UNIVERSITY OF CHITTAGONG

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Database Systems Exercises

Database Systems Exercises

Course Information

Course Code: 413

Course Title: Database Systems

Semester: 4th

Submitted By:

Md. Abdullah Al Mamun Emon
ID: 23701028
GitHub: emon4075

Submitted To:

Dr. Rudra Pratap Deb Nath
Associate Professor
Department of Computer Science & Engineering
University of Chittagong
Chittagong-4331

June 10, 2025

Table of Contents

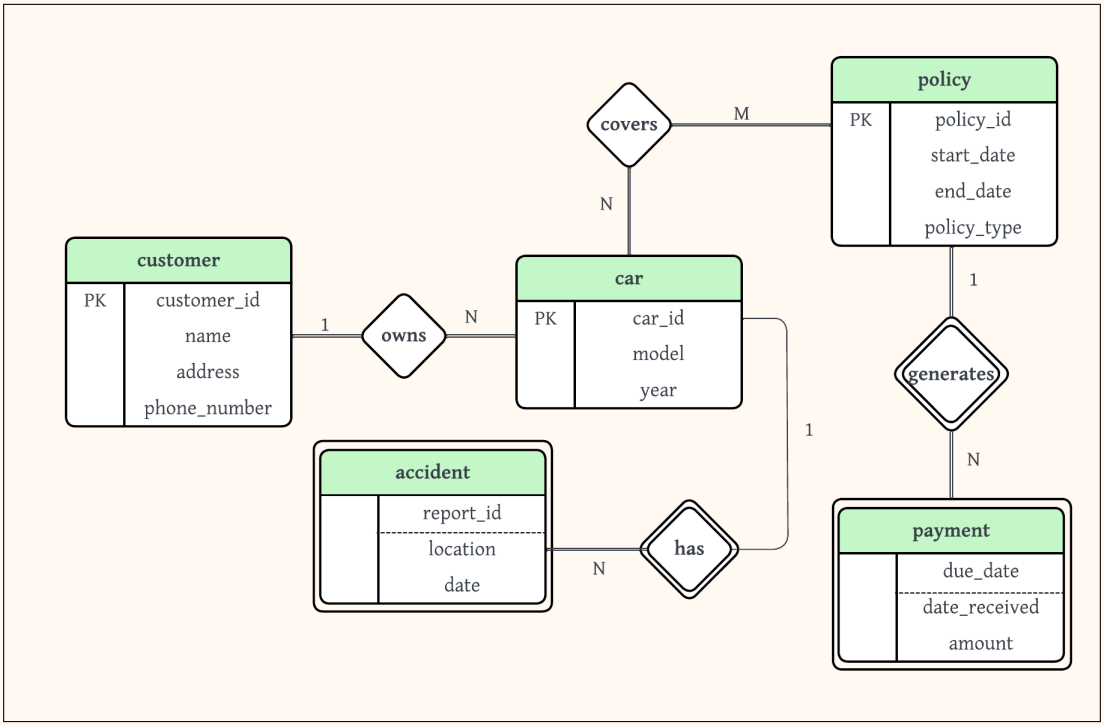
Contents

1 Chapter 6 - Database Design Using E-R Model

Question 6.1

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 6.1



Note: This answer contains an E-R diagram showing the relationships between Customer, Car, Accident, Insurance Policy, and Premium Payment entities with proper cardinalities and attributes as specified in the question.

