ER - DIAGRAM

LECTURE 1: Databases

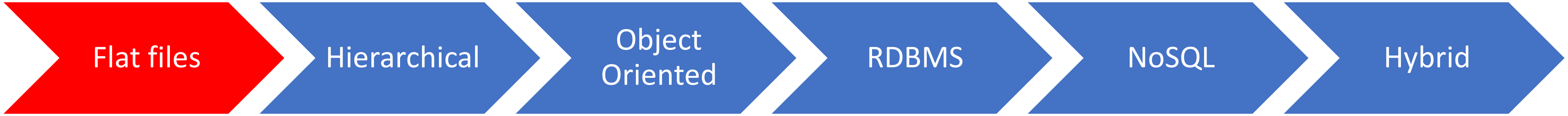
DBMS (Database Management System)

- System designed to define and manipulate data.
    - Storage.
    - Retrieval.
    - Updates.

DBMS (Database Management System)

- Avoid redundancy, inconsistency.
- Concurrent data access.
- Provides security and recovery.
- Declarative language to manipulate, query, define and control data.
- DDL, DML, DCL.
- Data dictionary: database providing info about database structure.

DBMS (Database Management System)



Flat files

Hierarchical

Object Oriented

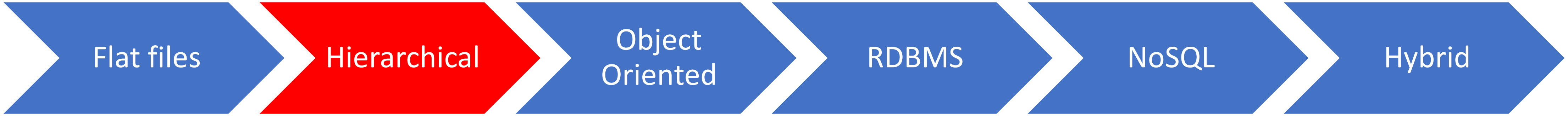
RDBMS

NoSQL

Hybrid

- Text database, example CSV format.
- Implemented in 1970 (IBM).
- File = table with a single record on each line.
- Read, store and send.
- Simple structure.
- Inefficient: slow, duplicated values, difficult to update etc.

DBMS (Database Management System)



Flat files

Hierarchical

Object Oriented

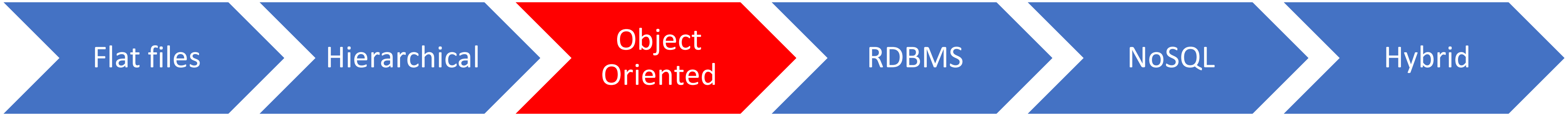
RDBMS

NoSQL

Hybrid

- Tree structure, examples: file system, Windows Registry
- IBM Information Management System (IMS)
- XML, XAML
- Used in mainframe era.
- Rigid structure.
- Only *One-to-many* relationship.
- Traversing very easy, moving a node difficult

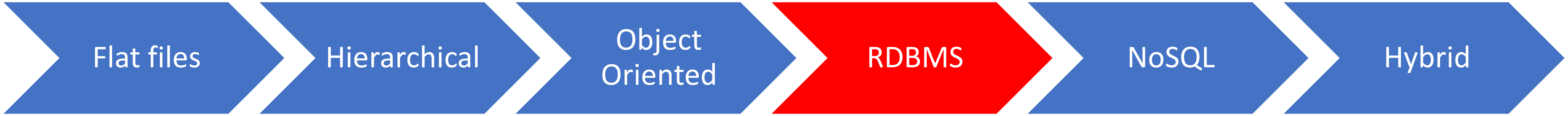
DBMS (Database Management System)



Flat files → Hierarchical → Object Oriented → RDBMS → NoSQL → Hybrid

- Hybrid relation + objects =>> tables of objects.
- Realm database for Android/IoS: classes used as schema definition, alternative for SQLite.
  - Lightweight, doesn’t need an ORM
  - MVCC architecture (multiversion concurrency control)
- MongoDB Realm sync data [1].

DBMS (Database Management System)



Flat files

Hierarchical

Object Oriented

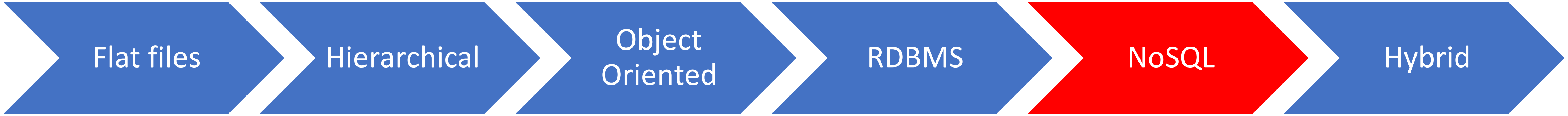
RDBMS

NoSQL

Hybrid

- Transaction oriented systems (example financial transactions).
- ACID: Atomicity, consistency, isolation, durability.
- Suitable for structured data.

