UNIVERSITY OF CHITTAGONG

Department of Computer Science & Engineering

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

Assignment_01

Topic: Entity Relationship Modelling

Course Title: DataBase System
Course Code: CSE - 413

Submitted To:

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

Submitted By:

Nilanjana Das Jui
ID: 23701011
Dept. of Computer Science & Engineering

Date of submission: June 10, 2025

Chapter - 6

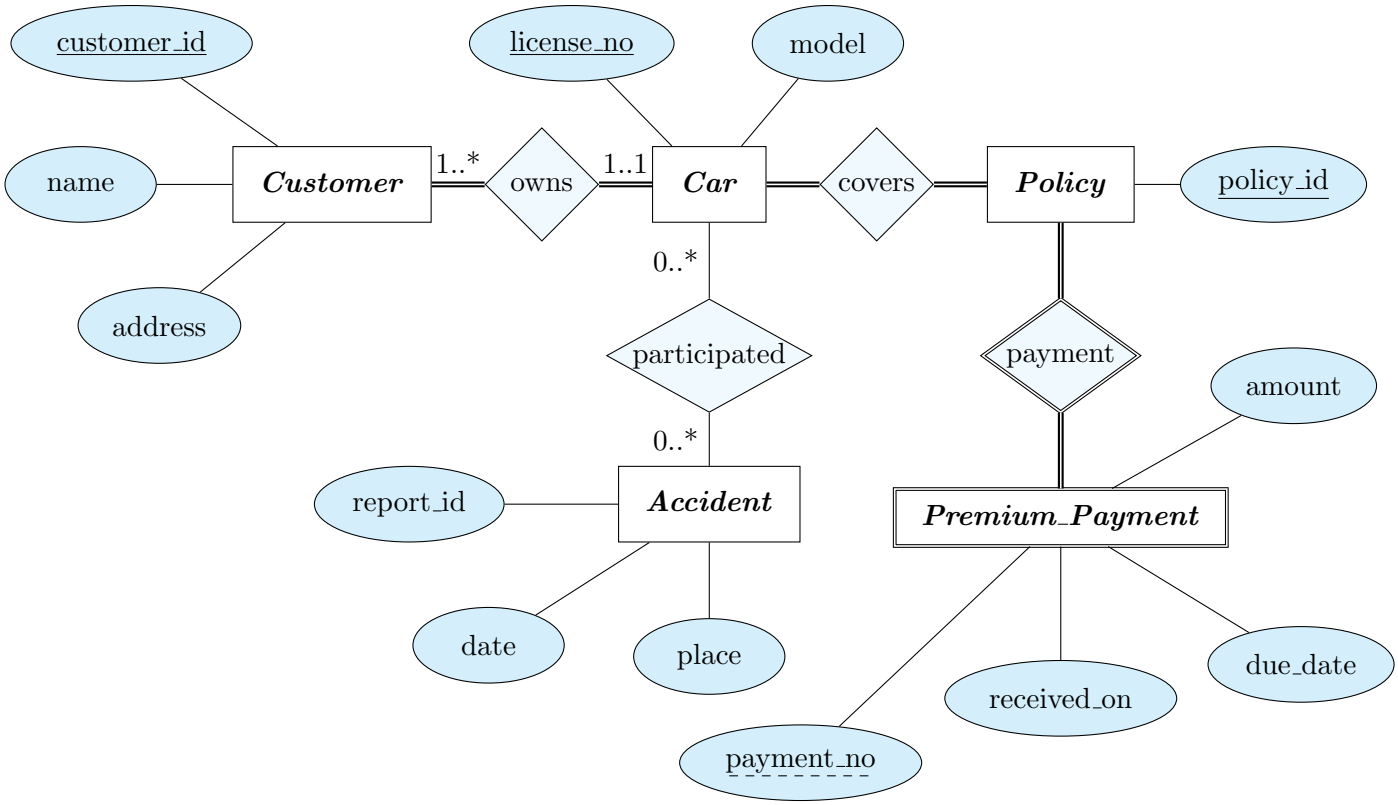
DataBase Design Using the E-R Model

1. **Question:**

   Construct an E-R diagram for a Car insurance company whose customers own one or more cars each. Each ar has associated with it zero to any number of recorded accidents. Each insurance policy overs 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 main entities of this E-R Diagrams are **Customer**, **Car**, **Accident**, **Policy**, and **Premium Payment**. Each customer can own one or more cars, and each car may be involved in zero or more accidents. Accidents have attributes like report ID, date, and place. Policies cover one or more cars, and each policy has one or more premium payments associated with it. A payment includes details such as due date, received date, and amount. The **Premium Payment** is a weak entity because it cannot exist without being linked to a policy. The key relationships in the diagram are **Owns** (between Customer and Car), **Participated** (between Car and Accident), **Covers** (between Policy and Car), and **Payment** (between Policy and 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.

