DBI

ASSIGNMENT

REPORT

Student: Nguyen Huy Vu

Student’s Roll Number: HE161239

Class: SE1703

Subject: Introduction to database (DBI202)

Instructor: Le Phuong Chi

**Table of contents**

1. Background…………………………………………………………………………………………………………………...4
2. General……………………………………………………………………………………………………………….4
3. Specification: ……………………………………………………………………………………………………..4
4. Destination…………………………………………………………………………………………………………4
5. Tools…………………………………………………………………………………………………………………..4
6. Preview…………………………………………………………………………………………………………………..........4
7. Introduce The Coursera Learning Online………………………………………………………….…5
8. Information For Coursera ……………………………………………………………………………….….6
9. Finances………………………………………………………………………………………………………........6
10. Funding…………………………………………………………………………………………………………......6
11. Courses……………………………………………………………………………………………………………….6
12. Degrees………………………………………………………………………………………………………………7
13. Entity Relationship Diagram (Using Chen’s Notation)……………………………………………………..8
14. Entity Relationship Diagram (ERD) For Database…………………………………………………8
15. Simplified Entity Relationship Diagram……………………………………………………………….8
16. Full Entity Relationship Diagram For Database……………………………………………………9
17. Explanations For Entities……………………………………………………………………………….….10
18. Subject……………………………………………………………………………………………………….…….10
19. Class section………………………………………………………………………………………………….….11
20. Course program…………………………………………………………………………………………..……12
21. Student………………………………………………………………………………………………………..…..13
22. Teacher………………………………………………………………………………………………………..…..14
23. Attend……………………………………………………………………………………………………..……….15
24. Users…………………………………………………………………………………………………………………15
25. Logical Design………………………………………………………………………………………………………………16
26. Relational Schema……………………………………………………………………………………………16
27. Database Diagram…………………………………………………………………………………………….17
28. Table Analysis……………………………………………………………………………………………………17
29. Subject……………………………………………………………………………………………………………..17
30. Course program………………………………………………………………………………………………..18
31. Teacher…………………………………………………………………………………………………………….18
32. Class section……………………………………………………………………………………………………..18
33. Student…………………………………………………………………………………………………………….19
34. Attend……………………………………………………………………………………………………………..19
35. Users………………………………………………………………………………………………………………..20
36. Database Statements used to create table………………………………………………………20
37. Subject table…………………………………………………………………………………………………….20
38. Class section table…………………………………………………………………………………………….20
39. Course program table……………………………………………………………………………………….21
40. Student table…………………………………………………………………………………………………….21
41. Teacher table…………………………………………………………………………………………………….22
42. Attend table………………………………………………………………………………………………………22
43. Users table………………………………………………………………………………………………………..22
44. Queries, Store procedures and trigger………………………………………………………………………….23
45. Query using ORDER BY………………………………………………………………………………………23
46. Query using INNER JOIN…………………………………………………………………………………….23
47. Query using AGGREGATE function…………………………………………………………………….24
48. Query using GROUP BY and HAVING clause………………………………………………………25
49. Query using sub-query as relation…………………………………………………………………….26
50. Query using sub-query in the where clause……………………………………………………….27
51. Query using partial matching in the where clause……………………………………………..28
52. Query using self-join………………………………………………………………………………………….29
53. **Background**
54. **General**
55. Design and develop a database for online sharing learningresources
56. The database must consist of at least six tables that have been populated with data
57. The database is to support queries that would typically be submitted to the system for the topical area that you have chosen
58. Self-investigation for the requirement of the system
59. Using UML, Chen’s or Crow’s Foot notation or any case tool to create an Entity Relationship (ER) model for your relational database
60. Using appropriate SQL commands create a set of database tables in MS SQL Server 2008+. You should also show all constraints used in the creation of the tables

**II. Specification**

1. Your database must contain one view, one trigger, on store procedure and an index (describe why)
2. Create 10 sample queries that demonstrate the expressiveness of your database system. Your queries must demonstrate different aspects of the system.

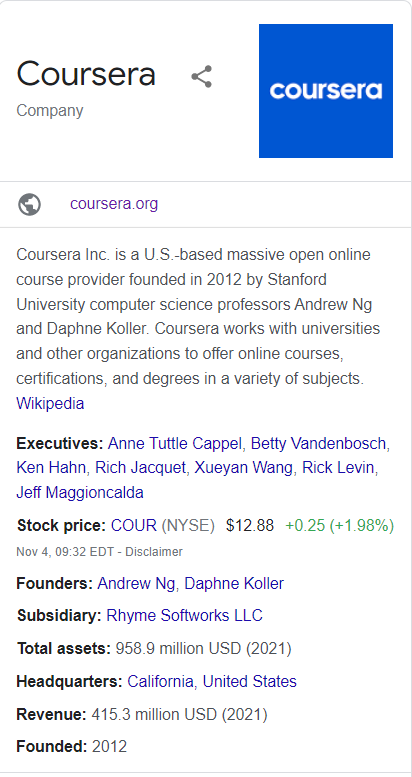
**III. Destination**

Design and develop a database for Coursera online sharing learning

IV. Tools

1. Microsoft SQL Database Management Studio
2. Microsoft word 2016
3. **Preview**
4. **Introduce The Coursera Learning Online**

