**Project Part 2**

***Question 1***

Normalize the following initial design to 3FN :

COURSE(CourseId, CourseName, Credit)

COURSE\_SECTION(Course-SectionNo, Course\_SectionName,(StudID, StudName, GRADE), CourseId, CourseName)

The initial design above, COURSE, is in 1FN.

The initial design above, COURSE\_SECTION, is not in 1FN because there is a repeating group of columns which are StudID, StudName and GRADE.

1. 1st Normal Form

These following rules must be respected for the table to be in 1FN :

* Each column of the table must be single-valued.
* Each column should have a unique name.
* A column should contain the same type of values.
* The order of the data stored does not matter.

COURSE

|  |  |  |
| --- | --- | --- |
| CourseId | CourseName | Credit |
| COMP251 | Algorithms and Data Structures | 3 |
| MATH240 | Discrete Structures | 3 |
| COMP308 | Computer Systems Lab | 1 |
| COMP535 | Computer Networks | 4 |

The primary key is CourseId because it is unique, not null and minimal.

COURSE\_SECTION1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Course-SectionNo | Course\_SectionName | StudID | StudName | GRADE | CourseId | CourseName |
| 001 | DS1 | 202001 | Chen | A | MATH240 | Discrete Structures |
| 002 | CN1 | 202002 | Tzu | A | COMP535 | Computer Networks |
| 005 | ADS1 | 202003 | Li | B | COMP251 | Algorithms and Data Structures |
| 004 | CSL1 | 202003 | Li | A | COMP308 | Computer Systems Lab |
| 003 | CN2 | 202004 | Gao | A | COMP535 | Computer Networks |

The primary key is a composite key of Course-SectionNo and StudID because they are in a many-to-many relationship : a course section (a course can have more than one section of the same type) can be opted by more than one student, and a student can opt for more than one course section.

Now we have the following design in 1FN :

COURSE\_SECTION1 (Course-SectionNo, Course\_SectionName, StudID, StudName, GRADE, CourseId, CourseName)

The design above, COURSE\_SECTION1, is not in 2FN because there are partial dependencies.

1. 2nd Normal Form

These following rules must be respected for the table to be in 2FN :

* The table should be in 1st Normal Form.
* The table should not have any partial dependencies.

By observing the table COURSE\_SECTION1, we notice four partial dependencies :

* The Course\_SectionName, CourseId and CourseName only depends on Course-SectionNo and not necessarily on StudID.
* The StudName only depends on StudID and not necessarily on Course-SectionNo.

GRADE is functionally dependent on the whole primary key (Course-SectionNo + StudID). It is impossible to find a specific GRADE record with Course-SectionNo only or with StudID only in this case, which means GRADE is not a partial dependency.

We are to divide the table COURSE\_SECTION1 in 3 new tables to remove the partial dependencies :

* CourseSection2 (Course-SectionNo, Course\_SectionName, CourseId, CourseName)
* Student2 (StudID, StudName)
* Grade2 (Course-SectionNo, StudID, GRADE)

|  |  |
| --- | --- |
| StudID | StudName |
| 202001 | Chen |
| 202002 | Tzu |
| 202003 | Li |
| 202004 | Gao |

Grade2 Student2

|  |  |  |
| --- | --- | --- |
| Course-SectionNo | StudID | GRADE |
| 001 | 202001 | A |
| 002 | 202002 | A |
| 005 | 202003 | B |
| 004 | 202003 | A |
| 003 | 202004 | A |

CourseSection2

|  |  |  |  |
| --- | --- | --- | --- |
| Course-SectionNo | Course\_SectionName | CourseId | CourseName |
| 001 | DS1 | MATH240 | Discrete Structures |
| 002 | CN1 | COMP535 | Computer Networks |
| 005 | ADS1 | COMP251 | Algorithms and Data Structures |
| 004 | CSL1 | COMP308 | Computer Systems Lab |
| 003 | CN2 | COMP535 | Computer Networks |