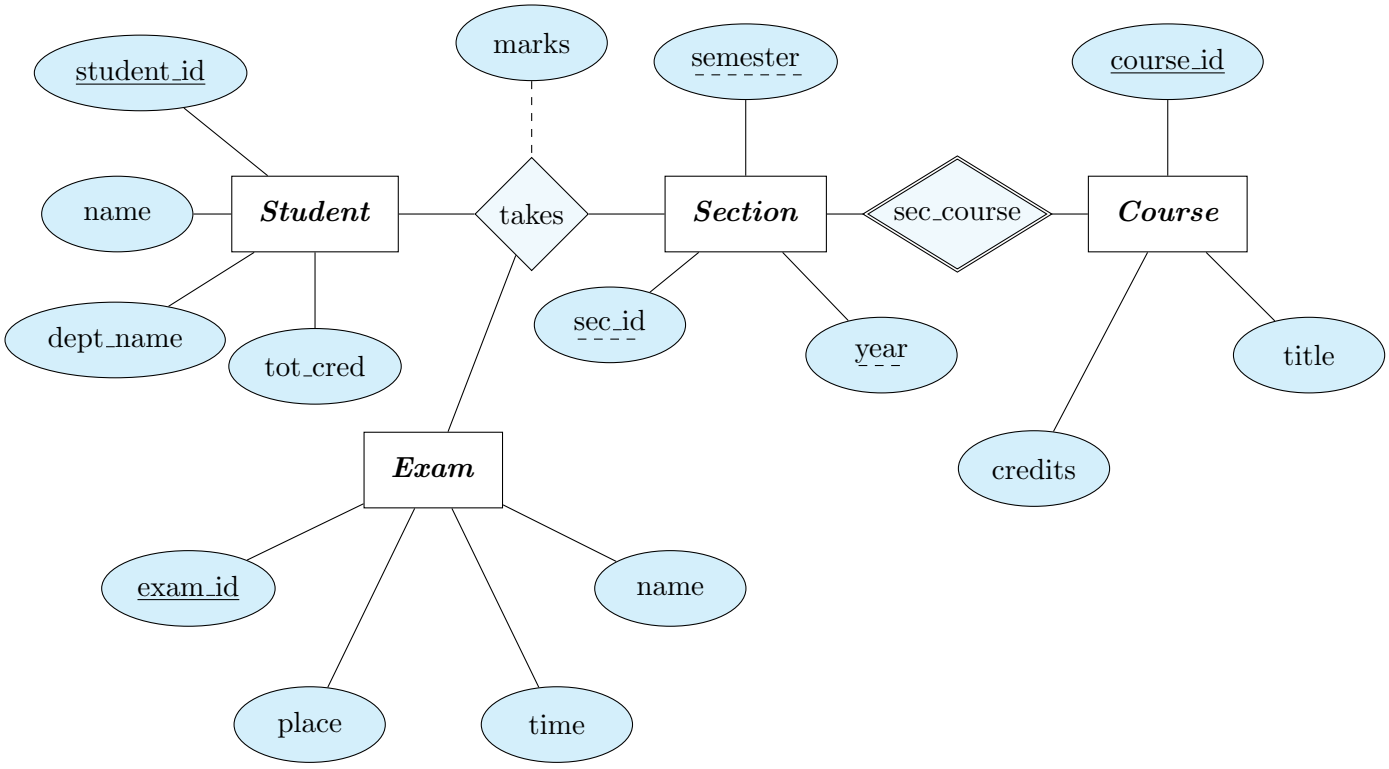
a) Construct an E-R diagram that models exams as entities and uses a ternary relationship as part of the design.

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:

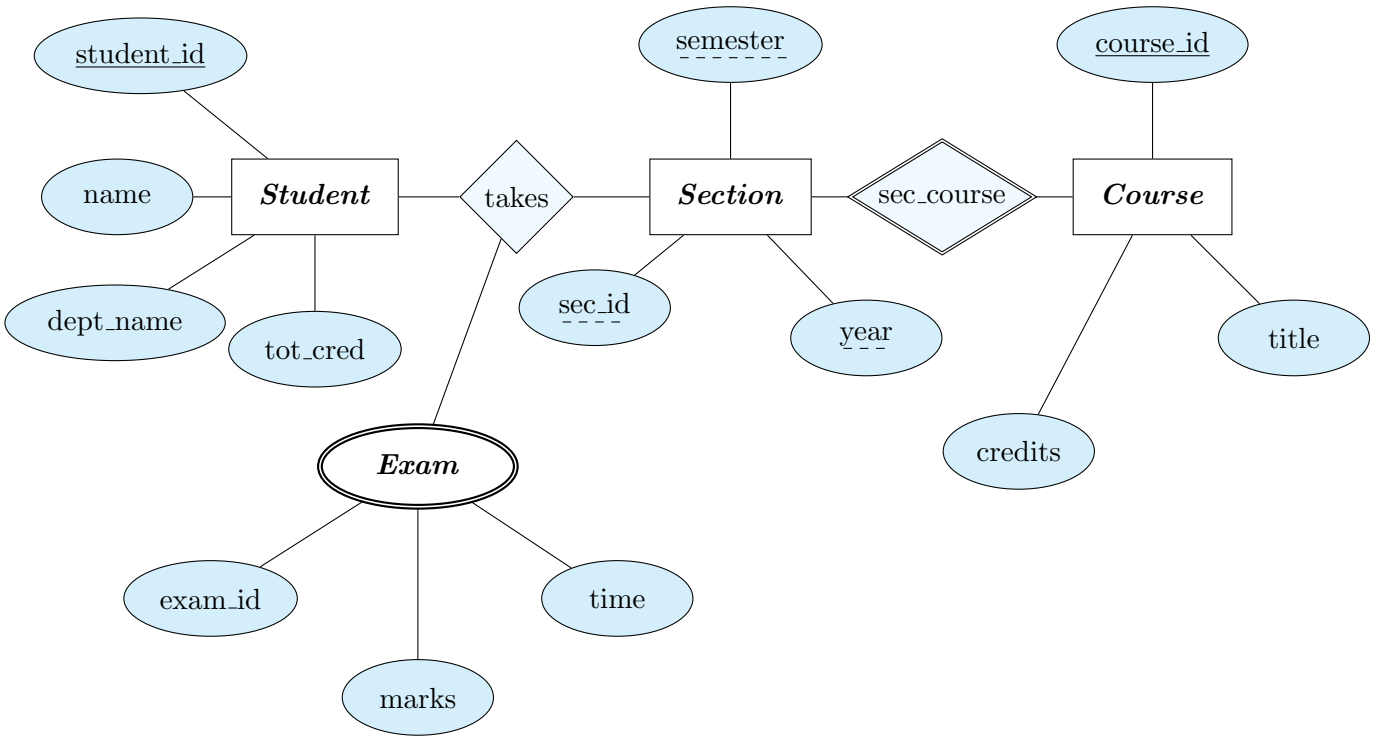
a) E-R Diagram with **Exams** as Entities and a Ternary Relationship:

In this E-R diagram, we consider **Student**, **Section**, and **Exam** as entity sets. Each **Exam** entity will have attributes like exam_id, name, place and time. To represent the marks that a student receives for a particular exam in a specific section, we introduce a ternary relationship named takes that connects **Student**, **Section**, and **Exam**. This ternary relationship includes an attribute marks to record the score. The ternary nature ensures that the relationship captures the context in which a student appears for a specific exam held in a specific section of a course. This design is useful when the same exam could exist across sections or when exams need to be treated as first-class entities.



