DATABASE SYSTEMS

WEEK 5 LECTURE 1

Topics to Cover

- Introduction to ER Model and its constructs
  - Entity
  - Different types of Entities
  - Attribute and its different types
- Concept of Key and its importance
- Different types of keys

Data Model

- **Data models** define how **data** is connected to each other and how they are processed and stored inside the system.

ER Data Model

- The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases.

Constructs of ER Model

- The E-R data model supports following major constructs:
- Entity
- Attribute
- Relationship

The Entity

- Entity is basic building block of the E-R data model. The term entity is used in three different meanings or for three different terms and that are:
  - Entity type
  - Entity instance
  - Entity set

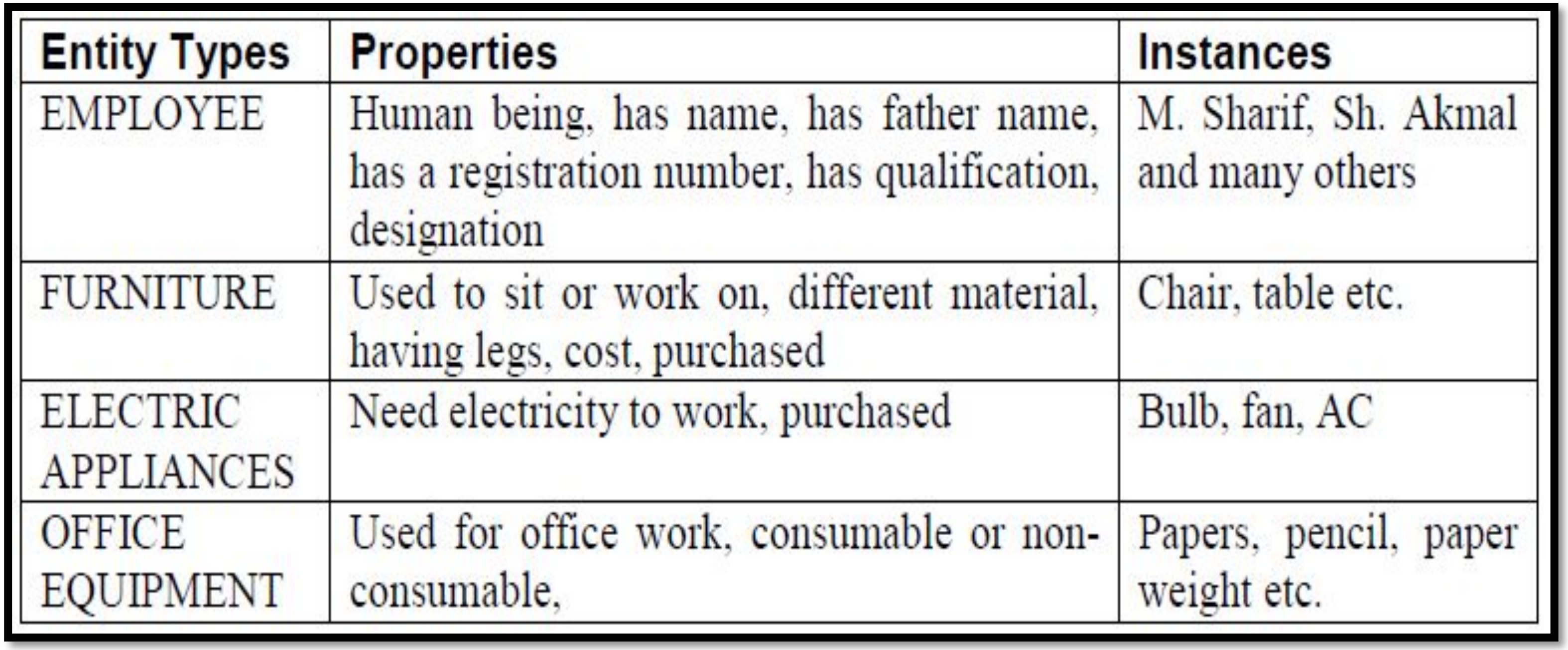
The Entity

- **Entity Type:** The entity type can be defined as a name/label assigned to items/objects that exist in an environment and that have similar properties.
- It could be person, place, event or even concept, that is, an entity type can be defined for physical as well as not-physical things.
- An entity type is distinguishable from other entity types based on properties and the same thing provides the basis for the identification of an entity type.

