# Physical Data Design for Licensing Self-Certification Portal (LSCP)

## Introduction

This document provides a comprehensive description of the physical data structure and process design for the Licensing Self-Certification Portal (LSCP) project. It serves as a blueprint for the implementation of the LSCP database, ensuring a robust and efficient data management system.

The document delves into the detailed relationships between key business entities within the LSCP. A clear and concise diagrammatic representation is provided to visually illustrate the connections between these entities. This section also includes a comprehensive description of each entity, its attributes, and the data types used to represent them. The data relationships are further explained, highlighting the primary keys, foreign keys, and any constraints imposed on the data.

## Objectives

The objectives of the proposed Licensing Self-Certification Portal (LSCP) are:

1. Provide user-friendly and meaningful messages to users.
2. Store all electronic and paper submissions from applicant and authorized person (AP)/registered structural engineer (RSE) applications of the requisite safety certificates for registration of non-purpose built schools and child care centres, and applications in related to non-purpose non-local higher and professional education courses.
3. BD departmental portal login for internal users (BD), and provide User ID and password as an alternative.
4. Support the latest web browsers.
5. Comply with the standards of the Government System Architecture and government IT security policy.

## Database Analysis: bd

Last Updated: 2025/3/4 ??10:10:39

### Database Statistics

* Database Size: 88.10 MB
* Collections: 12
* Total Documents: 1278983
* Total Data Size: 371.24 MB

### Collections Overview

| Collection | Document Count | Size (MB) | N/A | ----------------- | -------------- | --------- | N/A |---|---|---| N/A | tasks | 5523 | 0.99 | N/A |---|---|---| N/A | eminutes | 133 | 0.03 | N/A |---|---|---| N/A | submissions | 0 | 0.00 | N/A |---|---|---| N/A | applications | 381 | 0.36 | N/A |---|---|---| N/A | notifications | 1837 | 0.24 | N/A |---|---|---| N/A | bsblocks | 98397 | 6.40 | N/A |---|---|---| N/A | cases | 451 | 1.17 | N/A |---|---|---| N/A | oauthtokens | 3019 | 2.29 | N/A |---|---|---| N/A | sysfilerefs | 601808 | 204.70 | N/A |---|---|---| N/A | attachments | 370 | 0.13 | N/A |---|---|---| N/A | users | 116 | 0.04 | N/A |---|---|---| N/A | adrblkfilerefs | 566948 | 154.89 | N/A |---|---|---|

## Physical Data Structure Specification

This section documents the data model and its associated descriptions of the required system.

### Physical Data Structure

An entity-relationship diagram consists of three basic elements such as entity, relationship, and attribute. Along with these are more components based on their main elements like weak entity, multi-valued attribute, and many more. Notations used to make ERD diagram examples include cardinality and ordinality to define relationships in numbers.

**Note:** The provided image for the ER diagram is missing, so a textual description of the relationships will be provided instead.

**Entities:**

* **ApplicationCases:** Stores application numbers.
* **SchoolApp\_Infos:** Stores the latest update of submitted application data.
* **SchoolApp\_Submissions:** Stores the submission of each application.
* **ApplicationFiles:** Stores the paths of uploaded files.
* **LSCPMasterTable:** Stores metadata or parameter data for the frontend.
* **GenOtp:** Stores generated OTP codes and usage status.
* **AdrBlk:** Stores addresses imported from BCIS.
* **SYS\_META\_DATA:** Stores metadata imported from BCIS.
* **Aprse:** Stores AP/RSE information imported from MWMS 2.0.
* **Application:** Stores application data (MongoDB).
* **Submission:** Stores submission data (MongoDB).
* **Attachment:** Stores attachment data (MongoDB).
* **BsBlock:** Stores block IDs and BDGIS data (MongoDB).
* **Task:** Stores task information (MongoDB).
* **Case:** Stores case information (MongoDB).
* **Eminute:** Stores e-minute data (MongoDB).
* **OAuthToken:** Stores OAuth tokens (MongoDB).
* **User:** Stores user data (MongoDB).
* **Notification:** Stores notification data (MongoDB).
* **SysFileRef:** Stores system file references (MongoDB).
* **AdrBlkFileRef:** Stores address block file references (MongoDB).

**Relationships:**

* SchoolApp\_Submissions has a one-to-one relationship with SchoolApp\_Infos through ApplicationNo.
* ApplicationFiles has a one-to-one relationship with SchoolApp\_Submissions through SubmissionId.
* GenOtp has a one-to-one relationship with SchoolApp\_Submissions through ApplicationNo and UserId.
* Application has a one-to-many relationship with Task.
* Application has a one-to-many relationship with Case.
* Case has a one-to-many relationship with Eminute.
* Application has a one-to-many relationship with Attachment.
* AdrBlkFileRef has a one-to-one relationship with AdrBlk through adrBlkId.
* AdrBlkFileRef has a one-to-one relationship with SysFileRef through sysFileRefId.

