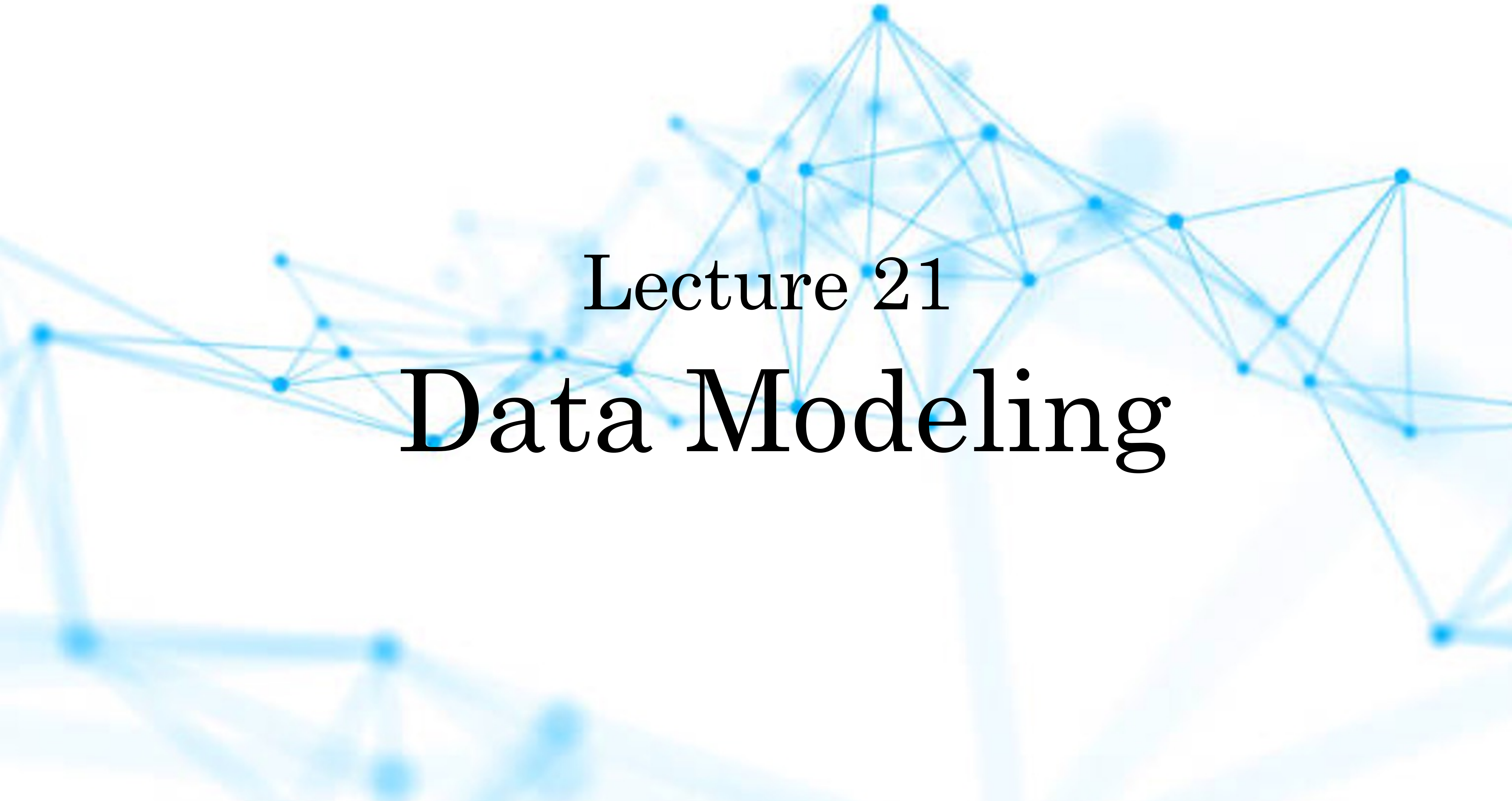
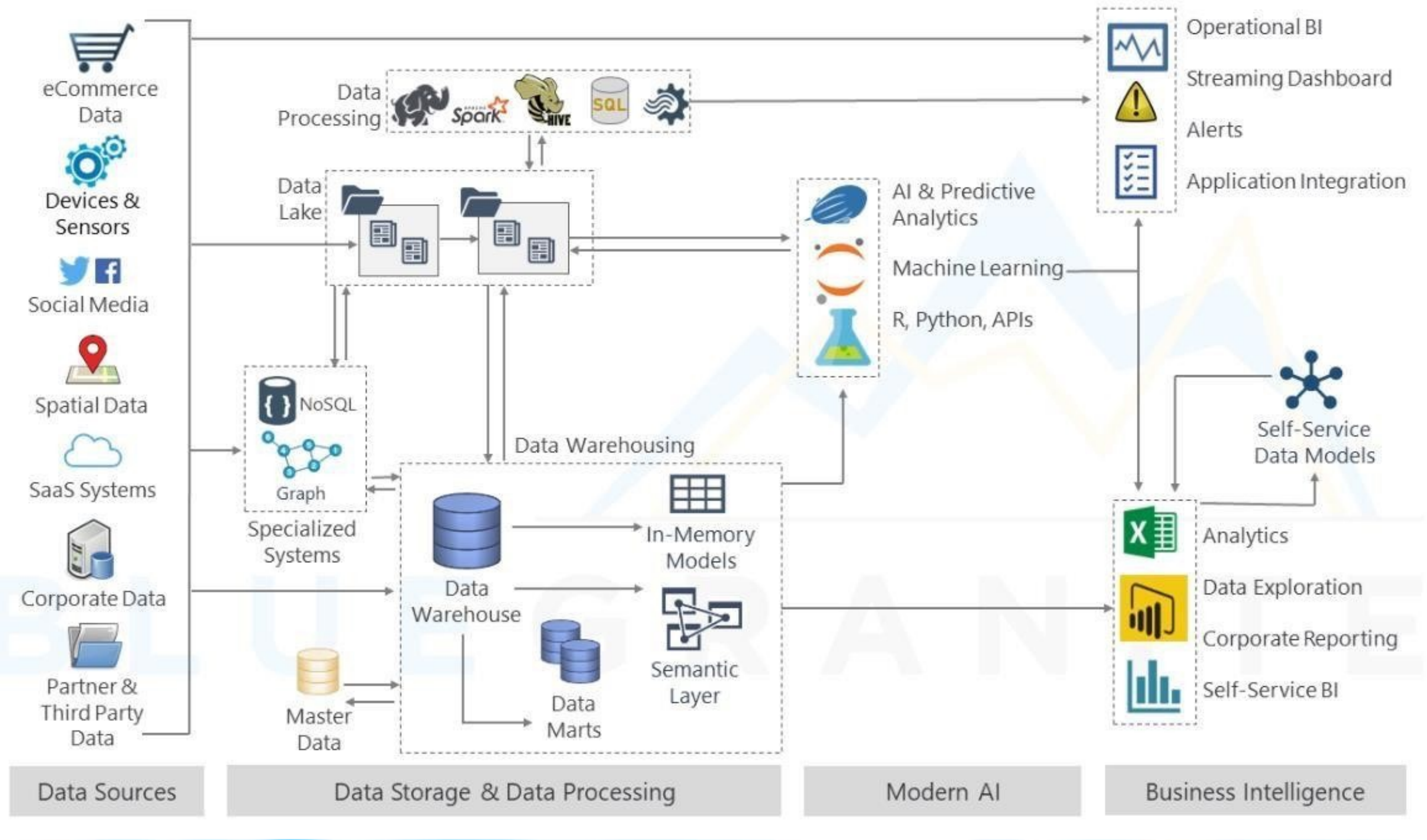


Lecture 21

Data Modeling

Data Architecture



Data Modeling

“Data Modeling is an abstraction that organizes elements of data and how they will relate to each other”

– Wikipedia

Example: Spreadsheets for household

- You define rows and columns
- You structure your data

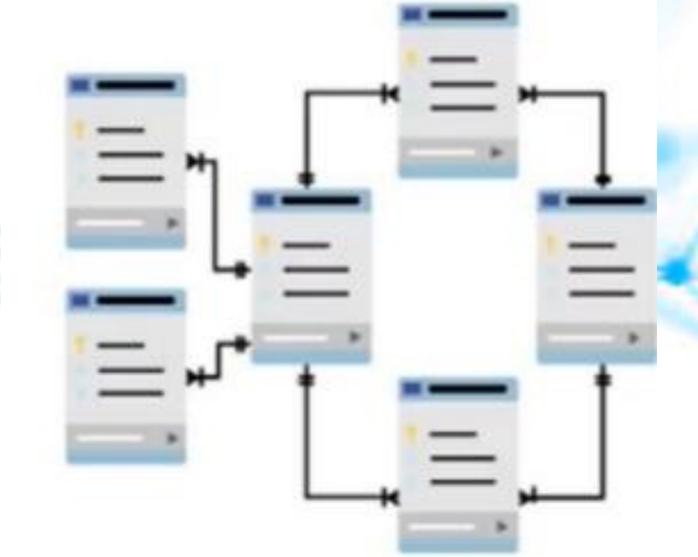
Process of Data Modeling

The process of data modeling is to

- Organize data into databases.
- To ensure that your data is persistent.
- To ensure that it is easily useable by you and your organization.

