DB → ER → SQL

marco milanesio
MScDSAI 2023-2024

Database

- **Structured** information
- Tables / documents
- relational, hierarchical, network, …
- SQL, NoSQL, Distributed, …
- Relations / links / pointers
- DB Server, DBMS, …

Relational Algebra

- Sets
  - Set Union, Set Difference, Cartesian product
- Unary operators
  - Projection, Selection, Rename
- Binary operators
  - Join (inner, outer, left-, right-, …), division

Tuple relational calculus

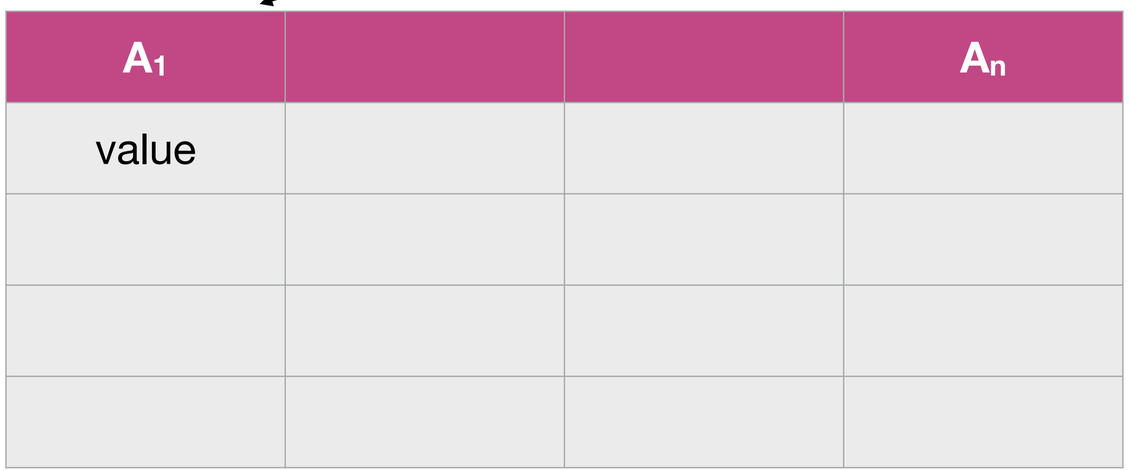
- not python’s tuple!!!
- foundation of a **declarative** query language
- tuple (t) == "finite sequence of attributes"
- orderer pairs (attribute, value)
- t.name = "Marco" —> attribute name of tuple t has value "Marco"
- Person(t) —> tuple t is present in the relation Person

