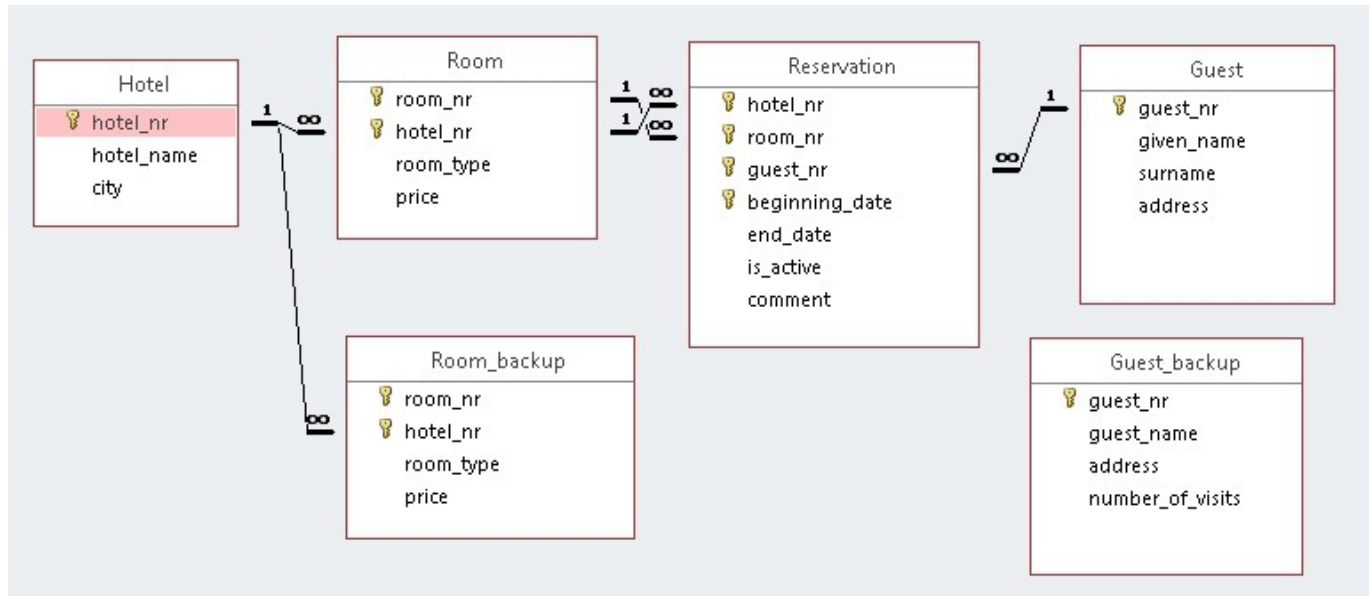


SQL Exercise 2

Database that is created by using MS Access: *Hotel* (**Hotel.mdb**). The structure of the base tables (tables in short) is the following.



All the presented tasks must be solved by using SQL statements. One can use multiple statements for solving a task if not stated otherwise. Save the statements in the MS Access database – for each statement you have to create a separate query object. Use the naming like *Task00*, *Task01*, *Task02*, etc. If the solution involves more than one SQL statement, then use the names like *Task01_1*, *Task01_2*, etc.

10. Find all the data about reservations. For each reservation, find also the concatenated given name and surname of the associated guest (in the column *guest_name*) as well as the name of the hotel where the reservation is made. Sort the result based on the hotel name in descending alphabetical order. If there are more than one reservation in the same hotel, then sort these rows based on the guest number in the ascending order. Use in the query correlation names (aliases) G, R, and H to refer to the tables *Guest*, *Reservation*, and *Hotel*, respectively.
11. Find the number of reservations in case of each hotel. For each hotel, present the hotel number, hotel name, and the number of reservations. If there are no reservations in a hotel, then the number of reservations must be 0. The column containing data values that represent the number of reservations must have the name *cnt*. Sort the result based on the number of reservations in the descending order.
12. Find all the data of hotels (from all the columns of the table *Hotel*) that have no associated reservations.

13. Register a new hotel in the table *Hotel*. The name of the hotel is "Viljandi Boatman" and the city is "Viljandi".
14. Insert to the table *Guest_backup* all the data about the guests (from the table *Guest*) whose address contains the word "Tallinn" or whose address is not registered. Please note that table *Guest_backup* must contain concatenated given name and surname of guests. In case of concatenating strings, remove from the result leading and trailing spaces by using *Trim* function.
15. Create a copy of the table *Reservation* by using a *SELECT ... INTO* statement. The name of the copy table must be *Reservation_backup*.
16. Define the primary key constraint in the table *Reservation_backup* that involves columns *hotel_nr*, *room_nr*, *guest_nr*, and *beginning_date* (in the specified order). The name of the primary key constraint must be *pk_reservation_backup*.
17. Delete all the rows from the table *Guest_backup*.
18. Delete from the table *Reservation_backup* data about all the reservations that have started more than 1600 days ago in the hotel "Viru".
19. Delete from the table *Reservation_backup* data about all the end dates of reservations where such dates are registered. In case of all the modified rows register also the comment: "End date is removed at X", where X is the current timestamp. The comment must be appended to the existing comment. Current timestamp must be found by using a function.