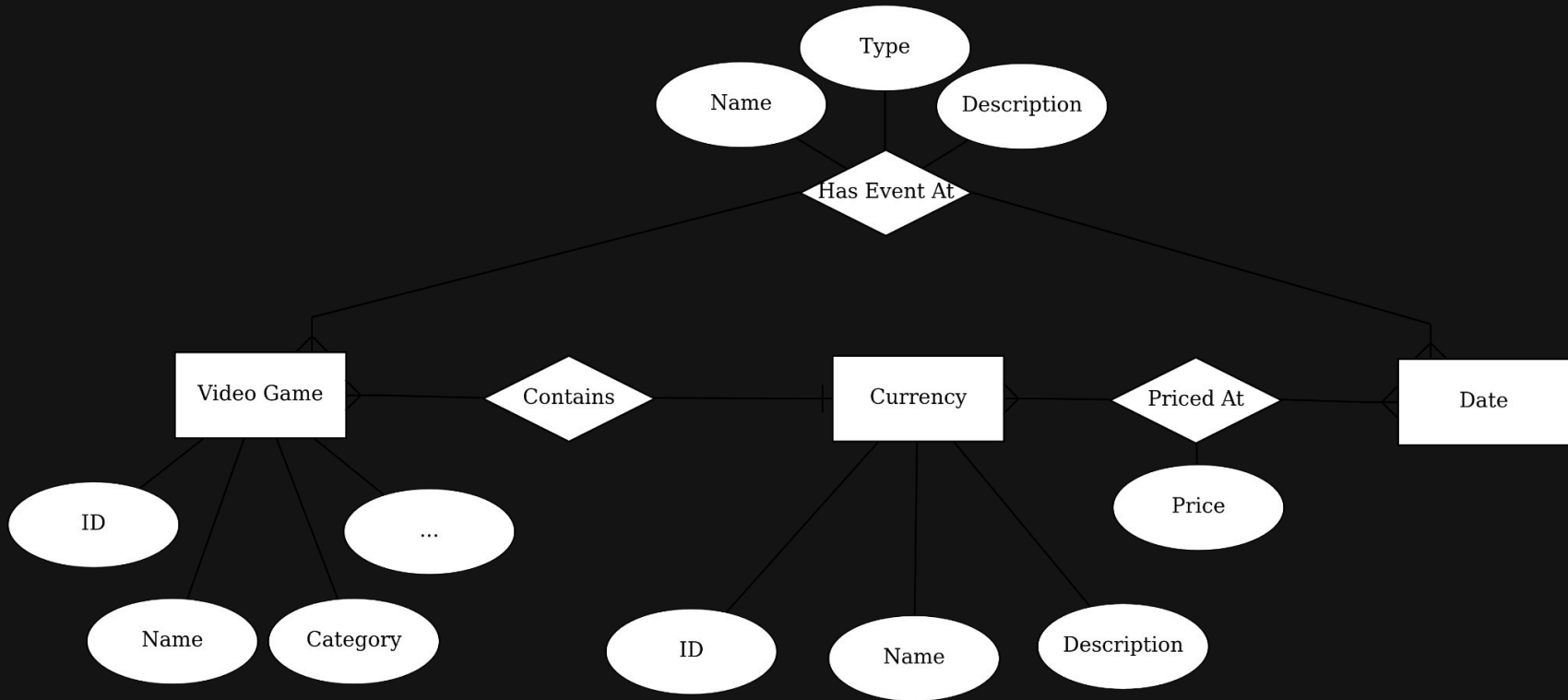


Construção de dataset para histórico de ***in-game currencies*** (Entrega Parcial)

