UNIVERSITY OF CHITTAGONG

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Program: **B.Sc.** (Engineering)
Session: 2022–2023
4th Semester

Assignment_01
(E-R modeling)

Course Title: DataBase Systems
Course Code: CSE-413

Submitted To

Dr. Rudra Pratap Deb Nath
Associate Professor
Department of Computer Science & Engineering
University of Chittagong

Submitted By

Subha Shesgin
ID: 23701036
Department of Computer Science & Engineering

Date of Submission: 10 June, 2025

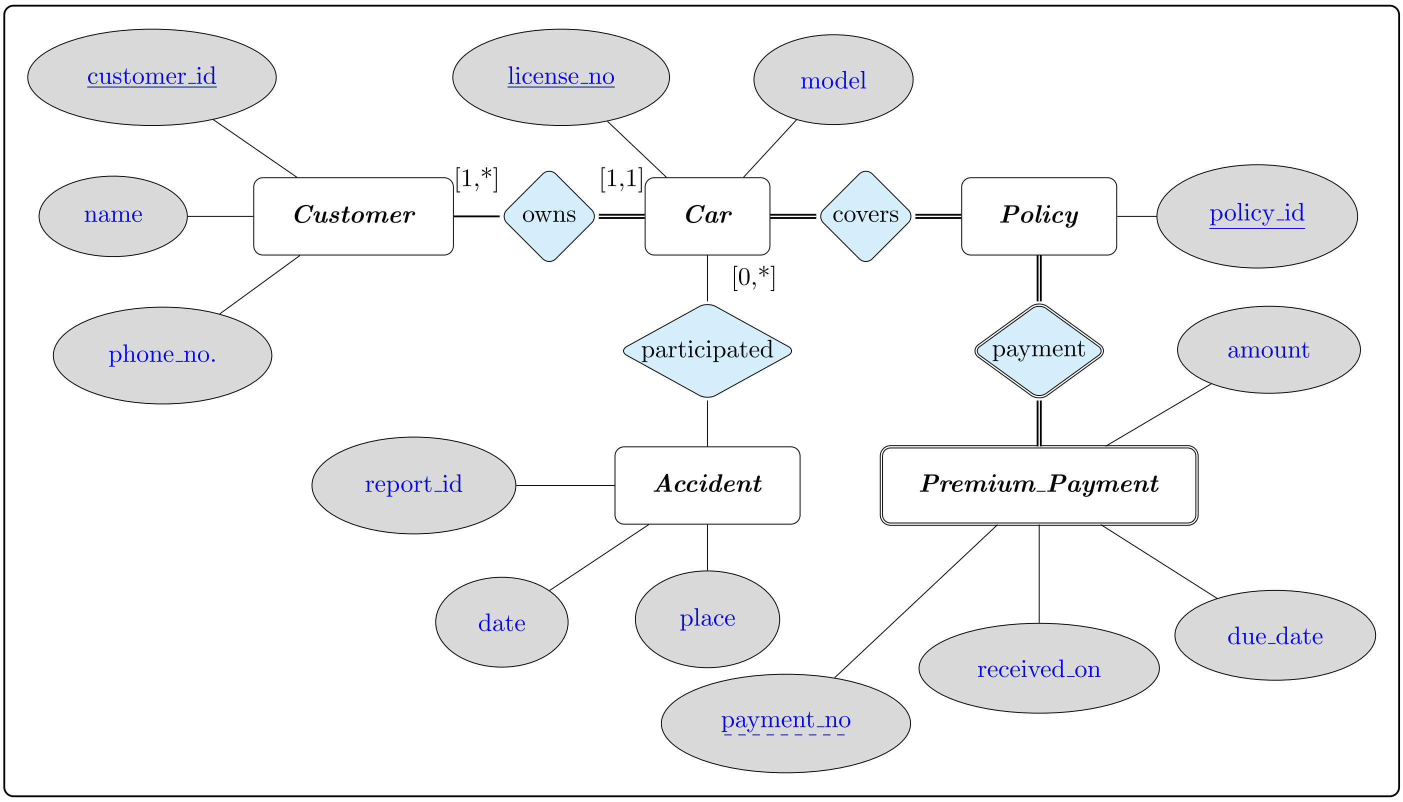
DATABASE SYSTEM CONCEPTS

Chapter - 6

DataBase Design Using the E-R Model

1. **Question:**
   Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars and has one or more premium payments associated with it. Each payment is for a particular period of time and has an associated due date and the date when the payment was received.

   Answer :
   The E-R diagram is centered around five main entities: **Customer, Car, Accident, Policy,** and **Premium Payment**. Each customer can own multiple cars, and a car may be involved in any number of accidents, including none. Accident records include details such as a report ID, date, and location. A policy can cover several cars and is associated with multiple premium payments. Each payment contains information like the due date, date of receipt, and payment amount. The **Premium Payment** is a weak entity, meaning it depends on the Policy entity for its existence. Key relationships in the model include Owns (linking Customer and Car), Participated (connecting Car and Accident), Covers (relating Policy to Car), and Payment (associating Policy with Premium Payment)



2. **Question:**

   Consider a database that includes the entity sets student, course, and section from the university schema and that additionally records the marks that students receive in different exams of different sections.