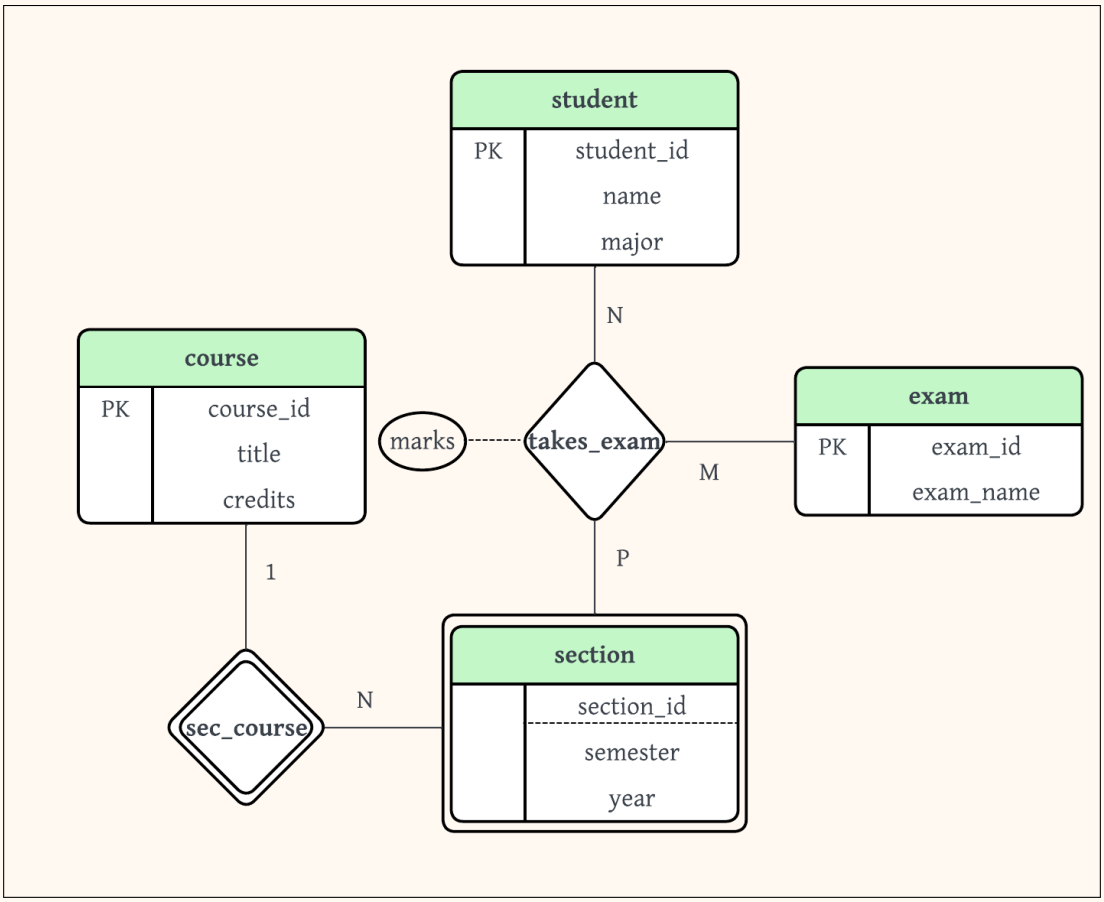
Question 6.2

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.

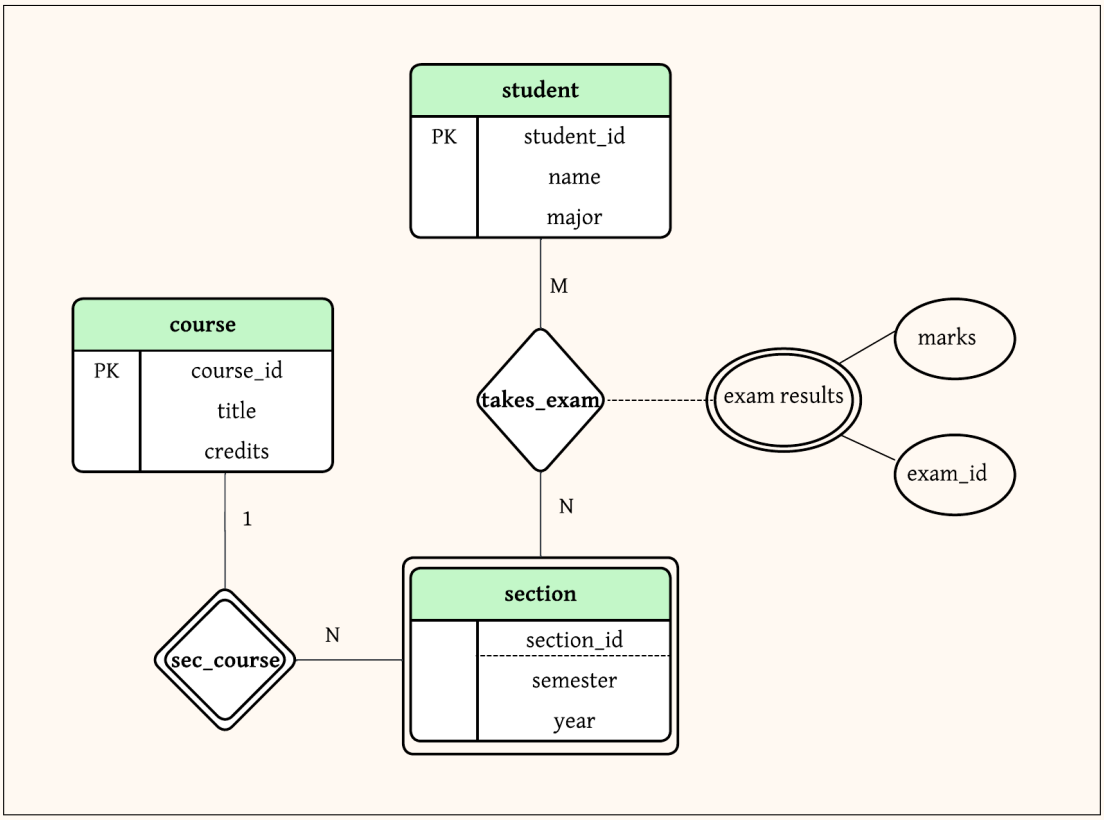
b. Construct an alternative E-R diagram that uses only a binary relationship between student and section. Make sure that only one relationship exists between a particular student and section pair, yet you can represent the marks that a student gets in different exams.

Answer 6.2 (a)



Note: This diagram shows a ternary relationship between student, exam, and section entities as requested in part (a).

Answer 6.2 (b)

