Ciobanu Mihaela Lavinia
Master Ingineria Datelor, An II

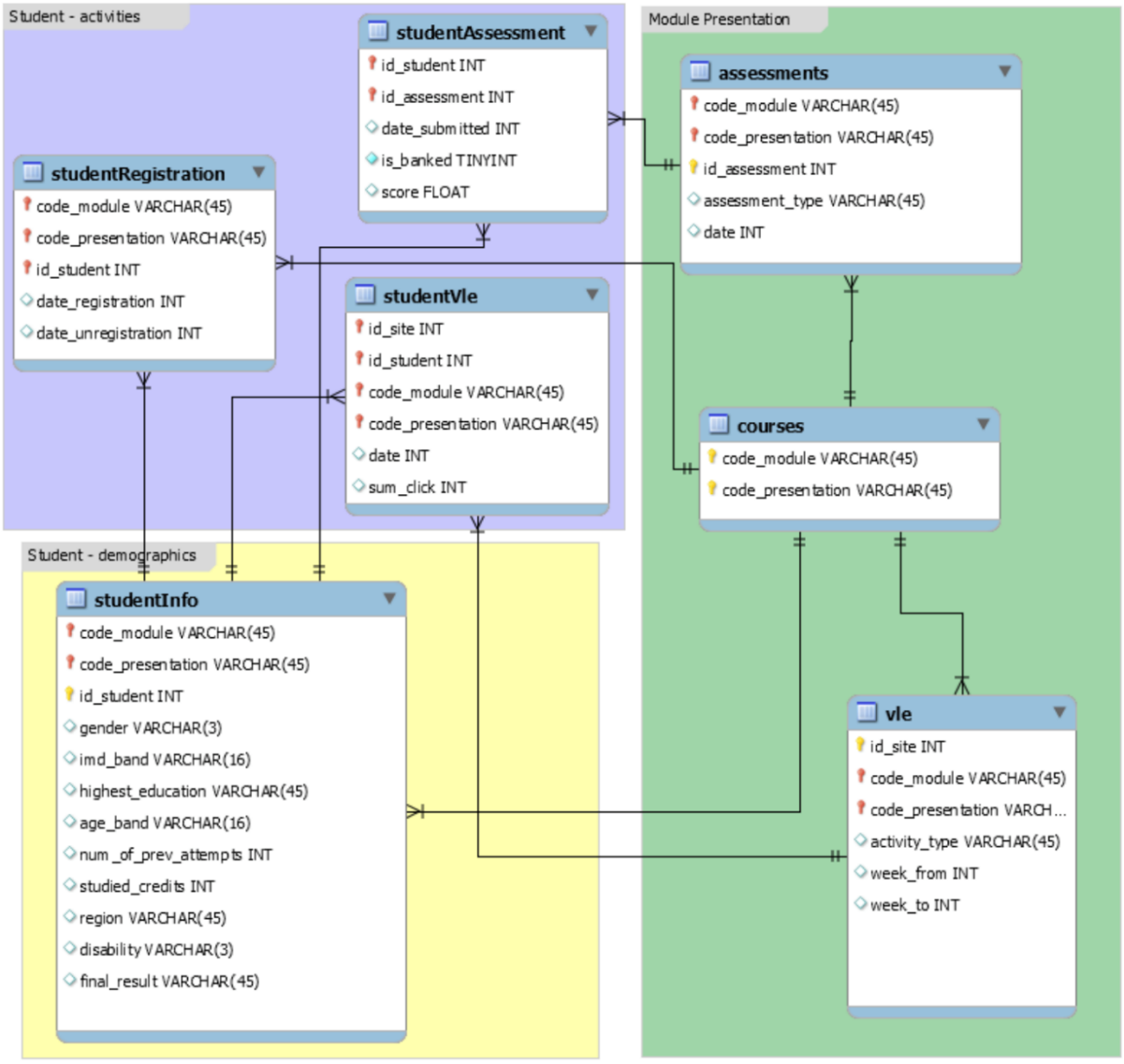
Proiect Tehnologii Big Data Federated Learning

1. Introducere și Descrierea Datelor

Proiectul vizează dezvoltarea unui sistem de procesare distribuită pentru analiza comportamentului studenților. Am utilizat setul de date OULAD (Open University Learning Analytics Dataset).

Acesta este un set de date relațional complex, care leagă informații demografice, rezultate ale evaluărilor și, cel mai important, jurnalele de activitate (log-uri) ale interacțiunii studenților cu platforma VLE (Virtual Learning Environment).

Pentru a înțelege relațiile dintre tabele și necesitatea procesării Big Data, avem diagrama bazei de date mai jos:



Provocarea tehnică principală a fost tabelul studentVle, care conține milioane de înregistrări tranzacționale, necesitând o arhitectură distribuită pentru procesare.

2. Setul de Date

Am implementat o arhitectură de tip ETL (Extract, Transform, Load) utilizând Apache Spark. Procesul este automatizat și divizat în trei etape majore, conform obiectivelor proiectului.

2.1. Ingestia și Curățarea Datelor (Obiectivul O1)

În prima etapă, am preluat fișierele CSV brute. Deoarece formatul CSV nu este eficient pentru interogări analitice, am realizat conversia în format Parquet.