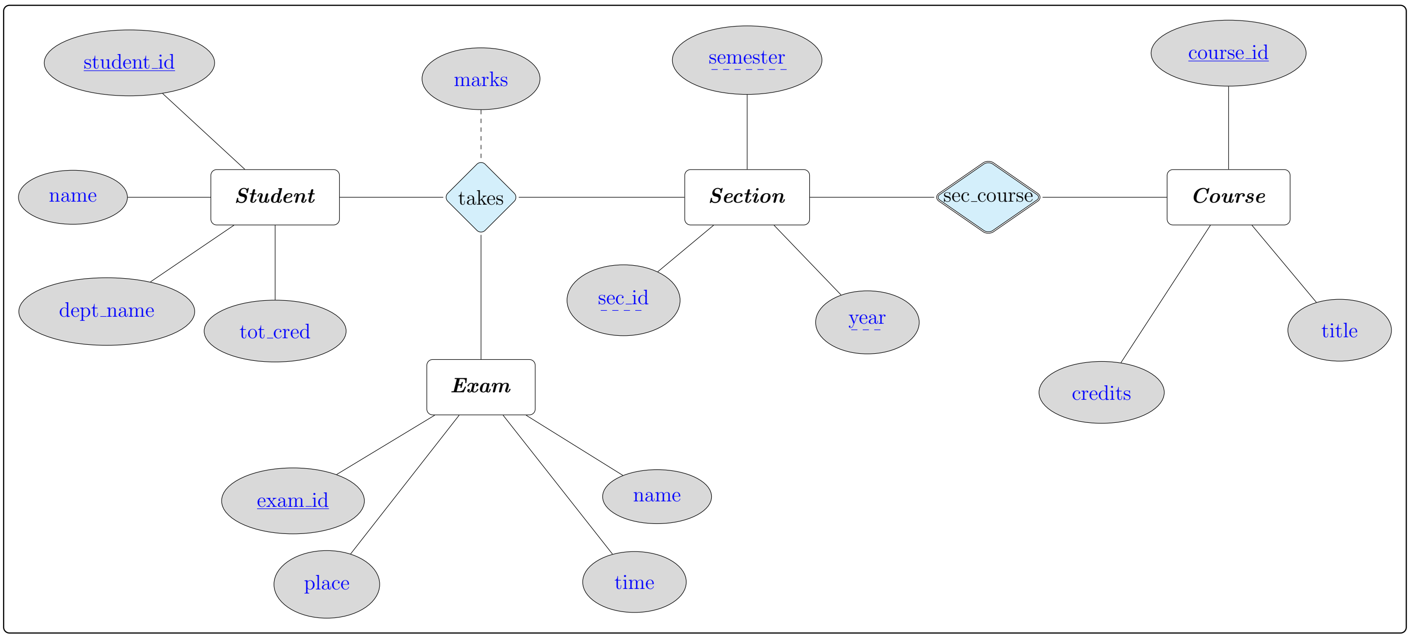
   a) Construct an E-R diagram that models exams as entities and uses a ternary relationship as part of the design.

   b) Construct an alternative E-R diagram that uses only a binary relationship between student and section. Make sure that only one relationship exists between a particular student and section pair, yet you can represent the marks that a student gets in different exams.

Answer:

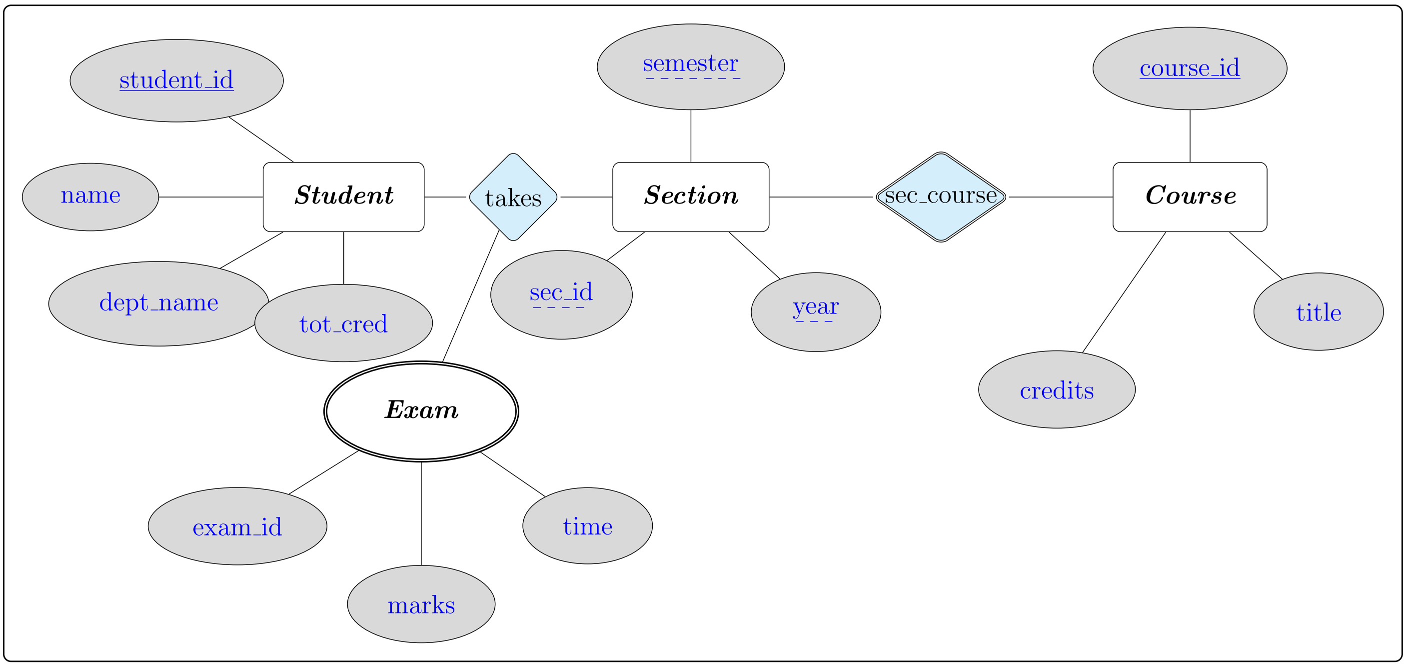
a) E-R Diagram Featuring **Exams** as Entities and a Ternary Relationship:

This E-R diagram models **Student**, **Section**, and **Exam** as distinct entity sets. The **Exam** entity includes attributes such as *exam_id*, *name*, *place*, and *time*. To capture the marks obtained by a student in a particular exam conducted within a specific section, a ternary relationship named **takes** is introduced. This relationship links the **Student**, **Section**, and **Exam** entities and includes the attribute *marks* to represent the score. The ternary relationship structure ensures that the participation context—where a student takes a particular exam in a specific section—is accurately represented. This model is especially effective in scenarios where an exam may be common across multiple sections or when exams are treated as independent entities.



b) E-R Diagram Using Only Binary Relationship Between **Student** and **Section**:

In this E-R diagram, we restrict ourselves to a binary relationship named **takes** between **Student** and **Section**. To avoid storing multiple records for the same student-section pair, we do not associate separate relationships for each exam. Instead, we treat **Exam** as a multi-valued composite attribute within the **takes** relationship. Each component of this composite attribute includes `exam_id`, `time`, and `marks`. This design treats exams not as independent entities, but as relevant only in the context of a student taking a section. As a result, the overall schema becomes simpler and more compact.