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

In this E-R diagram, we limit ourselves to a binary relationship between **Student** and **Section** called takes. Since we want to avoid multiple relationships between the same student-section pair, we cannot directly store multiple exam marks in this binary relationship. To solve this, we embed **Exam** as a multi-valued composite attribute within the takes relationship. Each entry in this composite attribute can contain exam_id, time, and marks. This approach assumes that exams do not exist independently as entities and are only relevant within the student-section context. It simplifies the schema.

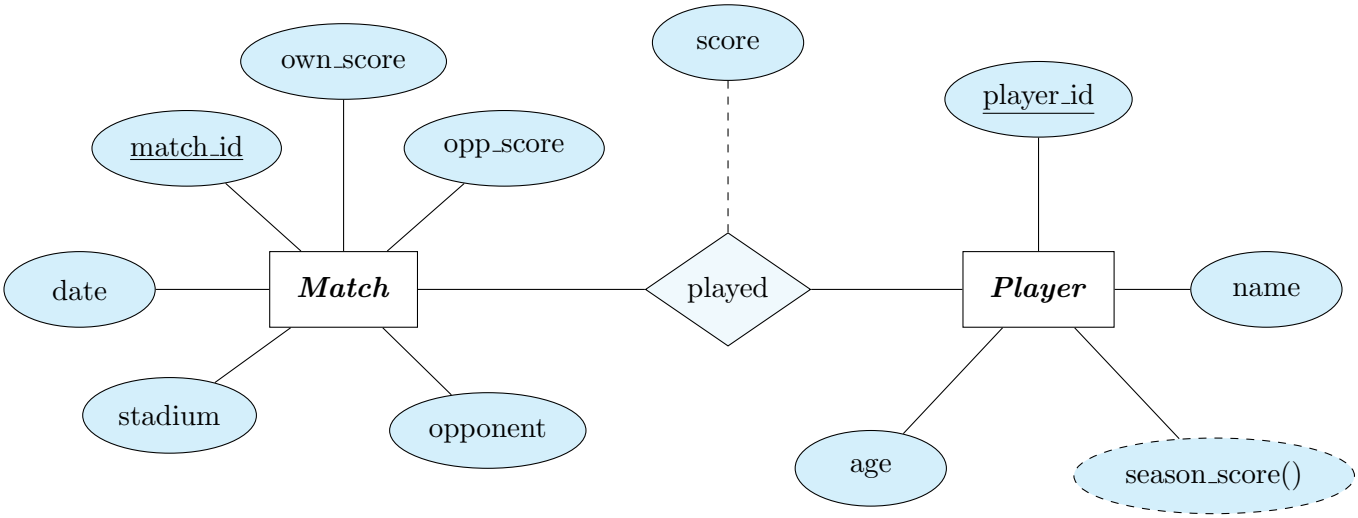


3. **Question:**

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 is designed to keep track of the scoring statistics of a sports team. It includes two main entities: **Match** and **Player**. The match entity stores details like match_id, date, stadium, opponent team, own team's score, and the opponent's score. The **Player** entity keeps information such as player_id, name, and age. There's also a **derived** attribute called **season_score()** in the **Player** entity, which represents the total score a player has in the season. This value isn't stored directly but is calculated by adding up the scores from all matches the player has participated in. The relationship played connects players to the matches they played in, showing which player took part in which match. The score attribute is part of this relationship and records how many points a player scored in a specific match. This design helps keep everything organized by linking players and matches while also making it easy to calculate individual and team performance.



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) If two entity sets in an E-R diagram are linked by a path, it means they are related, even if the connection is indirect. On the other hand, if the graph is disconnected, it shows that some entity sets have no relationship with each other at all. In terms of an enterprise, this means that these disconnected parts operate completely independently. If we divide the graph into its connected components, each component can be seen as a separate database representing an independent part of the enterprise.

b) A path between two entity sets in an E-R diagram represents a relationship between them, which might be direct or indirect. When a cycle exists in the graph, it means that entity sets within the cycle are connected through more than one distinct path, indicating multiple relationships. On the other hand, if the diagram is acyclic, there is only one path connecting any two entity sets, meaning each pair shares a single, unique relationship.

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:

If we convert a weak entity set into a strong entity set by including the primary key attributes of its identifying entity set as part of its own attributes, this will introduce redundancy because those key attributes will be stored repeatedly for every weak entity instance. Essentially, the identifying entity's key values get duplicated across all related weak entities, increasing data duplication and storage requirements. This redundancy can lead to anomalies during data updates, such as inconsistencies if the identifying key changes, and makes the database less efficient to maintain.

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:

If the advisor relationship is one-to-one, it means each advisor can only have one student, and each student can only have one advisor. To make sure this happens in the database, we need to add extra rules. First, the advisor's ID should be the primary key so it's unique. We also need to make sure the student's ID is unique too. This means no student can appear more than once in the advisor table. By making both IDs unique, the database will enforce that each advisor is linked to only one student and each student has only one advisor, keeping the one-to-one relationship intact.