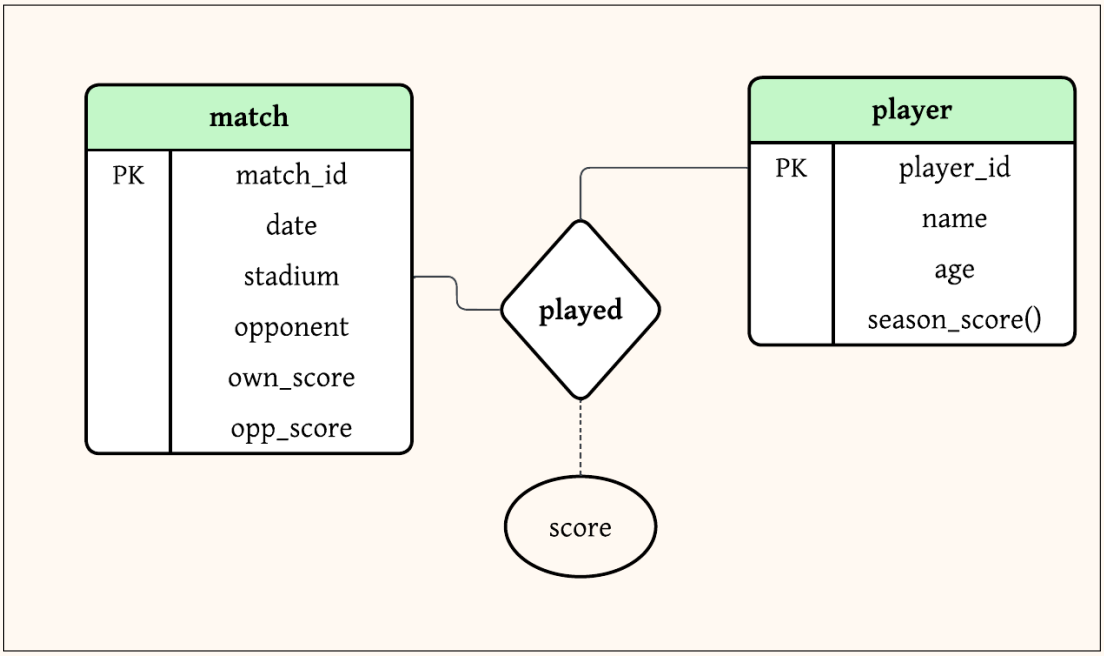


Note: This diagram shows an alternative design with binary relationships that still captures exam marks through proper entity modeling as requested in part (b).

Question 6.3

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 6.3



Note: This answer contains an E-R diagram for a sports team scoring statistics system, including entities for matches, players, and their relationships. The diagram shows how individual player statistics are tracked for each match, with derived attributes for summary statistics and explanations of their computation methods.

Question 6.5

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 6.5

a. The graph is disconnected.
This means the enterprise schema is partitioned into two or more independent sets of entities that have no relationships with each other. It represents distinct, unrelated parts of the business being modeled within the same database. For example, a company's HR system (Employees, Departments) might be completely separate from its Product Inventory system (Products, Warehouses).

b. The graph has a cycle.
This means there are multiple relationship paths between entities in the schema. This is common and often necessary to model complex, redundant relationships. For example, an Employee may work_for a Department, and a Department is also managed_by an Employee, creating a cycle. It indicates a higher degree of interconnectedness between the entities.

Question 6.7

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 6.7

When you add the owner's main key characteristics to a weak entity to make it a strong entity, it makes the database redundant.
This is because the owner-provided identifying information (like course title or credits) is now in every record of the weak entity (like parts of a course). Because of this, the same data is copied over and again.
For example, if the course CS101 - Database Systems contains five separate portions, the course title and credits will be saved five times, one for each section. Not only does this take up space, but it also makes it hard to keep track of changes. For example, if the course title changes, all five section records would need to be updated.
This kind of repetition is against the rules of database normalization, especially the Second Normal Form (2NF). It also makes it more likely that data will be inconsistent and that maintenance will be more difficult.
Example Comparison:
Before (Normalized):

```
course(course_id, title, credits)
section(course_id, section_id, semester)
```

After (Redundant):

```
section(course_id, title, credits, section_id, semester)
```

Question 6.9

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 6.9

The database must maintain a rigorous one-to-one connection to make sure that each student has only one adviser and that each advisor is only allocated to one student.
Without these kinds of limitations, problems like giving more than one adviser to the same student or more than one student to the same advisor may happen. For instance:

- A student like John may be connected to both Advisor Smith and Advisor Jones.
- Advisor Smith might work with more than one student, like John and Mary.

To prevent these things from happening, two rules are needed:

1. The primary key is student_id, which makes sure that a student only shows up once and that there aren't more than one adviser for each student.
2. Unique restriction on instructor_id makes guarantee that each instructor is only allocated to one student.

These two principles work together to make sure that the one-to-one connection is correct. If either one is absent, the connection might go from one to many in one direction.
Example:

student_id	instructor_id	Status
12345	Smith	OK
67890	Jones	OK
11111	Smith	Not allowed - Smith already advising
12345	Brown	Not allowed - Student already has advisor

Question 6.13

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 interval 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 advisor 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 6.13

[NOT YET SOLVED]

Question 6.14

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

Answer 6.14

Keys are employed in a database to make sure that records are unique. The differences are in how few and specific these keys are:

- A **superkey** is any set of properties that can uniquely identify a record. It could have additional, unneeded features.
  For example, {`student_id, name`} is a superkey even if `student_id` is enough on its own.
- A **candidate key** is a **minimal superkey**; that is, it doesn't have any other qualities. A table may have more than one candidate key.
  Example: Both {`student_id`} and {`ssn`} can uniquely identify a student and are the least amount of information needed.
- A **primary key** is **one selected candidate key** that is utilized as the principal identifier in the table. It is chosen for indexing, connections, and limits.
  Example: If both `student_id` and `ssn` are candidate keys, selecting `student_id` as the main key makes `ssn` an alternative key.

Visual Hierarchy:

"Superkeys $\supseteq$ Candidate Keys $\supseteq$ Primary Key"
Every main key is a candidate key, and every candidate key is a superkey. However, not all superkeys are candidate keys, as some may be redundant.