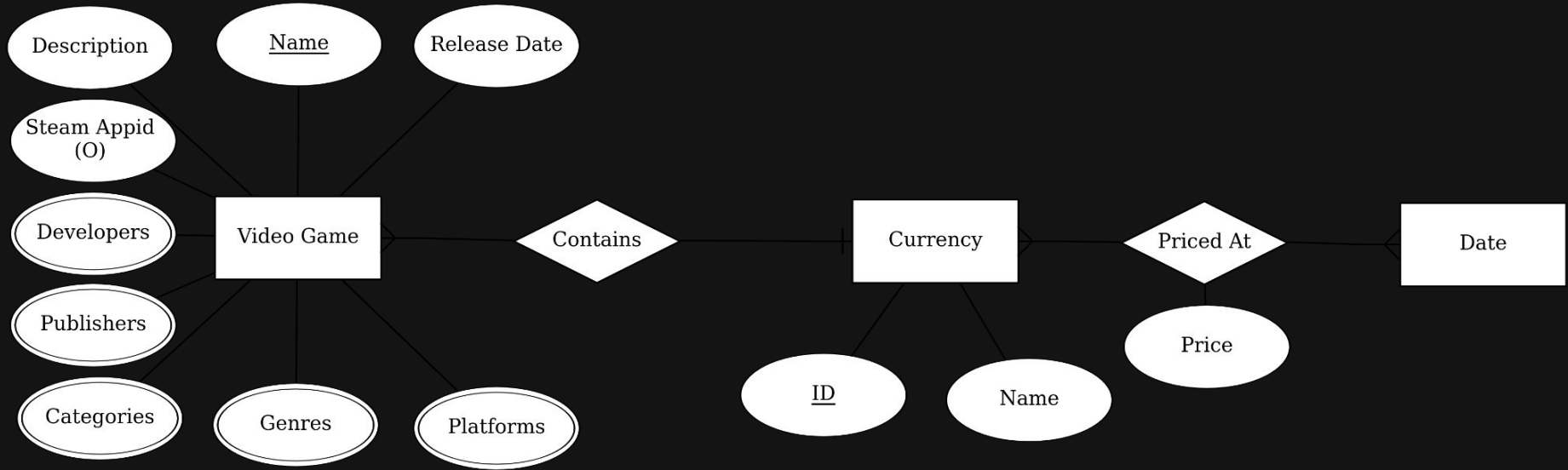
Resumo da Premissa

- ❖ *In-game currencies* são moedas virtuais utilizadas dentro de economias de jogos para compra/venda de itens. Geralmente possuem um valor atrelado em dinheiro real.
- ❖ De certa forma, são precursoras das atuais criptomoedas.
- ❖ Como existem a muito mais tempo, possuem uma quantidade maior de dados a serem explorados.
- ❖ Apesar disso, pesquisa no ramo é escassa e carece de fontes de dados compreensivas.
- ❖ **Proposta:** Um banco de dados contendo histórico de preços de moedas virtuais extraídos de diversas fontes, além de informações sobre seus respectivos jogos

Modelo Conceitual - Esboço Inicial



Modelo Conceitual - Esboço Atual



Modelos Lógicos

- ❖ Metadados de video games em formato JSON e Tabelas de Preços em formato CSV

Template utilizado
para guardar as
informações dos
jogos (Modelo de
documentos)

```
{ name: String,  
  description: String  
  steam: {  
    steam_game: bool,  
    appid: int  
  },  
  developers: [String],  
  publishers: [String],  
  platforms: [String],  
  genres: [String],  
  keywords: [String],  
  release_date: String,  
  currencies: [  
    {  
      id: String,  
      name: String,  
      price: {  
        fixed_price: int,  
        unit: int,  
        price_table: String  
      }  
    }  
  ]  
}
```

COIN(*Date*, *Price*)

Tabelas de Preços
(Modelo Relacional)

Modelos Lógicos

- ❖ Metadados de video games em formato JSON e Tabelas de Preços em formato CSV

Template utilizado
para guardar as
informações dos
jogos (Modelo de
documentos)

```
1 {
2   "name": "Final Fantasy XIV Online",
3   "description": "Take part in an epic and ever-changing the massively multiplayer online role-playing game (MMORPG) in a FINAL FANTASY game.\n\nForge friendships, make friends to undertake epic quests, battle familiar FINAL FANTASY enemies and love are waiting! Embark on the adventure of a lifetime!\n\n",
4   "steam": {
5     "steam_game": false,
6     "appid": null
7   },
8   "developers": [
9     "Square Enix"
10  ],
11  "publishers": [
12    "Square Enix"
13  ],
14  "platforms": [
15    "PC (Microsoft Windows)",
16    "PlayStation 3",
17    "PlayStation 4",
18    "PlayStation 5"
19  ],
20  "genres": [
21    "Role-playing (RPG)",
22    "Strategy",
23    "Adventure"
24  ],
25  "keywords": [],
26  "release_date": "2013-08-26",
27  "currencies": [
28    {
29      "id": "FFXIV",
30      "name": "Final Fantasy XIV Gil",
31      "price": {
32        "fixed_price": null,
33        "unit": 1000000.0,
34        "price_table": "../prices/FFXIV.csv"
35      }
36    }
37  ]
38 }
39
```

