# Education_Bronze_Generation_Notebook

New notebook

## Medallion Architecture - Bronze Tier

In [ ]:
```
from pyspark.sql import functions as F

# ====== CONFIG ======
BASE = "Files/send_education_synthetic"    #lakehouse location of raw csv files #"/
BRONZE_DB = "bronze"
SILVER_DB = "silver"
GOLD_DB   = "gold"

# Create databases (schemas)
''' This is a trial version so cannot create databases or schema
spark.sql(f"CREATE DATABASE IF NOT EXISTS {BRONZE_DB}")
spark.sql(f"CREATE DATABASE IF NOT EXISTS {SILVER_DB}")
spark.sql(f"CREATE DATABASE IF NOT EXISTS {GOLD_DB}")
'''

# If trial version, you cannot create databases or schema\nspark.sql(f"CREATE DATAB
```

## Read source files

In [ ]:
```
def read_csv_any(path: str):
    print(path)
    # Works for single CSV or folder of CSV parts
    return spark.read.option("header", True).option("inferSchema", False).csv(path)

def write_bronze(df, table_name: str):
    print(f"{BRONZE_DB}_{table_name}")
    (df.write.mode("overwrite").format("delta").saveAsTable(f"{BRONZE_DB}_{table_na

# ====== DIMENSIONS (BRONZE) ======
write_bronze(read_csv_any(f"{BASE}/dim_date.csv"),         "dim_date")
write_bronze(read_csv_any(f"{BASE}/dim_geography.csv"),    "dim_geography")
write_bronze(read_csv_any(f"{BASE}/dim_trust.csv"),        "dim_trust")
write_bronze(read_csv_any(f"{BASE}/dim_school.csv"),       "dim_school")
write_bronze(read_csv_any(f"{BASE}/dim_pupil.csv"),        "dim_pupil")
write_bronze(read_csv_any(f"{BASE}/dim_send_need.csv"),    "dim_send_need")
write_bronze(read_csv_any(f"{BASE}/dim_intervention.csv"), "dim_intervention")

# ====== FACTS (BRONZE) ======
write_bronze(read_csv_any(f"{BASE}/fact_enrolment.csv"),                 "fact_enro
write_bronze(read_csv_any(f"{BASE}/fact_attendance_daily.csv"),          "fact_atte
write_bronze(read_csv_any(f"{BASE}/fact_attainment_termly.csv"),         "fact_atta
write_bronze(read_csv_any(f"{BASE}/fact_behaviour_incidents.csv"),       "fact_beha
write_bronze(read_csv_any(f"{BASE}/fact_send_provision.csv"),            "fact_send
write_bronze(read_csv_any(f"{BASE}/fact_intervention_participation.csv"),"fact_inte
```

```
print("Bronze complete.")
```

- Not part of Bronze workflow - just a snippet to add column if adding a column is required

In [ ]:
```
from pyspark.sql.utils import AnalysisException

table_name = "gold_dimsendneed"
column_name = "PrimaryNeedDesc"

try:
    columns = [f.name for f in spark.table(table_name).schema]

    if column_name not in columns:
        spark.sql(f"""
            ALTER TABLE {table_name}
            ADD COLUMNS ({column_name} STRING)
        """)
        print("Column added.")
    else:
        print("Column already exists.")

except AnalysisException:
    print(f"Table {table_name} does not exist.")
```