Subject:
(310242)
Database Management Systems
(DBMS)

Scheme

Theory: DBMS

- **Teaching Scheme:** Lectures 3 Hrs/Week
- **Examination Scheme:**
  - ➢In Semester Assessment: 30 Marks
  - ➢End Semester Assessment: 70 Marks

Practical: DBMS Laboratory

- **Teaching Scheme:** Practical: 4 Hrs/Week
- **Examination Scheme:**
  - ➢Practical : 50 Marks
  - ➢Term Work: 25 Marks

Course Objectives:

- To understand the fundamental concepts of database management.

These concepts include aspects of database design, database languages, and database-system implementation

- To use Structured Query Language (SQL) and design relational database.

These concepts helps us to design good database and tell us how to handle it.

- To illustrate the concept of Transaction , Query processing and various Database Architectures.

These concepts explain various database architectures and query processing.

- To use scalable general purpose databases to handle big data.

These concepts explain how to handle big data.

Course Outcomes:

On completion of the course, student will be able to–

- summarize fundamental concepts of database management.
- apply SQL for Relational database management system and normalization techniques to normalize the database .
- analyze transaction management and classify various Database Architectures
- apply non-relational database techniques for storing and processing large volumes of unstructured data.

Contents of DBMS

- **Unit I:** Introduction to Databases: 07 Hrs
- **Unit II:** SQL and PL/SQL: 07 Hrs
- **Unit III:** Relational Database Design: 08 Hrs
- **Unit IV:** Database Transactions and Query Processing: 08Hrs
- **Unit V:** Parallel and Distributed Databases: 07Hrs
- **Unit VI:** NoSQL Database: 08Hrs

Books

Text Books:

- Abraham Silberschatz, Henry Korth, S.Sudarshan,"Database System concepts",5th Edition ,McGraw Hill International Edition.
- Connally T, Begg C., "Database Systems", Pearson Education, ISBN 81-7808-861-4
- Pramod J. Sadalage and Martin Fowler, “NoSQL Distilled”, Addison Wesley, ISBN-

10: 0321826620, ISBN-13: 978-0321826626

Reference Books:

1.C J Date, “An Introduction to Database Systems”, Addison-Wesley, ISBN: 0201144719

2.S.K.Singh, “Database Systems : Concepts, Design and Application”, Pearson, Education,ISBN 978-81-317-6092-5

3.Kristina Chodorow, Michael Dirolf, “MangoDB: The Definitive Guide” ,O’Reilly Publications, ISBN: 978-1-449-34468-9.

4.Adam Fowler, “NoSQL For Dummies”, John Wiley & Sons, ISBN-1118905628

5. Kevin Roebuck, “Storing and Managing Big Data - NoSQL, HADOOP and More”, Emereopty Limited, ISBN: 1743045743, 9781743045749

6. Joy A. Kreibich, “Using SQLite”, O'REILLY, ISBN: 13:978-93-5110-934-1

7. Garrett Grolemund, “Hands-on Programming with R”, O'REILLY, ISBN : 13:978-93-5110-728-6

Unit I

Introduction to Databases

Definition – Data

- **Data** is raw, unorganized facts that need to be processed.
- Data can be something simple and seemingly random and useless until it is organized.
- **Example:** Each student's test score is one piece of data.
- "Data" comes from a singular Latin word, datum, which originally meant "something given."
- Over time "data" has become the plural of datum.

Definition–Information

- When data is processed, organized, structured or presented in a given context so as to make it useful, it is called **information.**
- Information is the **processed data** on which decisions and actions are based.
- **Example:** The average score of a class or of the entire school is information that can be derived from the given data.

Definition – DBMS

- A **database** is an organized collection of data. It is the collection of tables, queries, reports, views and other objects.
- Database contains information's relevant to enterprise.
- **DBMS(Database Management System)** is a collection of interrelated data and a set of programs to access those data.
- It provides a way to store and retrieve database information in convenient and efficient manner.

Definition – DBMS Contd…

- **Management** of data involves:
  - Defining structures for storage of information
  - Providing mechanisms for the manipulation of information.
  - Ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.
  - If data are to be shared among several users, the system must avoid possible anomalous results.

- Well-known DBMSs include MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase and IBM DB2.

Application of DBMS

- Banking
- Airlines
- Universities
- Credit card transactions
- Telecommunication
- Finance
- Sales
- Manufacturing
- Human resources

Purpose of Database Systems
File-Processing System

- A **File processing system** is a collection of **files** and programs that access/modify these **files**.
- Typically, new **files** and programs are added over time (by different programmers) as new information needs to be stored and new ways to access information are needed.

Disadvantages of File Processing System

- Data redundancy and inconsistency
- Difficulty in accessing data
- Data isolation
- Integrity problems – consistency constraint
- Atomicity problems – atomic
- Concurrent-access anomalies
- Security problems – data abstraction

Advantage of DBMS

- Controlling Data Redundancy
- Sharing of Data
- Data Consistency
- Data Integrity
- Data Security
- Data Independence
- Backup and Recovery Procedures

Disadvantage of DBMS

- Increased costs
- Management complexity
- Frequent upgrade/replacement cycles
- Database Damage

View of Data

- Major purpose of a database system is to provide users with an *abstract* view of the data.

Data Abstraction

- Data abstraction is the reduction of a particular body of data to a simplified representation of the whole.

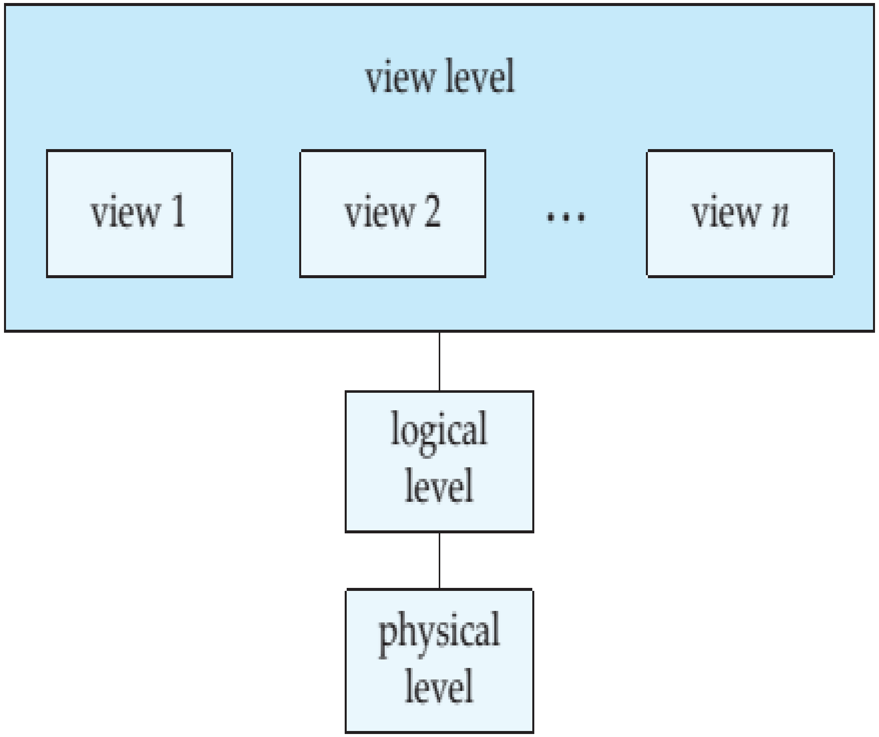


Fig: Three Level of data abstractions

View of Data Contd…

- **Data Abstraction**

- Physical level (Low Level)
  - ✓ How data are actually stored
  - ✓ Describe Complex low-level data structure
  - ✓ E.g. index, B-tree, hashing.
- Logical Level(Conceptual Level) (Middle Level)
  - ✓ What data are stored and what relationship exist among those data
- View Level (External Level) (High Level)
  - ✓ Describe only part of the entire database
  - ✓ E.g. tellers in a bank get a view of customer accounts, but not of payroll data.

View of Data Contd…

```
type instructor= record
                ID : char(5);
                name : char(20);
                deptname : char(20);
                salary : numeric(8,2);
                end;
```

This code defines a new record type called *instructor* with four fields.

- At the physical level, an *instructor* described as a block of consecutive storage locations. And compiler hide this level of details from programmers.
- At the logical level, each such record inserted by user.
- At the view level, computer users see a set of application programs that hide details of the data types.

Instances and Schemas

- The data stored in the database at any given time is an **Instance** of the database
- The overall design of the database is called the **database schema.**

Name	Account No	Balance	Address
Bob	102	1000	Mumbai
John	301	500	Chennai

- is an instance of a database with schema (Name, Account No, Balance, Address

Instances and Schemas

- **Database systems have schemas at each level of abstraction:**

- The physical schema describes the database design at the physical level

  i.e. as a file of records of a particular type

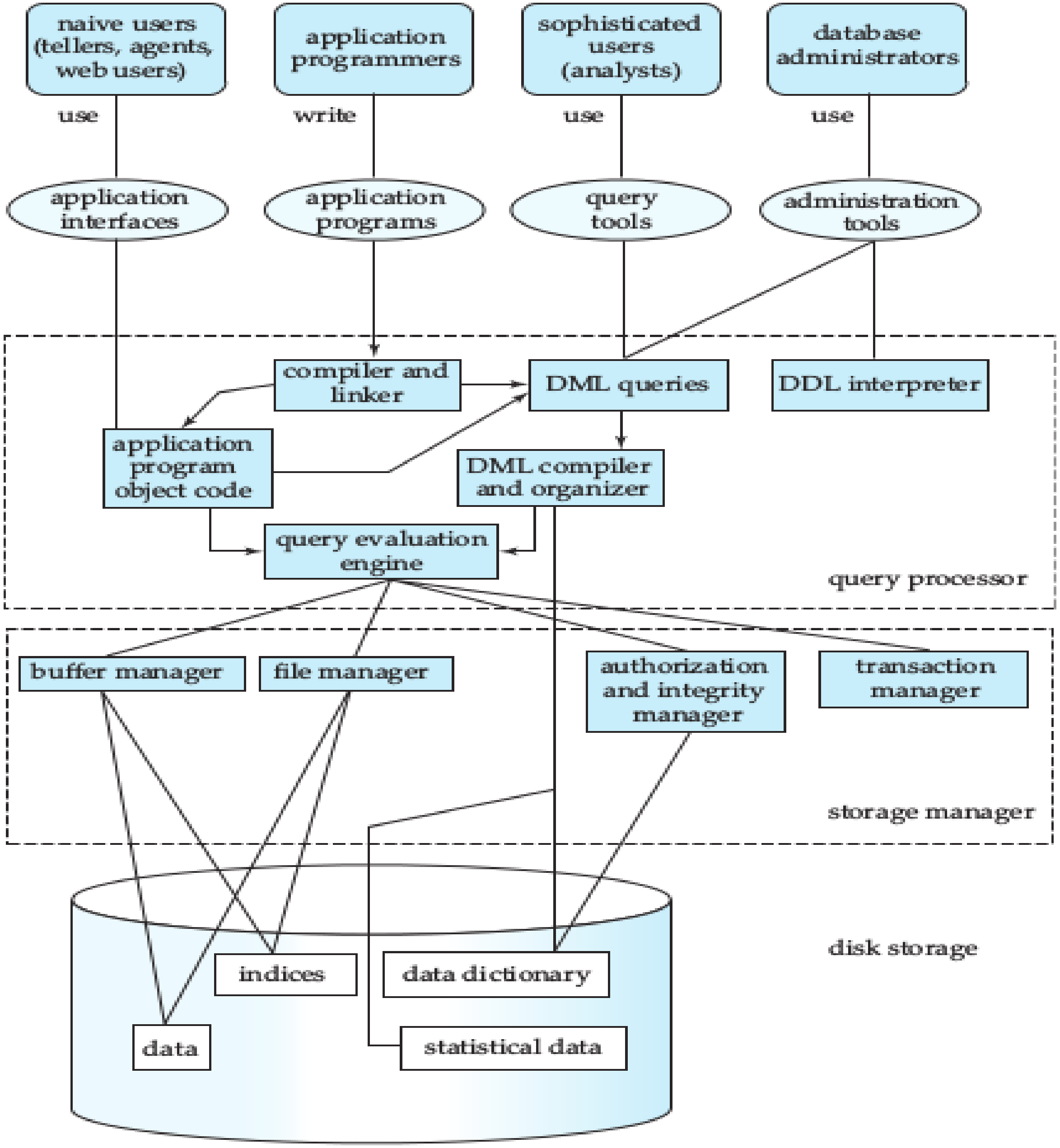
- The logical schema describes the database design at the logical level.

