



SAPIENZA
UNIVERSITÀ DI ROMA

Analysis, Modelling and Implementation
of a Domain-based...

Information Integration System of Videogame Industry Sources

Longo Valerio 1655653
Papi Alessio 1761063



<https://github.com/1655653/LSDM>





TABLE OF CONTENTS



DATA SOURCES

01.

04.

IMPLEMENTATION

GLOBAL SCHEMA

02.

05.

QUERIES

MODELLING

03.

06.

CONCLUSIONS

01.

DATA SOURCES



VGChartz delivers comprehensive game chart coverage, including:
sales data, news, reviews and game database for almost all game consoles.

From VGChartz we took:

- **vgsales-2021-04-09_16_51_31 (csv)** : list of the most sold 10000 games.
- **sales-by-console (csv)** : collection of **27** files, each holding vgsales information specific to every console supported by VGChartz.

Id	Game	Console	Publisher	Developer	Sales_Total	Na_sales	Pal_sales	Japan_sales	Other_Sales	Year	Last_Update
----	------	---------	-----------	-----------	-------------	----------	-----------	-------------	-------------	------	-------------



Id	Game	Console	Publisher	Developer	Sales	Year
----	------	---------	-----------	-----------	-------	------

VGCHARTZ



GAMEDEV



Contains geographic-related information about videogame developing and publishing companies.

This dataset will be a crucial utility in the design of geographic and market related tasks.

From gamedev.net we took:

- **gamedev (csv)**

X_Coord	Y_Coord	Name	Type	City	Country	Website
---------	---------	------	------	------	---------	---------

VGCHARTZ



GAMEDEV



METACRITIC



Metacritic is a website that aggregates reviews and averaged scores for films, TV shows, music albums, video games and formerly, books. We took from this source:

From metacritic we took:

- **metacritic (json)** : A list of all 18000 games present in the catalog

This dataset stresses out the averaged user score (**us**) and the metacritic score (**ms**) for each game of the dataset.

Id	Title	User_Score	MC_Score	Platform	Year
----	-------	------------	----------	----------	------

- **datagenreX (json)** : 16 different json files (based on 16 different genres)

Id	Title	Genre
----	-------	-------

VGCHARTZ



GAMEDEV



METACRITIC



Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment.

From kaggle we took:

- **HistoricalEsportData (csv)** : historical information about 500 games and related earnings in tournaments.

Date	Game	Earnings	Players	#Tournaments
------	------	----------	---------	--------------

- **GeneralEsportData (csv)** : general information about 500 games and related earnings in tournaments.

Game	ReleaseDate	Genre	TotalEarnings	OnlineEarnings	PricedPlayers	TotalTournaments
------	-------------	-------	---------------	----------------	---------------	------------------

VGCHARTZ



GAMEDEV



METACRITIC



KAGGLE



CORGIS



CORGIS is a “collection of Real-time, Giant, Interesting, Situated Datasets”.

From corgis we took:

- **sales_playtime_video_games (csv)** : list of 1200 games

Title	Multiplayer?	Online?	Sequel?	Genre	Publisher	Price	Console	Campaign_Avg_Duration	Other_Metadata
-------	--------------	---------	---------	-------	-----------	-------	---------	-----------------------	----------------



Title	Genre	Publisher	Console
-------	-------	-----------	---------

VGCHARTZ



GAMEDEV



METACRITIC



KAGGLE



CORGIS



Source Schema

VGCHARTZ



Id	Game	Console	Publisher	Developer	Sales	Year
----	------	---------	-----------	-----------	-------	------

vgsales[-by-console]

GAMEDEV



X_Coord	Y_Coord	Name	Type	City	Country	Website
---------	---------	------	------	------	---------	---------

gamedev

METACRITIC



Id	Title	User_Score	MC_Score	Platform	Year
----	-------	------------	----------	----------	------

metacritic

Id	Title	Genre
----	-------	-------

datagenreX

KAGGLE



Date	Game	Earnings	Players	#Tournaments
------	------	----------	---------	--------------

generalED

Game	ReleaseDate	Genre	TotalEarnings	OnlineEarnings	PricedPlayers	TotalTournaments
------	-------------	-------	---------------	----------------	---------------	------------------

