

ER - DIAGRAM

LECTURE 2: Databases

ER - Diagram

Entity Relationship model

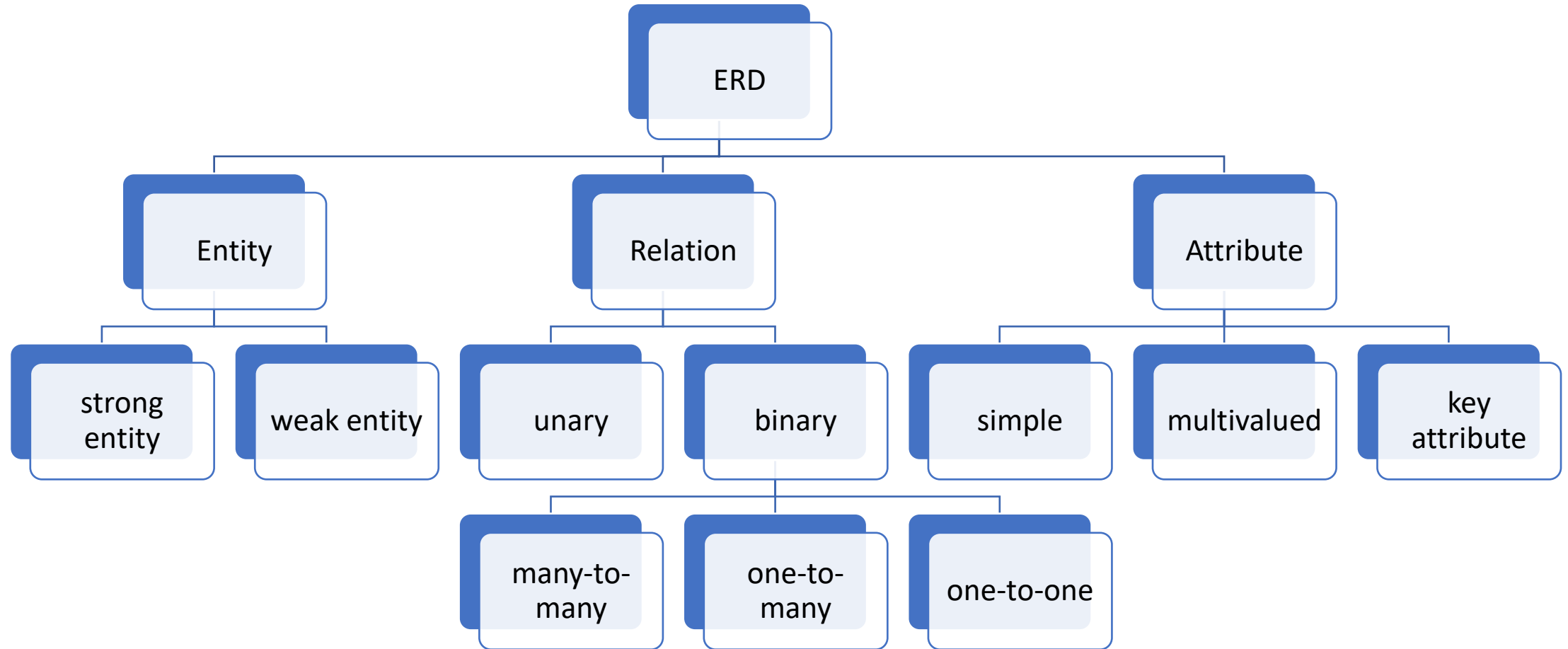
ER - Diagram

- Visual representation of the ER conceptual data model.
- High-level design.
- Not linked to the implementation or hardware.
- ERDs proposed by Peter Chen.

ER - Diagram

- User story/requirement analysis ➔ **ER** ➔ relational database schema.
- Easy to translate into relational tables.
- Describes the logical structure of a (relational) database.
- Suitable for structured systems (fixed, well-defined schema).

