

```
-- selecting the database to use
```

```
use project;
```

```
-- creating a new table and using varchar datatype for int to import the data fast and without error as  
the table has null values in it
```

```
create table airbnb(
```

```
id int,
```

```
NAME varchar(255) ,
```

```
host_id bigint ,
```

```
host_identity_verified varchar(100) ,
```

```
host_name varchar(255) ,
```

```
neighbourhood_group varchar(255) ,
```

```
neighbourhood varchar(255) ,
```

```
lat varchar(255) ,
```

```
longg varchar(255) ,
```

```
country varchar(100) ,
```

```
country_code varchar(100) ,
```

```
instant_bookable varchar(100) ,
```

```
cancellation_policy varchar(100) ,
```

```
room_type varchar(255) ,
```

```
Construction_year varchar(255) ,
```

```
price varchar(255) ,
```

```
service_fee varchar(255) ,
```

```
minimum_nights varchar(255) ,
```

```
number_of_reviews varchar(255) ,
```

```
last_review varchar(255) ,
```

```
reviews_per_month varchar(255) ,
review_rate_number varchar(255) ,
calculated_host_listings_count varchar(255) ,
availability_365 varchar(255) ,
house_rules longtext ,
license varchar(255));
```

```
-- now to import the data to the table using load file function -----
```

```
load data infile 'C:/Users/admin/OneDrive/data_analysis/Airbnb_Open_Data_mysql.csv'
```

```
into table project.airbnb
```

```
fields terminated by ',' OPTIONALLY ENCLOSED BY '"'
```

```
lines terminated by '\n'
```

```
ignore 1 rows
```

```
-- this is to import data even if the values are null
```

```
(id, NAME, host_id, host_identity_verified, host_name, neighbourhood_group, neighbourhood,
lat, longg, country, country_code, instant_bookable, cancellation_policy, room_type,
Construction_year, price, service_fee, minimum_nights, number_of_reviews, last_review,
reviews_per_month, review_rate_number, calculated_host_listings_count, availability_365,
house_rules, license)
```

```
set id = nullif(id, ''),
```