	date	price
0	2019-11-15	0.54512916
1	2019-11-16	0.54612894
2	2019-11-17	0.54773894
3	2019-11-18	0.53638545
4	2019-11-19	0.53478497
5	2019-11-20	0.52603063
6	2019-11-21	0.54710124
7	2019-11-22	0.56101775
8	2019-11-23	0.55097659
9	2019-11-24	0.54816689
10	2019-11-25	0.5280544
11	2019-11-26	0.52943581
12	2019-11-27	0.54993934
13	2019-11-28	0.55710786
14	2019-11-29	0.56868051
15	2019-11-30	0.5858149
16	2019-12-01	0.55394324
17	2019-12-02	0.54370801
18	2019-12-03	0.54212721
19	2019-12-04	0.55285276
20	2019-12-05	0.55023268

Tabelas de Preços
(Modelo Relacional)

Fontes de Dados - Preços



Fontes de Dados - Metadados



Obtenção e Transformação de Dados

- ❖ Até o momento, somente por APIs
- ❖ Nenhum site necessitou de scraping*
- ❖ Porém, para resolver problemas de compatibilidade entre dados foram necessárias diversas conversões ao longo do caminho

PlayerAuctions



Obtenção e Transformação de Dados

- ❖ Scripts para obter e converter automaticamente os dados

```
10 def get_steam_game_metadata(appid):
11     url = f"https://store.steampowered.com/api/appdetails?appids={appid}&cc=US"
12     req = requests.get(url).json()[str(appid)]
13     if req["success"]:
14         game = game_info.copy()
15         data = req["data"]
16         game["name"] = data["name"]
17         game["description"] = data["short_description"]
18         game["steam"]["steam_game"] = True
19         game["steam"]["appid"] = data["steam_appid"]
20         game["developers"] = data["developers"]
21         game["publishers"] = data["publishers"]
22         game["keywords"] = data["categories"]
23         game["platforms"] = data["platforms"]
24         game["genres"] = data["genres"]
25         game["release_date"] = datetime.strptime(data["release_date"]["date"], "%b %d, %Y").strftime("%Y-%m-%d")
26     return game
27 return None
```

Obtenção e Transformação de Dados

- ❖ Scripts para obter e converter automaticamente os dados

```
18 def get_igdb_games_metadata(ids, log=False):
36
37     req = requests.post(url=url, data=data, headers=headers).json()
38
39     for game in req:
40         game_json = game_info.copy()
41         game_json["name"] = game["name"]
42         if log:
43             print(f"Name: {game_json['name']}")
44         if "summary" in game:
45             game_json["description"] = game["summary"]
46             game_json["steam"]["steam_game"] = False
47             game_json["steam"]["appid"] = None
48
49         if "involved_companies" in game:
50             for company in game["involved_companies"]:
51                 ic_data = f"fields *; where id = {company};"
52                 ic_req = requests.post(url=ic_url, data=ic_data, headers=headers).json()[0]
53                 company_id = ic_req["company"]
54                 c_data = f"fields name; where id = {company_id};"
55                 company_name = requests.post(url=c_url, data=c_data, headers=headers).json()[0]["name"]
56                 if ic_req["developer"]:
57                     game_json["developers"].append(company_name)
58                 if ic_req["publisher"]:
59                     game_json["publishers"].append(company_name)
60
61         if log:
62             print("Developed by:")
63             for dev in game_json['developers']:
64                 print(f"- {dev}")
65             print("Published by:")
66             for pub in game_json['publishers']:
67                 print(f"- {pub}")
68
69         if "keywords" in game:
70             for keyword in game["keywords"]:
71                 k_data = f"fields name; where id = {keyword};"
72                 keyword_name = requests.post(url=k_url, data=k_data, headers=headers).json()[0]["name"]
73                 game_json["keywords"].append(keyword_name)
```