  Example: (Name, Account No, Balance, Address)

- A database may also have several schema's at the view level, sometimes called subschemas, that describe different views of the database.

  For example, (Name, Account No) is a subschema of (Name, Account No, Balance, Address)

Database System Structure



Functions of DBA

- Schema Definition.
- Storage structure and access method definition.
- Schema and physical organization modification.
- Granting of authorization for data access.
- Routine maintenance.

Data Models

- **Data Model** is a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.

- A data model provides a way to describe the design of a database at the **physical, logical, and view levels.**

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

Types of Data Models

1) **Record-based Data Models**

   The Relational Model

   The Network Model

   The Hierarchical Model

1) **Object-based Data Models**

   The E-R Model

   The Object-Oriented Model

1) **Physical Data Models**

Note: *1st & 2nd model describe data at the conceptual and view levels and 3rd at physical level*

Relational Model

- The Relational model uses a collection of tables to represent both data and the relationships among those data.
- Tables are also known as relations.
- Relation: made up of 2 parts:

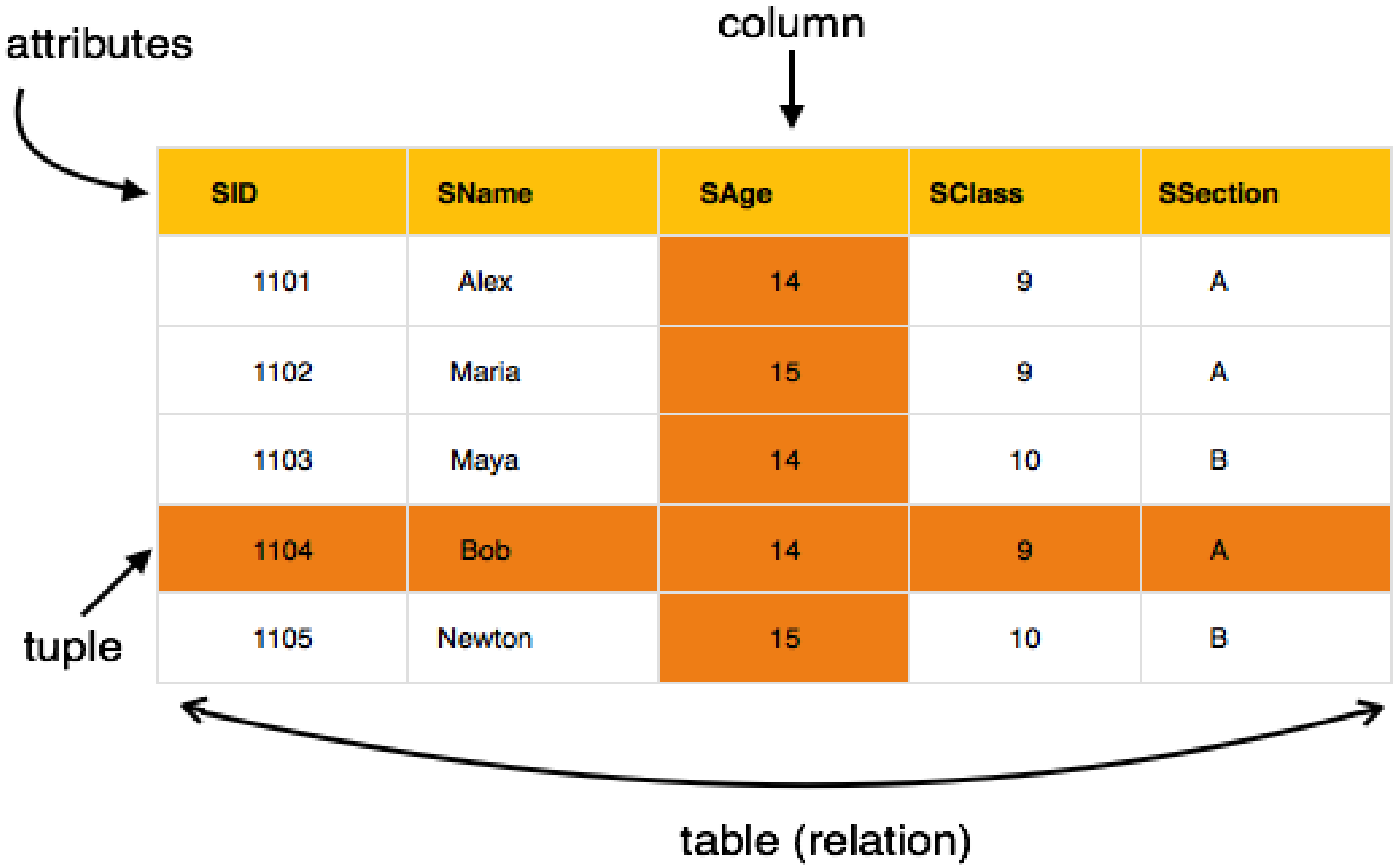
➢ Instance: a table, with rows and columns.

#rows =cardinality , #fields = degree / arity

➢ Schema: specifies name of relation, plus name and type of each column

E.g.: Students(*sid:* string, *name:* string, *login:* string, *age*: integer, *gpa:* real)

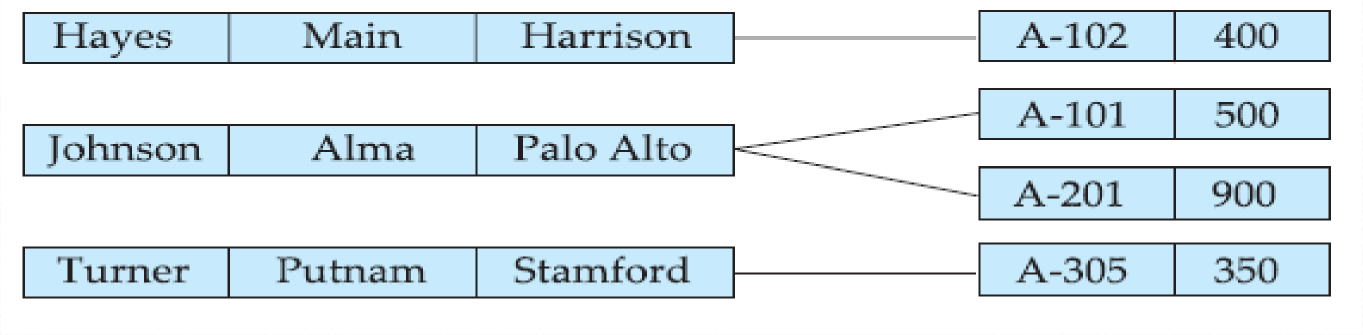
Relational Model Contd…



SID	SName	SAge	SClass	SSection
1101	Alex	14	9	A
1102	Maria	15	9	A
1103	Maya	14	10	B
1104	Bob	14	9	A
1105	Newton	15	10	B

Network Model

- In **Network Model** data are represented by collections of **records**, and relationships among data are represented by **links.**
- Each **record** is a collection of fields (attributes), each of which contains only one data value.
- A **link** is an association between precisely two records



Hayes Main Harrison

Johnson Alma Palo Alto

Turner Putnam Stamford

A-102 400

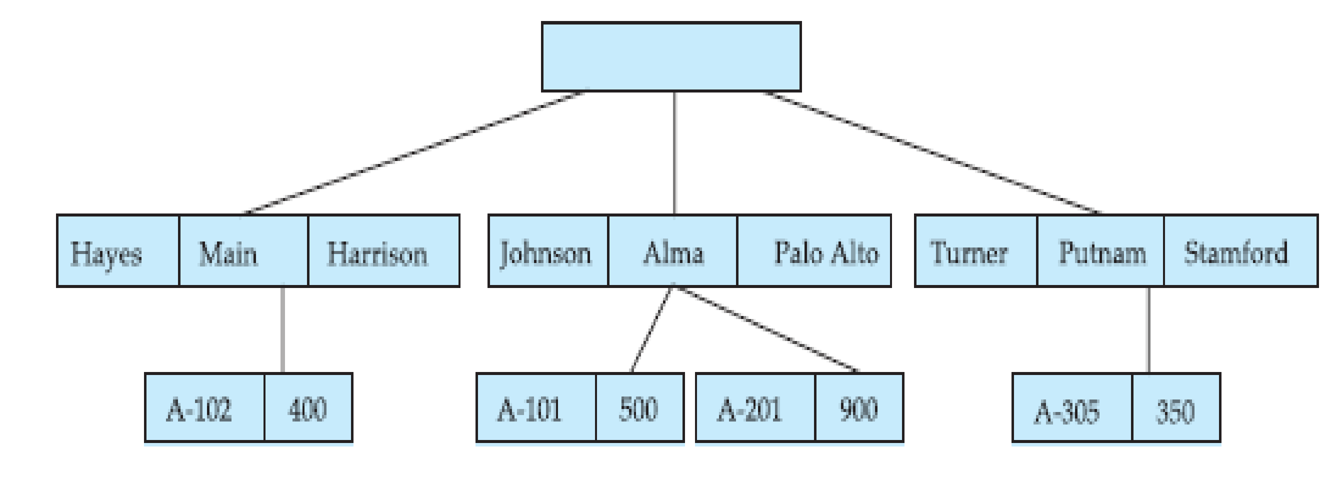
A-101 500

A-201 900

A-305 350

Hierarchical Model

- A **Hierarchical Model** consists of a collection of records that are connected to each other through links.
- A **record** is similar to a record in the network model.
- Each **record** is a collection of fields (attributes), each of which contains only one data value.
- A **link** is an association between precisely two records



Hierarchical Database Model

- The **Hierarchical Model** mandates that each child record has only one parent, whereas each parent record can have one or more child records.

- The relationships formed in the tree-structure diagram must be such that only one-to-many or one-to-one relationships exist between a parent and a child.

- In order to retrieve data from a hierarchical database the whole tree needs to be traversed starting from the root node.

Entity-Relationship(E-R) Model

- The entity-relationship (E-R) data model uses a collection of basic objects, called entities, and relationships among these objects.

- E-R Diagram is a visual representation of data, that describes how data is related to each other.

- **Entity** – An entity in an ER Model is a “thing” or “object ” in the real-world having properties called **attributes**.

E-R Model Contd…

- An **Entity set** is a set of entities of the same type that share the same properties, or attributes.

- **Attributes-** Entities are represented by means of their properties, called attributes. All attributes have values.

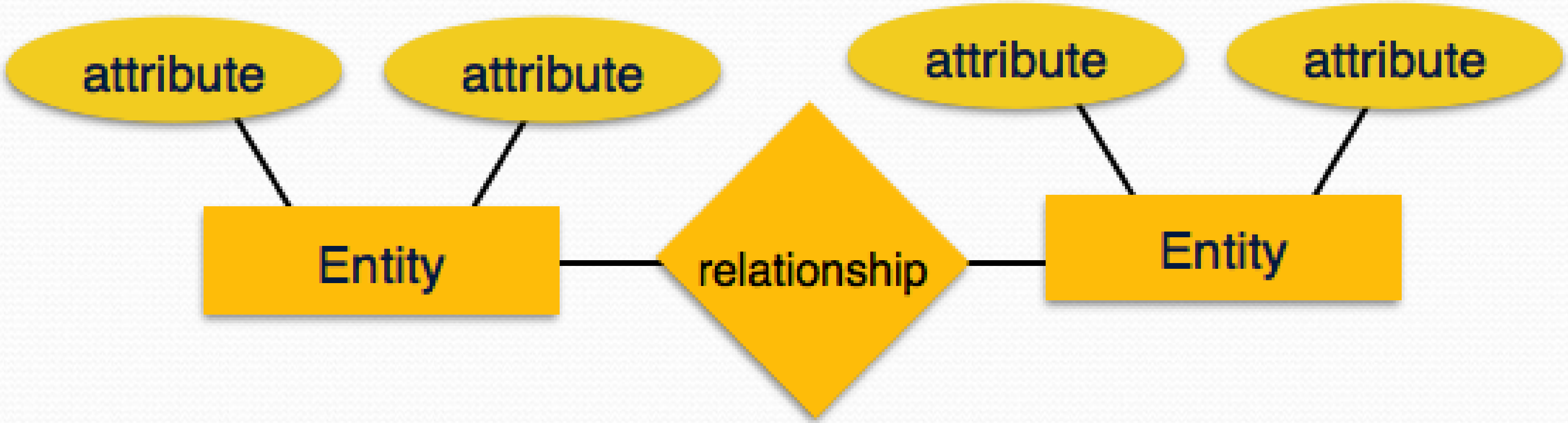
  For example, a student entity may have name, class, and age as attributes.

