BIG DATA MANAGEMENT

CZ/CE4123

Tutorial 2
Data Models

QUESTION1

Given the following relational schemas containing two tables (a.k.a., relations), and their primary keys are underlined:

Table1(A1, A2, A3)
Table2(A3, B1, B2)

Attribute A3 is the primary key of Table2 and is also the foreign key in Table1.

1) how do you convert Table 1 **alone** into key-value model? Give one possible solution.

2) how to convert them into key-value data model? Give one possible solution.

QUESTION1

Given the following relational schemas containing two tables (a.k.a., relations), and their primary keys are underlined:

Table1(A1, A2, A3)
Table2(A3, B1, B2)

Attribute A3 is the primary key of Table2 and is also the foreign key of Table2.

1) how do you convert Table 1 **alone** into key-value model? Give one possible solution.

Ans:
Key: A1
Value: A2;A3

QUESTION1

Given the following relational schemas containing two tables (a.k.a., relations), and their primary keys are underlined:

Table1(A1, A2, A3)
Table2(A3, B1, B2)

Attribute A3 is the primary key of Table2 and is also the foreign key of Table1. (You may assume that the query is always issued with respect to A1)

2) how to convert them into key-value data model? Give one possible solution.

Ans:
Step 1: (Left) join Table 1 and Table 2 using attribute A3, so that we have a bigger table T3(A1, A2, A3, B1, B2).
Step 2: Key: A1 Value: A2;A3;B1;B2

QUESTION2

Given the following relational schema containing three tables (primary keys are underlined):

Table1(A1, A2)

Table2(B1, B2)

Table3(A1, B1)

Please give one possible way to convert the above relational model into key-value model.
(You may assume that the query is always issued with respect to A1,B1)

QUESTION2

Given the following relational schema containing three tables (primary keys are underlined):

