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.

Good, this is clear. Boundary is well defined.

## Business Rules

* One teacher may teach different subjects, but one subject can only have one teacher.
* A subject can only belong to one course, but one course has many subjects.
* A student can enrol in many subjects and each subject can have many students.
* This department has many courses. One course can only belong to one department.

See below…

In your conceptual, original, conceptual design you have:

1 x one-to-many relationship

and a ternary relationship – which is

3 x many-to-many relationships through subjects

So, 4 relationships – you would have 8 business rules. One rule for each direction you read a relationship – see below for some examples…

# Entity Relationship Diagrams

A picture containing text, white

Description automatically generated

Above is great

In your original, conceptual design above you have:

1 x one-to-many relationship

and a ternary relationship – which is

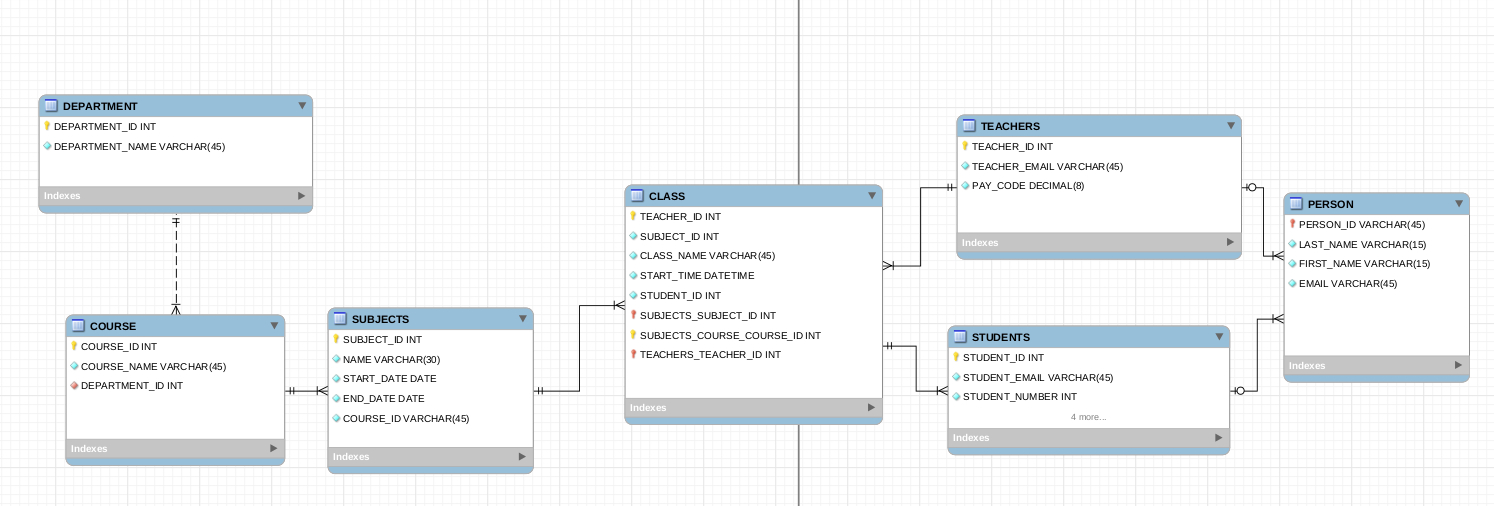
3 x many-to-many relationships through subjects

So, 4 relationships – you would have 8 business rules. One rule for each direction you read a relationship – see below for some examples…

* 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

# Final Data Structure Diagram

## Data Structure Diagram (Detailed ERD)



All fields are not null.

* good

See attached image with notes in red. There will be 10 relationships, so 20 business rules

* 5 x one-to-many
* 3 x many-to-many
* 2 x one-to-one

## 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/FK | int |  |  |  | N |
| TEACHER\_EMAIL |  | int | 45 |  |  | N |
| PAY\_CODE |  | decimal |  |  |  | N |
| PERSON\_ID |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table Name *–*  (tblTable \_ SUBJECTS) | | | | | | |
| Attribute | PK/FK | Data Type | Length | Acceptable Values | Default | Null Values Allowed |
| SUBJECT ID | PK | int |  |  |  | N |
| 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 | decimal |  |  |  | N |
| TEACHER\_ID | PK/FK1 | int |  |  |  | N |
| SUBJECT\_ID | PK/FK2 | varchar |  |  |  | N |
| START TIME |  | datetime |  |  |  | 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

All records are kept for 7 years. After 7 years:

* If the teacher resigns, the person table and teacher table’s relating record will be delated.
* If the student graduates, the person table and teacher table’s relating record will be delated.

Below records will be updated immediately.

* The teacher/student’s personal information is changed.
* The subject/ class’s detail is changed such as start/end time.
* The student/teacher transfers to another subject/class.
* Because the course is restructured, the relating subjects/classes table should be updated.

When you forward engineer your script, if you save it, you will see

ON DELETE NO ACTION

ON UPDATE NO ACTION

NO ACTION is the same as RESTRICT in MySQL, another option is CASCADE

These will be your rules

## Review Business Rules

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

* One teacher may teach different subjects, but one subject can only have one teacher.
* A subject can only belong to one course, but one course has many subjects.
* A student can enrol in many subjects and each subject can have many students.
* This department has many courses. One course can only belong to one department.

As above, you will have 10 relationships, therefore, 20 business rules.