Task 1
SQL Athena Query

As all details are not explained in the question itself so my solution is based on certain assumptions :

- After updating the checkout_date for any room, a new entry will be created with ***open*** status.
- Status column will be updated as **Closed** once we update the checkin_date whether the booking is for a future date.
- Status column will be in **Sequential** order e.g. first status will be open then close then again open.

Query to create a sample table

```
CREATE EXTERNAL TABLE IF NOT EXISTS `hotelbookings`.`reservations` (
  `room_number` int,
  `reservation_status` string,
  `checkin_date` date,
  `checkout_date` date
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
with serdeproperties(
    'serialization.format'=',',
    'field.delim'=','
)
LOCATION 's3://hotel-data/csv/'
TBLPROPERTIES ("skip.header.line.count"="1");
```

Sample structure of the table

room_number	reservation_status	checkin_date	checkout_date
1	open		
1	closed	2023-02-09	2023-02-21
1	open		
1	closed	2023-02-01	2023-02-03

1	open		
2	open		
2	closed	2023-02-01	2023-02-14
2	open		
3	closed	2023-02-01	
3	open		
4	open		
4	closed	2023-02-01	2023-02-05
4	open		
5	closed	2023-02-01	
5	open		
6	open		

Query for derived table from reservations table

```
CREATE TABLE derived_table
WITH (
  partitioned_by = ARRAY['checkin_date'],
  format = 'parquet',
  external_location = 's3://hotel-partitioned-data/derived-table/'
)
AS
select room_number,reservation_status,checkout_date,checkin_date
from (select *, row_number () over (partition by room_number) as rn
from reservations) as p where p.rn=1  and p.reservation_status='open';
```

Explanation

- Subquery will create result like this, row_number () window function gives a rank to each reservation status.

room_number	reservation_status	checkin_date	checkout_date	row_number
1	open			1
1	closed	2023-02-09	2023-02-21	2
1	open			3

1	closed	2023-02-01	2023-02-03	4
1	open			5
2	open			1
2	closed	2023-02-01	2023-02-14	2
2	open			3
3	closed	2023-02-01		1
3	open			2
4	open			1
4	closed	2023-02-01	2023-02-05	2
4	open			3
5	closed	2023-02-01		1
5	open			2
6	open			1

- Now we only need to show those rooms which are not reserved right now.

room_number	reservation_status	checkin_date	checkout_date	row_number
1	open			1
1	closed	2023-02-09	2023-02-21	2
1	open			3
1	closed	2023-02-01	2023-02-03	4
1	open			5
2	open			1
2	closed	2023-02-01	2023-02-14	2
2	open			3
3	closed	2023-02-01		1
3	open			2
4	open			1
4	closed	2023-02-01	2023-02-05	2
4	open			3
5	closed	2023-02-01		1
5	open			2
6	open			1

Optimization Method for Athena Query

As we know user will filter mainly on `*reservation_status*` column so If we wanted to reduce our data scan and improve query speed as well, we can do the following things :

1. We can bucket our Data on the basis of the `*reservation_status*` and `*room_number*` column.
2. We can use columnar file format to store the data like (Parquet and ORC).