- Every **attribute** is defined by its set of values called **domain**.

  For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.

E-R Model Contd…

- E-R Model is based on –
- ➢ **Entities** and their attributes.
- ➢ **Relationships** among entities.



E-R Model Contd…

Types of Attributes

- Simple attribute
- Composite attribute
- Derived attribute
- Single-valued attribute
- Multivalued attribute

E-R Model Contd…

- **Simple attribute −** Simple attributes consist of atomic values, which cannot be divided further.
  - For example, a student's phone number is an atomic value of 10 digits

- **Composite attribute −** Composite attributes are made of more than one simple attribute.
  - For example, a student's complete name may have first_name and last_name.

- **Derived attribute −** Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database.
  - For another example, age can be derived from data_of_birth.

E-R Model Contd…

- **Single-valued attribute −** Single-valued attributes contain single value.
  - For example − Social_Security_Number.

- **Multivalued attribute −** Multivalued attributes may contain more than one values.
  - For example, a person can have more than one phone number, email_address, etc

E-R Model Contd…

- **Relationship** – The logical association among entities is called ***relationship***.

- **Relationship Set-** A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes. These attributes are called ***descriptive attributes***.

E-R Model Contd…

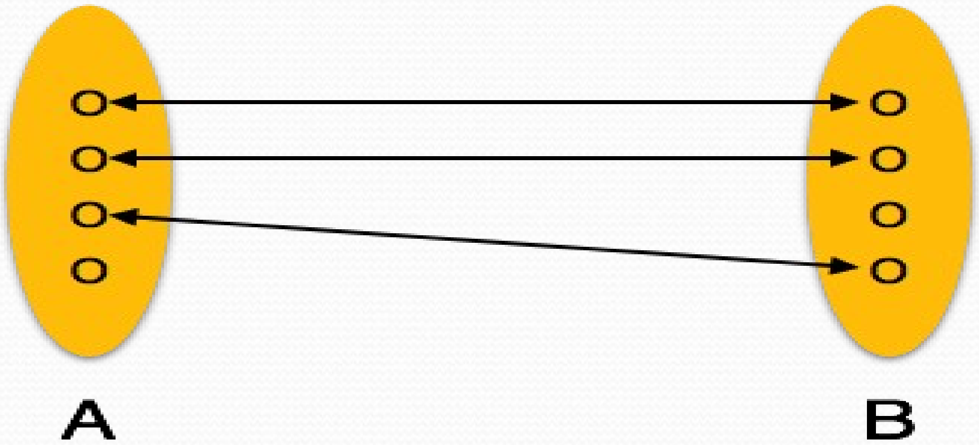
Mapping cardinalities –

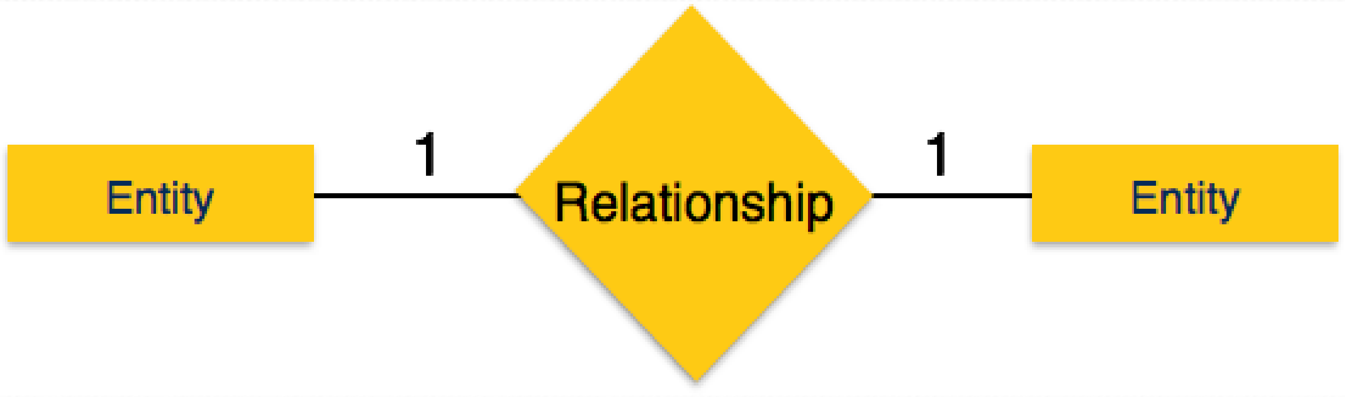
Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

- one to one
- one to many
- many to one
- many to many

E-R Model Contd…

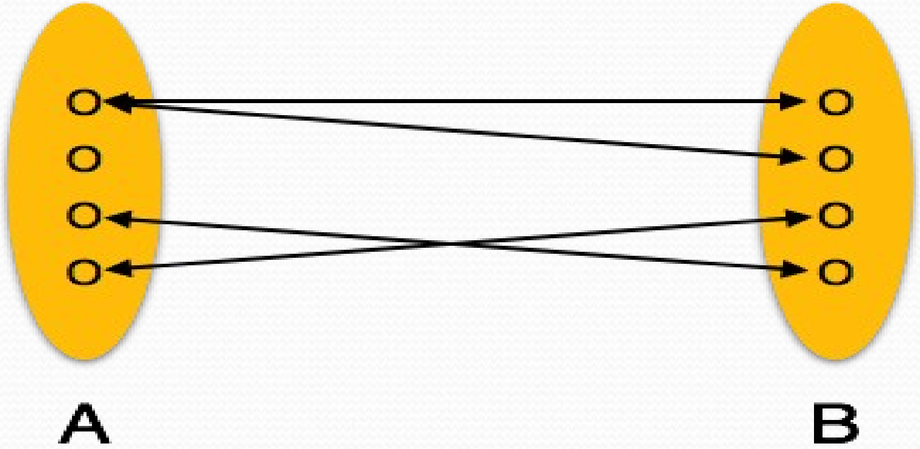
- **One-to-one -** One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

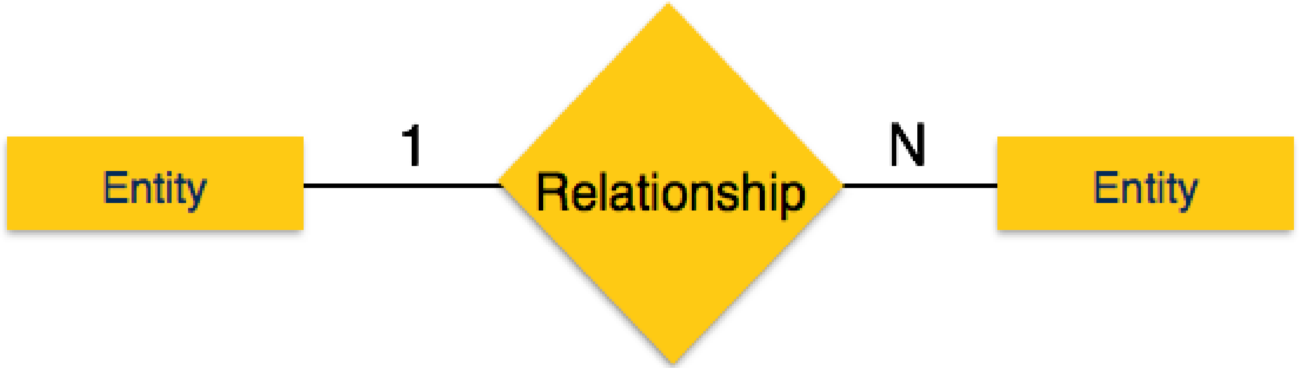




E-R Model Contd…

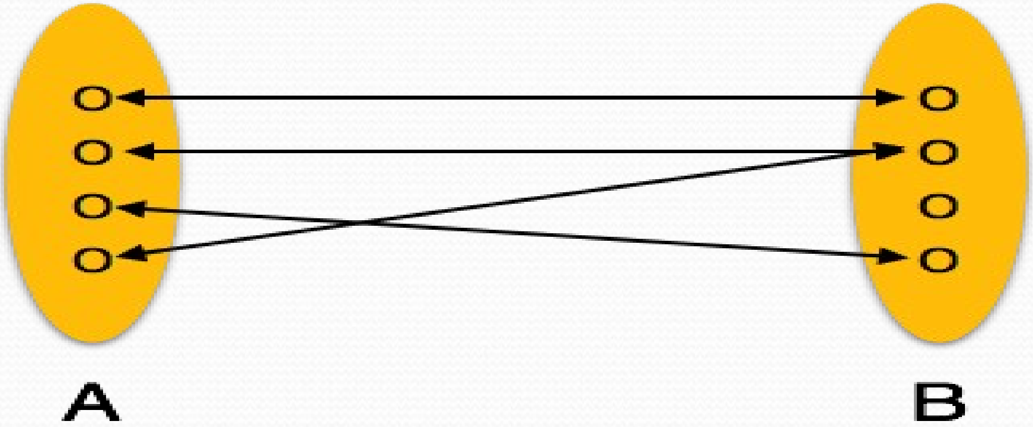
One-to-many – One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.

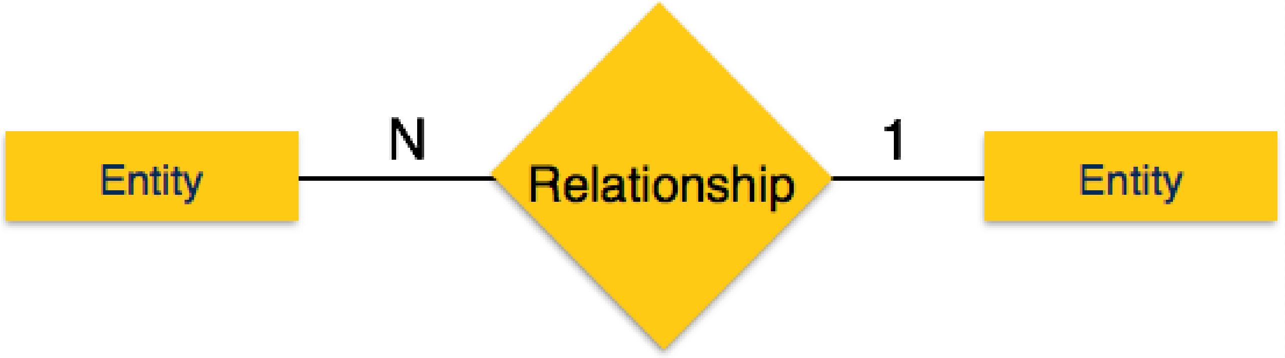




E-R Model Contd…

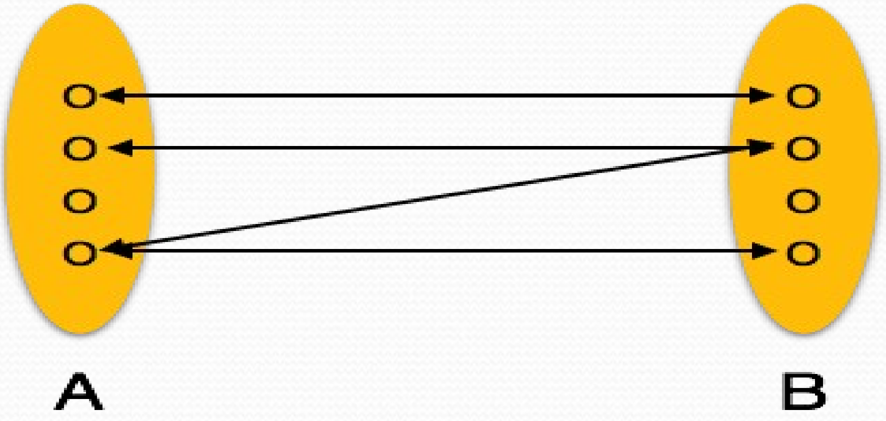
- **Many-to-one –** More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.