Data Modeling is also called **database modeling.**

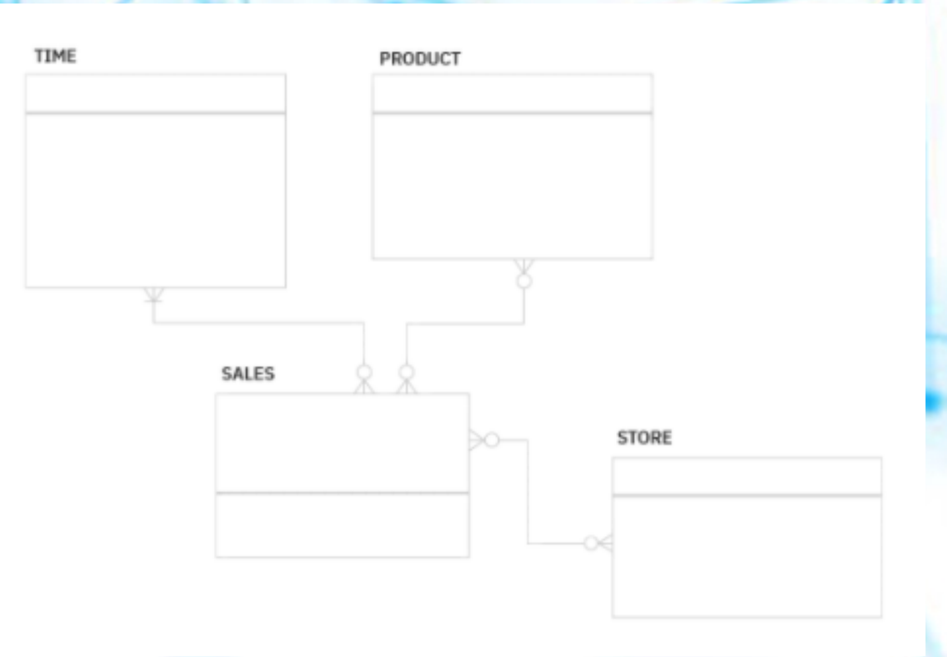
Data Modeling



- Process to support business and user applications
- Gather requirements
- Conceptual Data Modeling
- Logical Data Modeling
- Physical Data Modeling

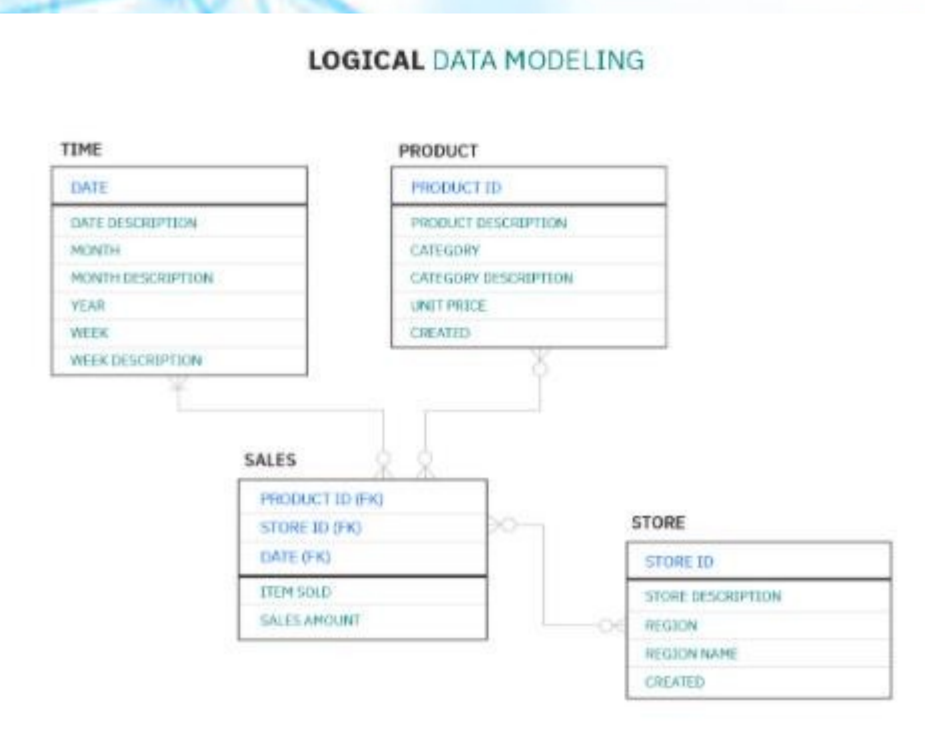
Conceptual Data Modeling

- Offers a big view picture of the business structure
- Created as part of the process of gathering initial project requirements
- Typically includes **entity classes,** their **characteristics** and **constraints** and the **relationships** between them



Logical Data Modeling

- Greater detail about the system
- More concerned about system implementation
- Data attributes in each entity are defined
- Data attributes, such as **data types** and **lengths and** relationships between **entities** are indicated



LOGICAL DATA MODELING

TIME
DATE
DATE DESCRIPTION
MONTH
MONTH DESCRIPTION
YEAR
WEEK
WEEK DESCRIPTION

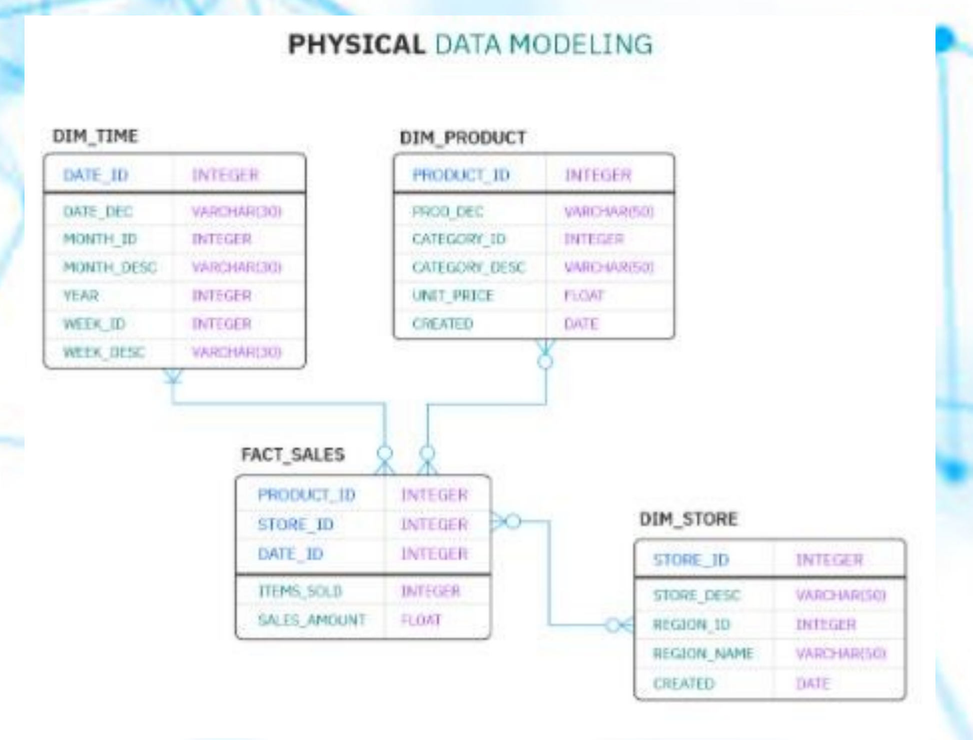
PRODUCT
PRODUCT ID
PRODUCT DESCRIPTION
CATEGORY
CATEGORY DESCRIPTION
UNIT PRICE
CREATED

SALES
PRODUCT ID (FK)
STORE ID (FK)
DATE (FK)
ITEM SOLD
SALES AMOUNT

STORE
STORE ID
STORE DESCRIPTION
REGION
REGION NAME
CREATED

Physical Data Modeling

- Demonstrates the low-level implementation details
- A finalized design is offered containing data types, primary and foreign keys
- Can include DBMS-specific properties, including performance tuning.



PHYSICAL DATA MODELING

DIM_TIME

DATE_ID	INTEGER
DATE_DEC	VARCHAR(30)
MONTH_ID	INTEGER
MONTH_DESC	VARCHAR(30)
YEAR	INTEGER
WEEK_ID	INTEGER
WEEK_DESC	VARCHAR(30)

DIM_PRODUCT

PRODUCT_ID	INTEGER
PROD_DEC	VARCHAR(50)
CATEGORY_ID	INTEGER
CATEGORY_DESC	VARCHAR(50)
UNIT_PRICE	FLOAT
CREATED	DATE

FACT_SALES

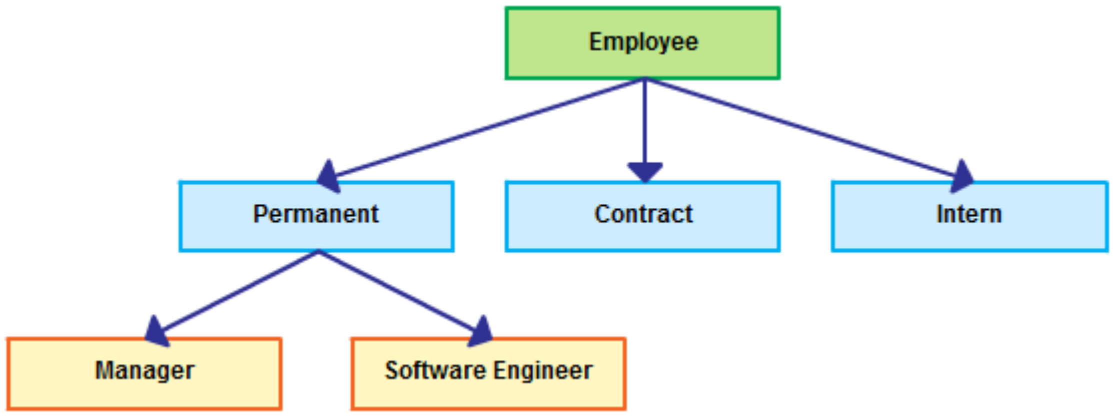
PRODUCT_ID	INTEGER
STORE_ID	INTEGER
DATE_ID	INTEGER
ITEMS_SOLD	INTEGER
SALES_AMOUNT	FLOAT

