Anonymous Data for Test Oracle HR Schemas

A Net 2000 Ltd. White Paper

Abstract

Data masking (or data scrubbing, data sanitization, data cleansing, data scrambling) is the process of making sensitive information in non-production databases safe for wider visibility. This White Paper is an overview of the issues and actions involved in cleansing the data in an Oracle HR schema.

An overview of the motivations for the creation of anonymous test Oracle HR schemas is provided and techniques which can be used in the cleansing process are listed. A comprehensive overview of the various issues which arise when sanitizing Oracle HR schemas is discussed. The remainder of the paper concerns operations that can be applied to specific tables in the HR schema in order to protect the information they contain from inappropriate visibility.

Implementation Details

This paper is a general discussion of the issues and techniques involved in the preparation of anonymous Oracle HR test instances. Net 2000 Ltd. sells a software tool called Data Masker which resolves the issues discussed herein and makes the masking of the information in Oracle HR schemas a simple and repeatable process. Pre-built rule sets, and evaluation copies of the software are available. Typical run times are between one and four hours for a complete masking operation.

Having said that, this paper really is a generic survey of the issues involved in sanitizing the data in an Oracle HR schema and there will be no further reference to any software. If you wish to know more, or have any questions about the issues and techniques described below please contact us.

Net 2000 Ltd.
Info@Net2000Ltd.com
http://www.Net2000Ltd.com

Table of Contents

Disclaimer

The contents of this document are for general information purposes only and are not intended to constitute professional advice of any description. The provision of this information does not create a business or professional services relationship. Net 2000 Ltd. makes no claim, representation, promise, undertaking or warranty regarding the accuracy, timeliness, completeness, suitability or fitness for any purpose, merchantability, up-to-dateness or any other aspect of the information contained in this paper, all of which is provided "as is" and "as available" without any warranty of any kind.

Oracle HR schemas vary widely and each has a unique configuration. Readers should take appropriate professional advice prior to performing any actions.

Masking Oracle HR Schemas

Why Provide Anonymous Information in Test Oracle HR Schemas?

The information in an HR schema is sensitive and nearly all countries place a legal obligation on the data holder which requires its protection. Besides the legal consequences, an escape of such information can cause considerable public relations damage.

So the information must be protected – of that there is no doubt. The issue then becomes a matter of implementation. In general, a policy of *minimum required access* is usually adopted.

In a production environment it is usually possible to protect the information by restricting access to the underlying data. Strict controls are in place and carefully designed user interfaces present a managed view. Test and development systems are different. They present an environment in which access is usually much wider. Information is visible to more people and those people often have greater privileges and low level access. From a data visibility standpoint, test schemas have exactly the same data security requirements as production systems yet they usually contain far more relaxed controls.

In general, a reasonable security assumption is that the more people who have access to the information, the greater the inherent risk of the data being compromised. If, as is typical with test systems, it is not possible to restrict the number of people working on the data then a useful technique to provide enhanced security is to modify the data so that no sensitive information remains. The process of modifying the data to remove data sensitivity issues is known by a number of names – data masking, data sanitization, data scrubbing or data cleansing.

Irregardless of the name used, the general technique is to modify the existing data in such a way as to remove all identifiable distinguishing characteristics thus rendering the data anonymous yet still usable as a test system.

In practise, masking the data in such a way that it remains functional requires a variety of techniques. These techniques are discussed in detail in the companion white paper available from Net 2000 Ltd. entitled Data Sanitization Techniques This paper will concentrate on the specific issues found in Oracle HR Schemas and the actions needed to sanitize the data in a selected group of tables. As the data in many of the tables is intricately interlinked and highly denormalized each operation applied to the schema must preserve the relationships. The methodology required is, in many cases, extremely subtle.

Sanitizing the Oracle HR Schema

Overview

There are a number of issues and techniques which arise when performing data cleansing operations on an Oracle HR schema. Often they appear in combination as well as singly. The sections below summarize each issue or technique. The names used are ours and were composed in order to provide a short descriptor for later reference. As far as we know there is no widely agreed nomenclature.

Techniques

Substitution. This technique consists of randomly replacing the contents of a column of data with information that looks similar but is completely unrelated to the real details. For example, employee surnames are replaced with surnames drawn from a random list. Substitution is very effective in terms of preserving the look and feel of the existing data. The downside is that a largish store of substitutable information must be maintained for each column. Substitution data can sometimes be very hard to find in large quantities. For example, if a million random street addresses are required, then just obtaining the substitution data can be a major exercise in itself.