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: a)

The E-R diagram models temporal data by keeping primary key attributes like **student_id** and **instructor_id** immutable, while allowing all other attributes to change over time. Time-varying attributes such as **name**, **dept_name**, **tot_cred**, and salary are represented as **multivalued composite attributes** with subattributes like **value**, **start_time**, and **end_time**. The value subattribute holds the actual data (e.g., name or salary) valid during a specific time interval. The advisor relationship also includes a **valid_time** composite attribute to indicate when the advising connection was active. This structure enables the system to track changes over time and answer temporal queries effectively.

b)

The E-R diagram is converted into relational schemas by turning each multivalued or time-varying attribute into a separate relation. These are named by combining the entity and attribute names, and include the entity's ID, attribute value, start time, and end time. For instance, student_name (student_id, value, start_time, end_time) tracks changes in a student's name over time. Relationships like advisor are also represented with temporal attributes. By default, the primary key includes the ID, value, start time, and end time, though end_time can be dropped if time intervals don't overlap. While this approach captures time-based changes accurately, it leads to a complex schema. To simplify, designers often ignore temporal aspects in the E-R model and handle time directly in the relational schema using history tables or timestamps.

Each multivalued or time-varying attribute is converted into a separate relation, named using the pattern:

$$< entity > \_ < attribute > (entity\_id, value, start\_time, end\_time)$$

Entities:

$$student(student\_id)$$
$$instructor(instructor\_id)$$

Temporal Attributes:

$$student\_valid\_times(student\_id, start\_time, end\_time)$$
$$student\_name(student\_id, value, start\_time, end\_time)$$
$$student\_dept\_name(student\_id, value, start\_time, end\_time)$$
$$student\_tot\_cred(student\_id, value, start\_time, end\_time)$$
$$instructor\_valid\_times(instructor\_id, start\_time, end\_time)$$
$$instructor\_name(instructor\_id, value, start\_time, end\_time)$$
$$instructor\_dept\_name(instructor\_id, value, start\_time, end\_time)$$
$$instructor\_salary(instructor\_id, value, start\_time, end\_time)$$

Relationship with Temporal Attributes:

$$advisor(student\_id, instructor\_id, start\_time, end\_time)$$

14. **Question:**

Explain the distinctions among the terms primary key, candidate key, and superkey.

Answer:

(a) *SuperKey:*

- A superkey is any set of one or more attributes that can uniquely identify a row in a relation or table.
- It might include extra attributes that are not necessary for uniqueness.
- Every relation has at least one superkey - the entire set of attributes is always a superkey (because together they uniquely identify tuples).

(b) *Candidate Key:*

- A candidate key is a minimal superkey - meaning it is a superkey with no unnecessary attributes.
- It contains the smallest number of attributes needed to uniquely identify tuples.
- A relation can have one or more candidate keys.

(c) *Primary Key:*

- A primary key is one of the candidate keys chosen by the database designer as the main means to uniquely identify tuples.
- It is a candidate key designated for practical use, often because it is simple, stable, or efficient.

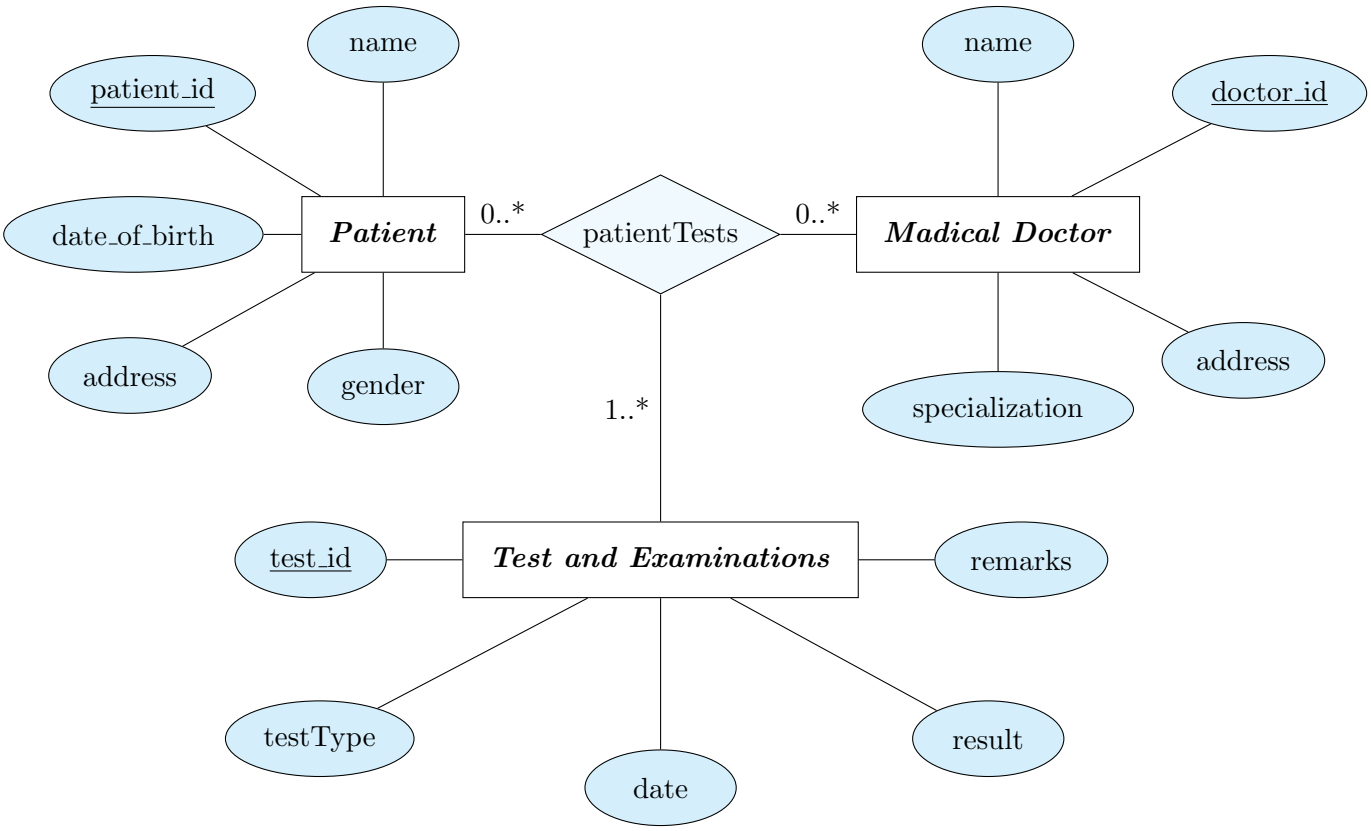
- There can be only one primary key per relation.
- Other candidate keys that are not chosen as primary are called alternate keys.