DIM_STORE

STORE_ID	INTEGER
STORE_DESC	VARCHAR(50)
REGION_ID	INTEGER
REGION_NAME	VARCHAR(50)
CREATED	DATE

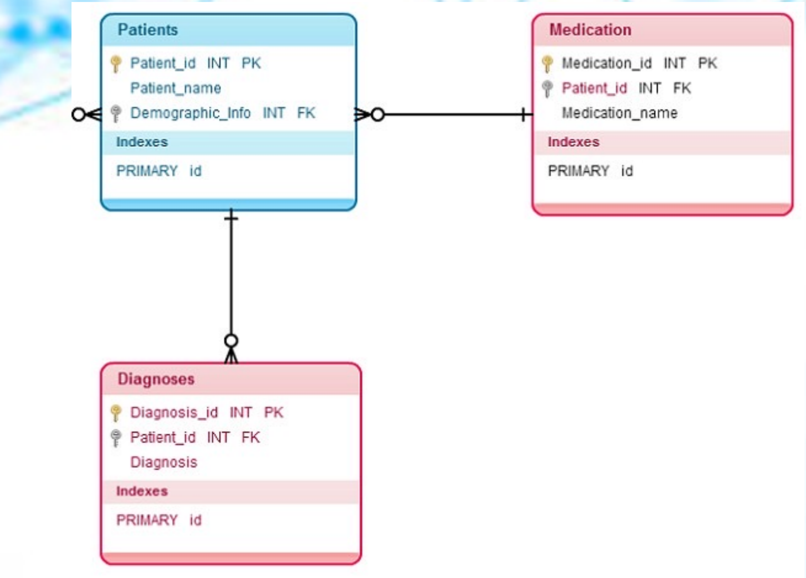
Types of Data Modeling

- Hierarchical Data Models
  - Relationships represented in a tree-like format
  - Each record has a single root/parent and maps to child tables



Types of Data Modeling

- Relational Data Models
  - Data segments are explicitly joined through the use of tables, reducing database complexity.



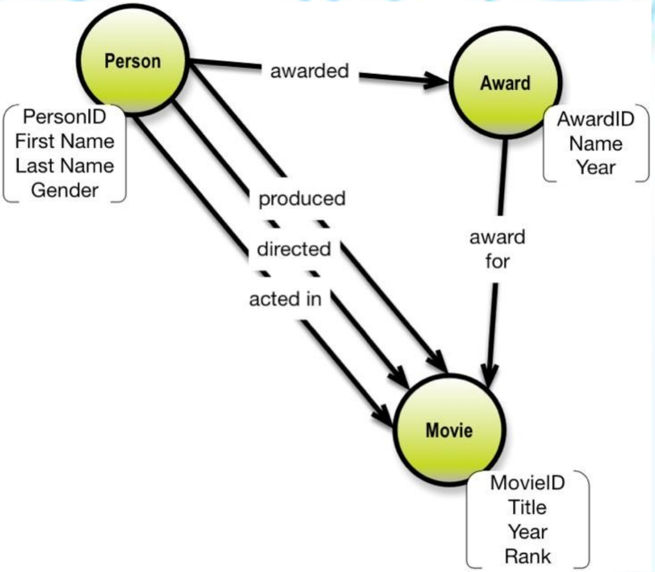
Patients
Patient_id INT PK
Patient_name
Demographic_Info INT FK
Indexes
PRIMARY id

Medication
Medication_id INT PK
Patient_id INT FK
Medication_name
Indexes
PRIMARY id

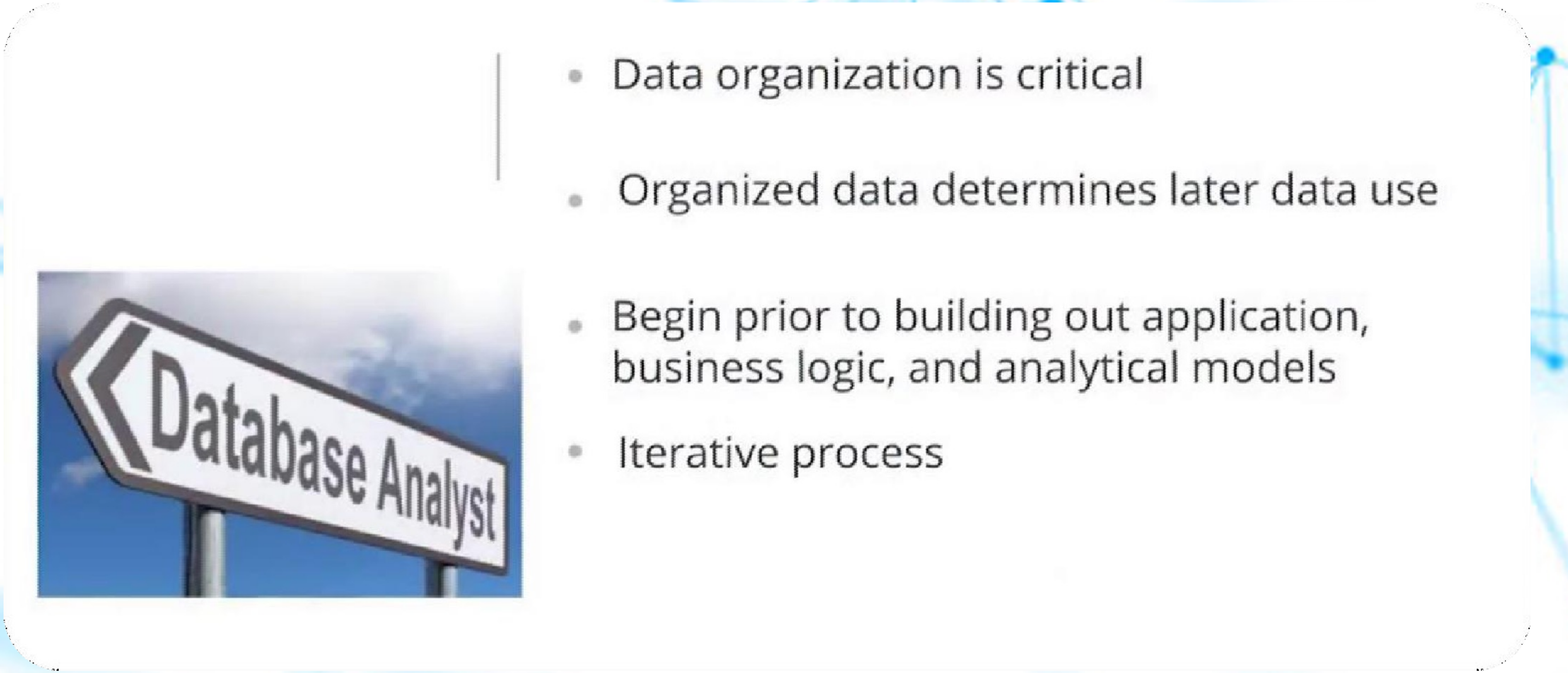
Diagnoses
Diagnosis_id INT PK
Patient_id INT FK
Diagnosis
Indexes
PRIMARY id

Types of Data Modeling

- Graph Data Models
  - Based on Graph Theory
  - Nodes and Edges in a graph are used to represent data



Why is data modeling important?

