Jonnatan Arias Garcia

Estructura del Curso

- Modulo 1: SQL y Data bases relacionales
  - Introducción
  - Información en data models
  - Tipos de relaciones
  - Mapeo de entidades a tablas
  - Conceptos de modelos relacionales
  - Modulo 1 Lab
- Modulo 2: Modelos relacionales, Constrains y Data objects
  - Constrains: Modelo relacional
  - Constrains: Modelo relacional Avanzado
- Modulo 3: Definición de data Lenguaje (DOL) y manipulación de data (DML)
  - CREATE TABLE
  - INSERT
  - SELECT
  - UPDATE y DELETE
  - Lab 3

Modulo 4: DOL y DML avanzado

Patrones de Strings, rangos y conjuntos

Sorting

Grouping

Modulo 4 lab

Modulo 5: Trabajando con múltiples tablas

Overview

Inner Join

Left Outer Join

Right Outer Join

Full Join

Modulo 5 lab

Modulo 1

Que es SQL?

Es un lenguaje de programación para almacenar y procesar información en una base de datos relacional.

Se usa como sistema de búsqueda de data

data es una colección de números palabras o pinturas

Databases?

- Es un repositorio de almacenamiento de datos
- Se puede adicionar, modificar o buscar data
- Diferentes tipos de data-bases permiten ordenar los datos en diferentes formas
- Almacenar data in forma tabular es una base de datos relacional

Table: customers

customer_id	first_name	last_name	phone	country
1	John	Doe	817-646-8833	USA
2	Robert	Luna	412-862-0502	USA
3	David	Robinson	208-340-7906	UK
4	John	Reinhardt	307-242-6285	UK
5	Betty	Taylor	806-749-2958	UAE

Que es RDBMS?

- Relational Database Management System
- Un conjunto de herramientas de software que controla la data
  - Acceso, Organización y Almacenamiento
- Ejemplo: MySQL, Oracle, DB2 Express-C

SQL comandos básicos

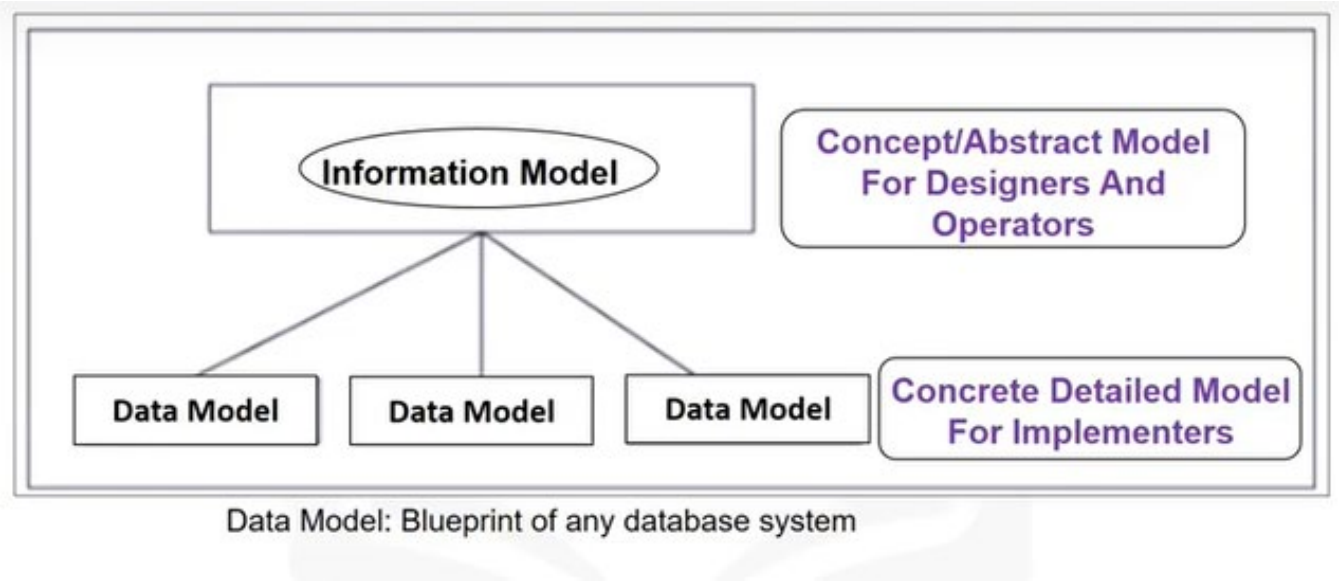
- Create
- Insert
- Select
- Update
- Delete

Information Model Vs. Data Model

Information model: es un resumen, una representación formal de las entidades que incluye.

Esta en un nivel conceptual

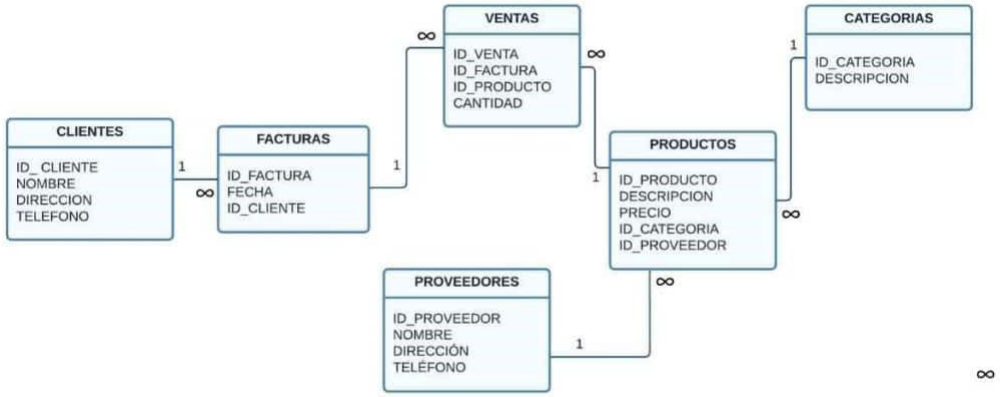
Data model: es la marca de una base de datos.

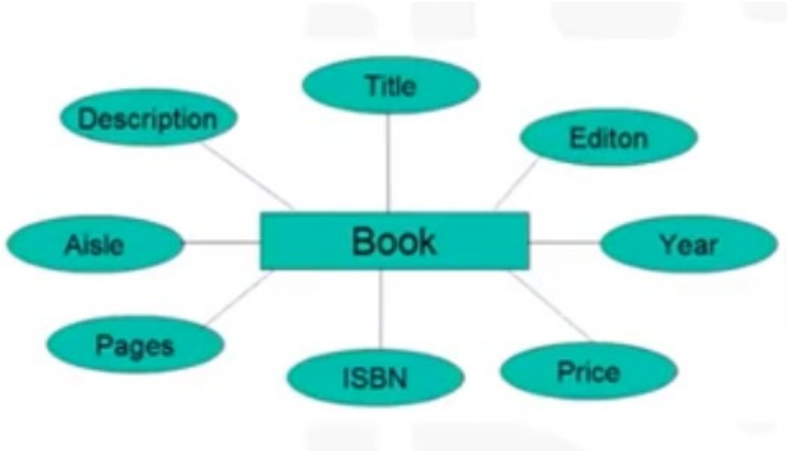


Data Model: Blueprint of any database system

Data Relacional

- Es la mas usado debido a su independencia
- El diagrama ER (entidad-relación) permite ver las relaciones de nuestros datos
- Los atributos se convierten en columnas y por cada atributo tenernos características por ejemplo para la entidad autor, sus atributos serian nombre, email, ciudad…





Tipos de relaciones

- 1 a 1
- 1 a muchos
- Muchos a muchos

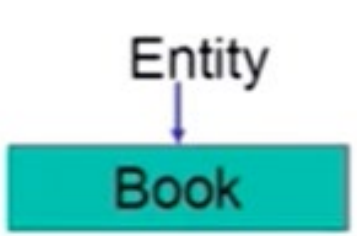
One-to-one Relationship

One-to-many Relationships

Many-to-many Relationships

Mapeando entidades en tablas

- Entidad: Nuestro punto central al cual se le adicionaran características o atributos



