

4.8

Foreign keys:

In table Room: hotelNo;

In table Booking: hotelNo, guestNo, roomNo;

Entity rules in these relations can be seen in the fact that in **none** of the primary keys is a null, while the referential integrity means that every foreign key can be found in the table it refers to or it is set to null.

4.9

Hotel

hotelNo	hotelName	city
666	Dalniela's heaven	Pasikurówice
44	Adamstron	Ghanzi

Room

roomNo	hotelNo	type	price
24	666	Single	40dkk
12	666	Double	80dkk
7	44	Family	150dkk
98	666	Single	39dkk
33	44	Double	100dkk
154	44	Single	12dkk

Booking

hotelNo	guestNo	dateFrom	dateTo	roomNo
44	88	29/12/2018	2/01/2019	7
666	68	12/12/2018	12/01/2019	24
44	96	15/03/2018	21/03/2018	154
666	72	6/03/2018	8/03/2018	24
44	68	9/07/2018	17/07/2018	33

Guest

guestNo	guestName	guestAddress
96	Peter Peterson	Honolulu Street
65	Mark Gsheeshulka	Mariastan Street
21	Boscena Tumidajewich	Floryda Street
72	Lorena Spienkier	Jolypop Street
45	Orfeus Zalad	Cyklop Street
88	Penelopa Candycane	Sia Street
68	Odetta Black	Swanlake Street

The dates of the bookings of the same room can not be overlapping; The same guest can not have two or more overlapping bookings; there can only be 5 hotels in the table;

7.10

```
CREATE TABLE Hotel (  
    hotelNo INT NOT NULL  
    ,hotelName VARCHAR(15)  
    ,city VARCHAR(15)  
    ,PRIMARY KEY (hotelNo)  
);
```

7.11

```
CREATE TABLE Room (  
    roomNo INT CHECK (VALUE BETWEEN 1 AND 100)  
    ,hotelNo INT CHECK (VALUE IN (SELECT hotelNo FROM Hotel))  
    ,type CHAR(6) CHECK(VALUE IN('Single', 'Double', 'Family'))  
    ,price INT CHECK (VALUE BETWEEN 10 AND 100)  
    ,PRIMARY KEY (roomNo, hotelNo)  
);
```

```
CREATE TABLE Booking (  
    hotelNo INT CHECK (VALUE IN (SELECT hotelNo FROM Hotel))  
    ,guestNo INT CHECK (VALUE IN (SELECT guestNo FROM Guest))  
    ,dateFrom DATE CHECK (VALUE > GETDATE())  
    ,dateTo DATE CHECK (VALUE > GETDATE())  
    ,roomNo INT CHECK (VALUE IN (SELECT roomNo FROM Room)) UNIQUE  
    ,PRIMARY KEY (hotelNo, guestNo, dateFrom)  
);
```

```
CREATE TABLE Guest (  
    guestNo INT  
    ,guestName VARCHAR(15)
```

```
,guestAddress VARCHAR(20)
,PRIMARY KEY (guestNo)
);
```

7.12

```
CREATE TABLE BookingArchive (
    hotelNo INT CHECK (VALUE IN (SELECT hotelNo FROM Hotel))
    ,guestNo INT CHECK (VALUE IN (SELECT guestNo FROM Guest))
    ,dateFrom DATE CHECK (VALUE > GETDATE())
    ,dateTo DATE CHECK (VALUE > GETDATE())
    ,roomNo INT CHECK (VALUE IN (SELECT roomNo FROM Room)) UNIQUE
    ,PRIMARY KEY (hotelNo, guestNo, dateFrom)
);
```

```
INSERT INTO BookingArchive
SELECT *
FROM Booking
WHERE dateTo > 1/01/2007;
```

6.7

```
SELECT *
FROM Hotel;
```

6.8

```
SELECT *
FROM Hotel
WHERE city = 'London';
```

6.9

```
SELECT guestName, guestAddress
FROM Guest
WHERE guesAddress = '%London%'
ORDER BY guestName;
```




6.10

```
SELECT *
```

FROM Room

WHERE type = 'family' OR type = 'double' AND price < 40.00

ORDER BY price; 

6.11

SELECT *

FROM Booking

WHERE dateTo IS NULL;

6.12

SELECT COUNT(hotelNo) AS noOfHotels

FROM Hotel;

6.13

SELECT AVG(price)

FROM Room;

6.14

SELECT SUM(price) AS revenue

FROM Room

WHERE type = 'Double'

GROUP BY roomNo;

6.15

SELECT COUNT(guestNo) AS myCount

FROM Booking

WHERE dateFrom > 31/07/2018 AND dateFrom < 1/09/2018;

6.16

SELECT price, type

FROM Room

WHERE hotelNo =

(SELECT hotelNo

FROM Hotel

WHERE hotelName = 'Grosvenor Hotel');

6.17



```
SELECT *  
FROM Guest  
WHERE guestNo IN  
    (SELECT guestNo  
     FROM Booking  
     WHERE hotelNo =  
         (SELECT hotelNo  
          FROM Hotel  
          WHERE hotelName = 'Grosvenor Hotel'));
```