An Example with a Student Entity

Attributes: {`student_id, ssn, name, email`}

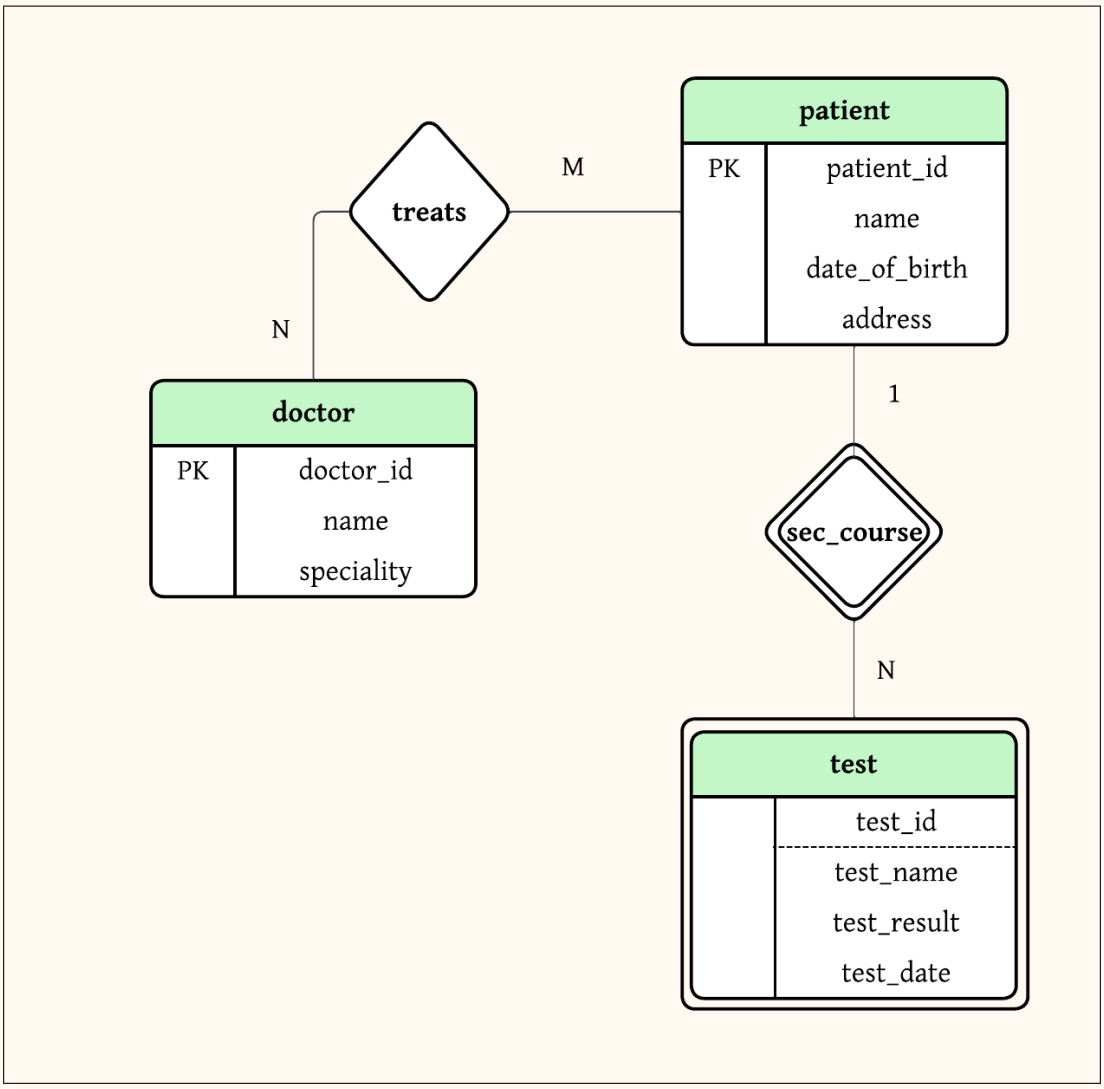
- **Superkeys**: {`student_id`}, {`ssn`}, {`student_id, name`}, {`ssn, email`}, and so on.
- **Candidate Keys**: {`student_id`}, {`ssn`} (unique and minimum)
- **Primary Key**: {`student_id`} (picked from the list of applicants)

This structure helps keep data safe and makes sure that every record can be found using a reliable and short key.

Question 6.15

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 6.15



Note: This answer contains an E-R diagram for a hospital system showing the relationships between Patient and Doctor entities, with a comprehensive log of tests and examinations. The diagram includes proper entities for tracking medical procedures, test results, and patient-doctor interactions as specified in the question.

Question 6.19

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 6.19

Weak entity sets are significant because they effectively describe real-world relationships and help maintain a clean, efficient database architecture. It is theoretically feasible to turn weak entities into strong ones by adding extra properties, including owner identifiers.

Main Reasons:

1. **Shows Real-World Dependency**: Some things naturally rely on other things. A *room* only exists in a certain *building*, for example. Modeling this using a weak object makes that reliance evident.
2. **Reduces Redundancy**: If there were no weak entities, you would have to copy the owner's characteristics into every connected record. For instance, each *section* would have to include all of the *course* information, such as the title and credits. This would waste space and might cause problems.
3. **Protects Data Integrity**: Weak entities can't exist without their owners. The database automatically enforces this existence dependence, which helps preserve **referential integrity**.
4. **Leads to Cleaner Schema Design**: Tables remain compact and on topic. Instead of putting all the information together in a messy way, qualities are rationally separated and neatly arranged.
5. **Makes Maintenance Easier**: When you need to change shared information, like the title of a course, you just need to do so in one place. Weak entity architecture keeps the danger of inconsistent changes across many records low.

