Handling Missing Data in ETL

SECTION A – THEORETICAL QUESTIONS

QUESTION 1

What are the most common reasons for missing data in ETL pipelines?

ANSWER

1) Source system data gaps

Optional fields left blank by users
Data not captured due to application logic
Legacy systems with incomplete records

2) extraction failures

Partial job failures
Network or API timeouts
File corruption or missing files

3) Schema or structure changes

Columns renamed, removed, or added
Data type changes

4. Incorrect data mapping

Source column not mapped to target
Mapping logic errors

5. Filtering during transformatio

WHERE clauses removing records
Business rules excluding data

QUESTION 2

Why is blindly deleting rows with missing values considered a bad practice in ETL?

ANSWER

1) Loss of valuable information

Missing values don’t always mean the entire record is useless.
Example:
A customer record missing phone number may still have valid name, email, and purchase data

2) Data bias and wrong analytics

Rows with missing values are often not random.

3) Incorrect KPIs and reporting

Dropping records reduces counts, totals, and averages.

4) Breaks referential integrity

Deleting rows can break relationships between tables.

5) Hides upstream data issues

Blind deletion masks problems in:
Source systems
ETL logic
Data collection processes

QUESTION 3

Explain the difference between: Listwise deletion Column deletion

ANSWER

If a row has at least one missing value, the entire row is removed.

Checks row by row
One null (whole record deleted)

If a column has many missing values, the entire column is removed.

Evaluate missing % per column
Drop columns exceeding threshold (e.g., 60%)

Listwise deletion removes entire records when any value is missing, while column deletion removes entire attributes when a column contains excessive missing data. Both cause data loss and should be used only after business validation.

QUESTION 4

Why is median imputation preferred over mean imputation for skewed data such as income?

ANSWER

1) Income data is highly skewed

Most people earn moderate incomes, while a few earn extremely high amounts

2) Mean is sensitive to outliers

A few very high incomes can dramatically increase the mean.

3) Median represents the “typical” value

The median reflects the middle of the distribution, not the extremes

4) Prevents distortion of analysis & models

Mean imputation:
Inflates income values
Skews KPIs like average salary
Misleads ML models

QUESTION 5

What is forward fill and in what type of dataset is it most useful?

ANSWER

If a value is missing at time t, we copy the value from time t-1.

Date	Stock Price
Jan 1	100
Jan 2	NULL
Jan 3	NULL
Jan 4	105

AFTER

Date	Stock Price
Jan 1	100
Jan 2	100
Jan 3	100
Jan 4	105

Forward fill is best for time-series or sequential datasets where values change gradually and the previous value is still meaningful.

Data has a clear order (time, sequence)

Values change gradually

Missing values imply “no new value recorded”

QUESTION 6

Why should flagging missing values be done before imputation in an ETL workflow?

ANSWER

You don't lose information about "missingness"

Once you impute, you can no longer tell which values were originally missing.

Missingness itself can be meaningful

The fact that a value is missing can carry business or behavioral insight.

Better debugging and data audits

If reports look off later, flags help answer:
Which data was original?
Which data was imputed?
How much was imputed?

Protects downstream analytics & ML models

Imputed values look like real data.
Model bias checks
Sensitivity analysis
Feature engineering using missing indicators

Supports different business rules later

Different teams may want different treatments.

Improves trust and governance

Clear visibility into data modifications builds stakeholder confidence and meets governance requirements.

QUESTION 7

Consider a scenario where income is missing for many customers.
How can this missingness itself provide business insights?

ANSWER

1) Income missing may reflect customer behavior

High-income or privacy-conscious customers often choose not to disclose income.

2) Missingness can indicate trust or engagement levels

Customers who skip income fields may:
Not trust the platform yet
Be early-stage users
Have low engagement

3) Segment-specific patterns

Missing income might be concentrated in:
Certain age groups
Specific regions
Particular acquisition channels

4) Risk & eligibility signals

In lending or insurance:
Missing income can itself be a risk indicator

5) Strategic ETL takeaway

Instead of deleting or blindly imputing:
Add income_missing_flag = 1
Analyze conversion, churn, or revenue by this flag