The Entity

- **Entity Instance:** A particular object belonging to a particular entity type.
- **Entity Set:** A group of entity instances of a particular entity type is called an entity set.

The Entity



Entity Types	Properties	Instances
EMPLOYEE	Human being, has name, has father name, has a registration number, has qualification, designation	M. Sharif, Sh. Akmal and many others
FURNITURE	Used to sit or work on, different material, having legs, cost, purchased	Chair, table etc.
ELECTRIC APPLIANCES	Need electricity to work, purchased	Bulb, fan, AC
OFFICE EQUIPMENT	Used for office work, consumable or non-consumable,	Papers, pencil, paper weight etc.

The Entity

- A particular object belonging to a particular entity type
  - Entity Type: Employee
  - Entity Instance: M. Sharif
  - Entity Set: All employees

Classification of Entity Types

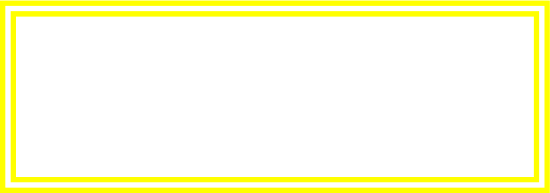
- Entity types (ETs) can be classified into two categories:
  - Regular/Strong ETs.
  - Weak ETs.

Classification of Entity Types

- **Weak Entity Types:** An entity type whose instances cannot exist without being linked with instances of some other entity type, i.e., they cannot exist independently.
- For example, in an organization we want to maintain data about the vehicles owned by the employees.
- Now a particular vehicle can exist in this organization only if the owner already exists there as employee.

Classification of Entity Types

- **Weak Entity Types:** Similarly, if employee leaves the job and the organization decides to delete the record of the employee then the record of the vehicle will also be deleted since it cannot exist without being linked to an instance of employee.
- Represented by:

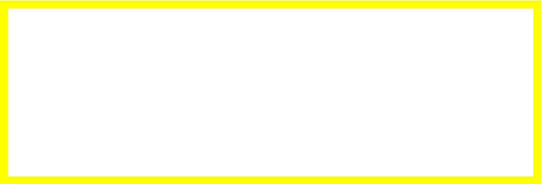


Classification of Entity Types

- **Regular/Strong Entity Types:** An entity type whose instances can exist independently, that is, without being linked to the instances of any other entity type is called strong entity type.
- A major property of the strong entity types is that they have their own identification, which is not always the case with weak entity types.
- For example, employee in the previous example is an independent or strong entity type, since its instances can exist independently.

Classification of Entity Types

- **Regular/Strong Entity Types are represented by:**



Naming Entity Types

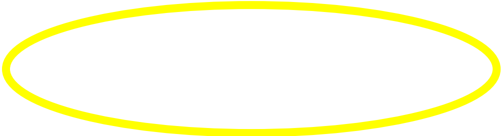
- Following are some recommendations for naming entity types. But they are just recommendations; practices considered good in general:
- Singular noun recommended.
- Organization specific names, like customer, client anything will work.
- Write in capitals.
- Abbreviations can be used, be consistent.

Attribute

- An attribute of an entity type is a defining property or quality of the instances of that entity type.
- Entity instances of same entity type have the same attributes. (E.g. Student Identification, Student Name).
- However, values of these attributes may be same or different. For example, all instances of the entity type STUDENT may have the attributes name, father name, age; but the values against each of these attributes for each instance may be different.

