SQL Assignment

Q1. What is the time period used?

```
8   -- Q1. What is the time period used?
9
10  ALTER TABLE airbnb_calendar MODIFY date DATE;
11  SELECT MIN(date) as min_date, MAX(date) as max_date,
12      DATEDIFF(MAX(date), MIN(date)) AS timeperiodused FROM airbnb_calendar;
13
```

100% 1:7

Result Grid | Filter Rows: Search | Export:

min_date	max_date	timeperiodused
2016-09-06	2017-09-05	364

Q2. How many properties have duplicate entries? Remove duplicate rows (say a row appears 3 times, remove 2 and keep 1)

```
14  -- Q2. How many properties have duplicate entries? Remove duplicate rows
15
16  SELECT listing_id,date,COUNT(*) c FROM airbnb_calendar GROUP BY listing_id, date
17      HAVING count(*)> 1;
18
19  DELETE t1 FROM (SELECT *, ROW_NUMBER() OVER(listing_id)  as rownum  FROM airbnb_calendar) t1
20  INNER JOIN (SELECT *, ROW_NUMBER() OVER(listing_id)  as rownum FROM airbnb_calendar) t2
21  WHERE
22      t1.date = t2.date AND
23      t1.listing_id = t2.listing_id AND
24      t1.rownum < t2.rownum;
25
```

100% 1:13

Result Grid | Filter Rows: Search | Export:

listing_id	date	c
12898806	2016-09-06	2
12898806	2016-09-07	2
12898806	2016-09-08	2
12898806	2016-09-09	2
12898806	2016-09-10	2
12898806	2016-09-11	2
12898806	2016-09-12	2
12898806	2016-09-13	2