Currently, if talking about the most popular online learning platform, coursera must be mentioned. Coursera guarantees the quality of its courses with certificates and degrees that are accepted by most of the world's famous universities. Online learning is becoming a popular trend because of its convenience and speed for both teachers and learners.

 Coursera Inc. is a large US-based open online course provider founded in 2012 by Stanford University computer science professors Andrew Ng and Daphne Koller. Coursera works with universities and other institutions to offer online courses, certificates, and degrees in a variety of subjects. By 2021, an estimated 150 universities will offer more than 4,000 courses through  
 Figure 1: Coursera Information

(Source: Wikipedia (<https://en.wikipedia.org/wiki/Coursera)>)

1. **Information for coursera**
2. **Finances**

Coursera's revenue has grown from $184 million in 2019 to $294 million in 2020. To date, Coursera has yet to make a profit. The company lost $66 million in 2020 as it ramped up marketing and advertising.

For the first quarter of 2021, Coursera reported revenue of $88.4 million, up 64% from a year earlier, with a net loss of $18.7 million, or $13.4 million. la on a non-GAAP basis. Coursera said consumer revenue was $51.9 million, up 61 percent, while corporate revenue was $24.5 million, up 63 percent, and degree programs had $12 million in revenue. dollars, up 81%.

For the third quarter of 2021, Coursera reported revenue of $109.9 million, up 33% from $82.7 million a year ago. Gross profit was $67.7 million, accounting for 61.6% of revenue. Net loss was $32.5 million or (29.5 percent of sales).

1. **Funding**

The startup has raised an initial $16 million funding round backed by Kleiner Perkins Caufield & Byers and the New Entrepreneurs Consortium. In 2013, GSV led in Series B investment, totaling $63 million. In 2015, NEA led the Series C venture capital round, totaling over $60 million. In 2017, the company raised $64 million from its existing investors in a Series D funding round. In 2019, the company raised $103 million in a Series E funding round. from SEEK Group, Future Fund and NEA. and updated its valuation to $2.5 billion.

1. **Courses**

Coursera courses last about four to twelve weeks, with one to two hours of video lectures per week. These courses offer quizzes, weekly assignments, graded and peer-reviewed assignments, an optional Honors assignment, and sometimes a project or final exam to complete. into a course. [46] Courses are also available upon request, in which case the user can take the time to complete the course with all the material available at once. As of May 2015, Coursera offers 104 on-demand courses. They also offer guided projects which are short 2-3 hour projects that can be done at home.

According to the 'Coursera Impact Report 2020', the five most popular courses people take from Coursera are contact tracking, digital marketing, project management, Python programming, and social psychology.

