|  |
| --- |
| 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.  3.8  Identify the foreign keys in this schema. Explain how the entity and referential integrity rules apply to these relations.  3.9  Produce some sample tables for these relations that observe the relational integrity rules. Suggest some general constraints that would be appropriate for this schema.  3.10  Analyze the RDBMSs that you are currently using. Determine the support the system provides for primary keys, identity integrity, foreign keys, relational integrity.  3.11  Implement the above schema in one of the RDBMSs you currently use. Implement, where possible, the primary, alternate and foreign keys, and appropriate relational integrity constraints.  Write your answers from 3.8 - 3.10 in a text/word file and push it to Github. When you are finished with exercise 3.11 make a sql dump of the database and push it to github. We will look at it in the next class.  *ISBN 0 321 21025 5 (p87)* |
|  |

3.8 Foreign keys are : In tables Room and Booking: hotelNo and in Guest: guestNo.

**Entity Integrity rules** says that no attribute of a primary key can be null so we can‘t have empty or data which can’t identify uniquely a tuple (row).

In this hotelNo example every hotel must be identified by unique attribute in the table (relation) because we want to identify that specific hotel. If we had the same hotelNo entry in different tuples we’ll get an error when trying to modify the data because the system will not know to which tuple refers to.

**Referential integrity rules** says that if a foreign key exists in a relation (table), either the foreign key must match a candidate key value of some tuple (row) in his home relation or the home key value must be wholly null.

In our example by using the foreign key hotelNo we can identify of which hotel a room and a booking belongs to and by using the foreign key guestNo what booking belongs to a specific guest. So we can get all the information containing what hotel he is staying at and what room have a guest booked.

3.9 General Constrains are additional rules specified by the users or database administrators of a database that define or constrain some aspect of the enterprise.

General constrains for this schema:

General constrains on range:

– for example if the biggest hotel has 50 rooms, then we shouldn’t be able to enter more than 50 tuples in roomNo and the DBMS should enforce that and display that the range 1 … 50 can’t be exceeded.

- for example the price should be also enforced by the DBMS if doesn’t match a range of values (f. ex. if it inserted by mistake an extra digit) in the attribute `price` in Room table.

General constrains on date:

– for example if a client is booking for a specific period, that period must be enforced by the DBMS that is placed in the future to eliminate typing error or some other kind of error for attributes `dateFrom` and `dateTo` in Booking table.

3.10 *Analyze the RDBMSs that you are currently using. Determine the support the system provides for primary keys, identity integrity, foreign keys, relational integrity.*

Support for primary keys: auto increment, not null option, unique index

Support for identity integrity: auto increment, not null

Support for foreign keys: no action on delete

Support for Relational Integrity: foreign keys can’t be deleted.