|  |  |
| --- | --- |
| *Lombard Risk International Limited*  **LRM Arch – RBMS Standards** | |
|  | Document No.:  Version: v0.1  Date: 27 Oct 2014  Author: Malcolm Arnold  Owner:  Classification: Internal Use Only  Project: ARCH |
| |  | | --- | | © Lombard Risk International Limited 2013 -2020. All rights reserved.  This document and the information contained in it is confidential and the property of Lombard Risk Management plc. The technologies herein may be the subject of patents pending and granted or other similar protection. This document must not be used for commercial purposes other than your own internal discussions nor copied, disclosed, reproduced, stored in a retrieval system or transmitted in any form or by any means (electronic, - mechanical, photocopying, recording or otherwise) or otherwise used, whether in whole or in part except in accordance with the prior written agreement of Lombard Risk Management plc. Lombard Risk Management plc reserves the right to request the return of this document at any time and any copies thereto. Lombard Risk Management plc standard terms and conditions apply unless specific supply terms have been mutually agreed in writing. | | |

**Revision History**

| Version | Author(s) | Date | Description |
| --- | --- | --- | --- |
| 0.1 | Malcolm Arnold | 2014-10-27 | Initial revision. |
| 0.2 | Jack Yang | 2015-01-19 | Change mapping for long/Long, int/Integer, add a support data link. |
| 0.3 | Malcolm Arnold | 2015-02-18 | Move link to data type mapping to footnote, add Boolean check constraint as mandatory |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1 Introduction 4

1.1 Purpose 4

2 Naming Standards 4

# Introduction

## Purpose

This documents sets out simple naming standards for Lombard Risk software against RDBMS databases, ie. SQL databases.

The target audience is the technical authority group, software and technical architects and senior developers.

# Naming Standards

1. All object names must be in capitals
2. Table name must be singular,  eg.  PRIVILEGE\_GROUPS -> bad, PRIVILEGE\_GROUP -> good
3. All table and column names should have words delimited by \_ characters, eg. PRIVILEGEGROUPNAME should be PRIVILEGE\_GROUP\_NAME
4. Table names and columns should use American English spelling, eg.  ACTIVE\_LICENCES -> bad; ACTIVE\_LICENSES -> good.
5. Table names should have 3 letter prefix for functional area (keeps a degree of coherent grouping even in same schema)
6. All constraints should be named, eg. ALTER TABLE "USER\_PREFERENCE" **ADD CONSTRAINT "USER\_PREFERENCE\_PK"** PRIMARY KEY ("ID") ENABLE
7. Primary key constraint should be named as ${tableName}\_PK, removing characters from right of table name if length too long
8. Unique key constraints should be named as ${tableName}\_UKn, removing characters from right of table name if length too long
9. Foreign key constraints should be named as ${tableName}\_FKn, removing characters from right of table name if length too long
10. Check constraints should be named as ${tableName}\_CKn, removing characters from right of table name if length too long
11. All indexes should be name as ${tableName}\_IDXn, removing characters from right of table name if length too long
12. Create check constraints wherever possible, eg. ALTER TABLE LIC\_ACTIVE\_LICENSES ADD CONSTRAINT LIC\_ACTIVE\_LICENSES\_CK1 (IS\_ENABLED IN (0, 1))
13. Enforce all foreign key relationships in the database
14. Indexes on foreign key columns
15. Lookup tables and many-to-many join tables should be index organised on Oracle, eg. USER\_INFO\_PRIVILEGE\_GROUPS should be index organised
16. Columns that map to 'long' should be defined NUMBER(19,0) NOT NULL. (Oracle binary types not supported 10g)1
17. Columns that map to 'Long' should be defined NUMBER(19,0). (Oracle binary types not supported 10g)[[1]](#footnote-1)
18. Columns that map to 'int' should be defined NUMBER(10) NOT NULL. (Oracle binary types not supported 10g)1
19. Columns that map to 'Integer' should be defined NUMBER(10,0). (Oracle binary types not supported 10g)1
20. Columns that map to 'BigDecimal' for monetary data should be defined NUMBER(38,10)
21. Booleans should be represented as 1 for true and 0 for false and defined NUMBER(1,0), and must have check constraint restricting their value to IN (0, 1)

1. <https://docs.oracle.com/cd/E19501-01/819-3659/gcmaz/> [↑](#footnote-ref-1)