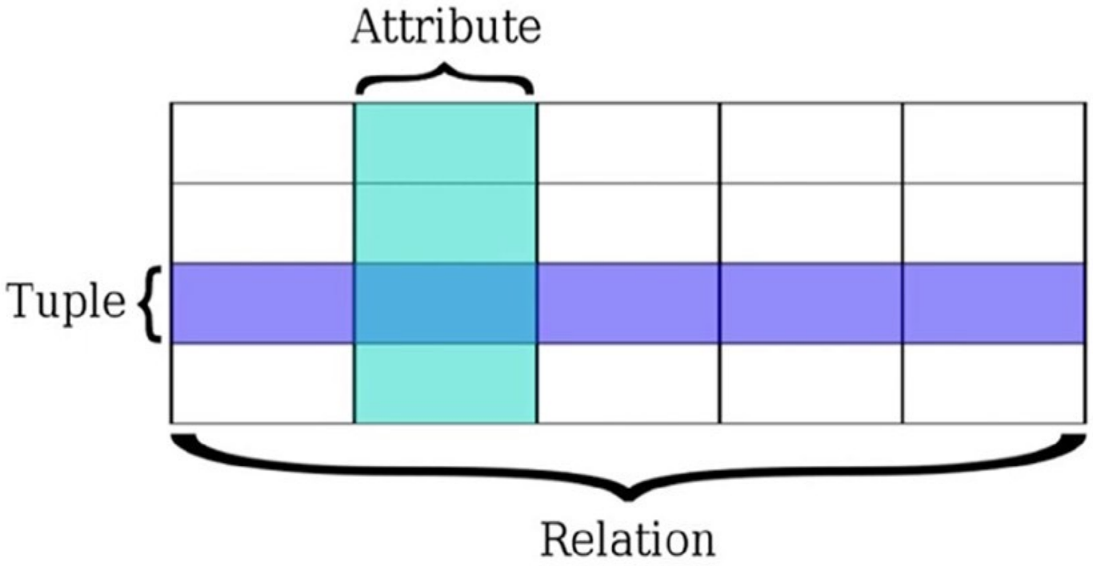


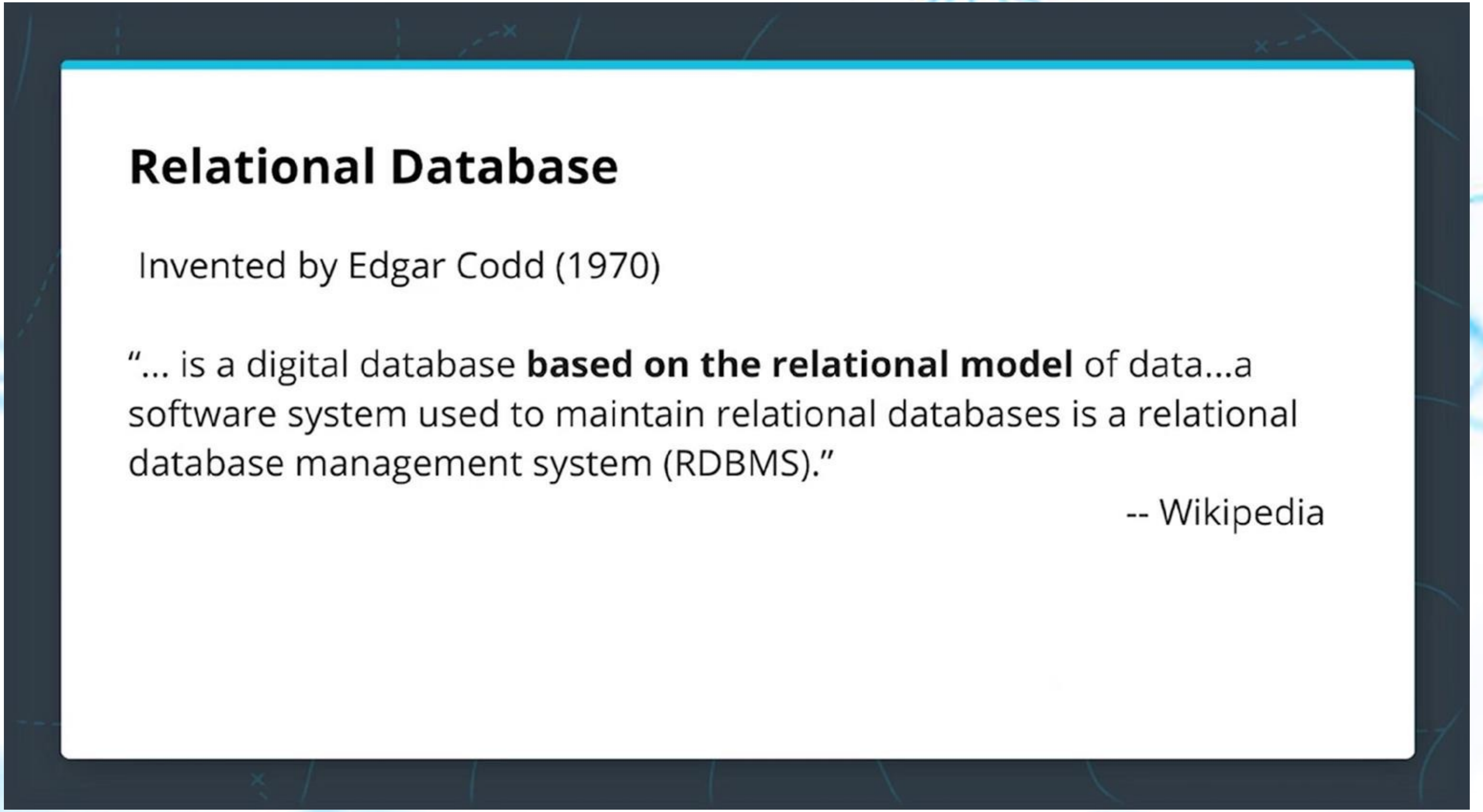
- Data organization is critical
- Organized data determines later data use
- Begin prior to building out application, business logic, and analytical models
- Iterative process

Relational and NoSQL Databases

Relational Model

“This model **organizes data into** one or more tables (or "**relations") of columns and rows**, with a **unique key identifying each row**. Generally, each table represents one "entity type" (such as customer or product).”



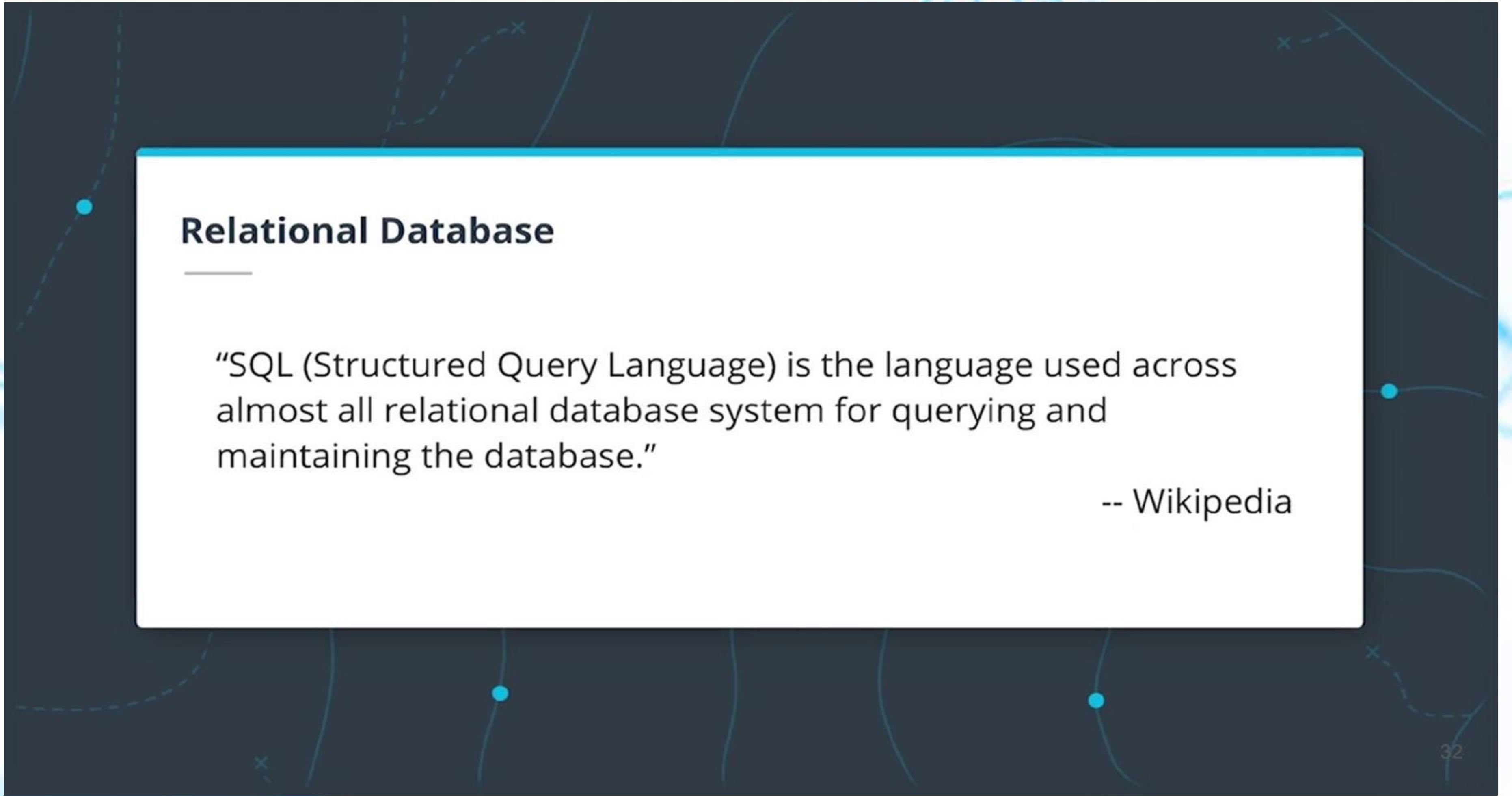


Relational Database

Invented by Edgar Codd (1970)