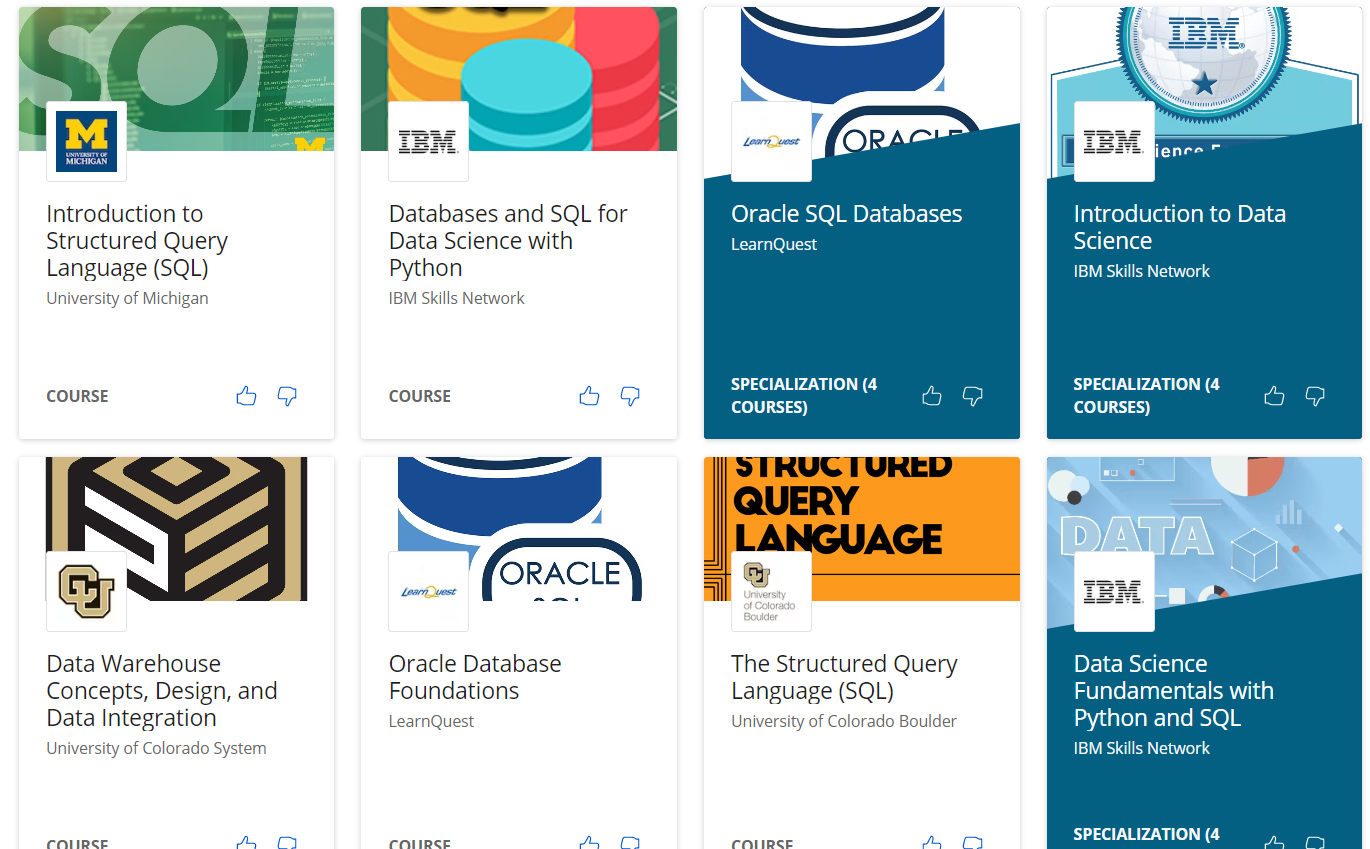


Figure 2: Coursera courses

(Source: Coursera (https://www.coursera.org))

1. **Degrees**

As of 2017, Coursera offers an entire master's degree. They started with a Master in Innovation and Entrepreneurship (OMIE) from HEC Paris and a Master of Accountancy (iMSA) from the University of Illinois but have moved on to offer a Master of Computer Science in Data Science and an MSc. MBA (iMBA), both from the University of Illinois. Also as part of their MBA programs, there are a number of courses offered separately that are included in the curriculum of certain MBAs when enrolled in classes such as the digital marketing course. their digital.

1. Entity Relationship Diagram (Using Chen’s Notation)
2. Entity Relationship Diagram (ERD) For Database
3. Simplified Entity Relationship Diagram