PK in CourseSection2 = Course-SectionNo

PK in Student2 = StudID

PK in Grade2 = Course-SectionNo + StudID (composite key)

FK1 in Grade2 = Course-SectionNo

FK2 in Grade2 = StudID

The following design is not in 3FN because there is a transitive dependency :

CourseSection2 (Course-SectionNo, Course\_SectionName, CourseId, CourseName)

Student2 (StudID, StudName)

Grade2 (Course-SectionNo, StudID, GRADE)

1. 3rd Normal Form

These following rules must be respected for the table to be in 3FN :

* The table should be in 2nd Normal Form.
* The table should not have any transitive dependencies.

By observing the table CourseSection2, we notice a transitive dependency :

The CourseName only depends on CourseId and not necessarily on the primary key of Course-SectionNo. CourseId is not part of the primary key in the table CourseSection2.

We are to divide the table CourseSection2 in two tables which results to this following design to remove the transitive dependency :

* CourseSection3 (Course-SectionNo, Course\_SectionName, CourseId)
* Student2 (StudID, StudName)
* Grade2 (Course-SectionNo, StudID, GRADE)
* COURSE (CourseId, CourseName, Credit)

Note : There should have been a table Course3 with the columns CourseId and CourseName (Course3 (CourseId, CourseName)) instead of COURSE (CourseId, CourseName, Credit). We notice there is a redundancy of attributes between the supposed Course3 (CourseId, CourseName) and COURSE (CourseId, CourseName, Credit). To remove this redundancy of attributes, we are prioritizing the entity that presents the most details (attributes). Therefore, Course3 (CourseId, CourseName) has been replaced by COURSE (CourseId, CourseName, Credit).

Grade2 Student2

|  |  |  |
| --- | --- | --- |
| Course-SectionNo | StudID | GRADE |
| 001 | 202001 | A |
| 002 | 202002 | A |
| 005 | 202003 | B |
| 004 | 202003 | A |
| 003 | 202004 | A |

|  |  |
| --- | --- |
| StudID | StudName |
| 202001 | Chen |
| 202002 | Tzu |
| 202003 | Li |
| 202004 | Gao |

CourseSection3 COURSE

|  |  |  |
| --- | --- | --- |
| Course-SectionNo | Course\_SectionName | CourseId |
| 001 | DS1 | MATH240 |
| 002 | CN1 | COMP535 |
| 005 | ADS1 | COMP251 |
| 004 | CSL1 | COMP308 |
| 003 | CN2 | COMP535 |

|  |  |  |
| --- | --- | --- |
| CourseId | CourseName | Credit |
| COMP251 | Algorithms and Data Structures | 3 |
| MATH240 | Discrete Structures | 3 |
| COMP308 | Computer Systems Lab | 1 |
| COMP535 | Computer Networks | 4 |

PK in COURSE = CourseId

PK in CourseSection3 = Course-SectionNo

FK in CourseSection3 = CourseId

PK in Student2 = StudID

PK in Grade2 = Course-SectionNo + StudID (composite key)

FK1 in Grade2 = Course-SectionNo

FK2 in Grade2 = StudID

***Question 2***

We need a database to store the following information :

Consultant info: Consultant ID, Name, DepartmentNumber, DepartmentName

Skill info: Skill ID, Description

Status of consultant’s skill info: Certified (Y/N)

Here are the Business`s Rules:

a/ One consultant can have many skills

b/ One skill can be obtained by many consultants

Part A: Design using the business rule a

Normalize your design to 3NF

Draw the E.R.D of the final design

Identify all keys of the final design

We have the following initial design :

Consultant(ConsultantID, Name, DepartmentNumber, DepartmentName, (Skill\_ID, Description, Certified))

The initial design above, Consultant, is not in 1FN because there is a repeating group of columns which are Skill\_ID, Description and Certified.

1. 1st Normal Form

These following rules must be respected for the table to be in 1FN :

* Each column of the table must be single-valued.
* Each column should have a unique name.
* A column should contain the same type of values.
* The order of the data stored does not matter.