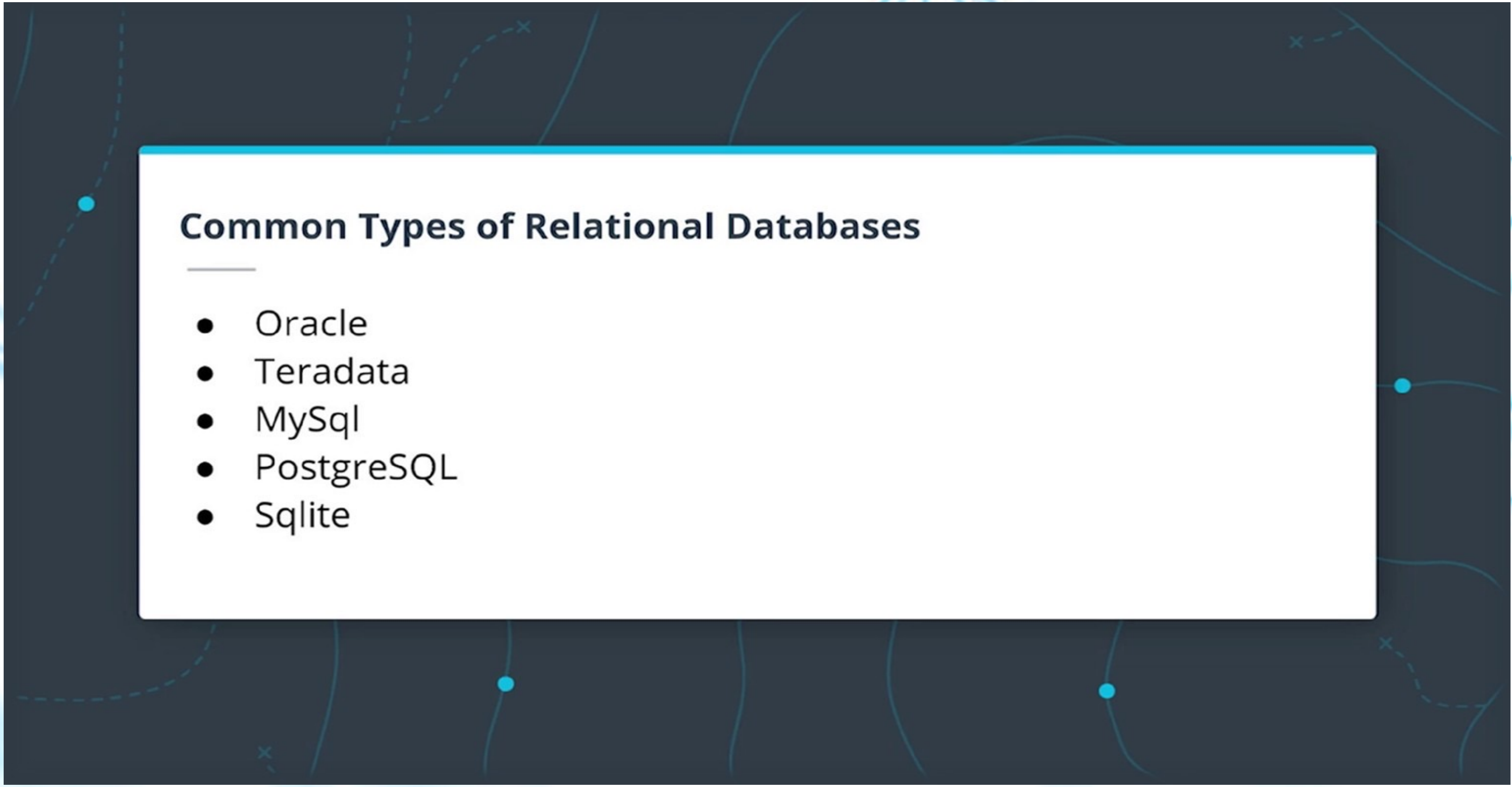
“… is a digital database **based on the relational model** of data…a software system used to maintain relational databases is a relational database management system (RDBMS).”

-- Wikipedia



Relational Database

> "SQL (Structured Query Language) is the language used across almost all relational database system for querying and maintaining the database."
>
> -- Wikipedia

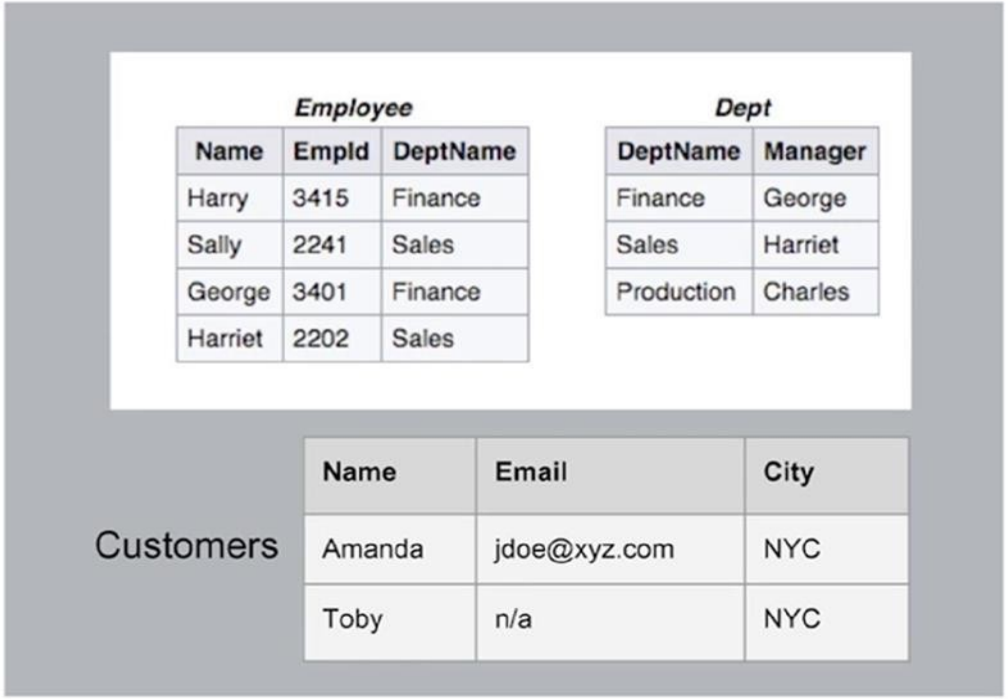


Common Types of Relational Databases

- Oracle
- Teradata
- MySql
- PostgreSQL
- Sqlite

The Basics

- Database/Schema
    - Collection of Tables
- Tables/Relation
    - A group of rows sharing the same labeled elements
        - Customers



Employee

Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales

Dept

DeptName	Manager
Finance	George
Sales	Harriet
Production	Charles

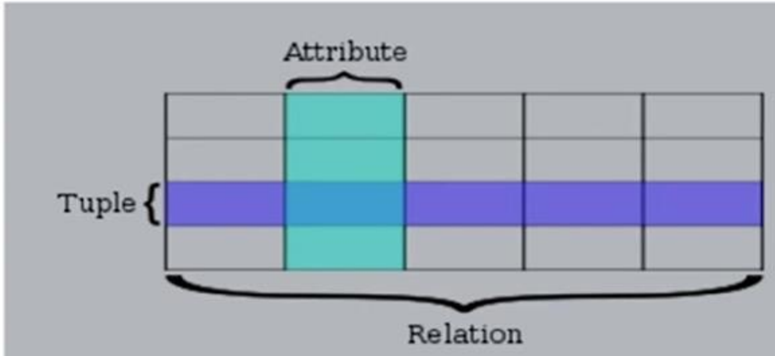
Customers

Name	Email	City
Amanda	jdoe@xyz.com	NYC
Toby	n/a	NYC

The Basics

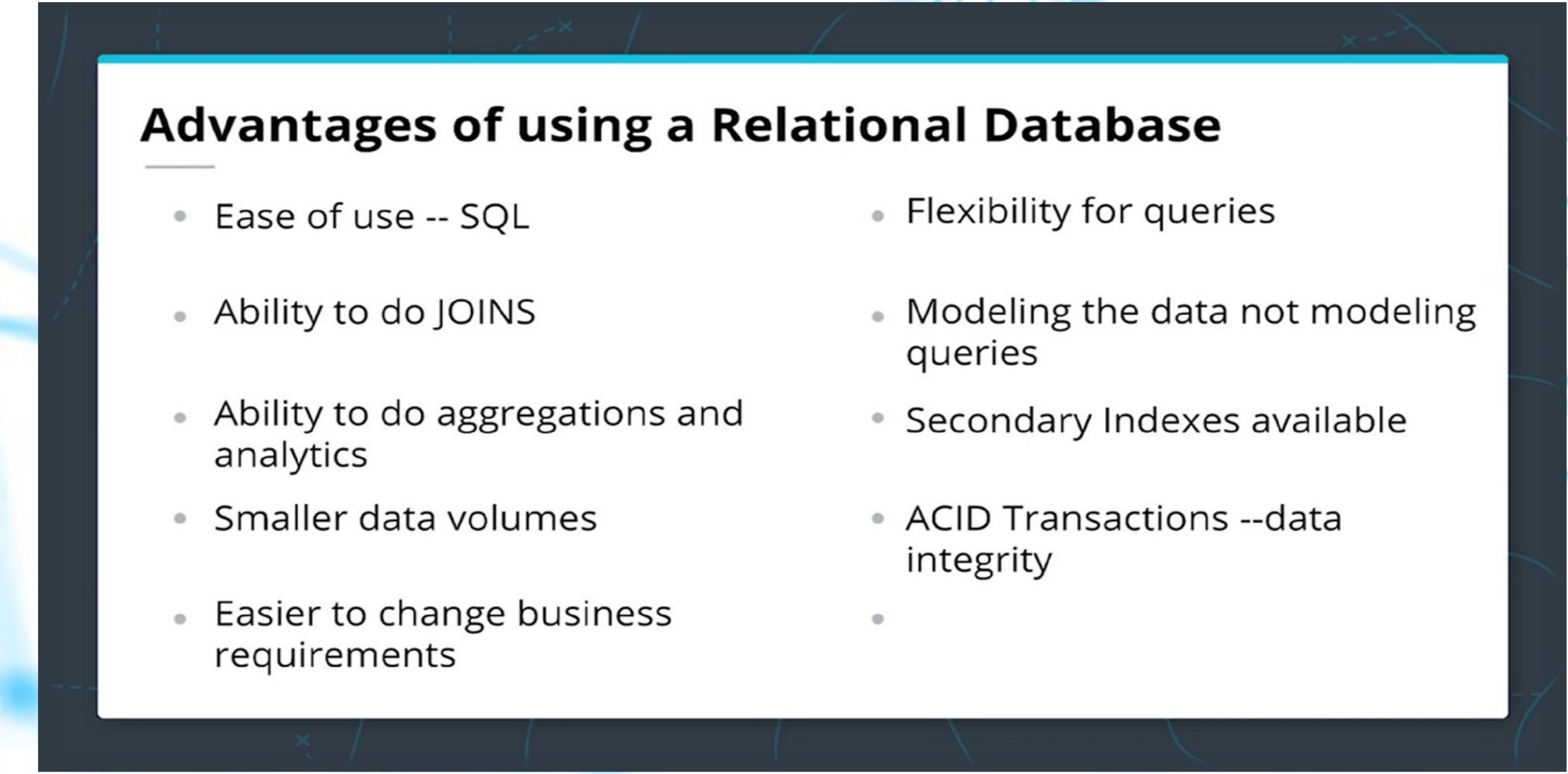
- Columns/Attribute
  - Labeled element
    - Name, email, city