Mapeando entidades en tablas

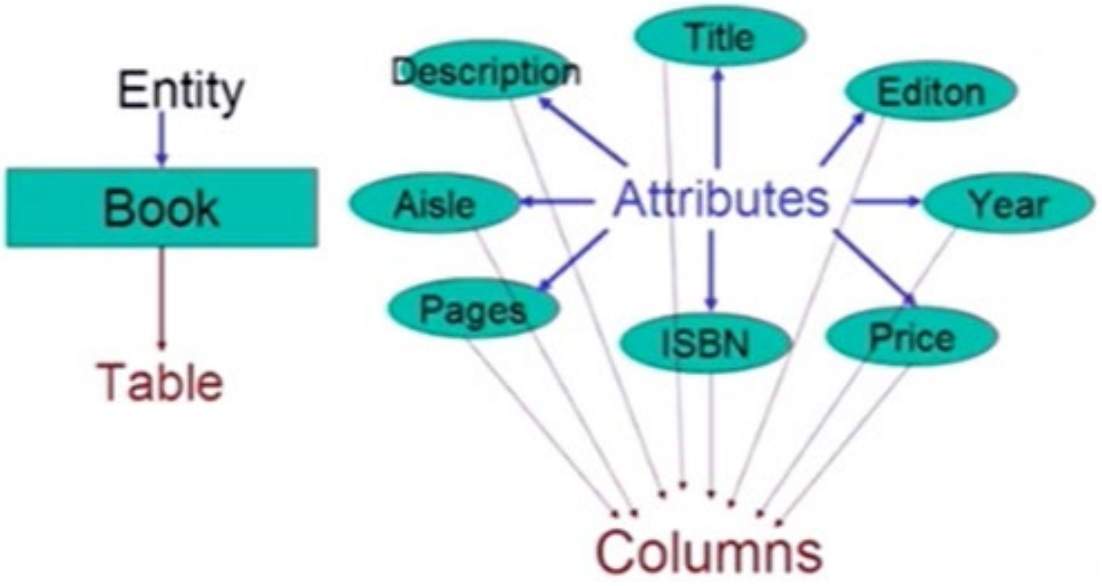
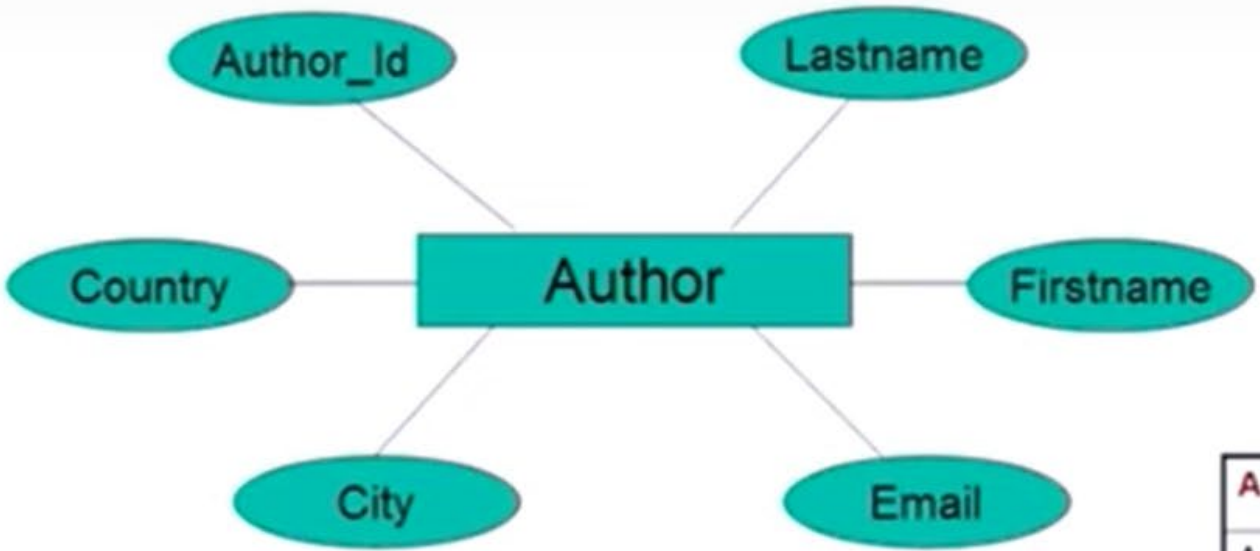


Table: Book

Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
Database Fundamentals	1	2010	24.99	978-0-98662 83-1-1	300	DB-A02	Teaches you the fundamentals of databases
Getting started with DB2 Express-C	1	2010	24.99	978-0-9866 283-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C, the free version of DB2



Author_ID	Lastname	Firstname	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	CA
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@univ.com	Transilvania	RO

Conceptos: El modelo relacional

El modelo relacional fue propuesto en 1970 basado en modelos matemáticos

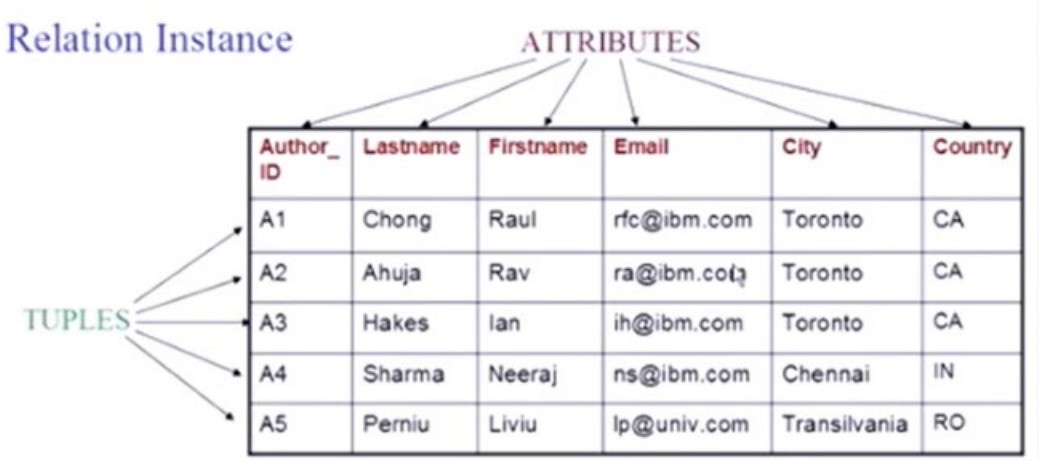
- Bloques
  - Relación
  - Conjuntos
- Conjunto
  - Colección desordenada de distintos elementos
  - Items del mismo tipo
  - Desordenada y sin duplicados

Relacional database

- Conjunto de relaciones
- Relación = Termino matemático para la tabla
- 2 partes
  - Esquema relacional: Especifica el nombre de la relación y el nombre y tipo de cada columna o atributo

AUTHOR (Author_ID:char, lastname: varchar, firstname: varchar, email: varchar, city: varchar, country:char)

  - Instancia relacional: Tabla hecha de filas (tupla) y columnas (atributos)



Author_ ID	Lastname	Firstname	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	CA
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@univ.com	Transilvania	RO