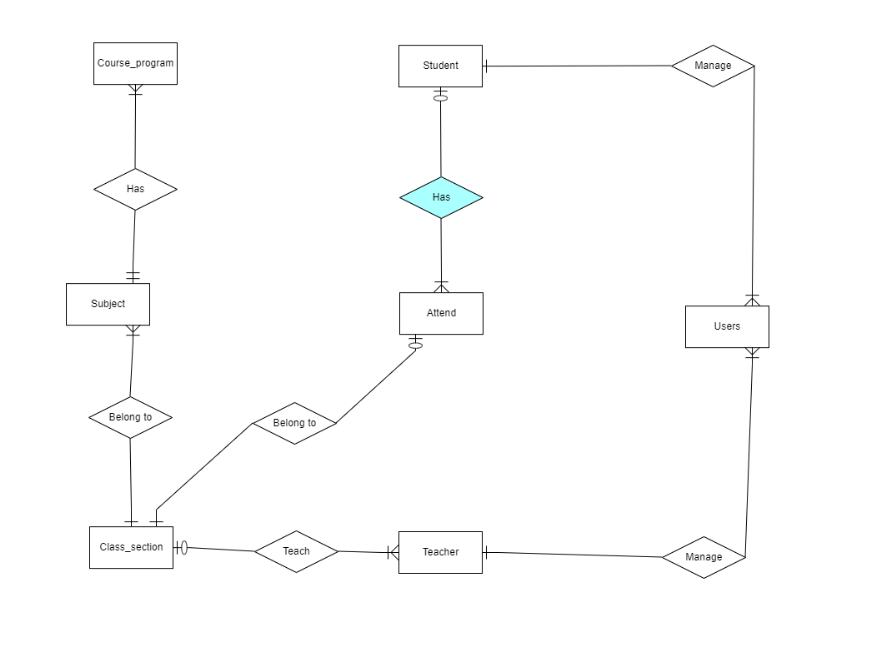


Figure 3: Simplified Entity Relationship Diagram

1. Full Entity Relationship Diagram For Database

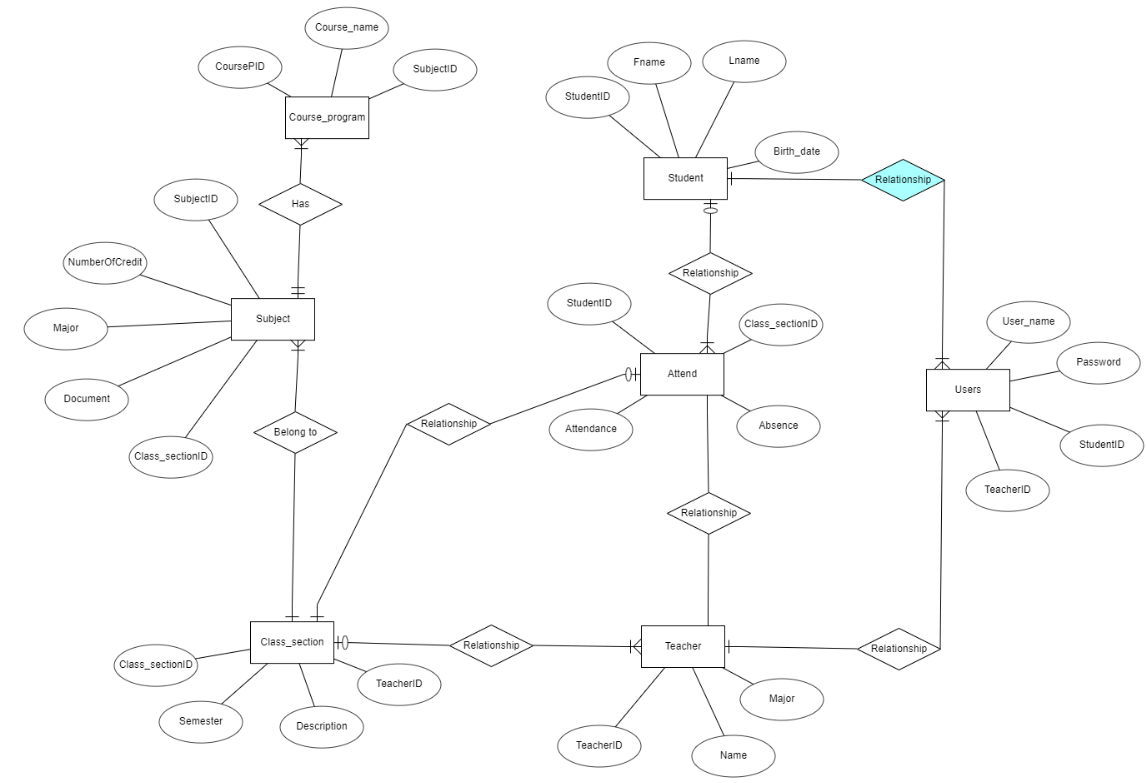


Figure 4: Full Entity Relationship Diagram For Database

1. Explanations For Entities
2. Subject

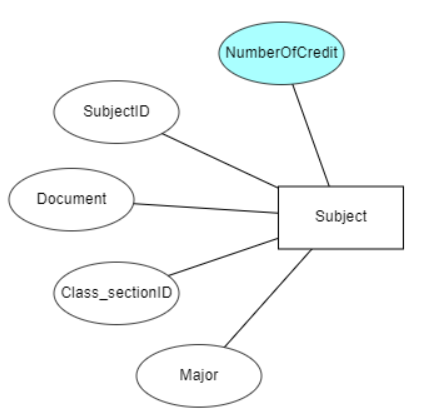


Figure 5: Subject entity

* Defination: Subjects often have many class sections. Divided by the number of slots per week. Includes many industry codes related to many topics such as: computer science, Data science, Business, Information technology, Arts and humanity,...
* Attributes of Subject: SubjectID, Document, NumberOfCredit, Class\_sectionID, Major
* SubjectID is unique (primary key of subject entity).
* Since students have to study a section of a subject, subject is a mandatory attribute.

