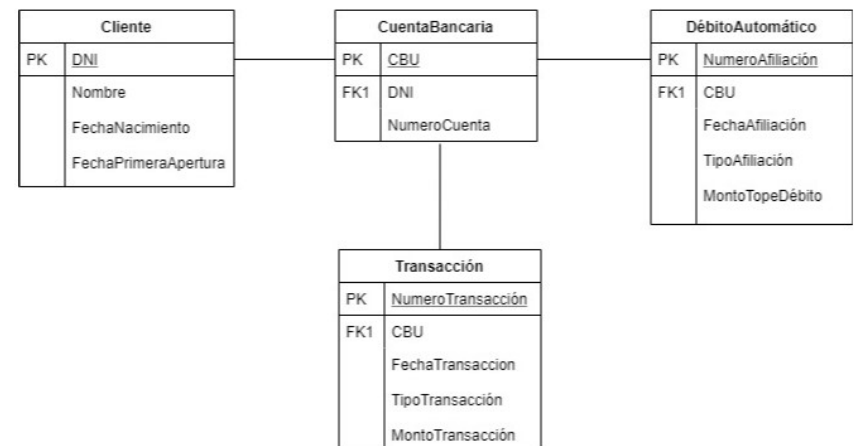
Consiste en garantizar que la clave foránea de una tabla siempre debe aludir a un registro válido de la tabla a la que hace referencia.



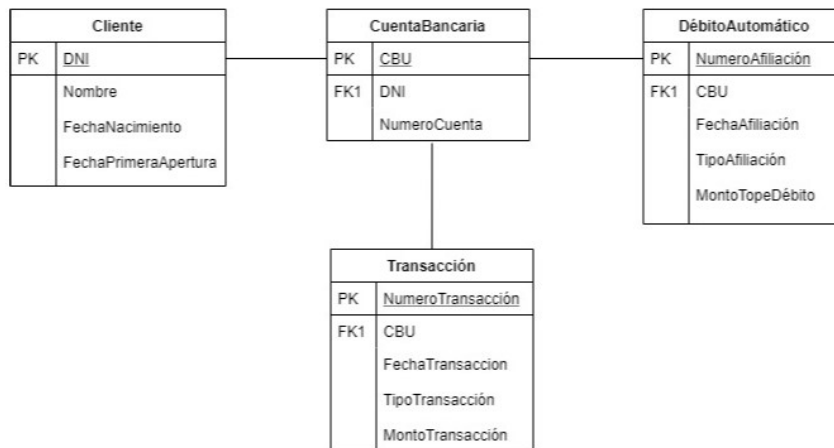
Permite asegurar que la relación entre las dos tablas permanezca sincronizada en cualquier operación que se haga en la base de datos. Entonces no se podrá eliminar un registro de la tabla referenciada si tiene algún registro de otra tabla que lo referencie.

CONSULTAS RELACIONANDO DISTINTAS TABLAS INNER JOIN

- El join es la instrucción que me permite obtener datos relacionados de más de una tabla en una única consulta.
- Por ejemplo, si necesito obtener todas las cuentas bancarias que tiene un cliente cuyo nombre es 'Antonio Perez'



CONSULTAS RELACIONANDO DISTINTAS TABLAS INNER JOIN



```
SELECT cb.cbu, cb.numerocuenta
FROM cliente c,
      cuentabancaria cb
WHERE c.dni = cb.dni
AND c.nombre = 'Antonio Perez';
Ó
SELECT cb.cbu, cb.numerocuenta
FROM cliente c
INNER JOIN cuentabancaria cb
ON (c.dni = cb.dni)
WHERE c.nombre = 'Antonio Perez';
```

LEFT JOIN

- Permite consultar dos o más tablas mostrando:
 - Todos los registros de la primera tabla referenciada en la consulta
 - Los registros de la segunda tabla que hacen match. Si algún registro de la primera tabla no hace match se muestra igual

LEFT JOIN

```
SELECT *  
FROM articulos a  
LEFT JOIN facturas f  
ON (a.ArticuloID = f.ArticuloID);
```