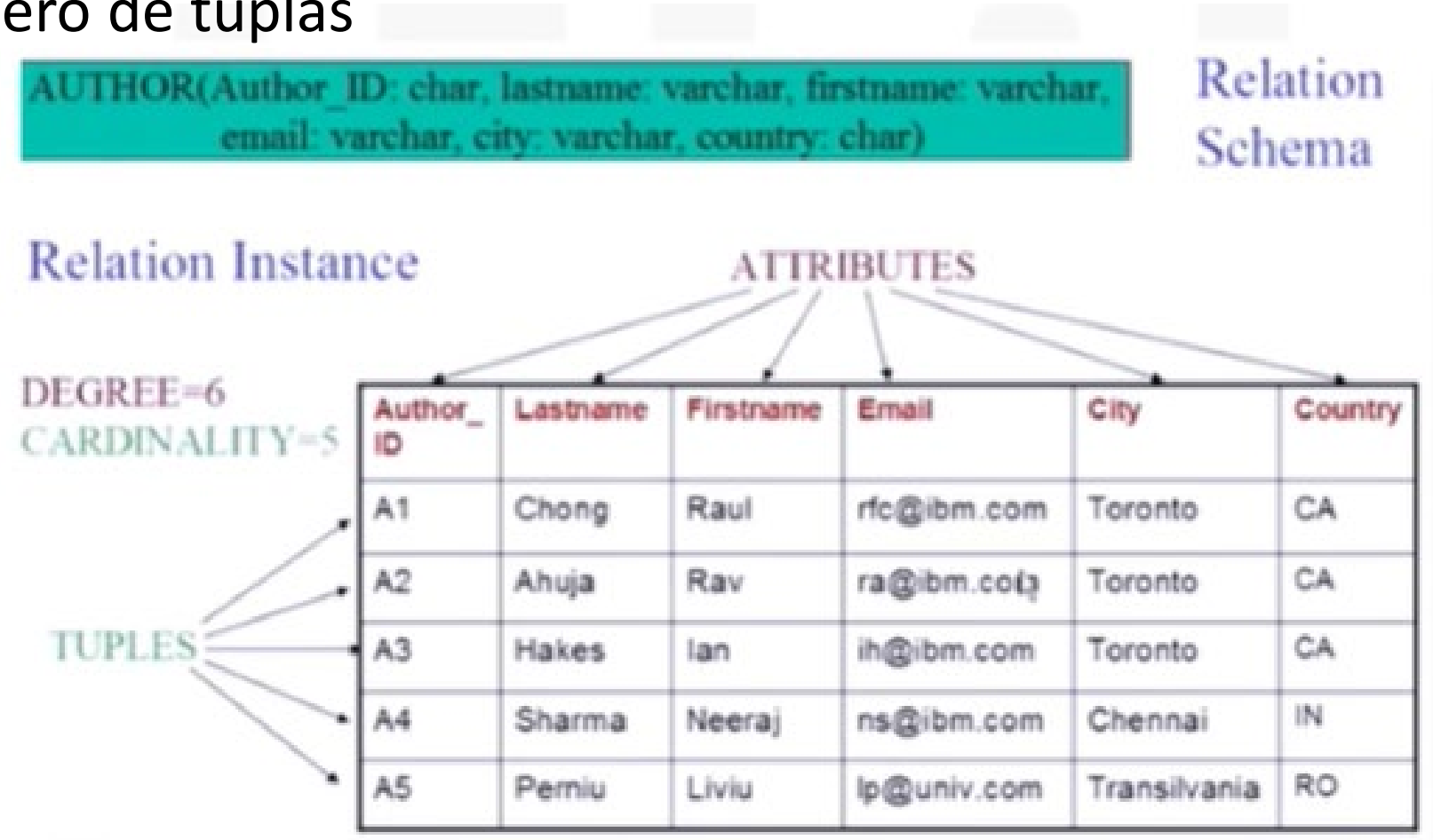
Relación

- Grado: Numero de atributos en una relación
- Cardinalidad: numero de tuplas

En el ejemplo

Degree = 6

Cardinality = 5



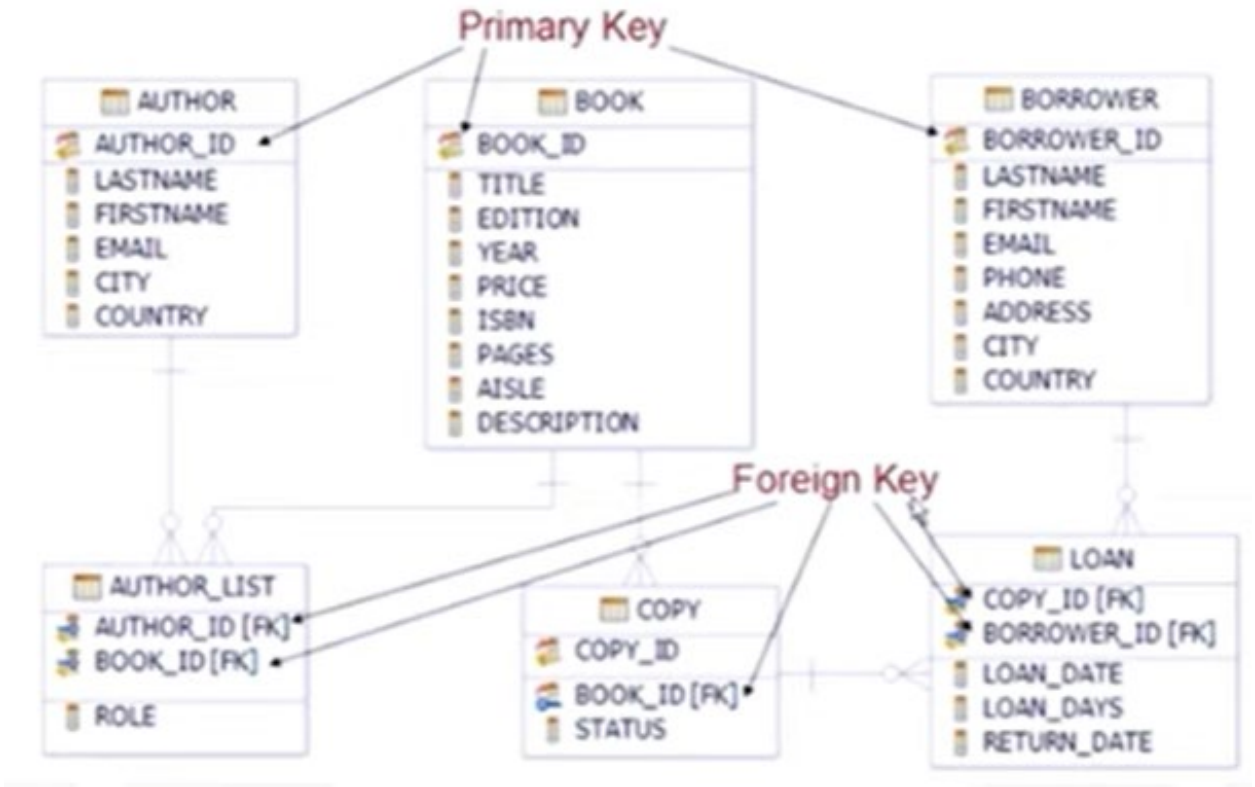
Actividad

- Preguntas
- Actividad de crear servidor en ibm puede ser o alguna db gratos (SQL)

Modulo 2

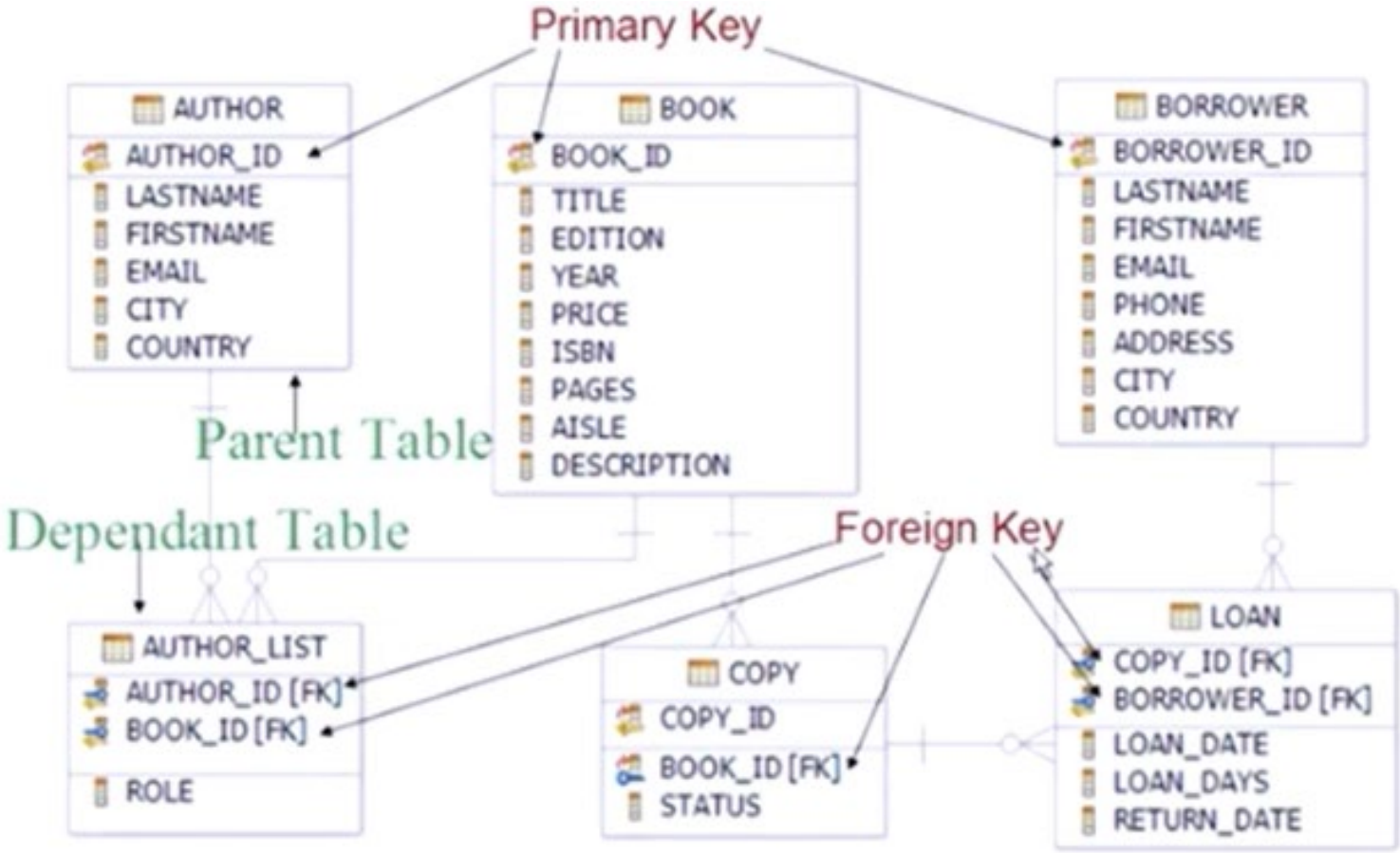
Relational model - limitaciones

- Primary key: identifica cada columna única
- Foreign key: son datos de identificación de otra entidad



Relational model - limitaciones

- Parent Table: Contiene una única primaria key
- Dependent table: Contiene una o mas keys extranjeras



Relational model - limitaciones

- Limitaciones:
  - Entity integrity: Es la clave primaria, permitiendo identificación única.
  - Referential Integrity: asegura que las relaciones entre tablas se mantengan. Esta restricción se implementa a través de claves foráneas (foreign keys).
  - Semantic Integrity: reglas que aseguran que los datos en una base de datos sean válidos en términos de formato, significado y lógica

  se basan en reglas de negocio que definen cómo deben comportarse los datos y cómo deben interactuar.

Relational model - limitaciones

- Limitaciones:
  - Domain: Especifica los valores permitidos para un atributo dado
  - Null: Reglas aplicadas a las columnas que determinan si los valores en esas columnas pueden ser nulos o no.
  - Check: Reglas aplicadas a las columnas de una tabla que definen condiciones específicas que deben cumplirse para que los datos en esas columnas sean válidos.

Adicional

- Reglas de las primary keys:
  - No pueden ser NULL
  - Es inmutable
  - Si se usa para enlazar múltiples atributos, ninguno de estos puede ser cambiado
- Reglas Semanticas
  - Data asociada (integers, real, char, bool, tamaño fijo, date, time, money,…)

Actividad

- Preguntas
- Carga de datos a database desde script, csv

Modulo 3

Tipos de SQL: DDL Vs. DML

Data Definition Language statements Vs. Data Manipulation Language statements

DDL son usado para definir, cambiar, o drop.