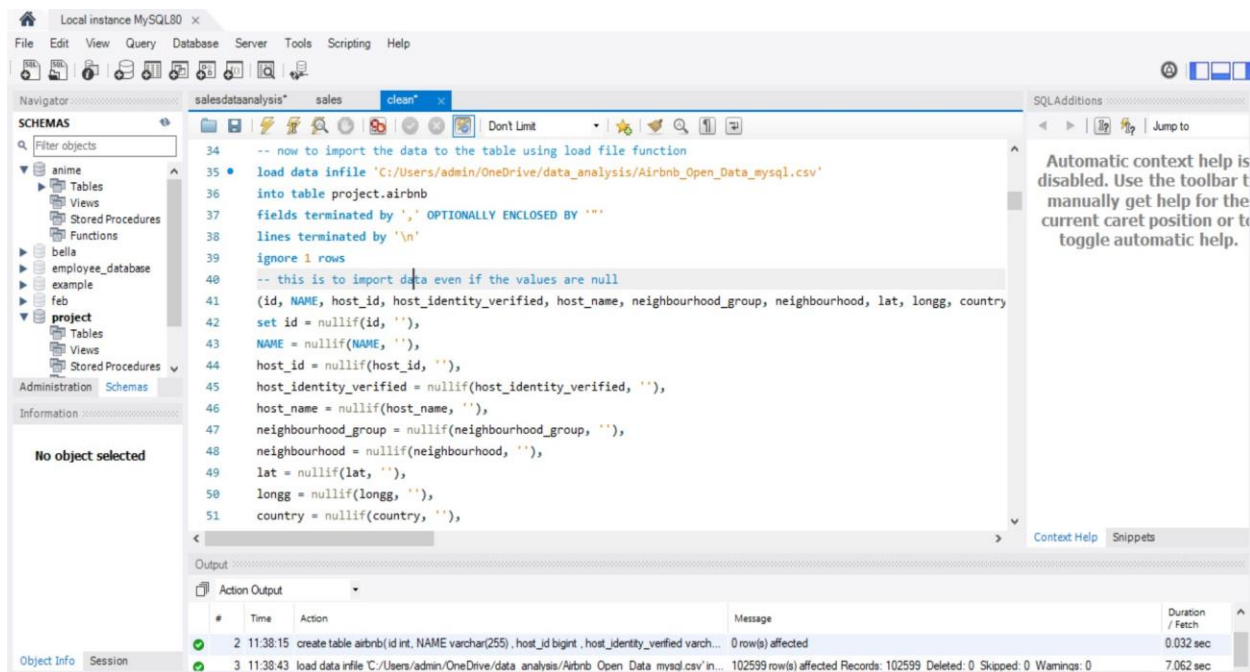
```
NAME = nullif(NAME, ''),
```

```
host_id = nullif(host_id, ''),
```

```
host_identity_verified = nullif(host_identity_verified, ''),
```

```
host_name = nullif(host_name, ''),
```

```
neighbourhood_group = nullif(neighbourhood_group, ""),
neighbourhood = nullif(neighbourhood, ""),
lat = nullif(lat, ""),
longg = nullif(longg, ""),
country = nullif(country, ""),
country_code = nullif(country_code, ""),
instant_bookable = nullif(instant_bookable, ""),
cancellation_policy = nullif(cancellation_policy, ""),
room_type = nullif(room_type, ""),
Construction_year = nullif(Construction_year, ""),
price = nullif(price, ""),
service_fee = nullif(service_fee, ""),
minimum_nights = nullif(minimum_nights, ""),
number_of_reviews = nullif(number_of_reviews, ""),
last_review = nullif(last_review, ""),
reviews_per_month = nullif(reviews_per_month, ""),
review_rate_number = nullif(review_rate_number, ""),
calculated_host_listings_count = nullif(calculated_host_listings_count, ""),
availability_365 = nullif(availability_365, ""),
house_rules = nullif(house_rules, ""),
license = nullif(license, "");
```



-- change the datatype of the columns

alter table airbnb modify column lat double null;

alter table airbnb modify column longg double null;

alter table airbnb modify column construction_year int null;

alter table airbnb modify column price int null;

alter table airbnb modify column service_fee int null;

alter table airbnb modify column minimum_nights int null;

alter table airbnb modify column number_of_reviews int null;

alter table airbnb modify column reviews_per_month int null;

alter table airbnb modify column review_rate_number int null;

alter table airbnb modify column calculated_host_listings_count int null;

alter table airbnb modify column availability_365 int null;

alter table airbnb modify column last_review date null;

-- to view the schema of the table

describe project.airbnb; # all the datatypes have been updated as the correct datatypes

Field	Type	Null	Key	Default	Extra
country	varchar(100)	YES		NULL	
country_code	varchar(100)	YES		NULL	
instant_bookable	varchar(100)	YES		NULL	
cancellation_policy	varchar(100)	YES		NULL	
room_type	varchar(255)	YES		NULL	
construction_year	int	YES		NULL	
price	int	YES		NULL	
service_fee	int	YES		NULL	
minimum_nights	int	YES		NULL	
number_of_reviews	int	YES		NULL	
last_review	date	YES		NULL	
reviews_per_month	int	YES		NULL	
review_rate_number	int	YES		NULL	
calculated_host_listin...	int	YES		NULL	
availability_365	int	YES		NULL	
house_rules	longtext	YES		NULL	
license	varchar(255)	YES		NULL	