6.18



```
SELECT guestName, r.*  
FROM Room r, guest g  
WHERE guestNo IN  
    (SELECT guestNo  
     FROM Booking b  
     WHERE b.guestNo IS NOT NULL AND roomNo IN  
         (SELECT roomNo  
          FROM Room  
          WHERE hotelNo =  
              (SELECT hotelNo  
               FROM Hotel  
               WHERE hotelName = 'Grosvenor Hotel')));
```

6.19



```
SELECT SUM(price) AS income
FROM Room
WHERE roomNo IN
    (SELECT roomNo
     FROM Booking
     WHERE dateFrom < GETDATE() AND dateTo > GETDATE() AND hotelNo =
        (SELECT hotelNo
         FROM Hotel
         WHERE hotelName = 'Grosvenor Hotel')));
```

6.20

```
SELECT *
FROM Room
WHERE roomNo IN
    (SELECT roomNo
     FROM Booking
     WHERE (dateFrom < GETDATE() AND dateTo < GETDATE()) OR (dateFrom > GETDATE() AND
        dateTo > GETDATE()) AND hotelNo =
        (SELECT hotelNo
         FROM Hotel
         WHERE hotelName = 'Grosvenor Hotel')));
```

6.21



```
SELECT SUM(price)
FROM Room
WHERE roomNo IN
    (SELECT roomNo
     FROM Booking
     WHERE (dateFrom < GETDATE() AND dateTo < GETDATE()) OR (dateFrom > GETDATE() AND
        dateTo > GETDATE()) AND hotelNo =
```

```
(SELECT hotelNo  
FROM Hotel  
WHERE hotelName = 'Grosvenor Hotel'));
```

6.22

```
SELECT hotelName, COUNT(roomNo) AS numberOfRooms  
FROM Hotel h, Room r  
GROUP BY hotelName;
```

6.23

```
SELECT hotelName, COUNT(roomNo) AS numberOfRooms  
FROM Hotel h, Room r  
WHERE city = 'London'  
GROUP BY hotelName;
```

6.24 

```
SELECT AVG(COUNT(hotelNo, guestNo, dateFrom)), hotelName  
FROM Booking b, Hotel h  
WHERE dateFrom > 31/07/2018 AND dateFrom < 1/09/2018  
GROUP BY hotelName;
```

6.25

```
SELECT type, MAX(COUNT(hotelNo, guestNo, dateFrom)), hotelName  
FROM Room r, Booking b, Hotel h  
WHERE city = 'London'  
GROUP BY hotelName;
```

6.26

```
SELECT SUM(price), hotelName  
FROM Room r, Hotel h  
WHERE roomNo IN  
    (SELECT roomNo  
    FROM Booking  
    WHERE (dateFrom < GETDATE() AND dateTo < GETDATE()) OR (dateFrom > GETDATE() AND  
    dateTo > GETDATE()))  
GROUP BY hotelName;
```

6.27

```
INSERT INTO Hotel VALUES ('666', 'Daniela's Heaven', 'Pasikurrowice');
INSERT INTO Hotel VALUES ('44', 'Adamstron', 'Ghanzi');
INSERT INTO Room VALUES ('24', '666', 'Single', '40dkk');
INSERT INTO Room VALUES ('24', '666', 'Single', '40dkk');
INSERT INTO Room VALUES ('12', '44', 'Family', '150dkk');
INSERT INTO Booking VALUES ('44', '88', '29/12/2018', '2/01/2019', '7');
INSERT INTO Booking VALUES ('666', '68', '12/12/2018', '12/01/2019', '24');
INSERT INTO Guest VALUES ('96', 'Peter Peterson', 'Honolulu Street');
INSERT INTO Guest VALUES ('65', 'Mark Gsheeshulka', 'Marianstan Street');
```

6.28

UPDATE Room

SET price = prise*1.05;

5.12

- a) Π hotelNo, hotelName, city (Hotel)
- b) Π roomNo, hotelNo, type, price (σ type=single \wedge price<20(Room))
- c) Π guestName, guestAddress (Guest)
- d) $(\Pi$ price, type(Room)) \bowtie (Π hotelName (σ hotelName = Grosvenor Hotel(Hotel)))
- e) $((\Pi$ (σ dateFrom <= NOW() AND dateTo > NOW() (Booking)) \bowtie Booking.hotelNo = Hotel.hotelNo (Π (σ hotelName = Grosvenor Hotel (Hotel))
- \bowtie Hotel.guestNo = Guest.guestNo (Π guestName (Guest))
- f) $((\Pi$ roomNo, hotelNo, price, type (Room)) \bowtie Room.hotelNo = Hotel.hotelNo (Π hotelName (σ hotelName = Grosvenor Hotel(Hotel))))
- \bowtie Room.hotelNo = Booking.hotelNo (Π (σ guestNo != null (Booking)) \bowtie Booking.guestNo = Guest.guestNo (Π guestName (Guest))
- e) $(\Pi$ guestNo, guestName, guestAddress (Guest)) \bowtie Guest.guestNo = Booking.guestNo (Π (Booking))
- \bowtie Booking.hotelNo = Hotel.hotelNo (Π (σ hotelName = Grosvenor Hotel (Hotel))