Departmental Database

Part A: Relational Database Design

[Name]

[Student Id]

**Document Name**: DatabaseTemplatePartA.docx

Document History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Version | Revision | Description | Author | Role |
| 31/08/2022 | 1 | 1 | First version document |  | System Analyst |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Table of Contents**

[1. Scope and Boundaries 3](#_Toc110947602)

[1.1. Scope and Boundaries 3](#_Toc110947603)

[1.2. Business Rules 3](#_Toc110947604)

[2. Entity Relationship Diagrams 3](#_Toc110947605)

[3. Final Data Structure Diagram 3](#_Toc110947606)

[3.1. Data Structure Diagram (Detailed ERD) 3](#_Toc110947607)

[3.2. Data Types and Constraints 3](#_Toc110947608)

[4. Validations 3](#_Toc110947609)

[4.1. Domain, Entity and Referential Constraints 3](#_Toc110947610)

[4.2. Deletion / Update Rules 3](#_Toc110947611)

[4.3. Review Business Rules 3](#_Toc110947612)

[4.4. Resolve Issues 3](#_Toc110947613)

# Scope and Boundaries

## Scope and Boundaries

In Australia, life time learning is encouraged by the government. Holmesglen is one of the largest providers of vocational and higher education in Victoria.

Holmesglen provides mostly formal education (e.g., Cert IV, Diploma), and some informal short courses for all ages. The courses are provided in many ways, but they also have things in common. There are subjects, lessons, teachers, and students. In this article, I will design a database to keep track of subject, course, students, and teacher.

The boundary of the project is the relating department.

## Business Rules

* 1a. One department has one or more courses
* 1b. One course is offered by one department
* 2a. One course has one or more students
* 2b. One student takes one or more courses
* 3a. One course has one or more teachers
* 3b. One teacher teaches one or more courses
* 4a. One teacher teaches one or more students
* 4b. One student has one or more teachers

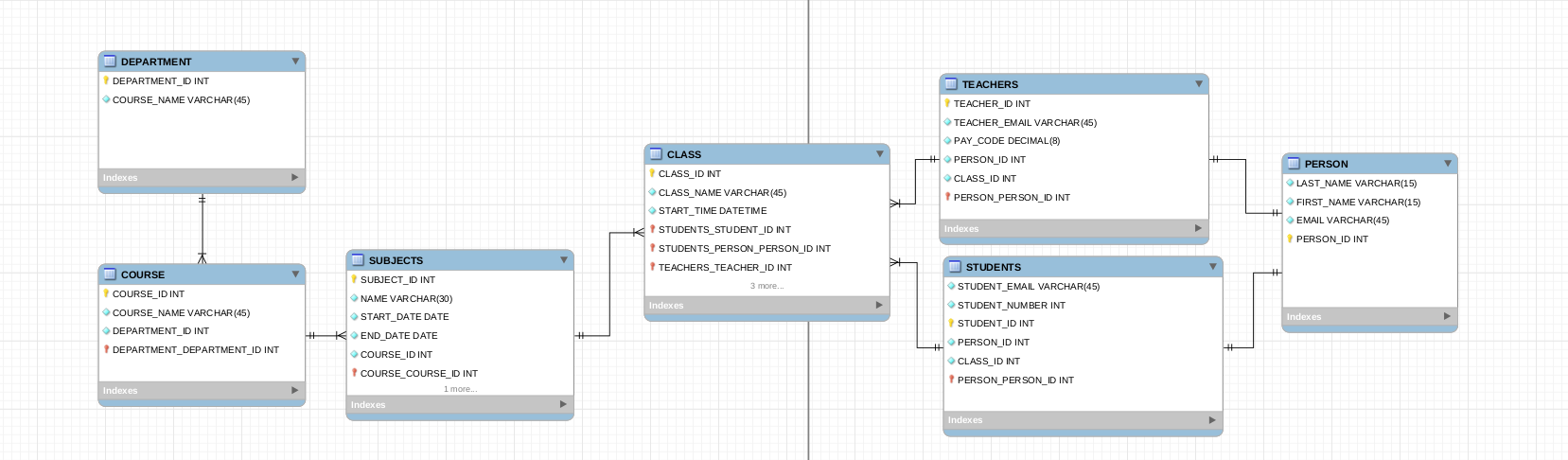
# Entity Relationship Diagrams

A picture containing text, white

Description automatically generated

# Final Data Structure Diagram

## Data Structure Diagram (Detailed ERD)



All fields are not null.

All primary key is INT and auto\_ increment.