Relational model

Relational Variable

R

attributes

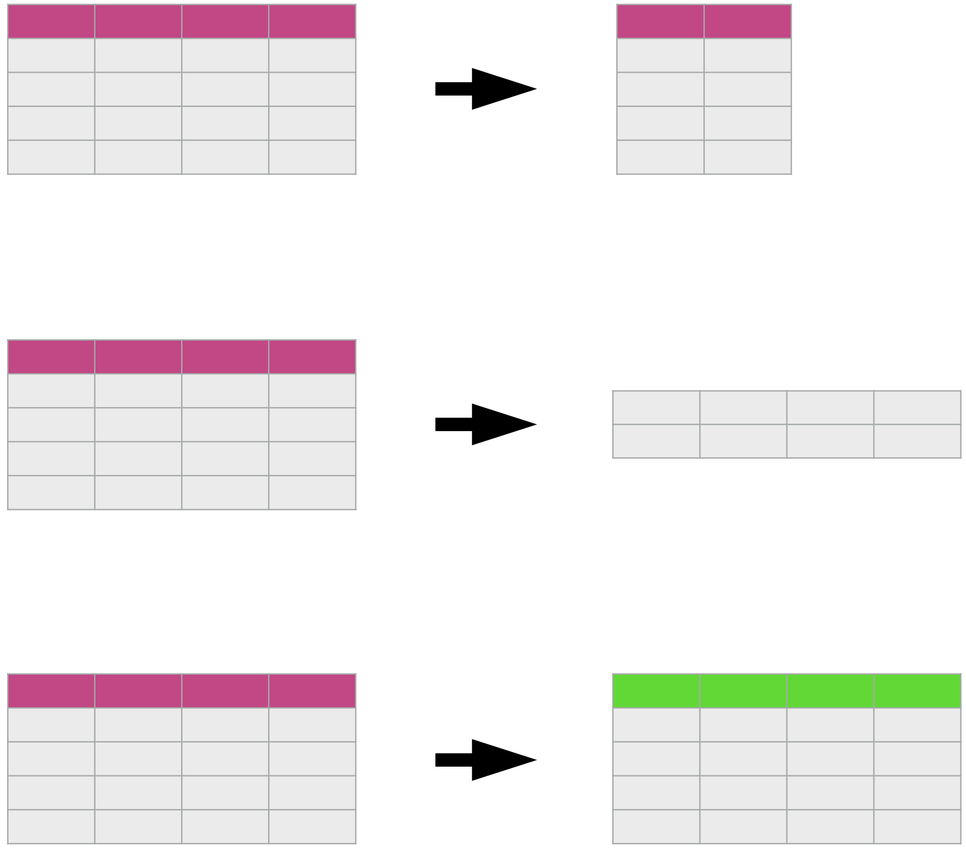


A_1			A_n
value			

RELATION

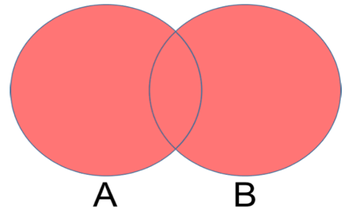
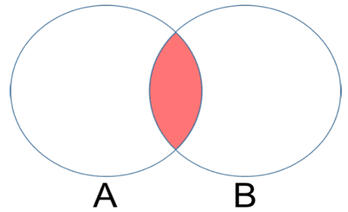
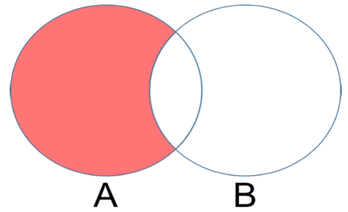
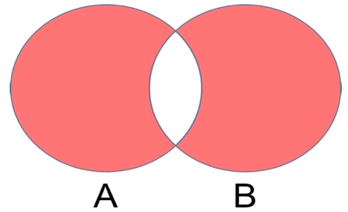
Unary operators (ER)

- **Projection**
  - **of attributes**
- **Selection**
  - **of tuples**
- **Rename**
  - **of attributes**



Binary operators

- **Cartesian Products**
- **Difference**

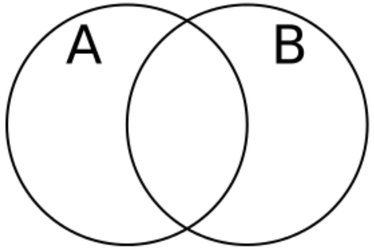
Set Operation	Venn Diagram	Interpretation
Union		$A \cup B$, is the set of all values that are a member of *A*, or *B*, or both.
Intersection		$A \cap B$, is the set of all values that are members of both *A* and *B*.
Difference		$A \setminus B$, is the set of all values of *A* that are not members of *B*
Symmetric Difference		$A \triangle B$, is the set of all values which are in one of the sets, but not both.

https://cdn-images-1.medium.com/max/1600/0*a02OPI3-TnbKXyub.png

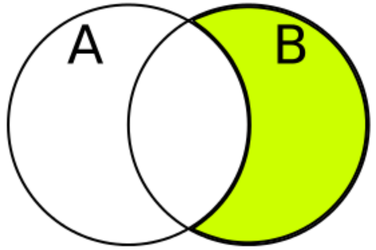
SQL JOINS

Arranged in a Karnaugh Map by Jason Charney (jrcharney@gmail.com)

No join (∅)