historicalED

CORGIS

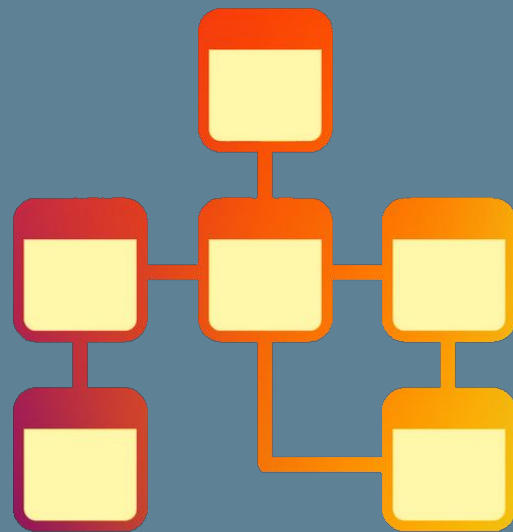


Title	Genre	Publisher	Console
-------	-------	-----------	---------

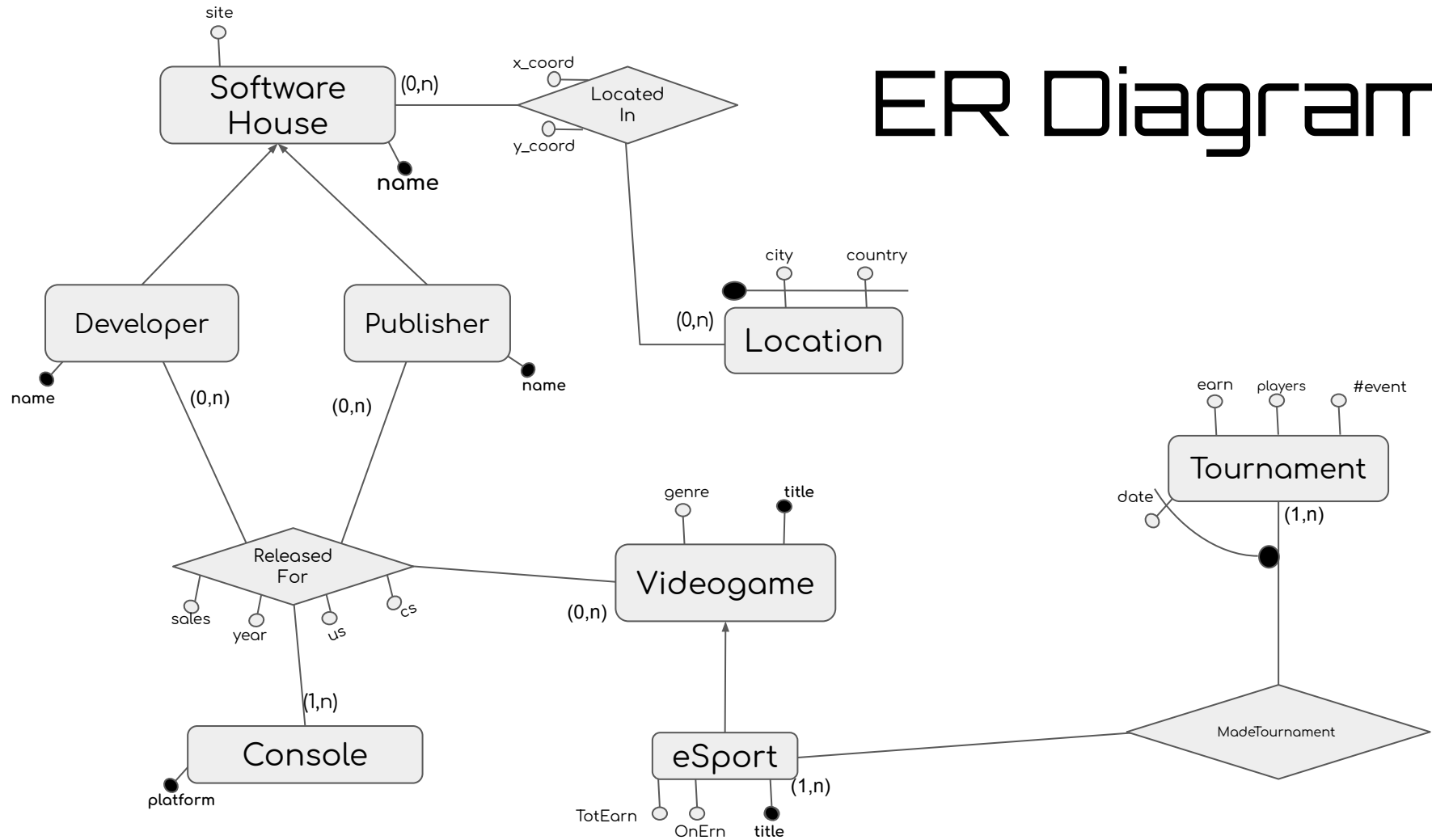
spvg

02.

GLOBAL SCHEMA



ER Diagram



Relational Schema

- SoftwareHouse(name,site)
- Location(city,country)
- LocatedIn(softwarehouse, city, country, x_coord, y_coord)
 - LocatedIn(softwarehouse) → SoftwareHouse(name)
 - LocatedIn(city.country) → Location(city.country)
- Videogame(title, genre)
- Console(platform)
- ReleasedFor(game, console,us,cs,sales,year,dev,pub)
 - ReleasedFor(game) → Videogame(title)
 - ReleasedFor(console) → Console(platform)
 - ReleasedFor(dev) → Developer(dev)
 - ReleasedFor(pub) → Publisher(pub)
- Developer(name)
 - Developer(name) → SoftwareHouse(name)
- Publisher(name)
 - Publisher(name) → SoftwareHouse(name)
- eSport(title,TotalEarnings,OnlineEarnings)
 - eSport(title) → Videogame(title)
- Tournament(date,game,earnings,#pricedplayers,#event)
 - Tournament(game) → eSport(title)

Global Schema

Global Schema Alphabet

- The set of all software houses:
- The set of all the Developing Software Houses:
- The set of all the Publishing Software Houses:
- The set of all Videogames:
- The set of all eSports Videogames:
- The set of eSports tournaments:
- The set of geographical locations
- Relation between software houses and their settlements:
- The set of all gaming consoles:
- Relation between the videogame and its console release:

SoftwareHouse_{/2}

Developer_{/1}

Publisher_{/1}

Videogame_{/2}

eSport_{/3}

Tournaments_{/5}

Location_{/2}

locatedIn_{/5}

Console_{/1}

releasedFor_{/8}

Global Schema

Global Schema Constraints (1)

Key constraints

- a1) $\forall \mathbf{t}. \forall ge. \forall ge'. \mathbf{Videogame}(\mathbf{t}, ge) \wedge \mathbf{Videogame}(\mathbf{t}, ge') \rightarrow ge = ge'$
- a2) $\forall \mathbf{n}. \forall w. \forall w'. \mathbf{SoftwareHouse}(\mathbf{n}, w) \wedge \mathbf{SoftwareHouse}(\mathbf{n}, w') \rightarrow w = w'$
- a3) $\forall \mathbf{g}. \forall \mathbf{c}. \forall us. \forall cs. \forall s. \forall y. \forall dev. \forall pub. \mathbf{ReleasedFor}(\mathbf{g}, \mathbf{c}, us, cs, s, y, dev, pub) \wedge \dots$
 $\dots \wedge \forall us'. \forall cs'. \forall s'. \forall y'. \forall dev'. \forall pub'. \mathbf{ReleasedFor}(\mathbf{g}, \mathbf{c}, us', cs', s', y', dev', pub')$
 $\rightarrow us = us' \wedge cs = cs' \wedge s = s' \wedge y = y' \wedge dev = dev' \wedge pub = pub'$
- a4) $\forall \mathbf{d}. \forall \mathbf{g}. \forall e. \forall pp. \forall ev. \forall e'. \forall pp'. \forall ev'. \mathbf{Tournament}(\mathbf{d}, \mathbf{g}, e, pp, ev') \wedge \dots$
 $\dots \wedge \mathbf{Tournament}(\mathbf{d}, \mathbf{g}, e', pp', ev') \rightarrow e = e' \wedge pp = pp' \wedge ev = ev'$
- a5) $\forall \mathbf{t}. \forall te. \forall oe. \forall te'. \forall oe'. \mathbf{eSport}(\mathbf{t}, te, oe) \wedge \mathbf{eSport}(\mathbf{t}, te', oe') \rightarrow te = te' \wedge oe = oe'$

Multiplicity Constraints

- a12) $\forall \mathbf{c}. \mathbf{Console}(\mathbf{c}) \rightarrow \exists g. \exists us. \exists cs. \exists s. \exists y. \exists dev. \exists pub. \mathbf{ReleasedFor}(g, \mathbf{c}, us, cs, s, y, dev, pub)$

Global Schema Constraints (2)

Referential Integrity Constraints

a6) $\forall d. \mathbf{Developer}(d) \rightarrow \mathbf{SoftwareHouse}(d)$

a7) $\forall p. \mathbf{Publisher}(p) \rightarrow \mathbf{SoftwareHouse}(p)$

a8) $\forall t. \mathbf{eSport}(t) \rightarrow \mathbf{Videogame}(t)$

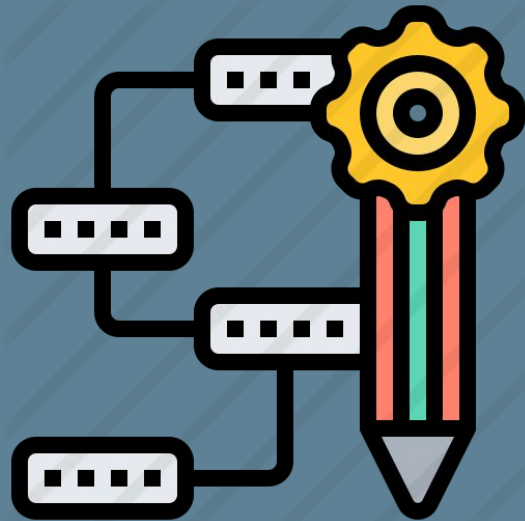
a9) $\forall d. \forall g. \forall e. \forall pp. \forall ev'. \mathbf{Tournament}(d, g, e, pp, ev') \rightarrow \mathbf{eSport}(g)$

a10) $\forall sh. \forall ci. \forall co. \forall x. \forall y. \mathbf{LocatedIn}(sh, ci, co, x, y) \rightarrow \mathbf{SoftwareHouse}(sh) \wedge \mathbf{Location}(ci, co)$

a11) $\forall g. \forall c. \forall us. \forall cs. \forall s. \forall y. \forall dev. \forall pub. \mathbf{ReleasedFor}(g, c, us, cs, s, y, dev, pub) \\ \rightarrow \mathbf{Console}(c) \wedge \exists ge. \mathbf{Videogame}(g, ge) \wedge \mathbf{Publisher}(pub) \wedge \mathbf{Developer}(dev)$

03.