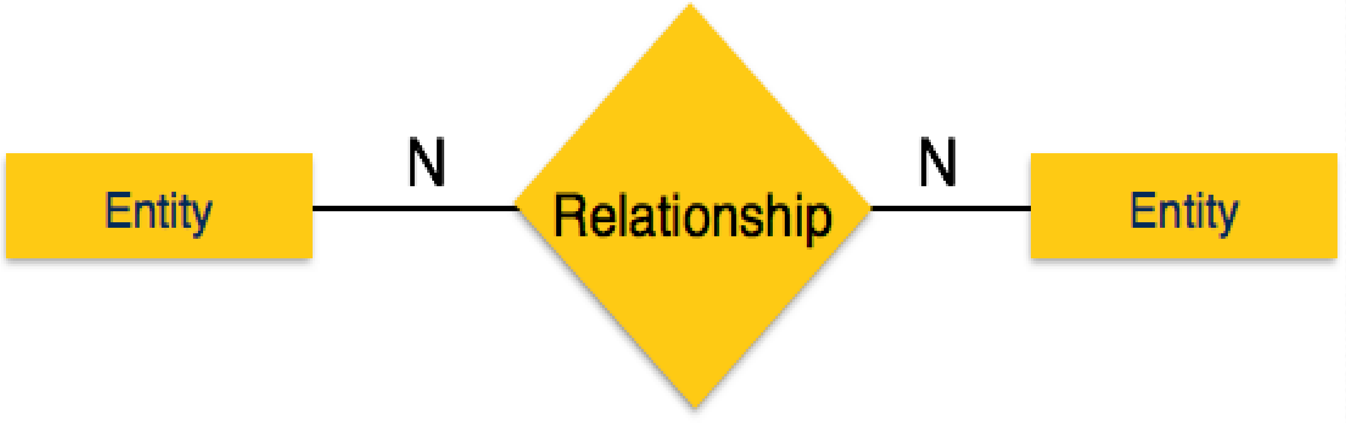




E-R Model Contd…

Many-to-many − One entity from A can be associated with more than one entity from B and vice versa.





E-R Model Contd…

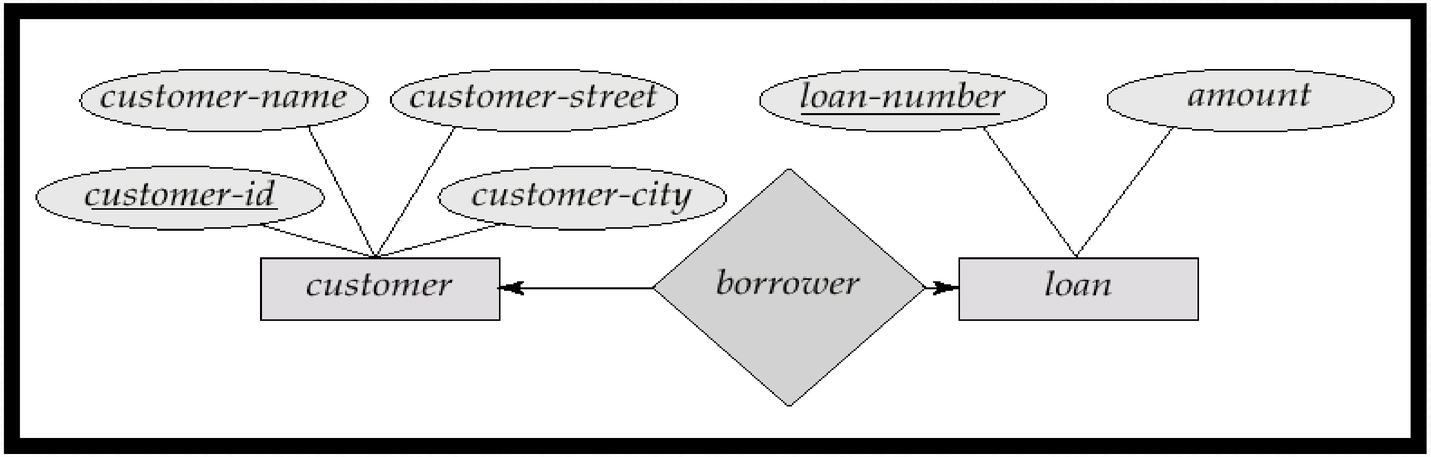
- **Cardinality Constraints**

➢We express cardinality constraints by drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

➢E.g.: **One-to-one relationship:**

✓A customer is associated with at most one loan via the relationship borrower

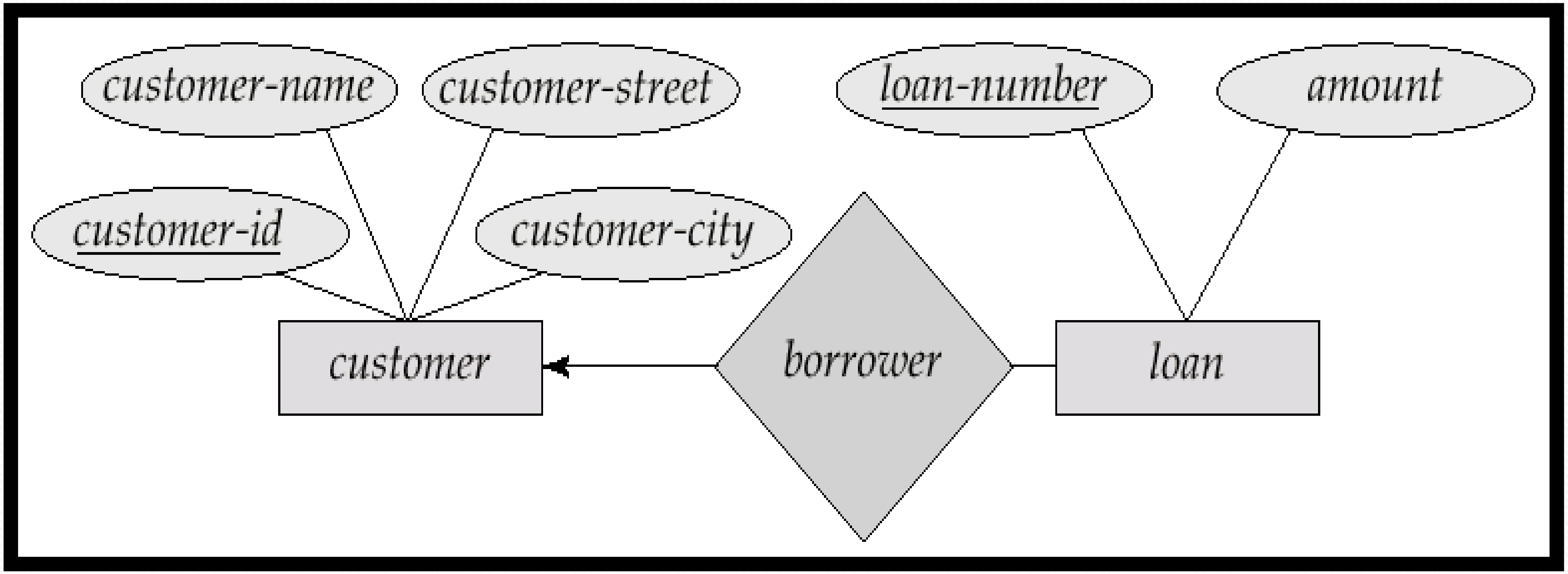
✓A loan is associated with at most one customer via borrower



E-R Model Contd…

- **Cardinality Constraints**

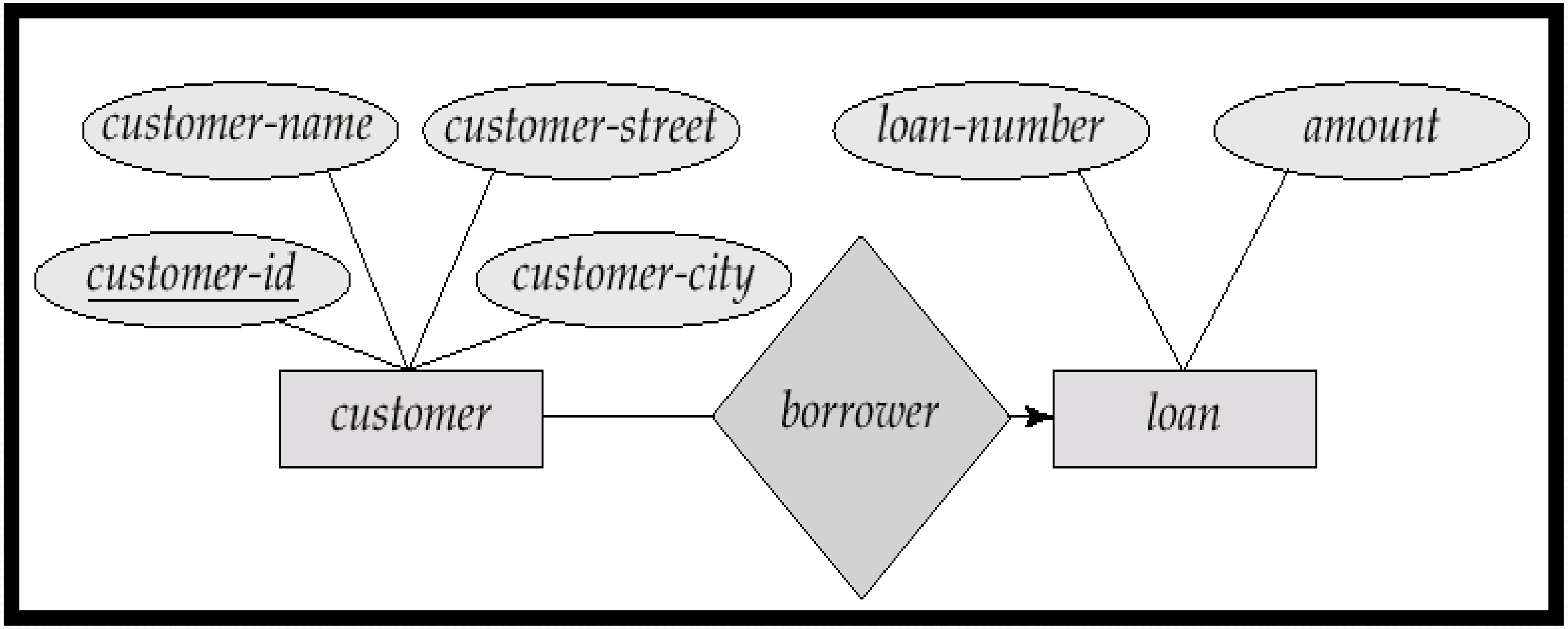
➢ In the **one-to-many relationship** a loan is associated with at most one customer via *borrower*, a customer is associated with several (including 0) loans via *borrower*



E-R Model Contd…

- **Cardinality Constraints**

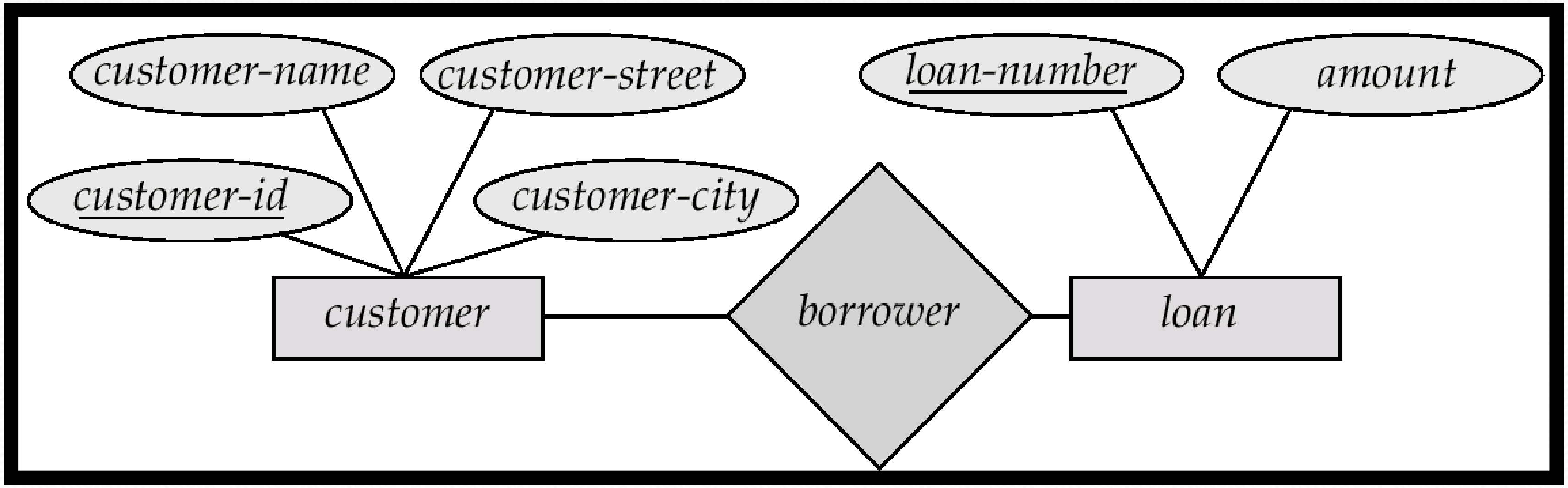
➢ In a **many-to-one relationship** a loan is associated with several (including 0) customers via *borrower*, a customer is associated with at most one loan via *borrower*