1. Course Program

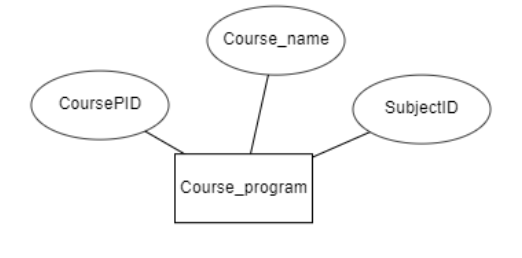
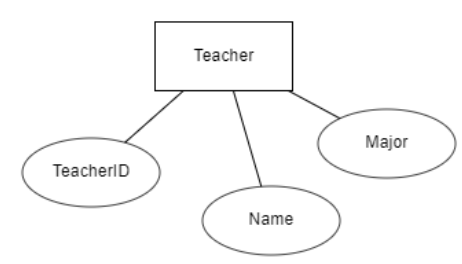


Figure 6: Course Program entity

* Defination: is a multi-subject management system, which stores brief information about each subject and each main idea
* Attributes of Course Program: CoursePID, Course\_name, SubjectID
* CoursePID is unique (primary key of Course Program entity).

1. Teacher

Figure 7: Teacher entity



* Defination: As a teacher who teaches for each subject and usually manages a part of students
* Attributes of Teacher: TeacherID, Name, Major
* TeacherID is unique (primary key of Teacher entity).

1. Class section

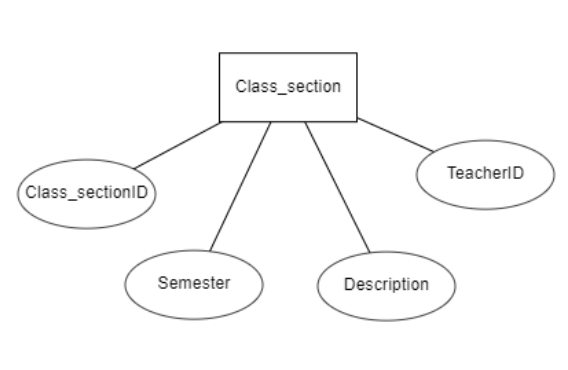


Figure 8: Class section entity

* Defination: is the module of each subject, where the number of semesters according to each teacher's id is recorded. And some description of the course.
* Attributes of Class section: Class\_sectionID, Semester, Description, TeacherID
* Class\_sectionID is unique (primary key of Class section entity).

1. Student

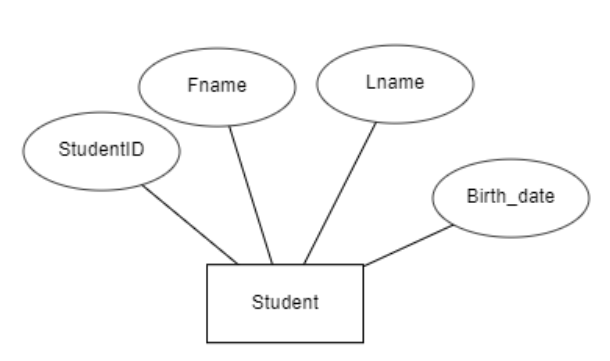


Figure 9: Student entity

* Defination: is the number of students divided by subject. It contains id, full name and date of birth.
* Attributes of Student section: StudentID, Fname, Lname, Birth\_date
* StudentID is unique (primary key of Class Student entity).

1. Attend

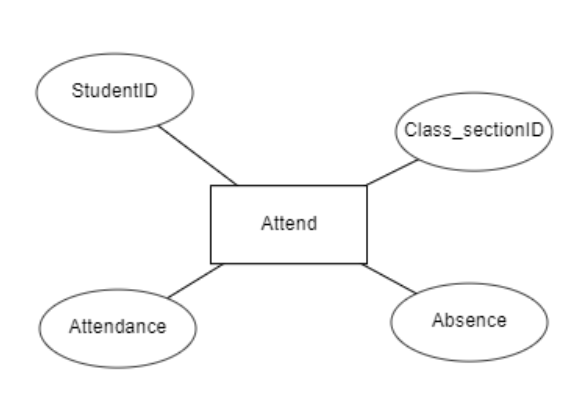


Figure 10: Attend entity

* Defination: to store information about the number of students' absences and attendance
* Attributes of Attend: Attendance, Absence, StudentID, Class\_sectionID
* Attendance is unique (primary key of Attend entity).

1. Users

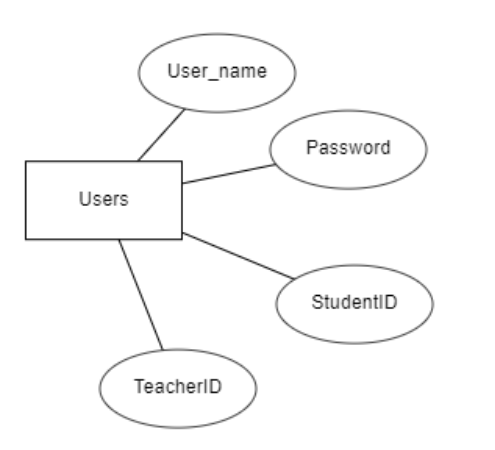


Figure 11: Users entity

* Defination: Is a system to manage the number of teachers and students in the form of ID. This is also the place to create passwords and accounts
* Attributes of Attend: User\_name, Password, StudentID, TeacherID
* User\_name is unique (primary key of Users entity).