- CREATE: crea tablas y define columnas
- ALTER: alterar tablas incluyendo adicion y drop columnas
- TRUNCATE: borrar data en una tabla pero no la tabla
- DROP: borrar tablas

DML son usados para leer y modificar tablas (CRUD, créate, read, update, delete)

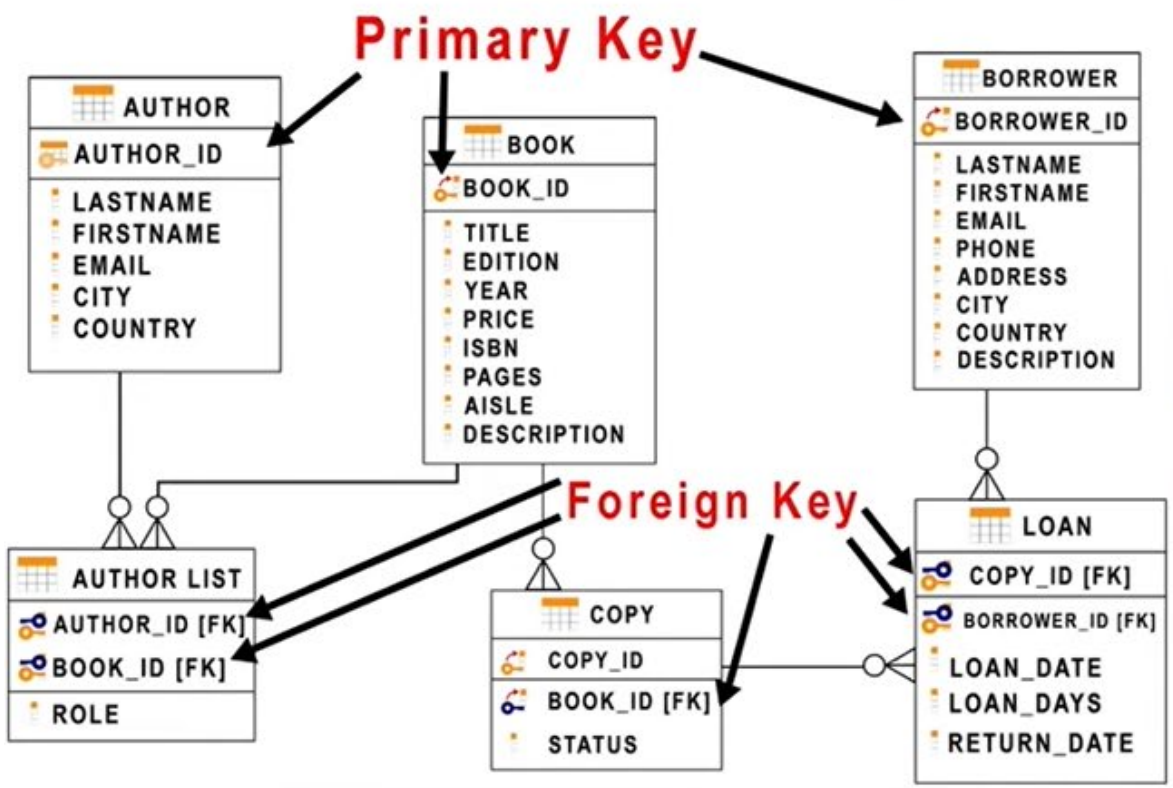
- INSERT: inserta filas
- SELECT: lee la fila seleccionada
- UPDATE: edita filas
- DELETE: remueve filas

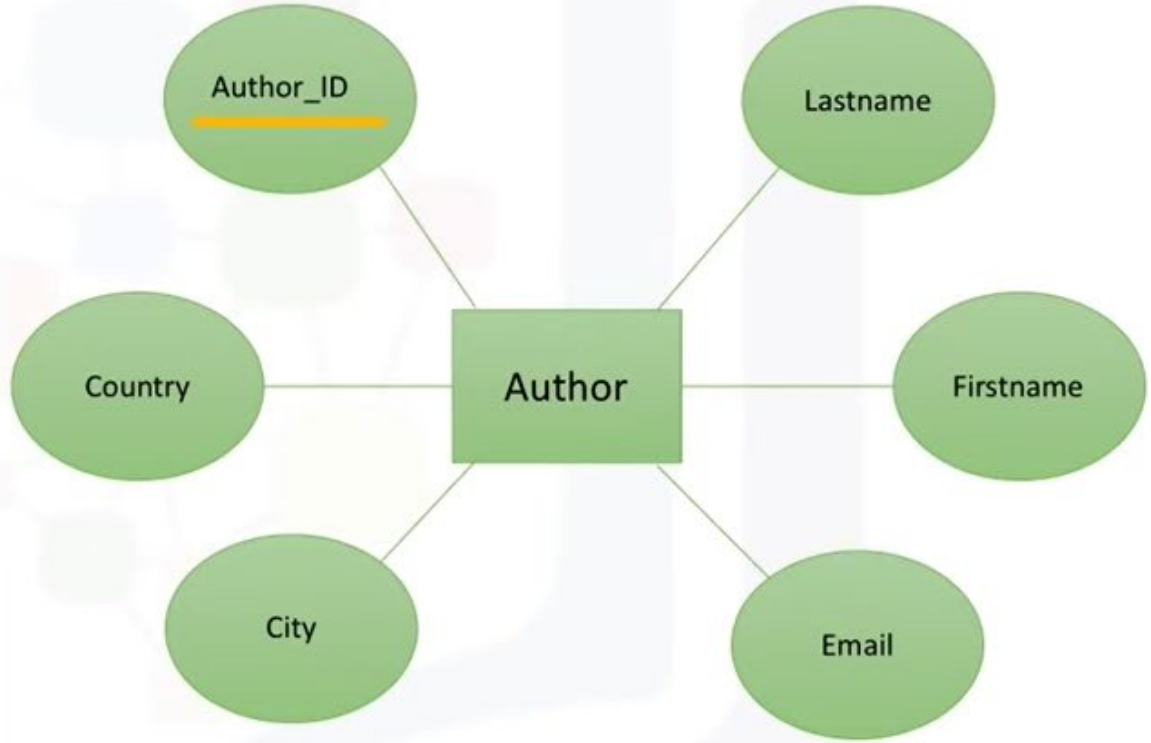
CREATE TABLE

```
CREATE TABLE table_name
   (
    column_name_1 datatype optional_parameters,
    column_name_2 datatype,
    …
    column_name_n datatype
   )
```

```
CREATE TABLE provinces(
    id char(2) PRIMARY KEY NOT NULL,
    name varchar(24)
   )
```

id *char(2)*	name *varchar(24)*
AB	ALBERTA
BC	BRITISH COLUMBIA
…	…

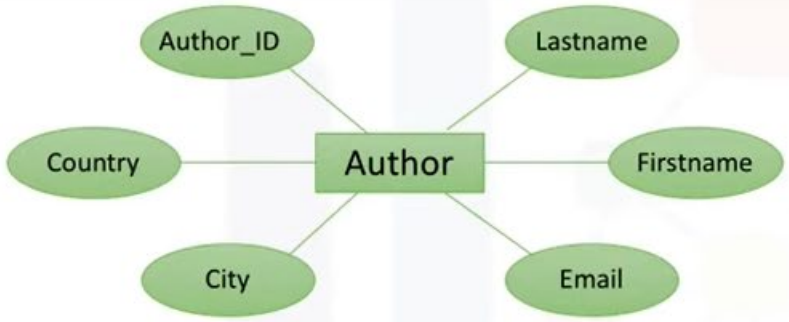





```
CREATE TABLE author (
    author_id CHAR(2) PRIMARY KEY NOT NULL,
    lastname VARCHAR(15) NOT NULL,
    firstname VARCHAR(15) NOT NULL,
    email VARCHAR(40),
    city VARCHAR(15),
    country CHAR(2)
   )
```

INSERT

- Se usa para Adicionarle filas a la tabla

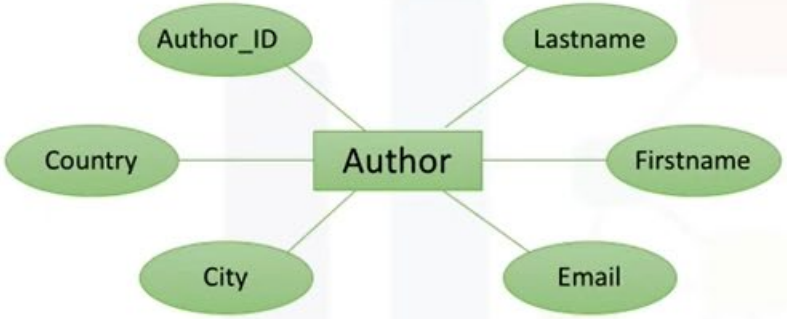


Author_ID	Lastname	Firstname	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	Ca
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

```
INSERT INTO [TableName]
    <([ColumnName],...)>
VALUES ([Value],...)
```

```
INSERT INTO AUTHOR
    (AUTHOR_ID, LASTNAME, FIRSTNAME, EMAIL, CITY, COUNTRY)
VALUES ('A1', 'Chong', 'Raul', 'rfc@ibm.com', 'Toronto', 'CA')
```

Insert múltiples filas



Author_ID	Lastname	Firstname	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	Ca
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

```
INSERT INTO AUTHOR
    (AUTHOR_ID, LASTNAME, FIRSTNAME, EMAIL, CITY, COUNTRY)
VALUES
    ('A1', 'Chong', 'Raul', 'rfc@ibm.com', 'Toronto', 'CA')
    ('A2', 'Ahuja', 'Rav', 'ra@ibm.com', 'Toronto', 'CA')
```

SELECT

- Método para recuperar información de la tabla, también es llamado Query

```
Select statement:    Query
Result from the query:   Result set/table

Select * from <tablename>
```

Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
Database Fundamentals	1	2010	24.99	978-0-9800628 3-1-1	300	DB-A02	Teaches you the fundamentals of databases
Getting started with DB2 Express-C	1	2010	24.99	978-0-9866628 3-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C