Consultant1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ConsultantID | Name | DepartmentNumber | DepartmentName | Skill\_ID | Description | Certified |
| 1 | Chan | 20201 | Engineering | 101 | Planning | N |
| 2 | Zhang | 20202 | Financial | 101 | Planning | Y |
| 3 | Lai | 20203 | Healthcare | 303 | Analyzing Management Practices | Y |
| 4 | Luo | 20204 | Human Resources | 404 | Implementing solutions | Y |
| 5 | Zhen | 20201 | Engineering | 505 | Designing | Y |

The primary key is a composite key of ConsultantID and Skill\_ID because they are in a many-to-many relationship : a consultant can opt for more than one skill, and a skill can be opted by more than one consultant.

Now we have the following design in 1FN :

Consultant1 (ConsultantID, Name, DepartmentNumber, DepartmentName, Skill\_ID, Description, Certified)

The design above, Consultant1, is not in 2FN because there are partial dependencies.

1. 2nd Normal Form

These following rules must be respected for the table to be in 2FN :

* The table should be in 1st Normal Form.
* The table should not have any partial dependencies.

By observing the table Consultant1, we notice four partial dependencies :

* The Name, DepartmentNumber and DepartmentName only depends on ConsultantID and not necessarily on Skill\_ID.
* The Description only depends on Skill\_ID and not necessarily on ConsultantID.

Certified is functionally dependent on the whole primary key (ConsultantID + Skill\_ID). It is impossible to find a specific Certified record with ConsultantID only or with Skill\_ID only in this case, which means Certified is not a partial dependency.

We are to divide the table Consultant1 in 3 new tables to remove the partial dependencies :

* Consultant2 (ConsultantID, Name, DepartmentNumber, DepartmentName)
* Skill2 (Skill\_ID, Description)
* Certification2 (ConsultantID, Skill\_ID, Certified)

Certification2 Skill2

|  |  |  |
| --- | --- | --- |
| ConsultantID | Skill\_ID | Certified |
| 1 | 101 | N |
| 2 | 101 | Y |
| 3 | 303 | Y |
| 4 | 404 | Y |
| 5 | 505 | Y |

|  |  |
| --- | --- |
| Skill\_ID | Description |
| 101 | Planning |
| 303 | Analyzing Management Practices |
| 404 | Implementing solutions |
| 505 | Designing |

Consultant2

|  |  |  |  |
| --- | --- | --- | --- |
| ConsultantID | Name | DepartmentNumber | DepartmentName |
| 1 | Chan | 20201 | Engineering |
| 2 | Zhang | 20202 | Financial |
| 3 | Lai | 20203 | Healthcare |
| 4 | Luo | 20204 | Human Resources |
| 5 | Zhen | 20201 | Engineering |

PK in Consultant2 = ConsultantID

PK in Skill2 = Skill\_ID

PK in Certification2 = ConsultantID + Skill\_ID (composite key)

FK1 in Certification2 = ConsultantID

FK2 in Certification2 = Skill\_ID

The following design is not in 3FN because there is a transitive dependency :

Consultant2 (ConsultantID, Name, DepartmentNumber, DepartmentName)

Skill2 (Skill\_ID, Description)

Certification2 (ConsultantID, Skill\_ID, Certified)

1. 3rd Normal Form

These following rules must be respected for the table to be in 3FN :

* The table should be in 2nd Normal Form.
* The table should not have any transitive dependencies.

By observing the table Consultant2, we notice a transitive dependency :

The DepartmentName only depends on DepartmentNumber and not necessarily on the primary key of ConsultantID. DepartmentNumber is not part of the primary key in the table Consultant2.