Shuffling. A technique similar to substitution except that the substitution data is derived from the column itself. Essentially shuffling means the data in a column is randomly moved between rows until there is no longer any reasonable correlation with the remaining information in the row. There is a certain danger inherent to the shuffling technique – see the discussion below regarding the *Isolated Case* and *Meta Information* issues. Shuffling is rarely effective when used on small amounts of data. For example, if there are only 5 rows in a table it probably will not be too difficult to figure out which item of shuffled data really belongs to which row.

Number and Date Variance. The Variance technique is useful on numeric and date columns. Simply put, the algorithm involves modifying each value in a column by some random percentage of its real value. This technique has the nice advantage of providing a reasonable disguise for the exact details while still keeping the range and distribution of values in the column within viable limits. For example, a column of birth dates might have a random variance of 10% placed on it. Some values would be higher, some lower but all would be not too far from their original range.

Issues

Relevant Data. In test Oracle HR systems, most data will eventually appear on a front end screen in one form or another. To be useful, the sanitised values must resemble the look-and-feel of the original information. For example, surnames should be replaced with random surnames. Usually it is not acceptable to insert random collections of meaningless text.

Row-Internal Data Synchronization. In many cases the contents of one column in a row are related in some way to the contents of the other columns in the same row. For example, if the gender field in the row is `'F'` then the `FIRST_NAME` column should really have a female first name. Sometimes a column contains duplicate (denormalized) or aggregate information from the other columns in the row. In this case, the column should reflect the equivalent details after anonymization. For an example of this issue please see the discussion of the `PER_ALL_PEOPLE_F` tables `FULL_NAME` column.

Table-Internal Data Synchronization. This issue is quite common in Oracle HR systems. The rows in the table are massively denormalized and contain information that must be identical among many rows. For example, in the `PER_ALL_PEOPLE_F` table each `PERSON_ID` can have multiple rows (one for every assignment that person has ever had). Each row has a `LAST_NAME` field. When sanitizing data, every distinct `PERSON_ID` must be updated with the same anonymous `LAST_NAME`. If this is not done, an employees name will appear on the front end screens as having changed dozens of times.

Table-Table Data Synchronization. Quite often in Oracle HR schemas, important information which must be obfuscated is used as a join key to a column in one (or more) other tables. This means data masked in one table must have synchronized data changes in a number of others. For example, changes to the `EMPLOYEE_NUMBER` column in the `PER_ALL_PEOPLE_F` table must trigger identical changes in the matching `ASSIGNMENT_ID` of the `PER_ALL_ASSIGNMENTS_F` table.

Multi-Synchronization. It is possible for a column of a table to be constrained by multiple simultaneous synchronization issues. The `EMPLOYEE_NUMBER` of the `PER_ALL_PEOPLE_F` table, for example, has both *Table-Internal* and *Table-Table Synchronization* issues.

Intelligent Keys. It often happens that data items have a structure which represents an internal meaning. An example of this is the checksum on a credit card number. It is undesirable to sanitise such data by replacing it with a random collection of digits. The problem arises in the front end screens – they check the content prior to update. Hence if the data value is not right according to its internal design, any screen which displays the value will never permit an update because the validity checks fail. Intelligent Keys are commonly found in such things as employee numbers, credit card numbers and ID numbers. There are two ways to resolve this issue – either generate data items that meet the validity standard or shuffle the data in the column among the rows so that no row contains its original data but each data item is valid internally. It is a matter of judgement whether shuffling the column data among the rows provides sufficient sanitization for the data.

Free Format Data. Textual data such as letters, memos, disciplinary notes etc are practically impossible to sanitize in-situ. Unless the masking algorithms are extremely clever, or the format of the text is fixed it is probable that some information will be missed during the sanitization process. The usual way of dealing with free format data is to replace all values with randomly generated

meaningless text (or simply null them) and then update certain selected data items with carefully hand sanitized examples. This will give the users of the test system some realistic looking information to work on while preserving anonymity in the remainder.

Consistent Masking. A common requirement for an Oracle HR sanitization process is to ensure that the output is consistent across multiple runs. In practice this means that if the name of employee Joe Smith gets changed to Bill Jones then the next time the database is cloned and sanitized Joe Smith should again appear as Bill Jones (not Jim Williams). Training teams, in particular, tend to require this feature as they use a lot of pre-scripted examples.