ERD - components



ER - Diagram



person, place, activity, event, concept, real world object etc.
usually a noun



ER - Diagram



ENTITY

person, place, activity, event, concept, real world object etc.
usually a noun



RELATION

links entities (unary, binary, ternary).
usually a verb



ATTRIBUTE

ER - Diagram



ENTITY

person, place, activity, event, concept, real world object etc.
usually a noun



RELATION

links entities (unary, binary, ternary).
usually a verb



ATTRIBUTE

describes entities or relations

Entities

- Unique names, uppercase characters
- Graphical representation: rectangles
- Relational database: entity ➔ table (line & columns)
- Primary key: attribute or group of attributes that uniquely identifies an entity instance

Entities

DEPARTMENTS

EMPLOYEES

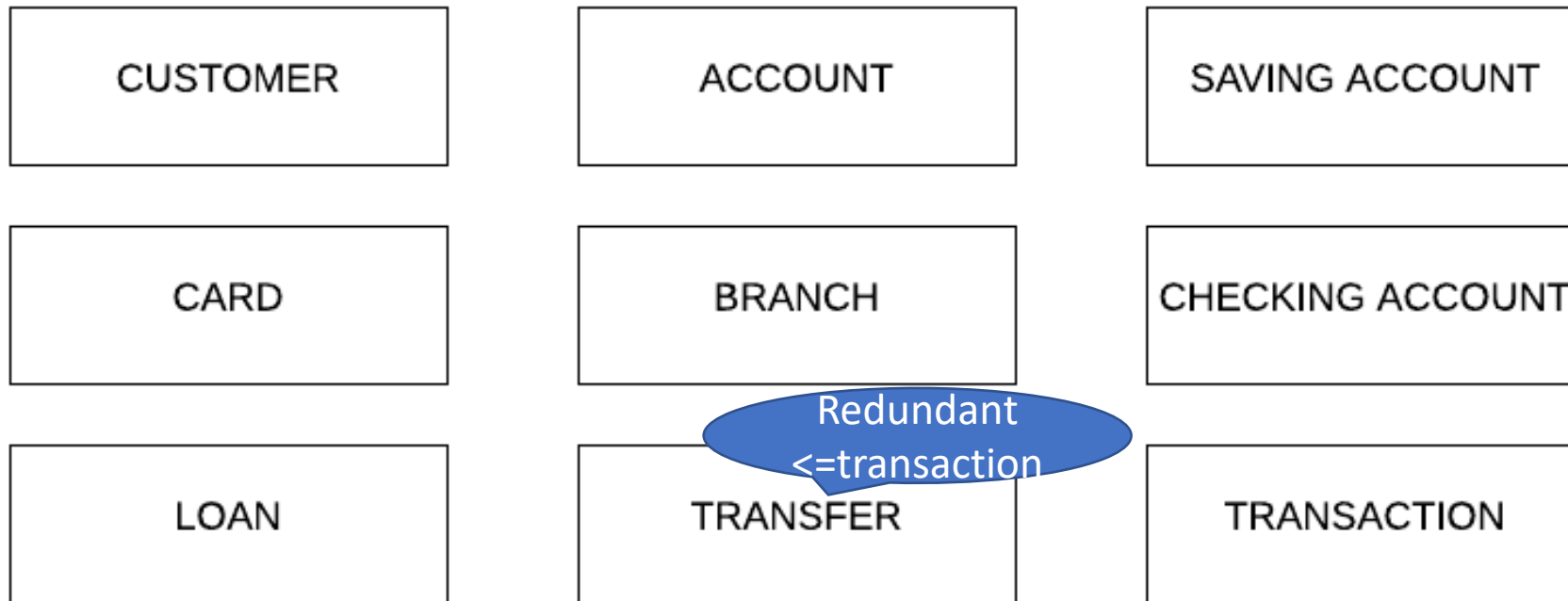
PROJECT

TASK

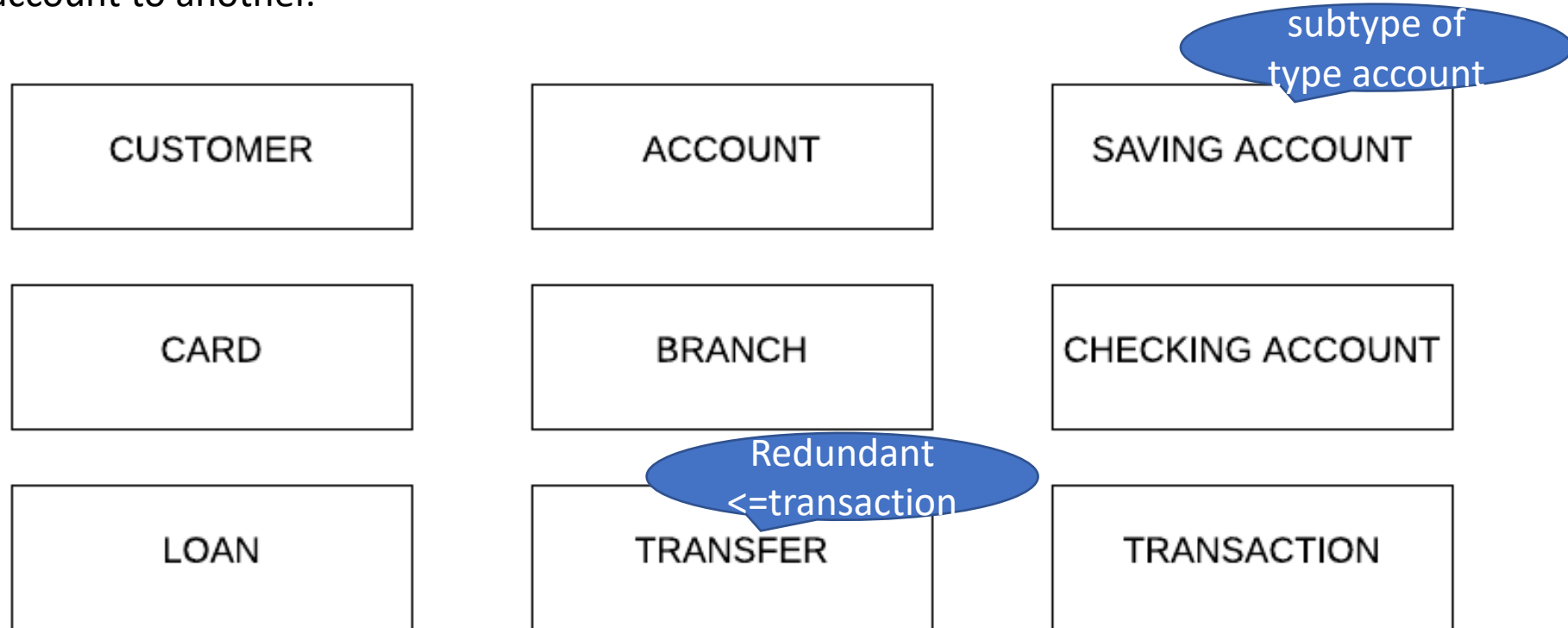
Banking -- Entities

- A customer opens a saving account or a checking account, at a bank branch. He may also access loans. For each checking account he has a card. Periodically he may withdraw money from his account or partially pay his loans. He may also transfer money from one account to another.

