Here’s a **comprehensive checklist** to validate **DB2-to-Oracle Q Replication (QREP)** data migration and detect **data mismatch issues**—critical when ensuring data accuracy, integrity, and type compatibility across platforms.

**✅ DB2-to-Oracle QREP Migration Validation Checklist**

**🔹 1. Pre-Migration: Schema & Data Type Mapping**

| **Task** | **Check** |
| --- | --- |
| 📋 Match data types | Ensure DB2 data types (e.g., BIGINT, DECIMAL, TIMESTAMP) map correctly to Oracle types (NUMBER, VARCHAR2, etc.) |
| 🔍 Validate column length & scale | VARCHAR(100) in DB2 ≠ VARCHAR2(100 CHAR) in Oracle if NLS settings differ |
| 🧠 Null constraints | NULL vs NOT NULL must be aligned, especially for columns with default values |
| ⚙ Primary/unique key presence | Ensure keys exist in Oracle for apply-side replication accuracy |
| 🎯 Index/constraint strategies | Decide whether to disable constraints/indexes before load for performance and re-enable after |

**🔹 2. Character Encoding / NLS Settings**

| **Task** | **Check** |
| --- | --- |
| 🧬 Charset compatibility | DB2 EBCDIC/UTF-8 vs Oracle AL32UTF8—ensure mapping doesn't corrupt special characters |
| 🧾 NLS\_LANG on Oracle client | Set NLS\_LANG to match Oracle DB charset to avoid garbage characters |
| 💡 String padding behavior | CHAR in DB2 is blank-padded, Oracle comparisons may behave differently |

**🔹 3. LOB and CLOB Data Handling**

| **Task** | **Check** |
| --- | --- |
| 📦 LOB transfer optimization | Enable LOB\_SEND\_OPTIMIZATION in Q Apply to stream large LOBs |
| 🔍 Truncate check | Oracle CLOB max size = 4 GB; verify source values don't exceed it |
| 📤 Inline vs out-of-line | Match storage options for performance tuning (SECUREFILE LOBs preferred in Oracle) |

**🔹 4. Numeric Precision & Overflow**

| **Task** | **Check** |
| --- | --- |
| 🧮 Precision match | DECIMAL(31,10) in DB2 must map to Oracle NUMBER(31,10) or greater |
| 🔁 Rounding/overflow checks | Validate Oracle handles values without overflow errors |
| 📊 Sample high-precision test | Run test cases with edge-value data (e.g., max/min decimal and BIGINT) |

**🔹 5. Date/Time Consistency**

| **Task** | **Check** |
| --- | --- |
| ⏱ Timestamp precision | DB2 supports microseconds; Oracle TIMESTAMP must match (TIMESTAMP(6) recommended) |
| 🌍 Time zone handling | Align DB2 and Oracle time zone awareness (DBTIMEZONE, SESSIONTIMEZONE) |
| 🔁 Date format transformation | Ensure consistent format if using intermediate staging (e.g., flat files or Kafka) |

**🔹 6. Transactional Consistency & Row Comparison**

| **Task** | **Check** |
| --- | --- |
| 🔄 Row counts | Validate row counts for each table on both sides (use COUNT(\*)) |
| 🔍 Checksum validation | Run checksum comparison on rows/columns using MD5/SHA1 |
| 🔎 Sampling spot checks | Randomly select and compare 100+ rows across tables |
| 📤 Logging enablement | Enable Q Apply trace (APPLY\_TRACE) and Q Capture logs to review rejected/missing rows |

**🔹 7. Q Apply Settings Validation**

| **Task** | **Check** |
| --- | --- |
| 🔁 Conflict detection enabled | Enable to handle duplicates, missing PKs, or constraint violations |
| ⏩ Performance tuning | Batch size (COMMIT\_COUNT, GROUP\_COMMITS) optimized for Oracle |
| ⚙ LOB\_SEND\_OPTIMIZATION | Ensures efficient LOB replication |
| 🗃 Target schema match | Q Apply must use the correct Oracle schema/table name mapping |

**🔹 8. Error Logging & Auditing**

| **Task** | **Check** |
| --- | --- |
| 📜 Q Apply error tables | Check IBMQREP\_APPLYTRC, IBMQREP\_APPLYERR for issues |
| 🧾 Oracle bad/discard logs | If using SQL\*Loader, check .bad and .log files |
| 🧭 Audit tables or triggers | Optional: Create audit logging to track data drift post-load |

**🔹 9. Testing and Cutover**

| **Task** | **Check** |
| --- | --- |
| 🧪 Parallel run test | Run QREP in parallel with production load for drift detection |
| 🧍 Shadow tables | Use shadow tables in Oracle for dry-run validation |
| ⛳ Cutover plan with rollback | Prepare for fallback in case of serious mismatch post-migration |

**🧰 Recommended Tools for Validation**

| **Tool** | **Use** |
| --- | --- |
| ✅ db2look | To extract schema definition from DB2 |
| ✅ Oracle SQL Developer / DBMS\_METADATA | Extract Oracle schema |
| ✅ DataStage / Python / custom scripts | Row-level comparisons, checksum |
| ✅ GoldenGate Veridata (if available) | Automated cross-platform data comparison |

Would you like:

* A **sample SQL script** for checksum validation across DB2 and Oracle?
* A template for a **parallel row comparison report**?

