

Lesson 2

Graph Models

Graph

- Defines by a set of vertices (nodes) \mathcal{V} and edges (links) \mathcal{E} define a graph \mathcal{G}
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
- Graph \mathcal{G} is a set, which contains two sub-sets \mathcal{V} and \mathcal{E} .

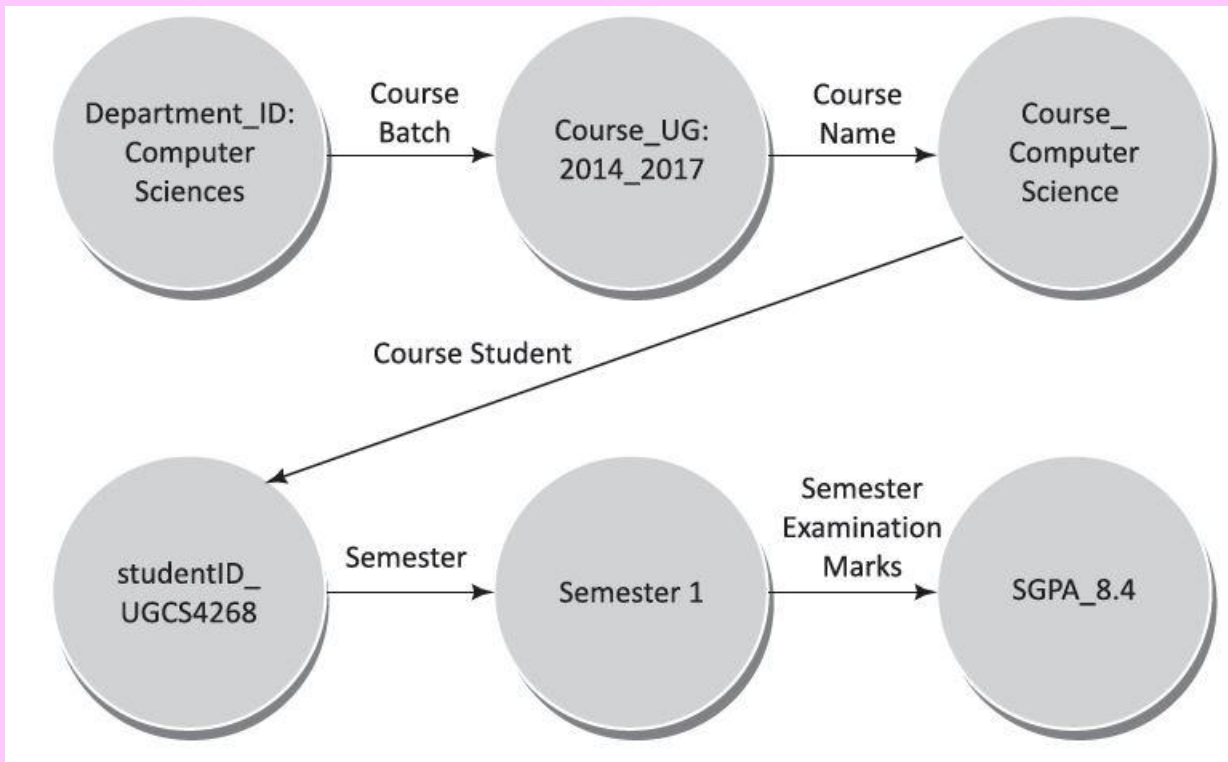
Elements of Vertices \mathcal{V}

- \mathcal{V} represents the entities
- A node or a vertex v in \mathcal{V} represents an entity, such as studentID [Example two nodes: ‘studentID’ and ‘studentSem1SGPA’.]

Set of Edges \mathcal{E}

- Represents the relations or associations
- An edge e connects the two nodes
- e represents a relation or association between the two entities
- An example of association is ‘**studies at**’.
- When entities are studentID and UG course at two interconnected nodes.

Figure 8.1 Graph model of a grade sheet



Distribution Function

- Represents the variation in degrees of the nodes on traversing from first to second, to third, to fourth, ...

A Label in graph model

- (i) A label near a vertex can specify identification for the **entity**. For example, *studentID*.
- (ii) A label near an edge can specify the context of **relation or association or value**. For example, “studies at” or “CS_Examination” or the grade point average in Semester 1, *Sem1SGPA*.

A weight in graph model

- (iii) A weight near the edge can specify the weight of a relationship with respect to other edges of the same kind.

Multiple Relations in a graph model

(iv) Multiple relationships can associate a pair of vertices interconnected by multiple edges.