DBMS (Database Management System)



Flat files

Hierarchical

Object Oriented

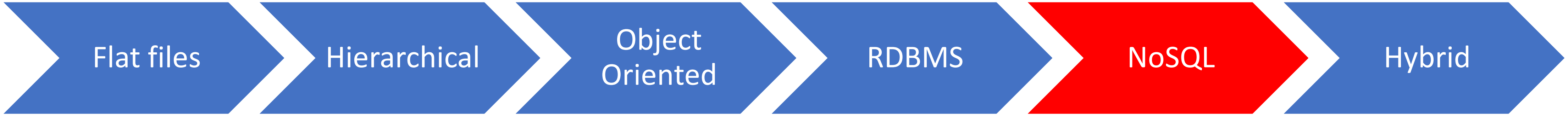
RDBMS

NoSQL

Hybrid

- RDBMS hard to scale (scale vertically, not horizontally).
- RDB Restrictive schemas =>> flexible structure.
- The state of the database can change.
- !!! availability, scalability, performance
- Sharding: distribute data on different servers

DBMS (Database Management System)



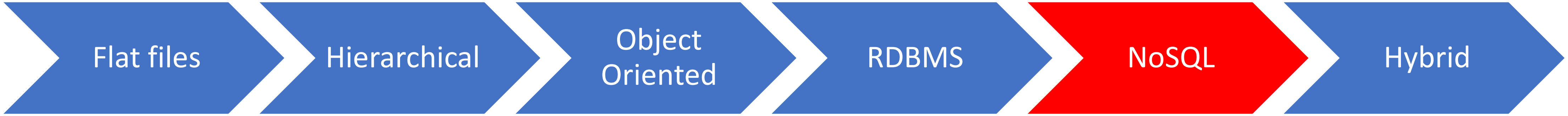
- Cloud and bigdata.
- BASE (Basically Available, Soft state, Eventually consistent)

  Database appears to work most of the time

  Replicas are not consistent all the time

  At some later point database will be consistent.

DBMS (Database Management System)



Flat files

Hierarchical

Object Oriented

RDBMS

NoSQL

Hybrid

- Cloud and bigdata.
- BASE (Basically Available, Soft state, Eventually consistent)
- Types:
  - key-value: Redis
  - Document: Mongo, Firestore
  - Column: Apache Cassandra, HBase, MariaDB
  - Graph: Neo4j

Sql or NoSQL

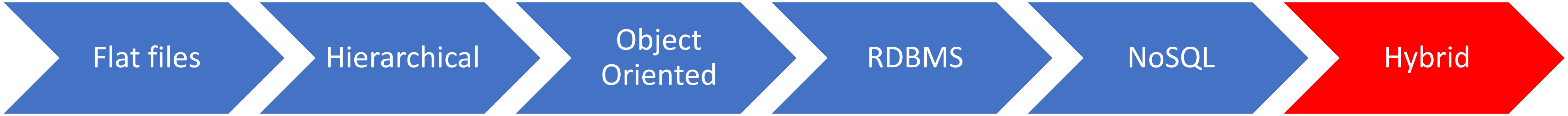
Relational

- Vertical scalability
- Consistency model: ACID
- pre-defined schema
- SQL language
- Normalized data

NoSql

- Horizontal scalability
- Consistency model: BASE
- Flexible schema
- No standard
- Collections, redundancy

DBMS (Database Management System)



Flat files → Hierarchical → Object Oriented → RDBMS → NoSQL → Hybrid

- Integration of Relational and NoSQL databases.
- Integration of in-memory DB and on-disk DB
- Altibase, Orient DB

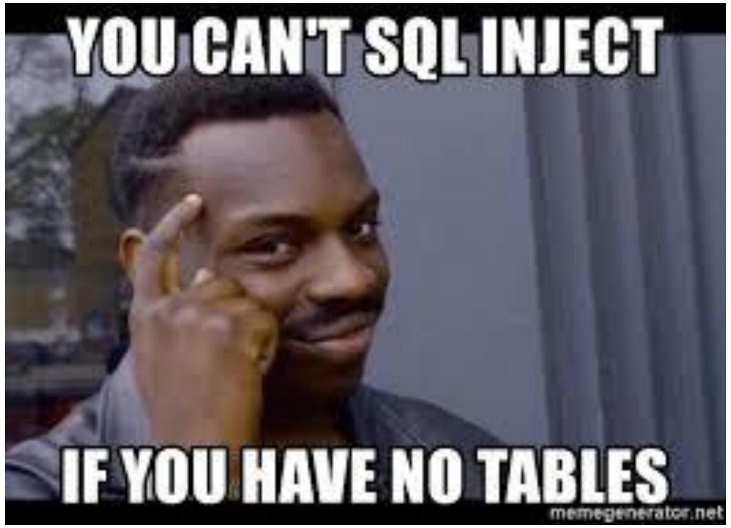
Course roadmap

- Database design (ERD, Relational model, normalization)
- Transactional systems, consistency models, concurrency control
- NoSql and BidData topics

- SQL (LMD, LDD, LCD)

- … & other topics …
- Please answer www.menti.com 8045 7859 Q1, Q2, Q3, Q4, Q5

Course roadmap



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.