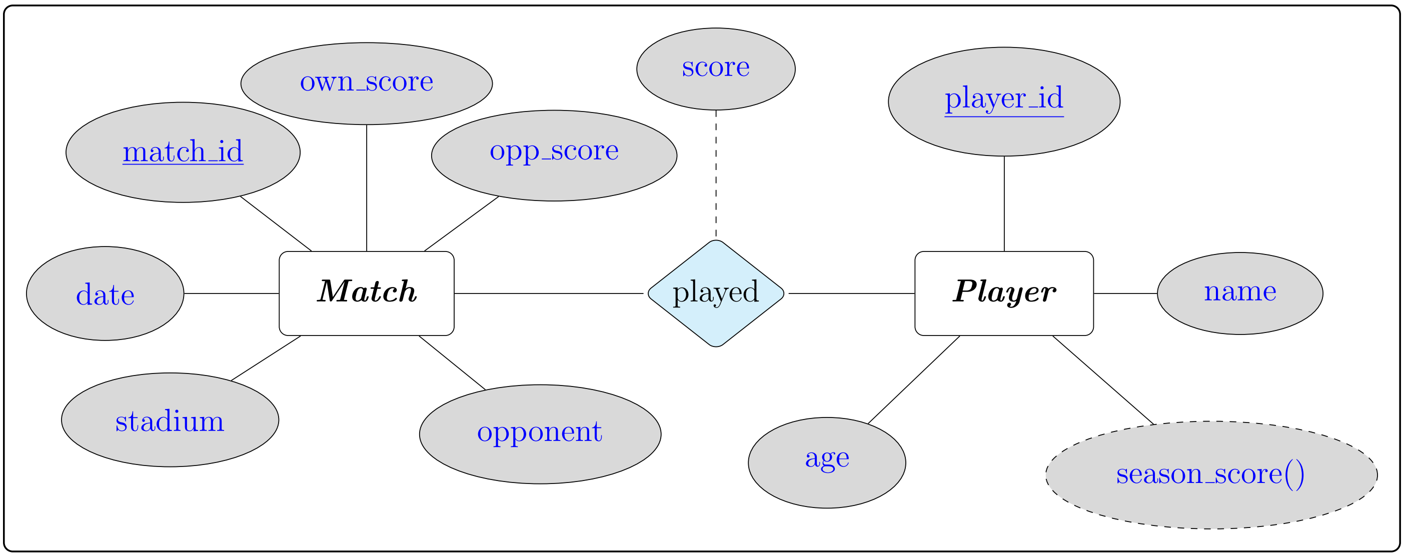


3. **Question:**

Design an E-R diagram for keeping track of the scoring statistics of your favorite sports team. You should store the matches played, the scores in each match, the players in each match, and individual player scoring statistics for each match. Summary statistics should be modeled as derived attributes with an explanation as to how they are computed.

Answer:

This E-R diagram models the scoring statistics of a sports team. It features two main entities: **Match** and **Player**. The **Match** entity includes attributes such as match_id, date, stadium, opponent team, own score, and opponent score. The **Player** entity contains player_id, name, and age, along with a derived attribute **season_score()**, which computes the total score a player has accumulated over the season. This value is not stored but derived from the scores across all matches the player has participated in. The **played** relationship connects players with the matches they played in and includes the attribute **score**, representing the points a player scored in that match. This design effectively organizes player and match data while allowing performance tracking.



5. **Question:**

   An E-R diagram can be viewed as a graph. What do the following mean in terms of the structure of an enterprise schema?

   a) The graph is disconnected.

   b) The graph has a cycle.

   Answer:

   a) When two entity sets in an E-R diagram are connected by a path, they are considered related, even if the relationship is indirect. A disconnected graph indicates that some entities have no connection and operate independently. In such cases, each connected component of the diagram can be treated as a separate database for a distinct part of the enterprise.

   b) A path between two entity sets signifies a direct or indirect relationship. If the diagram contains cycles, it indicates that there are multiple paths—and thus multiple relationships—between some entity sets. In contrast, an acyclic E-R diagram ensures that any two entity sets are connected by only one unique path, representing a single relationship between them.

7. **Question:**

   A weak entity set can always be made into a strong entity set by adding to its attributes the primary-key attributes of its identifying entity set. Outline what sort of redundancy will result if we do so.

   Answer:

   Converting a weak entity set into a strong one by incorporating the primary key of its identifying entity into its own attributes results in redundancy. This is because the identifying key values are duplicated for every instance of the former weak entity. Such duplication increases storage requirements and can lead to update anomalies, especially if the identifying key changes. Overall, this approach can reduce data consistency and make database maintenance more complex.

9. **Question:**

   Suppose the advisor relationship set were one-to-one. What extra constraints are required on the relation advisor to ensure that the one-to-one cardinality constraint is enforced?

   Answer:

   When the *advisor* relationship is one-to-one, it implies that each advisor is assigned to only one student, and each student has only one advisor. To enforce this constraint in the database, additional rules must be applied. Specifically, the advisor's ID should serve as a primary key to ensure uniqueness, and the student's ID must also be unique within the advisor relation. By enforcing uniqueness on both attributes, the database ensures that no advisor or student is associated with more than one counterpart, thereby preserving the one-to-one nature of the relationship.

