TASK1

Insertion anomalies :

Suppose we want to insert a new student, St3, into the table, but this student is not currently enrolled in any unit. Since the table does not allow for the insertion of a student without a corresponding unit, we would be unable to insert the record for St3.

Imagine we want to add a new tutor, Tut2, to the table. However, Tut2 has not yet tutored any students or units. Since the table requires a tutor to have at least one associated record in the table (with a student and unit), we would be unable to insert Tut2 without inventing a dummy student or unit.

Deletion anomalies:

Suppose we want to delete a specific student's record from the table, let's say St1. However, St1 is associated with multiple units and has received tutoring from different tutors. If we delete St1's record from the table, we will lose all information about the units they were enrolled in, the tutors who provided tutoring, and the corresponding grades. This results in a loss of relevant information that could be useful for analysis or historical purposes.

Imagine we want to delete a tutor's record, Tut1, from the table. However, Tut1 has tutored multiple students in various units. If we directly delete Tut1's record without considering the associated tuples, we may end up with incomplete deletion. The units Tut1 tutored and the corresponding grades will remain in the table, but the tutor's information will be removed. This incomplete deletion can lead to inconsistent or orphaned data.

Update anomalies:

Suppose we want to update the email address (TutEmail) for a specific tutor, Tut1, in the table. Let's say we want to change Tut1's email address from "tut1@fhbb.ch" to "tut1newemail@fhbb.ch". However, if there are multiple records in the table with the same TutEmail value (in this case, "tut1@fhbb.ch"), updating Tut1's email address may lead to inconsistencies.

Suppose we want to update the Grade for a particular record, such as changing the Grade for UnitID U1 and StudentID St1 from 4.7 to 4.5. However, if there are other records in the table with the same (UnitID, StudentID) combination, updating the Grade for one record may result in inconsistencies. For example, if there is another record with the same (UnitID, StudentID) but a different Grade, the update will lead to a discrepancy between the two records.

TASK2

UnitID, StudentID -> Grade

UnitID -> TutorID

UnitID ,Date -> Topic, Room,Book

TutorId -> TutEmail

Assumptions:

UnitID: This column represents the identification code of a specific unit or course. For each unitID one tutor is assigned. But one tutor can be assigned to multiple units

Date: This column indicates the date when a specific tutoring session took place

Book: This column specifies the name or reference of the book or learning material used during the tutoring session. It indicates the educational resource utilized for teaching or supporting the student's learning. Only one book is associated with each tutoring session.

Topic: This column specifies the topic or subject of the book.

TASK 3

2NF(if we consider that topic is assigned for the book)

UnitID	StudentID	Grade
U1	St1	4.7
U2	St1	5.1
U1	St4	4.3
U5	St2	4.9
U4	St2	5

UnitID	Date	Topic	Book	Room
U1	23.02.2019	GMT	Deumlich	629
U2	18.11.2018	Gln	Zehnder	631
U1	23.02.2019	GMT	Deumlich	629
U5	05.05.2019	PHF	Dummlers	632
U4	04.07.2019	AVQ	SwissTopo	621

UnitID	TutorID
U1	Tut1
U2	Tut3
U5	Tut3
U4	Tut5

TutorID	TutEmail
Tut1	tut1@fhbb.ch
Tut3	tut3@fhbb.ch
Tut1	tut1@fhbb.ch
Tut3	tut3@fhbb.ch
Tut5	tut5@fhbb.ch

3NF

Table 1

UnitID	StudentID	Grade
U1	St1	4.7
U2	St1	5.1
U1	St4	4.3
U5	St2	4.9
U4	St2	5

Table 2

Book	Topic
Deumlich	GMT
Zehnder	Gln
Dummlers	PHF
SwissTopo	AVQ

Table 3

UnitID	Date	Book	Room
U1	23.02.2019	Deumlich	629
U2	18.11.2018	Zehnder	631
U5	05.05.2019	Dummlers	632
U4	04.07.2019	SwissTopo	621

Table 4

UnitID	TutorID
U1	Tut1
U2	Tut3
U5	Tut3
U4	Tut5

Table 5

TutorID	TutEmail
Tut1	tut1@fhbb.ch
Tut3	tut3@fhbb.ch
Tut1	tut1@fhbb.ch
Tut3	tut3@fhbb.ch
Tut5	tut5@fhbb.ch

TASK 4

UnitID,StudentId -> Grade

Book -> Topic

UnitId,Date -> Book,Room

UnitId -> TutorId

TutorID -> TutorEmail

Primary keys:

Table1: UnitID,StudentId

Table2: Book

Table3: UnitId,Date

Table4: UnitID

Table5: TutorID

Foreign keys:

Table1:UnitID

Table3: UnitID,Book

Table4:TutorID