Isolated Case Phenomena. The end result of the sanitization on the Oracle HR schema is to preserve the privacy of the individual records. In general, anonymity is derived from the presence of a large number of similar records. If a record stands out in any way it could be attributable to an individual. For example, could the record for the organizational CEO be determined by finding the largest salary in the table? Sometimes each record is its own special case. An unmasked birth date could readily be attributable to a specific individual. Whether this issue is important depends largely on how the remainder of the information is masked.

Meta Information. Sometimes even if information is not attributable to a specific individual, the collection of information might well be sensitive. For example, salary figures may be anonymous because the associated employee names have been masked, but does it matter if someone can add up the salary figures for a department? It's a judgement which has to be made considering the specific circumstances.

WHERE Clause Skips. When conducting masking operations be careful how the data to be sanitized is selected with `WHERE` clauses. It is easy, by making assumptions about the content of data in the row, to leave data in some rows in its original state. As an example, consider a table with a `FIRST_NAME` column and a `GENDER` column. Don't replace all the `FIRST_NAME` fields where `GENDER='M'` with male first names and the `FIRST_NAME` fields where `GENDER='F'` with female first names unless you are absolutely sure that the `GENDER` column can contain only `'M'` or `'F'`. It is entirely possible that the `GENDER` field may contain some other character (including null). Masking only the `'M'` and `'F'` `GENDER` fields will leave the `FIRST_NAME` field in some rows unmasked. It is far better to mask all rows with one option (Male First Names for example) and then go through a second time to mask every `FIRST_NAME` fields where `GENDER='F'` with female first names. This ensures that all rows have some sort of masking operation applied – irregardless of the state of the `GENDER` field. Where Clause Skips can lead to some quite insidious omissions – be sure to use full coverage to ensure every record gets masked.

Granularity. Is it necessary to sanitize absolutely everything? Or is masking enough data to prevent attribution sufficient. For example, do job titles have to be masked? Perhaps just removing a few examples of the *Isolated Case Phenomena* is sufficient. Either way, decisions have to be made as to the depth of cleansing

required. Any sanitization process trades thoroughness for complexity – the deeper operation the harder it is to maintain synchronization.

Distribution Preservation. In many cases the distribution of the data (numbers and dates) is important. For example, if salary figures are randomly updated, the random number generator will almost certainly give an even distribution between the specified ranges rather than the usual pyramid of lots of small salaries and fewer larger ones. Whether the skewing of the data matters is an implementation decision. If the data distribution is important then the *Variance* techniques previously discussed are the tool to use.

Sparse Data. Not all columns in a table have data in all rows. For example, the `PREVIOUS_LAST_NAME` field in the `PER_ALL_PEOPLE_F` table will be mostly empty. In this case, in order to preserve usability, it is not appropriate to fill in every `PREVIOUS_LAST_NAME` – the majority must remain null.

Percentage Operations. The Sparse Data issue above highlighted the point that not every column in a table might necessarily have data. The column sparseness can be preserved by masking the not null values. However, it might be desirable to assign the `PREVIOUS_LAST_NAME` values randomly thus removing that association from any specific row. In cases like this, it is useful to be able to null everything and then randomly update just a specified percentage of the rows with the appropriate contents.

Sequential Operations. The order in which masking operations are applied is sometimes very important. For example, in the *Row Internal Synchronization* issue discussed above, a `FULL_NAME` field was built from various other columns in the row (`LAST_NAME`, `FIRST_NAME` etc). It is essential that the build of the `FULL_NAME` field is applied after all of the masking rules for each component have successfully completed. In general, it cannot be assumed that a sequence of masking operations are separable. Not only does each masking operation have to complete - they sometimes have to complete in a specific order.

Special Cases. Most masking operations sweep with a broad brush and tend to obliterate special cases. For example, if there are only a few examples of an employee that has quit and then been rehired it may well prove to be the case that they received the same `PERSON_ID` but a different `EMPLOYEE_NUMBER` when hired the second time. The *Table-Internal Synchronization* operation may well remove this special case. Whether this homogenisation of the information is important is a decision for each implementation – however it is useful to realize that the issue exists.

Flex Fields. Oracle assumes quite rightly that every site will have custom requirements for the storage of information and implemented flex fields for this purpose. These fields store site specific information and their usage varies widely between implementations. An analysis of the flex field contents is required in order to ensure the data is completely scrubbed clean of personal details.