We are to divide the table Consultant2 in two tables which results to this following design to remove the transitive dependency :

* Consultant3 (ConsultantID, Name, DepartmentNumber)
* Skill2 (Skill\_ID, Description)
* Certification2 (ConsultantID, Skill\_ID, Certified)
* Department3 (DepartmentNumber, DepartmentName)

Certification2 Skill2

|  |  |  |
| --- | --- | --- |
| ConsultantID | Skill\_ID | Certified |
| 1 | 101 | N |
| 2 | 101 | Y |
| 3 | 303 | Y |
| 4 | 404 | Y |
| 5 | 505 | Y |

|  |  |
| --- | --- |
| Skill\_ID | Description |
| 101 | Planning |
| 303 | Analyzing Management Practices |
| 404 | Implementing solutions |
| 505 | Designing |

Consultant3 Department3

|  |  |  |
| --- | --- | --- |
| ConsultantID | Name | DepartmentNumber |
| 1 | Chan | 20201 |
| 2 | Zhang | 20202 |
| 3 | Lai | 20203 |
| 4 | Luo | 20204 |
| 5 | Zhen | 20201 |

|  |  |
| --- | --- |
| DepartmentNumber | DepartmentName |
| 20201 | Engineering |
| 20202 | Financial |
| 20203 | Healthcare |
| 20204 | Human Resources |

PK in Department3 = DepartmentNumber

PK in Consultant3 = ConsultantID

FK in Consultant3 = DepartmentNumber

PK in Skill2 = Skill\_ID

PK in Certification2 = ConsultantID + Skill\_ID (composite key)

FK1 in Certification2 = ConsultantID

FK2 in Certification2 = Skill\_ID

Part B: Design using the business rule b

Normalize your design to 3NF

Draw the E.R.D of the final design

Identify all keys of the final design

We have the following initial design :

Skill(Skill\_ID, Description, Certified, (ConsultantID, Name), DepartmentNumber, DepartmentName)

The initial design above, Skill, is not in 1FN because there is a repeating group of columns which are ConsultantID and Name.

1. 1st Normal Form

These following rules must be respected for the table to be in 1FN :

* Each column of the table must be single-valued.
* Each column should have a unique name.
* A column should contain the same type of values.
* The order of the data stored does not matter.

Skill1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Skill\_ID | Description | Certified | ConsultantID | Name | DepartmentNumber | DepartmentName |
| 101 | Planning | N | 1 | Chan | 20201 | Engineering |
| 101 | Planning | Y | 2 | Zhang | 20202 | Financial |
| 303 | Analyzing Management Practices | Y | 3 | Lai | 20203 | Healthcare |
| 404 | Implementing solutions | Y | 4 | Luo | 20204 | Human Resources |
| 505 | Designing | Y | 5 | Zhen | 20201 | Engineering |

The primary key is a composite key of Skill\_ID and ConsultantID because they are in a many-to-many relationship : a skill can be opted by more than one consultant, and a consultant can opt for more than one skill.

Now we have the following design in 1FN :

Skill1 (Skill\_ID, Description, Certified, ConsultantID, Name, DepartmentNumber, DepartmentName)

The design above, Skill1, is not in 2FN because there are partial dependencies.

1. 2nd Normal Form

These following rules must be respected for the table to be in 2FN :

* The table should be in 1st Normal Form.
* The table should not have any partial dependencies.

By observing the table Skill1, we notice four partial dependencies :

* The Description only depends on Skill\_ID and not necessarily on ConsultantID.
* The Name, DepartmentNumber and DepartmentName only depends on ConsultantID and not necessarily on Skill\_ID.

Certified is functionally dependent on the whole primary key (ConsultantID + Skill\_ID). It is impossible to find a specific Certified record with ConsultantID only or with Skill\_ID only in this case, which means Certified is not a partial dependency.