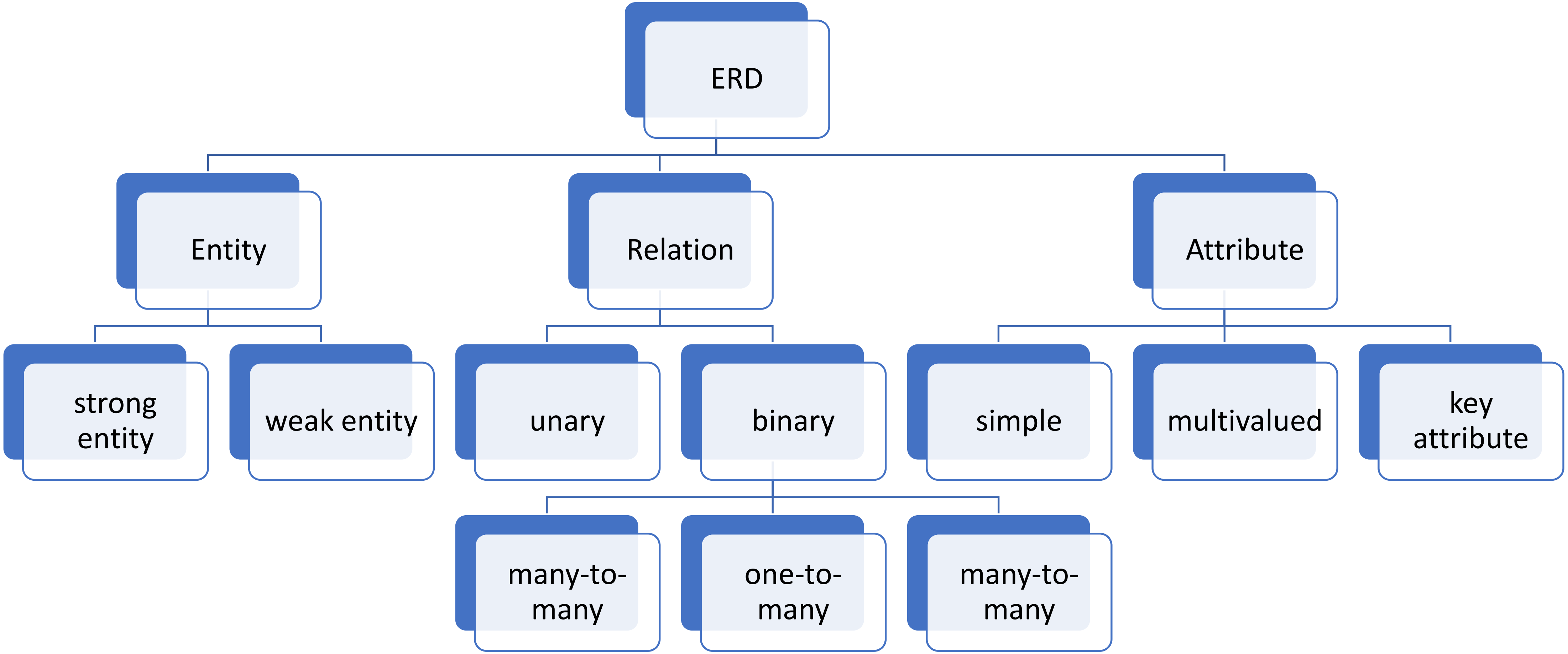
- Peter Chen proposed ERDs in 1976.

ER - Diagram

- User story/requirement analysis ➔ **ER** ➔relational database schema.
- Easy to translate into relational tables.

- Describes the logical structure of the (relational) database.
- Suitable for structured systems.

ERD - components



ER - Diagram

ENTITY

person, place, activity, event, concept, real world object etc.

usually a noun

RELATION

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

ER - Diagram

ENTITY

person, place, activity, event, concept, real world object etc.