-- now to view the data

select * from airbnb;

id	NAME	host_id	host_identity_verified	host_name	neighbourhood_group	neighbourhood
1002755	NULL	85098326012	unconfirmed	Garry	Brooklyn	Clinton Hill
1032817	NULL	18424415847	unconfirmed	Carina	Manhattan	Lower East Side
1033369	NULL	48877238684	NULL	Sophia	Manhattan	Hell's Kitchen
1033922	NULL	71987706184	verified	Rosie	Brooklyn	Greenpoint
1034474	NULL	85649449746	verified	Frederick	Manhattan	SoHo
1035026	NULL	22028915442	NULL	Agata	Brooklyn	Williamsburg
1035579	NULL	51930287368	unconfirmed	George	Manhattan	Chelsea
1036131	NULL	85658272266	unconfirmed	Kelsey	Manhattan	Upper East Side
1036683	NULL	91715353208	NULL	Sabrina	Brooklyn	Prospect Heights
1046072	NULL	73077753757	verified	Arthur	Brooklyn	Williamsburg
1054357	NULL	43086080920	verified	Joyce	Brooklyn	Bushwick
1054909	NULL	8644470108	verified	Sydney	Manhattan	Upper West Side
1055461	NULL	2531706698	NULL	Victoria	Manhattan	Harlem
1056014	NULL	31493374340	verified	NULL	Manhattan	Chelsea
1056566	NULL	47504416702	unconfirmed	NULL	Brooklyn	Bedford-Stuyvesant
1066507	NULL	87894881413	NULL	David	Manhattan	Upper West Side

Result Grid									
			Filter Rows:		Export:		Wrap Cell Content:		Fetch rows:
	price	service_fee	minimum_nights	number_of_reviews	last_review	reviews_per_month	review_rate_number	calculated_host_listings_count	ava
	368	74	30	270	2019-07-05	5	4	1	322
	1024	205	5	19	2019-05-18	0	4	1	NULL
	658	132	2	289	2019-06-09	2	3	2	NULL
	543	109	4	138	2019-06-04	1	1	3	NULL
	256	51	30	21	2019-06-29	0	5	1	NULL
▶	294	59	2	42	2019-06-30	0	3	1	NULL
	779	156	180	5	2018-11-03	0	2	1	NULL
	653	131	2	66	2019-03-30	1	5	2	NULL
	303	61	30	143	2019-01-26	1	1	2	NULL
	1057	211	3	61	2019-04-22	1	3	4	NULL
	380	76	NULL	118	2019-06-18	1	NULL	2	92
	343	69	NULL	NULL	2019-07-07	1	NULL	1	55
	160	32	NULL	1	2018-10-09	0	NULL	1	158
	528	106	NULL	30	2019-05-01	0	NULL	2	64
	739	148	NULL	139	2018-10-28	1	NULL	2	184
	1154	231	6	27	2019-04-27	0	NULL	NULL	237
	---	---	---	---	---	---	---	---	---

-- first is to check whether there is a duplicate

select id, count(*) as con from airbnb

group by id

having count(*) > 1; -- there is 541 duplicate id number

Result Grid		
	id	con
▶	6026161	2
	6026714	2
	6027266	2
	6027818	2
	6028371	2
	6028923	2
	6029475	2
	6030028	2
	6030580	2
	6031132	2
	6031684	2
	6032237	2
	6032789	2
	6033341	2
	6033894	2

-- this is used to create temp_table that has all unique values

create temporary table temp_table as

select distinct * from airbnb; -- this will select all the distinct values from all the columns

-- then deleting all the values from the existing table

delete from airbnb;

-- inserting all the unique values from the temp table

insert into airbnb

select * from temp_table; -- 102058 records affected

#	Time	Action	Message	Duration / Fetch
22	11:46:03	delete from airbnb	102599 row(s) affected	3.157 sec
23	11:46:10	insert into airbnb select * from temp_table	102058 row(s) affected Records: 102058 Duplicates: 0 Warnings: 0	5.062 sec

-- lets delete the temp table

drop table temp_table;

