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

**Version 0.1**

**Jan 2025**

? The Government of the Hong Kong Special Administrative Region

## 1. Introduction

This document outlines the physical data design for the Licensing Self-Certification Portal (LSCP) project. It provides a blueprint for implementing the LSCP database, ensuring a robust and efficient data management system. This document details the physical data structure, data relationships, entity attributes, and data types used.

## 2. Objectives

The LSCP aims to:

1. Provide user-friendly and meaningful messages to users.
2. Store electronic and paper submissions from applicants, authorized persons (AP), and registered structural engineers (RSE) for safety certificates related to non-purpose-built schools, child care centers, and non-local higher and professional education courses.
3. Enable secure login for internal users (BD) via departmental portal or User ID and password.
4. Support modern web browsers.
5. Comply with Government System Architecture and IT security policies.

## 3. Physical Data Structure Specification

### 3.1. Physical Data Structure

The system uses a relational database model. The following Entity-Relationship Diagram (ERD) illustrates the relationships between key entities:

[Insert ER Diagram Here - Diagrams are not possible in plain text. A visual ERD would be beneficial here]

**Diagram Explanation:**

* **Entities:** Represented as rectangles.
* **Attributes:** Listed within the entity rectangles.
* **Relationships:** Shown as lines connecting entities, with cardinality notations (e.g., one-to-many, many-to-many).

### 3.1.1. (GCIS) Frontend - Application Forms Submission

This section describes the data structure for application form submissions originating from the frontend.

### 3.1.2. (GCIS) Frontend - OTP Login Control

This section describes the data structure used for managing OTP-based user authentication.

### 3.1.3. (BD) Backend - TBC

This section is reserved for future documentation of backend-specific data structures.

## 4. Data Entity Description

This section details the conversion rules, assumptions, physical table names, and key data elements stored in the database. Microsoft SQL Server 2019 is the selected DBMS.

**LSCP Frontend Tables:**

| Table ID | LSCP Name | LSCP Entity Description | Key Nature | Key Data Item |
| --- | --- | --- | --- | --- |
| T-S-01 | ApplicationCaseModel | Table to store all the application number | PK | Id |
|  |  |  |  | ApplicationNo |
| T-S-02 | SchoolAppInfoModel | Table to store the latest update of the submitted application data as 1 row | PK | Id |
|  |  |  |  | ApplicationNo |
| T-S-03 | SchoolAppSubmissionModel | Table to store the submission of each application | PK | Id |
|  |  |  |  | ApplicationNo |
|  |  |  |  | SubmissionId |
| T-S-04 | ApplicationFileModel | Table to store all the path of applicant upload files | PK | Id |
|  |  |  |  | ApplicationNo |
|  |  |  |  | SubmissionId |
| T-S-05 | ScsMasterTableModel | Table to store meta-data or parameter data for frontend | PK | Id |
|  |  |  |  | Code |
|  |  |  |  | Type + Code |
| T-S-06 | GenOtpModel | Table to store generated OTP code and the usage status | PK | Id |
|  |  |  |  | ApplicationNo + userId + Otp |
| T-S-07 | AdrBlkModel | Table to store addresses that import from BCIS | PK | ADR\_BLK\_ID |
| T-S-08 | Sys\_Meta\_Data | Table to store meta data that import from BCIS | PK | SYS\_META\_DATA\_ID |
|  |  |  |  | REC\_TYPE |
|  |  |  |  | CODE |
| T-S-09 | ApRseModel | Table to store AP / RSE information that import from MWMS 2.0 | PK | Id |
|  |  |  |  | Name + RegistrationNumber |

**LSCP Backend Tables:**

| Table ID | LSCP Name | LSCP Entity Description | Key Nature | Key Data Item |
| --- | --- | --- | --- | --- |
| T-S-01 | Application | Table to store all the application number | PK | \_id |
|  |  |  |  | ApplicationNo |
| T-S-02 | Submission | Table to store submissions related to an application | PK | \_id |
|  |  |  |  | ApplicationNo |
|  |  |  |  | SubmissionType |
| T-S-03 | Attachment | Table to store attachments related to applications and cases | PK | \_id |
|  |  |  |  | application |
|  |  |  |  | submissionCase |
| T-S-04 | BsBlock | Table to store BS Block information | PK | \_id |
|  |  |  |  | blockId |
| T-S-05 | Task | Table to store tasks associated with applications and cases | PK | \_id |
|  |  |  |  | application |
|  |  |  |  | submissionCase |
| T-S-06 | Case | Table to store case details for applications | PK | \_id |
|  |  |  |  | application |
| T-S-07 | Eminute | Table to store electronic minutes related to cases | PK | \_id |
|  |  |  |  | submissionCase |
| T-S-08 | OAuthToken | Table to store OAuth tokens for authentication | PK | \_id |
|  |  |  |  | accessToken |
| T-S-09 | User | Table to store user information | PK | \_id |
|  |  |  |  | osdpLoginId |
| T-S-10 | Notification | Table to store notifications for users | PK | \_id |
|  |  |  |  | user |
| T-S-11 | SysFileRef | Table to store system file references | PK | \_id |
|  |  |  |  | sysFileRefId |
| T-S-12 | AdrBlkFileRef | Table to store address block file references | PK | \_id |
|  |  |  |  | adrBlkFileRefId |