Direction in A graph model

(v) Multiple relationships can associate a pair of vertices interconnected by multiple edges

Database as Graph

1. Graph models a database using graph parameters or properties assigned to each vertex, v and edge, e and hierarchical relations



Graph Database

- Graph database explicitly stores the relationships at each edge using hierarchy graph

Hierarchy in Graph

- A hierarchy graph explicitly stores hierarchal relationships
- Hierarchy relations between the tables, such as database tables do not store but implicit in the codes for a search or query.

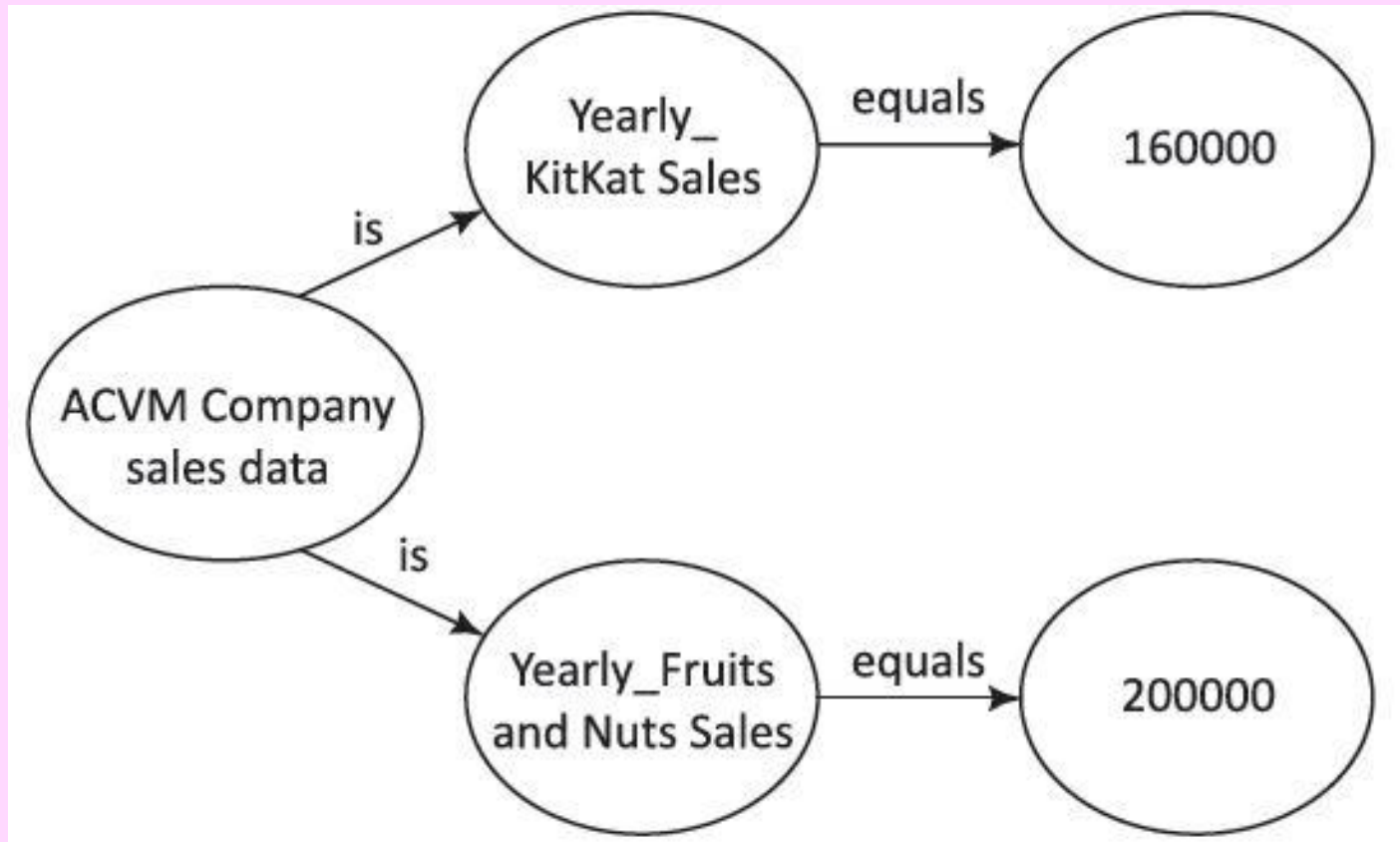
Representing a Graph as Triples

- Triple means a data entity consisting of three-components: subject, predicate and object. For example, consider a sentence, ‘Spark includes GraphX’. Here, **Spark** is the subject, **includes** is the predicate and **GraphX** is the object.