13. **Question:**

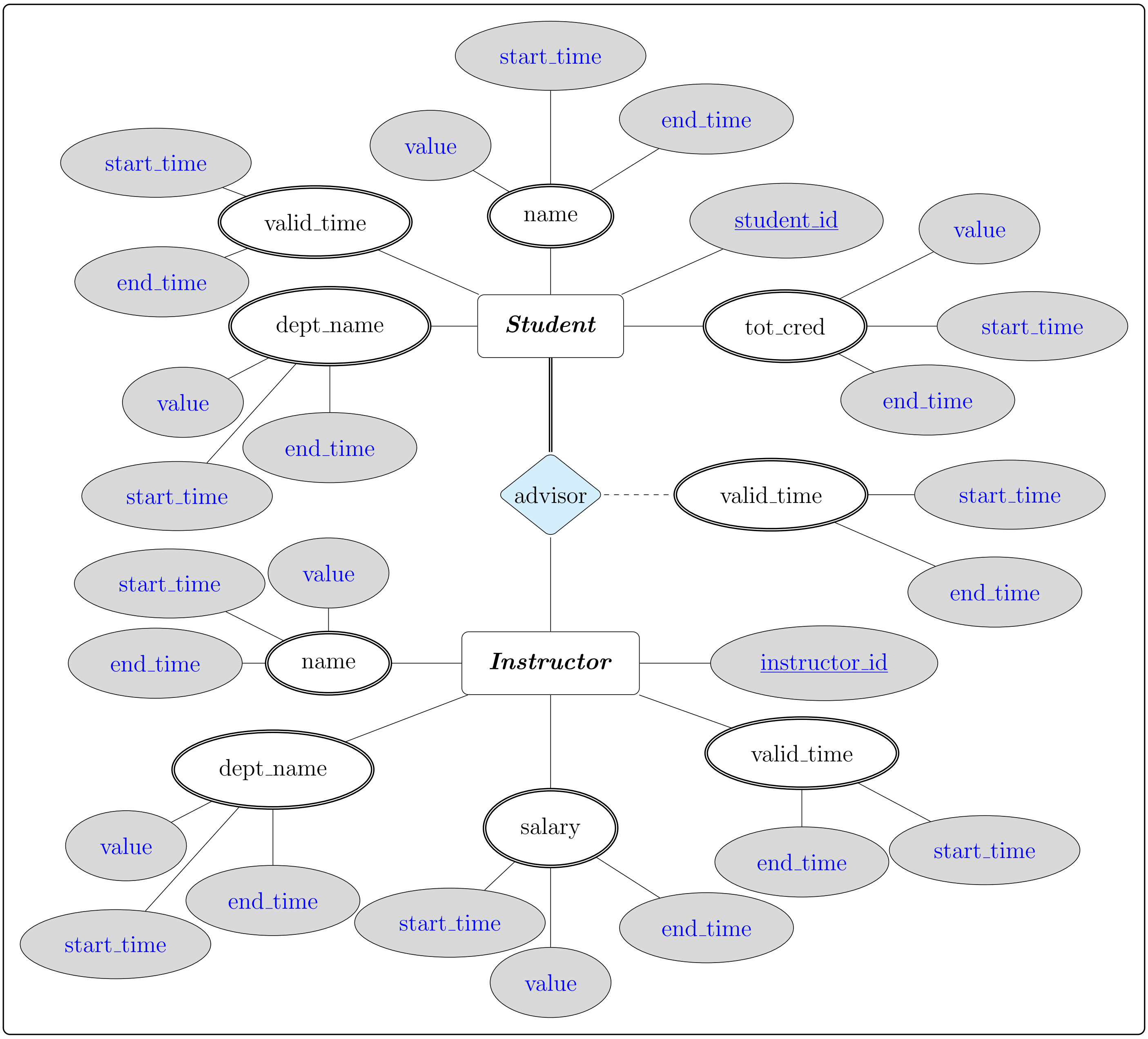
    An E-R diagram usually models the state of an enterprise at a point in time. Suppose we wish to track temporal changes, that is, changes to data over time. For example, Zhang may have been a student between September 2015 and May 2019, while Shankar may have had instructor Einstein as advisor from May 2018 to December 2018, and again from June 2019 to January 2020. Similarly, attribute values of an entity or relationship, such as title and credits of course, salary, or even name of instructor, and tot cred of student, can change over time. One way to model temporal changes is as follows: We define a new data type called valid time, which is a time interval, or a set of time intervals. We then associate a valid time attribute with each entity and relationship, recording the time periods during which the entity or relationship is valid. The end time of an interval can be infinity; for example, if Shankar became a student in September 2018, and is still a student, we can represent the end time of the valid time in- terval as infinity for the Shankar entity. Similarly, we model attributes that can change over time as a set of values, each with its own valid time.

    a) Draw an E-R diagram with the student and instructor entities, and the ad- visor relationship, with the above extensions to track temporal changes.

    b) Convert the E-R diagram discussed above into a set of relations.

    It should be clear that the set of relations generated is rather complex, leading to difficulties in tasks such as writing queries in SQL. An alternative approach, which is used more widely, is to ignore temporal changes when designing the E-R model (in particular, temporal changes to attribute values), and to modify the relations generated from the E-R model to track temporal changes.

    Answer :


start_time
end_time
value
name
start_time
valid_time
student_id
end_time
value
dept_name
Student
tot_cred
start_time
value
end_time
end_time
start_time
advisor
valid_time
start_time
start_time
value
end_time
end_time
name
Instructor
instructor_id
dept_name
valid_time
salary
value
end_time
start_time
end_time
start_time
start_time
end_time
value

b)

The conversion of an E-R diagram into relational schemas involves creating separate relations for attributes that are multivalued or change over time. Each such relation is named by combining the entity name with the attribute name and includes fields for the entity's identifier, the attribute value, and time intervals (start and end times). For example, the relation `student_name(student_id, value, start_time, end_time)` records how a student's name evolves over time. Temporal aspects of relationships—such as `advisor`—are also captured using this method. Typically, the primary key for these temporal relations includes all four fields, unless time periods do not overlap, in which case the `end_time` may be omitted.

Although this approach provides an accurate temporal model, it increases schema complexity. To reduce this, designers often skip representing temporal attributes in the E-R diagram and instead manage temporal data directly in the relational model through history tables or timestamp columns.

Each time-varying or multivalued attribute is mapped to a relation with the format:

```
<entity>_<attribute>(entity_id, value, start_time, end_time)
```

Entity Relations:

```
student(student_id)
instructor(instructor_id)
```

Temporal Attribute Relations:

```
student_valid_times(student_id, start_time, end_time)
student_name(student_id, value, start_time, end_time)
student_dept_name(student_id, value, start_time, end_time)
student_tot_cred(student_id, value, start_time, end_time)
instructor_valid_times(instructor_id, start_time, end_time)
instructor_name(instructor_id, value, start_time, end_time)
instructor_dept_name(instructor_id, value, start_time, end_time)
instructor_salary(instructor_id, value, start_time, end_time)
```

Temporal Relationship:

```
advisor(student_id, instructor_id, start_time, end_time)
```

14. **Question :**
Explain the distinctions among the terms primary key, candidate key, and superkey.

Answer :

A **primary key** is a candidate key selected by the database designer to serve as the main identifier for tuples in a relation.