E-R Model Contd…

Cardinality Constraints

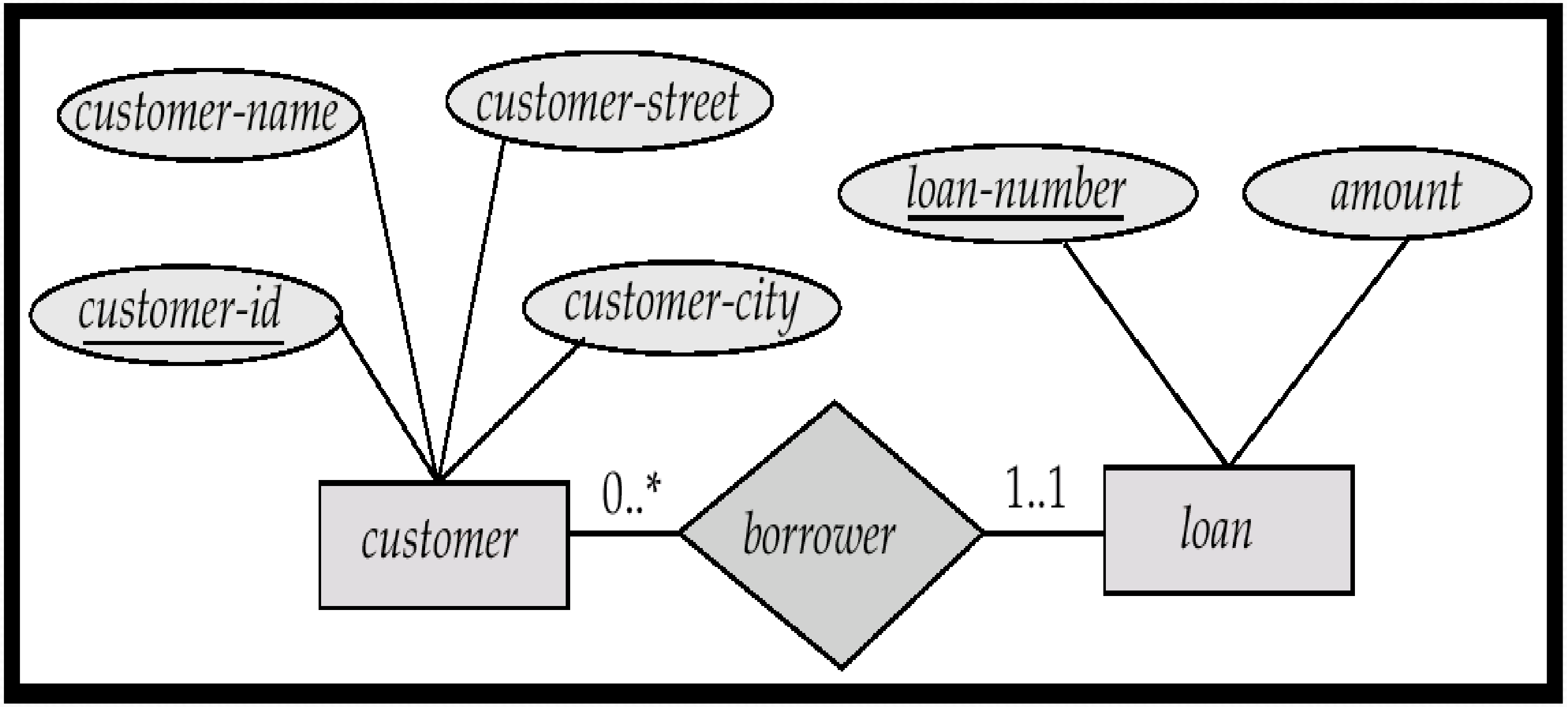
- A customer is associated with several (possibly 0) loans via borrower
- A loan is associated with several (possibly 0) customers via borrower



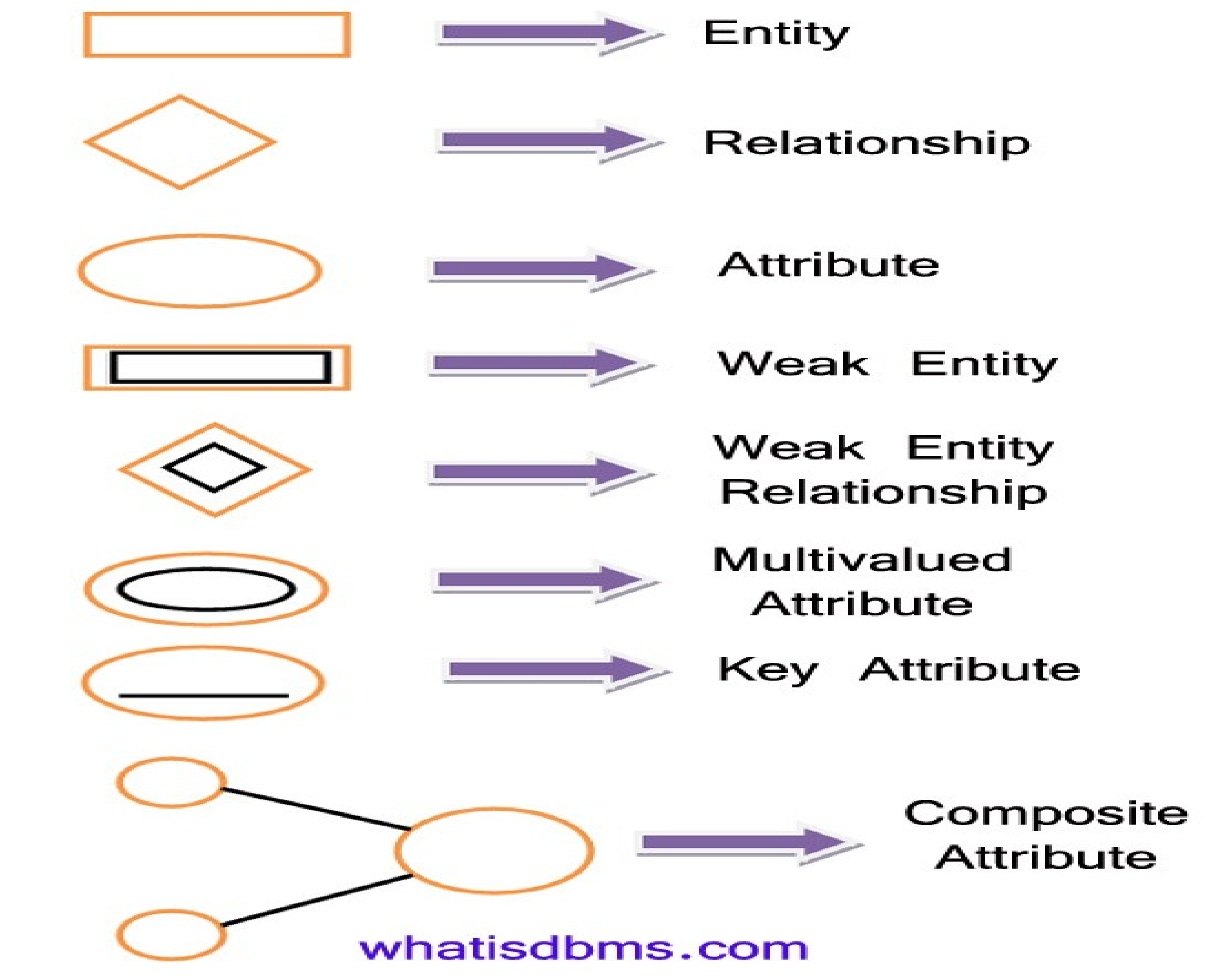
E-R Model Contd…

- **Alternative Notation for Cardinality Limits**

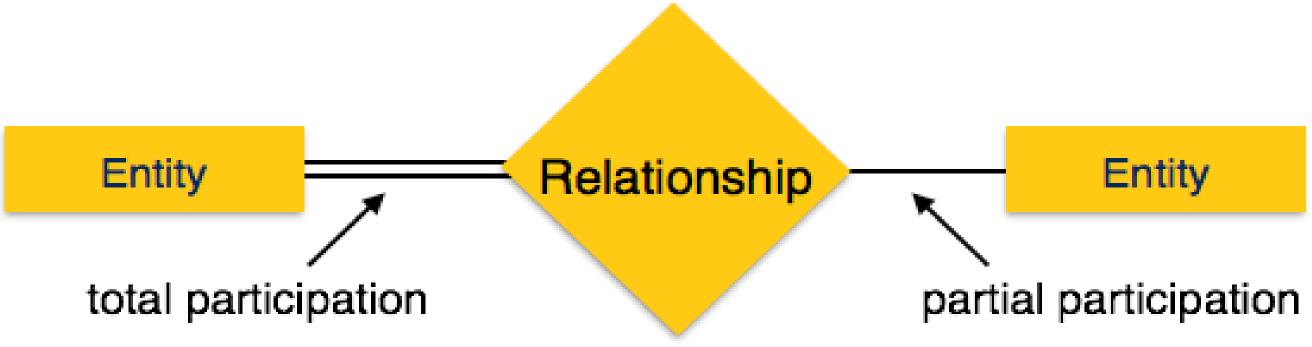
➢Cardinality limits can also express participation constraints



E-R Model Contd…



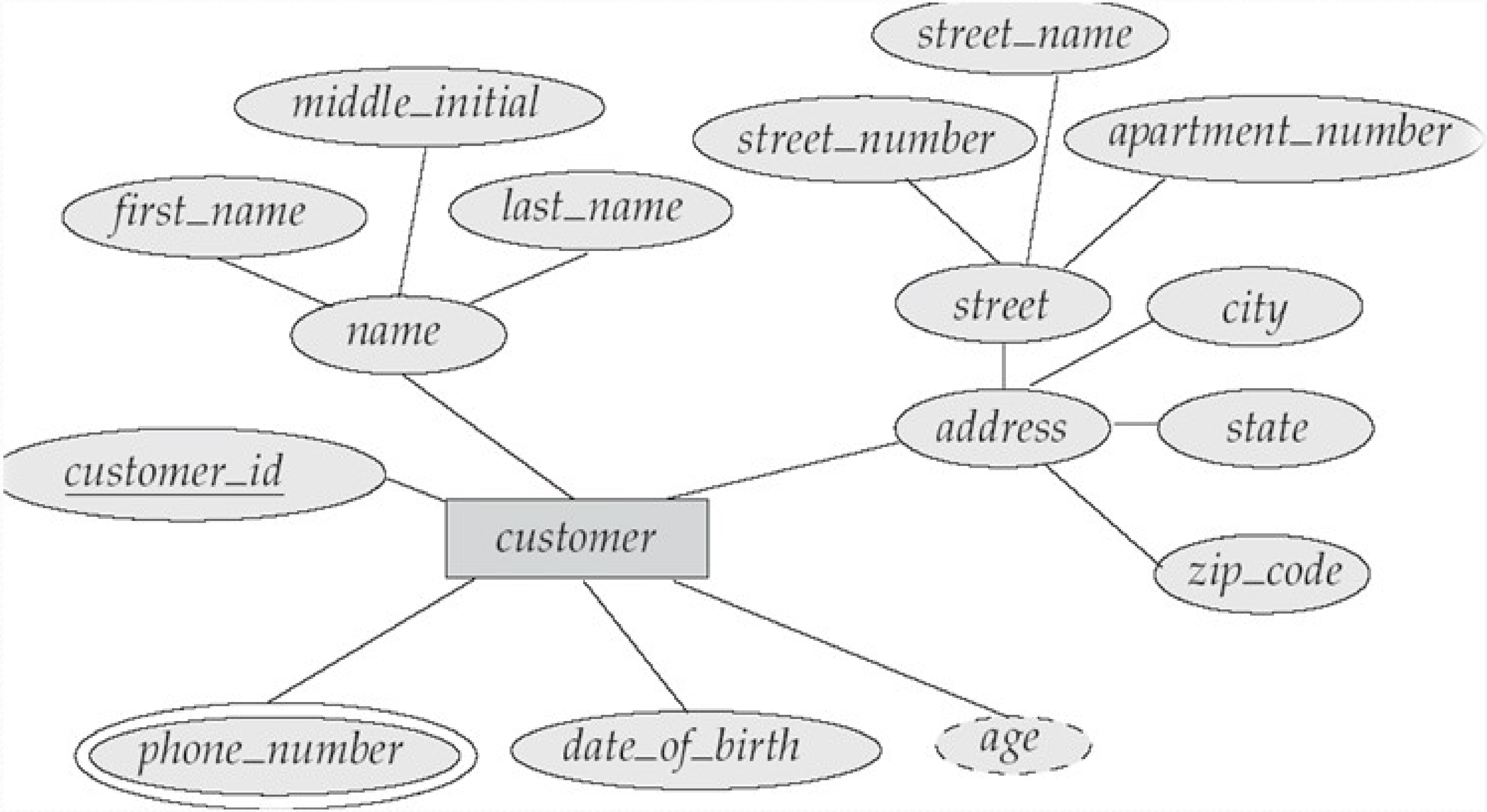
E-R Model Contd…



Total Participation − Each entity is involved in the relationship. Total participation is represented by double lines.