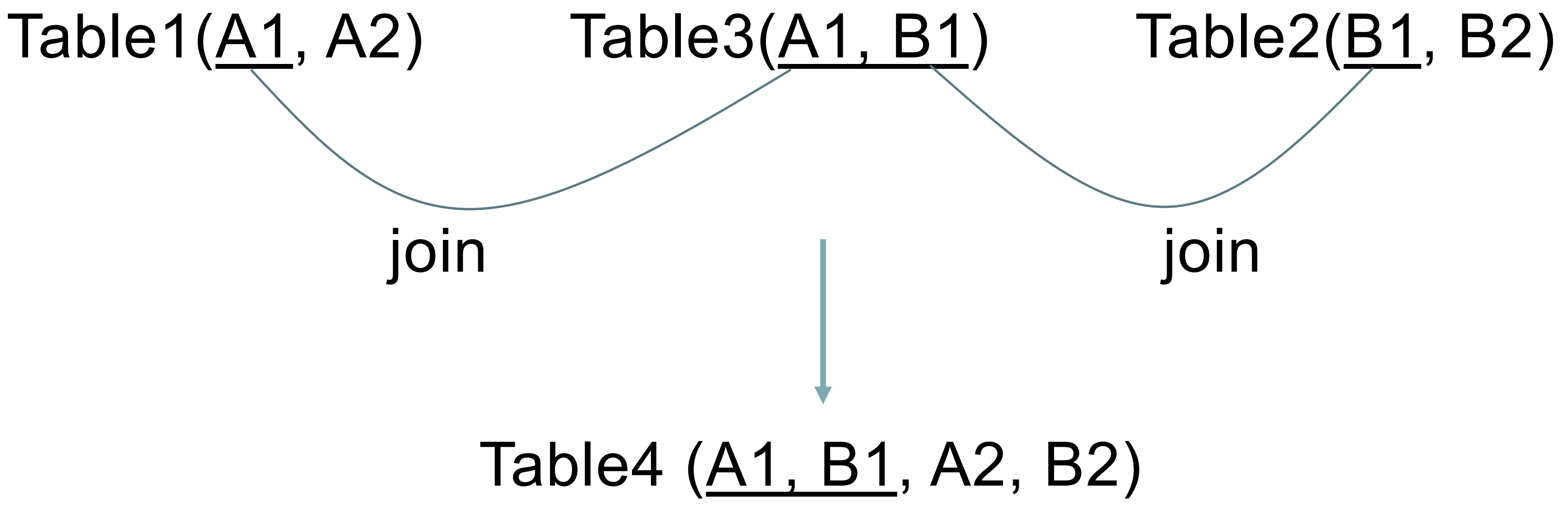
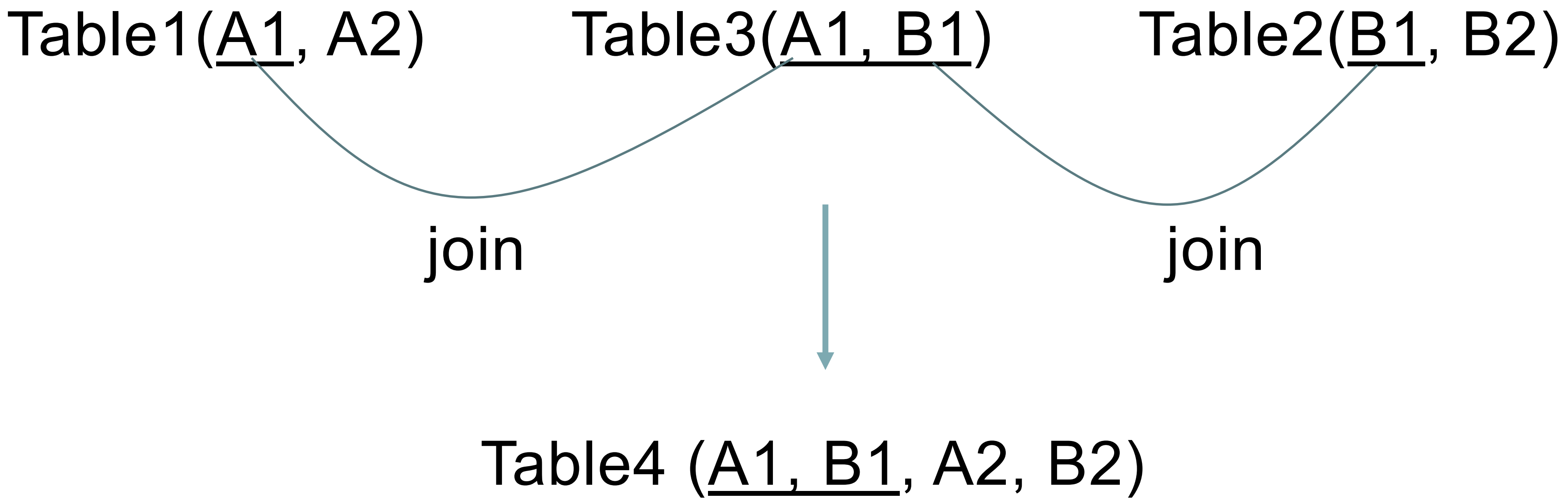


Table1($\underline{\text{A1}}$, A2) Table3($\underline{\text{A1, B1}}$) Table2($\underline{\text{B1}}$, B2)

Table4 ($\underline{\text{A1, B1}}$, A2, B2)

QUESTION2

Assume that the user is interested in querying the information for the combination of (A1,B1), then
Key: A1;B1
Value: A2;B2.



QUESTION 3

Can key-value model be converted into a relation? Why?

QUESTION 3

Can key-value model be converted into a relation? Why?