### (GCIS) Frontend - Application Forms Submission

(Diagram 3.1-1 is missing, so a textual description is provided)

This section describes the data structure for application form submissions from the frontend. Key entities include:

* **SchoolApp\_Infos:** Stores the latest update of the submitted application data as 1 row.
* **SchoolApp\_Submissions:** Stores the submission of each application.
* **ApplicationFiles:** Stores all the paths of applicant upload files.

### (GCIS) Frontend - OTP login control

(Diagram 3.1-2 is missing, so a textual description is provided)

This section describes the data structure for OTP login control from the frontend. Key entities include:

* **GenOtp:** Stores generated OTP code and the usage status.

### (BD) Backend - TBC

(Diagram 3.1-3 is missing, so a textual description is provided)

This section describes the data structure for the backend system. Key entities include:

* **ApplicationCases:** Stores all the application number.
* **SYS\_META\_DATA:** Stores meta data that import from BCIS.
* **Aprse:** Stores AP / RSE information that import from MWMS 2.0.

## Data Entity Description

This section states the conversion rules, the assumptions applied for the physical data design, the names of the physical data tables, the corresponding required system entities and key details to be stored into the database.

The database is a physical store of contract related information and textual data inside a database management system (DBMS). For LSCP, **Microsoft SQL Server 2019** is selected for the database management system. **All the spatial and textual entity will be stored into Microsoft SQL Server 2019.**

The following tables document how the Logical Data Model (LDM) can be mapped onto the physical data design.

**LSCP Frontend (SQL Server)**

| Table ID | LSCP Name | LSCP Entity Description | Key Nature | Key Data Item | N/A | :------- | :-------------------- | :-------------------------------------------------------------------------- | :--------- | :--------------------------- | N/A |---|---|---|---|---| N/A | T-S-01 | ApplicationCases | Table to store all the application number | PK | Id | N/A | | N/A | | N/A | ApplicationNo | N/A |---|---|---|---|---| N/A | T-S-02 | SchoolApp\_Infos | Table to store the latest update of the submitted application data as 1 row | PK | Id | N/A | | N/A | | N/A | ApplicationNo | N/A |---|---|---|---|---| N/A | T-S-03 | SchoolApp\_Submissions | Table to store the submission of each application | PK | Id | N/A | | N/A | | N/A | ApplicationNo | N/A | | N/A | | N/A | SubmissionId | N/A |---|---|---|---|---| N/A | T-S-04 | ApplicationFiles | Table to store all the path of applicant upload files | PK | Id | N/A | | N/A | | N/A | ApplicationNo | N/A | | N/A | | N/A | SubmissionId | N/A |---|---|---|---|---| N/A | T-S-05 | LSCPMasterTable | Table to store meta-data or parameter data for frontend | PK | Id | N/A | | N/A | | N/A | Code | N/A | | N/A | | N/A | Type + Code | N/A |---|---|---|---|---| N/A | T-S-06 | GenOtp | Table to store generated OTP code and the usage status | PK | Id | N/A | | N/A | | N/A | ApplicationNo + userId + Otp | N/A |---|---|---|---|---| N/A | T-S-07 | AdrBlk | Table to store addresses that import from BCIS | PK | ADR\_BLK\_ID | N/A |---|---|---|---|---| N/A | T-S-08 | SYS\_META\_DATA | Table to store meta data that import from BCIS | PK | SYS\_META\_DATA\_ID | N/A | | N/A | | N/A | REC\_TYPE | N/A | | N/A | | N/A | CODE | N/A |---|---|---|---|---| N/A | T-S-09 | Aprse | Table to store AP / RSE information that import from MWMS 2.0 | PK | Id | N/A | | N/A | | N/A | Name + RegistrationNumber | N/A |---|---|---|---|---| **LSCP Backend (MongoDB)**

| Table ID | LSCP Name | LSCP Entity Description | Key Nature | Key Data Item | N/A | :------- | :------------ | :-------------------------------------------------------------------------- | :--------- | :------------ | N/A |---|---|---|---|---| N/A | T-M-01 | Application | Stores application data | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-02 | Submission | Stores submission data | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-03 | Attachment | Stores attachment data | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-04 | BsBlock | Stores block IDs and BDGIS data | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-05 | Task | Stores task information | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-06 | Case | Stores case information | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-07 | Eminute | Stores e-minute data | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-08 | OAuthToken | Stores OAuth tokens | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-09 | User | Stores user data | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-10 | Notification | Stores notification data | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-11 | SysFileRef | Stores system file references | PK | \_id | N/A |---|---|---|---|---| N/A | T-M-12 | AdrBlkFileRef | Stores address block file references | PK | \_id | N/A |---|---|---|---|---| **Field Analysis for MongoDB Collections:**

Detailed field analysis for each MongoDB collection is provided in the database\_schema.md file. This analysis includes:

* Collection statistics (document count, size, average document size)
* Field analysis (types and occurrences) for each field in the collection.

This information is crucial for understanding the data structure and optimizing queries.

\*\*\* End of document\*\*\*