Comparison Example:

Good Design: Using a Weak Entity:

```
course(course_id, title, credits)
section(course_id, section_id, semester, year)
```

Changing to Strong Entity (Redundant Design):

```
section(course_id, title, credits, section_id, semester, year)
```

Conclusion:

Weak entity sets are not a restriction; they are a purposeful and effective method to depict **logical dependencies**, **prevent redundancy**, and **maintain normalization**. Making them powerful entities typically leads to schemas that are too big and hard to keep up with.

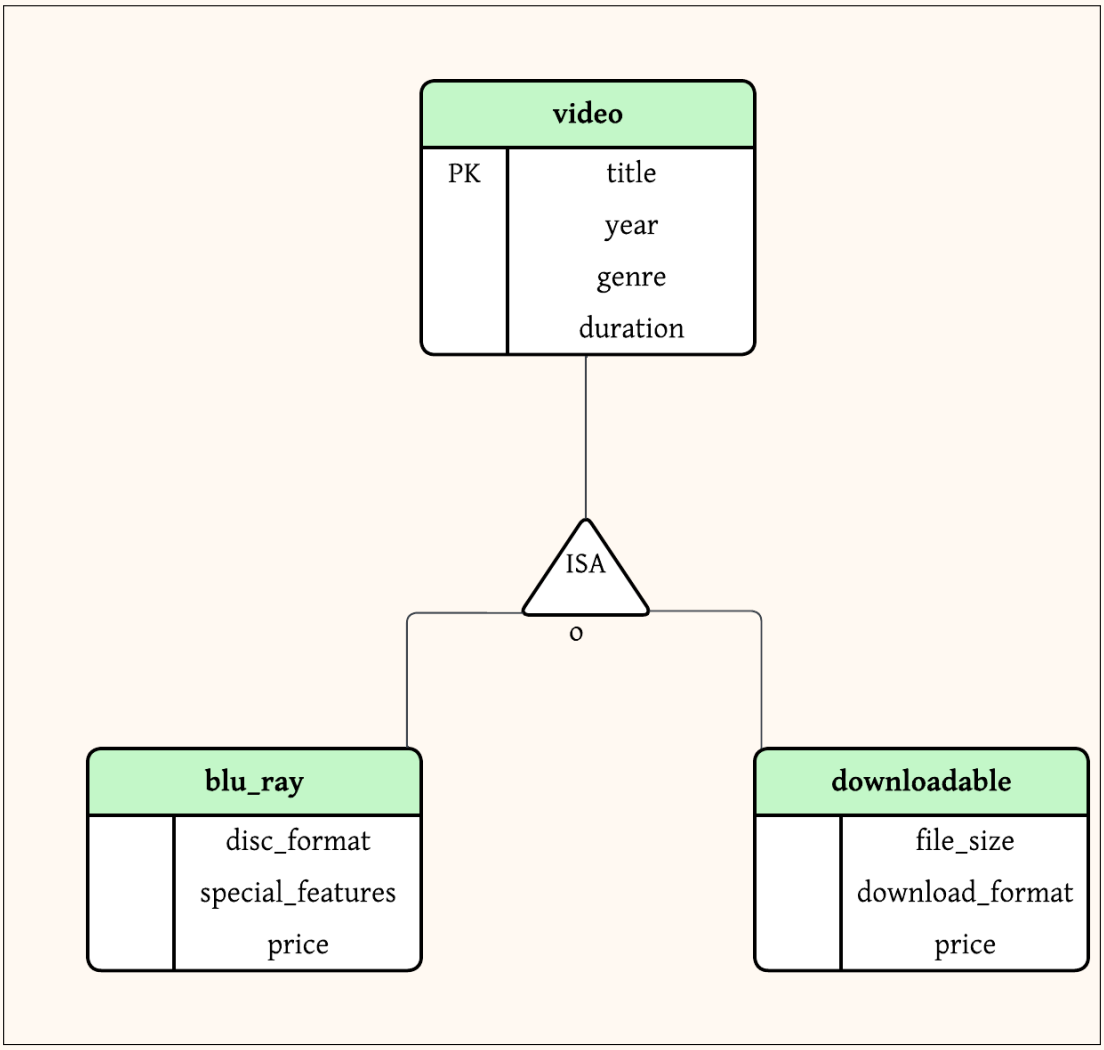
Question 6.21

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.