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 models a hospital system that tracks patients, medical doctors, and medical tests. It includes three main entities: **Patient**, **Medical Doctor**, and **Test and Examinations**. The Patient entity includes attributes like patient_id which is a primary key, name, date_of_birth, address, and gender. The Medical Doctor entity has doctor_id (primary key), name, specialization, and address. The Test and Examinations entity holds test_id (primary key), test type, date, result, and remarks. These entities are linked through a **ternary relationship** named patientTests, which records which doctor conducted which test for which patient. Patients and doctors can be linked to multiple test records, while each test must be associated with at least one patient and doctor. This structure effectively captures patient history and test details in a hospital environment.



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:

Altough we can convert any weak entity set into a strong entity set by adding appropriate attributes by including a foreign key from the related strong entity, weak entity sets are still used in E-R modeling for important conceptual reasons.

- Weak entities represent objects that cannot exist independently without being related to a strong entity.
- They make the model more intuitive and meaningful, clearly showing real-world relationships and dependencies.
- Converting a weak entity to a strong one may require unnatural or complex keys, which can clutter the design and reduce readability.
- Weak entities express scenarios where the existence of one entity relies on another, reinforcing referential integrity.
- They support a cleaner, more normalized structure and make it easier to enforce data integrity constraints.

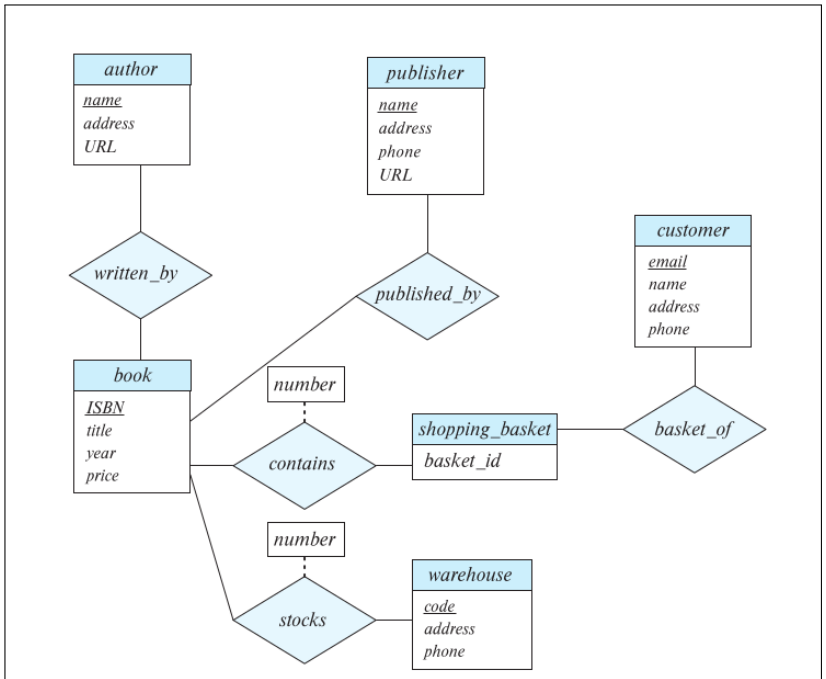


Figure 6.30 E-R diagram for modeling an online bookstore.

21. **Question:**

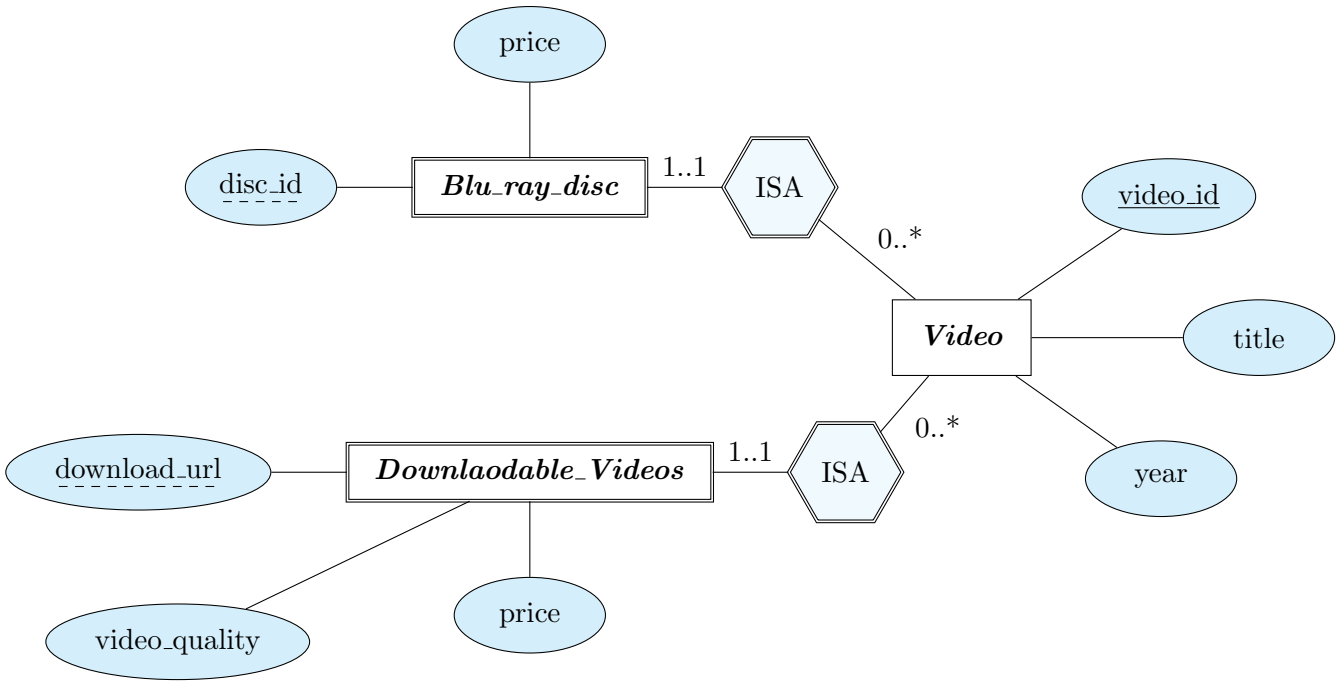
Consider the E-R diagram in Figure 6.30, which models an online bookstore.