Example: select * from Book

```
db2 => select * from Book
```

Book_ID	Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
B1	Getting started with DB2 Express-C	1	2010	24.99	978-0-98666283-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C
B2	Database Fundamentals	1	2010	24.99	978-0-98006283-1-1	300	DB-A02	Teaches you the fundamentals of databases
B3	Getting started with DB2 App Dev	1	2011	35.99	978-0-98086283-4-1	345	DB-A03	Teaches you the essentials of developing applications for DB2.
B4	Getting started with WAS CE	1	2010	49.99	978-0-98946283-3-1	458	DB-A04	Teaches you the essentials of WebSphere Application Server

```
4 record(s) selected.
```

* selecciona todo

Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
Database Fundamentals	1	2010	24.99	978-0-9800628 3-1-1	300	DB-A02	Teaches you the fundamentals of databases
Getting started with DB2 Express-C	1	2010	24.99	978-0-9866628 3-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C

Example: select <column 1, column 2, ..., column n from Book

```
db2 => select book_id, title, edition, year, price, ISBN, pages, aisle, description from Book
```

Book_ID	Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
B1	Getting started with DB2 Express-C	1	2010	24.99	978-0-98666283-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C
B2	Database Fundamentals	1	2010	24.99	978-0-98006283-1-1	300	DB-A02	Teaches you the fundamentals of databases
B3	Getting started with DB2 App Dev	1	2011	35.99	978-0-98086283-4-1	345	DB-A03	Teaches you the essentials of developing applications for DB2.
B4	Getting started with WAS CE	1	2010	49.99	978-0-98946283-3-1	458	DB-A04	Teaches you the essentials of WebSphere Application Server

```
4 record(s) selected.
```

Columnas especificas

```
SELECT <column 1>, <column 2> from Book
```

```
db2 => select book_id, title from Book
```

Book_ID	Title
B1	Getting started with DB2 Express-C
B2	Database Fundamentals
B3	Getting started with DB2 App Dev
B4	Getting started with WAS CE

```
4 record(s) selected.
```

SELECT + WHERE

Recupera información donde se cumpla una expresión

```
select book_id, title from Book
    WHERE book_id = 'B1'
```

Equal to	=
Greater than	>
Lesser than	<
Greater than or equal to	>=
Less than or equal to	<=
Not equal to	<>

UPDATE & DELETE

UPDATE: modificar la tabla

```
UPDATE [TableName]
SET [[ColumnName]=[Value]]
<WHERE [Condition]>
```

Author_Id	LastName	FirstName	Email	City	Country
A1	CHONG	RAUL	rfc@ibm.com	Toronto	CA
A2	AHUJA	RAV	ra@ibm.com	Toronto	CA
A3	HAKES	IAN	ih@ibm.com	Toronto	CA

```
UPDATE AUTHOR
SET LASTNAME='KATTA'
    FIRSTNAME='LAKSHMI'
  WHERE AUTHOR_ID='A2'
```

Author_Id	LastName	FirstName	Email	City	Country
A1	CHONG	RAUL	rfc@ibm.com	Toronto	CA
A2	KATTA	LAKSHMI	ra@ibm.com	Toronto	CA
A3	HAKES	IAN	ih@ibm.com	Toronto	CA

UPDATE & DELETE

DELETE: borra 1 o mas filas

DELETE FROM [TableName]
<WHERE [Condition]>

Author_Id	LastName	FirstName	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A2	Ahuja	Rav	ra@ibm.com	Toronto	CA
A3	Hakes	Ian	ih@ibm.com	Toronto	CA
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

```
DELETE FROM AUTHOR
    WHERE AUTHOR_ID IN ('A2', 'A3')
```

Author_Id	LastName	FirstName	Email	City	Country
A1	Chong	Raul	rfc@ibm.com	Toronto	CA
A4	Sharma	Neeraj	ns@ibm.com	Chennai	IN
A5	Perniu	Liviu	lp@ibm.com	Transylvania	RO

Preguntas y Lab

Task 0: Drop the table INSTRUCTOR from the database in case it already exists, so that we start from a clean state.

(Hint: Ignore the undefined error if this table does not already exist in your database)

Task 1: Create the INSTRUCTOR table as defined above. Have the ins_id be the primary key, and ensure the lastname and firstname are not null.

(Hint: ins_id is of type INTEGER, country of type CHAR(2), and rest of the fields VARCHAR)

Task 2A: Insert one row into the INSTRUCTOR table for the the instructor Rav Ahuja.

(Hint: values for the character fields require a singe quotation mark (‘) before and after each value)

Task 2B: Insert two rows at once in the INSTRUCTOR table for instructors Raul Chong and Hima Vasudevan.

(Hint: list the values for the second row after the first row)

Task 3: Select all rows from the INSTRUCTOR table.

Task 3B: Select the firstname, lastname and country where the city is Toronto

Task 4: Update the row for Rav Ahuja and change his city to Markham.

Task 5: Delete the row for Raul Chong from the table.

Task 5B: Retrieve all rows in the INSTRUCTOR table

```
-- 0. Drop table INSTRUCTOR in case it already exists
drop table INSTRUCTOR
;
--1. Create table INSTRUCTOR
CREATE TABLE INSTRUCTOR
  (ins_id INTEGER PRIMARY KEY NOT NULL,
   lastname VARCHAR(15) NOT NULL,
   firstname VARCHAR(15) NOT NULL,
   city VARCHAR(15),
   country CHAR(2)
  )
;
--2A. Insert single row for Rav Ahuja
INSERT INTO INSTRUCTOR
  (ins_id, lastname, firstname, city, country)
  VALUES
  (1, 'Ahuja', 'Rav', 'Toronto', 'CA')
;
--2B. Insert the two rows for Raul and Hima
INSERT INTO INSTRUCTOR
  VALUES
  (2, 'Chong', 'Raul', 'Toronto', 'CA'),
  (3, 'Vasudevan', 'Hima', 'Chicago', 'US')
;
--3. Select all rows in the table
SELECT * FROM INSTRUCTOR
;
--3b. Select firstname, lastname and country where city is
Toronto
SELECT firstname, lastname, country from INSTRUCTOR where
city='Toronto'
;
--4. Change the city for Rav to Markham
UPDATE INSTRUCTOR SET city='Markham' where ins_id=1
;
--5. Delete the row for Raul Chong
DELETE FROM INSTRUCTOR where ins_id=2
;
--5b. Retrieve all rows from the table
SELECT * FROM INSTRUCTOR
```

Modulo 4

SELECT – Avanzado

Solíamos usar WHERE para una búsqueda especifica, la cual nos debe retornar Falso, True ó Unknow.

Que pasa si no sabemos el valor exacto a buscar?

- Condiciones usando LIKE:
  - R%: Busca aquellos que inicien con R

```
db2 => select firstname from author
where firstname like 'R%'

FIRSTNAME
---------------
RAUL
RAV

  2 record(s) selected.
```

SELECT - Avanzado

Solíamos usar WHERE para una búsqueda especifica, la cual nos debe retornar Falso, True ó Unknow.

Que pasa si no sabemos el valor exacto a buscar?

- Condiciones usando LIKE:
  - R%: Busca aquellos que inicien con R
- Usando un rango (BETWEEN, AND, <,>,<=,>=)

```
db2 => select title, pages from book
where pages >= 290 AND pages <=300

TITLE                                PAGES
-------------------------------- -----------

Database Fundamentals                    300

Getting started with DB2 App Dev         298

  2 record(s) selected.
```

```
db2 => select title, pages from book
where pages between 290 and 300

TITLE                                PAGES
-------------------------------- --------
Database Fundamentals                 300
Getting started with DB2 App Dev      298

  2 record(s) selected.
```

SELECT - Avanzado

Solíamos usar WHERE para una búsqueda especifica, la cual nos debe retornar Falso, True ó Unknow.

Que pasa si no sabemos el valor exacto a buscar?

- Condiciones usando LIKE:
  - R%: Busca aquellos que inicien con R
- Usando un rango (BETWEEN, AND, <,>,<=,>=)
- Usando un conjunto de valores (OR, IN)

```
db2 => select firstname, lastname,
country from author where country='AU'
OR country='BR'

FIRSTNAME       LASTNAME        COUNTRY
--------------- --------------- -------
Xiqiang         Ji              AU
Juliano         Martins         BR

  2 record(s) selected.
```

```
db2 => select firstname, lastname,
  country from author where country
  IN ('AU','BR')

FIRSTNAME       LASTNAME        COUNTRY
--------------- --------------- -------
Xiqiang         Ji              AU
Juliano         Martins         BR

  2 record(s) selected.
```

SORTING Results

- ORDER BY: usado para organizar resultados según un conjunto u orden especifica.

```
db2 => select title from book

TITLE
---------------------------------
Getting started with DB2 Express-C
Database Fundamentals
Getting started with DB2 App Dev
Getting started with WAS CE

  4 record(s) selected.
```

```
db2 => select title from book
       order by title

TITLE
---------------------------------
Database Fundamentals
Getting started with DB2 App Dev
Getting started with DB2 Express-C
Getting started with WAS CE

  4 record(s) selected.
```

By default the result set is sorted in ascending order

SORTING Results

```
db2 => select title from book
       order by title

TITLE
------------------------------
Database Fundamentals
Getting started with DB2 App Dev
Getting started with DB2 Express-C
Getting started with WAS CE

  4 record(s) selected.
```

Ascending order by default

```
db2 => select title from book
       order by title desc

TITLE
------------------------------
Getting started with WAS CE
Getting started with DB2 Express-C
Getting started with DB2 App Dev
Database Fundamentals

  4 record(s) selected.
```