SECTION B – PRACTICAL QUESTIONS

QUESTION 8

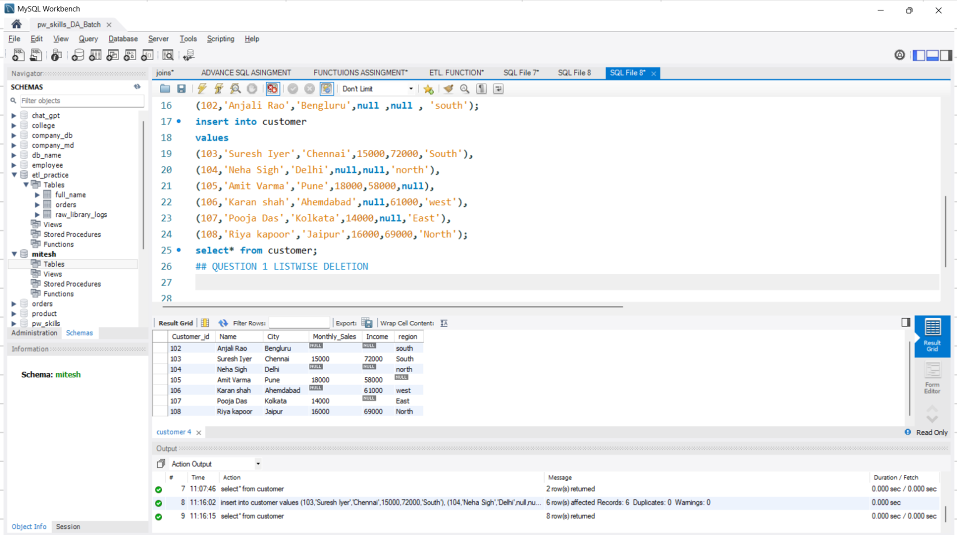
Listwise Deletion Remove all rows where Region is missing.

Identify affected rows

Show the dataset after deletion

Mention how many records were lost

ANSWER


MySQL Workbench
pw_skills_DA_Batch
File Edit View Query Database Server Tools Scripting Help
joins* ADVANCE SQL ASINGMENT FUNCTUIONS ASSINGMENT* ETL. FUNCTION* SQL File 7* SQL File 8 SQL File 8*
Don't Limit
Schema: mitesh

```
(102,'Anjali Rao','Bengluru',null ,null , 'south');
insert into customer
values
(103,'Suresh Iyer','Chennai',15000,72000,'South'),
(104,'Neha Sigh','Delhi',null,null,'north'),
(105,'Amit Varma','Pune',18000,58000,null),
(106,'Karan shah','Ahemdabad',null,61000,'west'),
(107,'Pooja Das','Kolkata',14000,null,'East'),
(108,'Riya kapoor','Jaipur',16000,69000,'North');
select* from customer;
## QUESTION 1 LISTWISE DELETION
```

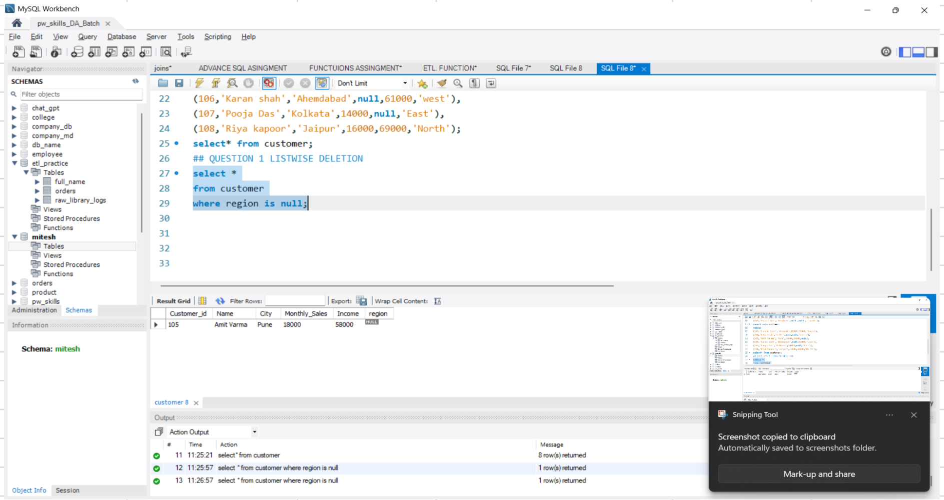
Customer_id	Name	City	Monthly_Sales	Income	region
102	Anjali Rao	Bengluru	NULL	NULL	south
103	Suresh Iyer	Chennai	15000	72000	South
104	Neha Sigh	Delhi	NULL	NULL	north
105	Amit Varma	Pune	18000	58000	NULL
106	Karan shah	Ahemdabad	NULL	61000	west
107	Pooja Das	Kolkata	14000	NULL	East
108	Riya kapoor	Jaipur	16000	69000	North

customer 4

```
#  Time      Action                                                                          Message                                                        Duration / Fetch
7  11:07:46  select* from customer                                                           2 row(s) returned                                              0.000 sec / 0.000 sec
8  11:16:02  insert into customer values (103,'Suresh Iyer','Chennai',15000,72000,'South'), (104,'Neha Sigh','Delhi',null,nu...  6 row(s) affected Records: 6  Duplicates: 0  Warnings: 0  0.000 sec
9  11:16:15  select* from customer                                                           8 row(s) returned                                              0.000 sec / 0.000 sec
```