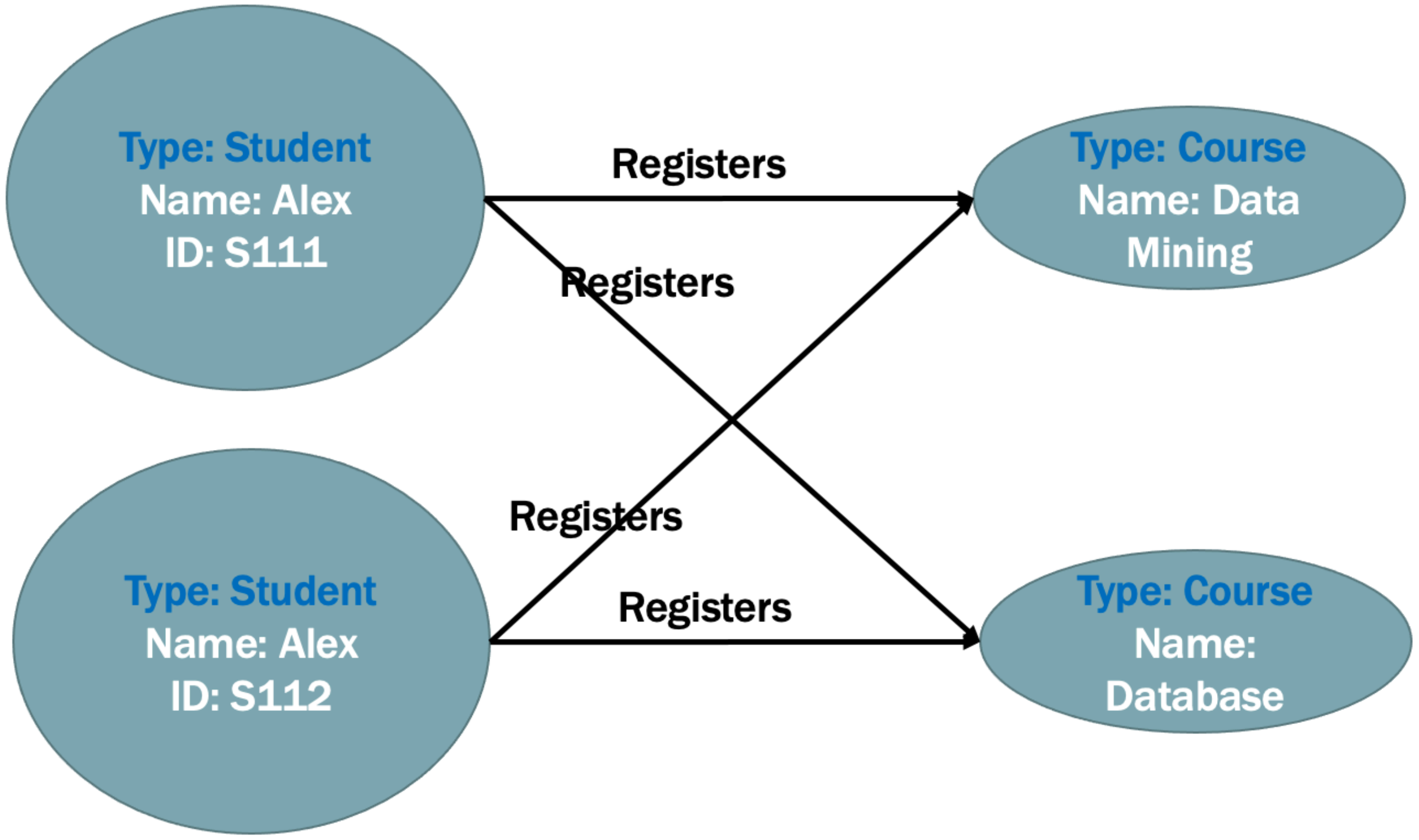
Ans: Key-value model can be trivially converted into a relation as follows

Table(<u>key</u>, value)

Note: This solution, however, is not able to uncover any schema that exists in **value**.