Triples as the Graph entities

- A triple in a directed graph is a sentence-like format
- A sentence three elements: ‘subject’ ‘predicate’ and ‘object’
- Similarly, a triple consists of three elements: source node, connecting edge, and destination node

Figure 8.2 Example of a graph model of yearly total sales of an ACVM company



Triple Format

- Format is instance identifier-property name-property value
- Example, **StudentID: 42629** obtained: “**GradePoint_Java** “8.2””.
StudentID: 42629 is the instance identifier. **Obtained** is the property name.

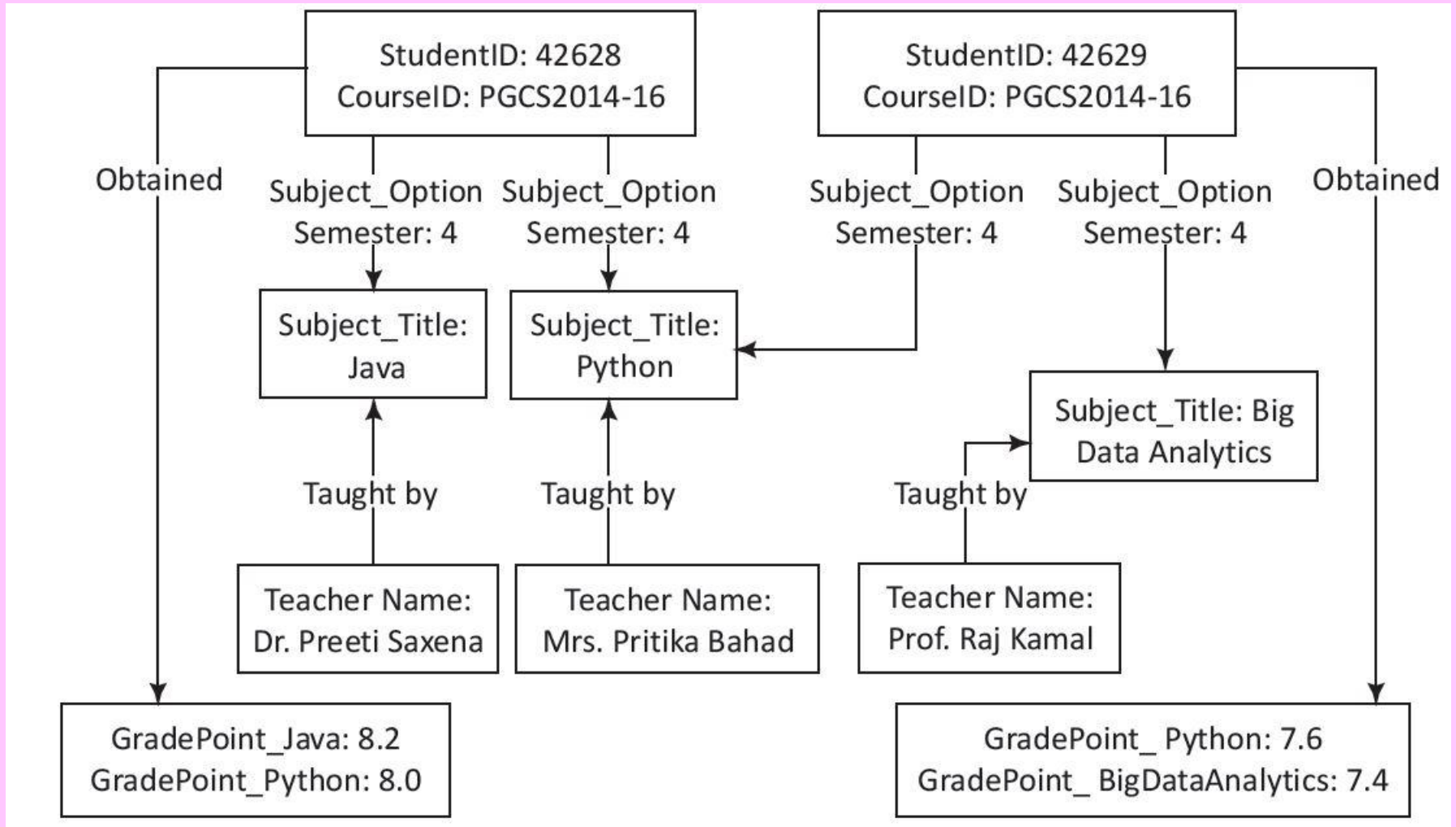
Triple Format

- Value of Property GradePoint_Java is 8.2.
- Graph model represents instance identifier and property value at two nodes and property name as the edge connecting them.

