- **Entity integrity** is a constraint that states that in a base relation no attribute of a primary key can be null. **Referential integrity** states that foreign key values must match a candidate key value of some tuple in the home relation or be wholly null. Apart from relational integrity, integrity constraints include required data, domain, and multiplicity constraints; other integrity constraints are called **general constraints**.
- A **view** in the relational model is a **virtual** or **derived relation** that is dynamically created from the underlying base relation(s) when required. Views provide security and allow the designer to customize a user's model. Not all views are updatable.

Review Questions

4.1 Discuss each of the following concepts in the context of the relational data model:
(a) relation
(b) attribute
(c) domain
(d) tuple
(e) intension and extension
(f) degree and cardinality.

4.2 Describe the relationship between mathematical relations and relations in the relational data model.

4.3 Describe the differences between a relation and a relation schema. What is a relational database schema?

4.4 Discuss the properties of a relation.

4.5 Discuss the differences between the candidate keys and the primary key of a relation. Explain what is meant by a foreign key. How do foreign keys of relations relate to candidate keys? Give examples to illustrate your answer.

4.6 Define the two principal integrity rules for the relational model. Discuss why it is desirable to enforce these rules.

4.7 What is a view? Discuss the difference between a view and a base relation.

Exercises

The following tables form part of a database held in a relational DBMS:

Hotel (hotelNo, hotelName, city)
Room (roomNo, hotelNo, type, price)
Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)
Guest (guestNo, guestName, guestAddress)

where Hotel contains hotel details and hotelNo is the primary key;
Room contains room details for each hotel and (roomNo, hotelNo) forms the primary key;
Booking contains details of bookings and (hotelNo, guestNo, dateFrom) forms the primary key;
Guest contains guest details and guestNo is the primary key.

4.8 Identify the foreign keys in this schema. Explain how the entity and referential integrity rules apply to these relations.

4.9 Produce some sample tables for these relations that observe the relational integrity rules. Suggest some general constraints that would be appropriate for this schema.

4.10 Analyze the RDBMSs that you are currently using. Determine the support the system provides for primary keys, alternate keys, foreign keys, relational integrity, and views.

4.11 Implement the previous schema in one of the RDBMSs you currently use. Implement, where possible, the primary, alternate, and foreign keys, and appropriate relational integrity constraints.

Exercises

For the following exercises, use the Hotel schema defined at the start of the Exercises at the end of Chapter 4.

5.8 Describe the relations that would be produced by the following relational algebra operations:

(a) $\Pi_{hotelNo}(\sigma_{price > 50}(Room))$

(b) $\sigma_{Hotel.hotelNo = Room.hotelNo}(Hotel \times Room)$

(c) $\Pi_{hotelName}(Hotel \bowtie_{Hotel.hotelNo = Room.hotelNo}(\sigma_{price > 50}(Room)))$

(d) $Guest \div (\sigma_{dateTo \geq '1-Jan-2007'}(Booking))$

(e) $Hotel \bowtie_{Hotel.hotelNo = Room.hotelNo}(\sigma_{price > 50}(Room))$

(f) $\Pi_{guestName, hotelNo}(Booking \bowtie_{Booking.guestNo = Guest.guestNo} Guest) \div \Pi_{hotelNo}(\sigma_{city = 'London'}(Hotel))$

5.9 Provide the equivalent tuple relational calculus and domain relational calculus expressions for each of the relational algebra queries given in Exercise 5.8.

5.10 Describe the relations that would be produced by the following tuple relational calculus expressions:

(a) {H.hotelName | Hotel(H) ∧ H.city = 'London'}

(b) {H.hotelName | Hotel(H) ∧ (∃R) (Room(R) ∧ H.hotelNo = R.hotelNo ∧ R.price > 50)}

(c) {H.hotelName | Hotel(H) ∧ (∃B) (∃G) (Booking(B) ∧ Guest(G) ∧ H.hotelNo = B.hotelNo ∧ B.guestNo = G.guestNo ∧ G.guestName = 'John Smith')}

(d) {H.hotelName, G.guestName, B1.dateFrom, B2.dateFrom | Hotel(H) ∧ Guest(G) ∧ Booking(B1) ∧ Booking(B2) ∧ H.hotelNo = B1.hotelNo ∧ G.guestNo = B1.guestNo ∧ B2.hotelNo = B1.hotelNo ∧ B2.guestNo = B1.guestNo ∧ B2.dateFrom ≠ B1.dateFrom}

5.11 Provide the equivalent domain relational calculus and relational algebra expressions for each of the tuple relational calculus expressions given in Exercise 5.10.

5.12 Generate the relational algebra, tuple relational calculus, and domain relational calculus expressions for the following queries:

(a) List all hotels.

(b) List all single rooms with a price below £20 per night.

(c) List the names and cities of all guests.

(d) List the price and type of all rooms at the Grosvenor Hotel.

(e) List all guests currently staying at the Grosvenor Hotel.

(f) List the details of all rooms at the Grosvenor Hotel, including the name of the guest staying in the room, if the room is occupied.

(g) List the guest details (guestNo, guestName, and guestAddress) of all guests staying at the Grosvenor Hotel.

5.13 Using relational algebra, create a view of all rooms in the Grosvenor Hotel, excluding price details. What are the advantages of this view?

The following tables form part of a database held in an RDBMS:

Employee	(<u>empNo</u>, fName, lName, address, DOB, sex, position, deptNo)
Department	(<u>deptNo</u>, deptName, mgrEmpNo)
Project	(<u>projNo</u>, projName, deptNo)
WorksOn	(<u>empNo, projNo, dateWorked</u>, hoursWorked)

where **Employee** contains employee details and empNo is the key.

Department contains department details and deptNo is the key. mgrEmpNo identifies the employee who is the manager of the department. There is only one manager for each department.

Project contains details of the projects in each department and the key is projNo (no two departments can run the same project).

and **WorksOn** contains details of the hours worked by employees on each project, and empNo/projNo/dateWorked form the key.