Attribute

- An attribute is any detail that serves to identify, qualify, classify, quantify, or otherwise express the state of an entity occurrence or a relationship.
- One entity type cannot have two attributes with the same name.
- Represented by:



Domain of an Attribute

- Every attribute has a name and a domain.
- Domain is the set of possible values that an attribute can have, that is, we specify a set of values either in the form of a range or some discrete values, and then attribute can have value out of those values.
- Domain is a form of a check or a constraint on attribute that it cannot have a value outside this set.

Domain of an Attribute

- For example, if we define a salary attribute of EMPLOYEE entity type to hold the salary of employees, the value assigned to this attribute should be numeric, it should not be assigned a value like ‘Ali’, or ‘10/10/2004.
- Further, even if we have declared it as numeric it will have numeric values, but about a value like 10000000000.
- It means not only you will specify that the value of salary will be numeric but also associate a range, a lower and upper limit.

Domain of an Attribute

- Domain is normally defined in form of data type and some additional constraints like the range constraint.
- Another important thing that needs to be mentioned here is that once we associate a domain to an attribute, all the attributes in all entity instances of that entity type will have the values from the same domain.

Types of Attributes

- Attributes may be of different types. They may be:
  - Simple or Composite
  - Single valued or multi-valued
  - Stored or Derived

Types of Attributes

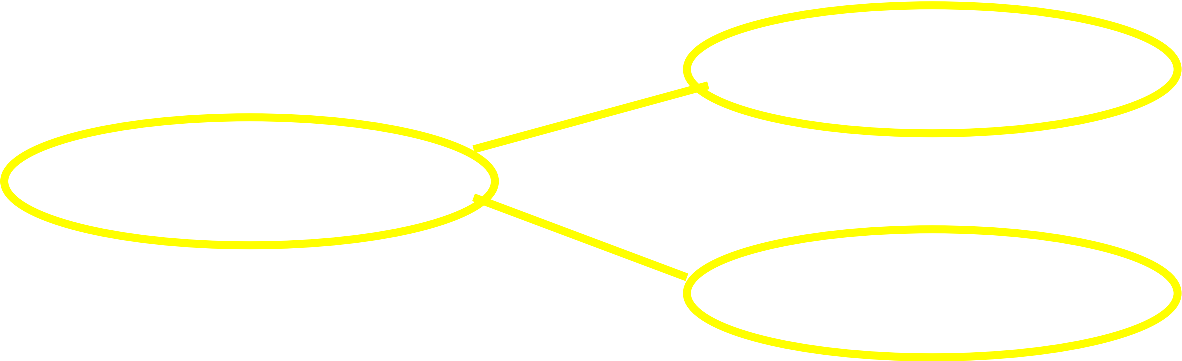
- Attributes may be of different types. They may be:
  - Simple or Composite
  - Single valued or multi-valued
  - Stored or Derived

Types of Attributes

- **SIMPLE ATTRIBUTE:** An attribute that is a single whole is a simple attribute. The value of a simple attribute is considered as a whole.
- Not as comprising of other attributes or components.
- For example, attributes stName, stFatherName, stDateOfBorth of an entity type STUDENT are example of simple attributes.

Types of Attributes

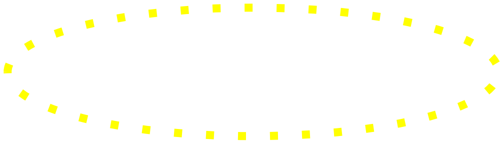
- **COMPOSITE ATTRIBUTE:** If an attribute consists of collection of other simple then it is called a composite attributes.
- For example, stAdres attribute may comprise of houseNo, streetNo, areaCode, city etc. In this case stAdres will be a composite attribute.
- Represented By:



Types of Attributes

- **STORED ATTRIBUTE:** Normally attributes are stored attributes, that is, their values are stored and accessed as such from the database.
- **DERIVED ATTRIBUTE:** However, sometimes attributes' values are not stored as such, rather they are computed or derived based on some other value.
- For example, we may store the name, father name, address of employees, but age can be computed from date of birth.
- Represented by:
- Doted line