1. Logical design
2. Relation schema

List of relations:

* Course\_program(**CoursePID**, Course\_name, SubjectID)
* Subject(**SubjectID**, Document, NumberOfCredit, Class\_sectionID, Major)
* Class\_section(**Class**\_**sectionID**, Semester, Description, TeacherID)
* Student(**StudentID**, Fname, Lname, Birth\_date)
* Attend(**Attendance**, Absence, StudentID, Class\_sectionID)
* Teacher(**TeacherID**, Name, Major)
* Users(**User\_name**, Password, StudentID, TeacherID)

1. Database diagram

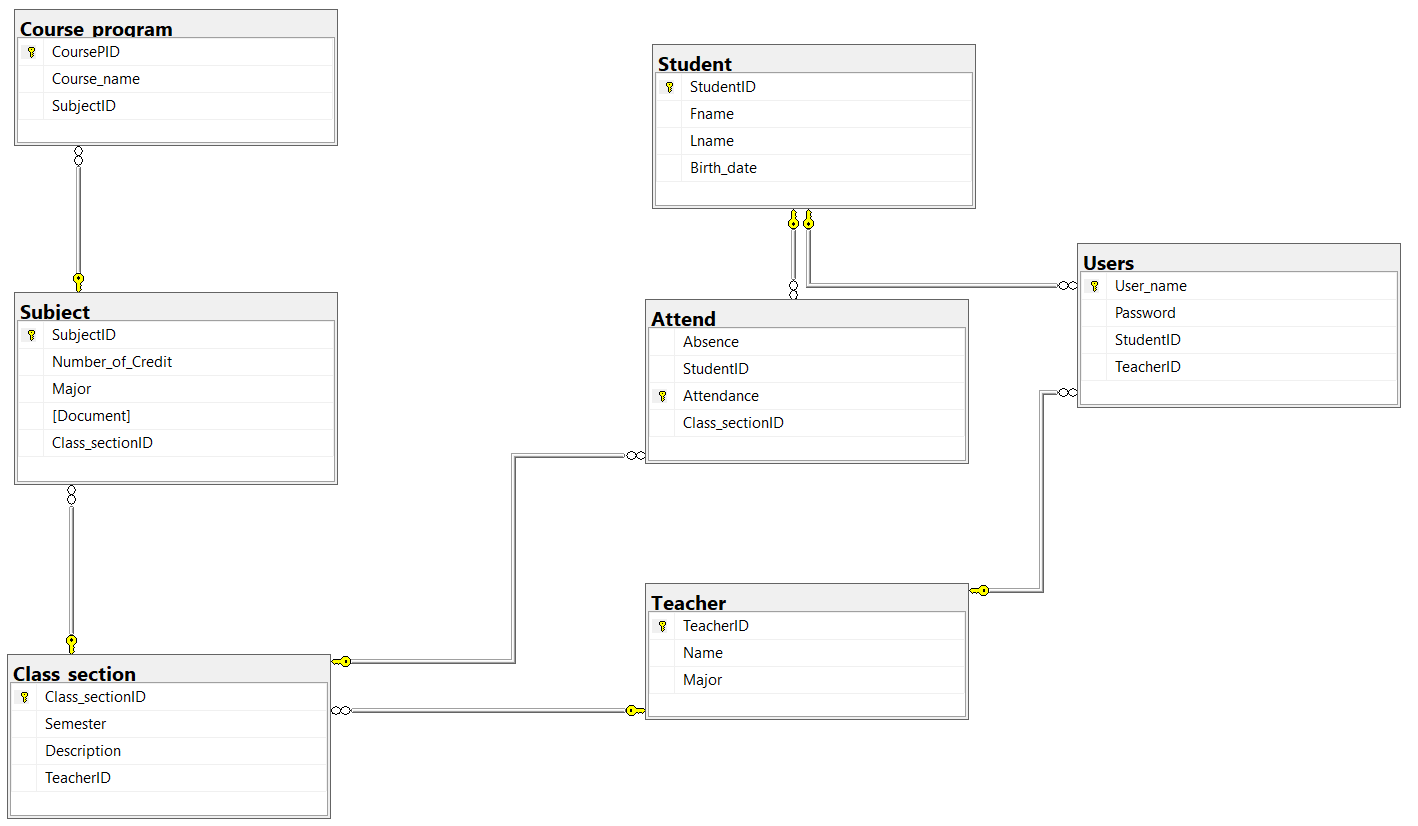


Figure 12: Logical design

1. Table analysis
2. Subject

|  |  |  |
| --- | --- | --- |
| Attributes | Data type | Allow null |
| SubjectID(Primary key) | int | No |
| Number\_of\_credit | Int | No |
| Major | Varchar(20) | No |
| Document | Varchar(20) | Yes |
| Class\_sectionID | int | No |