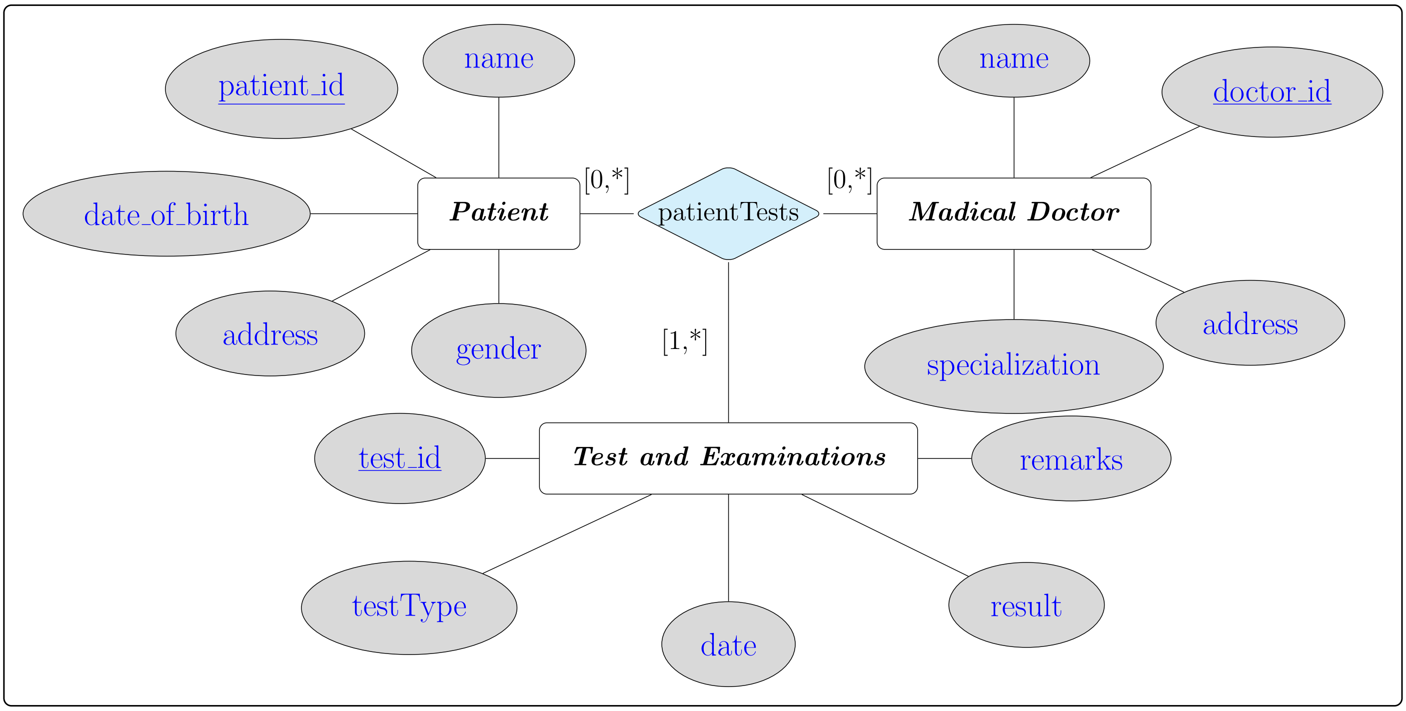
A **candidate key** is a minimal superkey, meaning that none of its proper subsets can uniquely identify tuples.

A **superkey** refers to a combination of one or more attributes that can uniquely distinguish each tuple in a relation.

15. **Question:**
Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.

Answer:

This E-R diagram represents a hospital management system involving patients, doctors, and medical tests. It includes three core entities: **Patient**, **Medical Doctor**, and **Test and Examinations**. The **Patient** entity has attributes such as *patient_id* (primary key), name, date of birth, address, and gender. The **Medical Doctor** entity includes *doctor_id* (primary key), name, specialization, and address. The **Test and Examinations** entity contains *test_id* (primary key), test type, date, result, and remarks. These entities are connected through a **ternary relationship** called *patientTests*, which indicates which doctor performed which test for which patient. Both patients and doctors can be involved in multiple test records, while each test must be associated with a patient and a doctor. This design effectively organizes medical records and test histories within a hospital context.



19. **Question:**

We can convert any weak entity set to a strong entity set by simply adding appropriate attributes. Why, then, do we have weak entity sets?

Answer:

While it is true that a weak entity set can be converted into a strong entity set by adding the primary key of its identifying entity set as part of its own attributes, weak entity sets exist for important modeling reasons. They help to naturally represent real-world scenarios where some entities cannot be uniquely identified without depending on another entity. For example, a *Dependent* entity (like a child or spouse) may not have a unique identifier without linking to an *Employee*. Using weak entity sets improves clarity, reflects dependency relationships, enforces referential integrity, and avoids redundancy by not duplicating identifying attributes unnecessarily. Thus, weak entity sets are a meaningful abstraction that enhances the semantic expressiveness of E-R diagrams.

21. **Question:**
Consider the E-R diagram in Figure 6.30, which models an online bookstore.