Speed. Some of the tables are big. One has to be careful how the masking operation is performed otherwise it will take an inordinate amount of time to complete. For

example, it is not possible to perform Inter-Table Data Synchronization directly on an un-indexed column in a large table. The process just never finishes. That is not to say it is impossible to do – one just has to use a carefully constructed intermediate table and a clever update statement. In Oracle HR schemas, make sure to disable all triggers before performing any updates or even operations on small tables will take forever to complete.

Repeatability. It is probable that any sanitization operation on an Oracle HR schema will need to be reproduced many times in the future as each new test instance is cloned. Make sure the masking process is designed to be simple and repeatable.

Oracle HR Tables Which Require Masking

Every organization has unique HR requirements. Hence there are considerable variations in the configuration of each Oracle HR implementation. In addition there are differences, both significant and minor, between the various versions of the Oracle HR schema. The tables discussed in this and the following section are current as of Oracle Applications version 11.5.8.

Some of the critical tables in an Oracle HR schema which contain user identifiable information are:

```
PER_ALL_PEOPLE_F
PER_ALL_ASSIGNMENTS_F
PER_PHONES
PER_ADDRESSES
PAY_EXTERNAL_ACCOUNTS
HR_COMMENTS
PER_ANALYSIS_CRITERIA
```

As a bare minimum, the data in the above tables will need to be sanitized. Usually other tables will also be required as well. The advice in the following section is only a suggestion as to a masking approach for some of the more tricky columns in the above tables. Doubtless there are other methodologies. Please be aware the discussion of the columns for the above tables is ***not*** complete! There are numerous important columns that have not been discussed for space reasons – for the most part their data sanitization requirements are pretty straightforward.

A careful analysis of each table and its contents will be required in order to completely mask the data. Typically, multiple operations are required for each table. The `PER_ALL_PEOPLE_F` table, for example, is a particularly complex case. Be sure to check the flex fields in the tables to see if they require masking. Since the usage of these columns are implementation defined, the only way to determine if they require sanitization is to look at them and find out what sort of data is in there.

Important Note
Needless to say, (but we are going to say it anyways), only mask rows in test instances. Even then, only mask schemas that you can recreate when necessary. In general, masking operations are not reversible - there is no “undo” other than a complete restore from backup.

The PER_ALL_PEOPLE_F Table

Overview

The PER_ALL_PEOPLE_F table holds personal information for just about everybody that comes into contact with the HR department. This includes employees, applicants, ex-employees, ex-applicants, contacts and usually other site specific categories of people.

In general, each distinct individual in the table is distinguished by an internal Oracle identifier called PERSON_ID which is unique to the individual. However, there is always more than one row per PERSON_ID in the table. The primary key is (PERSON_ID, EFFECTIVE_START_DATE, EFFECTIVE_END_DATE) and each time the user changes an assignment, a new row is added to PER_ALL_PEOPLE_F with a updated effective date range. The previous rows for the PERSON_ID remain as a history of previous assignments. This means that if the LAST_NAME field is masked for one row, the same last name will be required in each row with identical PERSON_ID. Getting this detail wrong really messes up some of the front end screens and reduces the functionality of the resulting test instance. This is an example of the *Table-Internal Synchronization* issue discussed previously, and it appears constantly in HR schema tables.

Decisions

Due to its many synchronization issues we tend to use PER_ALL_PEOPLE_F as the driver. That is, we modify PER_ALL_PEOPLE_F and synchronize every other table to it. This means some decisions have to be made regarding the PER_ALL_PEOPLE_F table.

Do you wish to preserve gender in the PER_ALL_PEOPLE_F *table?* In other words, should a record listed with a gender of 'M' necessarily remain an 'M' in the masked schema. It's a judgement call that has to be made based on the distribution of data within the schema. Typically, if there is a reasonable distribution of male and female records, one does not bother with masking the gender – the other fields will be enough to sanitize things. However, if there is only one or very few records of one type then the genders might need to be masked to prevent the identification via the *Isolated Case Phenomena*.

Do you wish to mask the EMPLOYEE_NUMBER *column?* Usually this is the case since people know their own employee number and would readily be able to identify their own records and possibly those of others. If all other columns associated with the record are masked, disguising the EMPLOYEE_NUMBER may not be required. If this column is masked, be aware that the EMPLOYEE_NUMBER column requires both *Table-Internal Data Synchronization* internally and *Table-Table Data Synchronization* with other tables (see the discussion of the PER_ASSIGNMENTS table for an example).