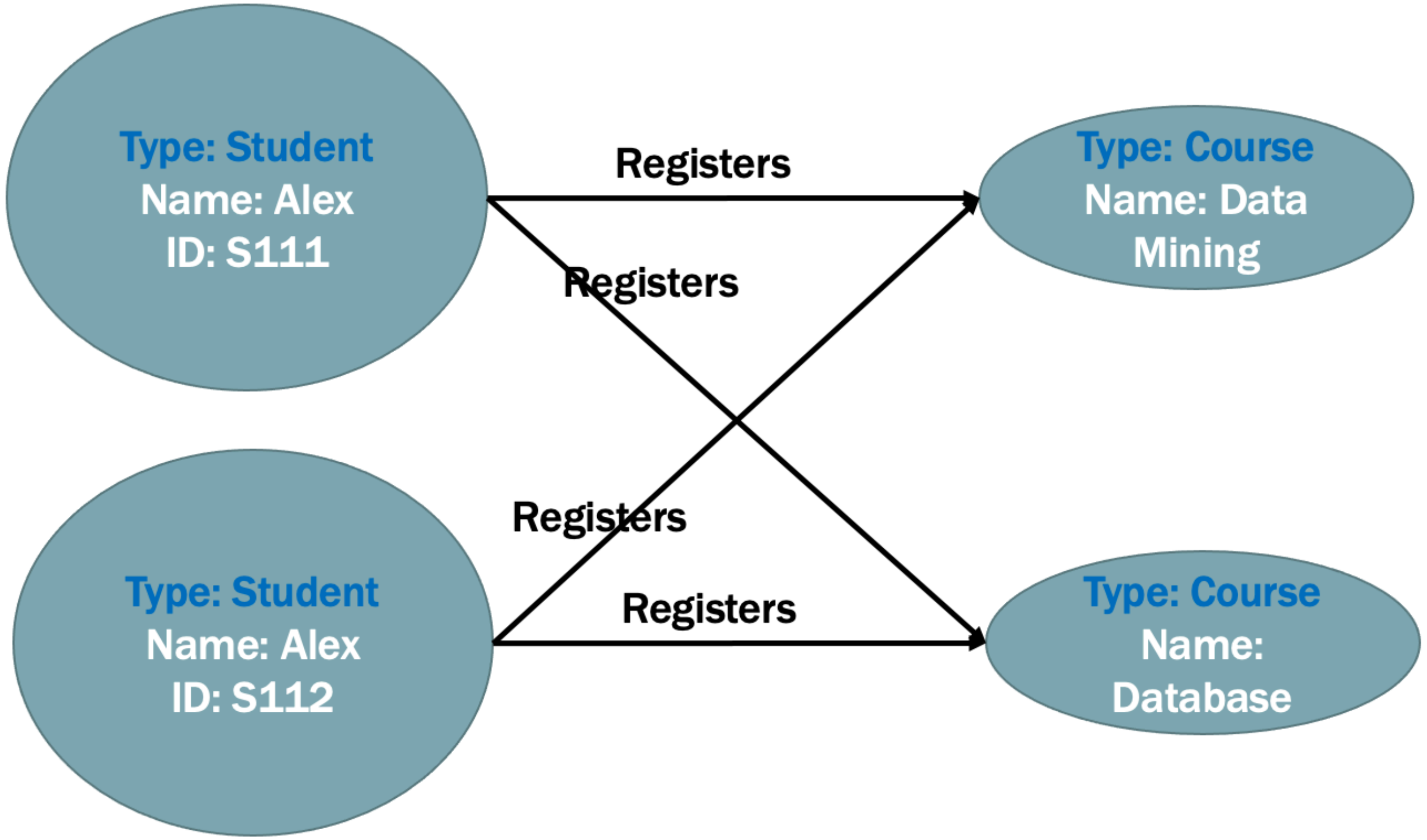
QUESTION 4

Given the following graph model, please convert it into relational data model.



QUESTION 4

Given the following graph model, please convert it into relational data model.