We are to divide the table Skill1 in 3 new tables to remove the partial dependencies :

* Skill2 (Skill\_ID, Description)
* Consultant2 (ConsultantID, Name, DepartmentNumber, DepartmentName)
* Certification2 (ConsultantID, Skill\_ID, Certified)

Certification2 Skill2

|  |  |  |
| --- | --- | --- |
| Skill\_ID | ConsultantID | Certified |
| 101 | 1 | N |
| 101 | 2 | Y |
| 303 | 3 | Y |
| 404 | 4 | Y |
| 505 | 5 | Y |

|  |  |
| --- | --- |
| Skill\_ID | Description |
| 101 | Planning |
| 303 | Analyzing Management Practices |
| 404 | Implementing solutions |
| 505 | Designing |

Consultant2

|  |  |  |  |
| --- | --- | --- | --- |
| ConsultantID | Name | DepartmentNumber | DepartmentName |
| 1 | Chan | 20201 | Engineering |
| 2 | Zhang | 20202 | Financial |
| 3 | Lai | 20203 | Healthcare |
| 4 | Luo | 20204 | Human Resources |
| 5 | Zhen | 20201 | Engineering |

PK in Consultant2 = ConsultantID

PK in Skill2 = Skill\_ID

PK in Certification2 = ConsultantID + Skill\_ID (composite key)

FK1 in Certification2 = ConsultantID

FK2 in Certification2 = Skill\_ID

The following design is not in 3FN because there is a transitive dependency :

Consultant2 (ConsultantID, Name, DepartmentNumber, DepartmentName)

Skill2 (Skill\_ID, Description)

Certification2 (Skill\_ID, ConsultantID, Certified)

1. 3rd Normal Form

These following rules must be respected for the table to be in 3FN :

* The table should be in 2nd Normal Form.
* The table should not have any transitive dependencies.

By observing the table Consultant2, we notice a transitive dependency :

The DepartmentName only depends on DepartmentNumber and not necessarily on the primary key of ConsultantID. DepartmentNumber is not part of the primary key in the table Consultant2.

We are to divide the table Consultant2 in two tables which results to this following design to remove the transitive dependency :

* Consultant3 (ConsultantID, Name, DepartmentNumber)
* Skill2 (Skill\_ID, Description)
* Certification2 (Skill\_ID, ConsultantID, Certified)
* Department3 (DepartmentNumber, DepartmentName)

Certification2 Skill2

|  |  |  |
| --- | --- | --- |
| Skill\_ID | ConsultantID | Certified |
| 101 | 1 | N |
| 101 | 2 | Y |
| 303 | 3 | Y |
| 404 | 4 | Y |
| 505 | 5 | Y |

|  |  |
| --- | --- |
| Skill\_ID | Description |
| 101 | Planning |
| 303 | Analyzing Management Practices |
| 404 | Implementing solutions |
| 505 | Designing |

Consultant3 Department3

|  |  |  |
| --- | --- | --- |
| ConsultantID | Name | DepartmentNumber |
| 1 | Chan | 20201 |
| 2 | Zhang | 20202 |
| 3 | Lai | 20203 |
| 4 | Luo | 20204 |
| 5 | Zhen | 20201 |

|  |  |
| --- | --- |
| DepartmentNumber | DepartmentName |
| 20201 | Engineering |
| 20202 | Financial |
| 20203 | Healthcare |
| 20204 | Human Resources |

PK in Department3 = DepartmentNumber

PK in Consultant3 = ConsultantID

FK in Consultant3 = DepartmentNumber

PK in Skill2 = Skill\_ID

PK in Certification2 = ConsultantID + Skill\_ID (composite key)

FK1 in Certification2 = ConsultantID

FK2 in Certification2 = Skill\_ID

Part C: What is your conclusion from the result of Part A and Part B.

We are able to identify the same primary key, the same foreign keys, and produce the same final design and the same entity relationship diagram (ERD) between Part A and Part B. We have started with the following initial designs :

* Consultant(ConsultantID, Name, DepartmentNumber, DepartmentName, (Skill\_ID, Description, Certified))
* Skill(Skill\_ID, Description, Certified, (ConsultantID, Name), DepartmentNumber, DepartmentName)

We notice that despite the name between both tables (entities) is different, both contain the same set of attributes each. We also notice that each initial design has a different primary key identified. When we start normalizing the initial designs, we notice that both ConsultantID and Skill\_ID makes together a composite primary key in Part A and Part B which ultimately leads us to a same or similar final design in both cases.