MODELLING



Source-FOL, GLAV, Sound Mapping

SoftwareHouse, Publisher, Developer, Location

m1) $\forall x. \forall y. \forall n. \forall ci. \forall co. \exists t. \exists w. \text{gamedev}(x, y, n, t, ci, co, w) \rightarrow \text{SoftwareHouse}(n, w) \wedge \text{locatedIn}(n, ci, co, x, y) \wedge \text{Location}(ci, co)$

m2) $\forall n. \forall t. \exists x. \exists y. \exists ci. \exists co. \exists w. \text{gamedev}(x, y, n, t, ci, co, w) \wedge (t = \text{publisher}) \rightarrow \text{Publisher}(n)$

m3) $\forall n. \forall t. \exists x. \exists y. \exists ci. \exists co. \exists w. \text{gamedev}(x, y, n, t, ci, co, w) \wedge (t = \text{developer}) \rightarrow \text{Developer}(n)$

m4) $\forall \text{pub}. \exists id. \exists g. \exists c. \exists dev. \exists s. \exists y. \text{vg-sales}(id, g, c, \text{pub}, dev, s, y) \wedge \neg(\text{pub} = \text{NULL}) \rightarrow \exists w. \text{SoftwareHouse}(\text{pub}, w) \wedge \text{Publisher}(\text{pub})$

m5) $\forall \text{dev}. \exists id. \exists g. \exists c. \exists pub. \exists s. \exists y. \text{vg-sales}(id, g, c, pub, dev, s, y) \wedge \neg(\text{dev} = \text{NULL}) \rightarrow \exists w. \text{SoftwareHouse}(\text{dev}, w) \wedge \text{Developer}(\text{dev})$

m6) $\forall p. \exists t. \exists g. \exists c. \text{spvg}(t, g, p, c) \wedge \neg(p = \text{NULL}) \rightarrow \text{SoftwareHouse}(p) \wedge \text{Publisher}(p)$

Videogame, Console, ReleasedFor

$$\begin{aligned} \text{m7)} \quad & \forall \mathbf{g}. \forall \mathbf{c}. \forall \mathbf{s}. \forall \mathbf{y}. \exists \mathbf{id}. \exists \mathbf{pub}. \exists \mathbf{dev}. \mathbf{vgsales}(\mathbf{id}, \mathbf{g}, \mathbf{c}, \mathbf{pub}, \mathbf{dev}, \mathbf{s}, \mathbf{y}) \\ & \rightarrow \exists \mathbf{ge}. \mathbf{Videogame}(\mathbf{g}, \mathbf{ge}) \wedge \mathbf{Console}(\mathbf{c}) \wedge \exists \mathbf{us}. \exists \mathbf{cs}. \exists \mathbf{dev}. \exists \mathbf{pub}. \mathbf{ReleasedFor}(\mathbf{g}, \mathbf{c}, \mathbf{us}, \mathbf{cs}, \mathbf{s}, \mathbf{y}, \mathbf{dev}, \mathbf{pub}) \end{aligned}$$

$$\begin{aligned} \text{m8)} \quad & \forall \mathbf{g}. \forall \mathbf{c}. \forall \mathbf{dev}. \forall \mathbf{s}. \forall \mathbf{y}. \exists \mathbf{id}. \exists \mathbf{pub}. \mathbf{vgsales}(\mathbf{id}, \mathbf{g}, \mathbf{c}, \mathbf{pub}, \mathbf{dev}, \mathbf{s}, \mathbf{y}) \wedge \neg(\mathbf{dev} = \mathbf{NULL}) \\ & \rightarrow \exists \mathbf{us}. \exists \mathbf{cs}. \exists \mathbf{pub}. \mathbf{ReleasedFor}(\mathbf{g}, \mathbf{c}, \mathbf{us}, \mathbf{cs}, \mathbf{s}, \mathbf{y}, \mathbf{dev}, \mathbf{pub}) \end{aligned}$$

$$\begin{aligned} \text{m9)} \quad & \forall \mathbf{g}. \forall \mathbf{c}. \forall \mathbf{pub}. \forall \mathbf{s}. \forall \mathbf{y}. \exists \mathbf{id}. \exists \mathbf{dev}. \mathbf{vgsales}(\mathbf{id}, \mathbf{g}, \mathbf{c}, \mathbf{pub}, \mathbf{dev}, \mathbf{s}, \mathbf{y}) \wedge \neg(\mathbf{pub} = \mathbf{NULL}) \\ & \rightarrow \exists \mathbf{us}. \exists \mathbf{cs}. \exists \mathbf{dev}. \mathbf{ReleasedFor}(\mathbf{g}, \mathbf{c}, \mathbf{us}, \mathbf{cs}, \mathbf{s}, \mathbf{y}, \mathbf{dev}, \mathbf{pub}) \end{aligned}$$

$$\begin{aligned} \text{m10)} \quad & \forall \mathbf{t}. \forall \mathbf{ms}. \forall \mathbf{p}. \forall \mathbf{r}. \exists \mathbf{us}. \mathbf{metacritic}(\mathbf{t}, \mathbf{us}, \mathbf{ms}, \mathbf{p}, \mathbf{r}) \\ & \rightarrow \exists \mathbf{ge}. \mathbf{Videogame}(\mathbf{t}, \mathbf{ge}) \wedge \mathbf{Console}(\mathbf{p}) \wedge \exists \mathbf{us}. \exists \mathbf{s}. \exists \mathbf{dev}. \exists \mathbf{pub}. \mathbf{ReleasedFor}(\mathbf{t}, \mathbf{p}, \mathbf{us}, \mathbf{ms}, \mathbf{s}, \mathbf{r}, \mathbf{dev}, \mathbf{pub}) \end{aligned}$$

$$\text{m11)} \quad \forall \mathbf{t}. \forall \mathbf{us}. \forall \mathbf{p}. \forall \mathbf{ms}. \forall \mathbf{r}. \mathbf{metacritic}(\mathbf{t}, \mathbf{us}, \mathbf{ms}, \mathbf{p}, \mathbf{r}) \wedge \neg(\mathbf{us} = \mathbf{tdb}) \rightarrow \exists \mathbf{s}. \exists \mathbf{dev}. \exists \mathbf{pub}. \mathbf{ReleasedFor}(\mathbf{t}, \mathbf{p}, \mathbf{us}, \mathbf{ms}, \mathbf{s}, \mathbf{r}, \mathbf{dev}, \mathbf{pub})$$

$$\begin{aligned} \text{m12)} \quad & \forall \mathbf{t}. \forall \mathbf{c}. \forall \mathbf{g}. \exists \mathbf{p}. \mathbf{spvg}(\mathbf{t}, \mathbf{g}, \mathbf{p}, \mathbf{c}) \\ & \rightarrow \mathbf{Videogame}(\mathbf{t}, \mathbf{g}) \wedge \mathbf{Console}(\mathbf{c}) \wedge \exists \mathbf{us}. \exists \mathbf{ms}. \exists \mathbf{s}. \exists \mathbf{y}. \exists \mathbf{dev}. \exists \mathbf{pub}. \mathbf{ReleasedFor}(\mathbf{t}, \mathbf{p}, \mathbf{us}, \mathbf{ms}, \mathbf{s}, \mathbf{y}, \mathbf{dev}, \mathbf{pub}) \end{aligned}$$

$$\begin{aligned} \text{m13)} \quad & \forall \mathbf{t}. \forall \mathbf{p}. \forall \mathbf{c}. \exists \mathbf{g}. \mathbf{spvg}(\mathbf{t}, \mathbf{g}, \mathbf{p}, \mathbf{c}) \wedge \neg(\mathbf{p} = \mathbf{NULL}) \\ & \rightarrow \exists \mathbf{us}. \exists \mathbf{cs}. \exists \mathbf{s}. \exists \mathbf{y}. \exists \mathbf{dev}. \mathbf{ReleasedFor}(\mathbf{t}, \mathbf{c}, \mathbf{us}, \mathbf{cs}, \mathbf{s}, \mathbf{y}, \mathbf{dev}, \mathbf{p}) \end{aligned}$$

$$\text{m14)} \quad \forall \mathbf{t}. \forall \mathbf{g}. \mathbf{datagenreX}(\mathbf{t}, \mathbf{g}) \rightarrow \mathbf{Videogame}(\mathbf{t}, \mathbf{g})$$

$$\text{m15)} \quad \forall \mathbf{g}. \forall \mathbf{ge}. \exists \mathbf{r}. \exists \mathbf{te}. \exists \mathbf{oe}. \exists \mathbf{pp}. \exists \mathbf{tt}. \mathbf{generalED}(\mathbf{g}, \mathbf{r}, \mathbf{ge}, \mathbf{te}, \mathbf{oe}, \mathbf{pp}, \mathbf{tt}) \rightarrow \mathbf{Videogame}(\mathbf{g}, \mathbf{ge})$$

Source-FOL, GLAV, Sound Mapping (3)

eSport, Tournament

m16) $\forall \mathbf{g}.\forall \mathbf{te}.\forall \mathbf{oe}.\exists \mathbf{r}.\exists \mathbf{ge}.\exists \mathbf{pp}.\forall \mathbf{tt}.\mathbf{generalED}(\mathbf{g}, \mathbf{r}, \mathbf{ge}, \mathbf{te}, \mathbf{oe}, \mathbf{pp}, \mathbf{tt}) \rightarrow \mathbf{eSport}(\mathbf{g}, \mathbf{te}, \mathbf{oe})$

m17) $\forall \mathbf{d}.\forall \mathbf{g}.\forall \mathbf{e}.\forall \mathbf{pp}.\forall \mathbf{ne}.\mathbf{historicalED}(\mathbf{d}, \mathbf{g}, \mathbf{e}, \mathbf{pp}, \mathbf{ne}) \rightarrow \mathbf{Tournament}(\mathbf{d}, \mathbf{g}, \mathbf{e}, \mathbf{pp}, \mathbf{ne})$

04.

IMPLEMENTATION

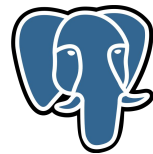


Design

Given our I.I.S. $J = \langle G = \{a1...a12\}, M = \{m1...m17\}, S = \emptyset \rangle$,
being A the set of axioms over A_G (**global schema constraints**),
being M our set of **Source-FOL**, **GLAV**, **sound mapping assertions...**

Materialization approach for computing Certain Answers:

- ETL: Pentaho
 - Winning **Physical Heterogeneity** (wrappers + extraction)
 - Winning **Conceptual Heterogeneity** (transformation)
 - **PostgreSQL** Materialization (loading)



VGCHARTZ



GAMEDEV



METACRITIC



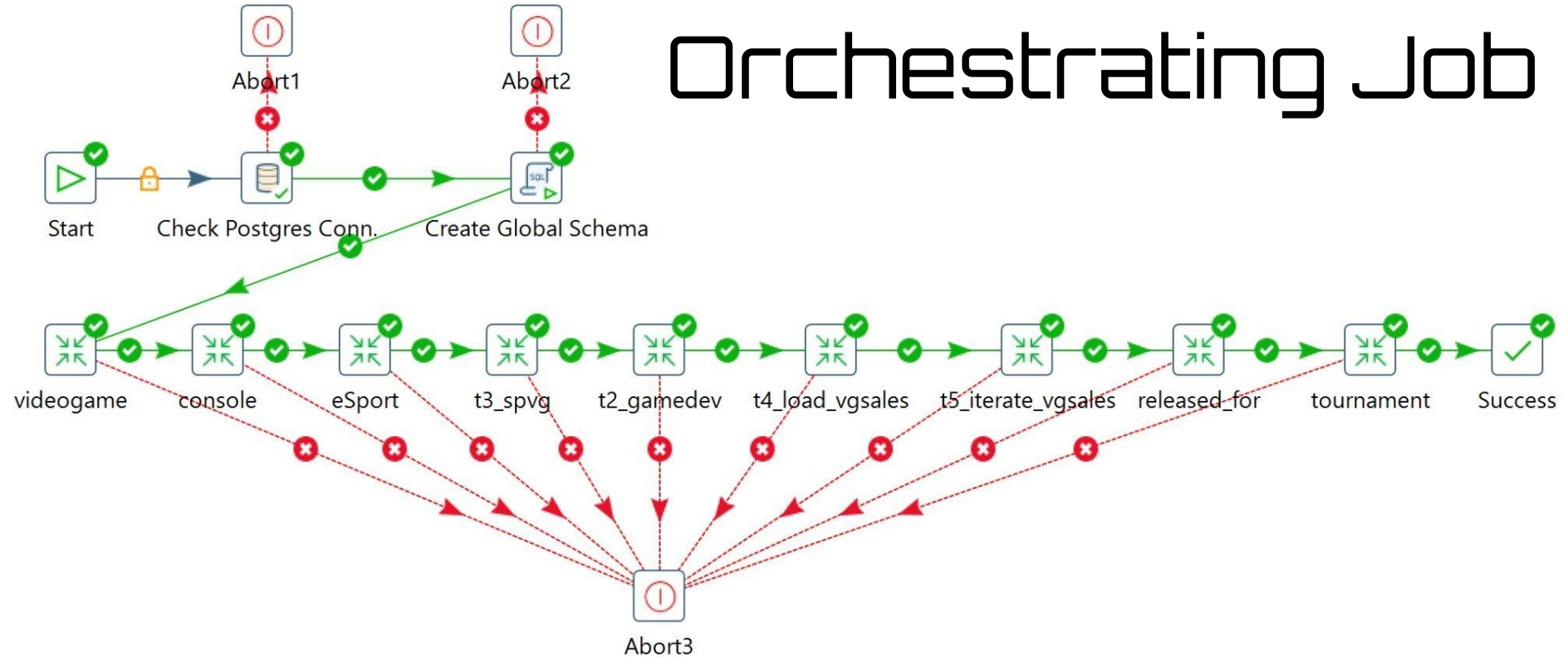
KAGGLE



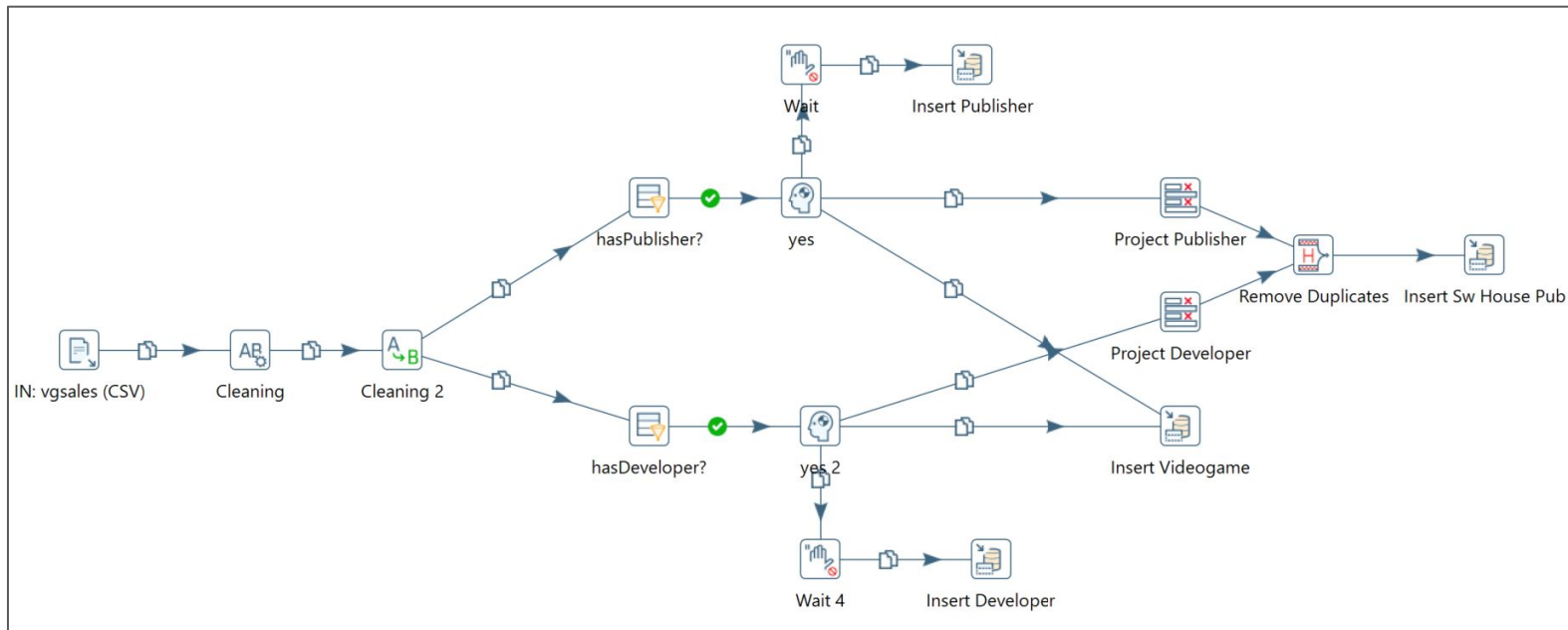
CORGIS



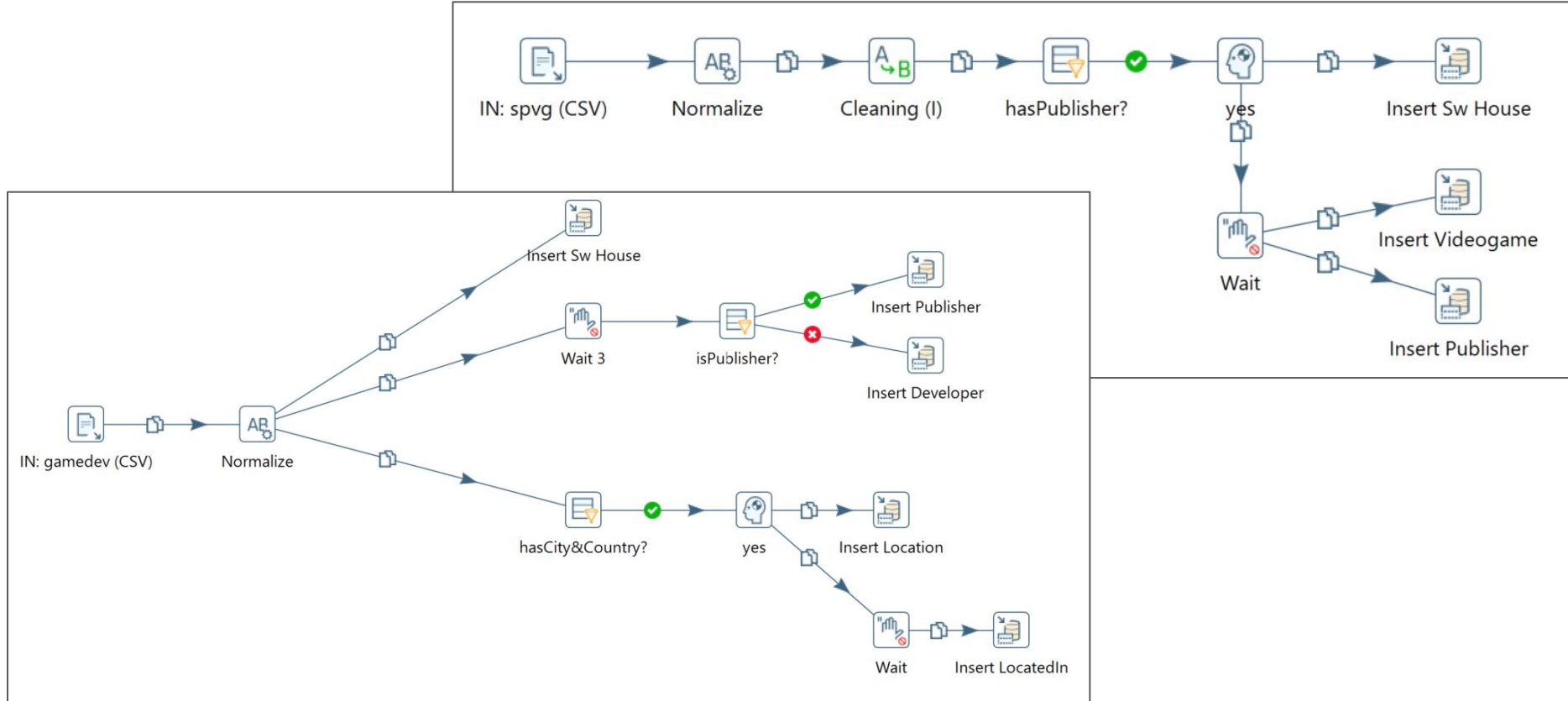
Pentaho Orchestrating Job



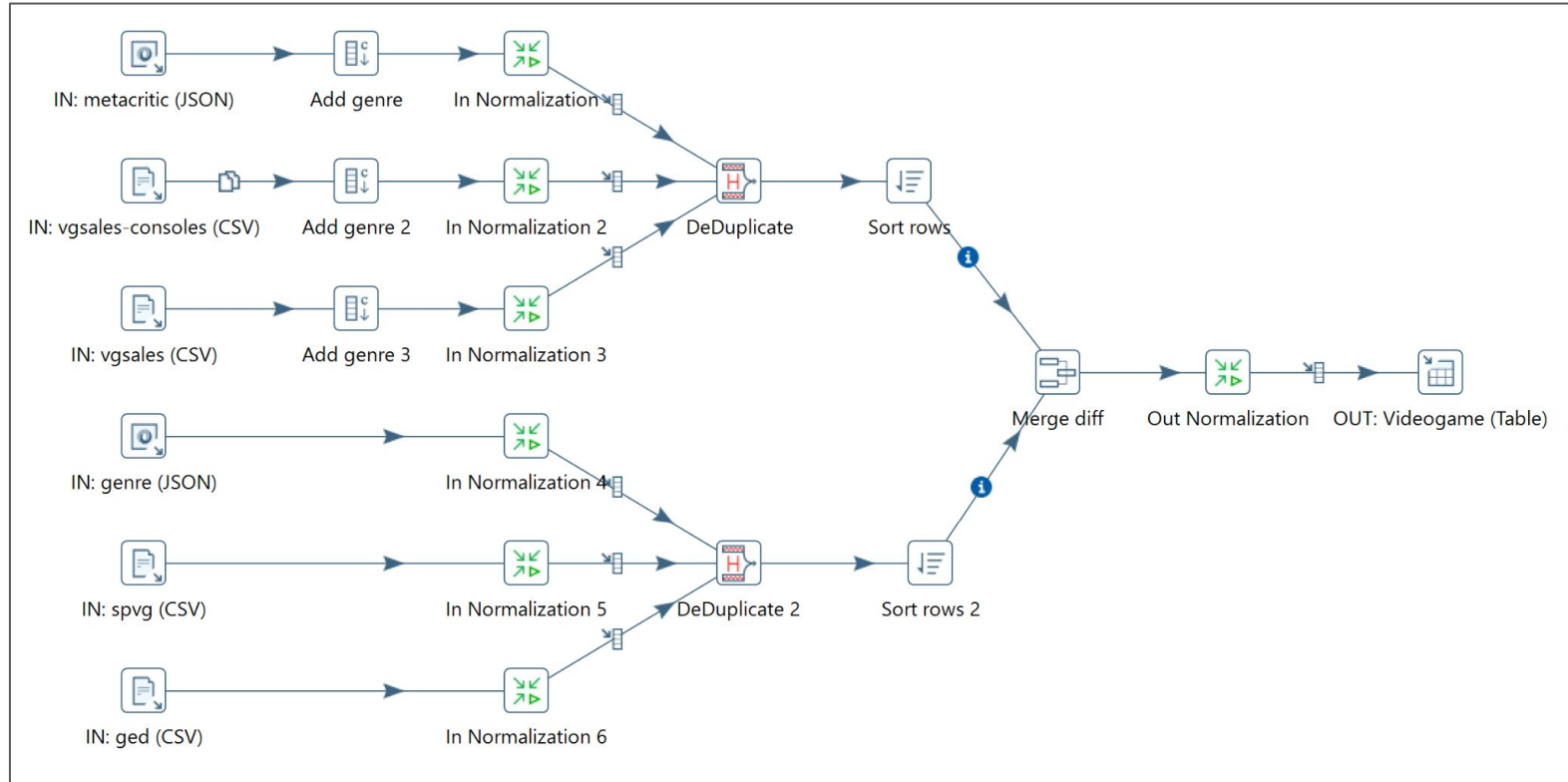
Pentaho: VGChartz Transformation



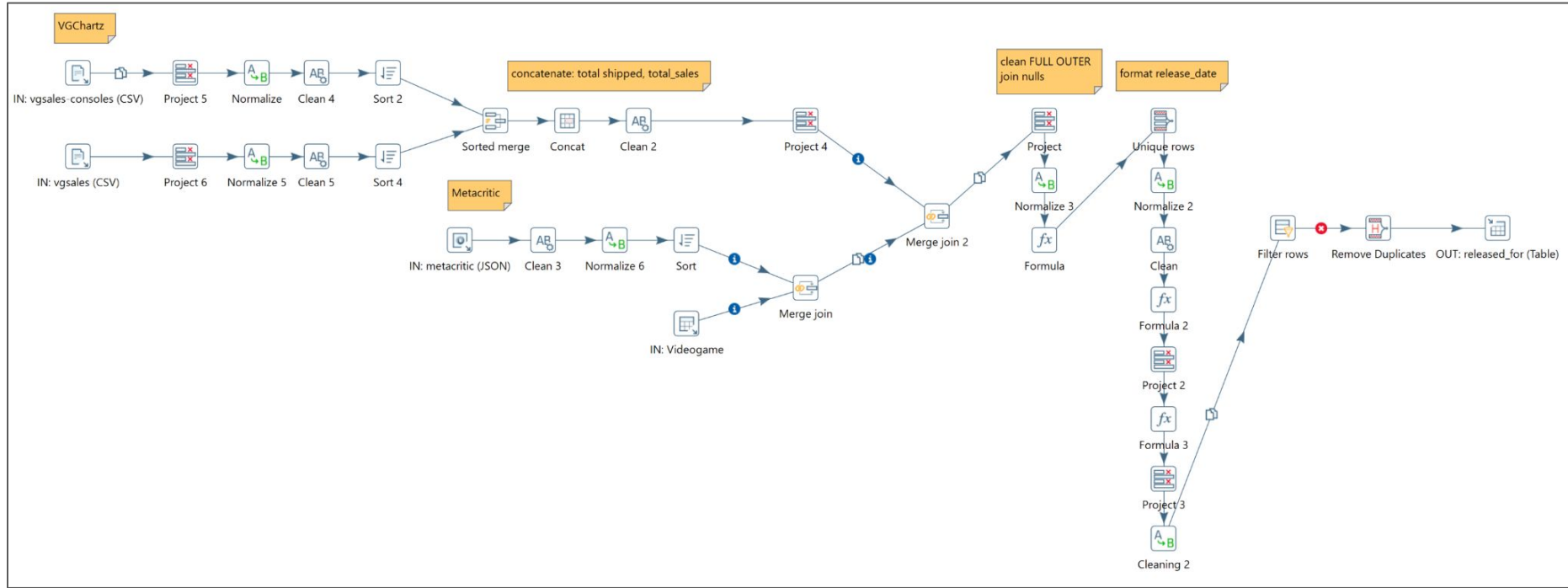
Pentaho: Gamedev, SPVG



Pentaho: Videogame



Pentaho: ReleasedFor



PostgreSQL: Materialized Solution












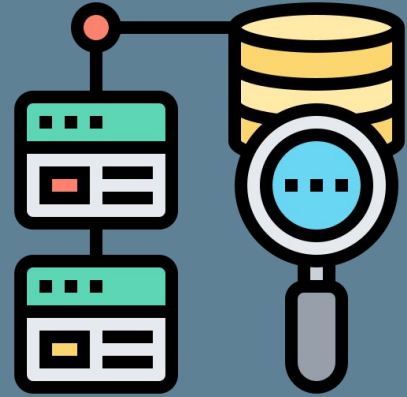
- ▼  Tables (10)
 - >  console
 - >  developer
 - >  esports
 - >  locatedin
 - >  location
 - >  publisher
 - >  released_for
 - >  softwarehouse
 - >  tournament
 - >  videogame

Table name	Tuples inserted
console	39
developer	8,050
esport	513
locatedin	4,140
location	1,052
publisher	1,852
released_for	37,535
softwarehouse	8,491
tournament	484
videogame	34,591

05.