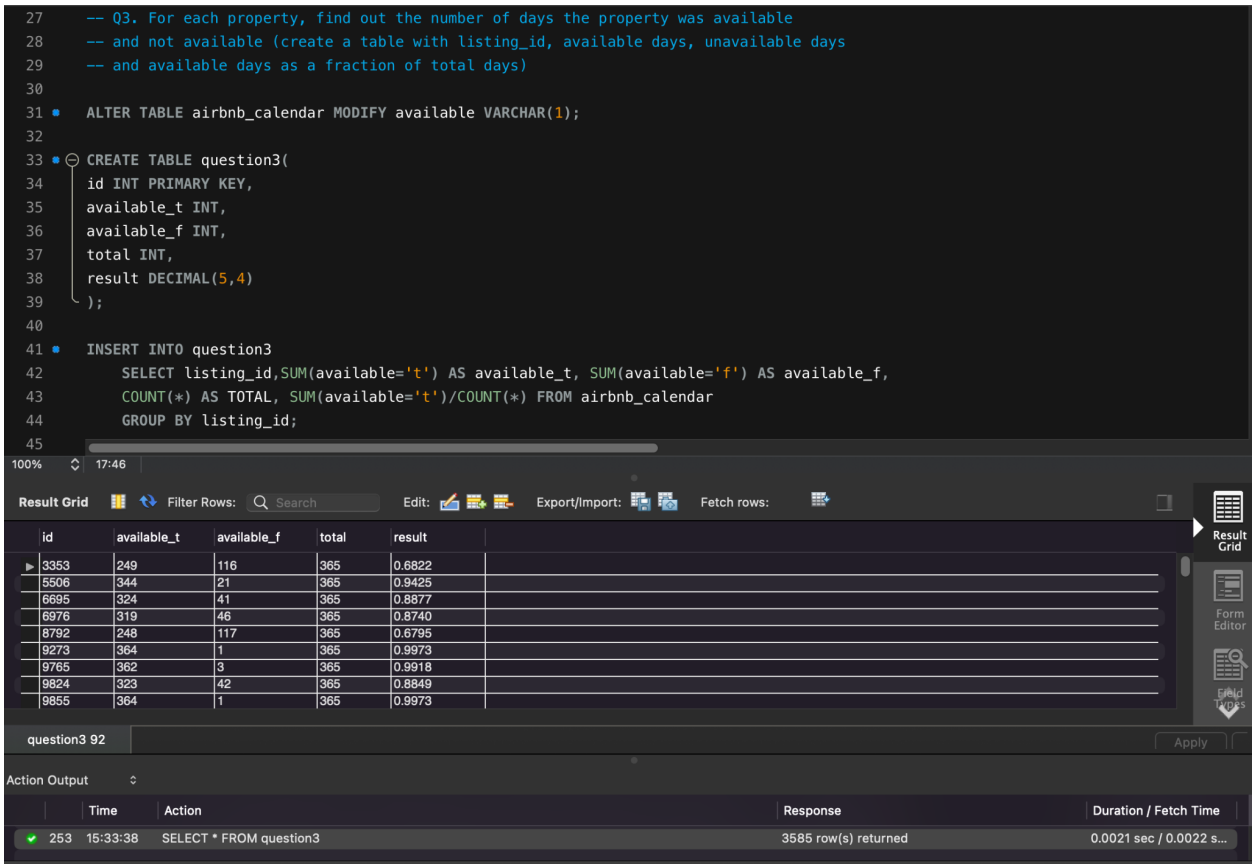
Result Grid | Form Editor

Result 91 | Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
252	15:31:46	SELECT listing_id,date,COUNT(*) c FROM airb...	365 row(s) returned	2.280 sec / 0.00015...

Q3. For each property, find out the number of days the property was available and not available (create a table with listing_id, available days, unavailable days and available days as a fraction of total days)



```
-- Q3. For each property, find out the number of days the property was available
-- and not available (create a table with listing_id, available days, unavailable days
-- and available days as a fraction of total days)

ALTER TABLE airbnb_calendar MODIFY available VARCHAR(1);

CREATE TABLE question3(
id INT PRIMARY KEY,
available_t INT,
available_f INT,
total INT,
result DECIMAL(5,4)
);

INSERT INTO question3
    SELECT listing_id,SUM(available='t') AS available_t, SUM(available='f') AS available_f,
    COUNT(*) AS TOTAL, SUM(available='t')/COUNT(*) FROM airbnb_calendar
    GROUP BY listing_id;
```

Result Grid

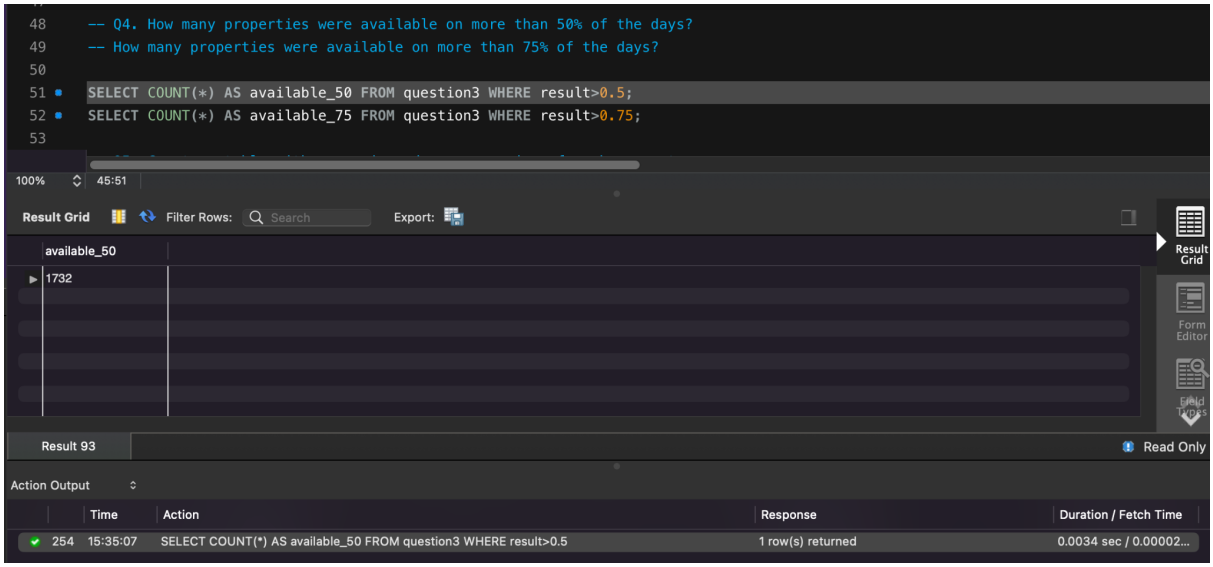
id	available_t	available_f	total	result
3353	249	116	365	0.6822
5506	344	21	365	0.9425
6695	324	41	365	0.8877
6976	319	46	365	0.8740
8792	248	117	365	0.6795
9273	364	1	365	0.9973
9765	362	3	365	0.9918
9824	323	42	365	0.8849
9855	364	1	365	0.9973