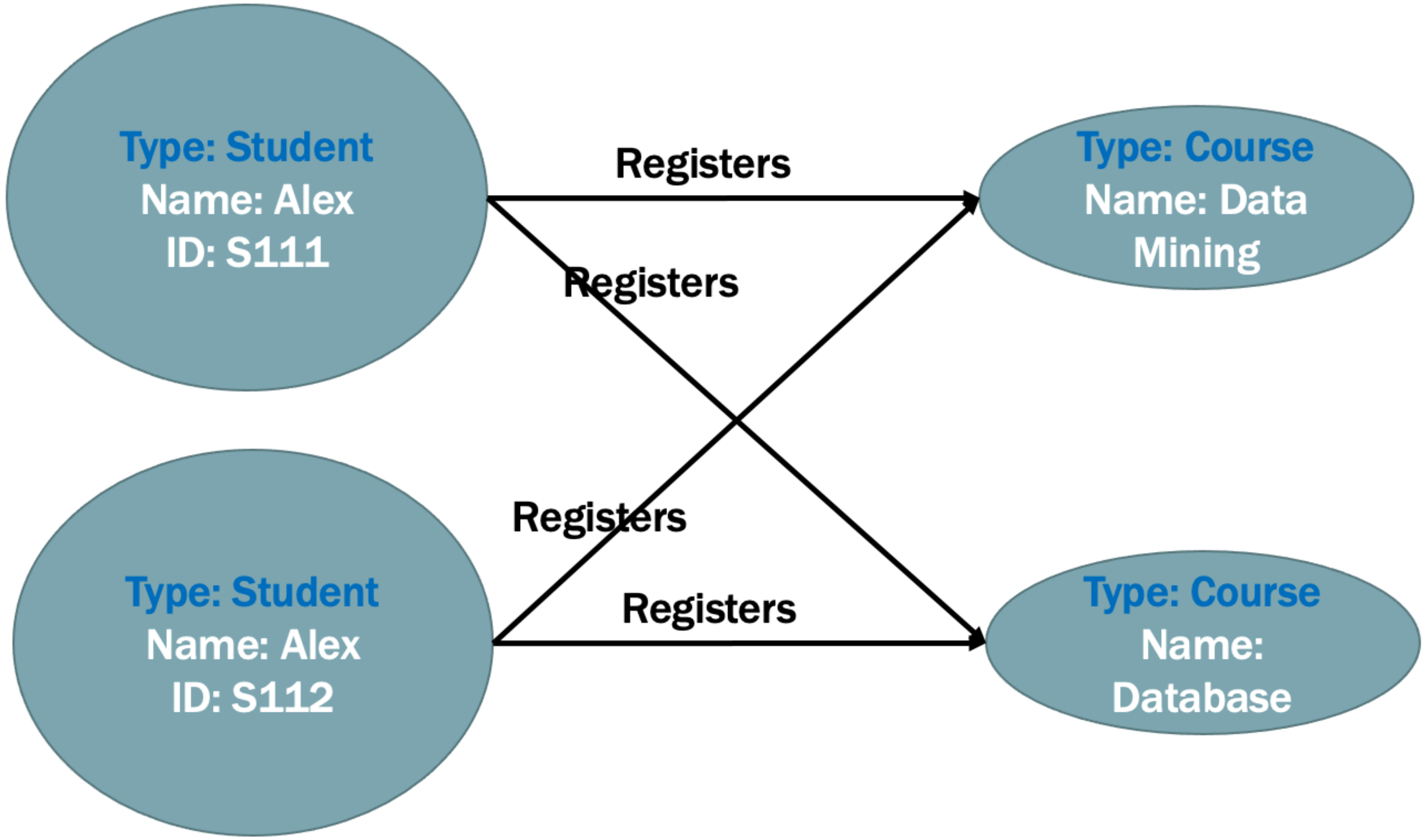


Step 1: consider how many tables are needed.

- ❑ *Type Student*→ **Student** Table
- ❑ *Type Course*→ **Course** Table

QUESTION 4

Given the following graph model, please convert it into relational data model.