- Rows / Tuple
  - A single item
    - Amanda, jdoe@xyc.com, NYC



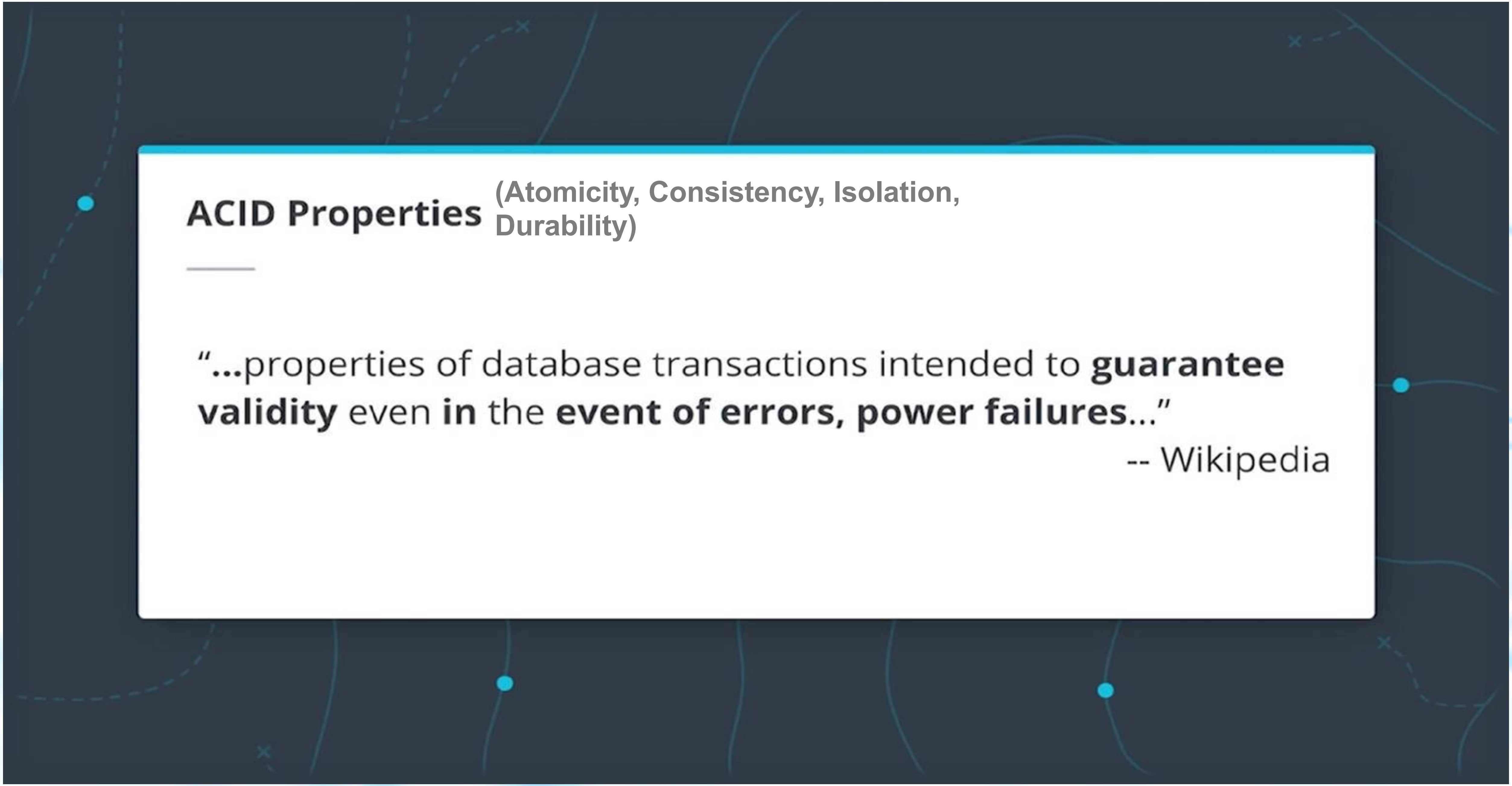
Customers

Name	email	City
Amanda	jdoe@xyz.com	NYC
Toby	n/a	NYC



Advantages of using a Relational Database

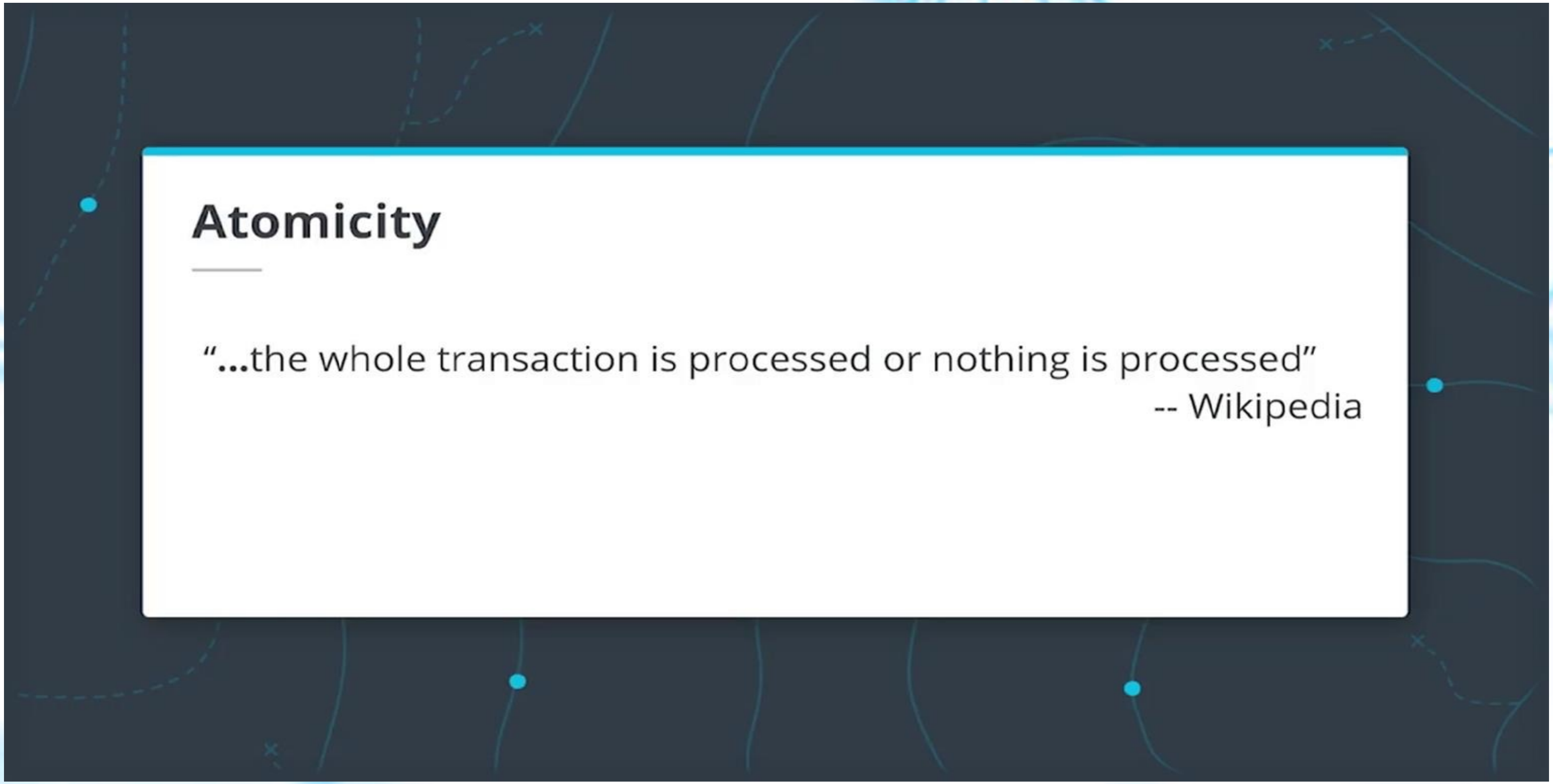
- Ease of use -- SQL
- Ability to do JOINS
- Ability to do aggregations and analytics
- Smaller data volumes
- Easier to change business requirements
- Flexibility for queries
- Modeling the data not modeling queries
- Secondary Indexes available
- ACID Transactions --data integrity



ACID Properties (Atomicity, Consistency, Isolation, Durability)

“**...**properties of database transactions intended to **guarantee validity** even **in** the **event of errors, power failures**...”