## data Types and Constraints

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name *–*  (tblTable \_ PERSONS) | | | | | | |
| Attribute | PK/FK | Data Type | Length | Acceptable Values | Default | Null Values Allowed |
| PERSON\_ID | PK | INT |  |  |  | N |
| FIRST\_NAME |  | VARCHAR | 15 |  |  | N |
| LAST\_NAME |  | VARCHAR | 15 |  |  | N |
| EMIAL |  | VARCHAR | 45 |  |  | N |
| MOBIEL |  | INT |  |  |  | N |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name *–*  (tblTable \_ TEACHERS) | | | | | | |
| Attribute | PK/FK | Data Type | Length | Acceptable Values | Default | Null Values Allowed |
| TEACHER\_ID | PK | INT |  |  |  | N |
| TEACHER\_EMAIL |  | INT | 45 |  |  | N |
| PAY\_CODE |  | VANCHAR |  |  |  | N |
| PERSON\_ID | FK |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name *– tblTableName* (tblTable \_ COURSES) | | | | | | |
| Attribute | PK/FK | Data Type | Length | Acceptable Values | Default | Null Values Allowed |
| COURSE\_ID | PK | INT |  |  |  | N |
| COURSE\_NAME |  | VARCHAR | 30 |  |  | N |
| DEPARTMENT\_ID | FK |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name *–*  (tblTable STUDENTS) | | | | | | |
| Attribute | PK/FK | Data Type | Length | Acceptable Values | Default | Null Values Allowed |
| STUDENT\_ID | PK | INT |  |  |  | N |
| STUDENT\_EMAIL |  | VARCHAR | 45 |  |  | N |
| STUENDT\_NUMBER |  | INT |  |  |  | N |
| PERSON\_ID | FK |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name *–*  (tblTable \_ SUBJECTS) | | | | | | |
| Attribute | PK/FK | Data Type | Length | Acceptable Values | Default | Null Values Allowed |
| SUBJECT ID | PK | INT |  |  |  | N |
| SUBJECT\_NAME |  | VARCHAR | 30 |  |  | N |
| START\_DATE |  | DATE |  |  |  | N |
| END\_DATE |  | DATE |  |  |  | N |
| COURSE\_ID | FK | INT |  |  |  | N |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name *– tblTableName* (tblTable \_ CLASS) | | | | | | |
| Attribute | PK/FK | Data Type | Length | Acceptable Values | Default | Null Values Allowed |
| CLASS\_ID | PK | INT |  |  |  | N |
| TEACHER\_ID | PK/FK1 | INT |  |  |  | N |
| SUBJECT\_ID | PK/FK2 | INT |  |  |  | N |
| STUDENT\_ID | PK/FK3 | INT |  |  |  |  |
| START TIME |  | VARCHAR |  |  |  | N |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name *– tblTableName* (tblTable \_ DEPARTMENT) | | | | | | |
| Attribute | PK/FK | Data Type | Length | Acceptable Values | Default | Null Values Allowed |
| DEPARTMENT\_ID | PK | INT |  |  |  | N |
| DEPARTMENT\_NAME |  | VARCHAR | 30 |  |  | N |

# Validations

## Domain, Entity and Referential Constraints

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. SUBJECTS | | | | |
| COLUMN | CONSTRAINT | Data Type | Length | Null Values Allowed |
| SUBJECT\_ID | **PK** | int |  | N |
| NAME | NO NULL | varchar | 30 | N |
| START\_DATE | NO NULL | date |  | N |
| END\_DATE | NO NULL | date |  | N |
| TEACHER\_ID | FK | int |  | N |
| COURSE\_ID | FK | int |  | N |

Entity Constraints: Each record in a table must be uniquely identified by a primary key. The primary key can be a single column or be a combination of two of more columns. In above table the primary key is subject\_id, it is a single column.

|  |  |  |  |
| --- | --- | --- | --- |
| STUDENT\_SUBJECTS | | | |
| Attribute | PK/FK | Data Type | Null Values Allowed |
| STUDENT\_ID | **FK** | INT | N |
| SUBJECT\_ID | **FK** | INT | N |

Another table is subject\_students. There is a many-to-many relationship between subjects and students. One student can enroll many subjects, and one subjects can have many students. Between students and subjects table there is a junction table student\_subject table. The student\_subject table combines two foreign keys student\_id and subject\_id to form one COMPOSITE KEY. The COMPOSITE KEY is a PRIMARY KEY and can be used to uniquely identify one students on one subject.

**Domain Integrity:** Each field in the database must contain a specific data type (from a domain). In above table, every field such as subject\_id, name, start\_date, contain a specific data type. They are int, varchar, date respectively.

**Referential integrity:** It is relation between tables. Because each table in a database must have a primary key, this primary key can appear in other tables because of its relationship to data within those tables. When a primary key from one table appears in another table, it is called a foreign key. Foreign keys join tables and establish dependencies between table.

for example: in table PERSON

TEACHER\_ ID:FK is the primary of teacher\_id in table teachers

STUDENT\_ ID:FK is the primary of student\_id in table students

## Deletion / Update Rules

There are two update/deletion rules, one is NO ACTION, the other is CASCADE, NO ACTION is the same as RESTRICT in MySQL. If we want to change the teacher’s data, we have to change the teacher’s all data in the relating table.

CASCADE is different from RESTRICT. we can use the keyword CASCADE in SQL statements, within a single command, we can change all relating tables simultaneously.

In our case, we can use CASCADE in SQL while deleting/updating person, all data relating to the teacher in different tables will be delete/update simultaneously.

## Review Business Rules

I have reviewed my design, it meets below business requirements.

* + 1a. One department has one or more courses
  + 1b. One course is offered by one department
  + 2a. One course has one or more students
  + 2b. One student takes one or more courses
  + 3a. One course has one or more teachers
  + 3b. One teacher teaches one or more classes
  + 4a. One student can choose many classes
  + 4b. One class have many students
  + 5a. One subject has many classes
  + 5b. One class only teaches one subject
  + 6a. One person can be one teacher
  + 6b. A teacher must be a person
  + 7a. One person can be one student
  + 7b. One student must be one person
  + 8a. One course has more subjects
  + 8b. One subject belongs to only one course
  + 9a. One teacher teaches one or more students
  + 9b. One student has one or more teachers