Generating Unique Identifiers/ Primary key in PySpark

1. UUID-Based Identifier

- Generates a **Universally Unique Identifier (UUID)** for each row.
- UUIDs are 128-bit randomly generated values that are unique across all systems.
- Useful when there is **no natural unique key** in the dataset.

Implementation in PySpark

```
from pyspark.sql.functions import expr

contacts_df = contacts_df.withColumn("contact_id", expr("uuid()"))
```

2. Composite Key

- Creates a **concatenated string** using multiple columns.
- Create a **natural key** when no single column is unique.

Implementation in PySpark

```
from pyspark.sql.functions import concat_ws

contacts_df = contacts_df.withColumn("contact_id", concat_ws("_", "first_name", "last_name", "email"))
```

3. Hash-Based Identifier

- Creates a **SHA-256 hash** from selected fields.

Implementation in PySpark

```
from pyspark.sql.functions import sha2

contacts_df = contacts_df.withColumn("contact_id", sha2(concat_ws("_", "first_name", "last_name", "email"), 256))
```