Do you need to worry about the TITLE *field?* Many sites use the Oracle supplied standard titles such as Mr. Mrs. Ms. etc and there is no real requirement to mask these fields. However, if you only have one "Reverend" in the system then you have a classic *Isolated Case Phenomena* to worry about. There are a number of sites (police

forces, military and other para-military) that have a variety of titles, the quantity of which gets progressively fewer as the rank increases. For example, there is probably no need to mask the title "Constable" but the single "Chief Constable" record might get re-titled back to an already existing type, or simply just deleted, to prevent identification.

Specific actions on PER_ALL_PEOPLE_F Columns

PERSON_ID – This column forms part of the primary key. It is an internal Oracle identifier and is generally meaningless. It can be masked, but there will be *Table-Table Data Synchronization* issues with over 60 tables. These tables will not be listed here, but if you wish to have this information just execute the query below:

```
select table_name from user_tab_columns
where column_name='PERSON_ID' order by table_name;
```

SEX – Decide if this column is to be masked. If it is, then typically every record gets set to 'M' then a defined percentage (50% perhaps) is updated to an 'F' value. Typically the PREVIOUS_LAST_NAME field needs to be synchronized so that the majority of not null PREVIOUS_LAST_NAME columns are associated with 'F' entries since this mirrors reality. It may also be necessary to update other tables containing details (such as pregnancy leave taken) to correlate with the re-gendered rows.

EMPLOYEE_NUMBER – Usually this column is rendered anonymous to prevent simple lookups on known values. This is a varchar2 field and its structure is site defined – any replacement value must conform to the same formatting otherwise the *Intelligent Key* issue will manifest itself and many things in the resulting schema will break. Care must also be taken not to update these values to collide with existing ranges otherwise unrelated rows can become associated with each other. The EMPLOYEE_NUMBER will need to be synchronized *Table-Internal* and also *Table-Table* with the PER_ALL_ASSIGNMENTS_F.ASSIGNMENT_NUMBER table at minimum and usually others depending on the implementation of the system.

FIRST_NAME – This sensitive column needs *Row-Internal* synchronization with the SEX column so that 'M' records get male first names and 'F' records get female first names. Also requires *Table-Internal* synchronization with the other rows with the same PERSON_ID.

MIDDLE_NAMES – Always masked and usually needs *Row-Internal* synchronization with the SEX column so that 'M' records get male names and 'F' records get female names. Also requires *Table-Internal* synchronization with the other rows with the same PERSON_ID.

LAST_NAME – This column requires *Table-Internal* synchronization with the other rows with the same PERSON_ID.

KNOWN_AS – A sparsely populated column that is always sanitized. A useful way of approaching this column is to null all values and substitute in a random percentage. Usually needs *Row-Internal* synchronization with the SEX column so that 'M' records

get male first names and 'F' records get female first names. May require *Table-Internal* synchronization with the other rows with the same PERSON_ID depending on how picky you wish to be.

PREVIOUS_LAST_NAME – Always masked, is sparsely populated and usually contains the previous last name of married (or divorced) females. A useful way of approaching this column is to null all values and substitute in a random percentage. May require *Table-Internal* synchronization with the other rows with the same PERSON_ID.

NATIONAL_IDENTIFIER – Usually a governmental ID data item which must be rendered anonymous (for example: SSN in the USA and NI number in the UK). This column will require *Table-Internal* synchronization with the other rows. Masking these types of ID usually involves coping with the *Intelligent Key* problem discussed previously.

EMAIL_ADDRESS – Needs to be cleansed in some manner. Probably should be updated with something random that looks like an email address.

DATE_OF_BIRTH – Often overlooked, but readily known and is a unique identifier in many cases. Probably should be masked, but take care not to put in values which are too old or too young. It is unlikely there are many 5 year olds on most HR systems and values which are out of range may well introduce validity issues on the front end screens.

START_DATE – A possible candidate. Make sure that this date is not sanitized to be unreasonable given the DATE_OF_BIRTH. It makes the internal HR validity checks unhappy if people are employed before they are born.

TITLE – May or may not be required to be masked – see the discussion in the Decisions section above. Masked TITLE columns may have to synchronized with the SEX field as appropriate and will require *Table-Internal* synchronization based on distinct PERSON_ID's.