-- lets fill the missing data for some columns to make the dataset clean

update airbnb set NAME = 'room'

where name is null; -- 249 rows updated. as the name depends on the type of structure the room has so there's no way to fill the accurate data

id	NAME	host_id	host_identity_verified	host_name	neighbourhood_group	neighbourhood
1031160	Greenpoint Place...Has It All!	52857769781	unconfirmed	Madaline	Brooklyn	Greenpoint
1031713	Beautiful Apartment in Manhattan!!!	39829029279	verified	Richard	Manhattan	Kips Bay
1032265	Williamsburg 1 bedroom Apartment	84933495312	unconfirmed	Paul	Brooklyn	Williamsburg
1032817	room	18424415847	unconfirmed	Carina	Manhattan	Lower East Side
1033369	room	48877238684	NULL	Sophia	Manhattan	Hell's Kitchen
1033922	room	71987706184	verified	Rosie	Brooklyn	Greenpoint
1034474	room	85649449746	verified	Frederick	Manhattan	SoHo
1035026	room	22028915442	NULL	Agata	Brooklyn	Williamsburg
1035579	room	51930287368	unconfirmed	George	Manhattan	Chelsea
1036131	room	85658272266	unconfirmed	Kelsey	Manhattan	Upper East Side
1036683	room	91715353208	NULL	Sabrina	Brooklyn	Prospect Heights
1037236	Double Room w Private Deck Clinton Hill Best A...	71078326464	unconfirmed	William	Brooklyn	Clinton Hill
1037788	Your Heaven in Hells Kitchen	20543275700	verified	Melissa	Manhattan	Hell's Kitchen
1038340	Light-filled 2B duplex in the heart of Park Slope!	42969787263	verified	Ellia	Brooklyn	Park Slope
1038892	Sunny, Modern room in East Village!	79667908130	unconfirmed	Walter	Manhattan	East Village
1039445	"HAVEN LOFT" - Entire Floor - Six Windows - Bri...	19095545297	unconfirmed	Sawyer	Brooklyn	Williamsburg
1039907	Love Room in Amazing East Village Apt	60704375200	unconfirmed	Melanie	Manhattan	East Village

```
update airbnb set host_identity_verified = 'under review'
```

```
where host_identity_verified is null; -- 289 rows updated
```

```
-- to find the host name with the neighbourhood group
```

```
select * from airbnb
```

```
where host_name in (select distinct(host_name) from airbnb); # there are thousand of records so it  
difficult to fill the missing data
```

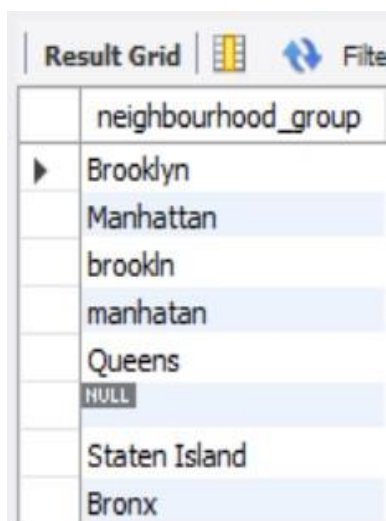
```
-- lets update the missing host name with under review
```

```
update airbnb set host_name = 'under review'
```

```
where host_name is null; # 404 rows updated
```

```
-- to find the distinct neighbourhood_group column
```

```
select distinct neighbourhood_group from airbnb; # we have Brooklyn, brooklyn, Manhattan ,  
manhattan. had 8 rows result
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a SQL query. The first column is labeled 'neighbourhood_group'. The rows contain the following values: Brooklyn, Manhattan, brooklyn, manhatan, Queens, NULL, Staten Island, and Bronx. The rows are alternatingly highlighted with light blue and white backgrounds.

neighbourhood_group
Brooklyn
Manhattan
brooklyn
manhatan
Queens
NULL
Staten Island
Bronx

-- lets update the column ---

```
update airbnb set neighbourhood_group = 'Brooklyn'
```

```
where neighbourhood_group = 'brookln';
```

```
update airbnb set neighbourhood_group = 'Manhattan'
```

```
where neighbourhood_group = 'manhatan';
```

-- lets check if there are records changed