Let me know and I’ll generate it!

When moving data from **DB2** to **Oracle**, several **issues** can arise due to fundamental differences in database architectures, data types, encoding, and constraints. Below is a categorized summary of common problems and how to address them.

**🔍 Common Issues When Migrating Data from DB2 to Oracle**

**1. 🧬 Data Type Mismatches**

| **DB2 Type** | **Oracle Equivalent** | **Issue/Note** |
| --- | --- | --- |
| BIGINT | NUMBER(19) | Risk of overflow if Oracle column is smaller |
| DECIMAL(p,s) | NUMBER(p,s) | Oracle may round or truncate if precision mismatches |
| CHAR(n) | CHAR(n) / VARCHAR2(n) | Fixed-length blanks in DB2 may be trimmed or miscompared |
| VARCHAR(n) | VARCHAR2(n) | Length semantics (bytes vs characters) can differ |
| DATE | DATE | Timezone differences and format differences |
| TIMESTAMP | TIMESTAMP(n) | Oracle supports up to nanoseconds; DB2 has microseconds |
| CLOB, BLOB | CLOB, BLOB | LOB streaming issues, size limits in Oracle |

✅ **Fix**: Map types explicitly, and test edge-case data (e.g. max decimals, longest strings).

**2. 🌍 Character Encoding Issues**

* **DB2 on z/OS** uses **EBCDIC** by default; Oracle expects **UTF-8 (AL32UTF8)**.
* Characters like é, ø, ’, or emojis may appear garbled if encoding is mismatched.
* Unicode to multibyte conversion issues may corrupt multi-language datasets.

✅ **Fix**: Convert data to UTF-8 before loading. Set Oracle client’s NLS\_LANG to match target DB charset.

**3. ⛔ NULL and Default Value Handling**

* Empty string ('') in DB2 ≠ NULL in Oracle.
* DB2 allows more permissive NULL values than Oracle.
* Oracle may reject rows with NULL into NOT NULL columns.

✅ **Fix**: Pre-transform values to meet Oracle constraints. Consider default value handling.

**4. 🔢 Numeric Precision and Overflow**

* Oracle silently **truncates** or **rounds** numeric fields if precision is exceeded.
* DB2 DECIMAL(31,15) must map to Oracle NUMBER(31,15) — Oracle allows up to 38 precision.

✅ **Fix**: Review precision/scale and widen Oracle target columns if needed.

**5. 📆 Date and Timestamp Format Differences**

* Oracle DATE includes time by default; DB2 DATE does not.
* DB2 timestamps default to 6-digit fractional seconds; Oracle must explicitly use TIMESTAMP(6).

✅ **Fix**: Convert date/timestamp fields using consistent format: YYYY-MM-DD HH24:MI:SS.FF6.

**6. ⚠️ Constraint and Key Differences**

* DB2 allows **duplicate NULLs** in unique keys; Oracle treats NULL = NULL in unique constraints.
* DB2 IDENTITY columns ≠ Oracle SEQUENCE; may break referential integrity.

✅ **Fix**:

* Load into staging tables first.
* Use Oracle sequences (SEQ.NEXTVAL) to simulate IDENTITY.
* Disable constraints during load, then re-validate.

**7. 🔄 LOB Data Handling**

* Oracle may have lower LOB limits or require special handling (e.g., DBMS\_LOB).
* Streaming large LOBs (e.g., > 32KB) may cause memory or network issues during replication or bulk load.

✅ **Fix**: Use Oracle SecureFiles and enable LOB streaming in tools like **Q Replication** or **SQL\*Loader**.

**8. 📦 Data Volume and Bulk Load Issues**

* DB2-to-Oracle migrations involving millions of rows require:
  + Parallel loads
  + Index disabling
  + Proper commit strategy

✅ **Fix**: Use tools like:

* Oracle **SQL\*Loader** or **External Tables**
* IBM **QREP**, **DataStage**, or **CDC**
* Apache **Sqoop** or **Spark** for big data volumes

**9. 🔒 Security and User Privileges**

* DB2 schema owners and Oracle schema owners may differ.
* Access control mechanisms (grants, roles) do not map 1:1.

✅ **Fix**: Define roles and privileges explicitly in Oracle post-load.

**10. 🧾 Index, Partitioning, and Performance Differences**

* DB2 indexes or partitions may not directly map to Oracle syntax.
* Query performance tuning required post-migration.

✅ **Fix**: Rebuild indexes and gather statistics (DBMS\_STATS.GATHER\_TABLE\_STATS).

**🧰 Tools for Migration and Validation**

| **Tool** | **Use Case** |
| --- | --- |
| **Oracle SQL\*Loader** | Bulk data load from CSV/flat files |
| **IBM Q Replication (QREP)** | Continuous replication |
| **Oracle GoldenGate** | Real-time replication alternative |
| **DataStage / Informatica** | ETL pipelines with transformations |
| **Python + cx\_Oracle + ibm\_db** | Custom migration scripts |