1. Course program

|  |  |  |
| --- | --- | --- |
| Attributes | Data type | Allow null |
| CoursePID(Primary key) | Int | No |
| Course\_name | Varchar(20) | No |
| SubjectID | int | No |

1. Teacher

|  |  |  |
| --- | --- | --- |
| Attributes | Data type | Allow null |
| TeacherID(Primary key) | Int | No |
| Name | varchar(20) | No |
| Major | varchar(20) | No |

1. Class section

|  |  |  |
| --- | --- | --- |
| Attributes | Data type | Allow null |
| Class\_sectionID(Primary key) | Int | No |
| Semester | Int | No |
| Description | Varchar(20) | No |
| TeacherID | Int | No |

1. Student

|  |  |  |
| --- | --- | --- |
| Attributes | Data type | Allow null |
| StudentID(Primary key) | Int | No |
| Fname | Varchar(20) | No |
| Lname | Varchar(20) | No |
| Birth\_date | Int | No |

1. Attend

|  |  |  |
| --- | --- | --- |
| Attributes | Data type | Allow null |
| Attendance (Primary key) | Int | No |
| Absence | Int | No |
| StudentID | Int | No |
| Class\_sectionID | Int | No |

1. Users

|  |  |  |
| --- | --- | --- |
| Attributes | Data type | Allow null |
| User\_name (Primary key) | varchar(20) | No |
| Password | varchar(20) | No |
| StudentID | Int | No |
| TeacherID | Int | No |

1. Database Statements used to create table
2. Subject table

CREATE TABLE Subject (

SubjectID int NOT NULL,

Number\_of\_Credit int NOT NULL,

Major varchar(20) NOT NULL,

Document varchar(20) NOT NULL,

Class\_sectionID int NOT NULL,

CONSTRAINT Subject\_pk PRIMARY KEY (SubjectID)

);

ALTER TABLE Subject ADD CONSTRAINT Subject\_Class\_section

FOREIGN KEY (Class\_sectionID)

REFERENCES Class\_section (Class\_sectionID);

1. Class section table

CREATE TABLE Class\_section (

Class\_sectionID int NOT NULL,

Semester int NOT NULL,

Description varchar(20) NOT NULL,

TeacherID int NOT NULL,

CONSTRAINT Class\_section\_pk PRIMARY KEY (Class\_sectionID)

);

ALTER TABLE Class\_section ADD CONSTRAINT Class\_section\_Teacher

FOREIGN KEY (TeacherID)

REFERENCES Teacher (TeacherID);

1. Course program table

CREATE TABLE Course\_program (

CoursePID int NOT NULL,

Course\_name varchar(20) NOT NULL,

SubjectID int NOT NULL,

CONSTRAINT Course\_program\_pk PRIMARY KEY (CoursePID)

);

ALTER TABLE Course\_program ADD CONSTRAINT Course\_program\_Subject

FOREIGN KEY (SubjectID)

REFERENCES Subject (SubjectID);

1. Student table

CREATE TABLE Student (

StudentID int NOT NULL,

Fname varchar(10) NOT NULL,

Lname varchar(10) NOT NULL,

Birth\_date date NOT NULL,

CONSTRAINT Student\_pk PRIMARY KEY (StudentID)

);

1. Teacher table

CREATE TABLE Teacher (

TeacherID int NOT NULL,

Name varchar(20) NOT NULL,

Major varchar(20) NOT NULL,

CONSTRAINT Teacher\_pk PRIMARY KEY (TeacherID)

);

1. Attend table

CREATE TABLE Attend (

Absence int NOT NULL,

StudentID int NOT NULL,

Attendance int NOT NULL,

Class\_sectionID int NOT NULL,

CONSTRAINT Attend\_pk PRIMARY KEY (Attendance)

);

ALTER TABLE Attend ADD CONSTRAINT Attend\_Class\_section

FOREIGN KEY (Class\_sectionID)

REFERENCES Class\_section (Class\_sectionID);

ALTER TABLE Attend ADD CONSTRAINT Attend\_Student

FOREIGN KEY (StudentID)

REFERENCES Student (StudentID);

1. Users table

CREATE TABLE "User" (

User\_name varchar(20) NOT NULL,

Password varchar(20) NOT NULL,

StudentID int NOT NULL,

TeacherID int NOT NULL,

CONSTRAINT User\_pk PRIMARY KEY (User\_name)

);

ALTER TABLE "User" ADD CONSTRAINT User\_Student

FOREIGN KEY (StudentID)