```
select neighbourhood_group from airbnb
```

```
where neighbourhood_group = 'manhaten'; -- error output so the data have been updated
```

```
select distinct neighbourhood, neighbourhood_group from airbnb
```

```
where neighbourhood_group is null; # 15 rows missing data
```

Result Grid			Filter Rows:
	neighbourhood	neighbourhood_group	
▶	Washington Heights	NULL	
	Clinton Hill	NULL	
	East Village	NULL	
	Upper East Side	NULL	
	Woodside	NULL	
	Williamsburg	NULL	
	Bushwick	NULL	
	Prospect Heights	NULL	
	Chelsea	NULL	
	East Harlem	NULL	
	Eastchester	NULL	
	Harlem	NULL	

---- this is to find the missing value from the above query

```
select distinct neighbourhood_group, neighbourhood from airbnb
```

```
where neighbourhood in('Washington Heights', -- man
```

```
'Clinton Hill', -- br
```

```
'East Village', -- man
```

```
'Upper East Side', -- man
```

```
'Woodside', -- queens
```

```
'Williamsburg', -- brook
```

```
'Bushwick', -- bro
```

```
'Prospect Heights', -- br
```

```
'Chelsea', -- br
```

```
'East Harlem', -- man
```

```
'Eastchester', -- bronx
```

```
'Harlem', -- man
```

```
'Chinatown', -- man
```

```
'Queens Village', -- queens
```

```
'Bedford-Stuyvesant', -- brooklyn
```

```
'Upper West Side'); -- manhattan
```

found the records from the previous query that had missing values and based on that updated the missing values

-- used case statement to update multiple rows with different values that had null

update airbnb set

neighbourhood_group =

case

when neighbourhood = 'Clinton Hill' then 'brooklyn'

when neighbourhood = 'East Village' then 'Manhattan'

when neighbourhood = 'Upper East Side' then 'Manhattan'

when neighbourhood = 'East Village' then 'Manhattan'

when neighbourhood = 'Woodside' then 'Queens'

when neighbourhood = 'Williamsburg' then 'Brooklyn'

when neighbourhood = 'Bushwick' then 'Brooklyn'

when neighbourhood = 'Prospect Heights' then 'Brooklyn'

when neighbourhood = 'Chelsea' then 'Brooklyn'

when neighbourhood = 'East Harlem' then 'Manhattan'

when neighbourhood = 'Eastchester' then 'Bronx'

when neighbourhood = 'Harlem' then 'Manhattan'

when neighbourhood = 'Chinatown' then 'Manhattan'

when neighbourhood = 'Queens Village' then 'Queens'

when neighbourhood = 'Bedford-Stuyvesant' then 'Brooklyn'

when neighbourhood = 'Upper West Side' then 'Manhattan' end

where neighbourhood in ('Clinton Hill', 'East Village', 'Upper East Side', 'East Village',

'Woodside', 'Williamsburg', 'Bushwick', 'Prospect Heights', 'Chelsea', 'East Harlem',

'Eastchester', 'Harlem', 'Chinatown', 'Queens Village', 'Bedford-Stuyvesant', 'Upper West Side')

and neighbourhood_group is null;

-- the null values in the neighbourhood group had been filled with relevant data from the neighbour column

	neighbourhood_group	neighbourhood
▶	Manhattan	Harlem
	Brooklyn	Clinton Hill
	Manhattan	East Harlem
	Brooklyn	Bedford-Stuyvesant
	Manhattan	Upper West Side
	Manhattan	Chinatown
	Brooklyn	Williamsburg
	Manhattan	Chelsea
	Manhattan	East Village
	Brooklyn	Bushwick
	Manhattan	Upper East Side
	Brooklyn	Prospect Heights

-- to check and update neighbourhood column missing data and update it

