

FOUNDATIONS OF SQL

BY VISHWA PATEL

ABOUT ME

- Vishwa Patel
- Graduated from Dal in Fall 2024!
- Full time employee at Bell as Technical Network Architect
- Feel free to connect with me on LinkedIn with any questions!



Vishwa Patel

Technical Network Architect | New Grad at Bell | BBM



NOTES

- You can take it the way you like or refer to my slides
- To make ER diagram use link -
<https://drawsql.app/diagrams>
- ER Model - <https://erdplus.com/>



COURSE OUTLINE

6th January

Week 1 – Introduction to ER Diagrams

- Terminologies
- Designing an ER Diagram
- Converting ER diagrams to Database

13th January

Week 2 – DDL commands and SQL Basics

- Constraints
- Creating tables
- Inserting data into tables

20th January

Week 3 – Joins and Basic Functions

- Basic queries
- Joins
- Aggregate functions

27th January

Week 4 – Intermediate SQL concepts

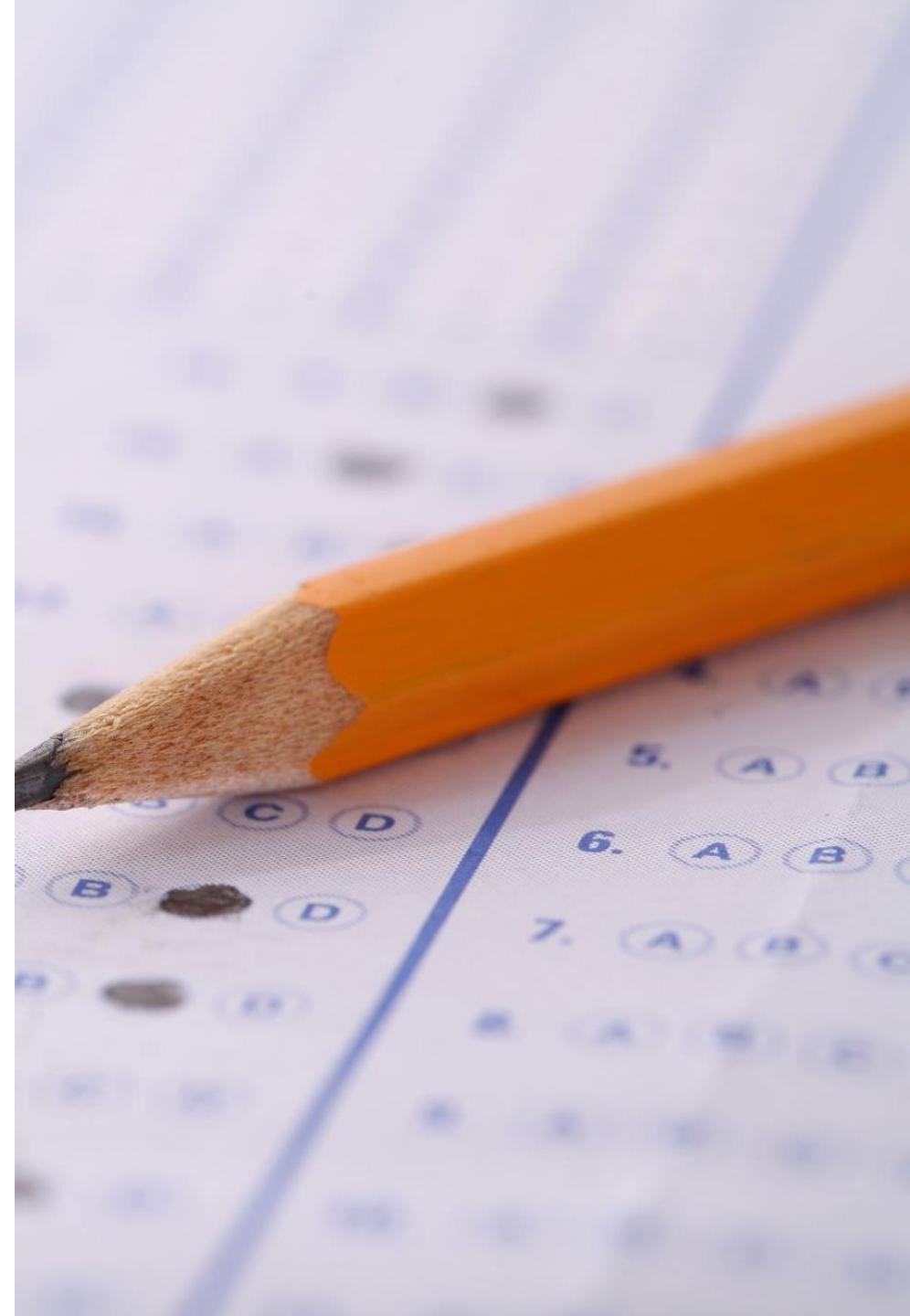
- Updating tables
- Subqueries
- Tips and Tricks

ASSESSMENT

Project submission required (further details to be provided soon!)

Criteria

- Need >85% in the project
- 75 % minimum attendance required (3/4 in-person classes)
- Deadline: 5th Feb 2025

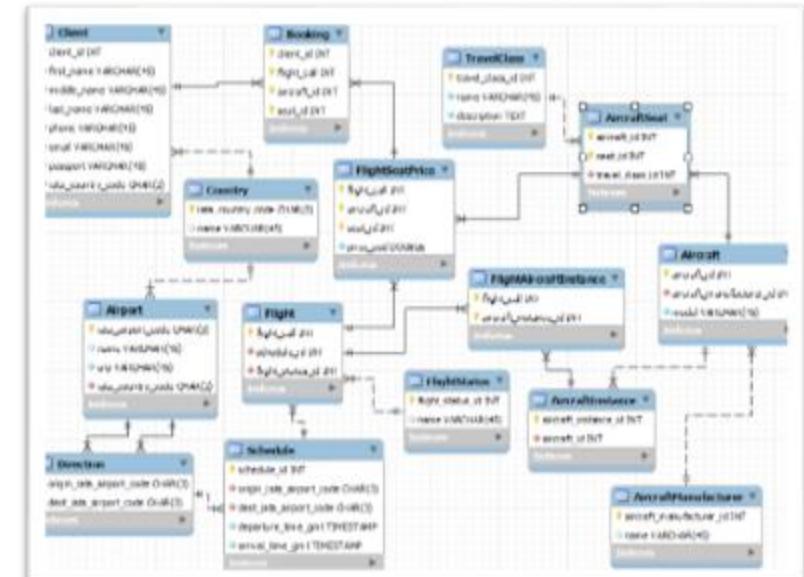


WHAT IS DATABASE (DB)?

- Any collection of related information
- Computers make it easier to handle large amounts of data

任何相关信息的集合

计算机使得处理大量数据变得更加容易



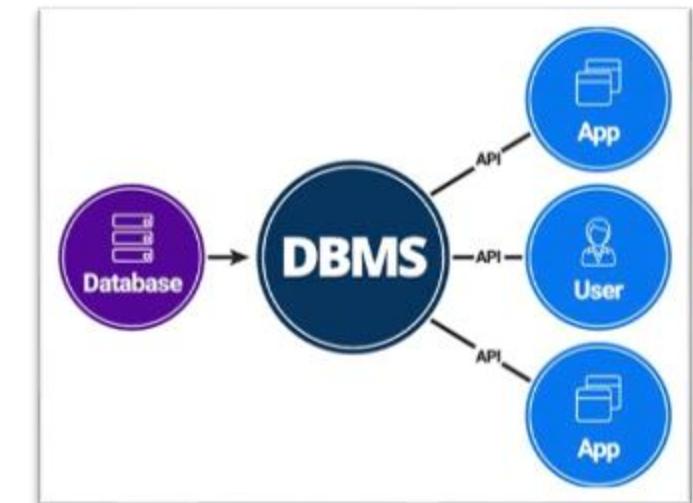
This Photo by Unknown Author is licensed under [CC BY-SA](#)

WHAT IS DBMS?

- Database Management System
- Special software program that helps users create and maintain database
 - Manages security of the data
 - Backups
 - Importing/Exporting Data
 - Easy to manage large amounts of data
 - Programming language can be used for interacting
- Performs CRUD operations – Create, Read/Retrieve, Update, Delete

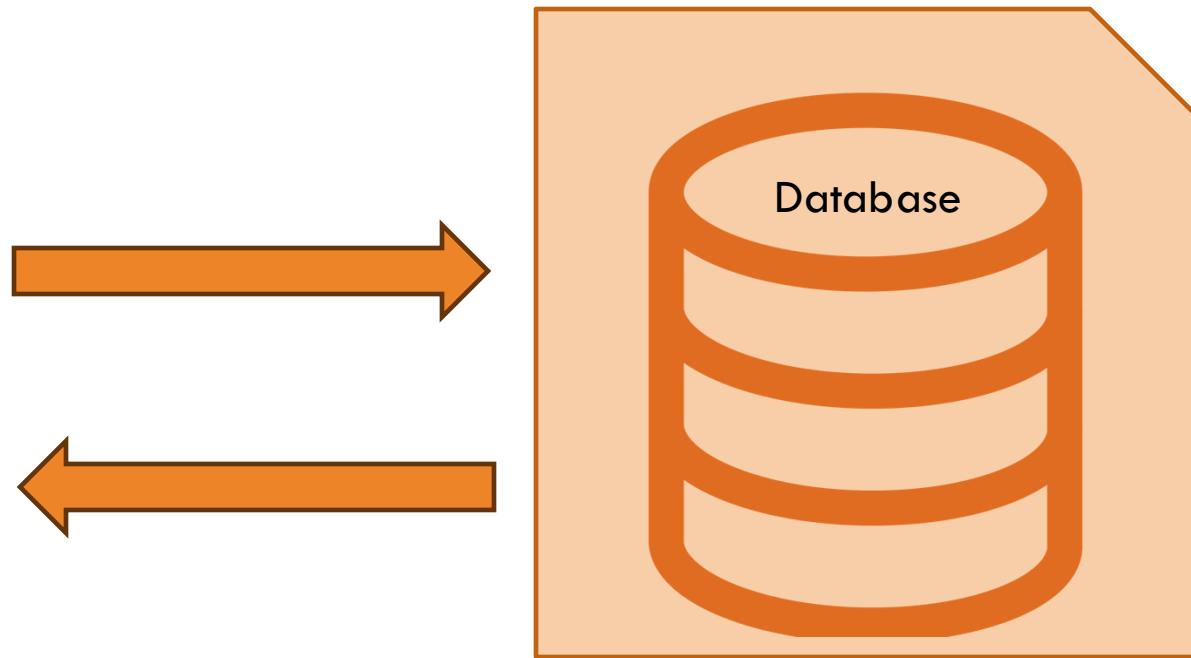
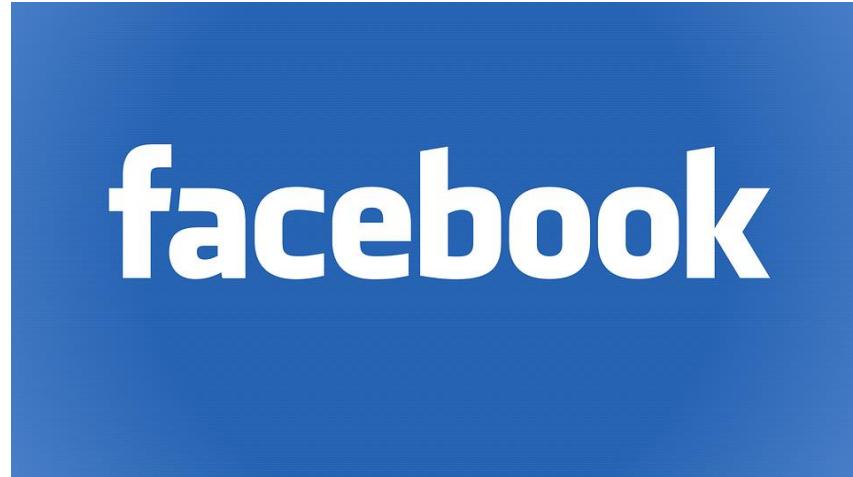
什么是DBMS?