Descending order with keyword

SORTING Results

```
db2 => select title, pages from book
            order by 2

TITLE                                PAGES
-----------------------------------  -------
Getting started with WAS CE          278
Getting started with DB2 Express-C   280
Getting started with DB2 App Dev     298
Database Fundamentals                300

  4 record(s) selected.
```

Ascending order by Column 2 (number of pages)

Eliminando duplicados

Sort para ver duplicados:

Si usamos DISTINCT(column), nos filtrara los duplicados

Si queremos agrupar por un tipo GROUP BY (atributo)

COUNT y AS COUNT nos calcula la cantidad (cuenta) y organiza según las clases

```
db2 => select country from author order by 1

COUNTRY
-------
AU
BR
...
CN
CN
...
IN
IN
IN
...
RO
RO

  20 rec
```

```
db2 => select distinct(country) from author

COUNTRY
-------
AU
BR
CA
CN
IN
RO

  6 recor
```

```
db2 => select country, count(country) from
author group by country

COUNTRY 2
------- -----------
AU                1
BR                1
CA                3
CN                6
IN                6
RO                3

  6
```

```
db2 => select country, count(country)
     as count from author group by country

COUNTRY COUNT
------- -----------
AU                1
BR                1
CA                3
CN                6
IN                6
RO                3

  6 record(s) selected.
```

Restringiendo resultados de GROUP BY

HAVING, es usada como combinación de GROUP BY y permite filtrar nuestra clase

```
db2 => select country, count(country)
     as count from author group by country

COUNTRY COUNT
------- -----------
AU                1
BR                1
CA                3
CN                6
IN                6
RO                3

  6 record(s) selected.
```

```
db2 => select country, count(country)
as count from author group by country
having count(country) > 4

COUNTRY COUNT
------- -----------
CN                6
IN                6

  2 record(s) selected.
```

Preguntas y Lab

1. Retrieve all employees whose address is in Elgin,IL.

2. Retrieve all employees who were born during the 1970’

3. Retrieve all employees in department 5 whose salary is between 60000 and 70000

4.Retrieve a list of employees ordered by department ID

5. Retrieve a list of employees ordered in descending order by department ID and within each department ordered alphabetically in descending order by last name

6.For each department ID retrieve the number of employees in the department

7. For each department retrieve the number of employees in the department, and the average employee salary in the department

8. Label the computed columns in the result set of SQL problem 2 (Exercise 3 Problem 2) as NUM_EMPLOYEES and AVG_SALARY

9. In SQL problem 3 (Exercise 3 Problem 3), order the result set by Average Salary

10. In SQL problem 4 (Exercise 3 Problem 4), limit the result to departments with fewer than 4 employees

```
SELECT EMP_ID, F_NAME, L_NAME
FROM employees
WHERE ADDRESS LIKE '%Elgin,IL%';
```

```
SELECT EMP_ID, F_NAME, L_NAME
FROM employees
WHERE B_DATE LIKE '%197%';
```

```
SELECT EMP_ID, F_NAME, L_NAME, SALARY
FROM employees
WHERE (SALARY BETWEEN 60000 AND 70000) AND DEP_ID =5;
```

```
SELECT EMP_ID, F_NAME, L_NAME, DEP_ID
FROM EMPLOYEES
ORDER BY DEP_ID;
```

```
SELECT EMP_ID, F_NAME, L_NAME, DEP_ID
FROM EMPLOYEES
ORDER BY DEP_ID DESC, L_NAME DESC;
```

```
SELECT DEP_ID, COUNT(*)
FROM EMPLOYEES
GROUP BY DEP_ID;
```

```
SELECT DEP_ID, COUNT(*), AVG(SALARY)
FROM EMPLOYEES
GROUP BY DEP_ID;
```

```
SELECT DEP_ID, COUNT(*) AS "NUM_EMPLOYEES",
AVG(SALARY) AS "AVG_SALARY"
FROM EMPLOYEES
GROUP BY DEP_ID;
```

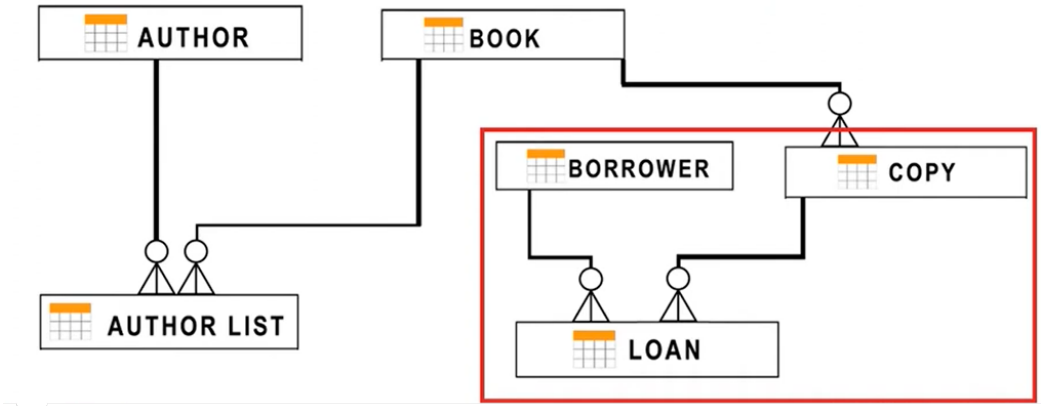
```
SELECT DEP_ID, COUNT(*) AS "NUM_EMPLOYEES",
AVG(SALARY) AS "AVG_SALARY"
FROM EMPLOYEES
GROUP BY DEP_ID
ORDER BY AVG_SALARY;
```

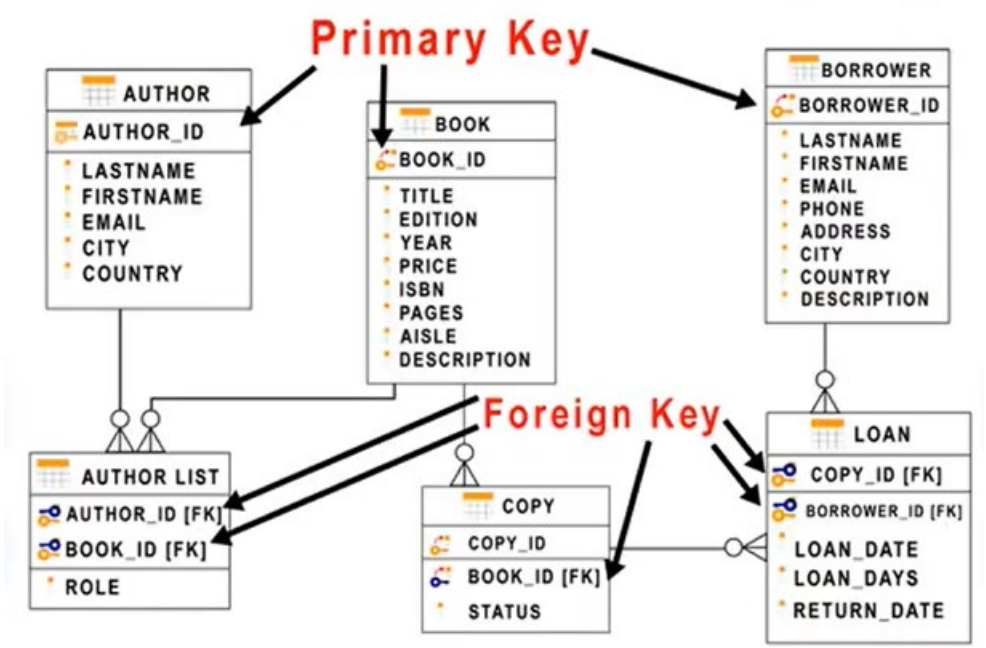
```
SELECT DEP_ID, COUNT(*) AS "NUM_EMPLOYEES",
AVG(SALARY) AS "AVG_SALARY"
FROM EMPLOYEES
GROUP BY DEP_ID
HAVING count(*) < 4
ORDER BY AVG_SALARY;
```

Modulo 5

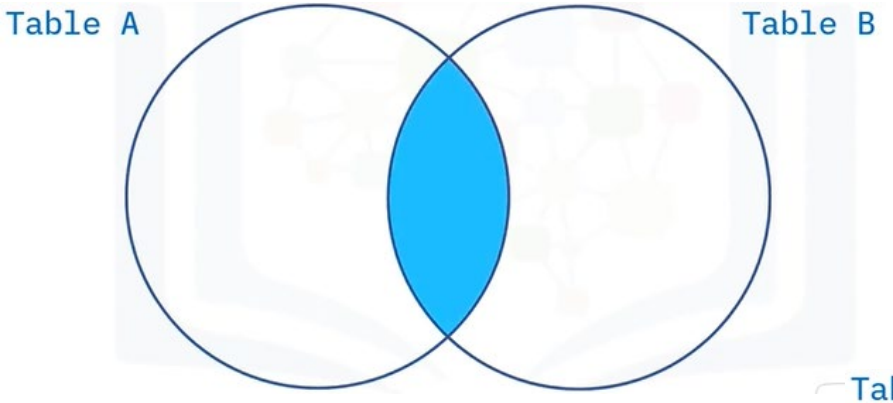
Operador Join

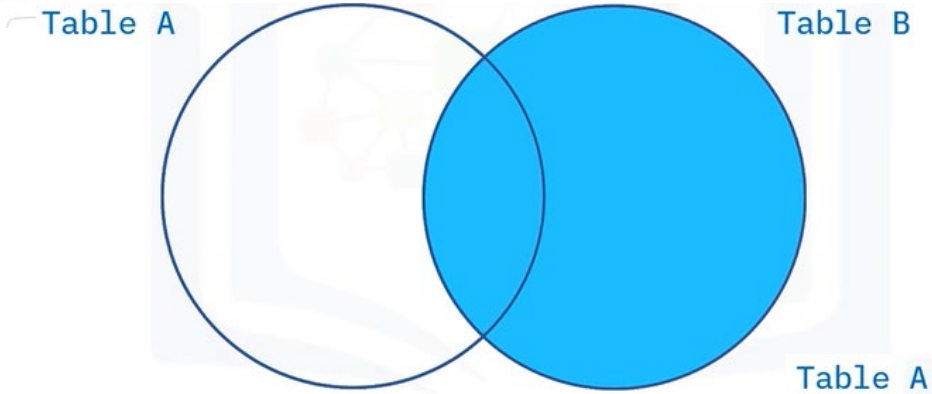
- Combinar filas de dos o mas tablas
- Se basa en las relaciones



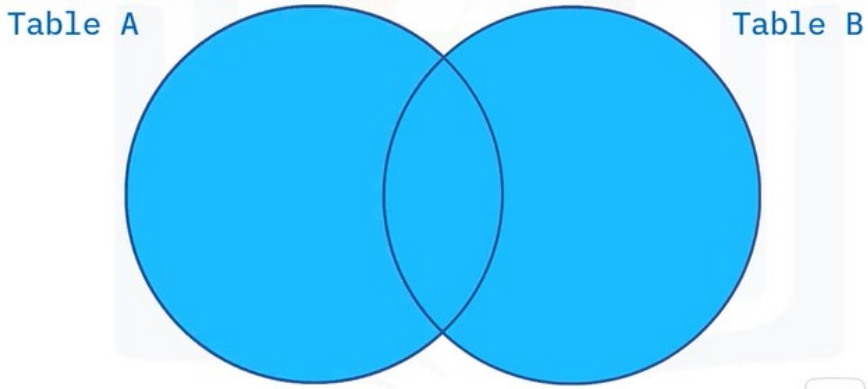


Tipos de Joins

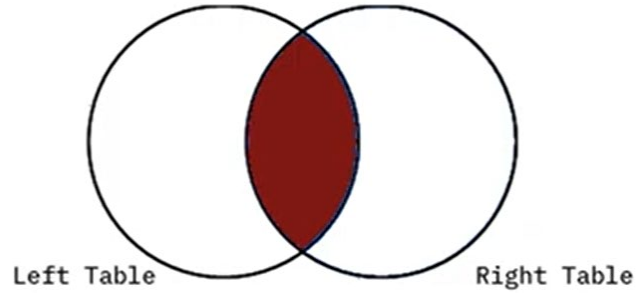




- Inner Join
- Outer Join
  - Left Outer Join
  - Right Outer Join
  - Full Outer Join

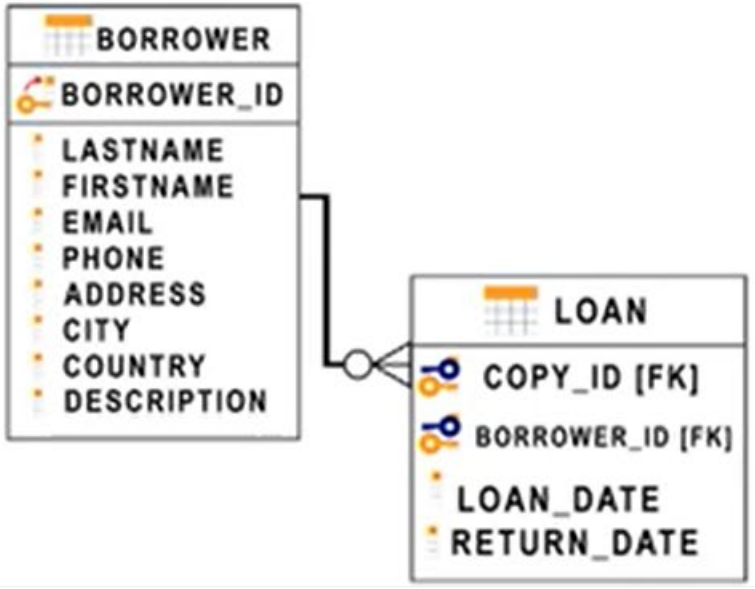


Inner Join

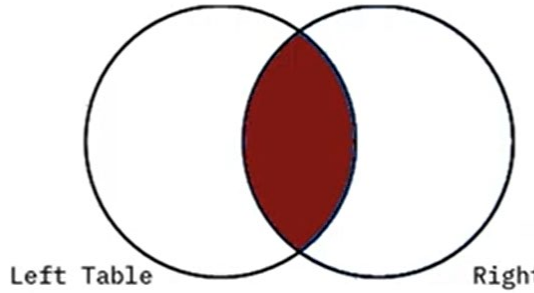


- Nos presenta el enlace entre las dos tablas

```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,
       L.BORROWER_ID, L.LOAN_DATE
FROM BORROWER B INNER JOIN LOAN L
    ON B.BORROWER_ID = L.BORROWER_ID
```



Inner Join




```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,
       L.BORROWER_ID, L.LOAN_DATE
FROM BORROWER B INNER JOIN LOAN L
    ON B.BORROWER_ID = L.BORROWER_ID
```

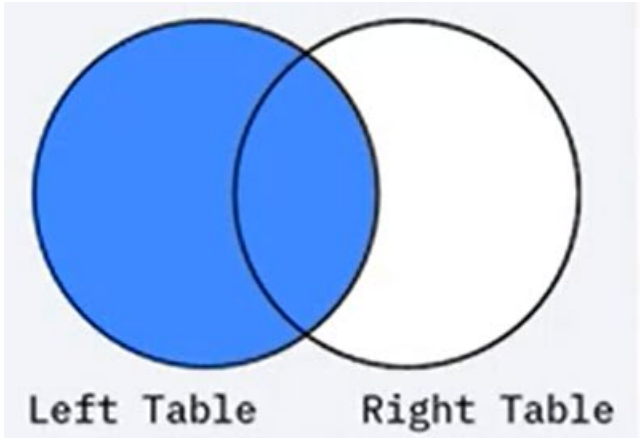
BORROWER_ID	LASTNAME	COUNTRY
D1	SMITH	CA
D2	SANDLER	CA
D3	SOMMERS	CA
D4	ARDEN	CA
D5	XIE	CA
~~D6~~	~~PETERS~~	~~CA~~
~~D7~~	~~LI~~	~~CA~~
~~D8~~	~~WONG~~	~~CA~~
~~D10~~	~~KIEVA~~	~~CA~~

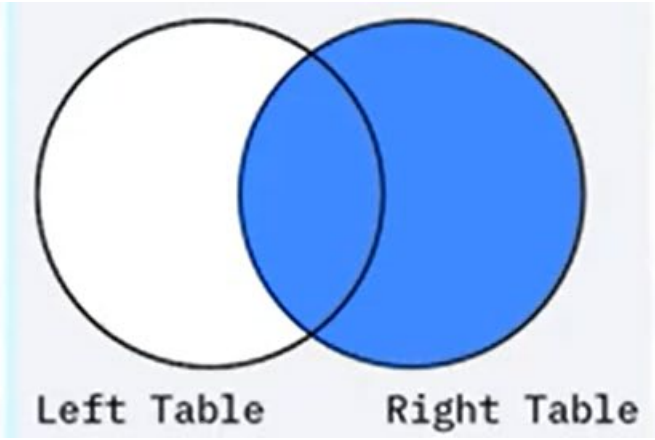
BORROWER_ID	LOAN_DATE
D1	11/24/2010
D2	11/24/2010
D3	11/24/2010
D4	11/24/2010
D5	11/24/2010
~~D9~~	~~11/24/2010~~

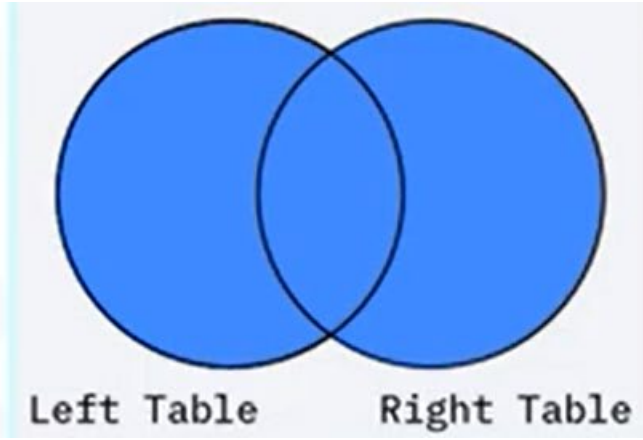
BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010