```
select neighbourhood, neighbourhood_group from airbnb
```

where neighbourhood is null; # 2 same rows missing data. since there are multiple neighbour for each neighbourhood group, we can update it as under review

Result Grid		
	neighbourhood	neighbourhood_group
▶	NULL	Brooklyn
	NULL	Manhattan
	NULL	Manhattan
	NULL	Manhattan
	NULL	Brooklyn
	NULL	Manhattan
	NULL	Manhattan
	NULL	Manhattan
	NULL	Brooklyn
	NULL	Brooklyn
	NULL	Brooklyn
	NULL	Brooklyn

```
update airbnb set neighbourhood = 'under review', neighbourhood = 'under review'
```

where neighbourhood_group in ('Brooklyn', 'Manhattan') and neighbourhood is null; -- all records have been updated

#	Time	Action	Message	Duration / Fetch
39	11:53:21	select neighbourhood, neighbourhood_group from airbnb where neighbourhood is null	16 row(s) returned	0.125 sec / 0.000 sec
40	11:53:54	update airbnb set neighbourhood = 'under review', neighbourhood = 'under review' wh...	16 row(s) affected Rows matched: 16 Changed: 16 Warnings: 0	0.453 sec

-- to fill the lat and longg values using neighbourhood column data

select distinct neighbourhood , lat, longg from airbnb

where lat is null and longg is null;

	neighbourhood	lat	longg
►	Crown Heights	NULL	NULL
	Greenpoint	NULL	NULL
	East Village	NULL	NULL
	West Village	NULL	NULL
	Elmhurst	NULL	NULL
	Flatiron District	NULL	NULL
	Upper West Side	NULL	NULL

select distinct neighbourhood , lat, longg from airbnb

where neighbourhood in ('Crown Heights','Elmhurst','Greenpoint','East Village','West Village','Flatiron District','Upper West Side');

Crown Heights 40.68 -73.95 Elmhurst 40.74 -73.88 Greenpoint 40.74 -73.95 East Village 40.73
-73.98 West Village 40.73 -74 Flatiron District 40.74 -73.99 Upper West Side 40.79
-73.96 this is the values to be updated

update airbnb set lat = 40.68

where neighbourhood = 'Crown Heights' and lat is null;

update airbnb set longg = -73.95

where neighbourhood = 'Crown Heights' and longg is null;

update airbnb set lat = 40.74, longg = -73.88

where neighbourhood = 'Elmhurst' and lat is null and longg is null;

update airbnb set lat = 40.74, longg = -73.95

where neighbourhood = 'Greenpoint' and lat is null and longg is null;

update airbnb set lat = 40.73, longg = -73.98

where neighbourhood = 'East Village' and lat is null and longg is null;

update airbnb set lat = 40.73, longg = -74

where neighbourhood = 'West Village' and lat is null and longg is null;

update airbnb set lat = 40.74, longg = -73.99

where neighbourhood = 'Flatiron District' and lat is null and longg is null;

update airbnb set lat = 40.79, longg = -73.96

where neighbourhood = 'Upper West Side' and lat is null and longg is null;

Result Grid			
Filter Rows:			
	neighbourhood	lat	longg
▶	Upper West Side	40.8	-73.97
	Upper West Side	40.8	-73.96
	West Village	40.74	-74.01
	Crown Heights	40.68	-73.95
	East Village	40.73	-73.99
	Greenpoint	40.72	-73.94
	East Village	40.72	-73.98
	Upper West Side	40.79	-73.98
	Upper West Side	40.79	-73.97
	Greenpoint	40.74	-73.95
	East Village	40.73	-73.98

-- lets update the missing values in country and country code column

select country , country_code from airbnb; # so both column have missing data relating to united states

Result Grid	
Filter Rows:	
country	country_code
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
United States	US