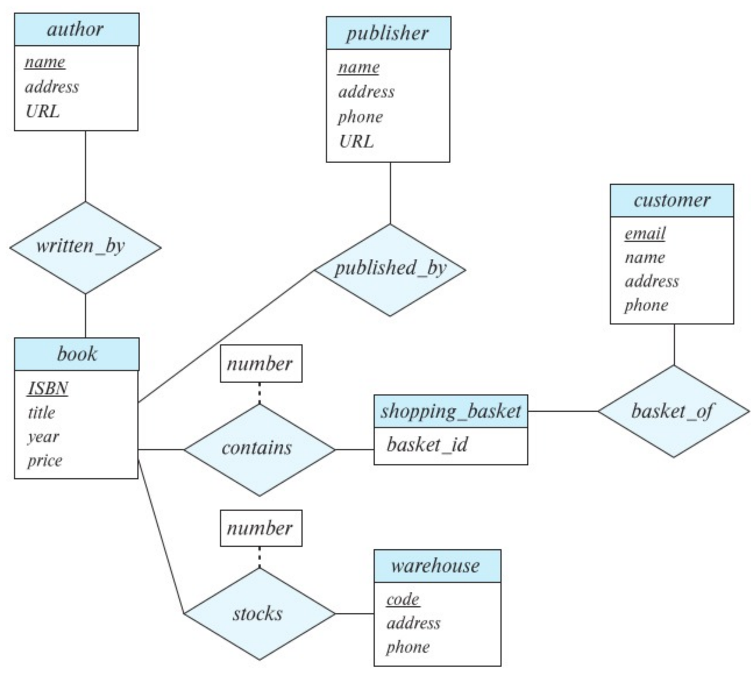


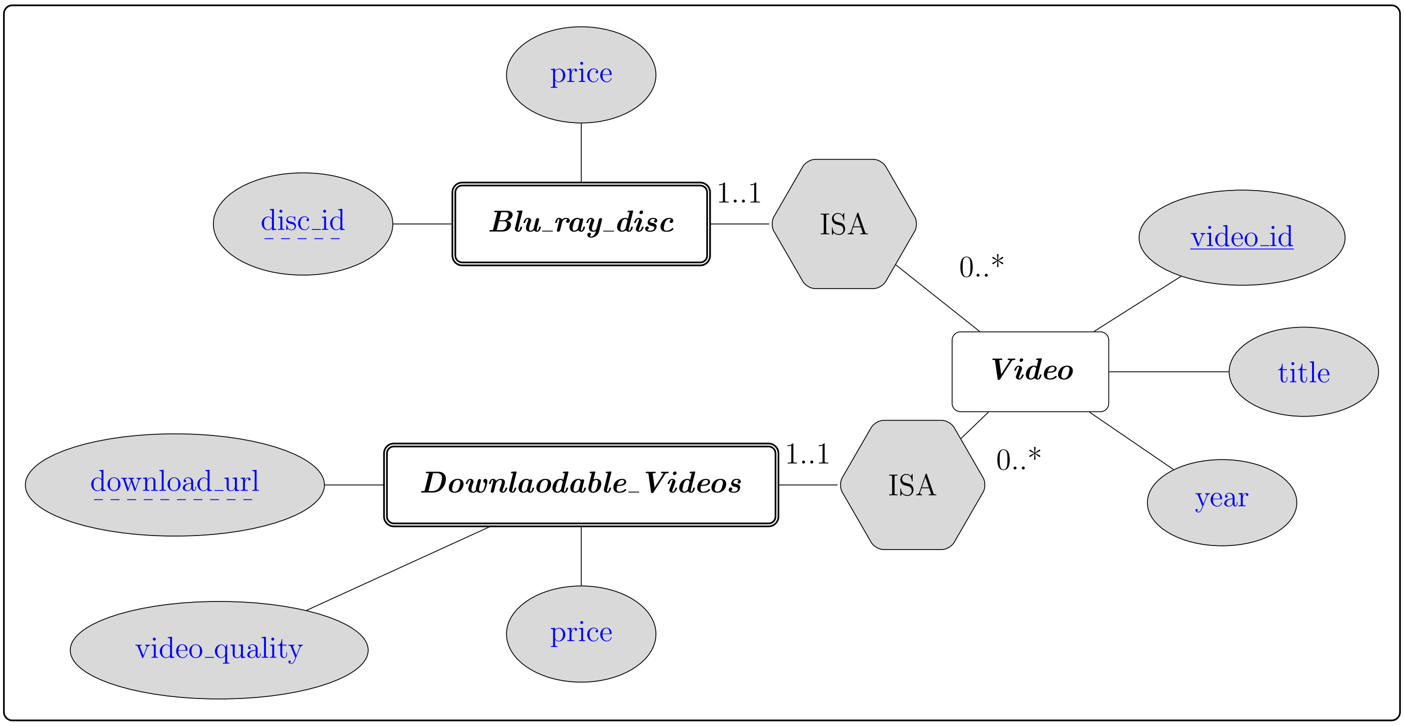
Figure 6.30 E-R diagram for modeling an online bookstore.

a) Suppose the bookstore adds Blu-ray discs and downloadable video to its collection. The same item may be present in one or both formats, with differing prices. Draw the part of the E-R diagram that models this addition, showing just the parts related to video.

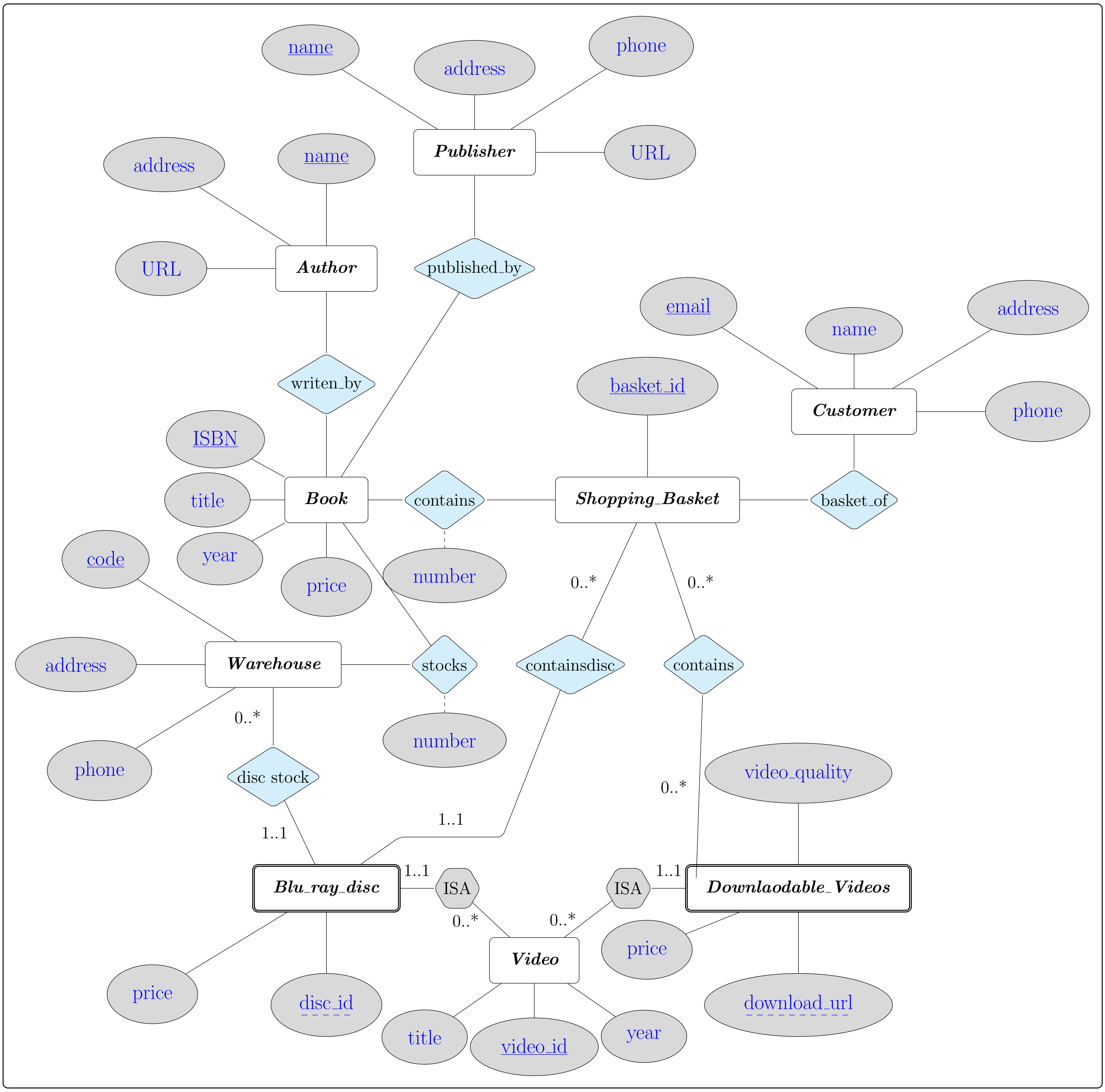
b) Now extend the full E-R diagram to model the case where a shopping basket may contain any combination of books, Blu-ray discs, or downloadable video.