question3 92

Action Output

	Time	Action	Response	Duration / Fetch Time
253	15:33:38	SELECT * FROM question3	3585 row(s) returned	0.0021 sec / 0.0022 s...

Q4. How many properties were available on more than 50% of the days? How many properties were available on more than 75% of the days?



```
-- Q4. How many properties were available on more than 50% of the days?
-- How many properties were available on more than 75% of the days?

SELECT COUNT(*) AS available_50 FROM question3 WHERE result>0.5;
SELECT COUNT(*) AS available_75 FROM question3 WHERE result>0.75;
```

Result Grid

available_50
1732

Result 93

Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
254	15:35:07	SELECT COUNT(*) AS available_50 FROM question3 WHERE result>0.5	1 row(s) returned	0.0034 sec / 0.00002...

```
48    -- Q4. How many properties were available on more than 50% of the days?
49    -- How many properties were available on more than 75% of the days?
50
51    SELECT COUNT(*) AS available_50 FROM question3 WHERE result>0.5;
52    SELECT COUNT(*) AS available_75 FROM question3 WHERE result>0.75;
53
54
```

100% 28:52

Result Grid | Filter Rows: Search | Export:

available_75
1429

Result 94 | Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
255	15:35:36	SELECT COUNT(*) AS available_75 FROM question3 WHERE result>0.75	1 row(s) returned	0.0022 sec / 0.00002...

Q5. Create a table with max, min and average price of each property.

```
-- Q5. Create a table with max, min and average price of each property.

56    SET SQL_SAFE_UPDATES=0;
57    UPDATE airbnb_calendar SET price=NULL WHERE price='';
58    UPDATE airbnb_calendar SET price=TRIM(LEADING '$' FROM price);
59    SET SQL_SAFE_UPDATES=1;
60    ALTER TABLE airbnb_calendar MODIFY price DECIMAL(5,2);
61
62    CREATE TABLE question5(
63        id int PRIMARY KEY,
64        min_price DECIMAL(5,2),
65        max_price DECIMAL(5,2),
66        avg_price DECIMAL(5,2)
67    );
68
69    insert into question5
70        select listing_id, min(price), max(price), avg(price) FROM airbnb_calendar
71        group by listing_id;
72
```

100% 20:73

Result Grid | Filter Rows: Search | Edit: | Export/Import: | Fetch rows:

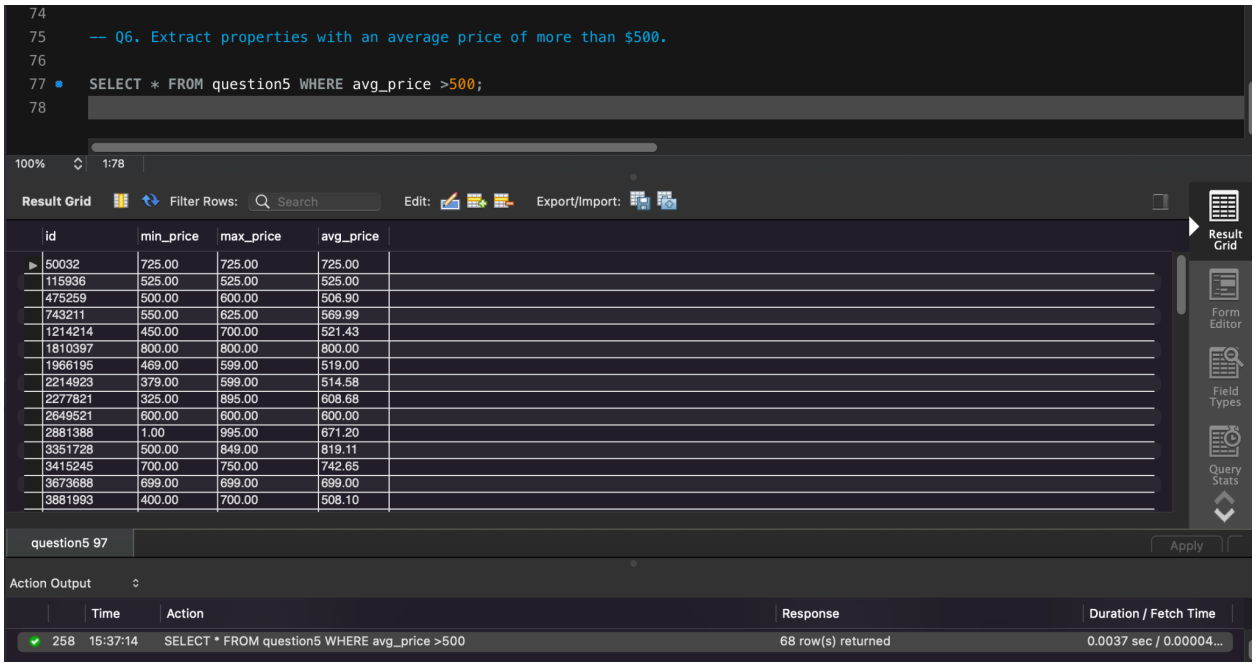
id	min_price	max_price	avg_price
6695	195.00	325.00	197.41
6976	65.00	65.00	65.00
8792	154.00	154.00	154.00
9273	225.00	225.00	225.00
9765	192.00	490.00	236.86
9824	209.00	490.00	222.32
9855	259.00	309.00	266.55
9857	301.00	702.00	363.39

question5 96 | Apply

Action Output

	Time	Action	Response	Duration / Fetch Time
257	15:36:45	SELECT * FROM question5	3585 row(s) returned	0.0022 sec / 0.0029...

Q6. Extract properties with an average price of more than $500.



```
74
75    -- Q6. Extract properties with an average price of more than $500.
76
77 •  SELECT * FROM question5 WHERE avg_price >500;
78
```

100% 1:78

Result Grid | Filter Rows: Search | Edit: | Export/Import:

id	min_price	max_price	avg_price
50032	725.00	725.00	725.00
115936	525.00	525.00	525.00
475259	500.00	600.00	506.90
743211	550.00	625.00	569.99
1214214	450.00	700.00	521.43
1810397	800.00	800.00	800.00
1966195	469.00	599.00	519.00
2214923	379.00	599.00	514.58
2277821	325.00	895.00	608.68
2649521	600.00	600.00	600.00
2881388	1.00	995.00	671.20
3351728	500.00	849.00	819.11
3415245	700.00	750.00	742.65
3673688	699.00	699.00	699.00
3881993	400.00	700.00	508.10

Result Grid | Form Editor | Field Types | Query Stats

question5 97 | Apply

Action Output

	Time	Action	Response	Duration / Fetch Time
258	15:37:14	SELECT * FROM question5 WHERE avg_price >500	68 row(s) returned	0.0037 sec / 0.00004...