1)Identify affected rows

2) AFTER THE DELETION RESULT


MySQL Workbench
pw_skills_DA_Batch
File Edit View Query Database Server Tools Scripting Help
joins* ADVANCE SQL ASINGMENT FUNCTUIONS ASSINGMENT* ETL. FUNCTION* SQL File 7* SQL File 8 SQL File 8*
Don't Limit
Schema: mitesh

```
(106,'Karan shah','Ahemdabad',null,61000,'west'),
(107,'Pooja Das','Kolkata',14000,null,'East'),
(108,'Riya kapoor','Jaipur',16000,69000,'North');
select* from customer;
## QUESTION 1 LISTWISE DELETION
select *
from customer
where region is null;
```

Customer_id	Name	City	Monthly_Sales	Income	region
105	Amit Varma	Pune	18000	58000	NULL

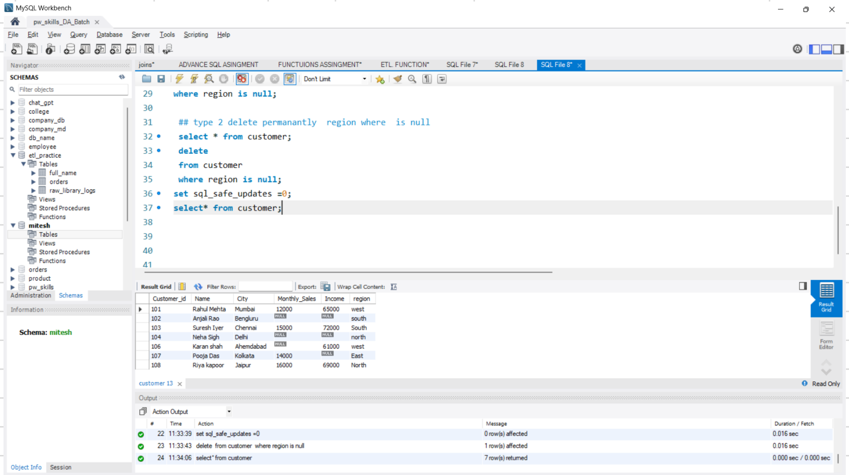
customer 8

```
#   Time      Action                                      Message
11  11:25:21  select* from customer                       8 row(s) returned
12  11:25:57  select * from customer where region is null 1 row(s) returned
13  11:26:57  select * from customer where region is null 1 row(s) returned
```

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Mark-up and share


MySQL Workbench
pw_skills_DA_Batch
File Edit View Query Database Server Tools Scripting Help
joins* ADVANCE SQL ASINGMENT FUNCTUIONS ASSINGMENT* ETL. FUNCTION* SQL File 7* SQL File 8 SQL File 8*
Don't Limit
Schema: mitesh

```
where region is null;

 ## type 2 delete permanantly  region where  is null
 select * from customer;
 delete
 from customer
 where region is null;
set sql_safe_updates =0;
select* from customer;
```

Customer_id	Name	City	Monthly_Sales	Income	region
101	Rahul Mehta	Mumbai	12000	65000	west
102	Anjali Rao	Bengluru	NULL	NULL	south
103	Suresh Iyer	Chennai	15000	72000	South
104	Neha Sigh	Delhi	NULL	NULL	north
106	Karan shah	Ahemdabad	NULL	61000	west
107	Pooja Das	Kolkata	14000	NULL	East
108	Riya kapoor	Jaipur	16000	69000	North

customer 13

```
#   Time      Action                                    Message             Duration / Fetch
22  11:33:39  set sql_safe_updates =0                   0 row(s) affected   0.016 sec
23  11:33:43  delete  from customer  where region is null  1 row(s) affected   0.016 sec
24  11:34:06  select* from customer                     7 row(s) returned   0.000 sec / 0.000 sec
```

QUESTION 9

Imputation

Handle missing values in Monthly_Sales using:

Forward Fill

Tasks:

Apply forward fill

Show before vs after values

Explain why forward fill is suitable here

ANSWER

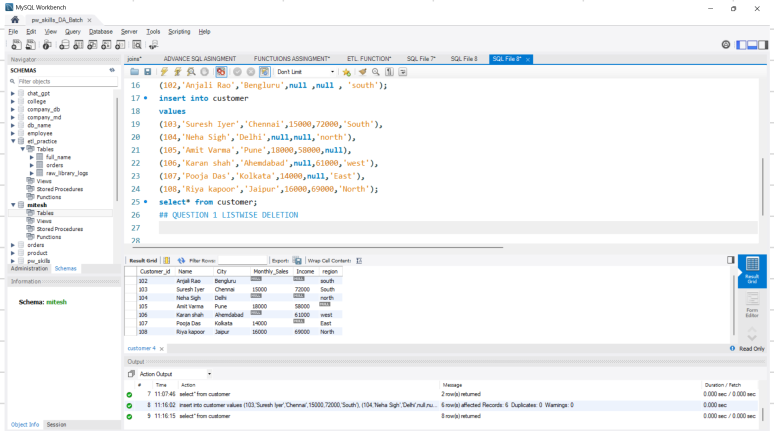
BEFORE

Customer_ID	Name	City	Monthly_Sales	Income	Region
1	Rahul Mehta	Mumbai	12000	65000	West
2	Neha Singh	Delhi	NaN	NaN	North
3	Anjali Rao	Bengaluru	NaN	NaN	South
4	Amit Verma	Pune	18000	58000	NaN
5	Pooja Das	Kolkata	14000	NaN	East
6	Suresh Iyer	Chennai	15000	72000	South
7	Karan Shah	Ahmedabad	NaN	61000	West
8	Riya Kapoor	Jaipur	16000	69000	North

TECHNIQUE 1	MEAN	15000

AFTER

Customer_ID	Name	City	Monthly_Sales	Income	Region
1	Rahul Mehta	Mumbai	12000	65000	West
2	Neha Singh	Delhi	15000	NaN	North
3	Anjali Rao	Bengaluru	15000	NaN	South
4	Amit Verma	Pune	18000	58000	NaN
5	Pooja Das	Kolkata	14000	NaN	East
6	Suresh Iyer	Chennai	15000	72000	South
7	Karan Shah	Ahmedabad	15000	61000	West
8	Riya Kapoor	Jaipur	16000	69000	North



```
(102,'Anjali Rao','Bengluru',null ,null , 'south');
insert into customer
values
(103,'Suresh Iyer','Chennai',15000,72000,'South'),
(104,'Neha Sigh','Delhi',null,null,'north'),
(105,'Amit Varma','Pune',18000,58000,null),
(106,'Karan shah','Ahemdabad',null,61000,'west'),
(107,'Pooja Das','Kolkata',14000,null,'East'),
(108,'Riya kapoor','Jaipur',16000,69000,'North');
select* from customer;
## QUESTION 1 LISTWISE DELETION
```

Customer_id	Name	City	Monthly_Sales	Income	region
102	Anjali Rao	Bengluru	NULL	NULL	south
103	Suresh Iyer	Chennai	15000	72000	South
104	Neha Sigh	Delhi	NULL	NULL	north
105	Amit Varma	Pune	18000	58000	NULL
106	Karan shah	Ahemdabad	NULL	61000	west
107	Pooja Das	Kolkata	14000	NULL	East
108	Riya kapoor	Jaipur	16000	69000	North

#	Time	Action	Message	Duration / Fetch
7	11:07:46	select* from customer	2 row(s) returned	0.000 sec / 0.000 sec
8	11:16:02	insert into customer values (103,'Suresh Iyer','Chennai',15000,72000,'South'), (104,'Neha Sigh','Delhi',null,nu...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec
9	11:16:15	select* from customer	8 row(s) returned	0.000 sec / 0.000 sec

Explain why forward fill is suitable here

DATA IS A TIME SERIES

PREVIOUS VALUES IS LOGICALLY VALID FOR FORWORD FILL

SALES DATA CHANGE OVER THE TIME

QUESTION 10

Flagging Missing Data

Create a flag column for missing Income

Tasks:

Create Income_Missing_Flag (0 = present, 1 = missing)

Show updated dataset

Count how many customers have missing income

AFTER CHANGING THE MISSING INCOME

FLAGGING MISSING DATA

ANSWER

Customer_ID	Name	City	Monthly_Sales	Income	MISSING INCOME	Region
1	Rahul Mehta	Mumbai	12000	65000	0	West
2	Neha Singh	Delhi	NaN	NaN	1	North
3	Anjali Rao	Bengaluru	NaN	NaN	1	South
4	Amit Verma	Pune	18000	58000	0	NaN
5	Pooja Das	Kolkata	14000	NaN	1	East
6	Suresh Iyer	Chennai	15000	72000	0	South
7	Karan Shah	Ahmedabad	NaN	61000	0	West
8	Riya Kapoor	Jaipur	16000	69000	0	North

COUNT HOW MANY CUSTOMERS HAVE MISSING INCOME =3

TOTAL NUMBERS OF MISSING DATA = '3'