Obtenção e Transformação de Dados

- ❖ Scripts para obter e converter automaticamente os dados

```
40 def price_history_json_to_csv(json_obj, filename):
41     df = price_history_json_to_dataframe(json_obj)
42     df.to_csv(filename)
43
44 def price_history_json_to_dataframe(json_obj):
45     x = []
46     y = []
47     prices = json_obj["prices"]
48     for price in prices:
49         price_date = datetime.strptime(price[0], "%b %d %Y %H: +0").date()
50         price_value = price[1]
51         x.append(price_date)
52         y.append(price_value)
53     return pd.DataFrame(np.array(list(zip(x, y))), columns=["date", "price"])
54
```

Obtenção e Transformação de Dados

- ❖ Scripts para obter e converter automaticamente os dados

```
25 def get_binance_data(symbol, date, interval=intervals['daily']):
26     base_url = "https://data.binance.vision"
27     year = date.year
28     month = date.month
29     url = f"{base_url}/data/spot/monthly/klines/{symbol}/{interval}/{symbol}-{interval}-{year}-{month:02}.zip"
30
31     req = requests.get(url, allow_redirects=True)
32
33     with ZipFile(BytesIO(req.content)) as zf:
34         zf.extractall("./extracted/")
35
36     with open(f"./extracted/{symbol}-{interval}-{year}-{month:02}.csv", "r") as file:
37         df = pd.read_csv(file, header=None, usecols=[0, 1], names=["date", "opening_price"])
38         df["date"] = df["date"].apply(lambda x: datetime.utcfromtimestamp(x/1000))
39     return df
40
```

Questões iniciais

- ❖ Como o comportamento dessas moedas virtuais se compara com o de moedas tradicionais?
- ❖ Similarmente, como seu comportamento se compara com o de criptomoedas?
- ❖ Como, e em que grau, elas são afetadas por eventos internos (como updates) e externos (como quedas no mercado)
- ❖ É possível prever seu comportamento com uso de algoritmos? (mais adequada para pesquisas de machine learning)

Questões baseadas em Consultas SQL

- ❖ Como conseguir o histórico da taxa de câmbio entre duas moedas?

$$1 \text{ A} = X \text{ USD} \rightarrow 1 \text{ USD} = 1/X \text{ A}$$

$$1 \text{ B} = Y \text{ USD} \rightarrow 1 \text{ USD} = 1/Y \text{ B}$$



$$1/X \text{ A} = 1/Y \text{ B}$$



$$1 \text{ A} = X/Y \text{ B}$$

Questões baseadas em Consultas SQL

- ❖ Como conseguir o histórico da taxa de câmbio entre duas moedas?

```
1 SELECT O.Date Date,  
2       (O.Price/F.Price) OSRS_FFXIV,  
3       (O.Price/G.Price) OSRS_GTA,  
4       (O.Price/B.Price) OSRS_BTC,  
5       (O.Price/E.Price) OSRS_ETH  
6 FROM OSRS O, FFXIV F, GTA G, BTC B, ETH E  
7 WHERE O.Date=F.Date AND  
8       O.Date=G.Date AND  
9       O.Date=B.Date AND  
10      O.Date=E.Date  
11      LIMIT 10;
```


	Date	OSRS_FFXIV	OSRS_GTA	OSRS_BTC	OSRS_ETH
1	2017-08-17	0.666667	4.571429	0.000300	0.004251
2	2017-08-18	0.742857	43.333333	0.000303	0.004305
3	2017-08-19	1.133929	42.333333	0.000309	0.004330
4	2017-08-20	0.915493	43.333333	0.000315	0.004492
5	2017-08-21	0.867133	124.000000	0.000305	0.004146
6	2017-08-22	1.272727	63.000000	0.000314	0.003925
7	2017-08-23	0.887324	63.000000	0.000312	0.004035
8	2017-08-24	0.879433	41.333333	0.000299	0.003927
9	2017-08-25	0.816993	62.500000	0.000290	0.003865
10	2017-08-26	0.765432	124.000000	0.000290	0.003789