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.

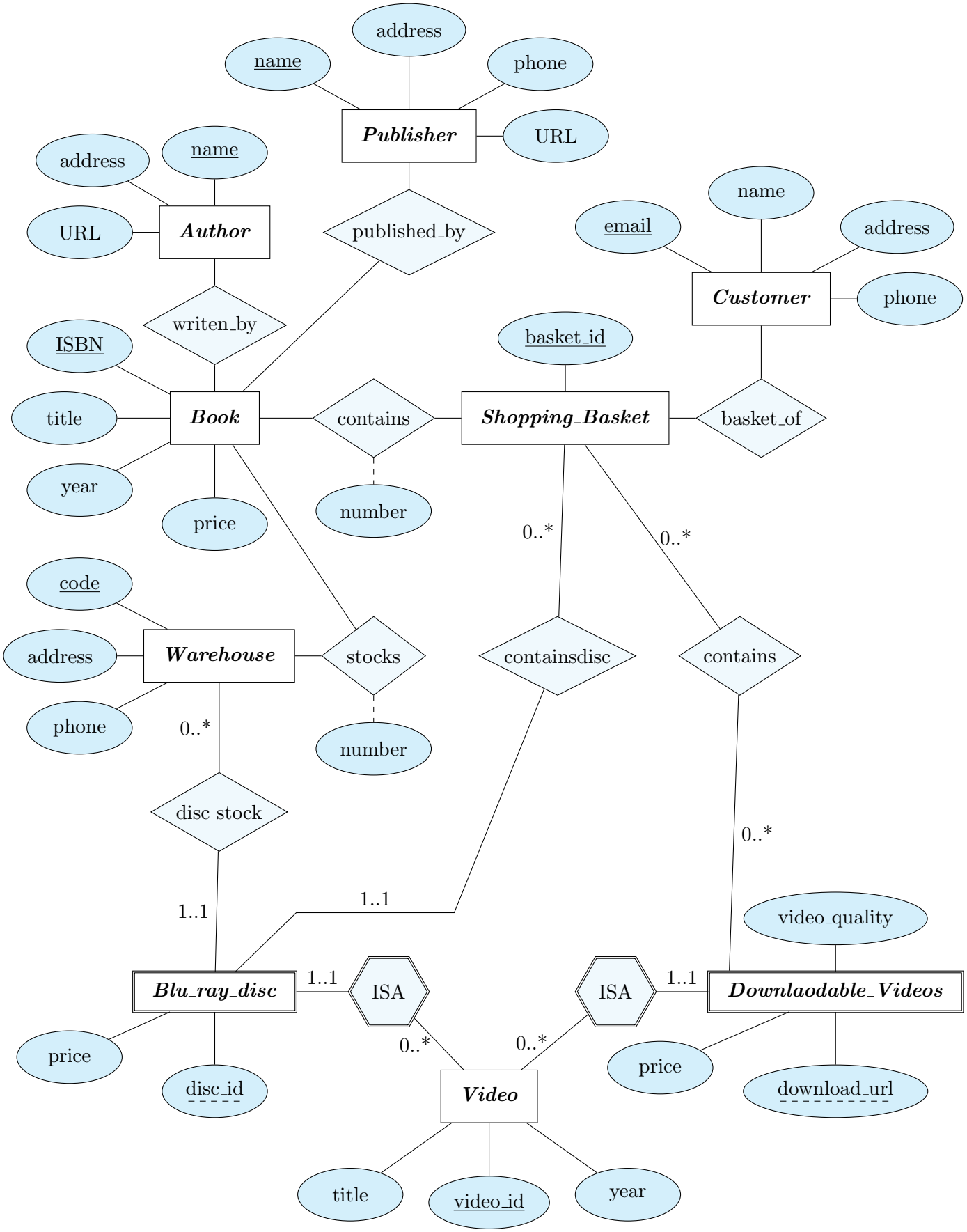
b) Now extend the full E-R diagram to model the case where a shopping basket may contain any combination of books, Blu-ray discs, or downloadable video.

Answer: a)

The E-R diagram adds support for video items by introducing a general entity **Video** with shared attributes: video_id, title, and year. Two specialized sub-entities - **Blu_ray_disc** and **Downloadable_Videos** are connected to Video through total **ISA relationships**. These represent different formats of the same content. Blu_ray_disc includes disc_id and price, while Downloadable_Videos includes download_url, video_quality, and its own price. This structure allows the same video to exist in one or both formats, each with unique attributes and pricing, while avoiding data duplication by centralizing common details in the Video entity.