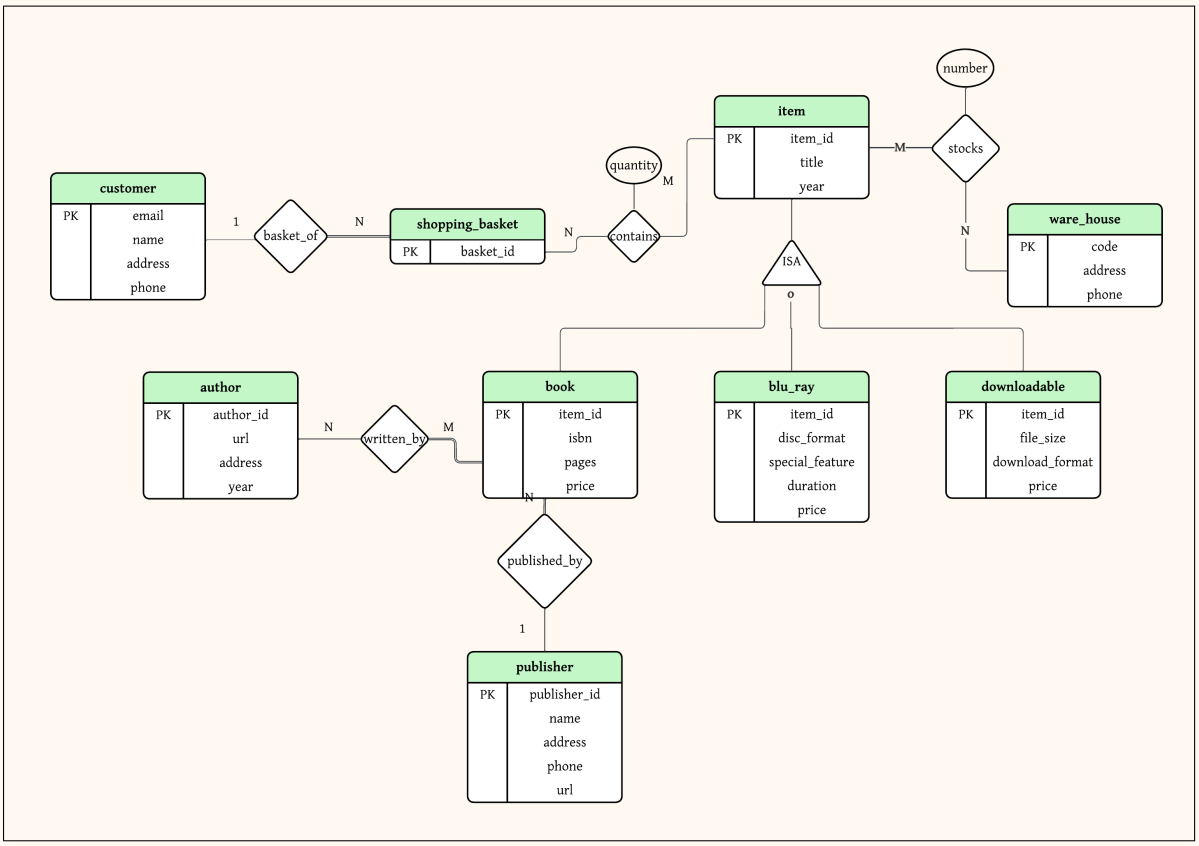
b. Now extend the full E-R diagram to model the case where a shopping basket may contain any combination of books, Blu-ray discs, or downloadable video.

Answer 6.21 (a)



Note: This diagram shows the E-R model for adding Blu-ray discs and downloadable video to the bookstore collection. It demonstrates how the same video item can exist in multiple formats with different pricing structures, focusing specifically on the video-related entities and relationships.

Answer 6.21 (b)

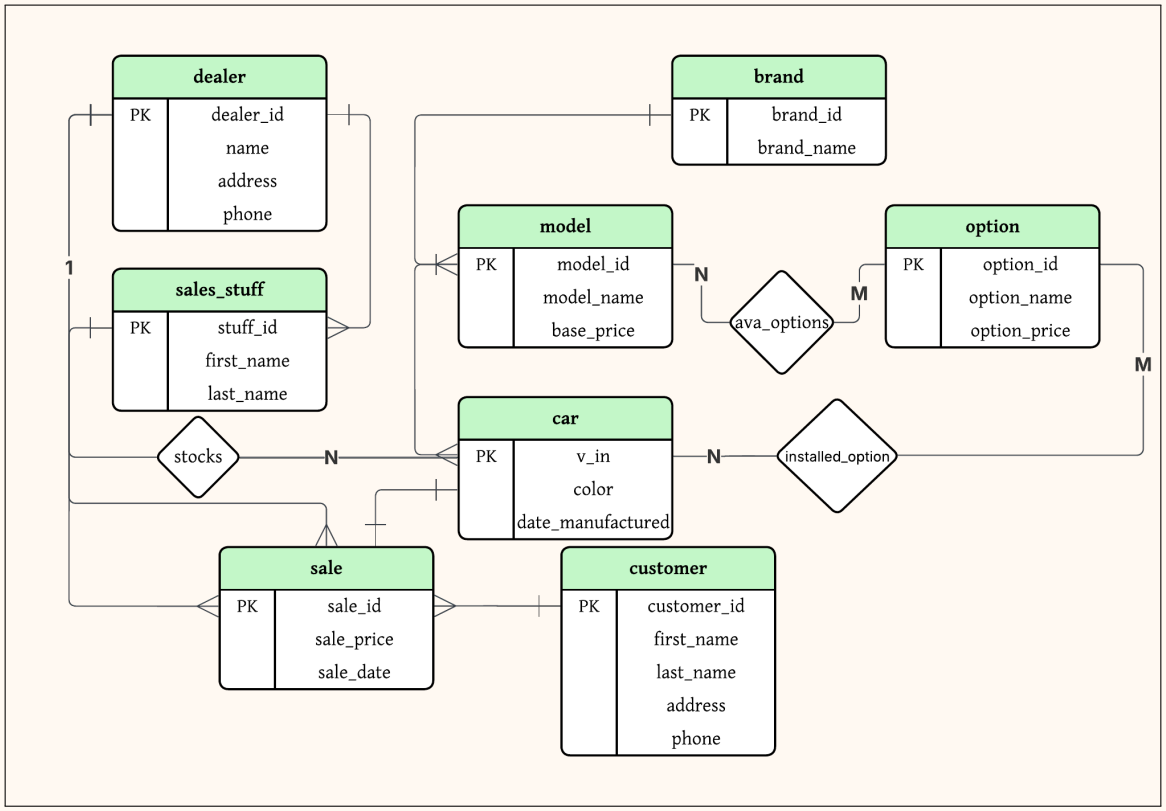


Note: This diagram shows the complete extended E-R model for the online bookstore, incorporating the shopping basket functionality that can contain any combination of books, Blu-ray discs, or downloadable video items. It demonstrates the comprehensive relationship structure for the multi-format inventory system.