FULL_NAME – A denormalized field requiring Row-Internal synchronization which contains the formatted contents of the LAST_NAME, FIRST_NAME, KNOWN_AS and TITLE fields. This data item must be built after the masking operations on the dependent fields are complete – see the previous discussion of the *Sequential Operations* issue.

The PER_ALL_ASSIGNMENTS_F Table

Overview

The PER_ALL_ASSIGNMENTS_F table holds information regarding employee assignments. Employees must have at least one employee assignment at all times in a period of service, and each assignment must have a unique number. The ASSIGNMENT_ID field here is the same as the EMPLOYEE_NUMBER in the PER_ALL_PEOPLE_F table and needs to be synchronized with it. There are multiple

assignments for any given PERSON_ID or EMPLOYEE_NUMBER so *Table-Internal Data Synchronization* is also called for.

Specific actions on PER_ALL_ASSIGNMENTS_F Columns

ASSIGNMENT_ID – If the EMPLOYEE_NUMBER column is masked in the PER_ALL_PEOPLE_F table then this value must also be masked. Each distinct PERSON_ID in the PER_ALL_PEOPLE_F table must have its EMPLOYEE_NUMBER match that of the ASSIGNMENT_ID for the same PERSON_ID in the PER_ALL_PEOPLE_F table.

The HR_COMMENTS Table

Overview

The HR_COMMENTS table contains a number of non-date dependent textual HR information.

Specific actions on HR_COMMENTS Columns

COMMENT_TEXT – Highly sensitive textual information associated with personnel records. Absolutely must be sanitized and exhibits the *Free Format Data* issue discussed previously. A typical action is to null the entire column or replace it with random textual gibberish. Carefully hand sanitized realistic looking data can be substituted into selected records if required.

The PER_PHONES Table

Overview

The PER_PHONES table holds phone numbers for current and ex-employees and other contacts.

Specific actions on PER_PHONES Columns

PHONE_NUMBER – Highly sensitive information which really must be replaced with realistic looking random data.

The PER_ADDRESSES Table

Overview

The PER_ADDRESSES table holds address information for current and ex-employees and other contacts. Be aware that there are some issues regarding the treatment of the primary address – see the Oracle documentation for detailed information.

Specific actions on PER_ADDRESSES Columns

ADDRESS_LINE_[1,2,3] – This is the employee address and is highly sensitive. Typically the first line is given a realistic looking random street address and the remainder set to null unless there is a requirement for multi-line street addresses.

TOWN_OR_CITY – It is much more useful to the end users if this column can be set from a list of town or city names rather than just random text.

REGION_[1,2,3] – These are usually information such as a state or county designator. As with the street address group of columns, the first value is given a realistic looking state name and the remainder set to null.

COUNTRY – It is debatable whether this column needs to be masked and is probably a decision that can be taken at implementation time. If this column is masked, be aware that the value used is required by the front end screens to be present in a pre-approved list. Usually this field is updated to a common value to eliminate all occurrences of the *Isolated Case Phenomena*.

POSTAL_CODE – This field needs to be masked and is usually an *Intelligent Key*. This means any replacement data must satisfy the validity checks or the front end screens will work improperly.

TELEPHONE_NUMBER_[1,2,3] – These are the telephone numbers of the employee and as such are highly sensitive. Typically the first line is given a realistic looking phone number and the rest are given null values.

The PAY_EXTERNAL_ACCOUNTS Table

Overview

The PAY_EXTERNAL_ACCOUNTS table stores bank account information for employees. All bank account details are stored in the flex fields so check these very carefully.

Specific actions on PAY_EXTERNAL_ACCOUNTS Columns

SEGMENT_[1…20] – Highly sensitive information which really must be replaced with realistic looking random data.

Summary

Given the legal and organizational operating environment of today, many test and development HR databases will require some form of sanitization in order to render the informational content anonymous.

There are a variety of techniques available, and an even larger number of issues of which to be aware. Some of the most critical issues are the *Row-Internal*, *Table-Internal* and *Table-Table Data Synchronization* requirements.

The demands of the Oracle HR schema require a sophisticated approach to the problem. In this paper the `PER_ALL_PEOPLE_F` table is the focus for the majority of the really complex masking requirements. A number of decisions for the PER_ALL_PEOPLE_F table were discussed and the outcome of these decisions has a major effect on the type of sanitization performed.

Some of the other important tables in the Oracle HR schema (from a data sanitization point of view) were listed and a discussion of how masking operations might be performed on selected columns from these tables was undertaken.