QUERIES



Queries

Given our I.I.S. $J = \langle G = \{a1...a12\}, M = \{m1...m17\}, S = \emptyset \rangle$,
being A the set of axioms over A_G (**global schema constraints**),
being M our set of **Source-FOL, GLAV, sound mapping assertions...**

We finally have a **materialized solution** (of the source database, w.r.t. the mapping M) to query in order to compute **certain answers**.

Note: being in a GLAV-materialization scenario, our expressive power is limited to **UCQs**.

Query + Certain Answers (1)

WEBSITES OF SOFTWARE HOUSES THAT ARE BOTH DEVELOPERS AND PUBLISHERS

$$q = \{(w) | \exists x. \text{SoftwareHouse}(x, w) \wedge \text{Publisher}(x) \wedge \text{Developer}(x)\}$$
$$\text{PROJ}_2(\text{SEL}_{1=3 \wedge 3=4}((\text{SoftwareHouse} \times \text{Developer}) \times \text{Publisher}))$$



SELECT website

FROM SoftwareHouse

*WHERE name IN (SELECT * FROM Publisher)*

*AND name IN (SELECT * FROM Developer)*

AND website NOTNULL

	website	
	character varying (300)	
84	http://www.enmasse.com	
85	http://www.encore.com	
86	http://www.engine-software.c...	
87	http://www.epicgames.com	
88	http://www.exorstudios.com	
89	http://www.facepunchstudios....	
90	http://www.farsightstudios.co...	
91	http://www.fatshark.se	
92	http://www.finji.co	
 Successfully run. Total query runtime: 134 msec. 337 rows affected.		

VIDEOGAMES THAT WERE RELEASED BEFORE 2000, FOR ANY CONSOLE

$$q = \left\{ (x, y) \mid \begin{array}{l} \exists ge. \mathbf{Videogame}(x, ge) \wedge (y < 2000) \wedge \dots \\ \dots \wedge \exists c. \exists us. \exists cs. \exists s. \exists dev. \exists pub. \mathbf{ReleasedFor}(g, c, us, cs, s, y, dev, pub) \end{array} \right\}$$

$PROJ_{1,8}(SEL_{1=3 \wedge 8 < "2000"}(Videogame \times ReleasedFor))$

SELECT title, release_date

FROM Videogame

NATURAL JOIN ReleasedFor

WHERE release_date < 2000

	title character varying (340)	release_date double precision
1	007 tomorrow never dies	1999
2	1 on 1	1998
3	10 pin bowling	1999
4	100 manyen quiz hunter	1998
5	10101 will the starship	1997
6	1080° teneighty snowboarding	1998
7	10yard fight	1985
8	1002 ueno betay ukeno reache	1999

Query + Certain Answers (2)

✓ Successfully run. Total query runtime: 78 msec. 7841 rows affected.

Query + Certain Answers (3)

VIDEOGAMES DEVELOPED OR PUBLISHED BY SOFTWARE HOUSES LOCATED IN ITALY

$$q = \left\{ (x) \left| \begin{array}{l} \exists c. \exists us. \exists cs. \exists s. \exists y. \exists dev. \exists pub. \textbf{ReleasedFor}(x, c, us, cs, s, y, dev, pub) \wedge \dots \\ \dots \wedge \exists ci. \exists x_co. \exists y_co. (\textbf{locatedIn}(dev, ci, "italy", x_co, y_co) \vee \textbf{locatedIn}(pub, ci, "italy", x_co, y_co)) \end{array} \right. \right\}$$

$PROJ_1 \left(SEL_{7=9 \wedge 11="italy"}(ReleasedFor \times locatedIn) \right) \cup PROJ_1 \left(SEL_{7=9 \wedge 11="italy"}(ReleasedFor \times locatedIn) \right)$

$SELECT \textit{distinct title}$
 $FROM ReleasedFor$
 $JOIN locatedIn \textit{ON publisher = softwarehouse}$
 $WHERE country = "italy"$
 $UNION$
 $SELECT \textit{distinct title}$
 $FROM ReleasedFor$
 $JOIN locatedIn \textit{ON developer = softwarehouse}$
 $WHERE country = "italy"$

	title	
	character varying (340)	
12	...	
13	rageball	
14	miami nights singles in the city	
15	we dare flirty fun for all	
16	ozzy drix	
17	srs street racing syndicate	
18	motto me de unou o kitaeru s...	
19	tomb raider the prophecy	
20	baldo	
✔ Successfully run. Total query runtime: 380 msec. 128 rows affected.		

Query + Certain Answers (4)

VIDEOGAMES PUBLISHED BY USA PUBLISHER

$$q = \left\{ (t, p) \mid \exists ci. \exists x. \exists y. \exists w. \exists y. \exists pl. \exists r. \exists us. \exists ms. \exists s. \exists d. \text{LocatedIn}(p, ci, \text{united states}, x, y) \right. \\ \left. \wedge (\text{ReleasedFor}(t, pl, r, us, ms, s, d, p)) \right\}$$

$PROJ_{1,6}(SEL_{1=13 \wedge 3=\text{"united states"}}(LocatedIn \times ReleasedFor))$

SELECT distinct title, softwarehouse

FROM locatedin, released_for

WHERE locatedin.softwarehouse = releasedfor.publisher AND country = 'united states'

	 title character varying (340)	 publisher character varying (400)	
1	lizzie mcguire 2 lizzie diaries	disney interactive studios	
2	un squadron	capcom	
3	road not taken	spry fox	
4	criminal girls 2 party favors	nis america	
5	1942	capcom	
6	ohno odyssey	big john games	
7	the bridge	the quantum astrophysicists ...	
8	dungeons dragons tower of d...	capcom	
9	harvest moon 3 gbc	natsume	
10	will o wisp ds	idea factory	
11	akira psychoball	atari	
12	the occupation	humble bundle	
13	blazblue continuum shift	aksys games	
14	hakuoki reimeiroku omouhas...	idea factory	
15	disney epic mickey the power ...	disney interactive studios	
16	ark survival evolved	studio wildcard	
17	the witcher enhanced edition	atari	
18	high school musical livin the ...	disney interactive studios	
19	disneys meet the robinsons	disney interactive studios	
20	disneys chip n dale rescue ran...	capcom	
21	zack wiki quest for barbaros t...	capcom	

Query - certain answer (5)

GENRES THAT ARE USED IN ESPORT GAMES

$q = \{(ge) | \exists t. \exists te. Videogame(t, ge) \wedge Esport(t, te, oe)\}$

$PROJ_2(SEL_{1=3}((Videogame \times Esport)))$

SELECT genre

FROM videogame

WHERE videogame.title IN (SELECT title FROM esport)

	genre character varying (200)	
1	multiplayer online battle arena	
2	strategy	
3	collectible card game	
4	first-person shooter	
5	racing	
6	sports	
7	puzzle game	
8	role-playing game	
9	fighting game	
10	third-person shooter	
11	battle royale	

Query - certain answer (6)

VIDEOGAMES THAT WERE RELEASED BOTH FOR PS3 AND PC

$$q = \left\{ (x) \mid \begin{array}{l} \exists c. \exists us. \exists cs. \exists s. \exists y. \exists dev. \exists pub. \text{ReleasedFor}(x, c, us, cs, s, y, dev, pub) \wedge (c = PS3) \wedge \dots \\ \dots \wedge \exists c'. \exists us'. \exists cs'. \exists s'. \exists y' \exists dev'. \exists pub'. \text{ReleasedFor}(x, c', us', cs', s', y', dev', pub') \wedge (c' = PC) \end{array} \right\}$$

$PROJ_1 \left(SEL_{2="PS3"}(ReleasedFor) \right) \cap PROJ_1 \left(SEL_{2="PC"}(ReleasedFor) \right)$

SELECT title

FROM ReleasedFor



WHERE console = "PC"

INTERSECT

SELECT title

FROM ReleasedFor

WHERE console = "PS3"

	title	
	character varying (340)	
15	resident evil revelations 2	
16	ruse	
17	the elder scrolls v skyrim	
18	timeshift	
19	far cry 4	
20	shovel knight	
21	sleeping dogs	
22	call of duty modern warfare 3	
 Successfully run. Total query runtime: 72 msec. 224 rows affected.		