Property Graph Model

- A property graph assigns property to the nodes and edges.
- Property graph model (Figure 8.4) clearly shows the relationships and associations as compared to RDBMS tables (Figure 8.5)

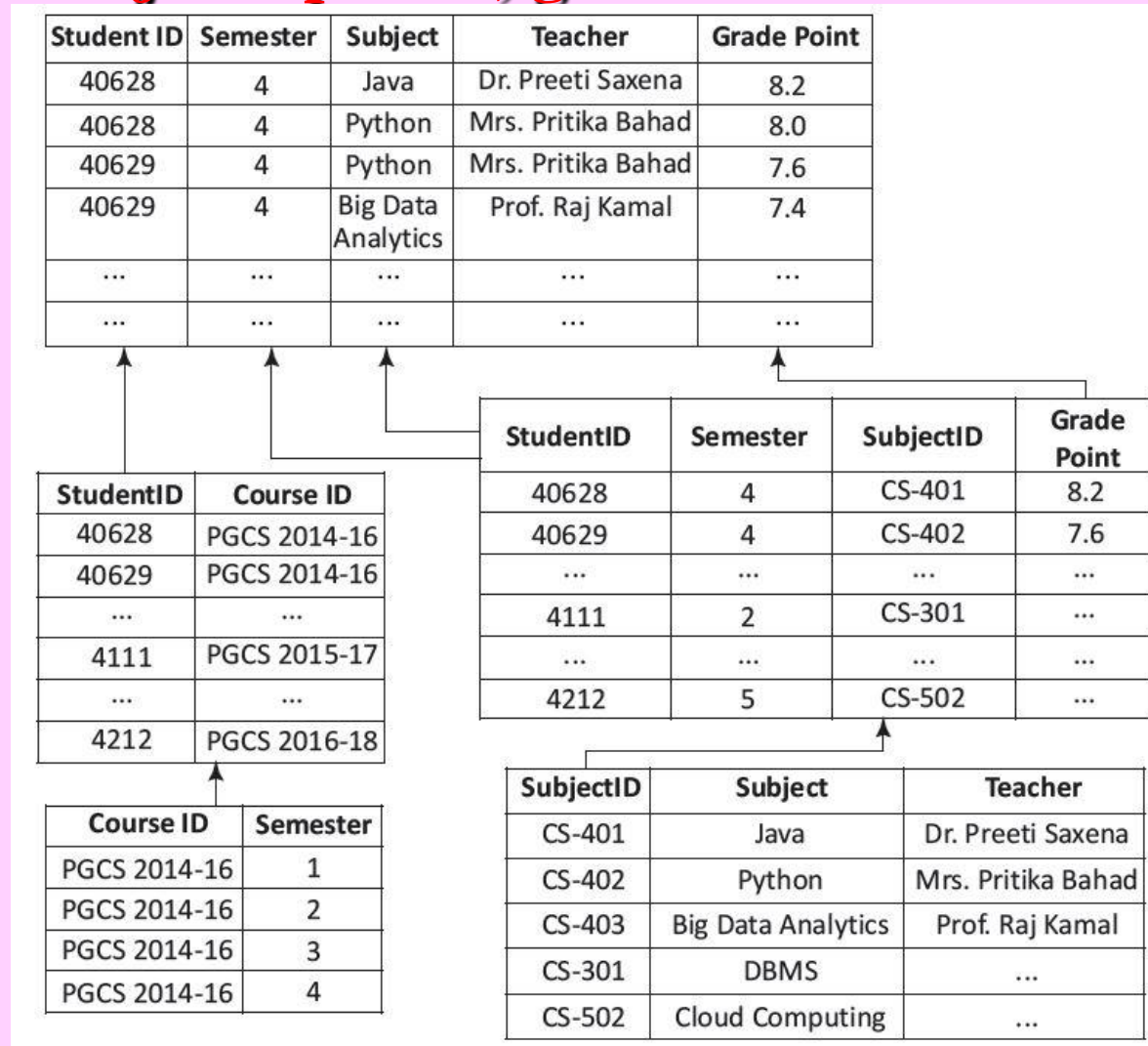
Figure 8.4 Property graph of students, semester, subject options, grades, and teachers



Property Graph Modeling a database

- Figure 8.5 RDBMS tables for students, semester, subject options, grades and teachers
- Property graph model represents the relationships or associations pictorially and are thus easy to interpret

Figure 8.5 RDBMS tables for students, semester, subject options, grades and teachers



Summary

We learnt:

- \mathcal{V} represents the entities and \mathcal{E} represents the relations or associations
- Labels at nodes and edges
- Graph model databases
- Property graph model represents the relationships or associations

Summary

We learnt:

- Triples as the Graph entities
- Format of Triple
- instance identifier
- property name
- property value

End of Lesson 2 on **Graph Models**