Artículos con el número de facturas

```
SELECT a.ArticuloID, a.Nombre, COUNT(I) num_facturas  
FROM articulos a  
LEFT JOIN facturas f  
ON (a.ArticuloID = f.ArticuloID)  
GROUP BY a.ArticuloID, a.Nombre;
```

ArticuloID	Nombre	Precio	Stock	Letra	Numero	ClienteID	ArticuloID	Fecha	Monto
1	Destornillador	25	50	A	1	1	1	2011-10-18	500
1	Destornillador	25	50	B	1	1	1	2011-11-18	5000
2	Pinza	35	22	A	2	2	2	2011-10-18	2500
2	Pinza	35	22	B	2	2	2	2011-11-18	200
3	Martillo	15	28	A	3	3	3	2011-10-18	320
3	Martillo	15	28	B	3	3	3	2011-11-18	3020
4	Maza	35	18	A	4	4	4	2011-10-18	120
4	Maza	35	18	B	4	4	4	2011-11-18	1200
5	Balde	55	13	A	5	5	5	2020-04-17	300
5	Balde	55	13	B	5	5	5	2020-04-24	3300
6	Cinta	110	50	NULL	NULL	NULL	NULL	NULL	NULL
7	Espatula	20	22	NULL	NULL	NULL	NULL	NULL	NULL
8	Fratas	35	28	NULL	NULL	NULL	NULL	NULL	NULL
9	Rearla	20	78	NULL	NULL	NULL	NULL	NULL	NULL
10	Tenaza	12	13	NULL	NULL	NULL	NULL	NULL	NULL

RIGHT JOIN

- Permite consultar dos o más tablas mostrando:
 - Todos los registros de la segunda tabla referenciada en la consulta
 - Los registros de la primera tabla que hacen match. Si algún registro de la segunda tabla no hace match se muestra igual

RIGHT JOIN

```
SELECT *  
FROM facturas f  
RIGHT JOIN clientes c  
ON (f.ClienteID = c.ClienteID);
```

Letra	Numero	ClienteID	ArticuloID	Fecha	Monto	ClienteID	Nombre	Apellido	Cuit	Direccion	Comentarios
A	1	1	1	2011-10-18	500	1	José	Diaz	20-35987452-0	Lima 458	NULL
B	1	1	1	2011-11-18	5000	1	José	Diaz	20-35987452-0	Lima 458	NULL
A	2	2	2	2011-10-18	2500	2	Ana	Lopez	20-37987854-0	Peru 32	NULL
B	2	2	2	2011-11-18	200	2	Ana	Lopez	20-37987854-0	Peru 32	NULL
A	3	3	3	2011-10-18	320	3	Cristian	Fraoa	20-36887498-0	Chile 6985	NULL
B	3	3	3	2011-11-18	3020	3	Cristian	Fraoa	20-36887498-0	Chile 6985	NULL
A	4	4	4	2011-10-18	120	4	Sol	Cabral	20-25982495-0	Lavalle 1201	NULL
B	4	4	4	2011-11-18	1200	4	Sol	Cabral	20-25982495-0	Lavalle 1201	NULL
A	5	5	5	2020-04-17	300	5	Ezequiel	Perez	20-21987111-0	Uruguay 25	NULL
B	5	5	5	2020-04-24	3300	5	Ezequiel	Perez	20-21987111-0	Uruguay 25	NULL
NULL	NULL	NULL	NULL	NULL	NULL	6	Mario	Pena	20-14987008-8	Aguero 635	NULL
NULL	NULL	NULL	NULL	NULL	NULL	7	Jorge	Rios	20-37562854-5	Nazca 1532	NULL
NULL	NULL	NULL	NULL	NULL	NULL	8	Valeria	Lagos	20-25487418-8	Chiope 5	NULL
NULL	NULL	NULL	NULL	NULL	NULL	9	Natalia	Peña	20-25982665-2	Jufre 3651	NULL
NULL	NULL	NULL	NULL	NULL	NULL	10	Juan	Khorn	20-23587171-9	Belarano 6525	NULL
NULL	NULL	NULL	NULL	NULL	NULL	11	Fernando	Castro	25-465768589-8	Avenida 12...	Nuevo Cliente