- 数据库管理系统
- 一种特殊的软件程序，帮助用户创建和维护数据库
 - 管理数据的安全性
 - 备份
 - 数据的导入/导出
 - 便于管理大量数据
 - 可以使用编程语言进行交互
- 执行CRUD操作——创建、读取/检索、更新、删除



[This Photo](#) by Unknown Author is licensed under [CC BY-SA-NC](#)

DATABASE INTERACTION



Database Management System (DBMS)

TYPES OF DATABASES

Relational Database	Non-Relational Database
SQL Database	No SQL/Not only SQL
Organizes data into one or more tables	Any database with information, not a table
Tables have rows and columns, more like excel sheet	Documents like JSON, XML file, etc.
Unique key for each row	Key – value storage

关系型数据库	非关系型数据库
SQL数据库	NoSQL/非SQL
将数据组织成一个或多个表格	任何包含信息的数据库, 不一定是表格形式
表格有行和列, 类似Excel表格	文档形式, 如JSON、XML文件等
每行都有唯一的键	键-值存储

存储, 更像是Excel表格

WHAT IS RDBMS?

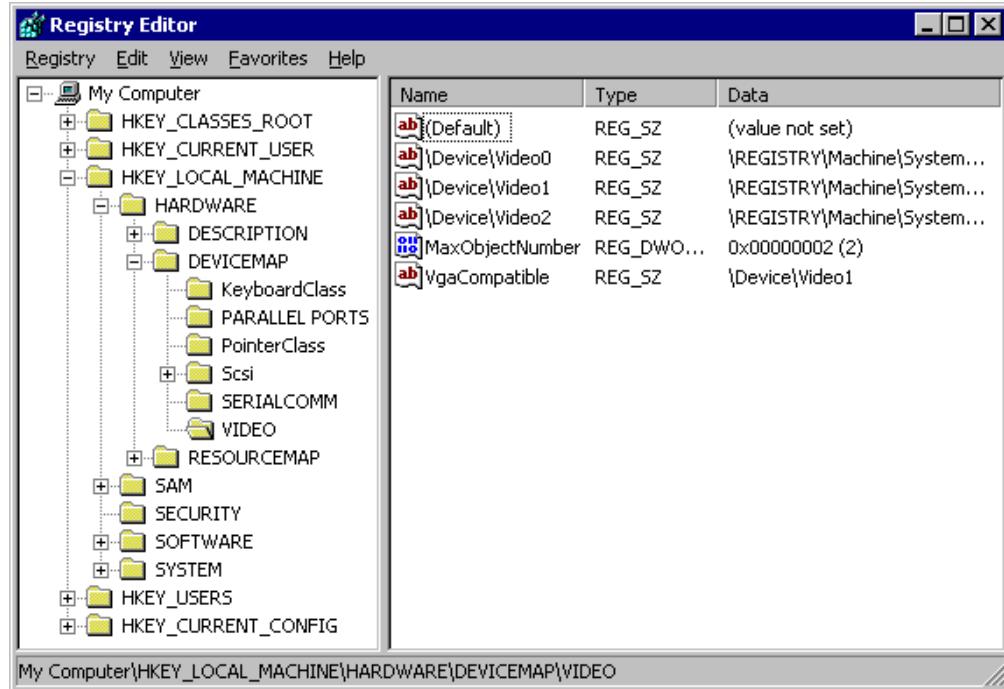
- **什么是RDBMS?**
 - 关系数据库管理系统 (Relational Database Management System)
 - 创建和维护关系数据库
 - 使用SQL —— 与RDBMS交互的标准化语言
 - SQL用于执行CRUD操作以及其他任务，如管理、备份等
 - 用于定义表格和结构
- Relational Database Management System
- Create and maintain relational databases
- Uses SQL – standardized language for interacting with RDBMS
- SQL used for CRUD operations and other tasks like administrative, backups, etc.
- Used to define tables and structure

DBMS VS RDBMS

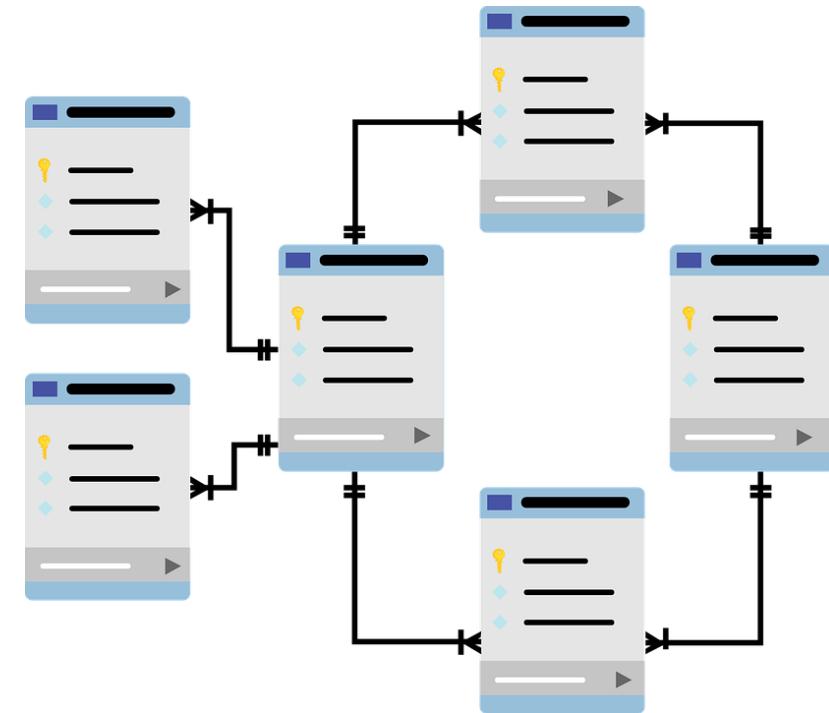
DBMS	RDBMS
Stores information as a file	Stores information in tables
Does not link data	Links data using keys, indexes, etc. to create relationships between tables.
Access one data at a time	Interact with multiple data elements at a time
Data redundancy is common	No Data redundancy
Deals with small amount of data	Can efficiently deal with large amounts of data
Example: Windows Registry	Examples: MySQL, PostgreSQL, Oracle, etc.

特征	DBMS	RDBMS
数据存储	作为文件存储	存储在表格中
数据关联	不建立数据之间的关联	使用键、索引等建立表格之间的关联
数据访问	一次访问一条数据	一次可以访问多条数据
数据冗余	数据冗余常见	数据冗余较少
数据量	处理小规模数据	能高效处理大规模数据
示例	Windows注册表	MySQL、PostgreSQL、Oracle等

DBMS VS RDBMS



<https://learn.microsoft.com/en-us/windows/win32/sysinfo/structure-of-the-registry>



[https://cdn.prod.website-files.com/644bb0d49c07b5dc9232d6f0/64aea5086620b6494985c654_Untitled%20\(4\).png](https://cdn.prod.website-files.com/644bb0d49c07b5dc9232d6f0/64aea5086620b6494985c654_Untitled%20(4).png)

WHAT IS NRDBMS?

什么是NRDBMS?

- 非关系数据库管理系统 (Non-Relational Database Management System)
- 帮助创建和维护非关系数据库
- 示例: MongoDB、Firebase等
- 它们有自己定义的语言, 用于与数据进行交互

- Non-Relational Database Management System
- Helps create and maintain non-relational databases
- [MongoDB, firebase, etc.](#)
- They have their own language defined to communicate with the data

RDBMS VS NRDBMS

Relational database storage

customer_id	name	email	dob	mobile
1	John Drake	john.drake@gmail.com	24/11/1982	7843241098
2	Mary Chile	mary.chile@outlook.com	05/06/1981	8903424531
3	Mac Adams	mac_1979@gmail.com	23/04/1979	0920421454
4	Jill Smith	jellyjill@gmail.com	14/02/1987	8795092014



Key value database storage

Key	Value
customer_1	{ "name": "John Drake", "email": "john.drake@gmail.com", "dob": "24/11/1982", "mobile": 7843241098 }
customer_2	{ "name": "Mary Chile", "email": "mary.chile@outlook.com", "dob": "05/06/1981", "mobile": 8903424531 }
customer_3	{ "name": "Mac Adams", "email": "mac_1979@gmail.com", "dob": "23/04/1979", "mobile": 0920421454 }
customer_4	{ "name": "Jill Smith", "email": "jellyjill@gmail.com", "dob": "14/02/1987", "mobile": 8795092014 }

<https://www.mongodb.com/resources/basics/databases/key-value-database#:~:text=key%2Dbased%20queries-,MongoDB%20as%20a%20key%20value%20store,for%20%E2%80%9Ckey%20value%E2%80%9D%20data.>

INTRODUCTION TO SQL

WHAT IS SQL?

什么是SQL？

- 1970年代由IBM开发
- 结构化查询语言 (Structured Query Language)
- 与关系数据库管理系统 (RDBMS) 交互
- MySQL是广泛使用的RDBMS
- 用于与数据库进行通信的语言

- Developed in 1970s by IBM
- Structured Query Language
- Interact with Relational Database Management System (RDBMS)
- MySQL is widely used as RDBMS
- Language to communicate with databases



WHY SQL?

- Faster processing speed
- Reliability and efficient
- Originally designed for non-programmers, makes it easy to learn
- Widely adopted
- Easily scalable

为什么选择SQL?

- 更快的处理速度
- 可靠且高效
- 最初为非程序员设计，易于学习
- 广泛采用
- 容易扩展

WHERE IS SQL USED?

SQL在哪里使用？

- 广泛应用于各个领域（医疗、金融、社交媒体等）
- Netflix、Instagram等顶级公司使用SQL进行数据分析、数据库升级和数据维护
- NASA使用SQL数据库管理航天探索任务的数据
- 在数据分析、工程和科学等职业中广泛使用

- Widely used in all sectors (Healthcare, Finance, Social Media)
- Top companies like Netflix, Instagram, etc. use it for data analysis, database upgrades and maintenance of data.
- NASA uses SQL databases to manage space exploration mission's data
- Used in careers like data analysis, engineering and science.

SUMMARY

数据库只是相关信息的集合。

- DBMS用于创建、管理数据库并执行CRUD操作。
- 数据库有两种类型：关系型和非关系型。
- 关系型数据库由RDBMS（如MySQL等）管理。
- RDBMS使用表格结构存储数据，而NRDBMS使用键值对格式存储数据。
- SQL用于数据分析和维护数据库。

- Databases are nothing but collection of related information.
- DBMS to create, manage databases and perform CRUD operations.
- Databases are of 2 types, relational and non-relational.
- Relational managed by RDBMS like [MySQL](#), etc.
- RDBMS stores data using table's structure whereas NRDBMS stores data in key-value formats.
- SQL is used in data analysis, maintaining the databases.

