4.1. Introduction to Data Redaction

Data Redaction enables you to mask (redact) data that is returned from queries issued by applications.

You can redact column data by using one of the following methods:

**Full redaction.** You redact all of the contents of the column data. The redacted value returned to the querying application user depends on the data type of the column. For example, columns of the `NUMBER` data type are redacted with a zero (0), and character data types are redacted with a single space.

![Full Data Redaction](image)

Figure 4.1 Full Data Redaction

**Partial redaction.** You redact a portion of the column data. For example, you can redact a Social Security number with asterisks (*), except for the last 4 digits.
Regular expressions. You can use regular expressions to look for patterns of data to redact. For example, you can use regular expressions to redact email addresses, which can have varying character lengths. It is designed for use with character data only.
• **Random redaction.** The redacted data presented to the querying application user appears as randomly generated values each time it is displayed, depending on the data type of the column.

![Figure 4.4 OEM for Data Redaction](image)

Oracle Database applies the redaction at runtime, when users access the data (that is, at query-execution time). This solution works well in a production system. During the time that the data is being redacted, all of the data processing is performed normally, and the back-end referential integrity constraints are preserved [33].

No redaction. The No redaction type option enables you to test the internal operation of our redaction policies, with no effect on the results of queries against tables with policies defined on them [32]. You can use this option to test the redaction policy definitions before applying them to a production environment.

Oracle Database applies the redaction at runtime, when users access the data (that is, at query-execution time). This solution works well in a production system. During the time that the data is being redacted, all of the data processing is performed normally, and the back-end referential integrity constraints are preserved [33].

Data redaction can help you to comply with industry regulations such as Payment Card Industry Data Security Standard (PCI DSS) and the Sarbanes-Oxley Act.
4.2. When to use Data Redaction

Use Oracle Data Redaction when you must disguise sensitive data that our applications and application users must access. Data Redaction enables you to easily disguise the data using several different redaction styles.

Oracle Data Redaction is ideal for situations in which you must redact specific characters out of the result set of queries of Personally Identifiable Information (PII) returned to certain application users. For example, you may want to present a U.S. Social Security number that ends with the numbers 4320 as ***-**-4320.[34]

Oracle Data Redaction is particularly suited for call center applications and other applications that are read-only. Take care when using Oracle Data Redaction with applications that perform updates back to the database, because redacted data can be written back to this database.

4.3. Benefits of Using Data Redaction

Oracle Data Redaction provides several benefits when you use it to protect our data.

These benefits are as follows:

- You have different styles of redaction from which to choose.
- Because the data is redacted at runtime, Data Redaction is well suited to environments in which data is constantly changing.
- You can create the Data Redaction policies in one central location and easily manage them from there.
- The Data Redaction policies enable you to create a wide variety of function conditions based on SYS_CONTEXT values, which can be used at runtime to decide when the Data Redaction policies will apply to the results of the application user's query.
4.4. Target Use Cases for Data Redaction

Oracle Data Redaction protects sensitive data that is displayed in database applications. Oracle Data Redaction fulfils common use case scenarios.

4.4.1. Using Oracle Data Redaction with Database Applications

Data Redaction is transparent to application users because it preserves the original data type and (optionally) the formatting. It is highly transparent to the database because the data remains the same in buffers, caches, and storage only being changed at the last minute just before SQL query results are returned to the caller. The redaction is enforced consistently across all of the applications that use the same underlying database. You can specify which application users should see only redacted data by checking application user information that is passed into the database through the `SYS_CONTEXT` function; you can redact data based on attributes of the current database or application user; and you can implement multiple logical conditions within a given redaction policy.

Data Redaction is implemented in a way that minimizes performance overhead. These characteristics make Oracle Data Redaction particularly well suited for usage by a range of applications, analytics tools, reporting tools, and monitoring tools that share common production databases. Although its primary target is redaction of production data for applications, Oracle Data Redaction also can be used in combination with Oracle Enterprise Manager Data Masking and Subsetting Pack for protecting sensitive data in testing and development environments.
4.4.2. Oracle Data Redaction with Ad Hoc Database Queries

You may encounter situations where it is convenient to redact sensitive data for ad hoc queries that are performed by database users.

For example, in the course of supporting a production application, a user may need to run ad hoc database queries to troubleshoot and fix an urgent problem with the application [35].

This is different from the application-based scenarios described in "Using Oracle Data Redaction with Database Applications" [52], which typically generate a bounded set of SQL queries, use defined database accounts, and have fixed privileges.

Even though Oracle Data Redaction is not designed to prevent data exposure to database users who run ad hoc queries directly against the database, it can provide an additional layer to reduce the chances of accidental data exposure [36]. Because such users may have rights to change data, alter the database schema, and circumvent the SQL query interface entirely, it is possible for a malicious user to bypass Data Redaction policies in certain circumstances.

Remember that the Oracle Database security tools are designed to be used together to improve overall security. By deploying one or more of these tools as a complement to Oracle Data Redaction, you can securely increase our overall security posture [37].