usually a noun

RELATION

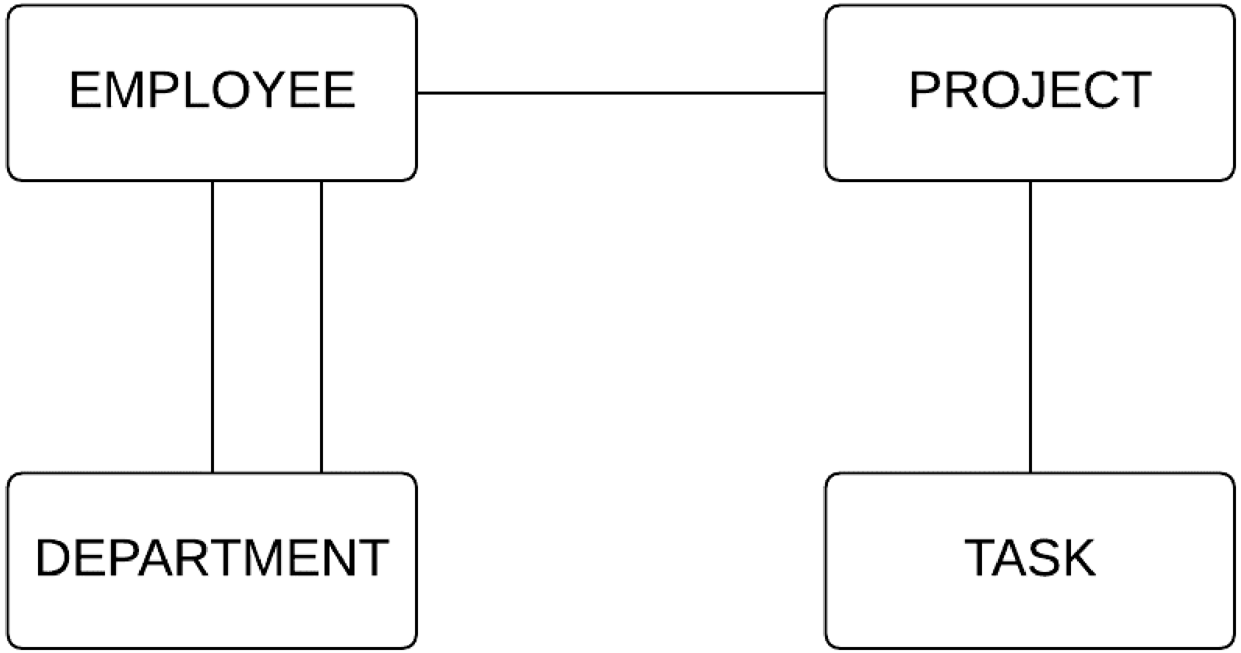
links entities (unary, binary, ternary).

usually a verb

ATTRIBUTE

describe entities or relations

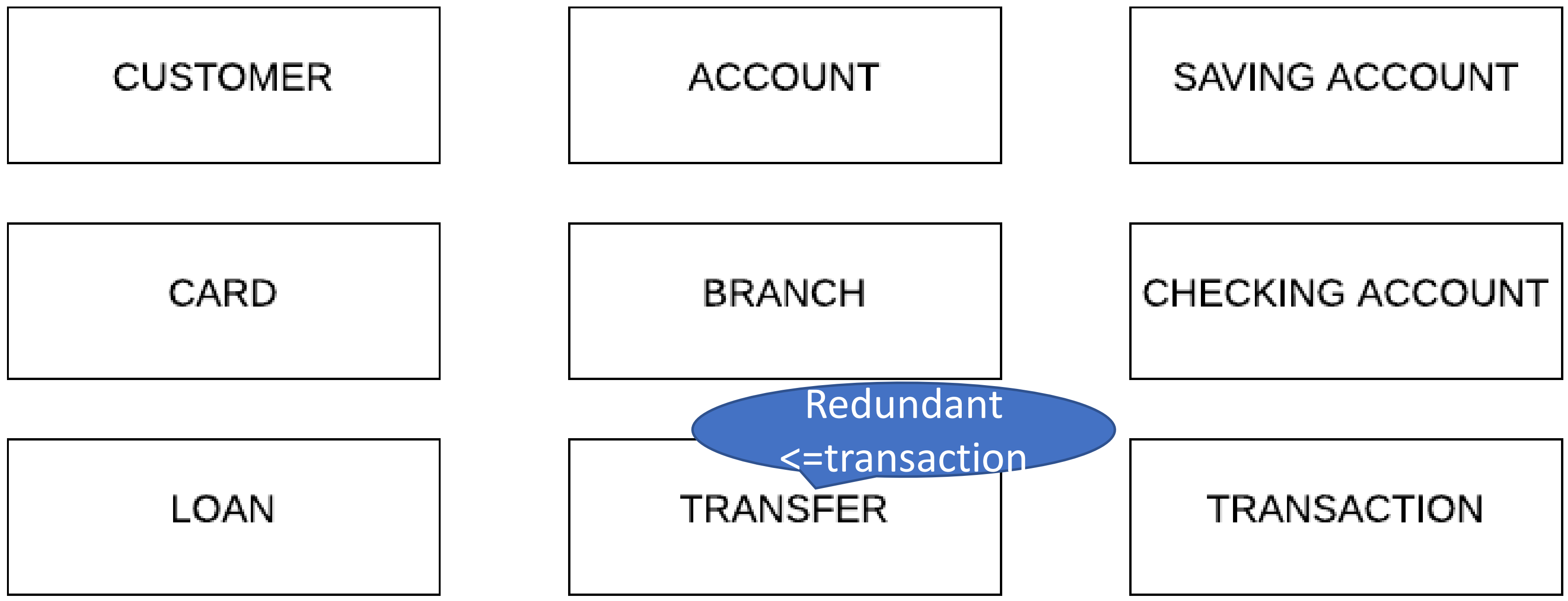
Entities



Banking (1)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.

- Please answer www.menti.com 8045 7859 Q6, Q7



CUSTOMER

ACCOUNT

SAVING ACCOUNT

CARD

BRANCH

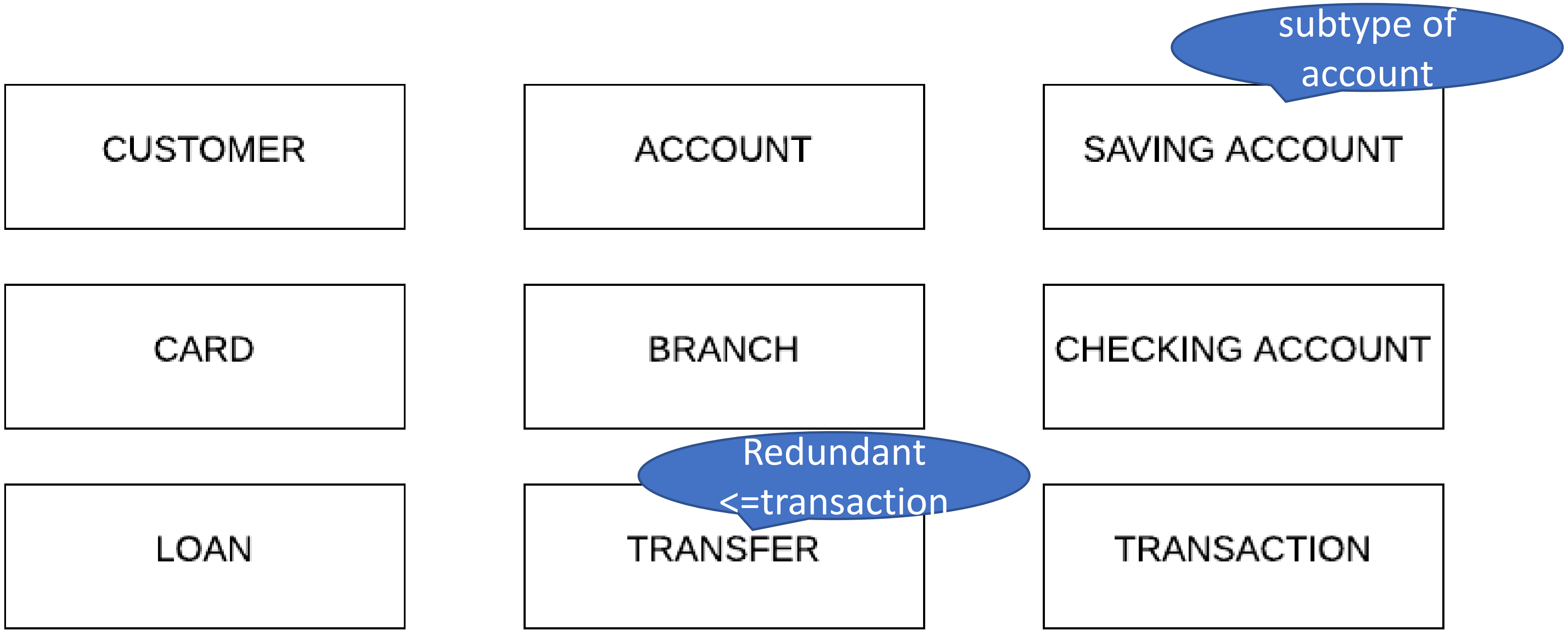
CHECKING ACCOUNT

LOAN

TRANSFER

TRANSACTION

Redundant <=transaction



CUSTOMER

ACCOUNT

SAVING ACCOUNT

subtype of account

CARD

BRANCH

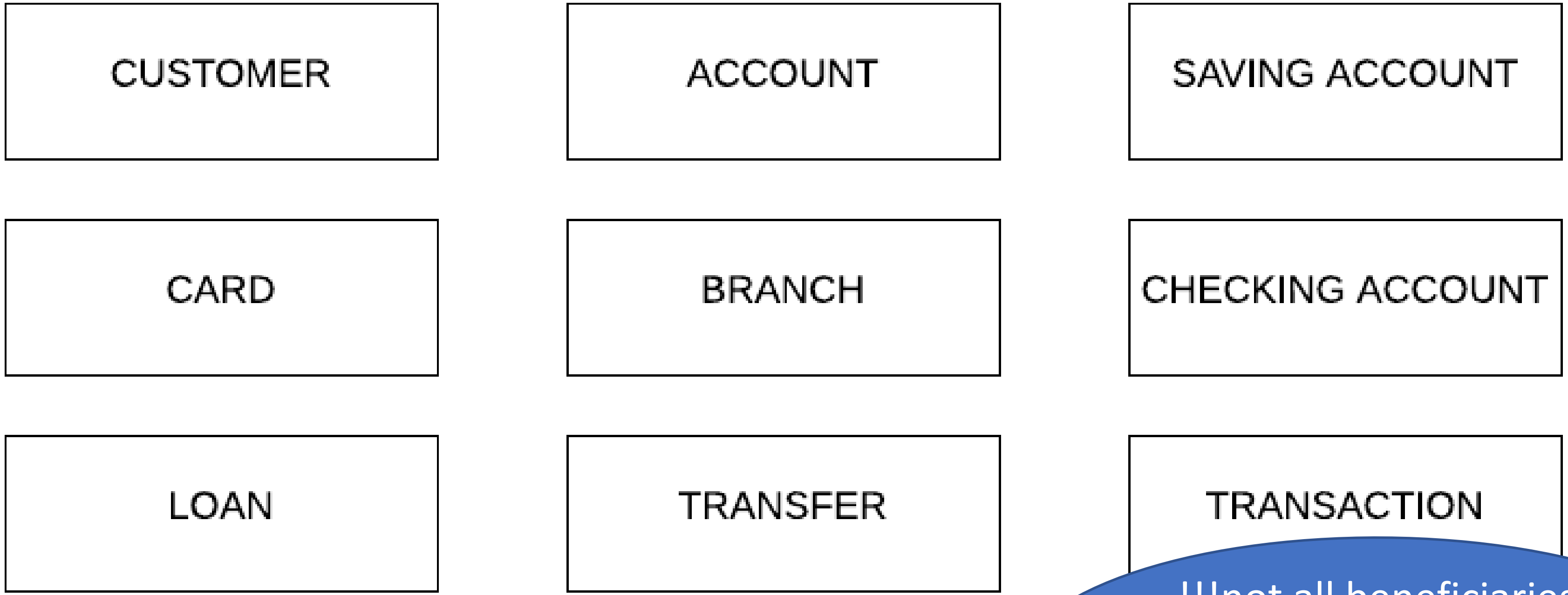
CHECKING ACCOUNT

LOAN

TRANSFER

Redundant <=transaction

TRANSACTION



CUSTOMER

ACCOUNT

SAVING ACCOUNT

CARD

BRANCH

CHECKING ACCOUNT

LOAN

TRANSFER

TRANSACTION

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

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

Primary key

- ***Unique*** identifier
- Must be known at any moment (***not null***)
- Simple
- No ambiguities
- Immutable

- Composed keys may be replaced with an *artificial key*.
- In many RDBMS we may use autoincremented values.

Primary key UUID/GUID

- universally unique identifier 128-bit
- Probability of collision (that a UUID is duplicated) is negligible.
- No need to change when merging to databases.
- Known before the insertion of a new row, without querying the database.

- Please answer www.menti.com 8045 7859 Q8

Primary key UUID/GUID

- universally unique identifier 128-bit
- Not the best solution for clusters (use sequential UUIDs).

- 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-adc1-0242ac120002
    type 4 a7bc2d72-7153-44a1-83df-d03dd298cf53

Candidate key

- *Relation* in relational model <> *relationship* in ERD
- Relation <-> set of tuples <-> tables <-> attributes of relation are columns

- Minimal *superkey* or o *relation* (minimal set of attributes) such that:
  1) There are no two distinct tuples sharing the same values for the *superkey* (unique)
  2) No proper subsets of the superkey has property (1)

- Attributes of the superkey are called prime attributes.
- Attributes that does not occur in any superkey are non-prime attributes.
- If there are no null values, since a relation is a *set* of tuples, each relation has at least one candidate key, i.e the set of all its attributes.

Candidate keys

BUYER_ID	PRICE	HALL_NO	DATE	TYPE
1	150	Coliseum	08/03/22	VIP
1	150	Lyttelton	14/04/22	A
2	140	Olivier	01/05/22	A
2	90	Coliseum	04/06/22	B
2	220	Lyttelton	08/03/22	VIP
3	140	Olivier	14/04/22	B
3	220	Olivier	20/03/22	VIP

Candidate keys

BUYER_ID	PRICE	HALL_NO	DATE	TYPE
1	150	Coliseum	08/03/22	VIP
1	150	Lyttelton	14/04/22	A
2	140	Olivier	01/05/22	A
2	90	Coliseum	04/06/22	B
2	220	Lyttelton	08/03/22	VIP
3	140	Olivier	14/04/22	B
3	220	Olivier	20/03/22	VIP

There are no candidate key with one attribute

Candidate keys

BUYER_ID	PRICE	HALL_NO	DATE	TYPE
1	150	Coliseum	08/03/22	VIP
1	150	Lyttelton	14/04/22	A
2	140	Olivier	01/05/22	A
2	90	Coliseum	04/06/22	B
2	220	Lyttelton	08/03/22	VIP
3	140	Olivier	14/04/22	B
3	220	Olivier	20/03/22	VIP

Unique keys: ~~(BUYER_ID, PRICE)~~

Candidate keys

BUYER_ID	PRICE	HALL_NO	DATE	TYPE
1	150	Coliseum	08/03/22	VIP
1	150	Lyttelton	14/04/22	A
2	140	Olivier	01/05/22	A
2	90	Coliseum	04/06/22	B
2	220	Lyttelton	08/03/22	VIP
3	140	Olivier	14/04/22	B
3	220	Olivier	20/03/22	VIP

Unique keys: ~~(BUYER_ID, HALL_NO)~~

Candidate keys

BUYER_ID	PRICE	HALL_NO	DATE	TYPE
1	150	Coliseum	08/03/22	VIP
1	150	Lyttelton	14/04/22	A
2	140	Olivier	01/05/22	A
2	90	Coliseum	04/06/22	B
2	220	Lyttelton	08/03/22	VIP
3	140	Olivier	14/04/22	B
3	220	Olivier	20/03/22	VIP

Unique keys: (BUYER_ID, DATE) (BUYER_ID, VIP) (PRICE, DATE) (HALL_NO, DATE) (HALL_NO, TYPE)
(BUYER_ID, PRICE, HALL_NO) (BUYER_ID, HALL_NO, DATE)
etc -- all sets with 3, 4 or 5 attributes

Candidate keys

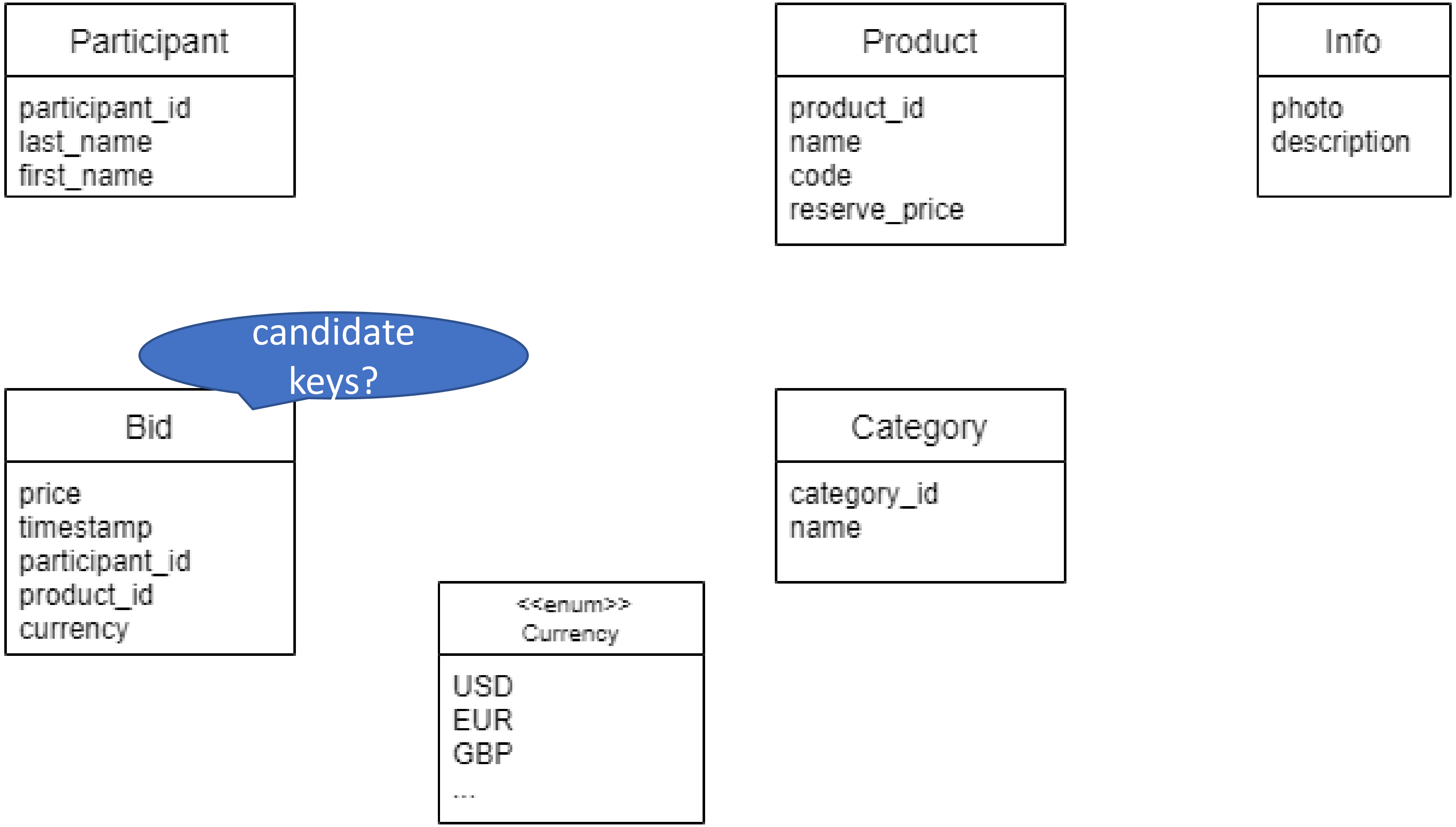
BUYER_ID	PRICE	HALL_NO	DATE	TYPE
1	150	Coliseum	08/03/22	VIP
1	150	Lyttelton	14/04/22	A
2	140	Olivier	01/05/22	A
2	90	Coliseum	04/06/22	B
2	220	Lyttelton	08/03/22	VIP
3	140	Olivier	14/04/22	B
3	220	Olivier	20/03/22	VIP

Candidate keys: (BUYER_ID, DATE) (BUYER_ID, VIP) (PRICE, DATE) (HALL_NO, DATE) (HALL_NO, TYPE) (BUYER_ID, PRICE, HALL_NO) (BUYER_ID, PRICE, TYPE) , ~~(PRICE_DATE, DATE, TYPE)~~

Candidate keys

BUYER_ID	PRICE	HALL_NO	DATE	TYPE
1	150	Coliseum	08/03/22	VIP
1	150	Lyttelton	14/04/22	A
2	140	Olivier	01/05/22	A
2	90	Coliseum	04/06/22	B
2	220	Lyttelton	08/03/22	VIP
3	140	Olivier	14/04/22	B
3	220	Olivier	20/03/22	VIP

Candidate keys: (BUYER_ID, DATE) (BUYER_ID, VIP) (PRICE, DATE) (HALL_NO, DATE) (HALL_NO, TYPE)
(BUYER_ID, PRICE, HALL_NO) (BUYER_ID, PRICE, TYPE) ,



Participant

participant_id
last_name
first_name

Product

product_id
name
code
reserve_price

Info

photo
description

candidate keys?

Bid

price
timestamp
participant_id
product_id
currency

<<enum>>
Currency

USD
EUR
GBP
...

Category

category_id
name

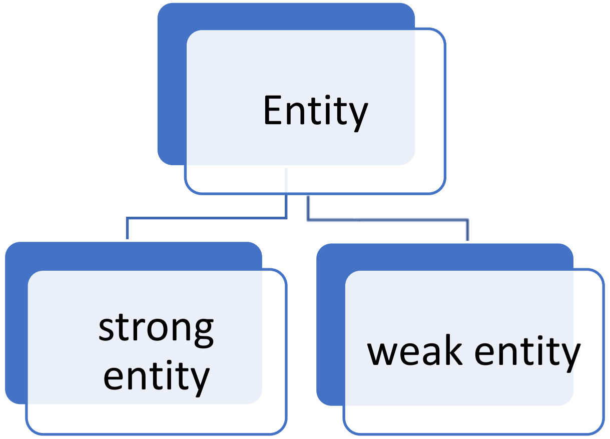
Please answer www.menti.com 44 32 42 0 Q8

Airline (1)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.

Please answer www.menti.com 8045 7859 Q10, Q11

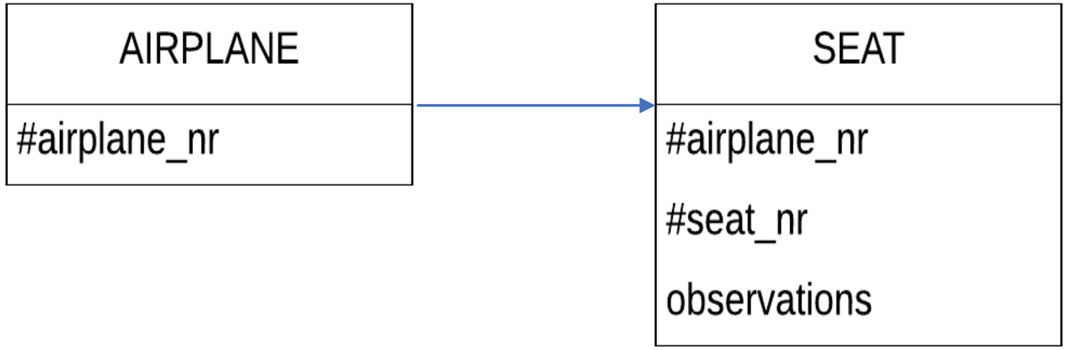
Entities



Entity

strong entity

weak entity



AIRPLANE

#airplane_nr

SEAT

#airplane_nr

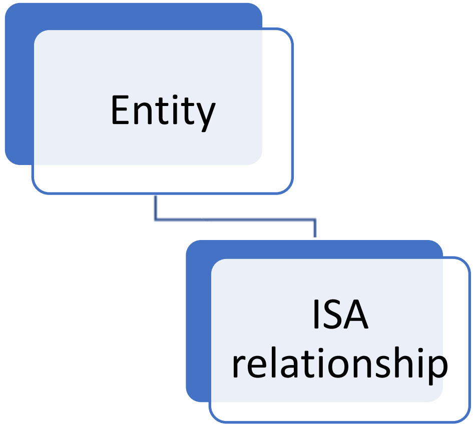
#seat_nr

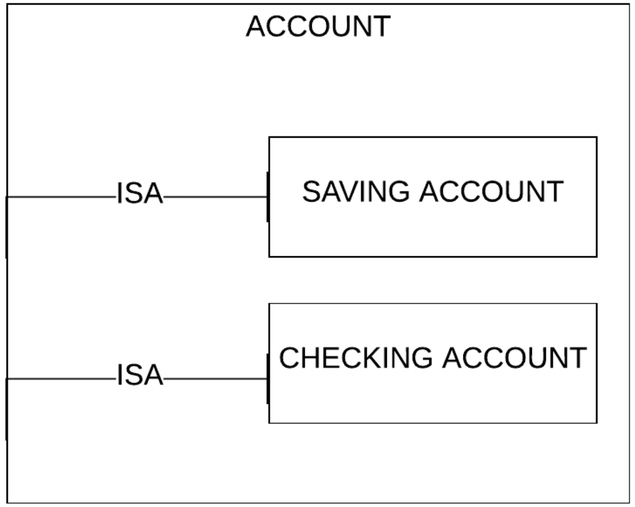
observations

- 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.

Please answer www.menti.com 8045 7859 Q12, 13

Entities





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

Please answer www.menti.com 8045 7859 Q14

Airline (1)Relationships

- 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.

Airline (1)Relationships

- 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. A flight may have one or more stops.

Airline (1)Relationships

- 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. A flight may have one or more stops. The passenger will pay for extra baggage.

- [1] https://docs.mongodb.com/realm/node/sync-data/

- [2] https://tomharrisonjr.com/uuid-or-guid-as-primary-keys-be-careful-7b2aa3dcb439

- [3] https://database.guide/newid-vs-newsequentialid-in-sql-server-whats-the-difference/

- [4] https://docs.mongodb.com/realm/sdk/android/advanced-guides/threading/#std-label-android-mvcc