## Detailed Table Schemas

The following sections provide detailed schemas for key tables, including field names, data types, constraints, and descriptions.

### Collection: tasks

* **Document Count:** 5523
* **Size:** 0.99 MB
* **Average Document Size:** 0.18 KB

| Field | Data Type | Description physical\_data\_design.md

# Physical Data Design for Licensing Self-Certification Portal (LSCP)  
  
\*\*Version 0.1\*\*  
  
\*\*Jan 2025\*\*  
  
? The Government of the Hong Kong Special Administrative Region  
  
## 1. Introduction  
  
This document outlines the physical data design for the Licensing Self-Certification Portal (LSCP) project. It provides a blueprint for implementing the LSCP database, ensuring a robust and efficient data management system. This document details the physical data structure, data relationships, entity attributes, and data types used.  
  
## 2. Objectives  
  
The LSCP aims to:  
  
1. Provide user-friendly and meaningful messages to users.  
2. Store electronic and paper submissions from applicants, authorized persons (AP), and registered structural engineers (RSE) for safety certificates related to non-purpose-built schools, child care centers, and non-local higher and professional education courses.  
3. Enable secure login for internal users (BD) via departmental portal or User ID and password.  
4. Support modern web browsers.  
5. Comply with Government System Architecture and IT security policies.  
  
## 3. Physical Data Structure Specification  
  
### 3.1. Physical Data Structure  
  
The system uses a relational database model with Microsoft SQL Server 2019 as the selected DBMS. The following Entity-Relationship Diagram (ERD) illustrates the relationships between key entities:

[Insert ER Diagram Here - Diagrams are not possible in plain text. A visual ERD would be beneficial here] ```

**Diagram Explanation:**

* **Entities:** Represented as rectangles.
* **Attributes:** Listed within the entity rectangles.
* **Relationships:** Shown as lines connecting entities, with cardinality notations (e.g., one-to-many, many-to-many).

### 3.1.1. (GCIS) Frontend - Application Forms Submission

This section describes the data structure for application form submissions originating from the frontend. These tables are managed using Sequelize.

### 3.1.2. (GCIS) Frontend - OTP Login Control

This section describes the data structure used for managing OTP-based user authentication. These tables are managed using Sequelize.

### 3.1.3. (BD) Backend - Data Storage

The backend data is stored in MongoDB.

## 4. Data Entity Description

This section details the conversion rules, assumptions, physical table names, and key data elements stored in the database.

**LSCP Frontend Tables (Sequelize Models):**

| Table ID | LSCP Name | LSCP Entity Description | Key Nature | Key Data Item |
| --- | --- | --- | --- | --- |
| T-S-01 | ApplicationCaseModel | Table to store all the application number | PK | Id |
|  |  |  |  | ApplicationNo |
| T-S-02 | SchoolAppInfoModel | Table to store the latest update of the submitted application data as 1 row | PK | Id |
|  |  |  |  | ApplicationNo |
| T-S-03 | SchoolAppSubmissionModel | Table to store the submission of each application | PK | Id |
|  |  |  |  | ApplicationNo |
|  |  |  |  | SubmissionId |
| T-S-04 | ApplicationFileModel | Table to store all the path of applicant upload files | PK | Id |
|  |  |  |  | ApplicationNo |
|  |  |  |  | SubmissionId |
| T-S-05 | ScsMasterTableModel | Table to store meta-data or parameter data for frontend | PK | Id |
|  |  |  |  | Code |
|  |  |  |  | Type + Code |
| T-S-06 | GenOtpModel | Table to store generated OTP code and the usage status | PK | Id |
|  |  |  |  | ApplicationNo + userId + Otp |
| T-S-07 | AdrBlkModel | Table to store addresses that import from BCIS | PK | ADR\_BLK\_ID |
| T-S-08 | Sys\_Meta\_Data | Table to store meta data that import from BCIS | PK | SYS\_META\_DATA\_ID |
|  |  |  |  | REC\_TYPE |
|  |  |  |  | CODE |
| T-S-09 | ApRseModel | Table to store AP / RSE information that import from MWMS 2.0 | PK | Id |
|  |  |  |  | Name + RegistrationNumber |
| T-S-10 | BackendUpdateModel | Table to store backend update information (e.g., file uploads) | PK | Id |