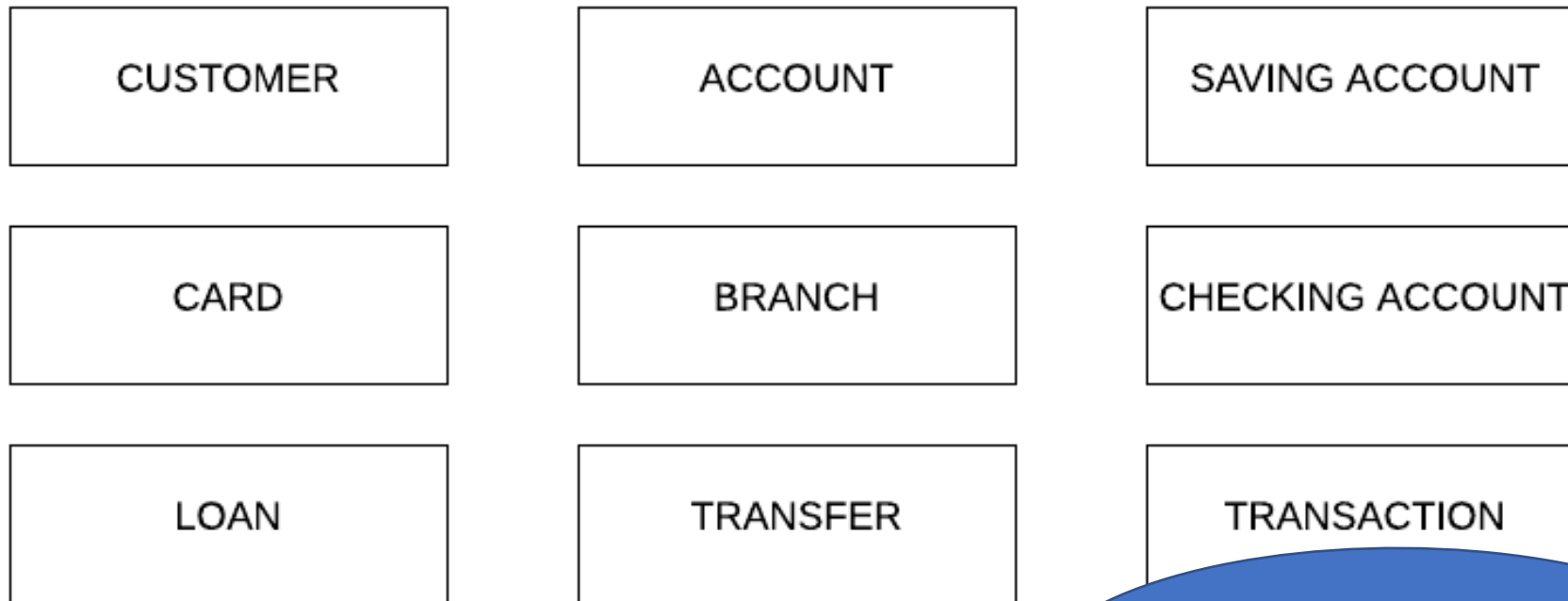
A customer opens a saving account or a checking account, at a bank branch. He may also access loans. For each checking account he has a card. Periodically he may withdraw money from his account or partially pay his loans. He may also transfer money from one account to another.



A customer opens a saving account or a checking account, at a bank branch. He may also access loans. For each checking account he has a card. Periodically he may withdraw money from his account or partially pay his loans. He may also transfer money from one account to another.



A customer opens a saving account or a checking account, at a bank branch. He may also access loans. For each checking account he has a card. Periodically he may withdraw money from his account or partially pay his loans. He may also transfer money from one account to another.

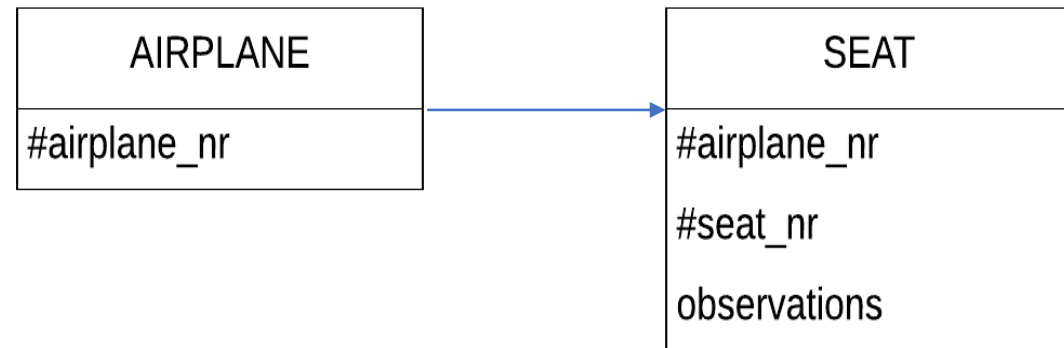
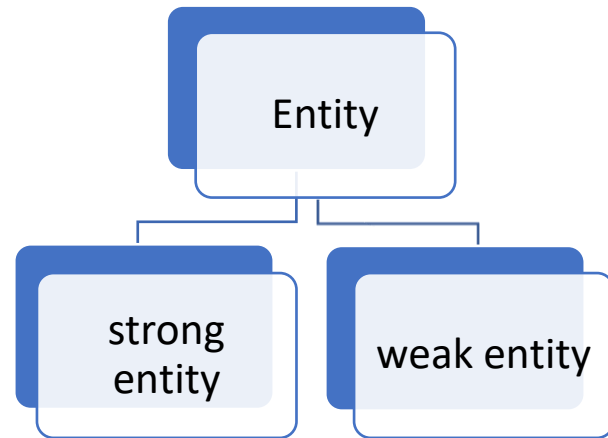