BUILD ER MODEL

LEARNING TERMINOLOGIES FOR STRUCTURING RELATIONAL DATABASE

WHAT IS TABLE?

什么是表格？

- 它是一个结构化的数据集合
- 按行和列组织
- 每一行都有一个唯一的ID，用于标识自身

customer_id	first_name	last_name	phone	email	street
1	Debra	Burks	NULL	debra.burks@yahoo.com	9273 Thome Ave.
2	Kasha	Todd	NULL	kasha.todd@yahoo.com	910 Vine Street
3	Tameka	Fisher	NULL	tameka.fisher@aol.com	769C Honey Creek St.
4	Daryl	Spence	NULL	daryl.spence@aol.com	988 Pearl Lane
5	Charolette	Rice	(916) 381-6003	charolette.rice@msn.com	107 River Dr.
6	Lyndsey	Bean	NULL	lyndsey.bean@hotmail.com	769 West Road
7	Latasha	Hays	(716) 986-3359	latasha.hays@hotmail.com	7014 Manor Station Rd.
8	Jacqueline	Duncan	NULL	jacqueline.duncan@yahoo.com	15 Brown St.
9	Genoveva	Baldwin	NULL	genoveva.baldwin@msn.com	8550 Spruce Drive
10	Pamelia	Newman	NULL	pamelia.newman@gmail.com	476 Chestnut Ave.

<https://www.sqlservertutorial.net/sql-server-basics/sql-server-select/>

WHAT IS AN ATTRIBUTE?

什么是属性？

- 表格的列
- 描述特定成员的值
- 例如: first_name (名字)、last_name (姓氏)、phone (电话) 等

- Column of the table
- Value to describe a specific member
- Here, **first_name**, **last_name**, **phone**, etc.

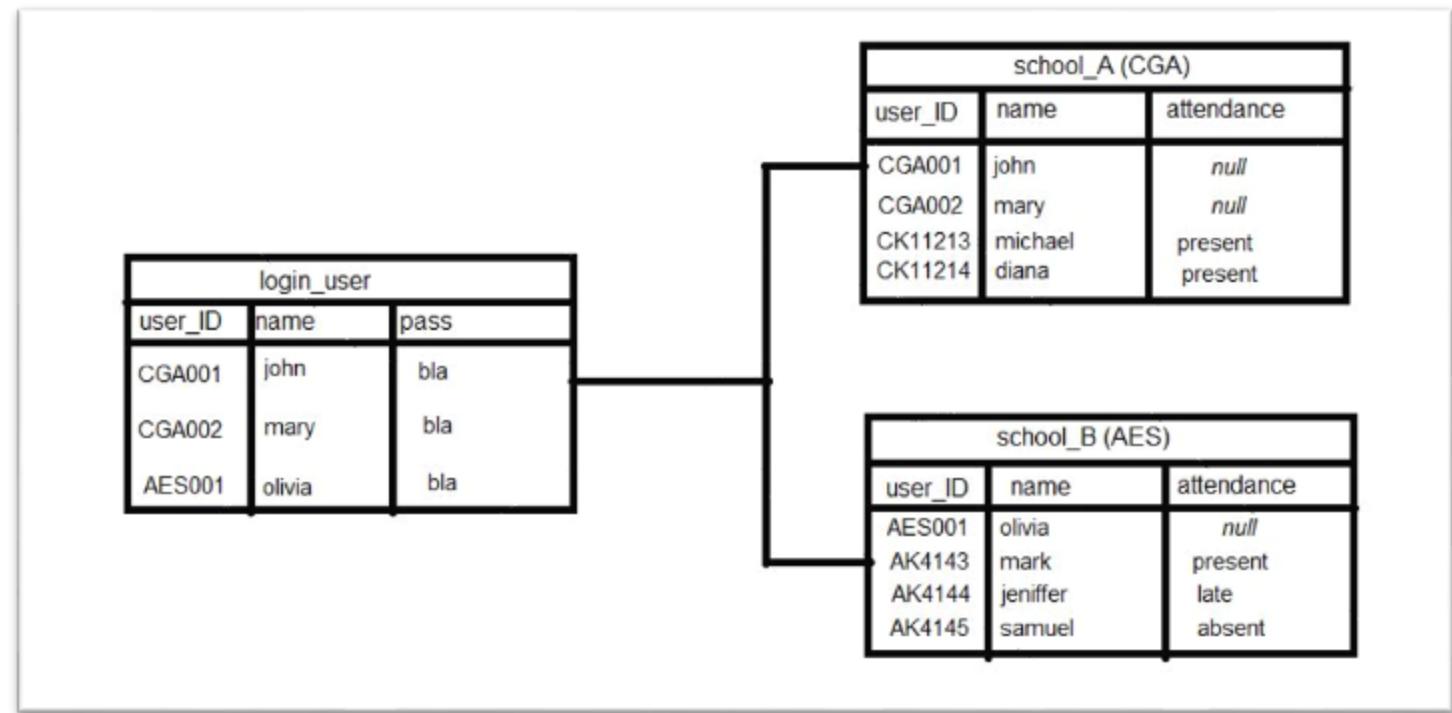
customer_id	first_name	last_name	phone	email	street
1	Debra	Burks	NULL	debra.burks@yahoo.com	9273 Thome Ave.
2	Kasha	Todd	NULL	kasha.todd@yahoo.com	910 Vine Street
3	Tameka	Fisher	NULL	tameka.fisher@aol.com	769C Honey Creek St.
4	Daryl	Spence	NULL	daryl.spence@aol.com	988 Pearl Lane
5	Charolette	Rice	(916) 381-6003	charolette.rice@msn.com	107 River Dr.
6	Lyndsey	Bean	NULL	lyndsey.bean@hotmail.com	769 West Road
7	Latasha	Hays	(716) 986-3359	latasha.hays@hotmail.com	7014 Manor Station Rd.
8	Jacqueline	Duncan	NULL	jacqueline.duncan@yahoo.com	15 Brown St.
9	Genoveva	Baldwin	NULL	genoveva.baldwin@msn.com	8550 Spruce Drive
10	Pamelia	Newman	NULL	pamelia.newman@gmail.com	476 Chestnut Ave.

<https://www.sqlservertutorial.net/sql-server-basics/sql-server-select/>

WHAT IS DATABASE SCHEMA?

- Structure of database in RDBMS
- Blueprint of the actual database
- Defines, tables, attributes and relationships.

什么是数据库模式?
- RDBMS中数据库的结构
- 实际数据库的蓝图
- 定义表格、属性和关系



<https://stackoverflow.com/questions/42217521/how-to-connect-each-person-in-a-table-to-another-table-respectively-in-a-database>



BUSINESS REQUIREMENT

The school system must maintain a database of students, capturing essential details such as their first name, last name, GPA, and a unique student number. Students can enroll in multiple classes, and each class can accommodate multiple students, reflecting a many-to-many relationship. For each class a student takes, their grades should be recorded to track academic performance. Additionally, classes may vary in structure, as some will include exams while others will not. This structure should support flexibility and accuracy in tracking academic records and class configurations.

业务需求

学校系统必须维护一个学生数据库，捕捉学生的基本信息，如名字、姓氏、GPA（平均绩点）和唯一的学生编号。学生可以注册多门课程，每门课程可以容纳多个学生，反映了多对多的关系。每个学生参加的课程，应该记录他们的成绩以跟踪学术表现。此外，课程结构可能会有所不同，有些课程包括考试，而有些则没有。该结构应支持灵活性和准确性，以便跟踪学术记录和课程配置。

PROBLEM STATEMENT

问题陈述

- 一所学校有许多学生。
 - 每个学生有多个属性，如名字、姓氏、GPA、学生编号。
 - 每个学生可以选修多门课程，课程可以包含多个学生。
 - 学生将获得他们所选课程的成绩。
 - 有些课程有考试，有些课程则没有。
- A school has many students.
 - Each student has multiple attributes like first name, last name, GPA, student number.
 - Each student can take multiple classes and classes can contain multiple students.
 - A student will get a grade for the classes they take.
 - Some classes will have exams, and some might not.



SOLUTION

Build an ER Model!

ER MODEL – ENTITY RELATIONSHIP MODEL

ER模型 – 实体关系模型

- 设计数据库模式 —— 所有不同的表格和表格中的属性
- 定义它们之间的关系
- 实际数据库实现与业务需求之间的中介
- 绘制实体、属性和关系的映射图

- Designing database schema – all different tables and attributes in the table
- Defining relationships among them
- Middleman between actual database implemented and the business requirements
- Map out entities, attributes and relationships

ENTITY

实体

- 可以定义并存储信息的事物或对象
- 可以将它们看作名词
- 它们被组成表格
- 示例：学生、教师、教授、部门等

- A thing or an object that can be defined and have stored information
- Think of them as nouns
- They are formed into tables
- Example: student, teacher, professor, department, etc.

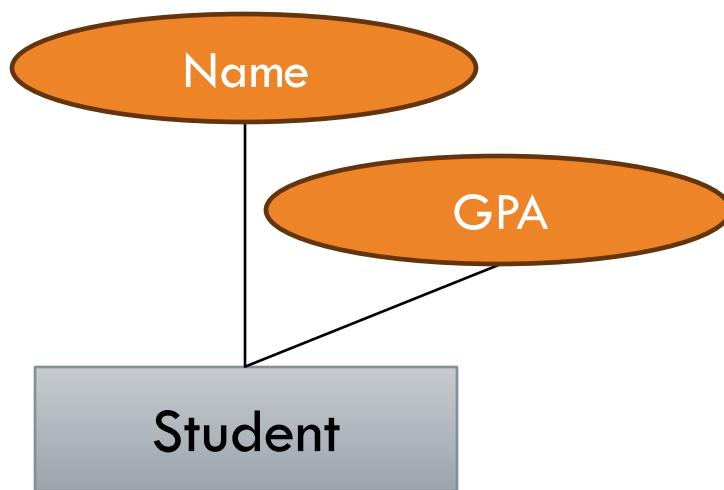
Student

ENTITY ATTRIBUTES

实体属性

- 关于实体的具体信息
- 它们描述实体
- 表格的列

- Specific information about the entity
- They describe the entity
- Columns of the table

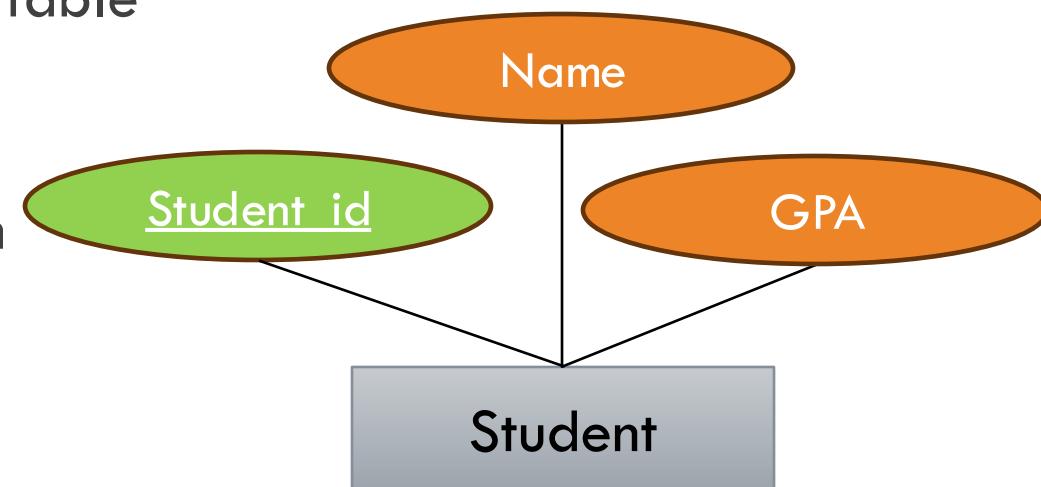


PRIMARY KEY

主键

- 唯一标识表格中条目的属性
- 通常主键不会以不同颜色显示
- 在ER图中，主键会被下划线标识
- 在这里，`Student_id` 将是主键

- An attribute that uniquely identifies an entry in the table
- Generally primary key is not in a different color
- Primary key is underlined to identify in ER diagram
- **Student_id** will be the primary key here