**LSCP Backend Collections (MongoDB):**

| Table ID | LSCP Name | LSCP Entity Description | Key Nature | Key Data Item |
| --- | --- | --- | --- | --- |
| T-S-01 | Application | Table to store all the application number | PK | \_id |
|  |  |  |  | ApplicationNo |
| T-S-02 | Submission | Table to store submissions related to an application | PK | \_id |
|  |  |  |  | ApplicationNo |
|  |  |  |  | SubmissionType |
| T-S-03 | Attachment | Table to store attachments related to applications and cases | PK | \_id |
|  |  |  |  | application |
|  |  |  |  | submissionCase |
| T-S-04 | BsBlock | Table to store BS Block information | PK | \_id |
|  |  |  |  | blockId |
| T-S-05 | Task | Table to store tasks associated with applications and cases | PK | \_id |
|  |  |  |  | application |
|  |  |  |  | submissionCase |
| T-S-06 | Case | Table to store case details for applications | PK | \_id |
|  |  |  |  | application |
| T-S-07 | Eminute | Table to store electronic minutes related to cases | PK | \_id |
|  |  |  |  | submissionCase |
| T-S-08 | OAuthToken | Table to store OAuth tokens for authentication | PK | \_id |
|  |  |  |  | accessToken |
| T-S-09 | User | Table to store user information | PK | \_id |
|  |  |  |  | osdpLoginId |
| T-S-10 | Notification | Table to store notifications for users | PK | \_id |
|  |  |  |  | user |
| T-S-11 | SysFileRef | Table to store system file references | PK | \_id |
|  |  |  |  | sysFileRefId |
| T-S-12 | AdrBlkFileRef | Table to store address block file references | PK | \_id |
|  |  |  |  | adrBlkFileRefId |

### Data Types and Field Analysis

Detailed field analysis is available in the database\_schema.md file. This section provides a summary and highlights key considerations for data types and constraints.

* **ObjectIds:** Used extensively in MongoDB to reference related documents (e.g., application, submissionCase, user).
* **Strings:** Used for various text-based fields. Consider length constraints and validation rules (e.g., email format).
* **Dates:** Store date and time information. Ensure consistent time zone handling.
* **Numbers:** Used for numerical data, such as counts and sizes. Consider appropriate data types (e.g., Integer, BigInt) based on expected ranges.
* **Booleans:** Represent true/false values.
* **Objects and Arrays:** Used to store structured data and lists of related items.

## 5. Indexing Strategy

To optimize query performance, consider creating indexes on frequently queried fields, especially foreign keys and fields used in search queries. For example:

* ApplicationNo on the Application and SchoolAppSubmissionModel collections.
* sysFileRefId on the Attachment collection.
* blockId on the BsBlock collection.
* role and bdgis on the User collection.

## 6. Data Migration

Data migration scripts are used to import data from external sources (e.g., BCIS) into the LSCP database. These scripts are located in the scripts directory of the backend codebase.

* importBsBlock.js: Imports BS Block data.
* importFileRef.js: Imports System File Reference data.
* importAdrFileRef.js: Imports Address Block File Reference data.
* importUsers.js: Imports User data.
* migrateGroupAndDepartment.js: Migrates group and department data for users.