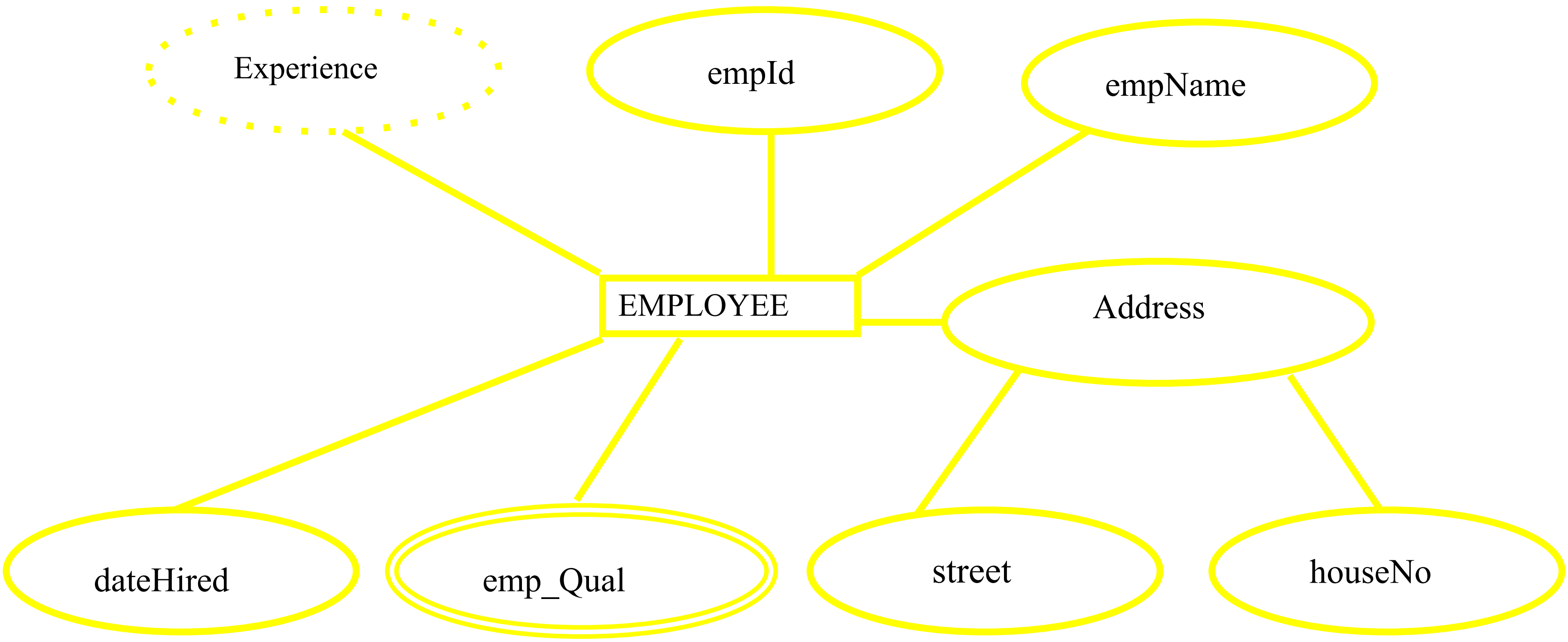
Derived



ACTIVITY

- Consider Employee as an entity type and associate different types of attributes with the entity type of Employee.
- Take a look into the system where you work or study or live, identify different entity types in that environment. Associate different types of attributes with these entity types.

Solution


Experience
empId
empName
EMPLOYEE
Address
dateHired
emp_Qual
street
houseNo

Keys

- A key is a set of attributes that can be used to identify or access a particular entity instance of an entity type.
- An entity type may have many instances, from a few to several thousands and even more.
- Now out of many instances, when and if we want to pick a particular/single instance, and many times we do need it, then key is the solution.

Keys

Following are the major types of keys:

- Candidate Key
- Primary Key
- Super Key
- Alternate Key
- Secondary Key
- Foreign Key

Candidate Key

- A Candidate Key can be any column or a combination of columns that can qualify as unique key in database. There can be multiple Candidate Keys in one table. Each Candidate Key can qualify as Primary Key.
- For Example Employee ID and Social Security Number (SSN) is a candidate key as both of these attributes can uniquely identify an instance of an entity type **EMPLOYEE.**

Primary Key

- A candidate key chosen by the database designer to act as key is the primary key.
- The columns you choose to maintain uniqueness in a table.
- Here in Employee table you can choose either Employee ID or SSN columns, Employee ID is preferable choice, as SSN is a secure value.

Super Key

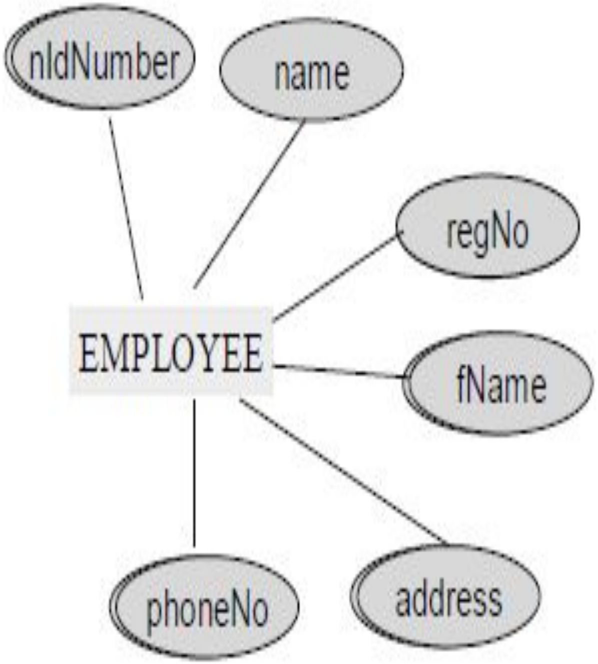
- **Super Key:** A super key is a set of one or more attributes which taken collectively, allow us to identify uniquely an entity instance in the entity set.
- If you add any other column/attribute to a Primary Key then it become a super key, like Employee ID + Full Name is a Super Key.

Super Key

- Once specific characteristic with super key is that, as per its definition any combination of attributes with the key is a super key.
- For example, consider the entity type STUDENT with attributes registration number, name, father name, address, phone, class, admission date. Now which attribute can we use that can uniquely identify any instance of STUDENT entity type.
- All the combinations “regNo, name”, “regNo, fName, address”, “name, fName, regNo” and many others, all are super keys.

Alternate Keys

- Candidate keys which are not chosen as the primary key are known as alternate keys.
- For example, we have two candidate keys of Employee in figure , regNo and nIdNumber, if we select regNo as PK then the nIdNumber will be alternate key.

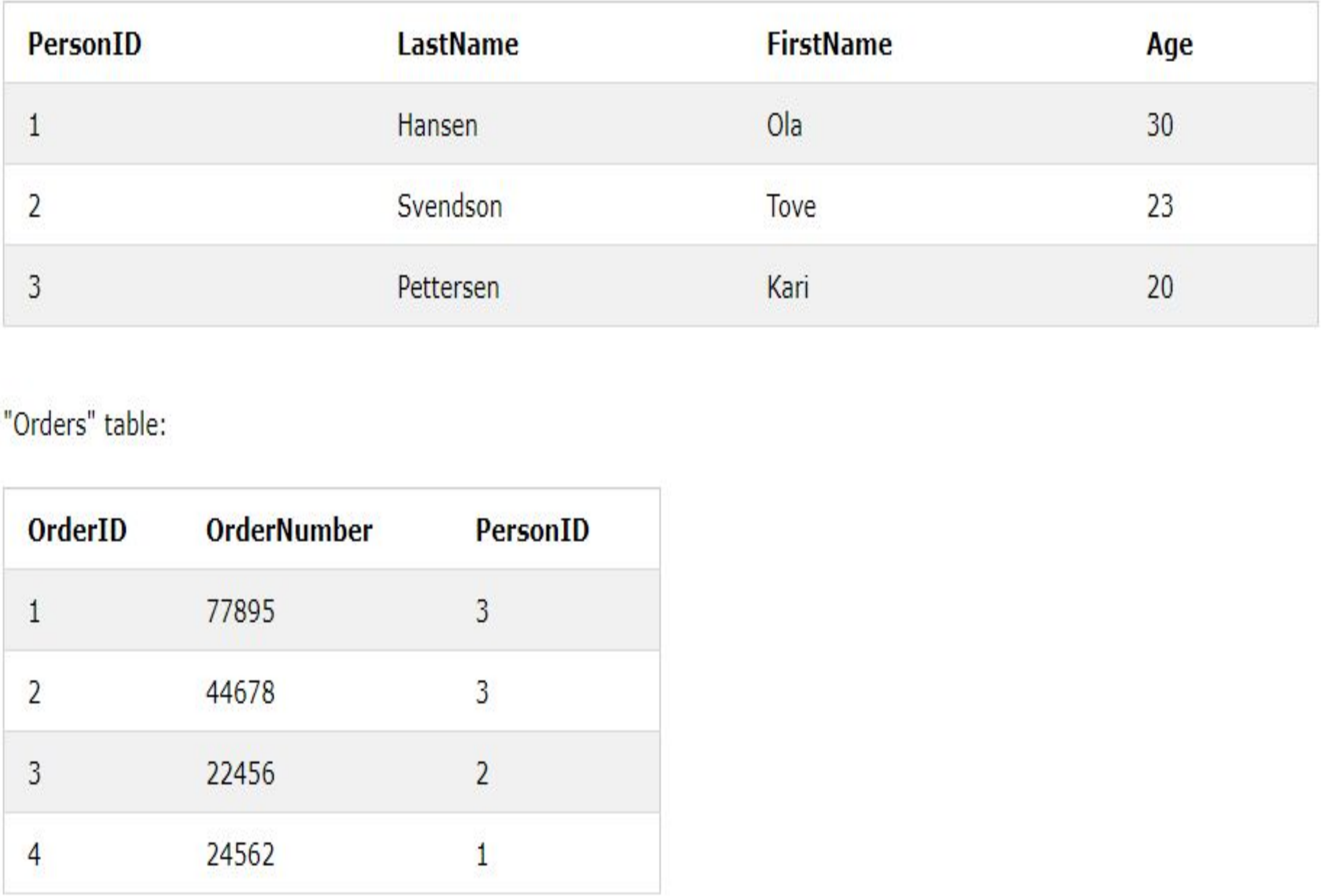


Secondary Key

- The attributes on which we need to access the instances of an entity type that may not necessarily return unique instance is called the secondary key (Using non-key attributes) .
- For example, we want to see how many of our students belong to Multan, in that case we will access those instances of the STUDENT entity type that contain “Multan” in their address. In this case address will be called secondary key.
- Keep one thing in mind here, that a particular access on the value of a secondary key MAY return a single instance, but that will be considered as chance or due to that particular state of entity set.

Foreign Key

- A FOREIGN KEY is a key used to link two tables together.
- A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.
- The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.



PersonID	LastName	FirstName	Age
1	Hansen	Ola	30
2	Svendson	Tove	23
3	Pettersen	Kari	20

"Orders" table:

OrderID	OrderNumber	PersonID
1	77895	3
2	44678	3
3	22456	2
4	24562	1

Next Lecture

- Relationships in E-R Data Model
- Types of Relationships