REFERENCES Student (StudentID);

ALTER TABLE "User" ADD CONSTRAINT User\_Teacher

FOREIGN KEY (TeacherID)

REFERENCES Teacher (TeacherID);

1. Queries, Store procedures and trigger
2. Query using ORDER BY

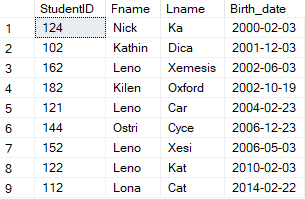
* Query:

select \* from Student s

order by YEAR(s.Birth\_date)

* Result:

Figure 13: Result of query using order by



1. Query using INNER JOIN

* Query:

select s.CoursePID, s.Course\_name, s.SubjectID,

st.Number\_of\_Credit, st.Document, cs.Semester

from Course\_program s

inner join Subject st on s.SubjectID = st.SubjectID

inner join Class\_section cs on cs.Class\_sectionID = st.Class\_sectionID

* Result:

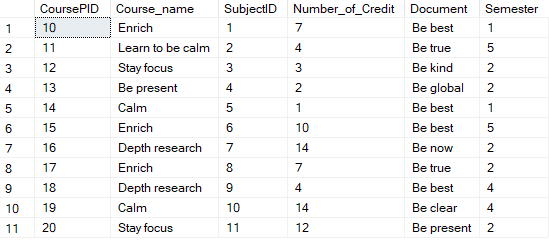


Figure 14: Result of query using inner join

1. Query using AGGREGATE function

* Query:

select count(st.Number\_of\_Credit) [Number of credit], st.Major, st.Document

from Subject st

group by st.Major, st.Document

* Result:

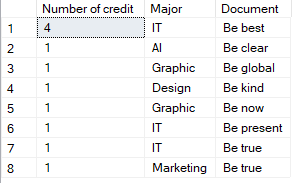


Figure 15: Result of query using AGGREGATE function

1. Query using GROUP BY and HAVING clause

* Query:

select Max(st.Number\_of\_Credit) [Maximum number of credit]

, st.Major, st.Document

from Subject st

group by st.Major, st.Document

Having Max(st.Number\_of\_Credit) > 3

* Result:

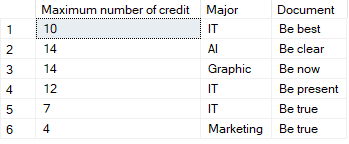


Figure 16: Result of query using GROUP BY and HAVING clause

1. Query using sub-query as relation

* Query:

select \*

from Subject s

where s.SubjectID in (select s.SubjectID

from Subject where s.Class\_sectionID < 10)

* Result:

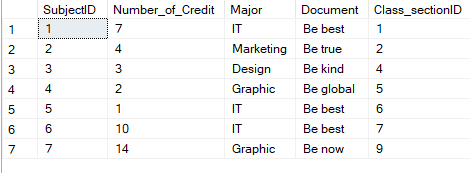


Figure 17: result of query using sub-query as relation

1. Query using sub-query in the where clause

* Query:

select \* from Teacher t

where t.TeacherID in

(select t.TeacherID from Teacher where t.Major = 'IT')

* Result

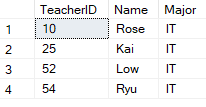


Figure 18: Result of query using sub-query in the where clause

1. Query using partial matching in the where clause

* Query:

select \* from Subject s

where s.Document like '%t%'

* Result:

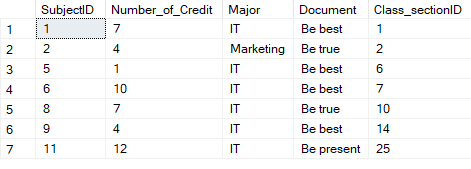


Figure 19: Result of query using partial matching in the where clause

1. Query using self-join

* Query:

select s1.SubjectID, s1.Document, s2.Document

from Subject s1

inner join Subject s2

on s1.SubjectID = s2.Class\_sectionID

and s1.Document > s2.Document

* Result:



Figure 20: Result of query using self-join

**Self-Assessment Sheet**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | % | NoAttemptto VeryPoor | Poor | Fair | Good | Very good | Excellent |
| ConceptualDatabaseDesign (EER) | 25 |  |  |  | ✓ |  |  |
| Relationalschema | 15 |  |  |  | ✓ |  |  |
| DatabaseImplementation | 12 |  |  |  | ✓ |  |  |
| Views andQueries | 15 |  |  |  |  | ✓ |  |
| Data Used | 5 |  |  |  |  | ✓ |  |
| DatabaseObjectsImplementation | 15 |  |  |  | ✓ |  |  |
| DemonstrationQuality | 15 |  |  |  |  |  |  |
| Total | 100 |  |  |  |  |  |  |