## 7. Code References

The following files in the codebase are relevant to the physical data design:

* **Models:** Defines the data structures and schemas for each entity.
  + bd-scs-backend-backend-main/models/Application.js
  + bd-scs-backend-backend-main/models/Attachment.js
  + bd-scs-backend-backend-main/models/BsBlock.js
  + bd-scs-backend-backend-main/models/Case.js
  + bd-scs-backend-backend-main/models/Eminute.js
  + bd-scs-backend-backend-main/models/Notification.js
  + bd-scs-backend-backend-main/models/OAuthToken.js
  + bd-scs-backend-backend-main/models/SysFileRef.js
  + bd-scs-backend-backend-main/models/Task.js
  + bd-scs-backend-backend-main/models/User.js
  + bd-scs-backend-backend-main/models/AdrBlkFileRef.js
  + bd-scs-nodejs-frontend-main/src/models/AdrBlk.js
  + bd-scs-nodejs-frontend-main/src/models/ApplicationCase.js
  + bd-scs-nodejs-frontend-main/src/models/ApplicationFile.js
  + bd-scs-nodejs-frontend-main/src/models/ApRse.js
  + bd-scs-nodejs-frontend-main/src/models/BackendUpdate.js
  + bd-scs-nodejs-frontend-main/src/models/GenOtp.js
  + bd-scs-nodejs-frontend-main/src/models/IamSmart.js
  + bd-scs-nodejs-frontend-main/src/models/LogEvents.js
  + bd-scs-nodejs-frontend-main/src/models/SchoolAppInfo.js
  + bd-scs-nodejs-frontend-main/src/models/SchoolAppSubmission.js
  + bd-scs-nodejs-frontend-main/src/models/ScsMasterTable.js
  + bd-scs-nodejs-frontend-main/src/models/Staff.js
  + bd-scs-nodejs-frontend-main/src/models/Sys\_Meta\_Data.js
  + bd-scs-nodejs-frontend-main/src/models/Test.js
* **Routes:** Defines the API endpoints for accessing and manipulating data.
  + bd-scs-backend-backend-main/routes/applications.js
  + bd-scs-backend-backend-main/routes/auth.js
  + bd-scs-backend-backend-main/routes/cases.js
  + bd-scs-backend-backend-main/routes/fileReferences.js
  + bd-scs-backend-backend-main/routes/tasks.js
  + bd-scs-backend-backend-main/routes/users.js
  + bd-scs-nodejs-frontend-main/src/routes/ApplicationController.js
  + bd-scs-nodejs-frontend-main/src/routes/AuthController.js
  + bd-scs-nodejs-frontend-main/src/routes/ESignController.js
* **Scripts:** Contains scripts for data migration and database setup.
  + bd-scs-backend-backend-main/scripts/importBsBlock.js
  + bd-scs-backend-backend-main/scripts/importFileRef.js
  + bd-scs-backend-backend-main/scripts/importAdrFileRef.js
  + bd-scs-backend-backend-main/scripts/importUsers.js
  + bd-scs-backend-backend-main/scripts/migrateGroupAndDepartment.js
  + bd-scs-backend-backend-main/scripts/setUpDb.js

## 8. Database Statistics (as of 2025/3/4)

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

## 9. Collection Statistics (MongoDB)

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

## 10. Field Analysis (MongoDB)

The following sections provide a detailed analysis of the fields within each MongoDB collection. This includes the data types used and the frequency of occurrence.

### Collection: tasks

* **Document Count:** 5523
* **Size:** 0.99 MB
* **Average Document Size:** 0.18 KB

| Field | Types | Occurrences |
| --- | --- | --- |
| \_\_v | objectId, int | 1000 |
| \_id | objectId | 1000 |
| application | objectId | 998 |
| createdAt | date | 1000 |
| status | string | 1000 |
| submissionCase | objectId | 998 |
| taskType | string | 998 |
| team | string | 835 |
| user | string, objectId | 713 |

### Collection: eminutes

* **Document Count:** 133
* **Size:** 0.03 MB
* **Average Document Size:** 0.24 KB

| Field | Types | Occurrences |
| --- | --- | --- |
| \_\_v | int | 133 |
| \_id | objectId | 133 |
| comment | string | 64 |
| content | string | 133 |
| createdAt | date | 133 |
| efolio | string | 100 |
| eminuteId | string | 133 |
| from | objectId, string | 133 |
| status | string | 133 |
| subject | string | 133 |
| submissionCase | objectId | 133 |
| sysFileRefId | string | 65 |
| to | objectId, string | 129 |

### Collection: submissions

* **Document Count:** 0
* **Size:** 0.00 MB
* **Average Document Size:** 0.00 KB

*(No field analysis available as the collection is empty)*

### Collection: applications

* **Document Count:** 381
* **Size:** 0.36 MB
* **Average Document Size:** 0.96 KB

| Field | Types | Occurrences |
| --- | --- | --- |
| APP13 | object, array | 172 |
| AddressOfPremiseCN | string | 267 |
| AddressOfPremiseCNFloor | string | 147 |
| AddressOfPremiseCNUnit | string | 147 |