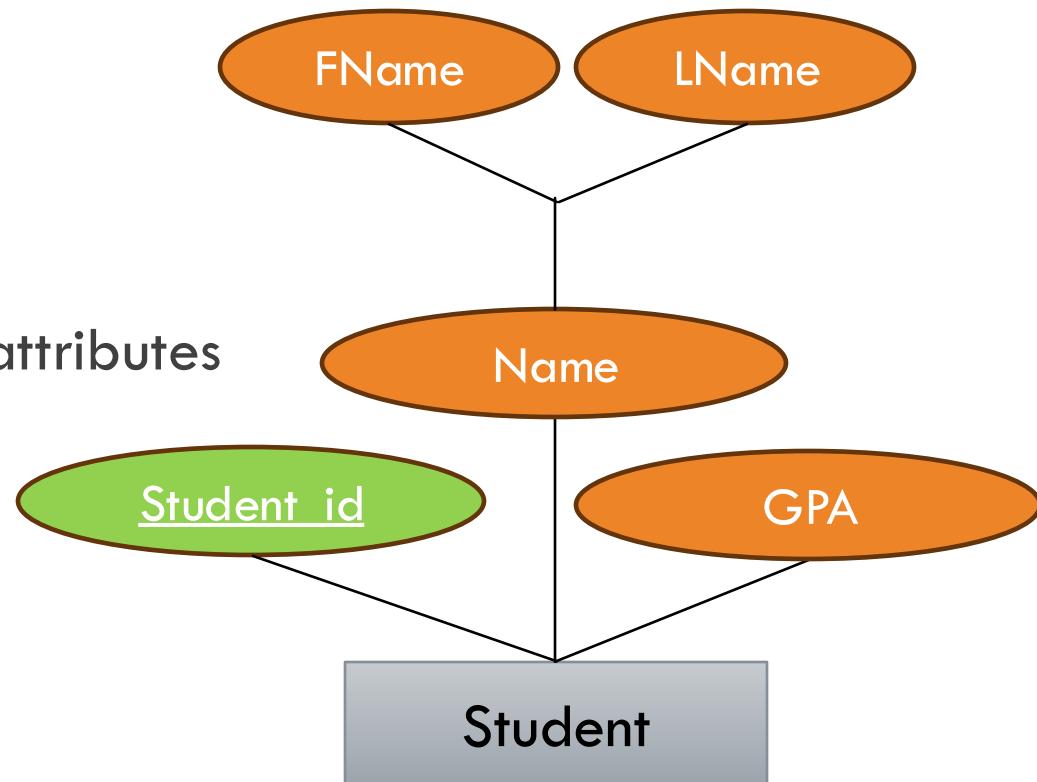


COMPOSITE ATTRIBUTE

复合属性

- 可以拆解成更小属性的属性
- 如姓名中的`FName`和`Lname`
- 主要属性与子属性连接

- An attribute that can be broken down into smaller attributes
- Like **FName** and **Lname** for **Name**
- Main attribute connected to the sub attributes

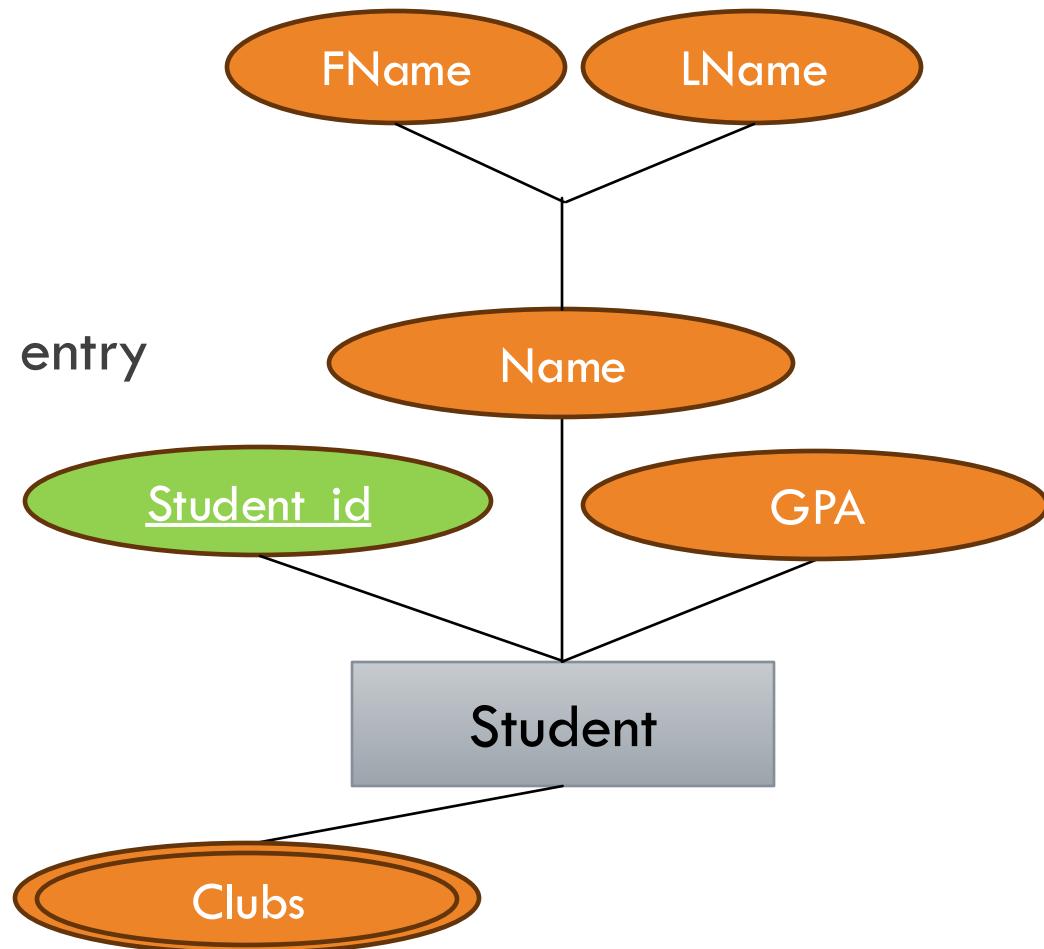


MULTI-VALUED ATTRIBUTES

多值属性

- 对于同一条目，具有多个值的属性
- 看起来像是带有额外圆圈的属性
- 学生可以参加多个社团

- An attribute with more than one value for the same entry
- Looks like attribute with an extra circle
- Students can be included in **more than one Clubs**

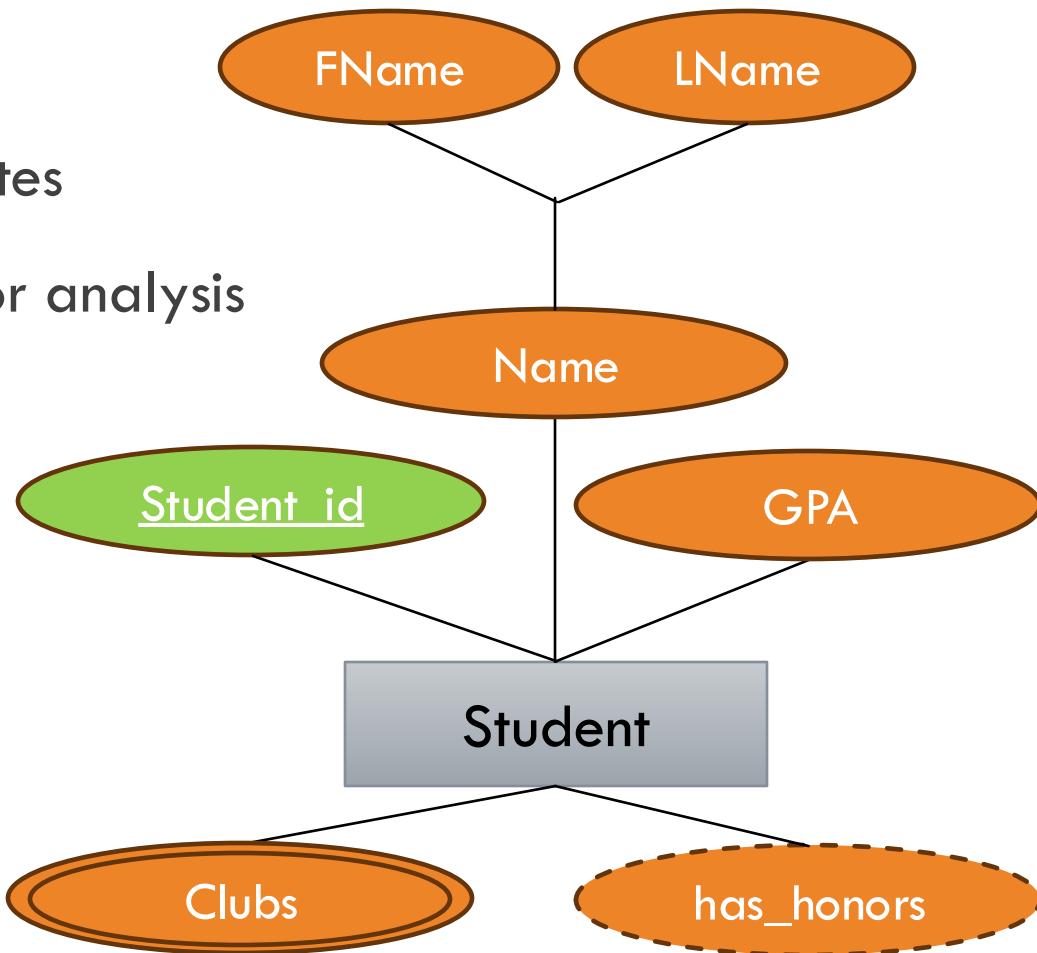


DERIVED ATTRIBUTE

派生属性

- 可以从其他属性推导出来的属性
- 我们不需要追踪它，只需在分析时推导出来
- `has_honors`，从GPA推导
- 不需要手动输入该属性的值
- 在检索时计算
- 以虚线边框的圆圈表示

- An attribute that can be derived from other attributes
- We won't keep track of it, we could just derive it for analysis
- **has_honors, derived from GPA**
- Don't need to manually enter the value for it
- Calculate it at time of retrieval
- Circle with dashed border