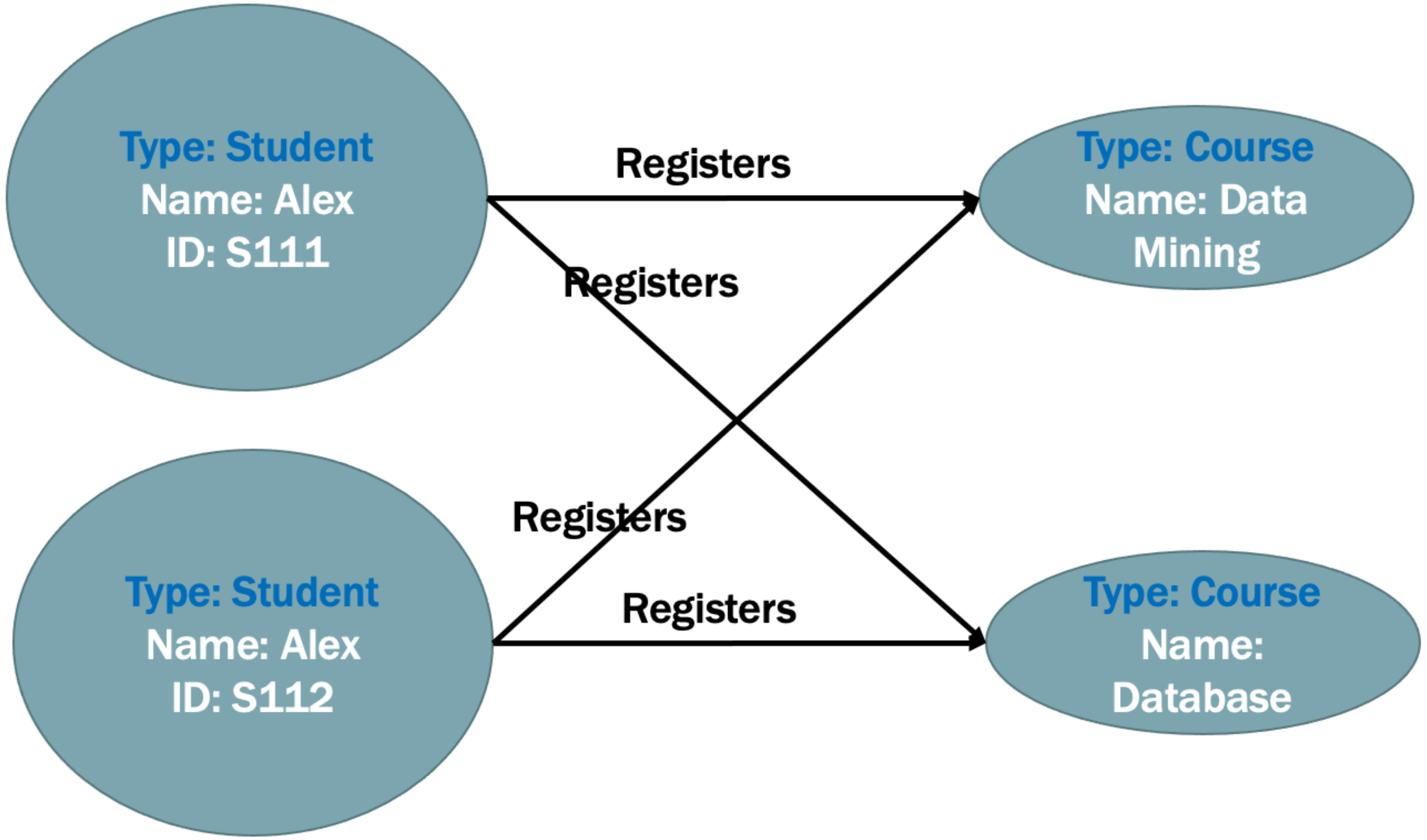


Step 1: consider how many tables are needed.

- ❑ *Type "Student"*→ **Student** Table
- ❑ *Type "Course"*→ **Course** Table
- ❑ *Relationship "Registers"*→ **Register** Table

QUESTION 4

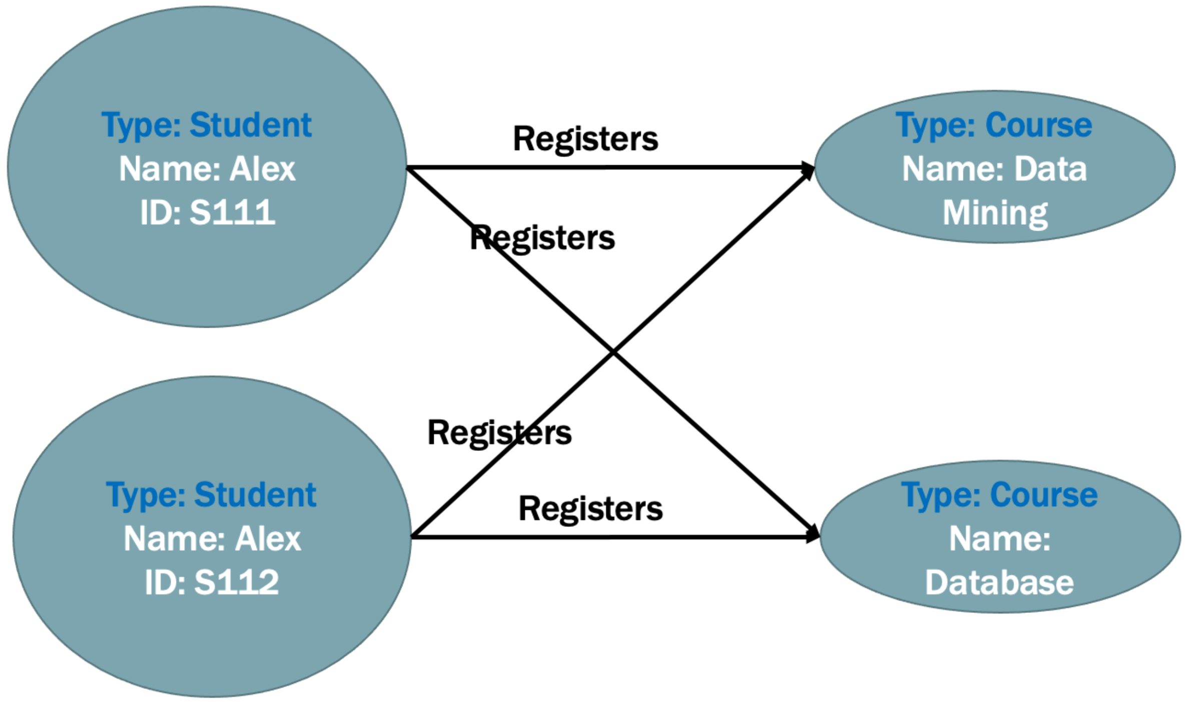
Given the following graph model, please convert it into relational data model.



Step 2: find attributes for each table.

Student (Name, <u>ID</u>)
Course (<u>Name</u>)
Register (<u>StudentID, CourseName</u>)

QUESTION 4



Student

ID	Name
S111	Alex
S112	Alex

Course

Name
Data Mining
Database

Register

StudentID	Name
S111	Data Mining
S111	Database
S112	Data Mining
S112	Database

QUESTION 5

Why we need the graph model? Discuss it from the physical storage’s perspective.

[redacted]

Ans: Graphs can be stored in the order based on adjacency list. In fundamental graph related operations such as graph traversal, the data access order aligns with the adjacency list. In contrast, relational tables often require costly “join” for these operations.

Example: Reconsider the example for Q4, we have two students S1, S2, and two courses C1, C2.

Nodes: S1, S2, C1, C2; **Edges**: (S1, C1), (S1, C2), (S2, C1), (S2, C2)

Adjacency list:

S1’s neighbor list: C1, C2
S2’s neighbor list: C1, C2
C1’s neighbor list: S1, S2 (if we do not consider the edge direction)
C2’s neighbor list: S1, S2 (if we do not consider the edge direction)