Outer Join

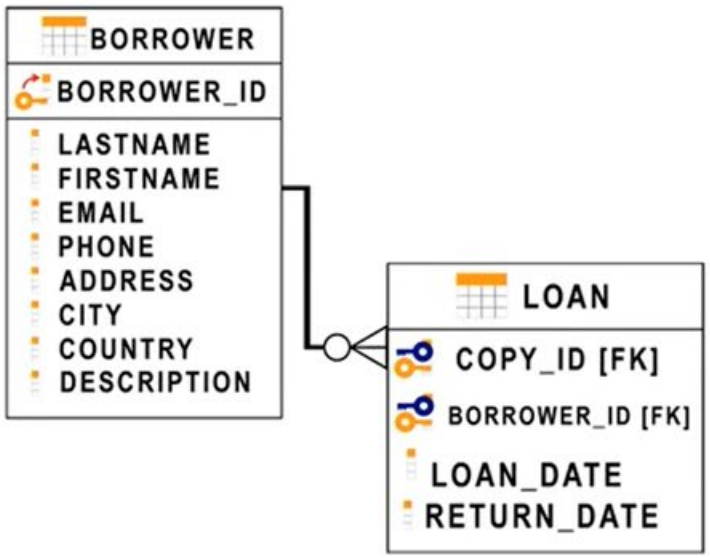
- Retorna información que no este compartida entre las tablas
  - Left Outer
  - Right Outer
  - Full Outer







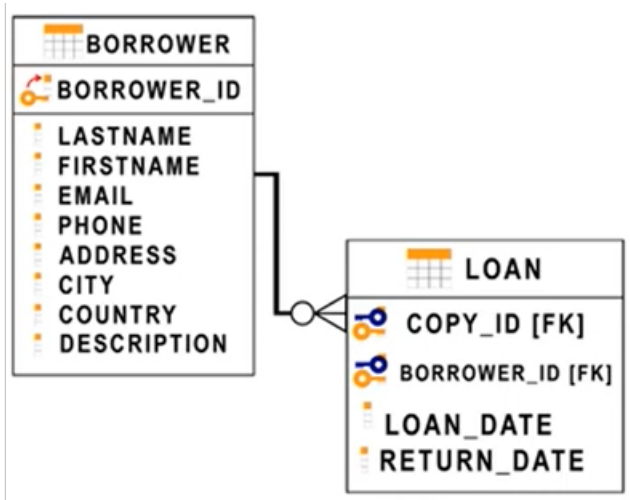
Left Outer Join




```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,
       L.BORROWER_ID, L.LOAN_DATE
FROM BORROWER B LEFT JOIN LOAN L
    ON B.BORROWER_ID = L.BORROWER_ID
```

BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010
D6	PETERS	CA	NULL	NULL
D7	LI	CA	NULL	NULL
D8	WONG	CA	NULL	NULL

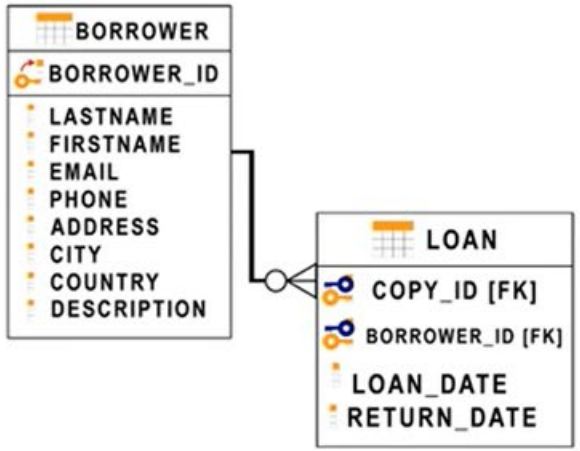
Right Outer Join




```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,
       L.BORROWER_ID, L.LOAN_DATE
FROM BORROWER B RIGHT JOIN LOAN L
    ON B.BORROWER_ID = L.BORROWER_ID
```

BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010
NULL	NULL	NULL	D9	11/24/2010

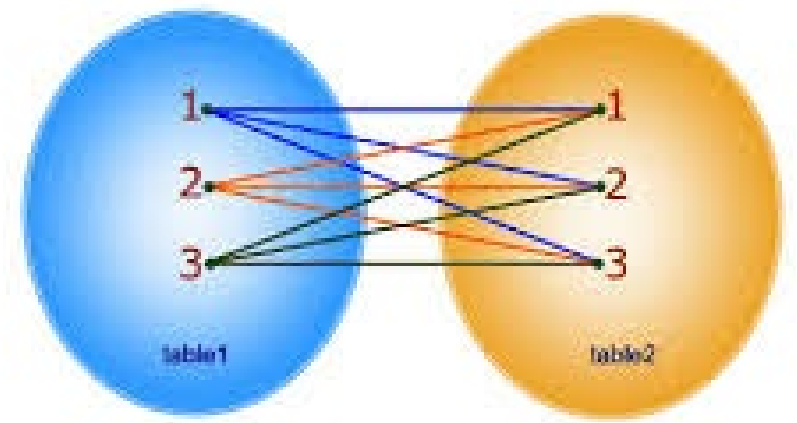
Full Joint




```
SELECT B.BORROWER_ID, B.LASTNAME, B.COUNTRY,
       L.BORROWER_ID, L.LOAN_DATE
FROM BORROWER B FULL JOIN LOAN L
    ON B.BORROWER_ID = L.BORROWER_ID
```

BORROWER_ID	LASTNAME	COUNTRY	BORROWER_ID	LOAN_DATE
D1	SMITH	CA	D1	11/24/2010
D2	SANDLER	CA	D2	11/24/2010
D3	SOMMERS	CA	D3	11/24/2010
D4	ARDEN	CA	D4	11/24/2010
D5	XIE	CA	D5	11/24/2010
D6	PETERS	CA	NULL	NULL
D7	LI	CA	NULL	NULL
D8	WONG	CA	NULL	NULL
NULL	NULL	NULL	D9	11/24/2010

CROSS JOIN ó CARTESIAN JOIN




```sql
SELECT column_name(s)
FROM table1
CROSS JOIN table2;
```

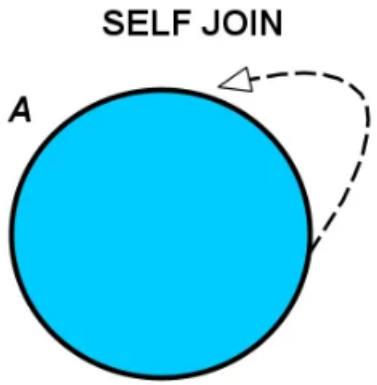
Meals
Omlet
Fried Egg
Sausage

Drinks
Orange Juice
Tea
Cofee

CROSS JOIN

MealName	DrinkName
Omlet	Orange Juice
Fried Egg	Orange Juice
Sausage	Orange Juice
Omlet	Tea
Fried Egg	Tea
Sausage	Tea
Omlet	Coffee
Fried Egg	Coffee
Sausage	Coffee

Self Join




```
SELECT column_name(s)
FROM table1 T1, table1 T2
WHERE condition;
```

Color	Name	Color
Blue	John	Red
Green	Alex	Blue
Red	Simon	Green

+

Color	Name	Color
Blue	John	Red
Green	Alex	Blue
Red	Simon	Green

=

Name	Secret_Santa
John	Simon
Alex	John
Simon	Alex

Preguntas y Lab

1. Select the names and job start dates of all employees who work for the department number 5
2. Select the names, job start dates, and job titles of all employees who work for the department number 5
3. Perform a Left Outer Join on the EMPLOYEES and DEPARTMENT tables and select employee id, last name, department id and department name for all employees
4. Re-write the previous query but limit the result set to include only the rows for employees born before 1980
5. Re-write the previous query but have the result set include all the employees but department names for only the employees who were born before 1980
6. Perform a Full Join on the EMPLOYEES and DEPARTMENT tables and select the First name, Last name and Department name of all employees
7. Re-write the previous query but have the result set include all employee names but department id and department names only for male employees

```
SELECT E.F_NAME,E.L_NAME, JH.START_DATE
FROM EMPLOYEES as E
INNER JOIN JOB_HISTORY AS JH on E.EMP_ID=JH.EMPL_ID
WHERE E.DEP_ID='5';
```

```
SELECT E.F_NAME,E.L_NAME, JH.START_DATE, J.JOB_TITLE
FROM EMPLOYEES as E
INNER JOIN JOB_HISTORY AS JH on E.EMP_ID=JH.EMPL_ID
INNER JOIN JOBS as J on E.JOB_ID=J.JOB_IDENT
WHERE E.DEP_ID='5';
```

```
SELECT E.EMP_ID,E.L_NAME,E.DEP_ID,D.DEP_NAME
FROM EMPLOYEES as E
LEFT OUTER JOIN DEPARTMENTS AS D ON E.DEP_ID=D.DEPT_ID_DEP;
```

```
SELECT E.EMP_ID,E.L_NAME,E.DEP_ID,D.DEP_NAME
FROM EMPLOYEES as E
LEFT OUTER JOIN DEPARTMENTS AS D ON
E.DEP_ID=D.DEPT_ID_DEP
WHERE YEAR(E.B_DATE) < 1980;
```

```
select E.EMP_ID,E.L_NAME,E.DEP_ID,D.DEP_NAME
from EMPLOYEES AS E
LEFT OUTER JOIN DEPARTMENTS AS D ON
E.DEP_ID=D.DEPT_ID_DEP
AND YEAR(E.B_DATE) < 1980;
```

```
select E.F_NAME,E.L_NAME,D.DEP_NAME
from EMPLOYEES AS E
FULL OUTER JOIN DEPARTMENTS AS D ON
E.DEP_ID=D.DEPT_ID_DEP;
```

```
select E.F_NAME,E.L_NAME,D.DEPT_ID_DEP, D.DEP_NAME
from EMPLOYEES AS E
FULL OUTER JOIN DEPARTMENTS AS D ON
E.DEP_ID=D.DEPT_ID_DEP AND E.SEX = 'M';
```

Gracias

- Mas comandos

https://www.w3schools.com/sql/