Partial participation − Not all entities are involved in the relationship. Partial participation is represented by single lines.

E-R Model Contd…



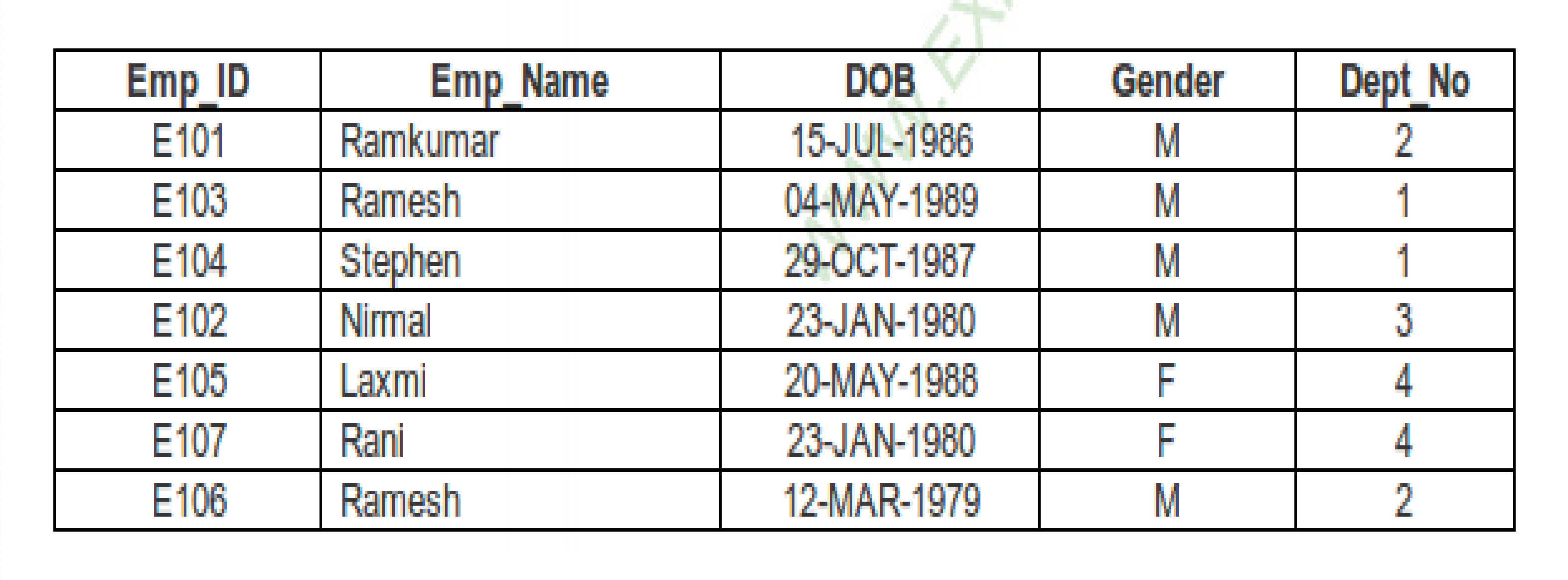
Keys

- Key – A key for an entity is a set of attributes that suffice to distinguish entities from each other or uniquely define that entity.

- Types -
  - Superkey
  - Candidate Key
  - Primary Key
  - Foreign Key

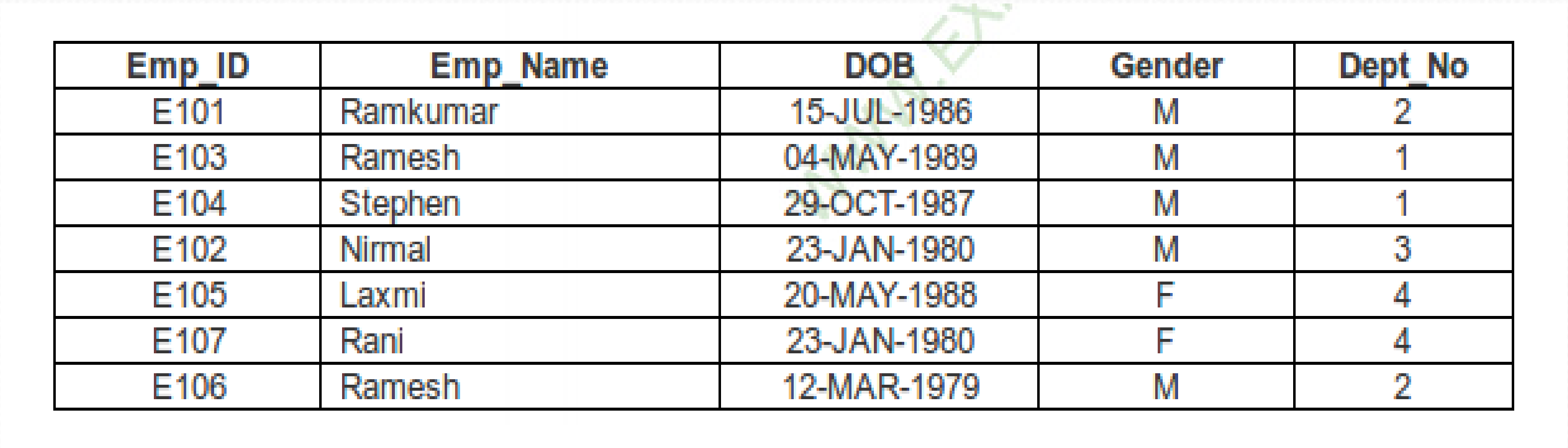
Superkey

- A superkey is a set of one or more attributes that, taken collectively, allow us to uniquely identify a tuple in the relation.



Emp_ID	Emp_Name	DOB	Gender	Dept_No
E101	Ramkumar	15-JUL-1986	M	2
E103	Ramesh	04-MAY-1989	M	1
E104	Stephen	29-OCT-1987	M	1
E102	Nirmal	23-JAN-1980	M	3
E105	Laxmi	20-MAY-1988	F	4
E107	Rani	23-JAN-1980	F	4
E106	Ramesh	12-MAR-1979	M	2

Superkey Example



Emp_ID	Emp_Name	DOB	Gender	Dept_No
E101	Ramkumar	15-JUL-1986	M	2
E103	Ramesh	04-MAY-1989	M	1
E104	Stephen	29-OCT-1987	M	1
E102	Nirmal	23-JAN-1980	M	3
E105	Laxmi	20-MAY-1988	F	4
E107	Rani	23-JAN-1980	F	4
E106	Ramesh	12-MAR-1979	M	2

SuperKey- {Emp_ID}
{Emp_ID,Emp_Name},{Emp_ID,DOB},
{Emp_Name,DOB},
{Emp_Name, DOB, Gender},...

Candidate Key

Candidate key is nothing but minimal super keys for which no proper subset is a super key.

SuperKey- {Emp_ID}
{Emp_ID,Emp_Name},{Emp_ID,DOB},
{Emp_Name,DOB},
{Emp_Name, DOB, Gender},...

Candidate Key – {Emp_ID}
{Emp_Name, DOB}

Primary Key

- The term primary key is used to denote a candidate key that is chosen by database designer as principal means of identifying entities within an entity set.

- Primary key entity in the set cannot have the same value for two or more tuples i.e. unique and it cannot be null.

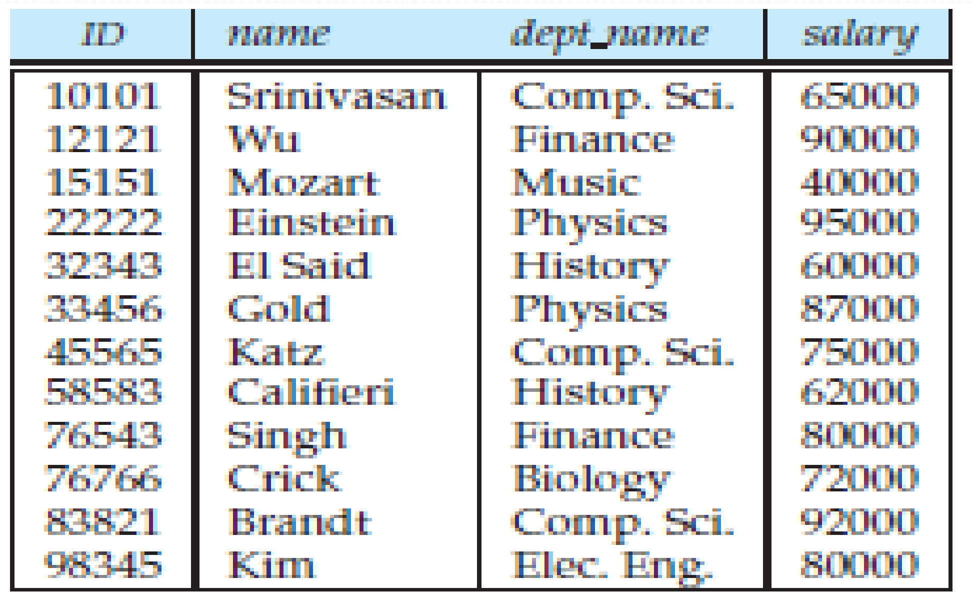
- Primary Key - {Emp_ID}

Foreign Key

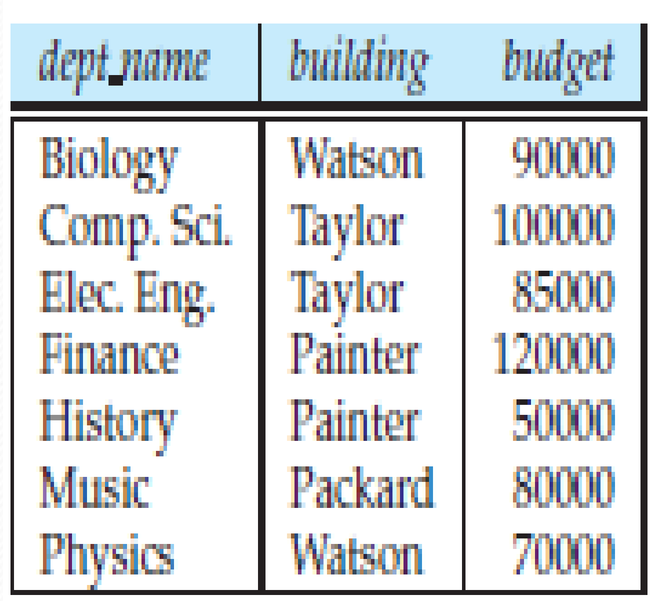
- A relation, say r1, may include among its attributes the primary key of another relation, say r2. This attribute is called a foreign key from r1, referencing r2.