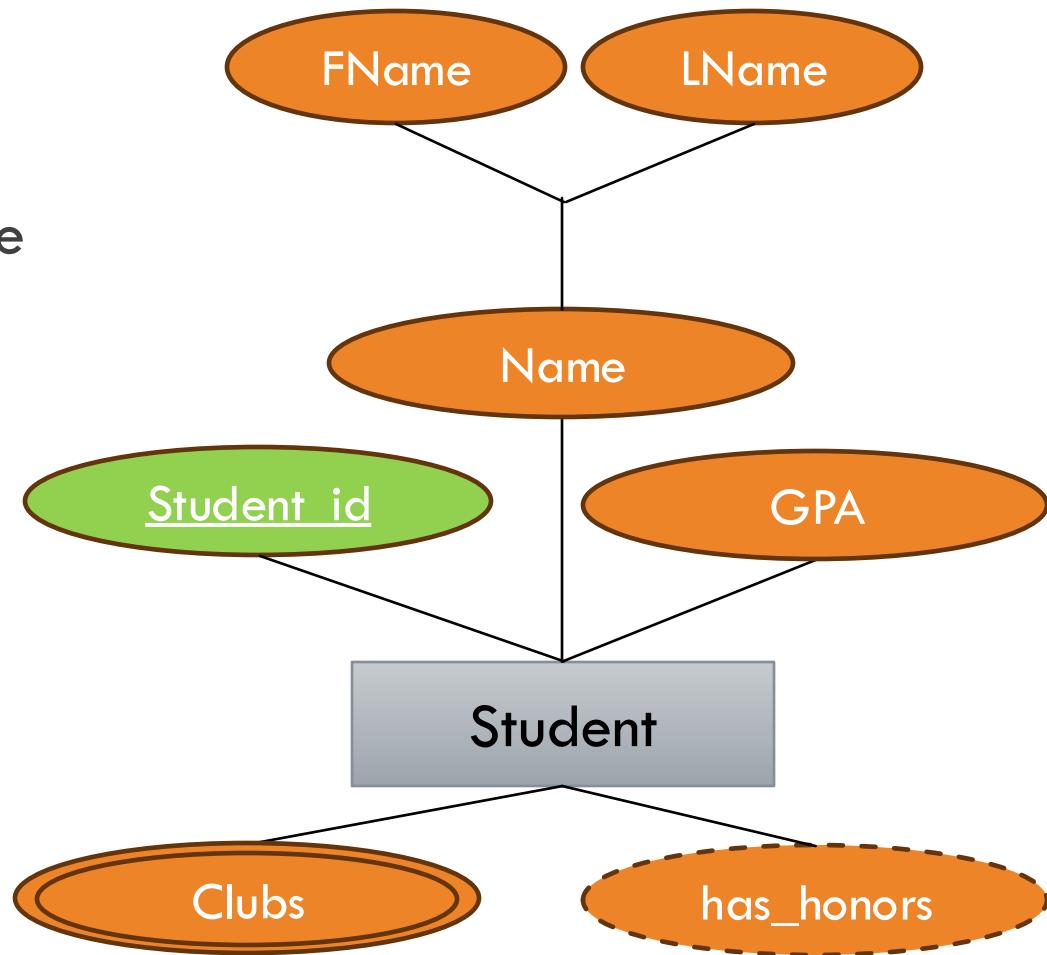
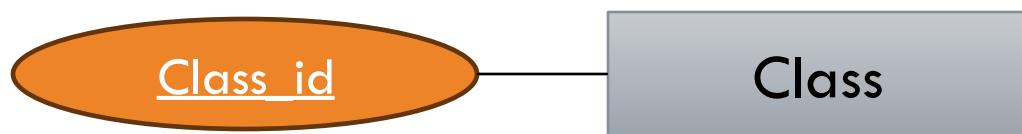


MULTIPLE ENTITIES

多个实体

- 根据问题陈述，您可以拥有任意数量的实体
- 我们可以有一个课程实体，用于存储关于课程的详细信息，如`class_id`、`name`等

- You can have as many entities as you like or require according to the problem statement
- We can have a class entity for storing the details about class like **class_id, name, etc.**



RELATIONSHIPS

关系

- 当我们有多个实体时，我们需要定义它们之间的关系
- 关系用菱形表示
- 它是一个动词，例如：`takes`（参加）、`works`（工作）、`controls`（控制）等

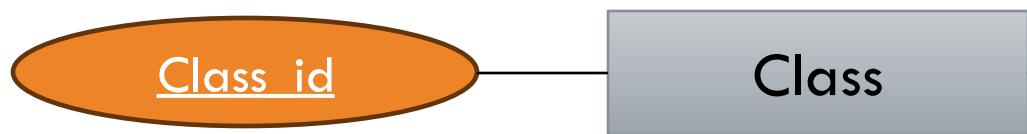
- When we have multiple entities, we would want to define how they are related to each other
- Define a relationship in a diamond shape
- It's a verb, like takes, works, controls, etc.

RELATIONSHIPS

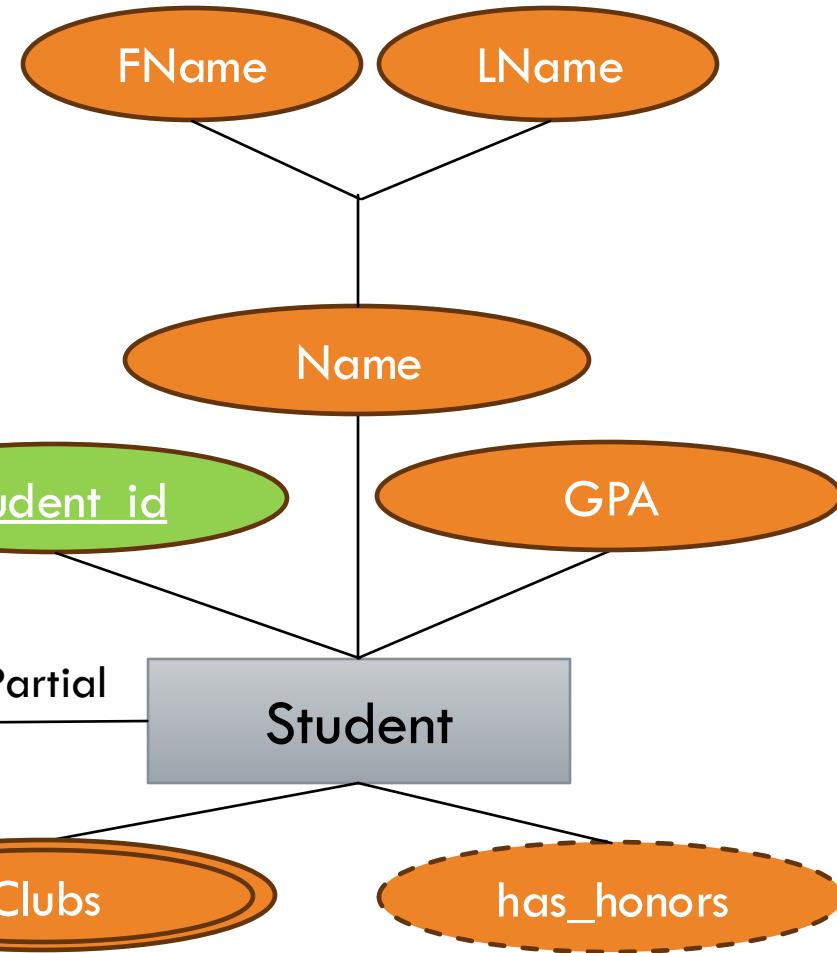
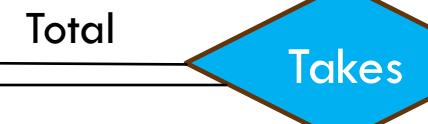
关系

- 学生参加课程，或者课程由学生参加
- 并非所有学生都需要参加课程（部分参与）
- 课程必须由学生参加（完全参与）
- 所有课程都需要有一些学生参加特定课程

- The student takes a class, or a class is taken by students
- Not all students need to take a class (Partial Participation)
- A class needs to be taken by students (Total participation)
- All classes need some students taking a particular class



Total



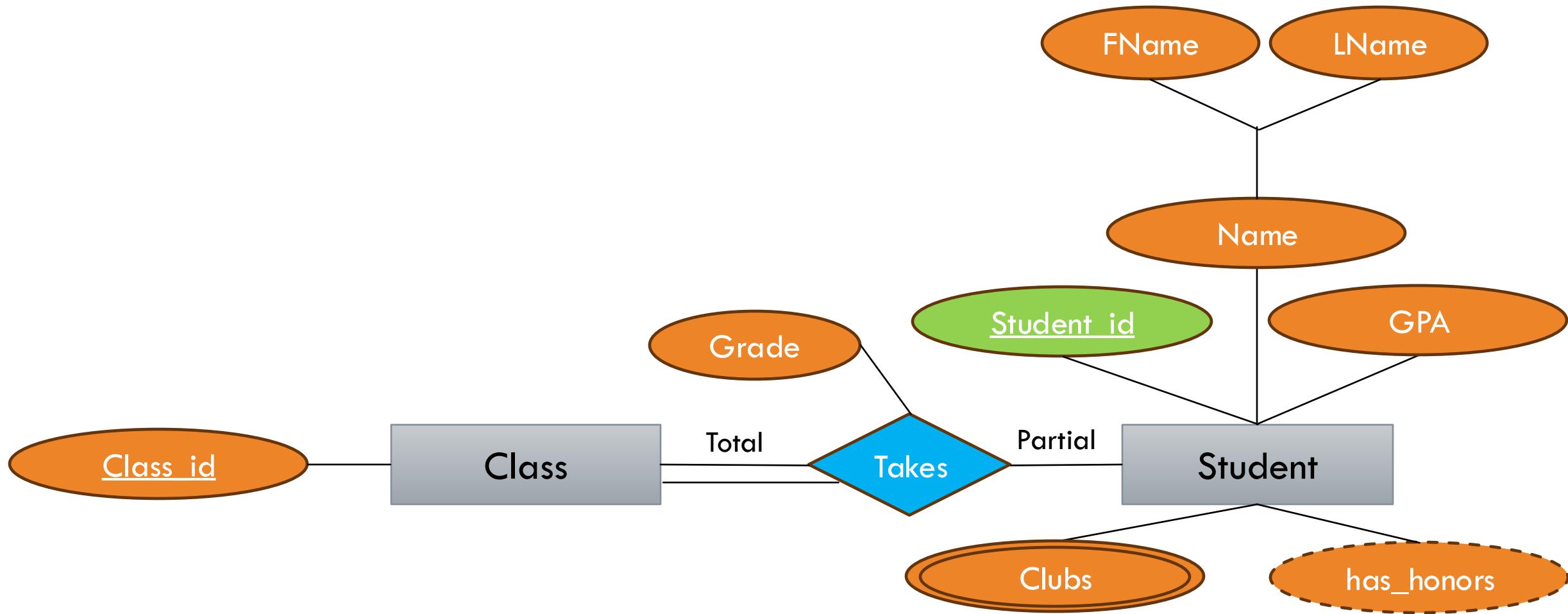
RELATIONSHIP ATTRIBUTE

关系属性

- 关于关系的属性
- 在这里，`take` 关系会为学生赋予成绩
- 成绩不存储在学生或课程实体中；只有当学生参加该课程时，成绩才会存在。

- An attribute about the relationship
- Here, take relationship will give student a grade
- Grade isn't stored on student or class entity; it is only if a student takes it.

RELATIONSHIP ATTRIBUTE



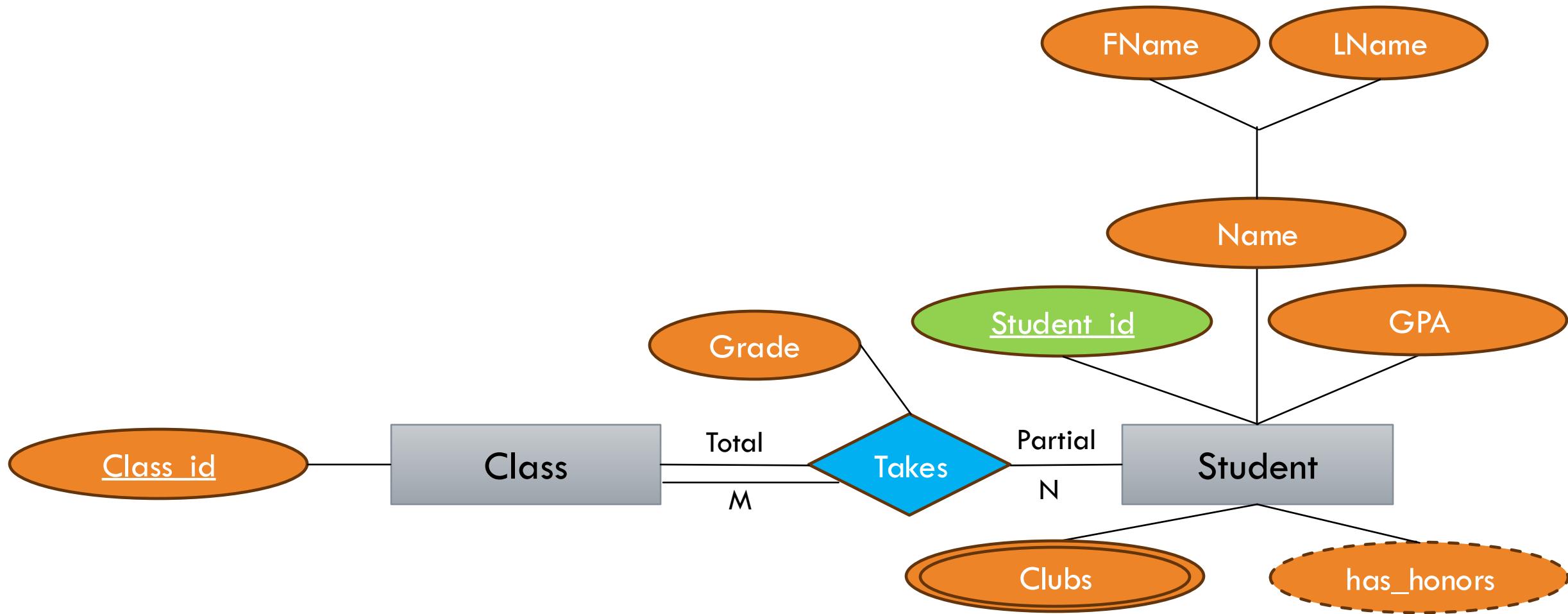
RELATIONSHIP CARDINALITY

关系基数

- 两个实体之间相关的行数
- 1:1 - 一名学生只能选修一门课程，且一门课程只能由一名学生选修
- 1:N - 一名学生可以选修一门课程，且一门课程可以被多名学生选修
- N:M - 一名学生可以选修多门课程，且一门课程可以被任意数量的学生选修
- 学生和课程之间是多对多关系

- Number of rows related between two entities
- 1:1 – A student can take one class, and class can be taken by one student
- 1:N – A student can take one class and a class can be taken by many students
- N: M – A student can take many classes and a class can be taken by any number any number of students
- Student and class are in many-to-many relationship

RELATIONSHIP CARDINALITY



WEAK ENTITY

弱实体

- 无法仅通过其属性唯一标识的实体
- 它依赖于另一个实体
- 例如，考试是一个实体，但考试无法独立存在，它必须与课程关联
- 用双线矩形表示

- An entity that cannot be uniquely identified by its attributes
- It is dependent on another entity
- Like, exam is an entity, but exam can't exist without a class, it must be associated with a class
- It's a double lined rectangle

IDENTIFYING RELATIONSHIP

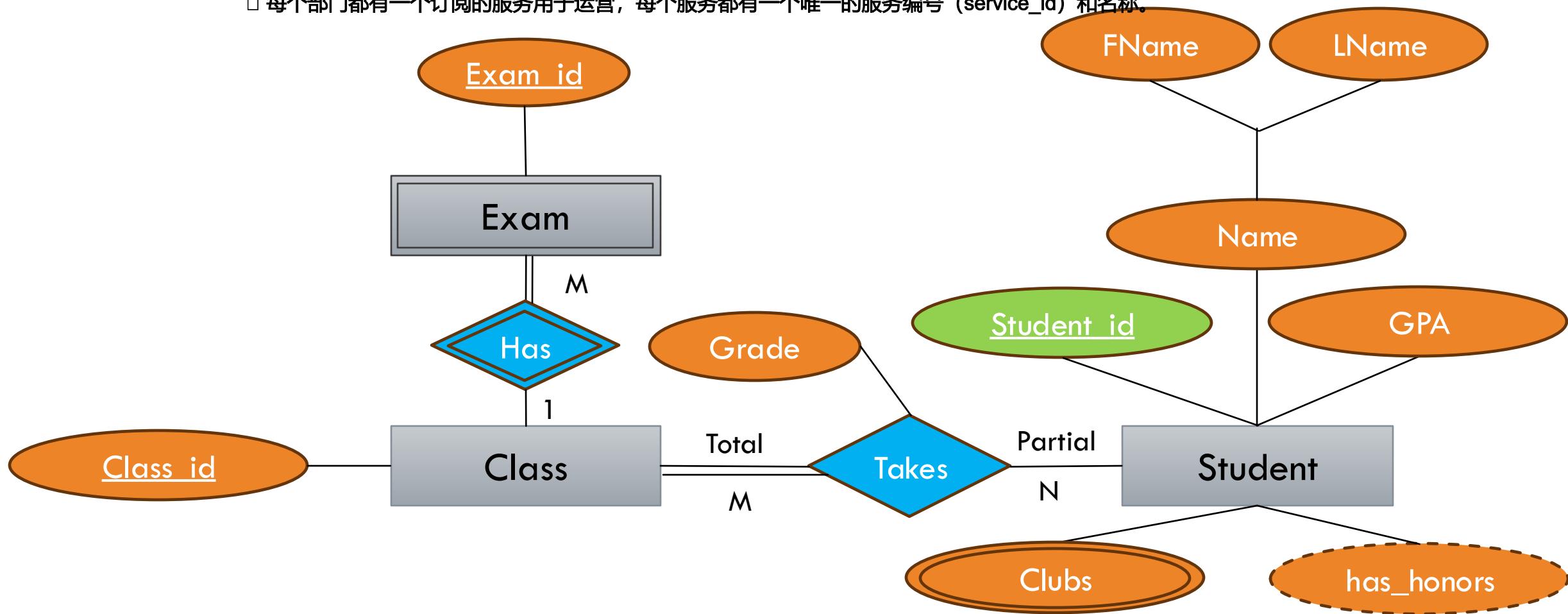
标识关系

- 用于唯一标识弱实体的关系
- 以双线菱形表示

- A relationship that serves to uniquely identify the weak entity
- It is a double lined diamond shape

业务需求

- 大学由多个部门组成。每个部门都有一个唯一的编号 (dept_id) 和名称。
- 每个部门有许多教师和学生。每个学生都有一个唯一的学号、姓名、出生日期和年龄来进行标识。每个教师通过姓名和工号来标识。
- 每个教师一次只在一个部门工作，每个部门由一名教师管理，我们需要跟踪其开始工作的日期。
- 每个部门都有一个订阅的服务用于运营，每个服务都有一个唯一的服务编号 (service_id) 和名称。



IMPORTANT NOTE

Whenever we have a weak entity and an identifying relationship,
the weak entity always must have a total participation in the
relationship.

Means all exams must have a class, but not all classes will have an
exam.

每当我们有一个弱实体和一个标识关系时，弱实体必须在该关系中有完全参与。

这意味着所有的考试必须有一个班级，但并非所有的班级都会有考试。



PRACTICE ER MODELLING

LET'S DO AN EXAMPLE!



INTRODUCTION

简介

- 文档描述了所有不同的信息片段以及它们之间的关系
- 通常是一段话
- 根据文档创建ER图/模型，然后将其转换为数据库
- 信息 -> ER图/模型 -> 数据库模式

- Document describing all different pieces of information and relation between them
- Usually a paragraph
- Make an ER diagram/Model from the document and then convert it into a database
- Information -> ER Diagram/Model -> Database Schema

BUSINESS REQUIREMENTS

The university system is structured into departments, each identified by a unique department number (dept_id) and name. Departments consist of multiple instructors and students. Students are uniquely identified by their student number, name, birth date, and age. Similarly, instructors are identified by their name and id. Each instructor is associated with only one department at a time, and every department is managed by a single instructor whose start date in the role must be tracked. Additionally, each department utilizes a subscribed service to support its operations, and these services are identified by a unique service_id and name.

业务需求

大学系统由多个部门组成，每个部门都有一个唯一的部门编号（dept_id）和名称。部门由多个教师和学生成组。学生通过学号、姓名、出生日期和年龄唯一标识。同样，教师通过姓名和工号进行标识。每个教师一次只关联一个部门，每个部门由一个主管教师管理，且必须跟踪该教师开始担任该职务的日期。此外，每个部门都使用订阅服务来支持其运营，这些服务通过唯一的服务编号（service_id）和名称进行标识。



BUSINESS REQUIREMENTS

- A university is organized into departments. Each department has a unique number (dept_id), name.
- Each department has many instructors and many students. Every student has a unique number, name, birth_date and age to identify it. Every instructor has name and id.
- An instructor can be working for only one department at a time and each department will be managed by one instructor working it and we want to keep track of start date for them.
- Each department has a subscribed service they use for functioning. Each service has a service_id and name.

业务需求

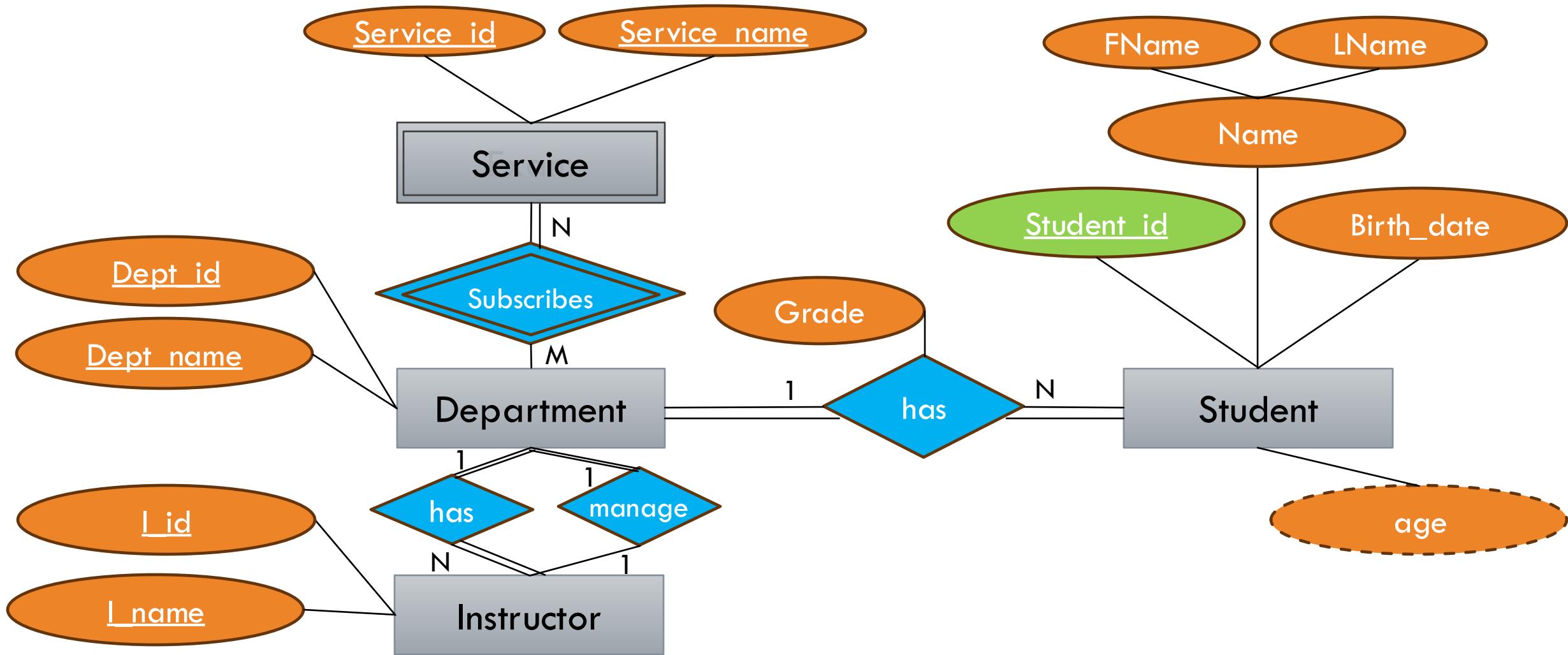
- 大学由多个部门组成。每个部门都有一个唯一的编号 (dept_id) 和名称。
- 每个部门有许多教师和学生。每个学生都有一个唯一的学号、姓名、出生日期和年龄来进行标识。每个教师通过姓名和工号来标识。
- 每个教师一次只在一个部门工作，每个部门由一名教师管理，我们需要跟踪其开始工作的日期。
- 每个部门都有一个订阅的服务用于运营，每个服务都有一个唯一的服务编号 (service_id) 和名称。



STEPS TO BUILD AN ER MODEL

- Identify the entities
- Identify the attributes for each entity
- Identify the primary key, derived attributes, etc.
- Identify relationship between each entity set

COMPLETE ER MODEL



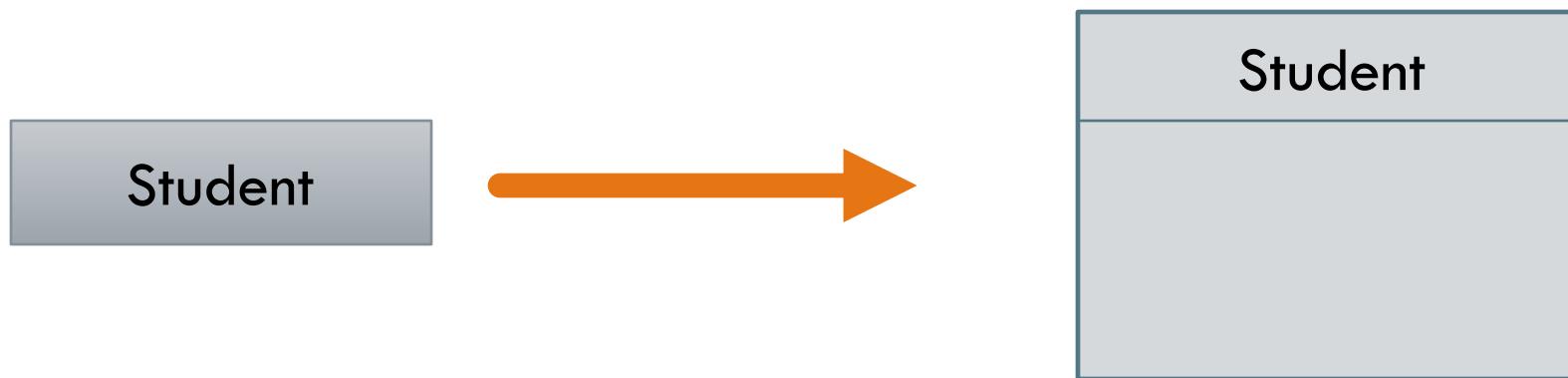


CONVERTING TO ER DIAGRAM

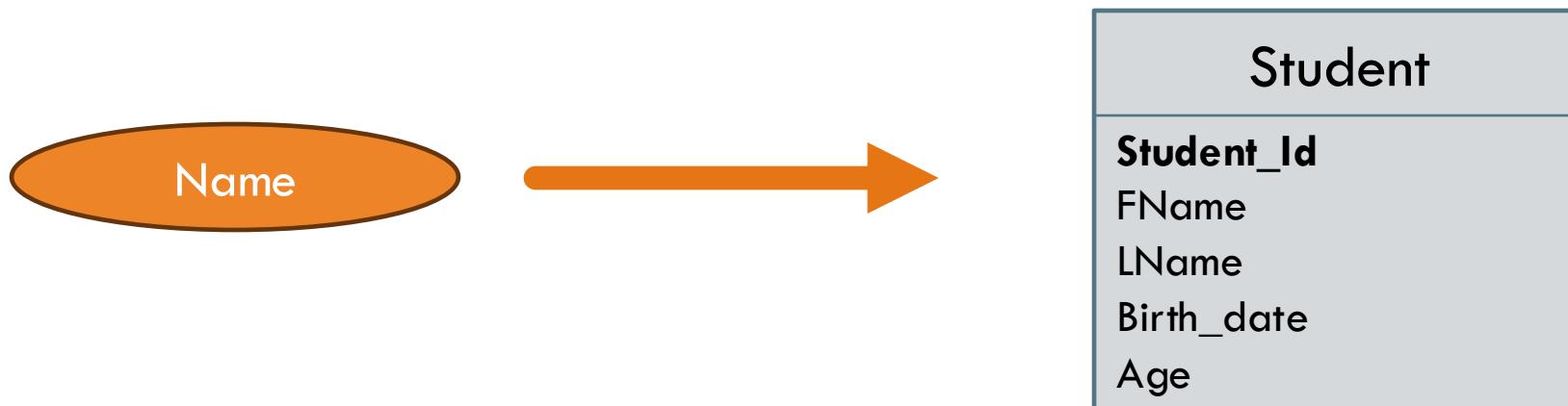
ER MODEL TO ER DIAGRAM



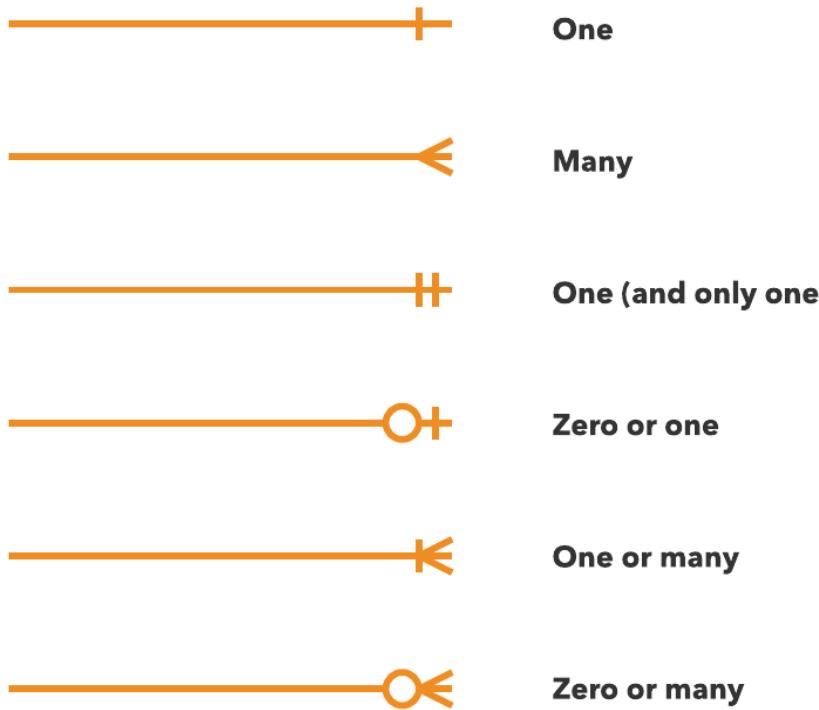
ENTITY



ATTRIBUTES



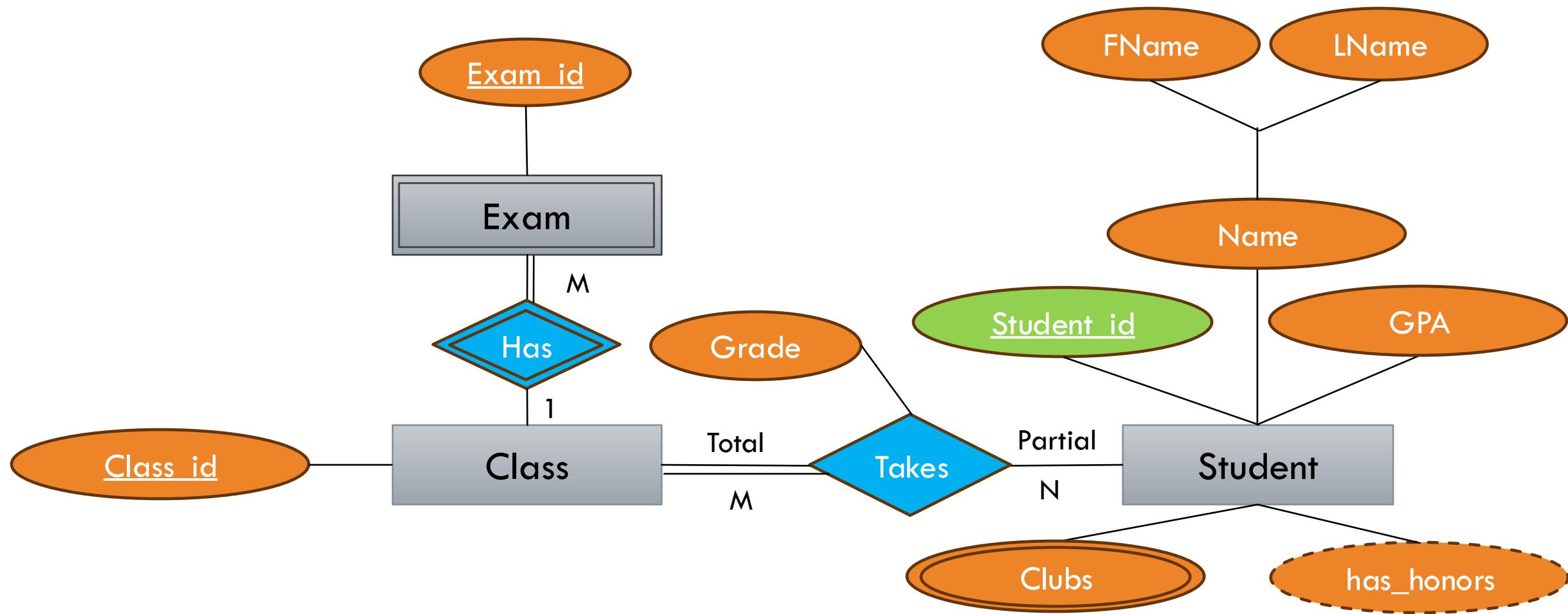
RELATIONSHIP CARDINALITY



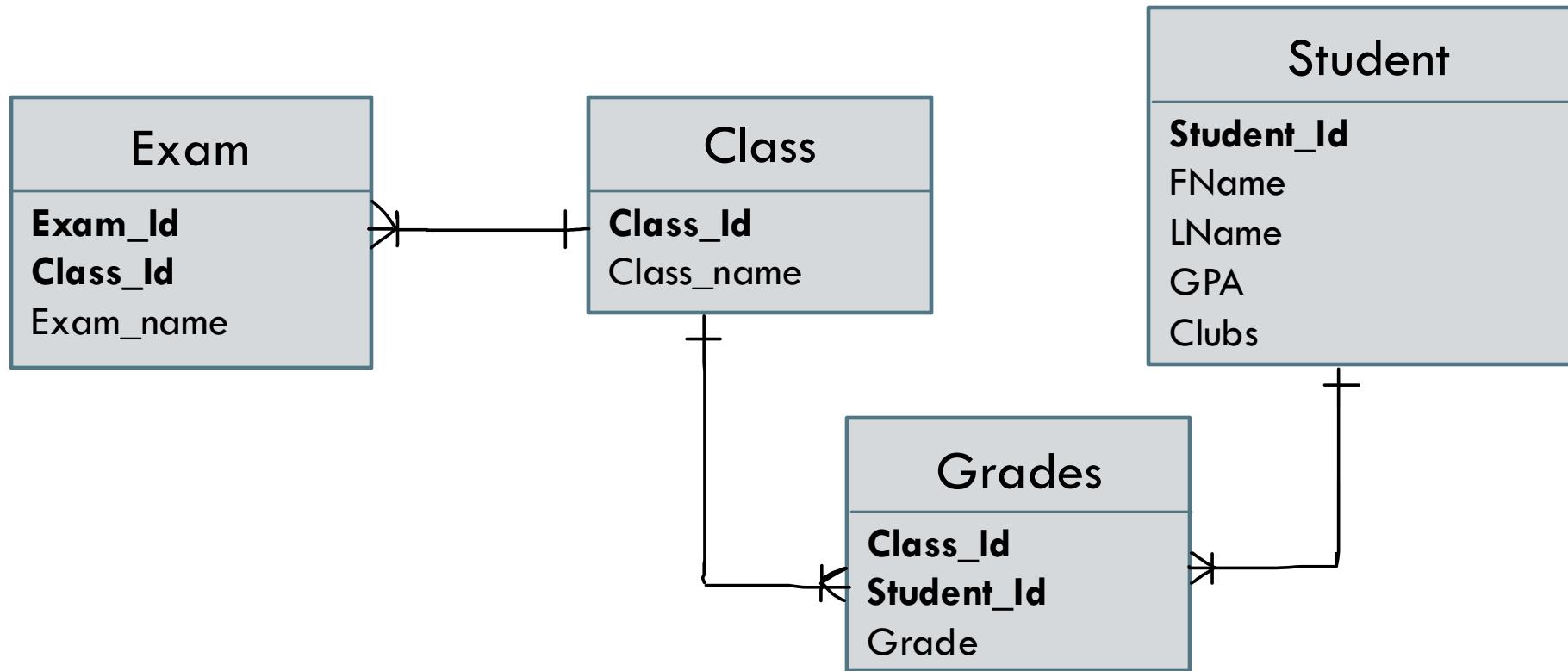
FOREIGN KEY

- 外键：用于描述两个表之间链接的关键字段。
- 它是您要链接到的另一个表的主键。
- 在 Exams 表中存储 class_id，以查找哪些课程有考试。

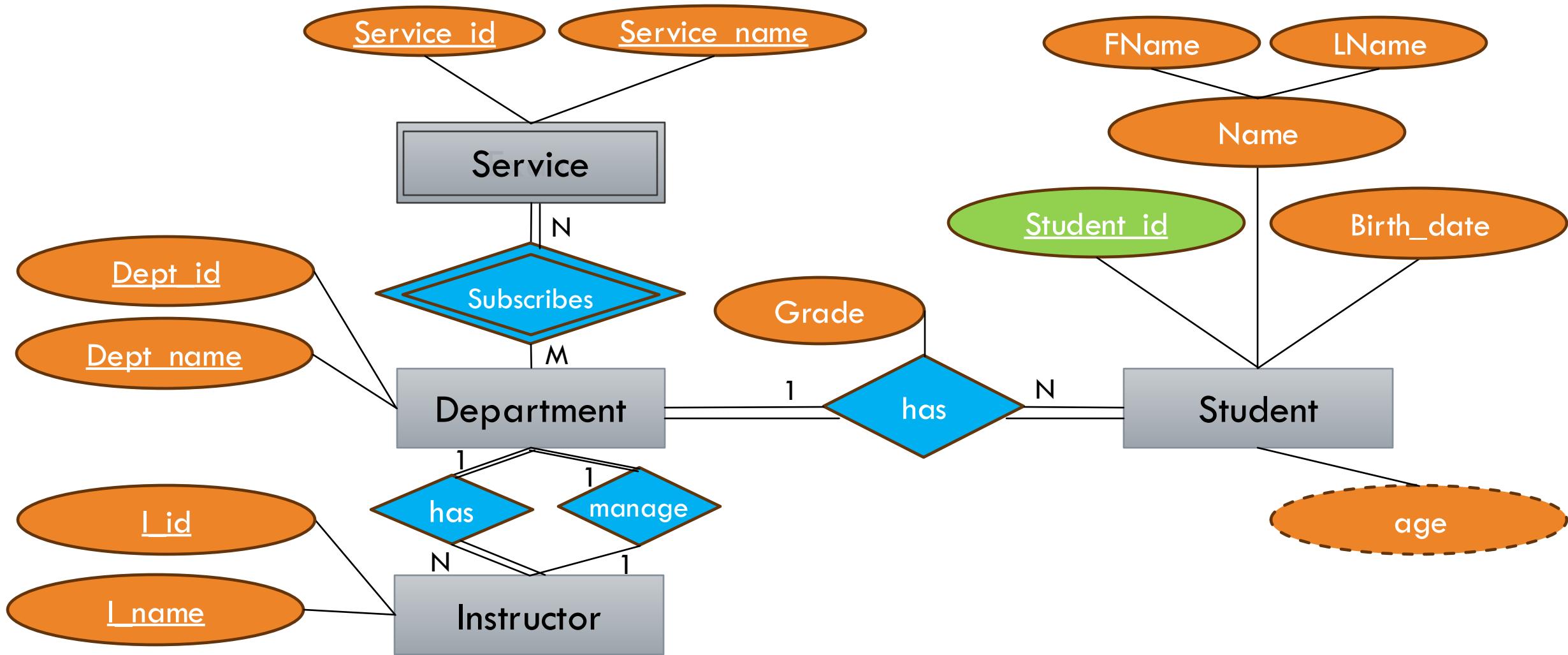
- Key to describe the link between 2 tables
- It is a primary key of another table that you are linking to
- Storing class_id in Exams table to find which classes have exams.



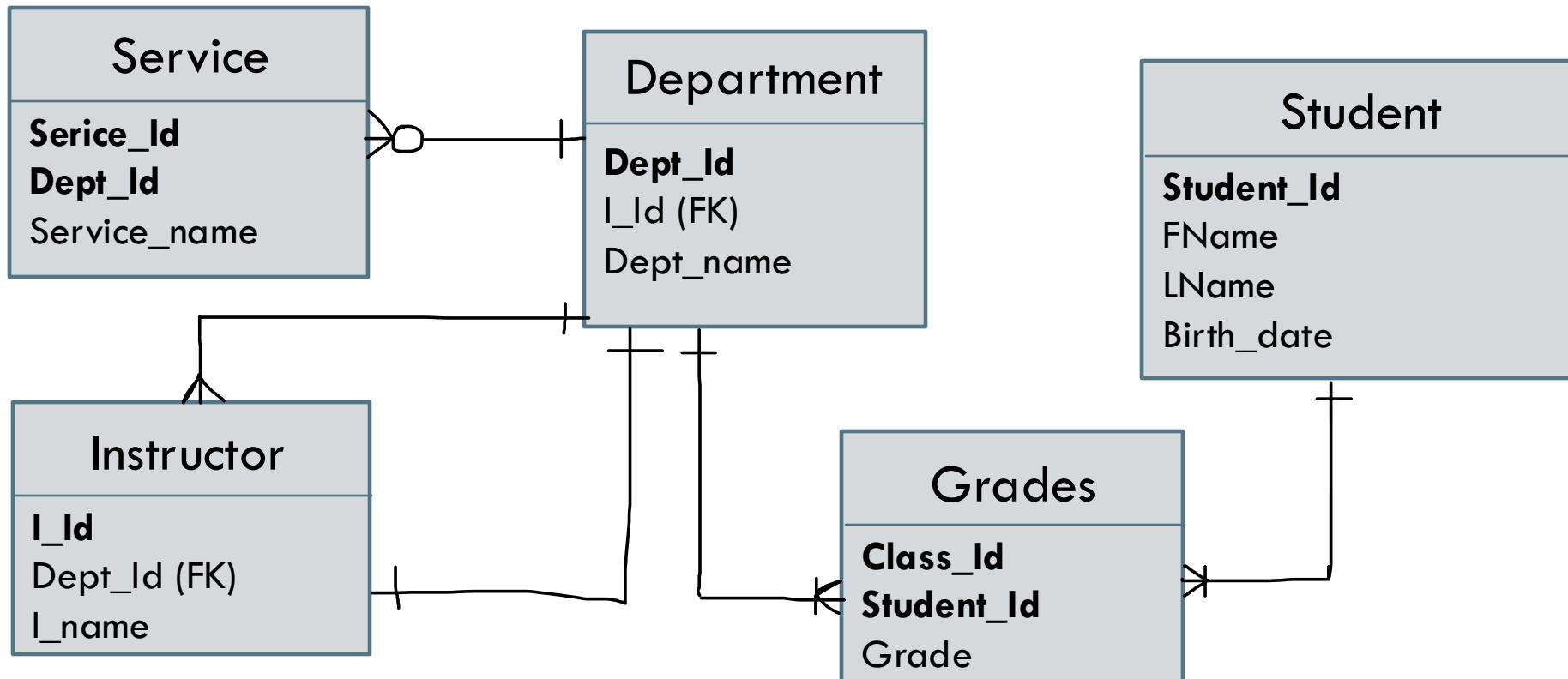
CONVERTING TO ER DIAGRAM



DESIGN ER DIAGRAM



UNIVERSITY ER DIAGRAM



SUMMARY

- ER模型包含实体、属性和关系。
- ER图/模型是业务需求和实际数据库模式实现之间的中间环节。
- 主键是用于唯一标识表中每一行的属性。
- 外键是用于将表中的值与另一个表使用其主键进行链接的属性。

- ER Model contains, entities, attributes and relationships.
- ER Diagram/Model is the middleman between business requirement and the actual implementation of the database schema.
- Primary Key is the unique attribute to identify every row in the table.
- Foreign key is an attribute that links the table values to another table using their primary key