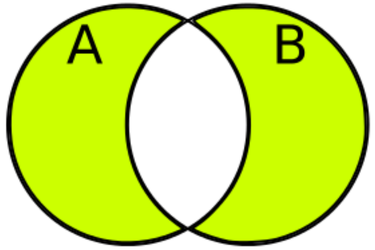


[Exclusive] Right Join (¬A)



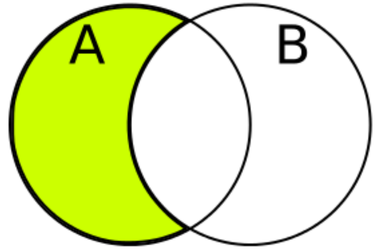

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

[Exclusive] Full Join (A⊕B)



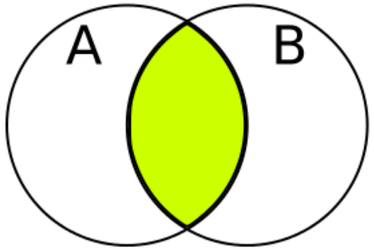

```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
WHERE B.key IS NULL
OR A.key IS NULL
```

[Exclusive] Left Join (¬B)



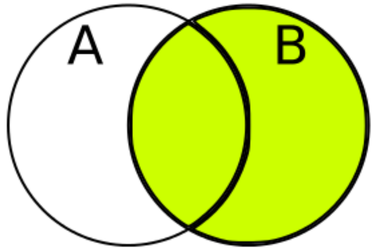

```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

Inner Join (A∧B)



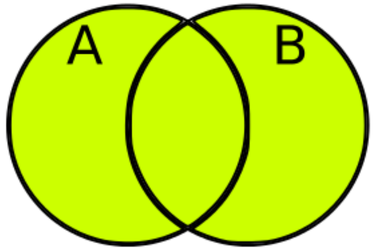

```
SELECT *
FROM A
INNER JOIN B ON A.key = B.key
```

[Inclusive] Right Join (B)



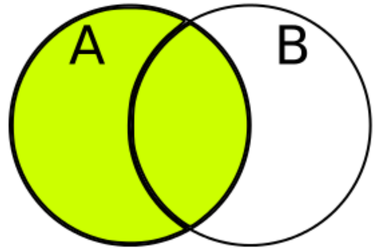

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
```

[Inclusive] Full Join (A∨B)




```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
```

[Inclusive] Left Join (A)