- r1 - referencing relation
- r2- referenced relation

Foreign Key Example



ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000



dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

Referencing relation –Instructor

Referenced relation –Department

Foreign Key - dept_name

E-R Model Contd…

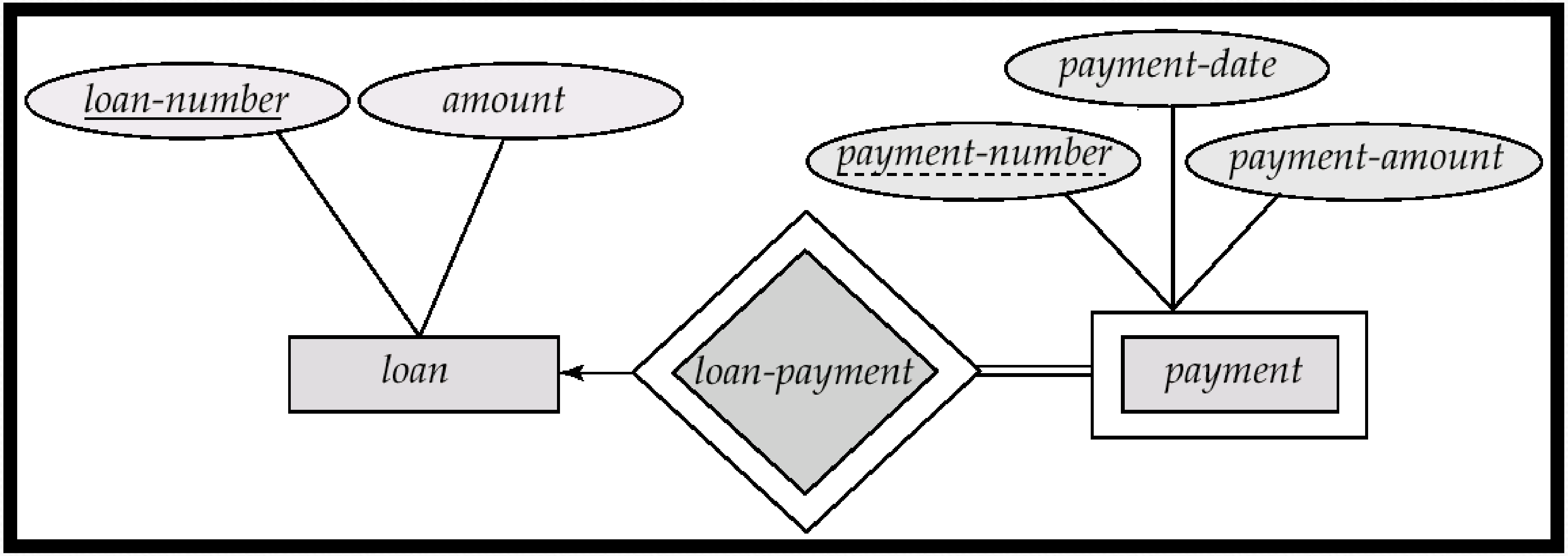
- **Weak Entity Sets**

- An entity set that does not have a primary key is referred to as a *weak entity set*.
- The existence of a weak entity set depends on the existence of a *identifying entity set*
- The *discriminator* *(or partial key)* of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.
- The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's discriminator.

E-R Model Contd…

Weak Entity Sets

- We depict a weak entity set by double rectangles.
- We underline the discriminator of a weak entity set with a dashed line.
- *payment-number* – discriminator of the payment entity set
- Primary key for *payment* – (*loan-number, payment-number*)



Extended E-R Diagram(EER)

- An enhanced entity-relationship model, also known as an extended **entity-relationship model**, is a type of database diagram that's similar to regular ERDs.
- Enhanced ERDs are high-level conceptual models that accurately represent the requirements of complex databases.
- In addition to E-R diagram, EERDs include:

- Subtypes and supertypes (sometimes known as subclasses and superclasses)
- Attribute and relationship inheritance
- Specialization or generalization
- Aggeregation

Extended E-R Diagram (EER) Contd…

- **Subclasses and Super-classes**

- An entity type may have additional meaningful sub-groupings of its entities
- **Example:** EMPLOYEE may be further grouped into SECRETARY, ENGINEER, MANAGER, TECHNICIAN, SALARIED_EMPLOYEE, HOURLY_EMPLOYEE,…
- ✓ Each of these groupings is a subset of EMPLOYEE entities
- ✓ Each is called a subclass of EMPLOYEE
- ✓ EMPLOYEE is the superclass for each of these subclasses
- These are called super-class/subclass relationships.

Extended E-R Diagram (EER) Contd…

- **Example:**

  EMPLOYEE/SECRETARY,

  EMPLOYEE/TECHNICIAN

- These are also called IS-A relationships

  (SECRETARY IS-A EMPLOYEE, TECHNICIAN IS-A EMPLOYEE, …).

Extended E-R Diagram (EER) Contd…

- **Specialization**

- Top-down design process; we designate subgrouping within an entity set that are distinctive from other entities in the set.

- These subgrouping become lower-level entity sets that have attributes or participate in relationships that do not apply to the higher-level entity set.

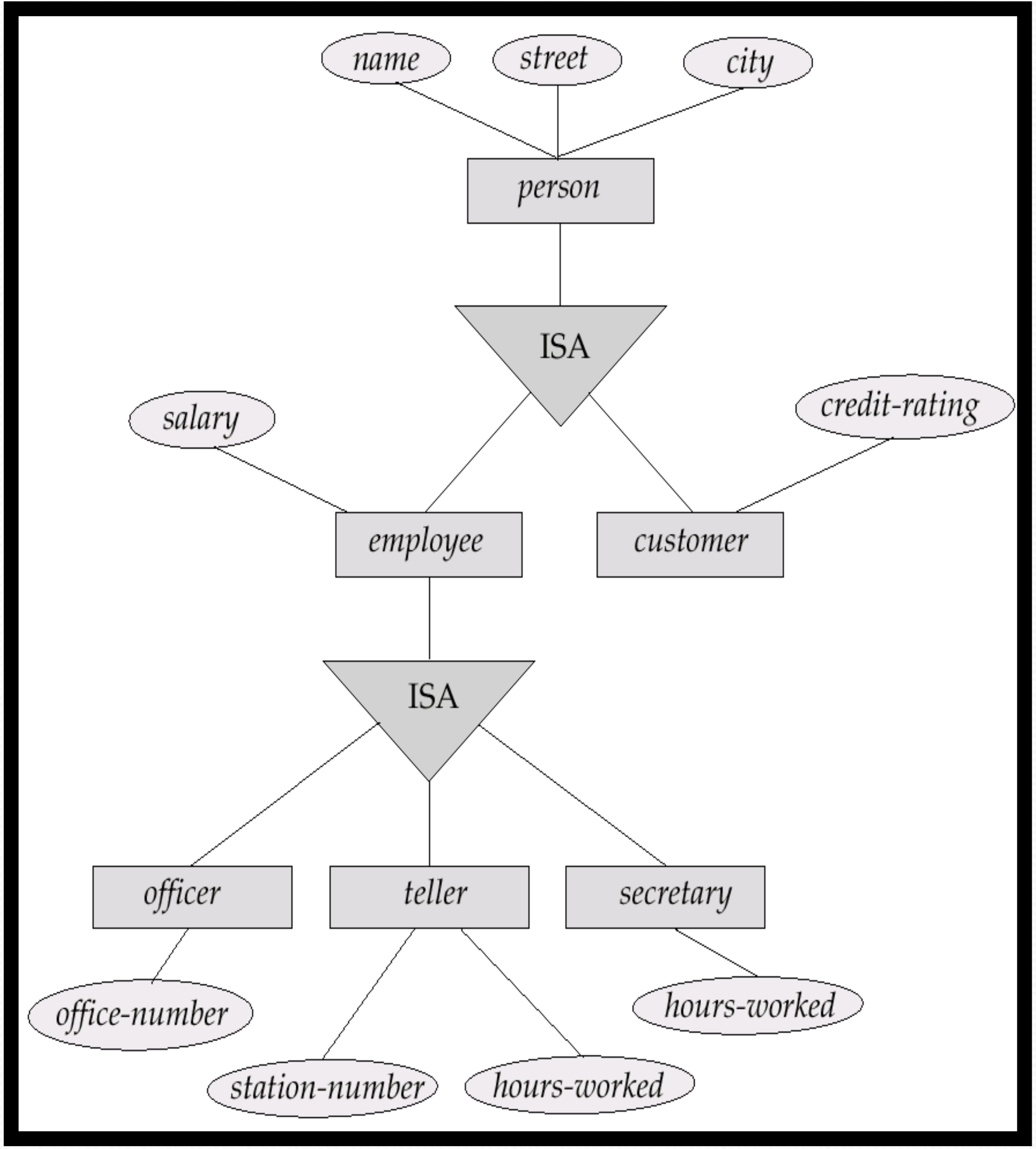
- Depicted by a *triangle* component labeled ISA (E.g. *customer* "is a" *person*).

- **Attribute inheritance** – a lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked.

Extended E-R Diagram (EER) Contd…

- **Specialization**

  Example



Extended E-R Diagram (EER) Contd…

- **Generalization**

- A bottom-up design process – combine a number of entity sets that share the same features into a higher-level entity set.

- **Specialization and generalization** are simple inversions of each other; they are represented in an E-R diagram in the same way.

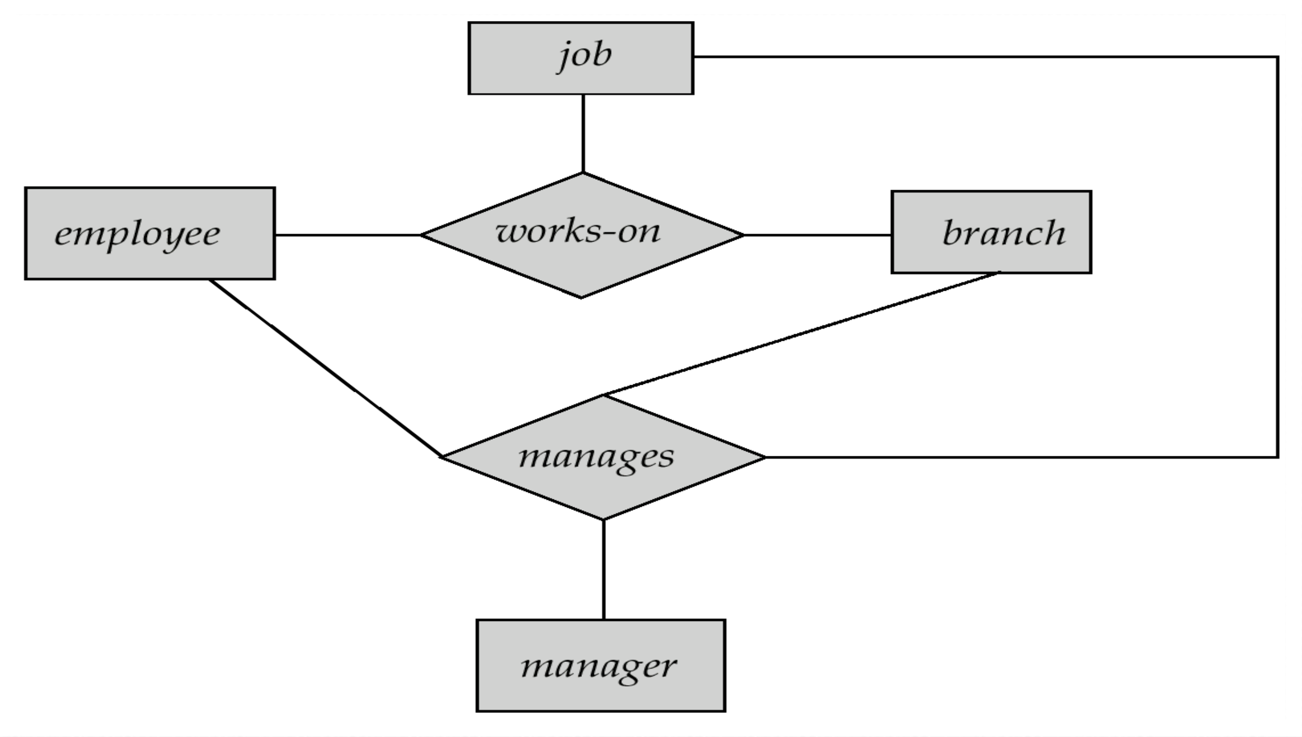
- The terms specialization and generalization are used interchangeably.

Extended E-R Diagram (EER) Contd…

- ## Aggregation

- One limitation of the E-R model is that it cannot express relationships among relationships.