United States	US
United States	US
United States	US
United States	US

-- so lets update both the entire column as the data is showing us

update airbnb set country = 'united states';

update airbnb set country_code = 'us'; -- all the missing data have been updated

-- lets update instant bookable and cancellation policy column since we cannot derive the relationship between them lets update it to under review for null values

update airbnb set instant_bookable = 'under review' where instant_bookable is null; -- 105 rows values updated

update airbnb set cancellation_policy = 'under review' where cancellation_policy is null; -- 76 rows values updated

all the missing values values have been updated

-- lets update construction year since there is no correlation to fill the missing data lets update it as 1 since its a date datatype

update airbnb set construction_year = 1

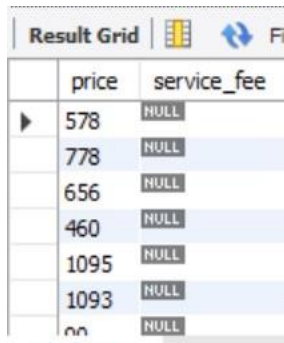
where construction_year is null; # 214 rows updated

all the missing date have been filled

-- lets fill the missing data from price and service fee column

select price, service_fee from airbnb

where service_fee is null;-- 273 null rows

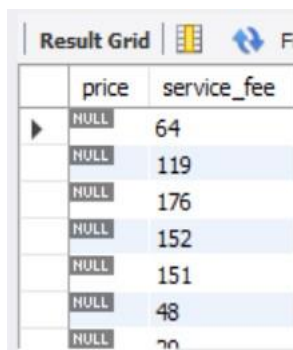


A screenshot of a database result grid titled "Result Grid". It shows a table with two columns: "price" and "service_fee". The "service_fee" column contains "NULL" for all rows shown. The "price" column has values: 578, 778, 656, 460, 1095, 1093, and 00.

	price	service_fee
▶	578	NULL
	778	NULL
	656	NULL
	460	NULL
	1095	NULL
	1093	NULL
	00	NULL

select price, service_fee from airbnb

where price is null; -- 247 null rows



A screenshot of a database result grid titled "Result Grid". It shows a table with two columns: "price" and "service_fee". The "price" column contains "NULL" for all rows shown. The "service_fee" column has values: 64, 119, 176, 152, 151, 48, and 00.

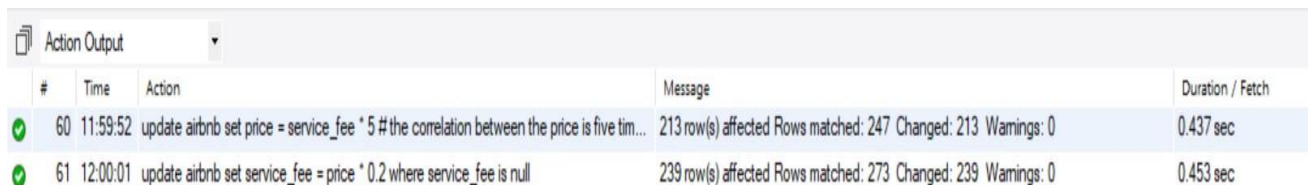
	price	service_fee
▶	NULL	64
	NULL	119
	NULL	176
	NULL	152
	NULL	151
	NULL	48
	NULL	00

update airbnb set price = service_fee * 5; # the correlation between the price is five times the rate of price, for example 100 is the service fee so the price is 100 * 5 and the result is the actual price.

where price is null; -- 213 rows updated

update airbnb set service_fee = price * 0.2

where service_fee is null;-- 239 rows updated



A screenshot of an "Action Output" table. It has four columns: "#", "Time", "Action", and "Message". The first row shows an update of price based on service_fee, affecting 213 rows. The second row shows an update of service_fee based on price, affecting 239 rows.

#	Time	Action	Message	Duration / Fetch
60	11:59:52	update airbnb set price = service_fee * 5 # the correlation between the price is five tim...	213 row(s) affected Rows matched: 247 Changed: 213 Warnings: 0	0.437 sec
61	12:00:01	update airbnb set service_fee = price * 0.2 where service_fee is null	239 row(s) affected Rows matched: 273 Changed: 239 Warnings: 0	0.453 sec

select price, service_fee from airbnb

where price is null and service_fee is null; -- 34 rows null

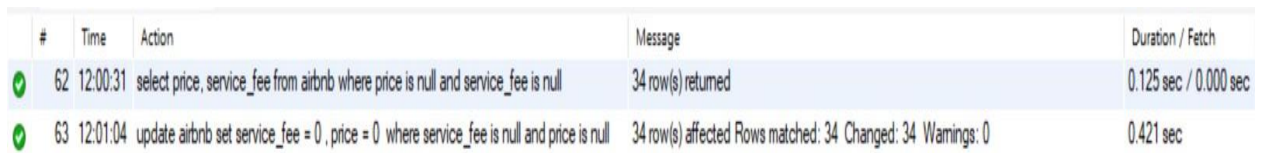


The screenshot shows a 'Result Grid' window with two columns: 'price' and 'service_fee'. There are 10 rows, all containing 'NULL' values. The window has a toolbar with icons for grid, chart, and refresh, and a filter icon.