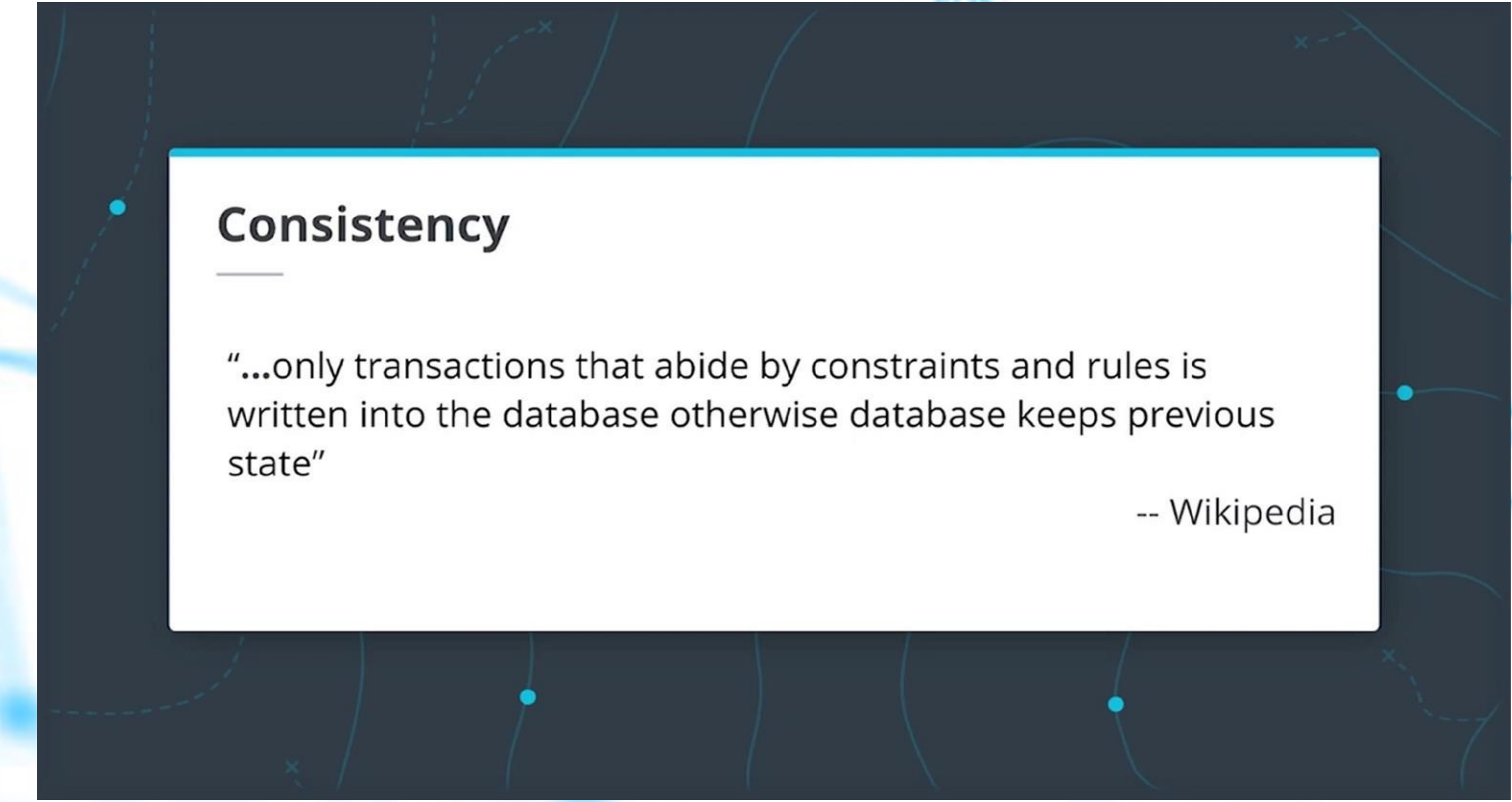
-- Wikipedia



Atomicity

"...the whole transaction is processed or nothing is processed"

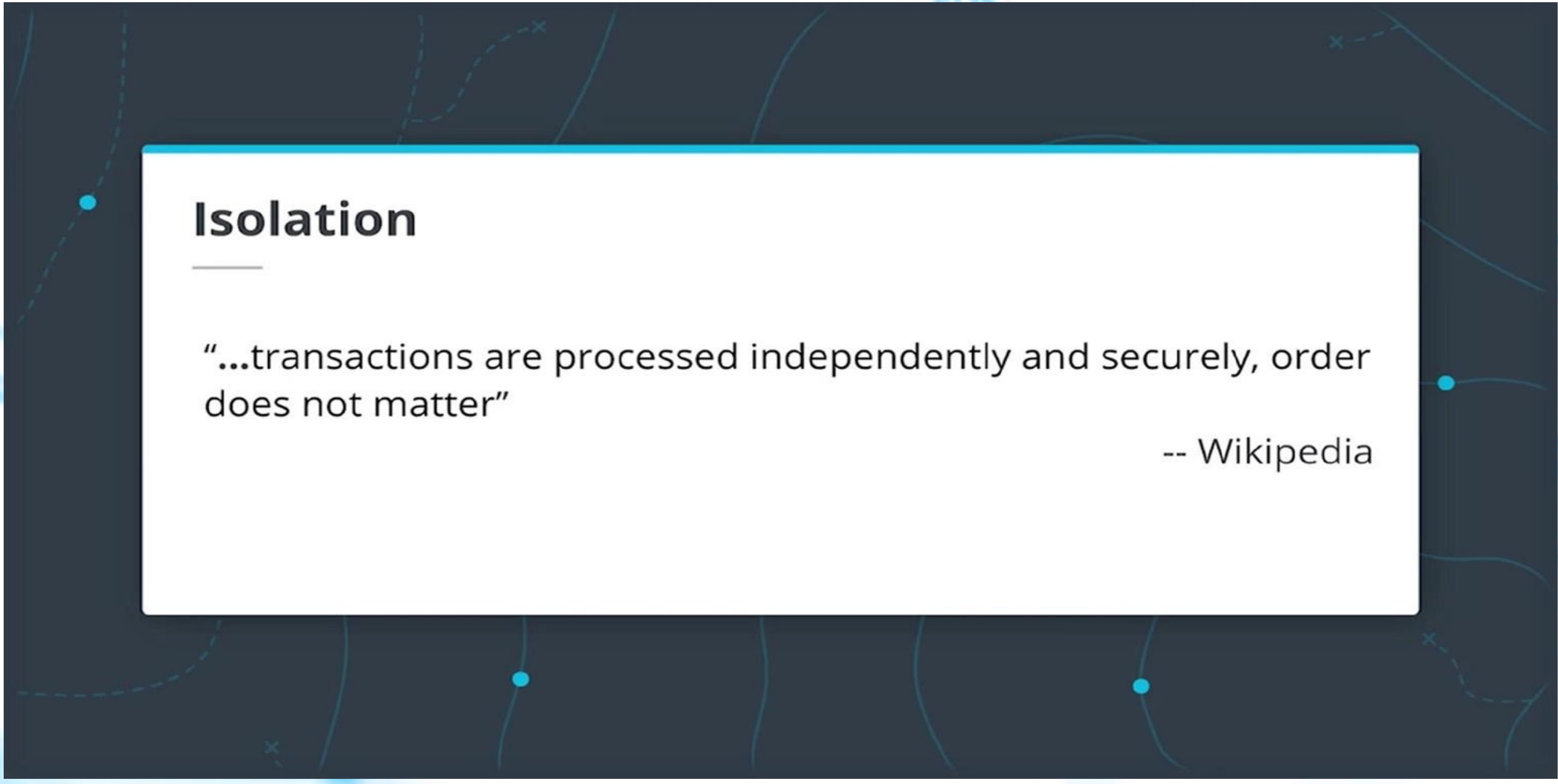
-- Wikipedia



Consistency

"...only transactions that abide by constraints and rules is written into the database otherwise database keeps previous state"