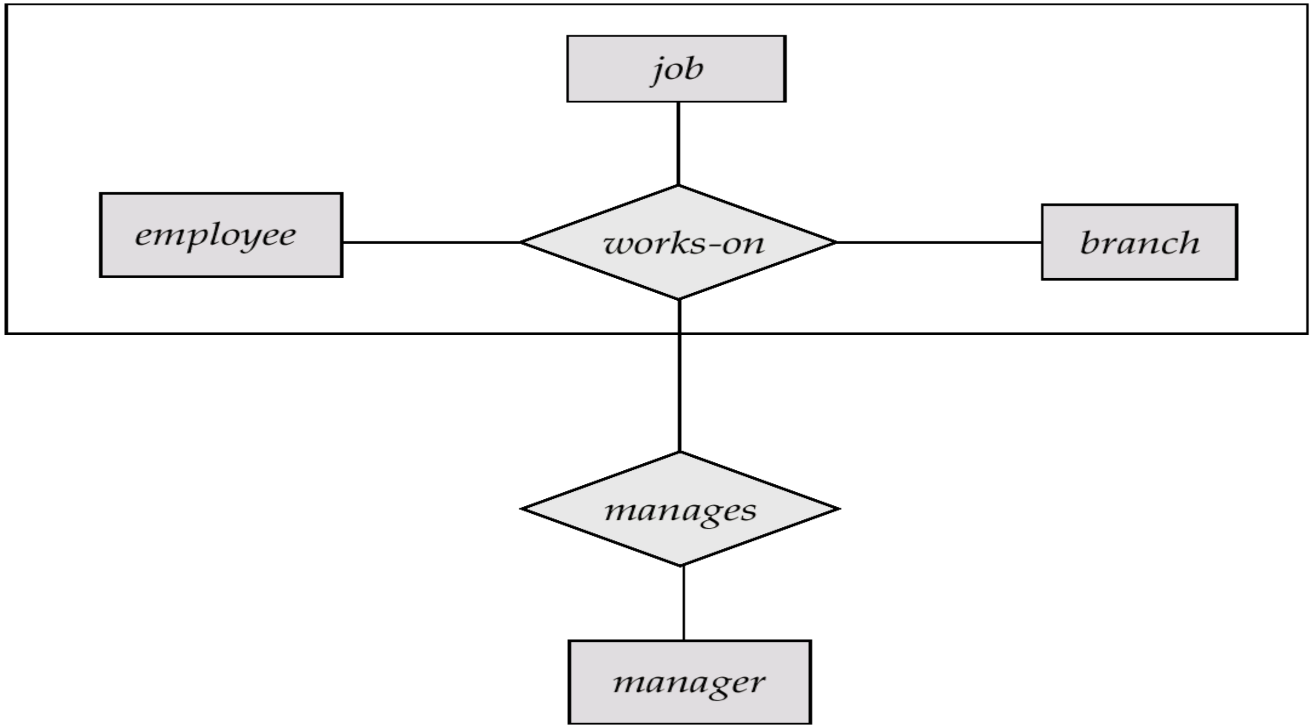
- Consider the ternary relationship *works-on:*



Extended E-R Diagram (EER) Contd…

- **Aggregation-** Aggregation is an abstraction through which relationships are treated as higher-level entities.

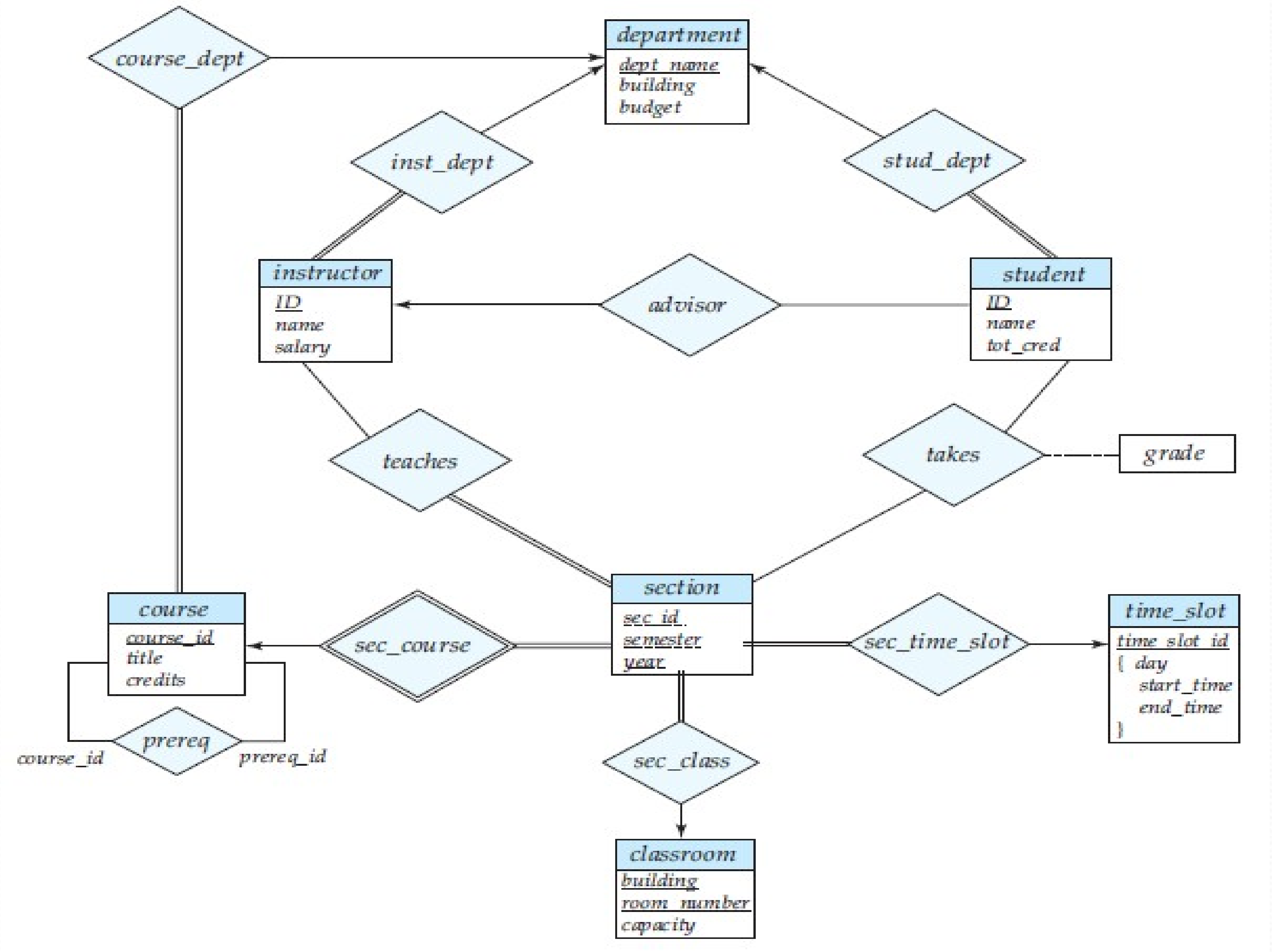
- Without introducing redundancy, the following diagram represents:
  - ✓ An employee works on a particular job at a particular branch
  - ✓ An employee, branch, job combination may have an associated manager



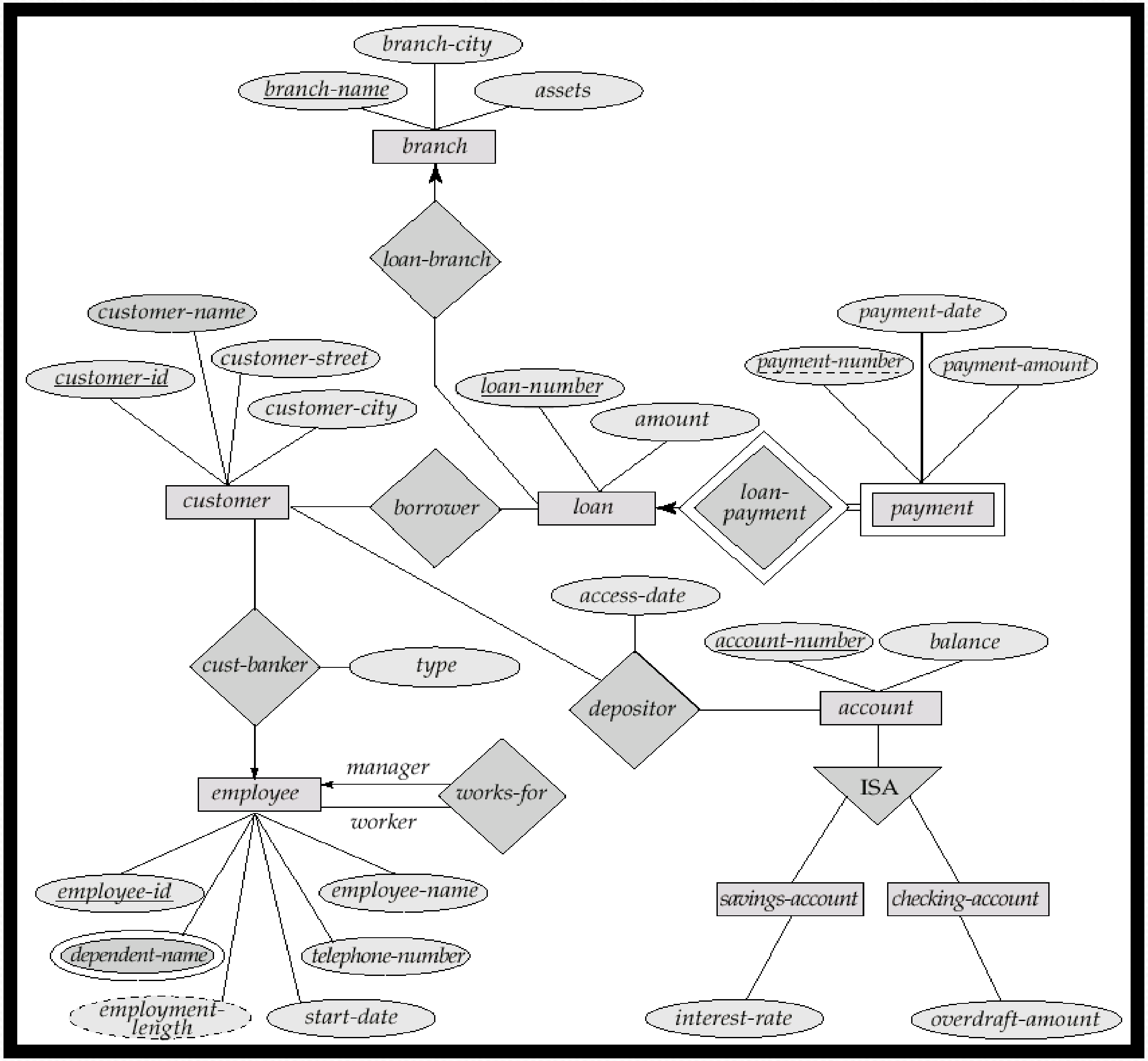
E-R Diagram for a University Database

classroom(building, room number, capacity)
department(dept name, building, budget)
course(course id, title, dept name, credits)
instructor(ID, name, dept name, salary)
section(course id, sec id, semester, year, building, room number, time slot id)
teaches(ID, course id, sec id, semester, year)
student(ID, name, dept name, tot cred)
takes(ID, course id, sec id, semester, year, grade)
advisor(s ID, i ID)
time slot(time slot id, day, start time, end time)
prereq(course id, prereq id)

E-R Diagram for a University Database



E-R Diagram for a Banking Enterprise



Converting ER diagram into tables

- Strong Entity Set – Individual table for each entity set with all attributes.

- Attributes -

➢ Simple/Single valued – column in table

➢ Composite – represented as individual columns in table

➢ Multi-valued – separate table for attribute with two fields *(Primary key of table and Multivalued attribute)*

- Weak Entity Set – Separate table for weak entity with all attributes along with primary key of identifying entity.

Converting EER diagram into tables

- Relationship set – Separate table consist of primary key of all entities participating in relation.

- Specialization/Generalization – Separate table for higher level and lower level entity set.

eg. person(ID,name,street,city)

employee(ID,salary)

student(ID, marks)

- Aggregation – Consist of all primary keys for aggregate relationship and entity.

END

OF

UNIT I