	price	service_fee
▶	NULL	NULL
	NULL	NULL
	NULL	NULL
	NULL	NULL
	NULL	NULL
	NULL	NULL
	NULL	NULL
	NULL	NULL
	NULL	NULL
	NULL	NULL

update airbnb set service_fee = 0 , price = 0

where service_fee is null and price is null; -- all the rows which was missing values are updated with 0



The screenshot shows a log table with columns: #, Time, Action, Message, and Duration / Fetch. It contains two entries, both marked with a green checkmark icon.

#	Time	Action	Message	Duration / Fetch
62	12:00:31	select price, service_fee from airbnb where price is null and service_fee is null	34 row(s) returned	0.125 sec / 0.000 sec
63	12:01:04	update airbnb set service_fee = 0 , price = 0 where service_fee is null and price is null	34 row(s) affected Rows matched: 34 Changed: 34 Warnings: 0	0.421 sec

-- lets update the minimum nights column since there no relationship from other columns to find the missing value lets update it with 1

update airbnb set minimum_nights = 1

where minimum_nights is null; -- 400 rows updated

-- lets update the number of reviews, last review , reviews per month, review rate number and calculated host columns since there no relationship from other columns to find the missing value lets update it with 0

update airbnb set number_of_reviews = 1 where number_of_reviews is null; -- 183 rows updated

alter table airbnb modify column last_review varchar(100) null; -- since it was date datatype to fill the missing values, converted it to varchar

update airbnb set last_review = 'under review' where last_review is null; -- 15832 rows updated

select last_review from airbnb

where last_review > '2022';-- since there values where last review year is after 2022 so that data must be updated with under review

	last_review
▶	2058-06-16
	2040-06-16
	2026-03-28
	2025-06-26
	2024-08-15
	2022-05-21
	2022-05-05
	2022-05-04

update airbnb set last_review = 'under review' where last_review > '2022'; -- 26206 rows updated

--- Since the last review column would be updated later so I left it as varchar ----

update airbnb set reviews_per_month = 1 where reviews_per_month is null; -- 15818 rows updated

update airbnb set review_rate_number = 1 where review_rate_number is null; -- 319 rows updated

update airbnb set calculated_host_listings_count = 1 where calculated_host_listings_count is null; -- 319 rows updated

-- lets update the availability_365 column as there are values that exceed the 365days so lets change that values to 365

	availability_365
▶	3677
	426
	426
	426
	426
	426
	426
	426
	426

update airbnb set availability_365 = 365

where availability_365 > 365; -- 2754 rows updated

✓	74	12:04:42	update airbnb set availability_365 = 365 where availability_365 > 365	2754 row(s) affected Rows matched: 2754 Changed: 2754 Warnings: 0	0.485 sec
---	----	----------	---	---	-----------

-- lets update the house_rules column since there are null values which can mean either there no rules or not mentioned

update airbnb set house_rules = 'no rules or not mentioned'

where house_rules is null; -- 51842 rows updated

-- since whole license column is null lets update it with under review

update airbnb set license = 'under review'; -- 102058 rows updated

----- XXXXXX -----

-- all the data have been cleaned and missing values have been filled ---