În secvența de cod de mai jos, se observă cum Spark citește datele brute și le salvează în format optimizat, adăugând un marcaj de timp pentru trasabilitate:

```
# Citim fisierul CSV original din folderul raw
# inferSchema=True permite Spark sa detecteze automat daca o coloana e numar sau text
df = spark.read.csv(cfg["paths"]["raw"] + ds["input_file"], header=True, inferSchema=True)

# Adaugam o coloana cu data si ora procesarii (pentru trasabilitate)
df = df.withColumn("ingest_timestamp", current_timestamp())

# Definim calea de iesire catre zona 'interim'
out_dir = cfg["paths"]["interim"] + ds["interim_dir"]

#Salvam datele in format Parquet
ensure_dir(out_dir)
df.write.mode("overwrite").parquet(out_dir)
```

Un pas critic în această etapă a fost Feature Engineering-ul pentru scenariul federat. Am creat un identificator unic compus **(client_id)** pentru a putea separa datele în funcție de curs, fără a le amesteca.

```
# Identifica unic o instanta de curs (ex: AAA_2013J) care va deveni un "client" federat
if "code_module" in df.columns and "code_presentation" in df.columns:
    df = df.withColumn("client_id", concat_ws("_", col("code_module"), col("code_presentation")))
```

2.2. Integrarea și Agregarea Datelor

Tabelul de activitate (student_vle) conține date la nivel de click. Pentru a prezice promovabilitatea, a trebuit să transformăm aceste date în profiluri la nivel de student.

Am realizat o agregare masivă folosind Spark SQL, reducând dimensiunea datelor de la milioane de rânduri la câteva mii de profiluri unice.

```
# Transformam datele TRANZACTIONALE (click-uri individuale) in date COMPORTAMENTALE (profil student).
# Aici are loc reducerea numarului de randuri
df = spark.sql("""
    SELECT v.client_id, v.id_student, SUM(v.sum_click) as total_clicks,
    COUNT(DISTINCT v.date) as days_active, i.label, AVG(s.score) as avg_score
    FROM vle v
    JOIN info i ON v.id_student = i.id_student AND v.client_id = i.client_id
    LEFT JOIN sa s ON v.id_student = s.id_student
    GROUP BY v.client_id, v.id_student, i.label
""")
```

Raportul generat automat de sistem evidențiază reducerea drastică a volumului de date prin sinteză, fără pierderea informației relevante:

```
1. DATE PROCESATE
----------------------------------------------------
Rânduri în dataset-ul principal (student_vle): 8459320
Rânduri în dataset-ul final de analiză dupa join: 29228
----------------------------------------------------
```

3. Scenarii de Analiză și Rezultate

Pe baza datelor procesate, am rulat trei scenarii analitice.

3.1. Analiza Exploratorie (EDA)

Am comparat comportamentul digital al studenților care promovează versus cei care pică examenele. Rezultatele statistice extrase din sistem arată o corelație clară:

Rezultate Statistice:

- **Studenții promovați (Label 1):** Medie de **11.023 click-uri** și **69 zile** de activitate.
- **Studenții respinși (Label 0):** Medie de **3.879 click-uri** și **35 zile** de activitate.

```
----------------------------------------------------

3. ANALIZA DATELOR
----------------------------------------------------
EDA REPORT
Row(label=1, count=22550, avg_clicks=11023.217294900222, avg_days=69.80727272727273, avg_score=75.2713961337254)
Row(label=0, count=6678, avg_clicks=3879.733303384247, avg_days=35.03519017669961, avg_score=64.48216969482488)

----------------------------------------------------
```

Studenții care promovează sunt de aproape 3 ori mai activi pe platformă decât cei care eșuează.

3.2. Învățare Automată Centralizată vs. Federată

Am antrenat un model de **Regresie Logistică** în două moduri distincte.

A. Abordarea Centralizată (Baseline):

Antrenarea modelului pe toate datele la un loc a stabilit performanța de referință.

Rezultat ML Centralizat: AUC Score = **0.754**

```
4. REZULTATE MACHINE LEARNING CENTRALIZAT
------------------------------------------------
Model: Logistic Regression
AUC Score: 0.7547410506109425

------------------------------------------------
```

B. Simulare Federated Learning: Am simulat un mediu în care datele nu părăsesc serverul cursului (client_id). Sistemul a iterat prin toate cursurile disponibile și a antrenat modele locale.

```
------------------------------------------------

5. SIMULARE FEDERATED LEARNING
------------------------------------------------
REZULTATE SIMULARE FEDERATED LEARNING
Total clienti (cursuri): 22
Modele antrenate cu succes: 22
Clienti ignorati (date insuficiente): 0
================================================================================
```

4. Concluzii

Proiectul a demonstrat succesul implementării unui pipeline Big Data complet. Prin utilizarea Apache Spark, am reușit:

1. Să procesăm eficient peste **8.4 milioane** de înregistrări.
2. Să asigurăm o calitate a datelor de 100% (0 valori nule în raportul final).
3. Să validăm ipoteza că arhitecturile federate sunt o alternativă viabilă tehnic pentru protejarea datelor educaționale, reușind antrenarea descentralizată a modelelor pe 22 de partiții distincte.