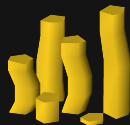
Questões baseadas em Consultas SQL

- ❖ Como é o comportamento médio de uma moeda X ao longo dos anos?

```
1 SELECT YEAR(O.Date) Year,  
2     AVG(O.Price) AVERAGE,  
3     MIN(O.Price) MIN,  
4     MAX(O.Price) MAX,  
5     STDDEV(O.Price) VOLATILITY  
6 FROM OSRS O  
7 GROUP BY Year;  
8
```



	Year	AVERAGE	MIN	MAX	VOLATILITY
1	2017	1.171367	0.98	1.76	0.101313
2	2018	0.950329	0.74	2.12	0.173994
3	2019	0.728466	0.53	1.23	0.151648
4	2020	0.624454	0.44	0.88	0.112294
5	2021	0.543822	0.48	0.63	0.02833



Questões baseadas em Consultas SQL

- ❖ Como é o comportamento médio de uma moeda X ao longo dos anos?

VOLATILITY
4287.182375
2415.151228
2649.926123
4206.256496
9617.358708



VOLATILITY
142.850054
296.044856
50.528983
142.680594
851.260247



VOLATILITY
0.518749



VOLATILITY
0.101313
0.173994
0.151648
0.112294
0.02833



VOLATILITY
0.646746
1.075212
0.778538
0.702268
0.1721



VOLATILITY
0.104157
0.152044
0.14021
0.031183
0.011673



Questões baseadas em Consultas SQL

- ❖ (Baseada na anterior) Como aumentar a precisão? (Meses/Dias)

```
1 SELECT YEAR(A.Date) Year,  
2     MONTH(A.Date) Month,  
3     AVG(A.Price) AVERAGE,  
4     MIN(A.Price) MIN,  
5     MAX(A.Price) MAX,  
6     STDDEV(A.Price) VOLATILITY  
7 FROM ADA A  
8 GROUP BY Year, Month  
9 ORDER BY Year, Month;
```



	Year	Month	AVERAGE	MIN	MAX	VOLATILITY
1	2021	3	1.170000	1.03	1.38	0.082345
2	2021	4	1.249667	1.09	1.48	0.098437
3	2021	5	1.640968	1.27	2.30	0.259657
4	2021	6	1.491333	1.16	1.85	0.180365
5	2021	7	1.282581	1.05	1.46	0.096151
6	2021	8	2.067097	1.31	2.95	0.571897
7	2021	9	2.455000	1.99	2.97	0.280021
8	2021	10	2.157742	1.91	2.28	0.086945



Questões baseadas em Consultas SQL

- ❖ Como a diferença entre moedas evolui ao longo dos anos?

```
1 CREATE VIEW OSRS_ETH AS SELECT O.Date, (O.Price/E.Price) Price FROM OSRS O, ETH E WHERE O.Date=E.Date;
2
3 SELECT YEAR(O.Date) Year,
4         AVG(O.Price) AVERAGE,
5         MIN(O.Price) MIN,
6         MAX(O.Price) MAX,
7         STDDEV(O.Price) VOLATILITY
8 FROM OSRS_ETH O
9 GROUP BY Year;
```



	Year	AVERAGE	MIN	MAX	VOLATILITY
1	2017	0.0032936788	0.001301	0.006083	0.0009942517
2	2018	0.0030780247	0.000935	0.011701	0.0024948681
3	2019	0.0044294548	0.002121	0.009673	0.0018996765
4	2020	0.0026501120	0.000643	0.006472	0.0014735126
5	2021	0.0002525888	0.000118	0.000727	0.0001015943

Coisas serem consideradas

- ❖ Stablecoins
- ❖ “Meme” coins
- ❖ Diferenças de preço da mesma moeda entre servers
- ❖ Diferenças de valor entre fontes de dados
- ❖ Uniformidade dos períodos de tempo dos dados

Problemas - API Backpack.tf

Internal Server Error



Something Went Wrong

An internal error occurred. This problem has been logged and will be investigated.