Question 6.22

Design a database 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.
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 6.22

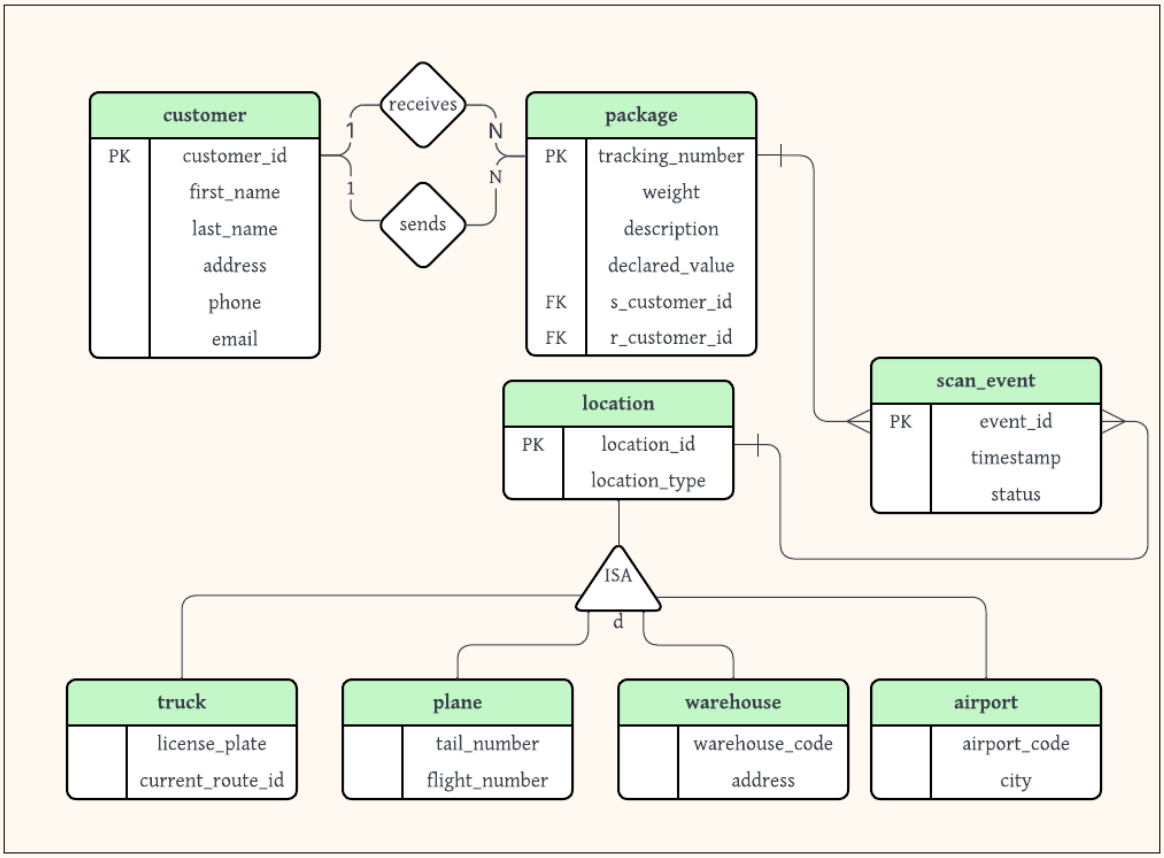


Note: This answer contains a comprehensive database design for an automobile company including an E-R diagram, relational schemas, and constraints. The design covers vehicle identification (VIN), brand-model relationships, optional features, dealer management, customer records, and sales operations with proper primary-key and foreign-key constraints as specified in the requirements.

Question 6.23

Design a database for a worldwide package delivery company (e.g., DHL or FedEx). The database must be able to keep track of customers who ship items and customers who receive items; some customers may do both. Each package must be identifiable and trackable, so the database must be able to store the location of the package and its history of locations. Locations include trucks, planes, airports, and warehouses.
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 6.23