Query - insight (1)

CONSOLES WITH THE MOST SELLS AND THEIR RELATIVE MOST SELLER GAME

```
select t1.platform, max(t1.sales)as most_sells, foo.title as title_most_sold,count(foo.title) as Ngames
from released_for as t1
inner join (
  select title, sales, platform
  from released_for as t2
  where (platform, sales) IN
  (select platform, max(sales)
   from released_for
   group by(platform)
  )
) as foo
on t1.platform = foo.platform
group by(t1.platform,foo.title)
order by (most_sells) DESC
```

	platform character varying (10)	most_sells double precision	title_most_sold character varying (340)	ngames bigint
1	PC	4000	counterstrike global offensive	1084
2	Wii	3738	mario kart wii	812
3	NS	3341	mario kart 8 deluxe	1653
4	GB	3138	pokémon red green blue vers...	1473
5	DS	3080	new super mario bros	3153
6	NES	2831	duck hunt	839
7	X360	2400	kinect adventures	423
8	SNES	2061	super mario world	791
9	PS4	2000	marvels spiderman	997
10	3DS	1892	mario kart 7	1181
11	GBA	1622	pokémon ruby sapphire versi...	1437
12	GEN	1500	sonic the hedgehog	643
13	PS2	1489	gran turismo 3 aspec	2655
14	PS3	1195	gran turismo 5	1182
15	N64	1191	super mario 64	284

Query - insight (2)

COUNTRIES WHO EARNED THE MOST FROM SALES PER YEAR

```
select release_date, country, concat(ceil(sum(sales)), ' mln')
from (select distinct country, sales, release_date
      from hascountry
      join locatedin on hascountry.city = locatedin.city
      join publish on locatedin.softwarehouse = publish.publisher
      join released_for on videogame=title
      where sales notnull and release_date notnull) as x
group by country, x.release_date
having round(sum(sales)) >= (
  select round(sum(sales))
  from (select distinct country, sales, release_date
        from hascountry
        join locatedin on hascountry.city = locatedin.city
        join publish on locatedin.softwarehouse = publish.publisher
        join released_for on videogame=title
        where sales notnull and release_date notnull) as y
  where x.release_date = y.release_date
  group by country
  order by round(sum(sales)) desc limit 1)
order by release_date desc
```

	release_date double precision	country character varying (300)	concat text
1	2021	japan	500 mln
2	2021	united states	500 mln
3	2020	japan	5547 mln
4	2019	japan	6259 mln
5	2018	japan	6943 mln
6	2017	japan	12336 mln
7	2016	japan	5294 mln
8	2015	japan	2843 mln
9	2014	japan	6025 mln
10	2013	japan	8136 mln
11	2012	japan	5540 mln
12	2011	japan	9785 mln
13	2010	united states	8731 mln
14	2009	japan	14516 mln
15	2008	japan	10753 mln
16	2007	japan	12328 mln
17	2006	japan	10687 mln

Queries

Query - insight (3)

COUNTRIES WHO EARNED THE MOST FROM TOURNAMENTS

```
select country, concat(ceil(sum(cast(totalearnings as decimal))/1000000),' mln')
from (
  select distinct country, totalearnings
  from hascountry
  join locatedin on hascountry.city = locatedin.city
  join publish on locatedin.softwarehouse = publish.publisher
  join esports on videogame=title) as x
group by country
order by sum(cast(totalearnings as decimal)) desc
```

	country character varying (300)	concat text
1	united states	176 mln
2	china	35 mln
3	south korea	34 mln
4	singapore	33 mln
5	france	33 mln
6	italy	32 mln
7	spain	32 mln
8	netherlands	32 mln
9	brazil	32 mln
10	australia	32 mln
11	ireland	32 mln
12	taiwan	32 mln
13	japan	15 mln

Query - insight (4)

COUNTRIES WHO PUBLISHED MOST VIDEOGAMES

```
select country, count(*)
from (
  select distinct country, videogame
  from hascountry
  join locatedin on hascountry.city = locatedin.city
  join publish on locatedin.softwarehouse = publish.publisher
) as x
group by country
order by count(*) desc
```

	 country character varying (300)	 gamespublished bigint
1	japan	3533
2	united states	2290
3	germany	1595
4	england	634
5	canada	525
6	france	121
7	netherlands	98
8	south korea	98
9	singapore	95
10	china	95
11	spain	81
12	italy	77
13	poland	66

Queries

Query - insight (5)

COUNTRIES WHO PUBLISHED MOST VIDEOGAMES

```
select country, count(name)
from softwarehouse
join locatedin on softwarehouse.name = locatedin.softwarehouse
join hascountry on locatedin.city = hascountry.city
group by country
order by count desc, country asc
```

	country character varying (300)	count bigint
1	united states	1423
2	canada	397
3	england	366
4	france	258
5	germany	190
6	japan	159
7	china	115
8	australia	92
9	sweden	90
10	spain	86

Queries

Query - insight (6)

PUBLISHERS WHO EARNED THE MOST FROM SALES PER YEAR

```
select release_date, publisher, concat(ceil(sum(sales)), ' mln')
from (select distinct publisher, sales, release_date
      from publish
      join released_for on videogame=title
      where sales notnull and release_date notnull) as x
group by publisher, release_date
having round(sum(sales)) >= (
  select round(sum(sales))
  from (select distinct publisher, sales, release_date
        from publish
        join released_for on videogame=title
        where sales notnull and release_date notnull) as y
  where x.release_date = y.release_date
  group by publisher
  order by round(sum(sales)) desc limit 1)
order by release_date desc
```

	release_date double precision	publisher character varying (300)	concat text
1	2021	capcom	500 mln
2	2020	nintendo	5503 mln
3	2019	nintendo	6250 mln
4	2018	nintendo	6545 mln
5	2017	nintendo	11321 mln
6	2016	nintendo	4228 mln
7	2015	nintendo	2107 mln
8	2014	nintendo	5463 mln
9	2013	nintendo	6938 mln
10	2012	nintendo	4989 mln
11	2011	nintendo	8174 mln
12	2010	nintendo	5522 mln
13	2009	nintendo	12394 mln

Queries

06.

CONCLUSIONS

We presented an Information Integration System based on multiple videogame industry sources.

At first we gathered all the sources we needed, then we have organized them in order to build a global schema and through mapping, axioms and the ER-diagram, we realized all the fundamental formalisms in order to be capable of implementing a **robust** database built with Pentaho connected to PostgreSQL.

In the end we tested this system with different queries getting very interesting results.

Conclusions



SAPIENZA
UNIVERSITÀ DI ROMA

Thanks for the attention!

Longo Valerio 1655653
Papi Alessio 1761063



<https://github.com/1655653/LSDM>