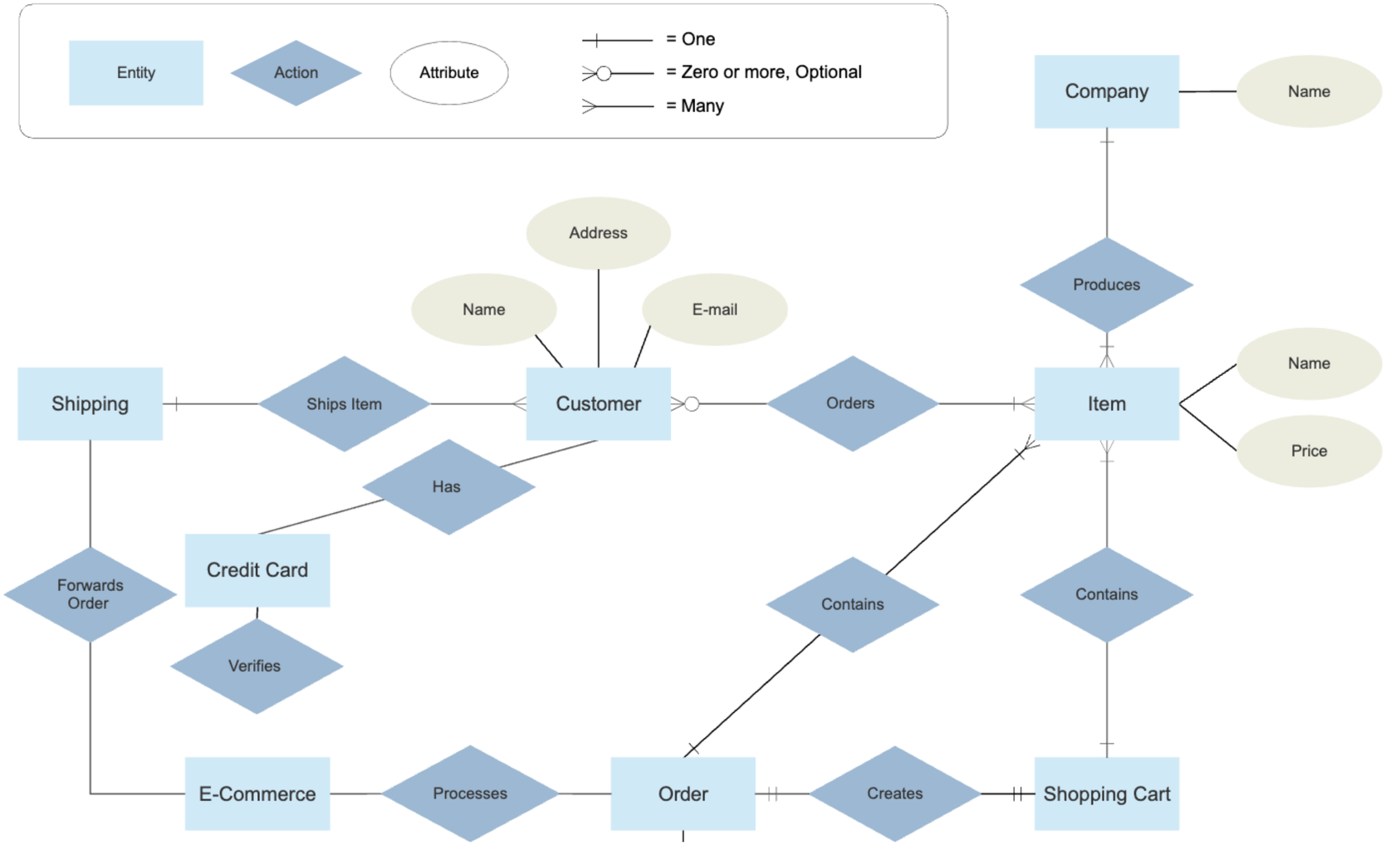



```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
```

https://i.pinimg.com/originals/bc/0c/8b/bc0c8ba4d12051502a68bade9bba4bc5.png

ER

- **Entity relationship**
- Table-based views
- SQL
- RDBMS



https://www.smartdraw.com/entity-relationship-diagram/

SQL

- **Structured** Query Language (born ~1970)
- Relational algebra
- Data Query Language
  - DQL: queries
- Data Definition Language
  - DDL: schemas
- Data Control Language
  - DCL: access
- Data Manipulation Language
  - DML: insert, update, delete

SQL

- **Direct**
  - **SQL Commands directly into Python code**
  - **Python interpreter converts and executes**
- **ORM**
  - **SqlAlchemy**

3 major databases

- **MySQL (MariaDB)**
  - **mysql-connector-python**
- **PostGreSQL**
  - **psycopg2**
- **SQLite**
  - **sqlite3 (in core library !!)**

SQLite

- **self-contained,**
- **serverless,**
- **zero-configuration,**
- **transactional.**

SQlite3

- **Easy prototyping**
- **Fully ER**
- **Works as-is**
- **CLI + GUI available for all OSs**
  - **https://www.sqlite.org/index.html**
- **Optimize later**

SQLAlchemy

- **ORM**
  - **Object Relational Mapper**
- **Python objects**
- **NO Sql commands directly ingested**
- **Whole architecture changes**

SQLAlchemy

- **ORM**
  - **Object Relational Mapper**
- **Python objects**
- **NO Sql commands directly ingested**
- **Whole architecture changes**
- **Any (i.e., ALL) database**
  - **https://www.sqlalchemy.org/features.html**

Hands-on