Databases.

Relational Databases.

RDBMS (Relational Database Management System) crash course …

Relational Model

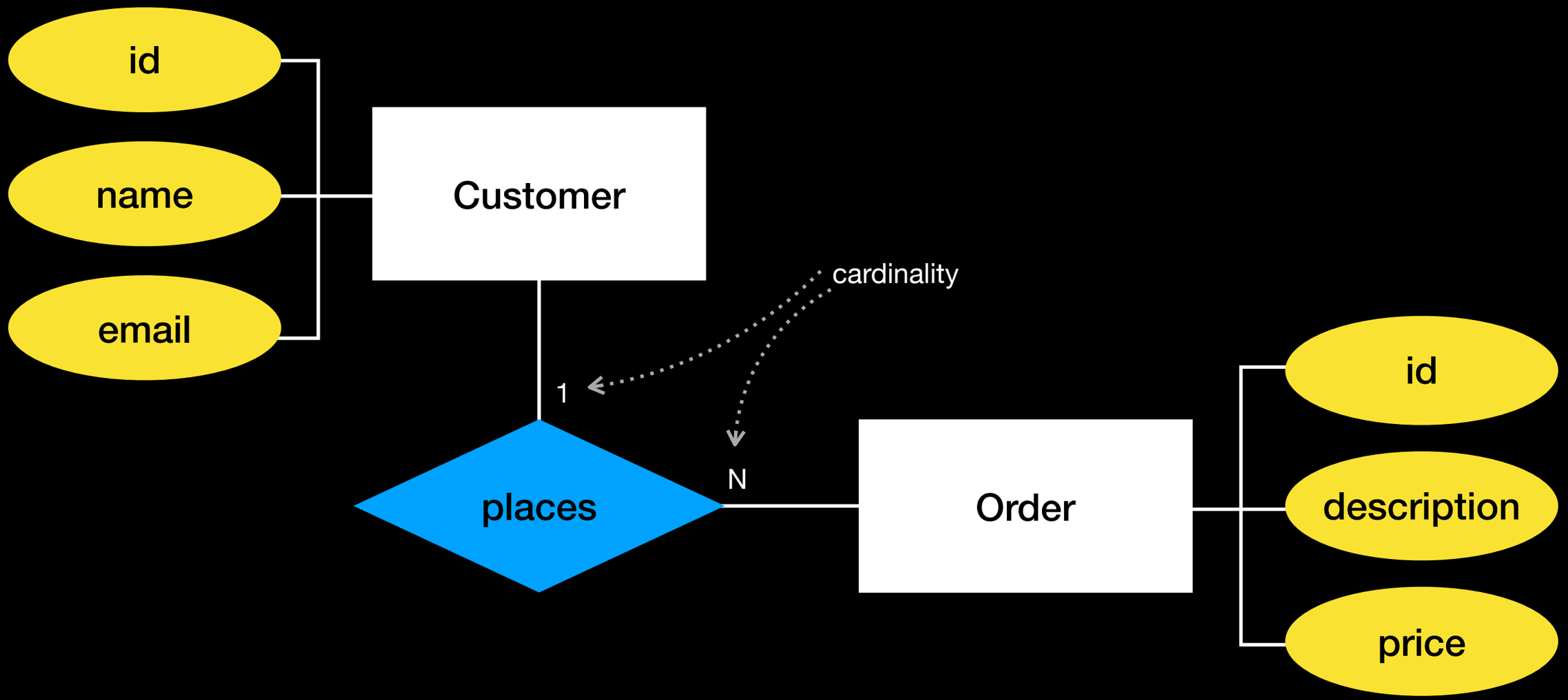
- Approach to describing and modeling relationships between data
- Individual records are grouped as **Tuples**
  - e.g., (“Jane Lee”, 12345678, “jane.lee@foo.edu")
- Collections of tuples are grouped into **Relations**
  - Relations also describe the attributes of its records
  - e.g., (Name: Text, Id: Integer, Email: Text)

Relational Algebra

- Defines **mathematical operations** on/over relations
  - **Selection** filters out select records
  - **Projection** selects only specific attributes
  - **Join** combines data from multiple relations
- Supports **rigorous reasoning** about data analysis and manipulation!

Entity-Relationship (ER) model

- A way of visually describing relationships between data



Relations = Tables

- Practically, a **Table** represents a relation in a database
  - Each **Row** in a table is a record/tuple
    - **Columns** of the table delineate attributes of the data
- An RDBMS is a database designed around principles established by the relational model
  - Operations generally align with relational algebra

Structured Query Language (SQL)

- Standard language used to interact with RDBMSs
  - Declarative; i.e., it describes **what** operation to perform, not **how** to carry it out (lots of room for optimization!)
  - Operations: creating tables, inserting/updating/deleting records, querying (fetching) records, etc.
- Most major RDBMSes (e.g., MySQL, PostgreSQL, Oracle, MS SQL Server) support their own “dialect” of SQL

SQLite

“an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.”

Sample Database: Orders

```sql
CREATE TABLE customer (
    id INTEGER PRIMARY KEY,
    name TEXT,
    email TEXT
);

CREATE TABLE purchase_order (
    id INTEGER PRIMARY KEY,
    description TEXT,
    price REAL,
    customer_id INT,
    FOREIGN KEY (customer_id) REFERENCES customer(id)
);
```

Sample Queries

```sql
SELECT * FROM customer;

SELECT * FROM purchase_order
WHERE customer_id = 1234;
```

```sql
SELECT c.name, o.description
FROM customer c
JOIN purchase_order o ON c.id = o.customer_id
WHERE o.id = 2345;
```

Sample Updates

```
INSERT INTO customer (id, name, email)
VALUES (1234, 'Jane Doe', 'jane@foo.com');

UPDATE purchase_order
SET description = 'Atomizer'
WHERE id = 2345;

DELETE FROM purchase_order WHERE id = 2345;
```