!!!not all beneficiaries are customers of the same bank (missing from story)

Airline -- Entities

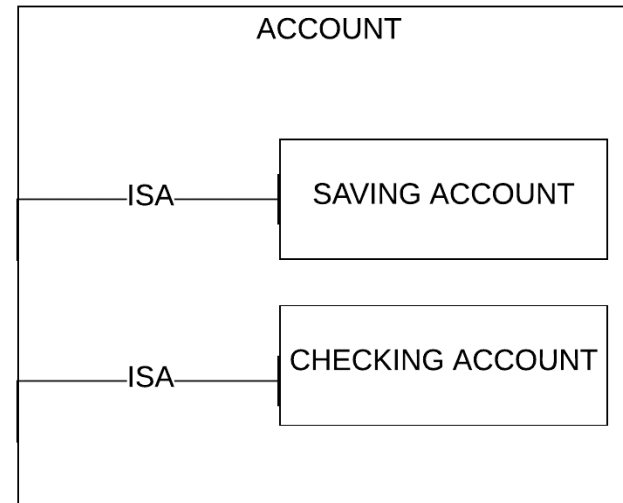
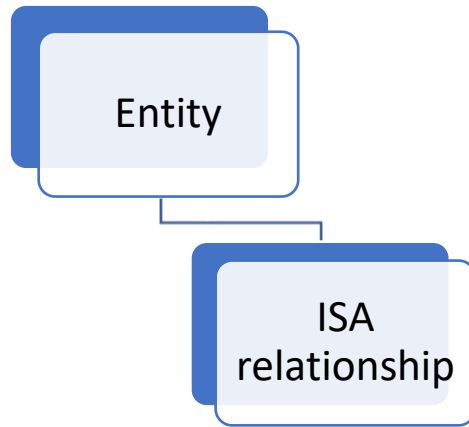
- The airline has one or more airplanes. An airplane has a model number, and capacity. Each flight is carried out by airplanes. An airplane is uniquely identified by its Registration_no and a flight is identified by its Flight_no. A passenger can book a ticket for a flight.

Entities



- Weak entity is an entity that depends on another entity.
- The primary key of a weak entity contains the primary key of the strong entity that it depends on + description/partial key.

Entities



- A sub-entity has the same key as the *super*-entity and all its attributes and relationships.

Primary key

- ***Unique*** identifier
 - ***Not null*** Must be known at any moment
 - ***Immutable***
 - Simple
 - No ambiguities
-
- Composed keys may be replaced with an *artificial key*.
 - In many RDBMS we may use autoincremented values.

Populating a Primary Key

- **Identity:** automatically assigns a unique sequence number to each row inserted.
- **Sequence or Table:** values are generated with special objects or store in a table.
- **Functions:** functions that generate uuid.
- Examples:
 - Postgres SERIAL (populate with sequence values)
 - MySql AUTO_INCREMENT.
 - Oracle Sequence
 - My Sql Server IDENTITY
 - Mongo ObjectID

Primary key UUID/GUID

- **universally unique identifier** 128-bit
- Not the best solution for clusters (sequential UUIDs might be used).
- Types:
 - Type 1 : 4 bytes + 2 bytes + 2 bytes + 2 bytes + 6 bytes = time + node
 - Type 4 : 122 bits randomly generated, 6 bits reserved for version and variant.
- Bit for type
 - type 1 2ad1db02-2ff0-11eb-**a**dc1-0242ac120002
 - type 4 a7bc2d72-7153-44a1-**8**3df-d03dd298cf53

Primary key UUID/GUID in MongoDB

- A 4-byte timestamp, representing the ObjectId's creation, measured in seconds since the Unix epoch.
- A 5-byte random value generated once per process. This random value is unique to the machine and process.
- A 3-byte incrementing counter, initialized to a random value.

<https://www.mongodb.com/docs/v6.2/reference/method/ObjectId/>



Databases C2 Entity Relationship Diagram