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

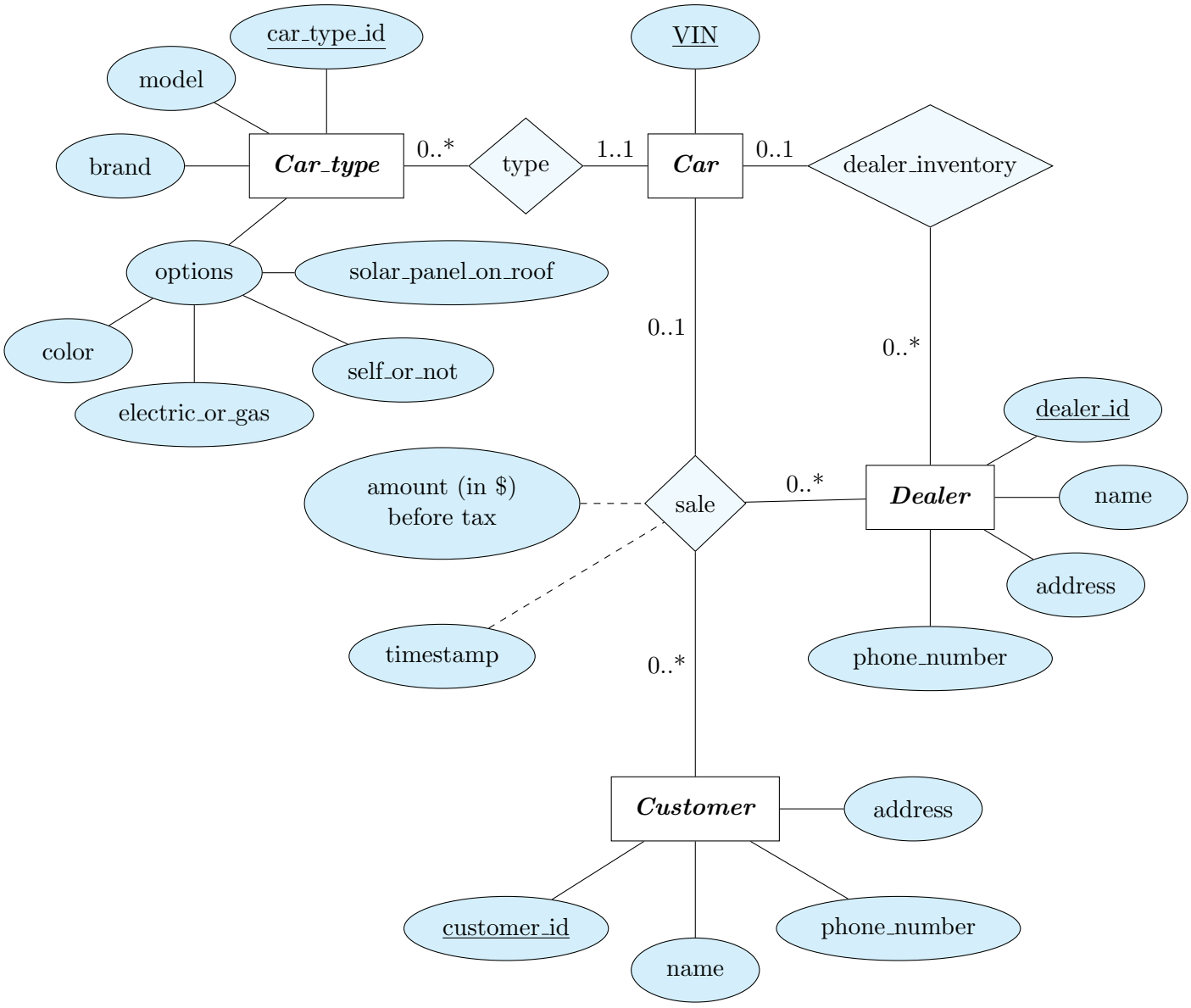


22. **Question:**

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 dealers, customers, and cars.

Answer:

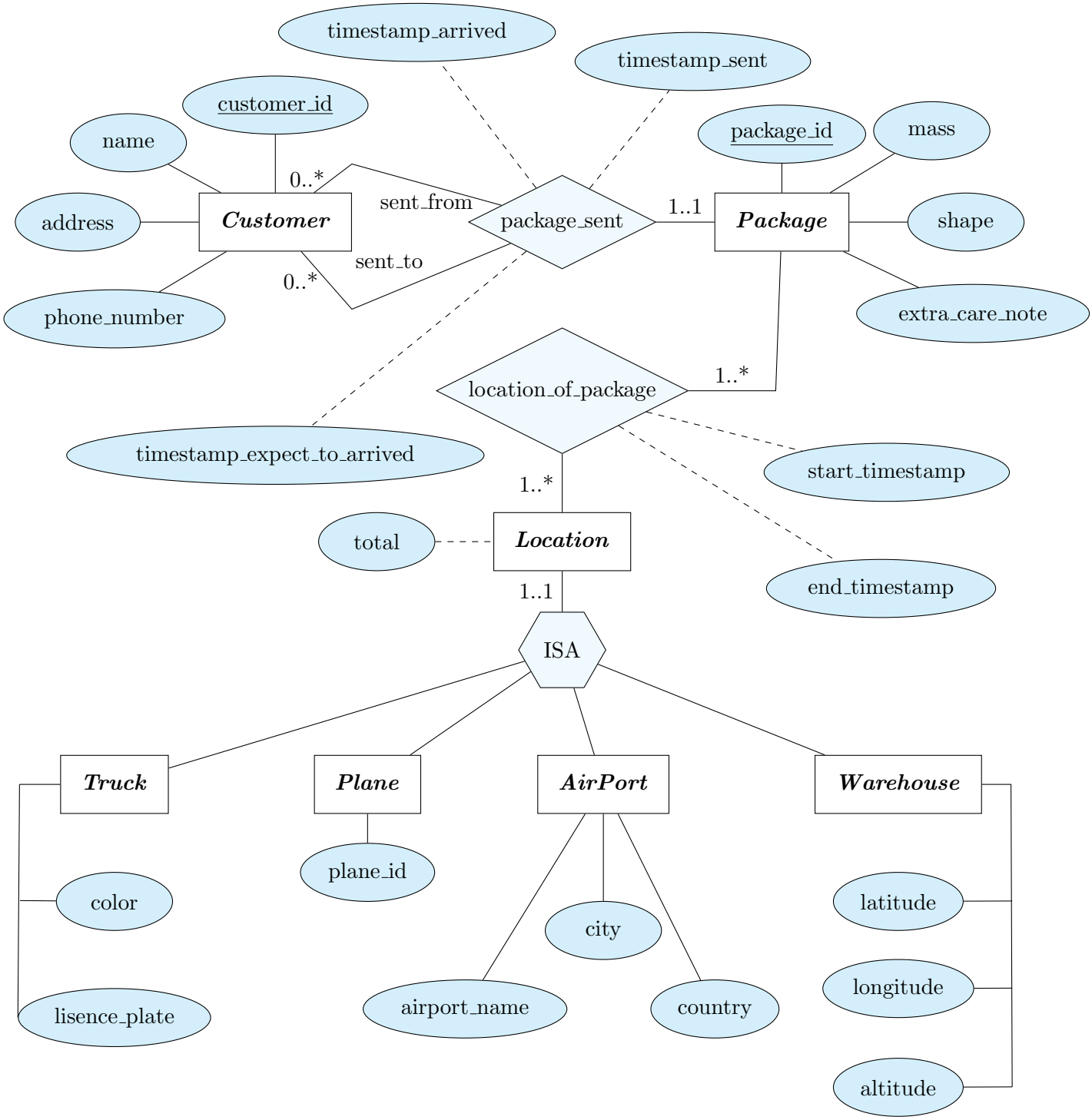
This E-R diagram represents a database design for an automobile company, tailored to manage cars, customers, dealers, inventory, and sales.



23. **Question:**

Design an 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:

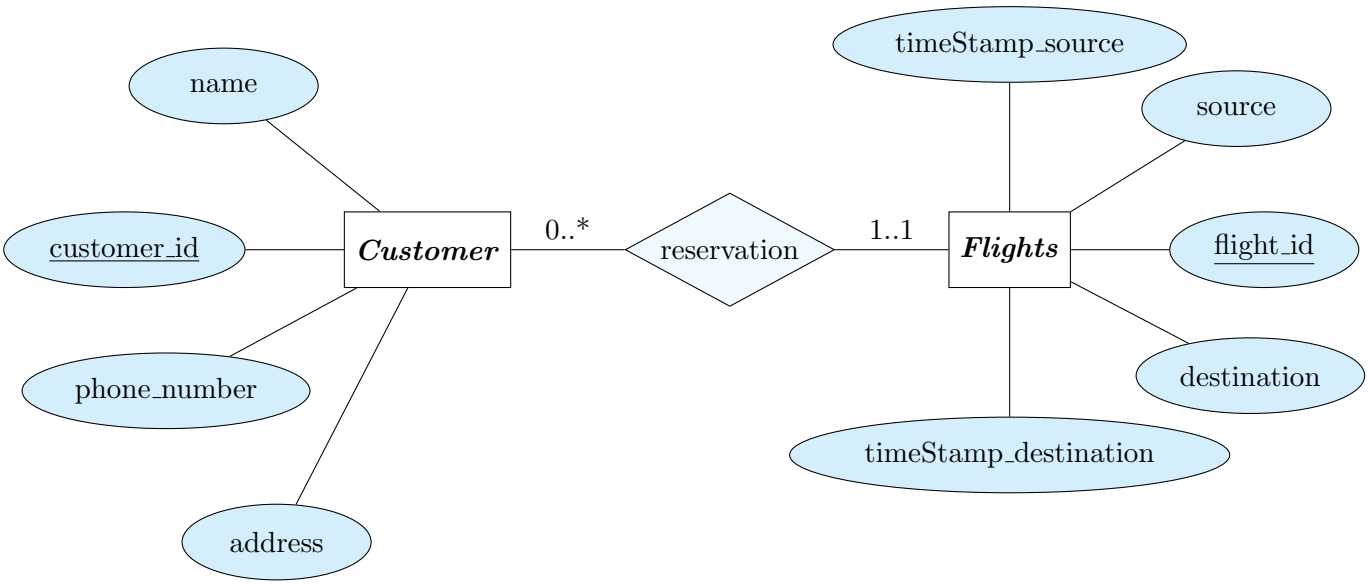


24. **Question :**

Design an 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 ER diagram models an airline database with two main entities: **Customer** and **Flights**. The Customer entity includes customer_id (primary key), name, phone_number, and address. The Flights entity contains flight_id (primary key), source, destination, and timestamps for departure and arrival. These are linked by a reservation relationship, where each customer can have multiple reservations, and each reservation is associated with exactly one flight (cardinality 0..* to 1..1).



26. **Question:**

Design a generalization–specialization hierarchy for a motor vehicle sales company. 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:

The ER diagram presents a generalized structure for vehicles using an ISA hierarchy. The main entity, **Vehicle**, includes shared attributes such as vehicle_id (primary key), max_speed, price, model, company, and made_in. The **ISA** relationship connects this entity to four subtypes **Motorcycles**, **Passenger_Cars**, **Vans**, and **Buses**, each representing a specialized category of vehicles. These sub-entities inherit the general attributes of Vehicle and also have their own specific attributes, such as diameter_of_motor_tyre for Motorcycles. This structure supports organized data storage by separating common and specific vehicle characteristics while maintaining a clear hierarchical relationship.