Databases and Information Systems

Designing an ER Diagram – Streaming Platform

Prof. Ing. Lelio Campanile, PhD

Data Analytics Bachelor
Universita degli Studi della Campania Luigi Vanvitelli

Guided classroom exercise

Learning goals

- Design an ER schema for a **realistic, complex domain**
- Model a **ternary relationship** and discuss when it cannot be decomposed
- Introduce **composite attributes** in BCN notation
- Apply **generalisation** to specialise an entity into subtypes
- Combine **relationship attributes**, **N:M**, **1:N** and **1:1** cardinalities

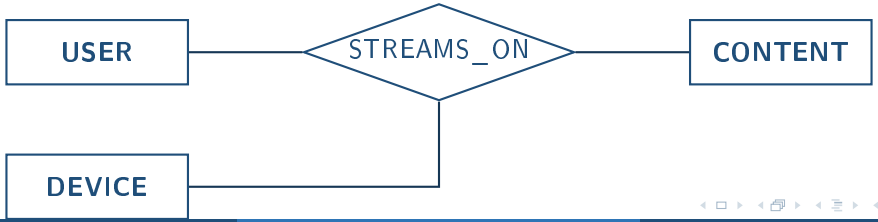
New constructs

Ternary relationship

Definition

A **ternary relationship** associates **three** entities simultaneously. An occurrence links one instance of each of the three entities at the same time.

- Used when the information depends on the **combination** of all three entities, not on any pair in isolation
- Example: a user streams a content on a specific device — the quality level and progress depend on that exact triple
- In BCN notation: same diamond symbol, three arcs

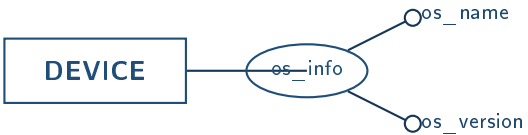


Composite attribute

Definition

A **composite attribute** is an attribute made up of two or more sub-attributes that together describe a single concept.

- In BCN notation: an **ellipse** connected to the entity, with **circles** for each sub-attribute
- Example: *os_info* of a device is described by *os_name* and *os_version* together

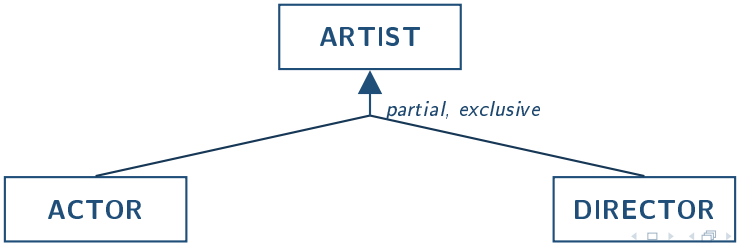


Generalisation

Definition

A **generalisation** represents a parent entity E that is more general than its child entities $E_1, \ldots, E_n$. Children **inherit** all properties of the parent.

- **Total**: every parent occurrence is also a child occurrence
- **Partial**: some parent occurrences belong to no child
- **Exclusive**: each parent occurrence belongs to at most one child
- In BCN notation: **arrow with filled triangle** from child to parent



Guided example

Case study description

A company wants to build a database for its **streaming platform**.

The platform has **users**, each identified by their **email** and characterised by a username and registration date. Users can activate **subscriptions**: each subscription has a unique ID, a start date, an end date, and a price. Each subscription activates exactly one **plan**, identified by its name and described by a monthly price, the maximum number of simultaneous screens, and the streaming quality offered.

The platform offers **contents** (films and series), each identified by a content ID and described by a title, duration, and release year. Each content belongs to exactly one **genre**, identified by its name. Contents are performed by **artists**: for each artist–content pair the role is recorded (actor, director, producer). Artists are identified by an artist ID and have a stage name and nationality. Artists are specialised into **actors** (with an associated agency) and **directors** (with a list of awards); this is a partial and exclusive generalisation.

Users stream contents on **devices**. Each device is identified by a device ID, has a type, and its operating system is described by a name and a version (composite attribute). For each streaming session — identified by the combination of user, content, and device — the platform records the date, the percentage of content watched, and the streaming quality used.

Users can write **reviews** about contents. Each review is identified by a review ID and stores a rating, a comment, and a date. Each review is written by exactly one user and refers to exactly one content.

Problem description

- **Users**: email, username, registration_date
- **Subscriptions**: subscription_id, start_date, end_date, price; each activates one **Plan** (plan_name, monthly_price, max_screens, quality)
- **Contents**: content_id, title, duration, release_year; belong to one **Genre** (genre_name)
- **Artists**: artist_id, stage_name, nationality; specialised into **Actor** (agency) and **Director** (awards); for each artist–content pair the **role** is stored
- **Devices**: device_id, type, os_info (os_name + os_version)
- Each streaming session (user + content + device) stores: stream_date, progress_pct, quality_level
- **Reviews**: review_id, rating, comment, review_date

Step 1 - Identify the entities

Question

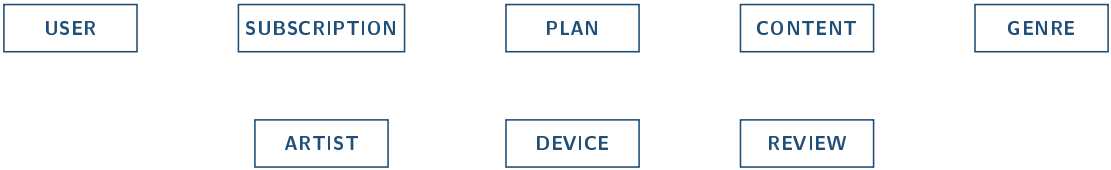
Which **main objects** of the domain do we need to store?

Think for a moment before looking at the solution.

Step 1 - Identify the entities

Answer — eight entities

- **USER**
- **SUBSCRIPTION**
- **PLAN**
- **CONTENT**
- **GENRE**
- **ARTIST** (+ ACTOR, DIRECTOR)
- **DEVICE**
- **REVIEW**



Step 2 - Add the attributes

Question

Which properties should we store for each entity?

List the attributes for each entity.

Step 2 - Add the attributes

USER

email, username, registration_date

CONTENT

content_id, title, duration, release_year

DEVICE

device_id, type, *os_info*: os_name + os_version (composite)

SUBSCRIPTION

subscription_id, start_date, end_date, price

GENRE

genre_name

REVIEW

review_id, rating, comment, review_date

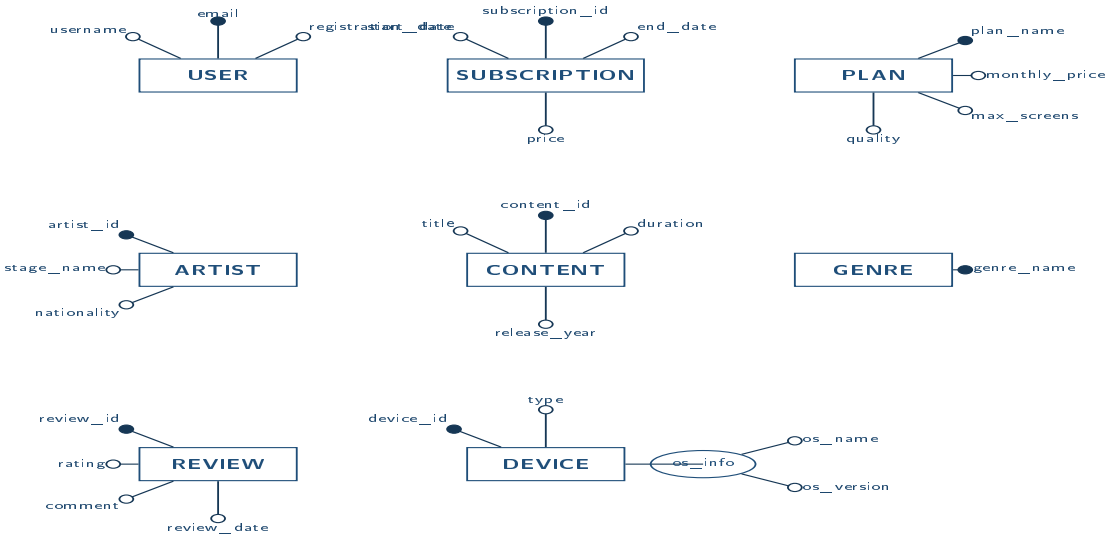
PLAN

plan_name, monthly_price, max_screens, quality

ARTIST

artist_id, stage_name, nationality

Step 2 - Attributes on the diagram



Step 3 - Identify the relationships

Question

How are the entities connected?

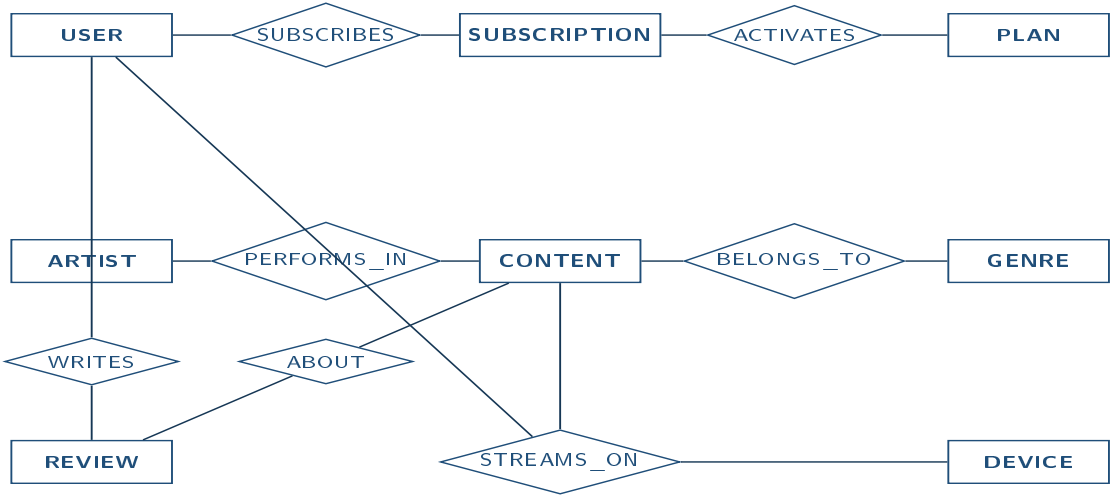
Look for verbs and business rules in the description.

Step 3 - Identify the relationships

Answer — seven relationships

SUBSCRIBES	USER → SUBSCRIPTION	a user activates subscriptions
ACTIVATES	SUBSCRIPTION → PLAN	each subscription has a plan
BELONGS_TO	CONTENT → GENRE	a content has one genre
PERFORMS_IN	ARTIST ↔ CONTENT	artists appear in contents
STREAMS_ON	USER + CONTENT + DEVICE	**ternary**: streaming session
WRITES	USER → REVIEW	a user authors a review
ABOUT	REVIEW → CONTENT	a review targets a content

Step 3 - Relationships on the diagram



Step 4 - Determine the cardinalities

Question

For each relationship, how many instances participate on each side?

- Can a user have more than one subscription over time?
- Can a subscription be linked to more than one plan?
- Can a content belong to more than one genre?
- Can an artist appear in more than one content? Can a content have more than one artist?
- Can a user stream the same content on different devices?
- Can a user write more than one review?

Step 4 - Cardinalities

SUBSCRIBES 1:N
User $(0, N)$ — Subscription $(1, 1)$

ACTIVATES N:1
Subscription $(1, 1)$ — Plan $(0, N)$

BELONGS_TO N:1
Content $(1, 1)$ — Genre $(0, N)$

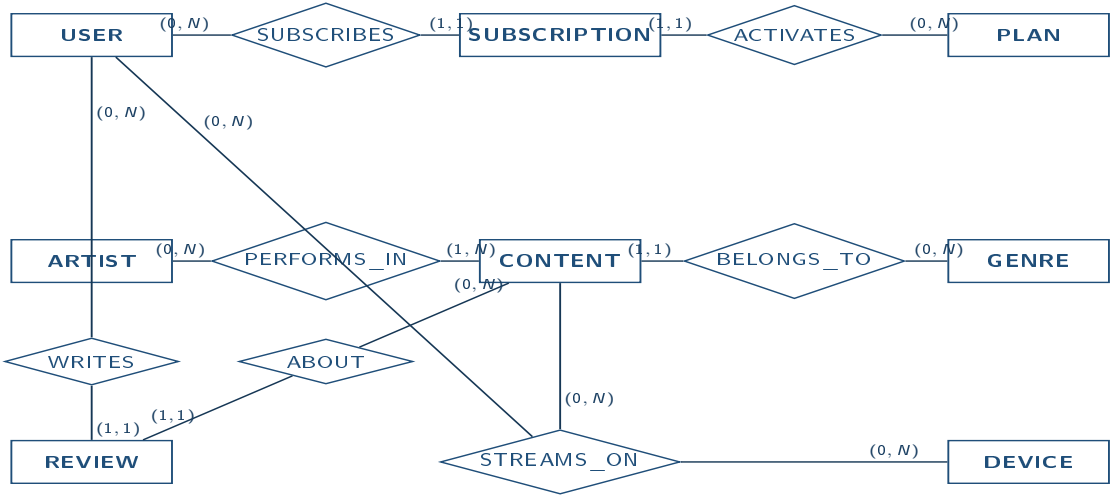
STREAMS_ON ternary
User $(0, N)$ — Content $(0, N)$ — Device $(0, N)$

PERFORMS_IN N:M
Artist $(0, N)$ — Content $(1, N)$

WRITES 1:N
User $(0, N)$ — Review $(1, 1)$

ABOUT N:1
Review $(1, 1)$ — Content $(0, N)$

Step 4 - Cardinalities on the diagram



Step 5 - Attributes of relationships

Question

Where should we store **role**, **stream_date**, **progress_pct**, **quality_level**?

- Is **role** a property of ARTIST alone? Of CONTENT alone?
- Do **stream_date** and **progress_pct** belong to USER, CONTENT or DEVICE alone?
- Or do they depend on a *specific combination* of entities?

Step 5 - Attributes of relationships

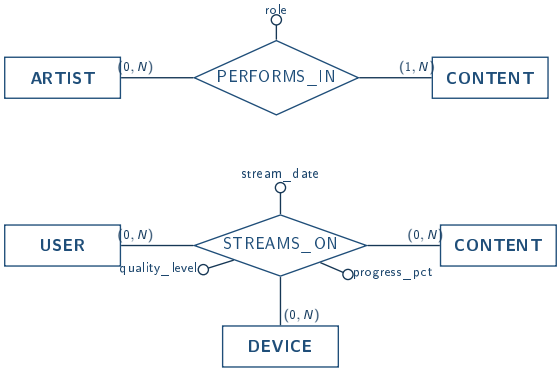
PERFORMS_IN — *role*

The role of an artist in a content depends on the specific *(Artist, Content)* pair.

The same artist can be actor in one content and director in another.

STREAMS_ON — *stream_date, progress_pct, quality_level*

These values describe one specific streaming session: they depend on the exact *(User, Content, Device)* triple.



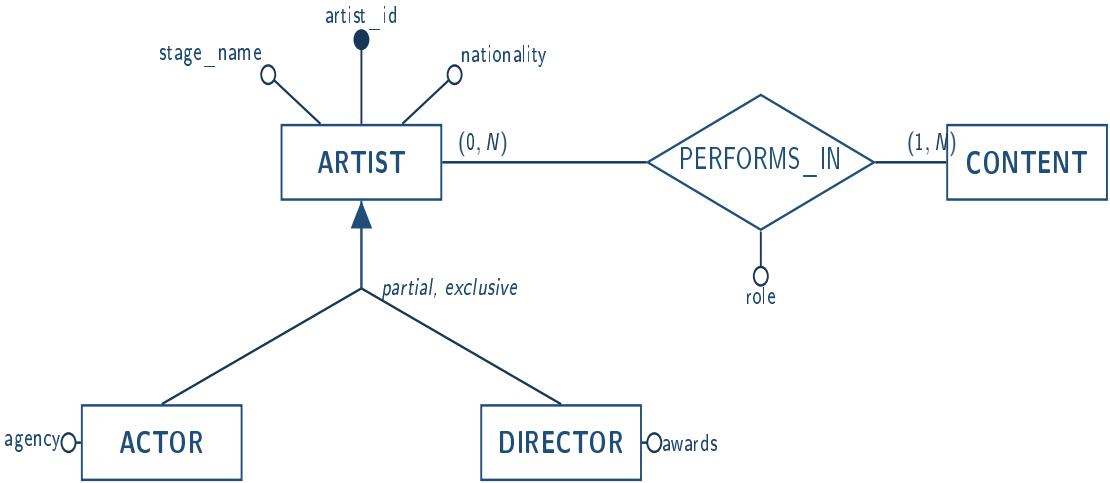
Step 6 - Generalisation: ARTIST

Question

Actors and directors are both artists, but have different properties. How do we model this?

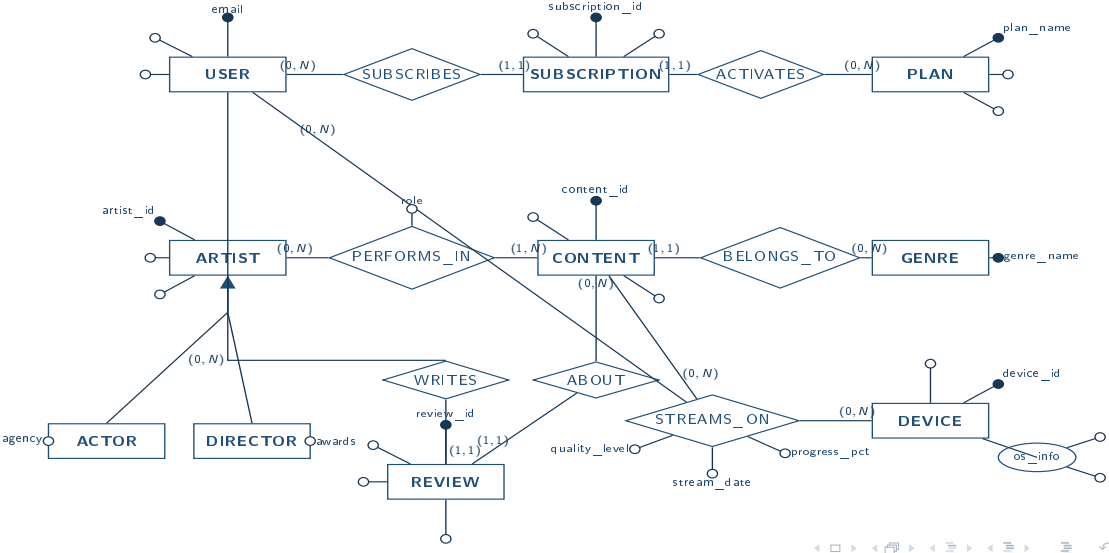
- Every **ACTOR** and every **DIRECTOR** is also an **ARTIST**
- They **inherit** artist_id, stage_name, nationality and the PERFORMS_IN relationship
- ACTOR adds: **agency**
- DIRECTOR adds: **awards**
- The generalisation is **partial** (some artists may be neither) and **exclusive** (none can be both)

Step 6 - Generalisation on the diagram



ACTOR and DIRECTOR inherit all ARTIST properties and relationships.

Final ER diagram



How to reason during design

1. Highlight the **main nouns** – possible entities
2. Search for **properties** – attributes; check for **composite** ones
3. Search for **verbs** – relationships; ask if any involve **three** entities simultaneously
4. For each relationship: *how many?* – cardinality (min, Max)
5. Check whether attributes belong to a **relationship** rather than an entity
6. Ask: are there subtypes with **different properties**? $\Rightarrow$ **generalisation**

Questions for the class

- Why is **STREAMS_ON** a ternary relationship? Could it be decomposed into two binary ones without losing information?
- Why is **role** an attribute of PERFORMS_IN and not of ARTIST or CONTENT?
- Why is the generalisation ARTIST $\rightarrow$ ACTOR / DIRECTOR **partial** and **exclusive**?
- Where would you store the **language** of a subtitle track, if a content can have multiple subtitles?
- If a user could have multiple active subscriptions simultaneously, how would the cardinality of SUBSCRIBES change?

In-class extension

How would you extend this model to support:

- **Download**: a user can download a content on a device for offline viewing (date and expiry)
- **Playlist**: users can create playlists and add contents to them
- **Season / Episode**: a series is organised into seasons, each with multiple episodes
- **Recommendation**: the platform recommends contents to users with a score and a reason

For each: identify the new entity or attribute, where it belongs, and the cardinality.

Summary

- We designed a complete ER schema for a **streaming platform**
- **8 entities**, **7 relationships** (including one ternary)
- **Composite attribute**: os_info (os_name + os_version) on DEVICE
- **Relationship attributes**: role on PERFORMS_IN; stream_date, progress_pct, quality_level on STREAMS_ON
- **Generalisation**: ARTIST $\rightarrow$ ACTOR / DIRECTOR (partial, exclusive)
- All diagrams in **BCN notation** (Batini-Ceri-Navathe)

Questions?