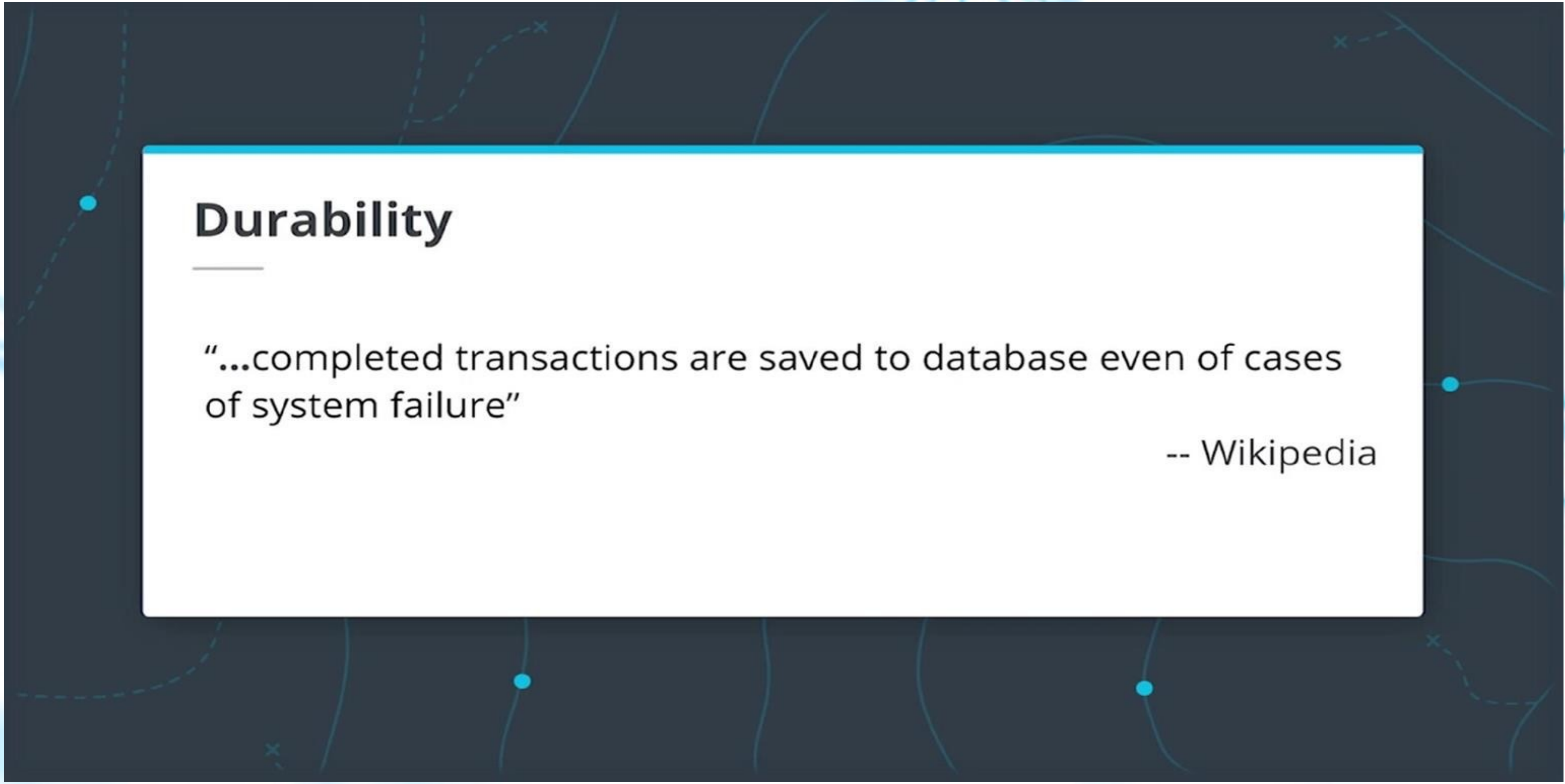
-- Wikipedia



Isolation

"...transactions are processed independently and securely, order does not matter"

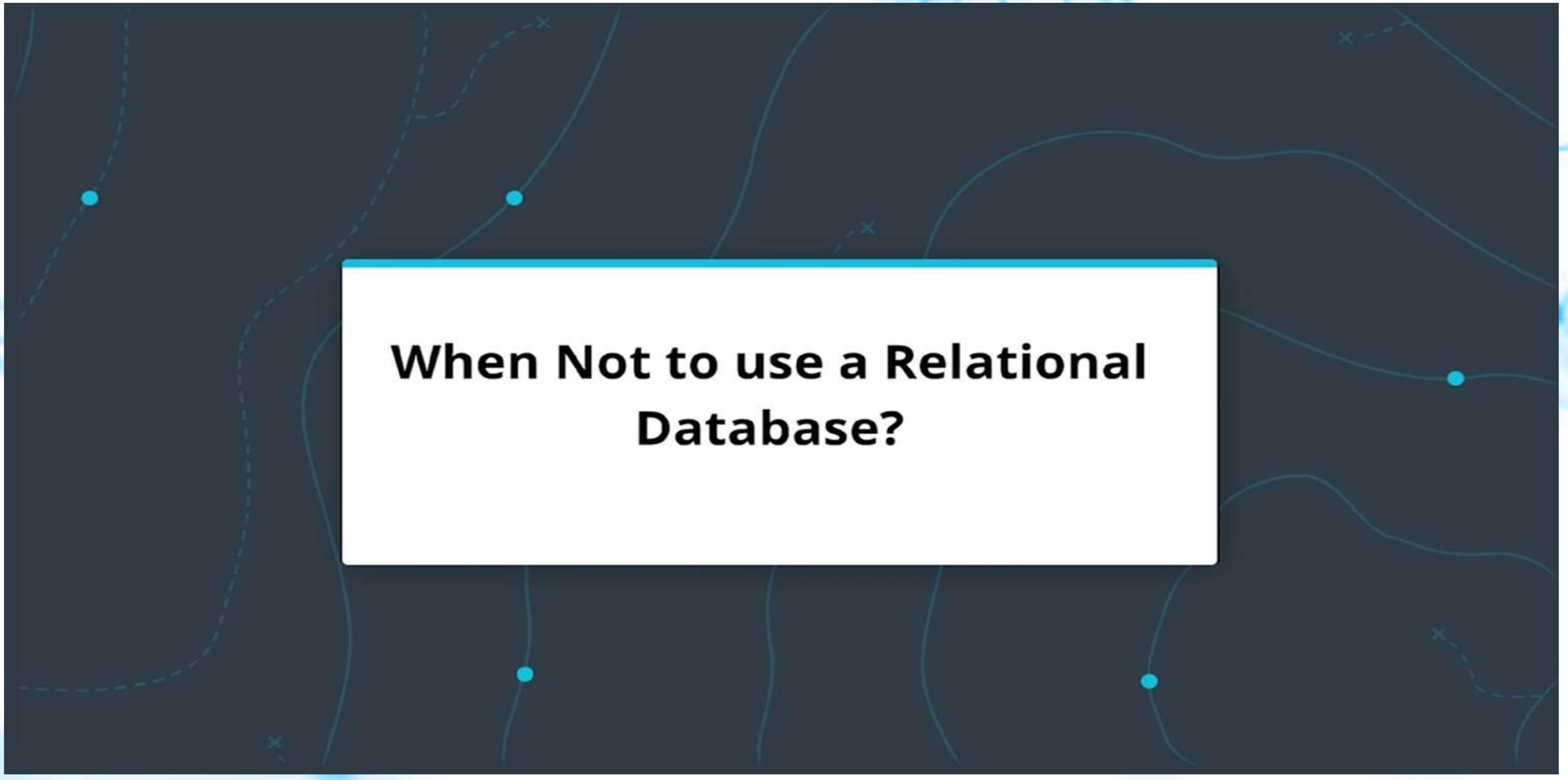
-- Wikipedia



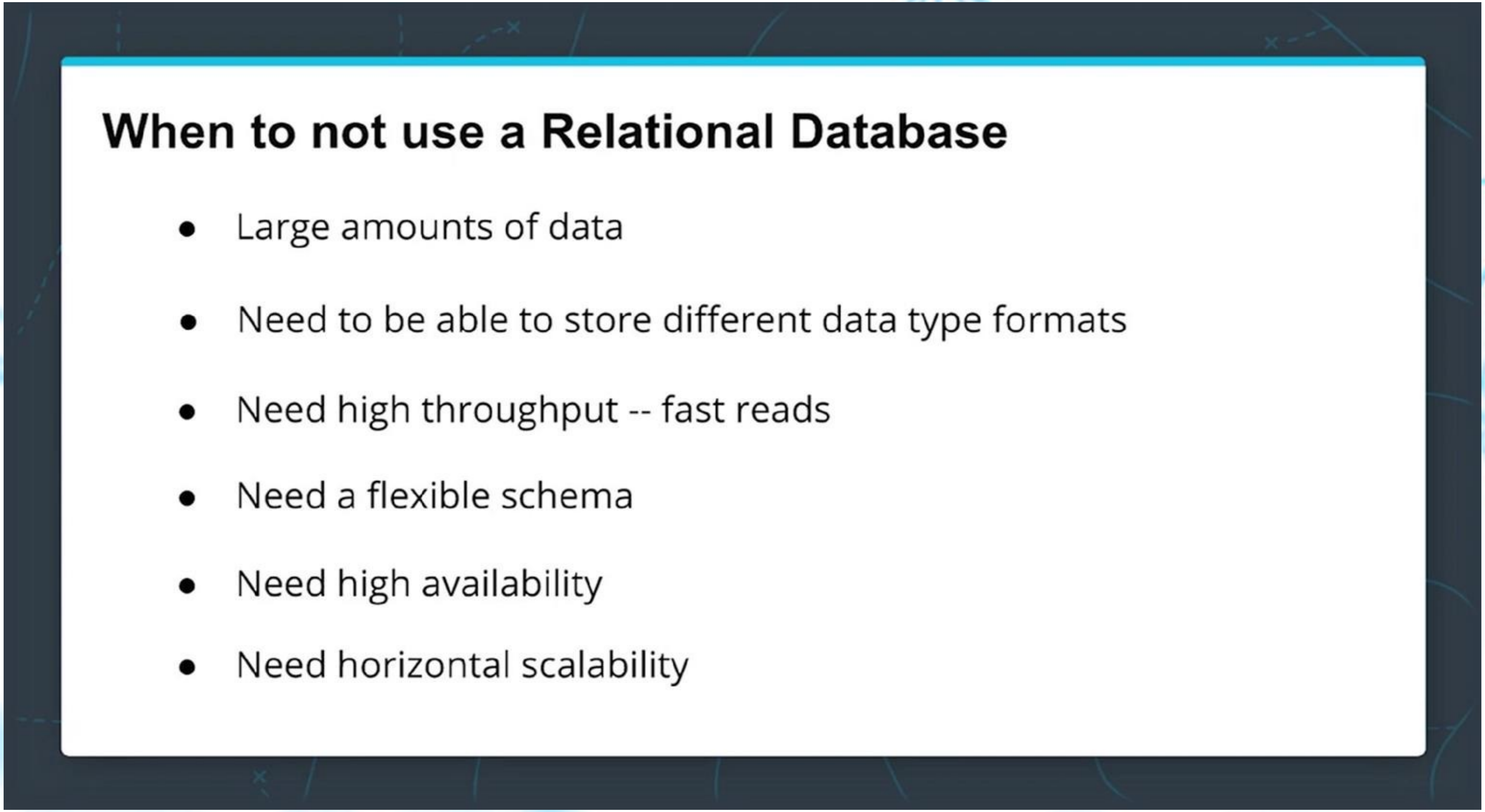
Durability

"...completed transactions are saved to database even of cases of system failure"

-- Wikipedia



When Not to use a Relational Database?



When to not use a Relational Database

- Large amounts of data
- Need to be able to store different data type formats
- Need high throughput -- fast reads
- Need a flexible schema
- Need high availability
- Need horizontal scalability