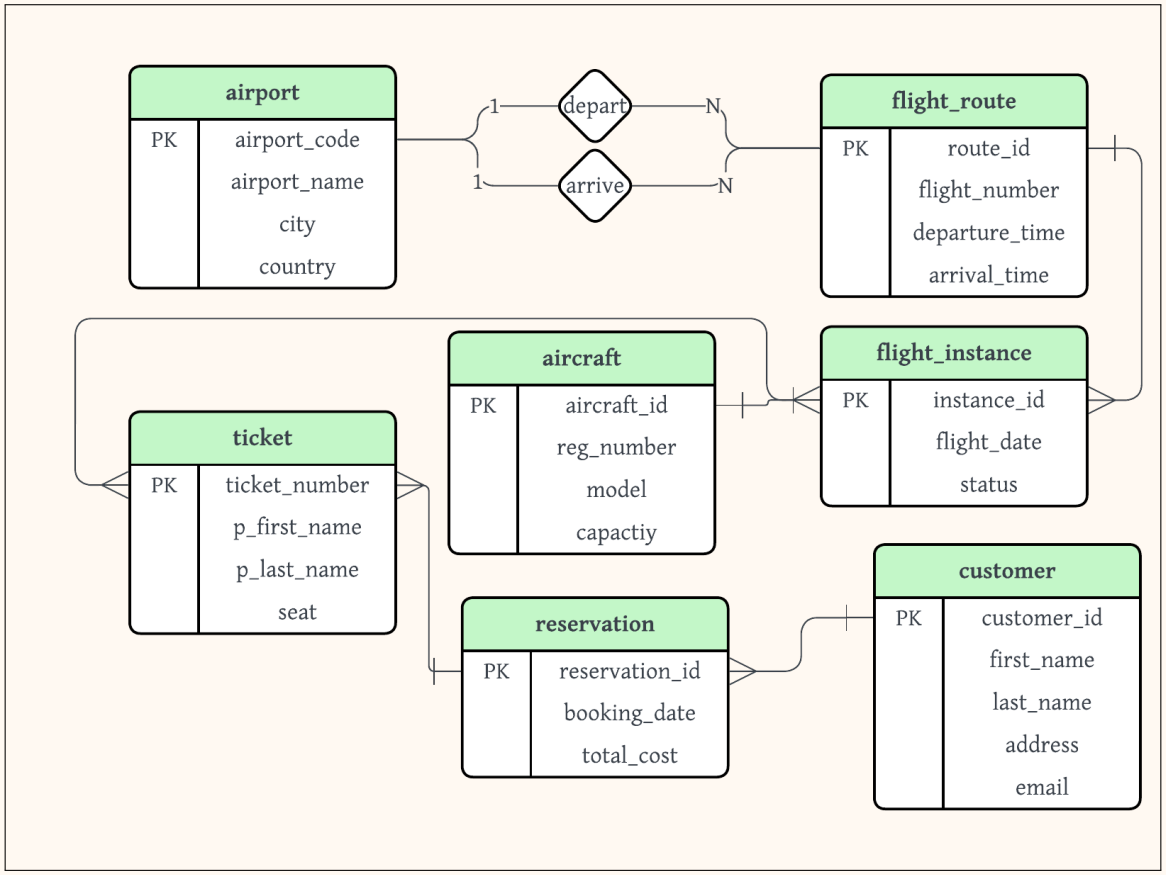


Note: This answer contains a comprehensive database design for a worldwide package delivery company including an E-R diagram, relational schemas, and constraints. The design covers customer management (shippers and receivers), package identification and tracking, location history (trucks, planes, airports, warehouses), and delivery operations with proper primary-key and foreign-key constraints as specified in the requirements.

Question 6.24

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 6.24

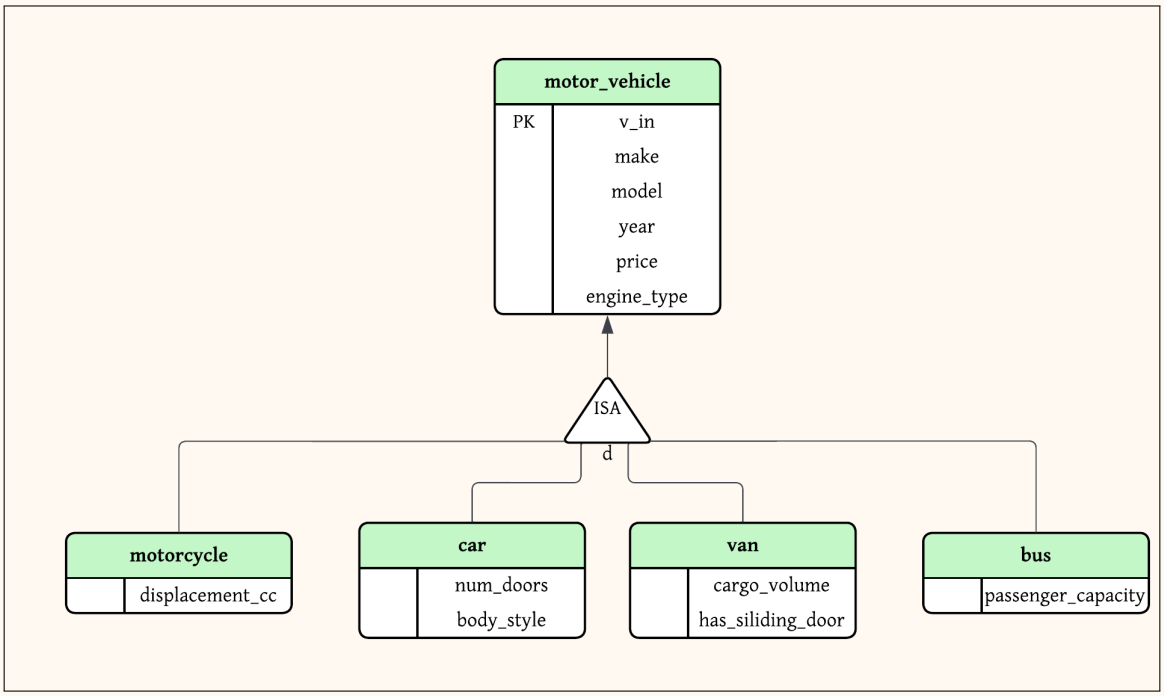


Note: This answer contains a comprehensive database design for an airline system including an E-R diagram, relational schemas, and constraints. The design covers customer management, reservation tracking, flight status monitoring, seat assignment management, and flight scheduling/routing operations with proper primary-key and foreign-key constraints as specified in the requirements.

Question 6.26

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 6.26



Note: This answer contains a generalization-specialization hierarchy design for a motor vehicle sales company. The diagram shows the inheritance structure from the general Vehicle entity down to specific vehicle types (motorcycles, passenger cars, vans, and buses), with proper attribute placement at each hierarchy level. It includes justifications for why attributes are placed at specific levels and explanations of the inheritance relationships between the different vehicle categories.