Answer:

a) The E-R diagram incorporates video items by defining a general entity **Video** that holds common attributes such as video_id, title, and year. It connects to two specific sub-entities— **Blu_ray_disc** and **Downloadable_Videos**—through total **ISA relationships**. These sub-entities represent different formats of the same video content. The Blu_ray_disc entity contains attributes like disc_id and price, while the Downloadable_Videos entity includes attributes such as download_url, video_quality, and its own price. This design enables a video to be available in one or both formats, each with distinct characteristics, while maintaining shared information in a centralized manner to minimize redundancy.



b) An E-R diagram for a bookstore where baskets can hold books, Blu-ray-discs, and downloads:

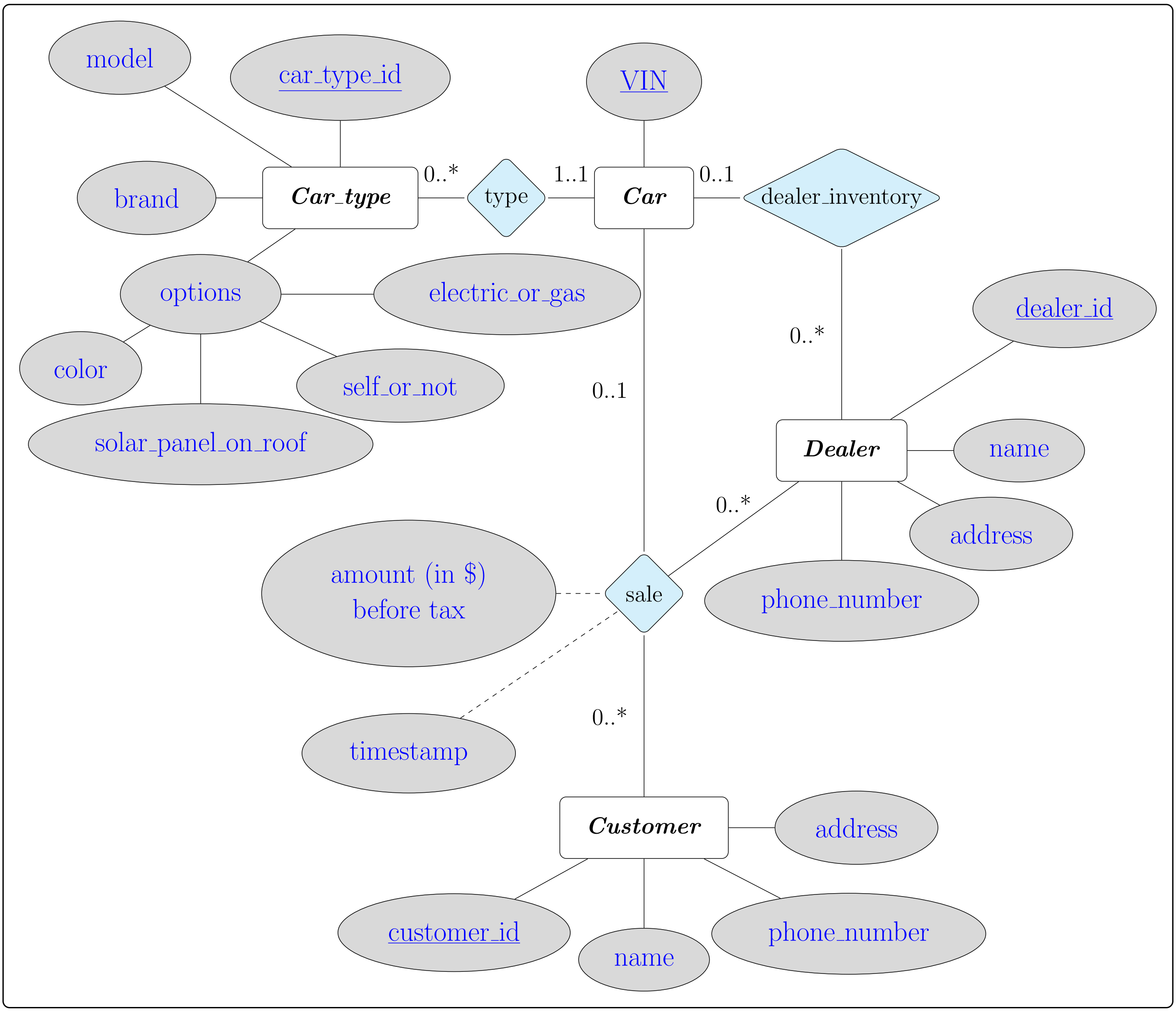


22. **Question:**

Design an E-R diagram for an automobile company to provide to its dealers to assist them in maintaining customer records and dealer inventory and to assist sales staff in ordering cars. Each vehicle is identified by a vehicle identification number (VIN). Each individual vehicle is a particular model of a particular brand offered by the company (e.g., the XF is a model of the car brand Jaguar of Tata Motors). Each model can be offered with a variety of options, but an individual car may have only some (or none) of the available options. The database needs to store information about models, brands, and options, as well as information about individual dealer

Answer:

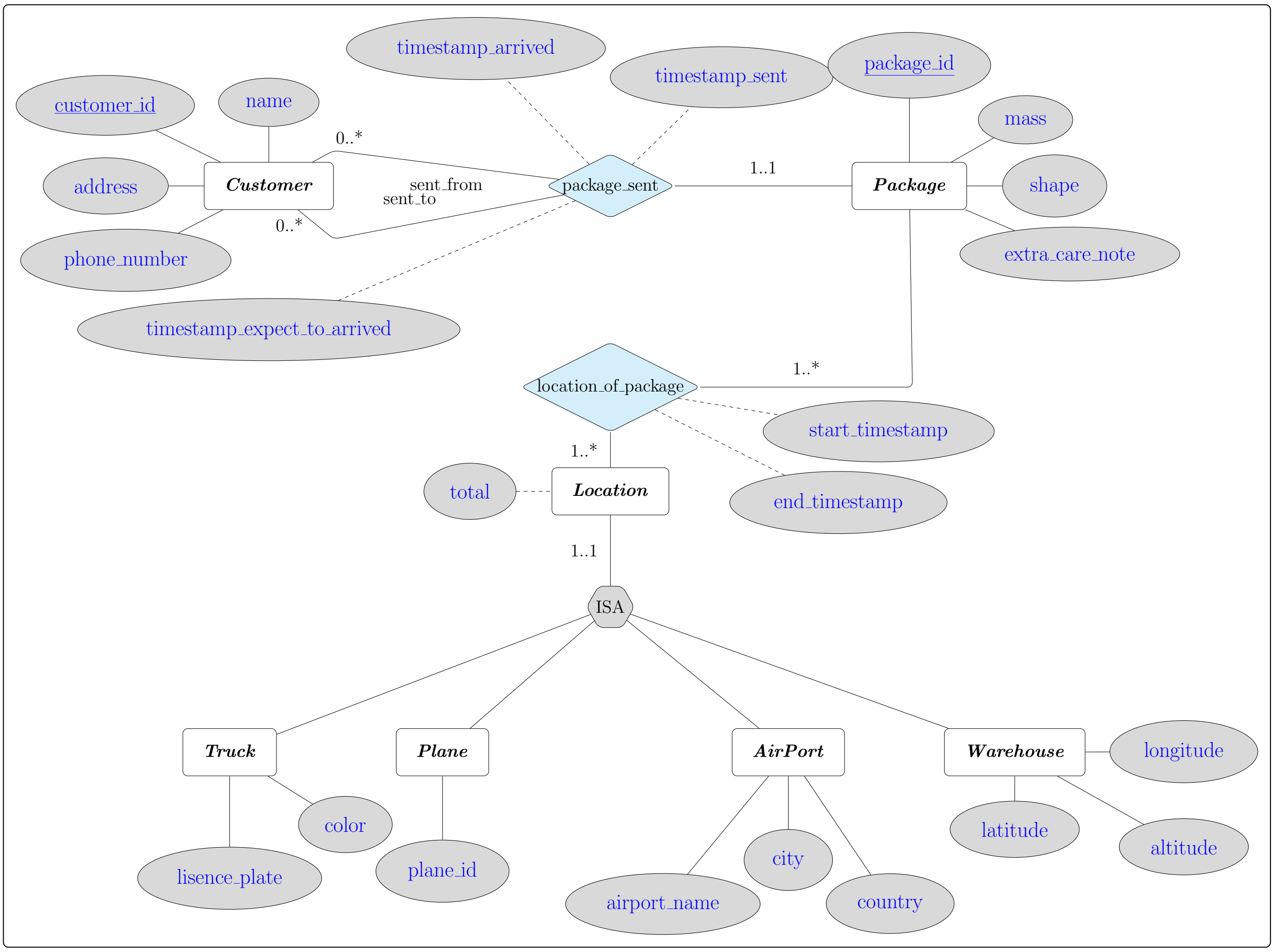
This E-R diagram represents a database design for an automobile company, tailored to manage cars, customers, dealers, inventory, and sales.



23. **Question:**

    Design a E-R diagram for an airline. The database must keep track of customers and their reservations, flights and their status, seat assignments on individual flights, and the schedule and routing of future flights.

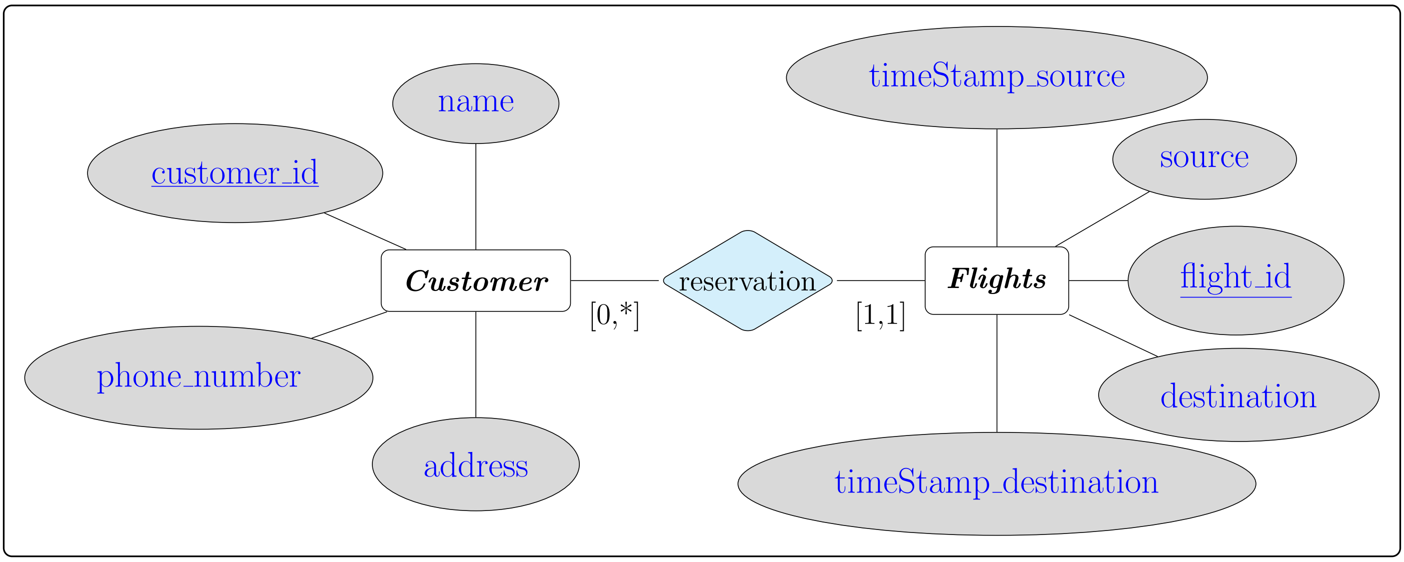
    Answer:
    The E-R diagram of Airline:



24. **Question :**

Design a database for an airline. The database must keep track of customers and their reservations, flights and their status, seat assignments on individual flights, and the schedule and routing of future flights. Your design should include an E-R diagram, a set of relational schemas, and a list of constraints, including primary-key and foreign-key constraints.

Answer :



26. **Question :**

Design a generalization–specialization hierarchy for a motor vehicle sales com- pany. The company sells motorcycles, passenger cars, vans, and buses. Justify your placement of attributes at each level of the hierarchy. Explain why they should not be placed at a higher or lower level.

Answer: