

Practical Exam: Sample SQL Associate

Tech Solutions Inc. is a leading technology company specializing in software development and IT consulting services. The company prides itself on delivering innovative solutions to clients across various industries. With a dedicated team of skilled professionals, TechSolutions has earned a reputation for excellence in the tech industry.

Tech Solutions Inc. has been experiencing a decline in customer satisfaction ratings over the past few months. Customer feedback surveys and support tickets indicate an increase in dissatisfaction among clients. The company is concerned about this trend as it directly impacts customer retention, reputation, and overall business growth.

You are working with the customer support team to provide data to managers to help the company take proactive measures to address these concerns effectively.

Data

The following schema diagram shows the tables available.

Support		Survey	
id	int	survey_id	int
customer_id	int	customer_id	int
category	str	rating	int
status	str	timestamp	int
creation_date	str		
response_time	int		
resolution_time	int		

Unknown integration DataFrame as `Support_overview`

-- Overview of the support table: preview first 10 rows

```
SELECT *
FROM support
LIMIT 10;
```

...	↑↓	...	↑↓	cus...	...	↑↓	category	...	↑↓	sta...	...	↑↓	creation_date	...	↑↓	respon...	...	↑↓	resolution...	...
0		1		1062			Installation Problem			In Progress			2023-01-26T00:00:00.000				6		0 hours	
1		2		892			Billing enquiry			Open			2023-06-18T00:00:00.000				3		0 hours	
2		3		433			Feedback			Open			2023-08-17T00:00:00.000				1		0 hours	
3		6		764			Billing enquiry			Open			2023-01-16T00:00:00.000				3		0 hours	
4		7		1144			Billing enquiry			Open			2023-06-01T00:00:00.000				2		0 hours	
5		8		288			Feedback			Open			2023-01-22T00:00:00.000				2		0 hours	
6		9		1495			Bug			In Progress			2023-02-05T00:00:00.000				1		0 hours	
7		10		1090			Bug			In Progress			2023-05-09T00:00:00.000				3		0 hours	
8		11		1397			Feedback			In Progress			2023-09-17T00:00:00.000				2		0 hours	
9		12		54			Feedback			Open			2023-09-13T00:00:00.000				2		0 hours	

Rows: 10

 Expand

Unknown integration DataFrame as Survey_overview

-- Overview of the survey table: preview first 10 rows

```
SELECT *
FROM survey
LIMIT 10;
```

index	...	↑↓	id	...	↑↓	customer_id	...	↑↓	rating	...	↑↓	timestamp	...	↑↓
0			1			693			3			2023-12-01T00:00:00.000		
1			2			1814			5			2023-12-01T00:00:00.000		
2			3			1932			5			2023-12-01T00:00:00.000		
3			4			1789			5			2023-12-01T00:00:00.000		
4			5			1332			2			2023-12-01T00:00:00.000		
5			6			400			2			2023-12-01T00:00:00.000		
6			7			139			2			2023-12-01T00:00:00.000		
7			8			948			4			2023-12-01T00:00:00.000		
8			9			1375			5			2023-12-01T00:00:00.000		
9			10			33			2			2023-12-01T00:00:00.000		

Rows: 10

↗ Expand

Unknown integration DataFrame as tabl

-- Understanding the table structure and data types of all the columns

```
SELECT
    column_name,
    data_type
FROM information_schema.columns
WHERE table_name IN('support', 'survey')
```

...	↑↓	column_n...	...	↑↓	d	...	↑↓
0		id			integer		
1		customer_id			integer		
2		rating			integer		
3		timestamp			date		
4		id			integer		
5		customer_id			integer		
6		creation_date			date		
7		response_time			integer		
8		resolution_time			text		
9		category			text		
10		status			text		

Rows: 11

↗ Expand

Unknown integration DataFrame as i

-- Checking id column for missing, NULL, or inconsistent values

```
SELECT DISTINCT id
FROM support
WHERE id IS NULL;
```

Your query ran successfully but returned no results.

Unknown integration DataFrame as

```
-- Checking customer_id column for missing, NULL, or inconsistent values
```

```
SELECT DISTINCT customer_id
FROM support
ORDER BY customer_id;
```

...	↑↓	cus...	...	↑↓
0		1		
1		4		
2		5		
3		7		
4		8		
5		11		
6		12		
7		14		
8		15		
9		16		
10		18		
11		20		
12		21		
13		22		
14		24		
15		26		

Rows: 1,237

↗ Expand

Unknown integration DataFrame as

```
-- Checking category column for missing, NULL, or inconsistent values
```

```
SELECT DISTINCT category
FROM support;
```

...	↑↓	category	...	↑↓
0		Other		
1		Bug		
2		Feedback		
3		Billing enquiry		
4		Installation Problem		

Rows: 5

↗ Expand

Unknown integration DataFrame as

```
-- Checking status column for missing, NULL, or inconsistent values
```

```
SELECT DISTINCT status
FROM support;
```

...	↑↓	sta...	...	↑↓
0		Open		
1		Resolved		
2		In Progress		
3		-		

Rows: 4

↗ Expand

Unknown integration DataFrame as

```
-- Checking creation_date column for missing, NULL, or inconsistent values
```

```
SELECT DISTINCT creation_date
FROM support
ORDER BY creation_date;
```

...	↑↓	creation_date	...	↑↓
0		2023-01-01T00:00:00.000		
1		2023-01-02T00:00:00.000		
2		2023-01-03T00:00:00.000		
3		2023-01-04T00:00:00.000		
4		2023-01-05T00:00:00.000		
5		2023-01-06T00:00:00.000		
6		2023-01-07T00:00:00.000		
7		2023-01-08T00:00:00.000		
8		2023-01-09T00:00:00.000		
9		2023-01-10T00:00:00.000		
10		2023-01-11T00:00:00.000		
11		2023-01-12T00:00:00.000		
12		2023-01-13T00:00:00.000		
13		2023-01-14T00:00:00.000		
14		2023-01-15T00:00:00.000		
15		2023-01-16T00:00:00.000		

Rows: 334

↗ Expand

Unknown integration DataFrame as

```
-- Checking response_time column for missing, NULL, or inconsistent values
```

```
SELECT DISTINCT response_time
FROM support
ORDER BY response_time;
```

...	↑↓	respon...	...	↑↓
0		1		
1		2		
2		3		
3		4		
4		5		
5		6		
6		7		
7		8		
8		9		
9		10		
10		11		
11		12		
12		13		
13		16		
14		17		

Rows: 15

↗ Expand

 Unknown integration DataFrame as

```
-- Checking resolution_time column for missing, NULL, or inconsistent values
```

```
SELECT DISTINCT resolution_time
FROM support
ORDER BY resolution_time;
```

...	↑↓	resolution...	...	↑↓
0		0.12 hours		
1		0.14 hours		
2		0.15 hours		
3		0.17 hours		
4		0.18 hours		
5		0.19 hours		
6		0.21 hours		
7		0.25 hours		
8		0.29 hours		
9		0.2 hours		
10		0.36 hours		
11		0.41 hours		
12		0.44 hours		
13		0.46 hours		
14		0.47 hours		
15		0.4 hours		

Rows: 246

 Expand

After thoroughly exploring each column of the support table, we observed the following:

1. id: PERFECT! 1987 ids. No NULLS, inconsistencies, or missing values. The PRIMARY KEY of the table.
2. customer_id: 1237 customer ids. No NULLS, inconsistencies, or missing values.
3. category: 5 distinct categories. If there are any NULLS, missing, or inconsistent values, replace them with 'Others'.
4. status: 4 distinct status updates. One inconsistent value found: '-'. Need to replace it with 'Resolved'.
5. creation_date: 334 distinct dates. If there are any NULLS, missing, or inconsistent values, replace them with '2023-01-01'.
6. response_time: 15 distinct days. If there are any NULLS, missing, or inconsistent values, replace them with '0'.
7. resolution_time: 246 distinct hours. If there are any NULLS, missing, or inconsistent values, replace them with '0'.

Make sure to cast the columns carefully.

Unknown integration DataFrame as

```
-- Checking customer_id column for missing, NULL, or inconsistent values
```

```
SELECT DISTINCT customer_id
FROM survey
ORDER BY customer_id;
```

...	↑↓	cus...	...	↑↓
0		11		
1		29		
2		33		
3		34		
4		51		
5		59		
6		68		
7		87		
8		125		
9		139		
10		159		
11		161		
12		167		
13		190		
14		214		
15		215		

Rows: 192

↗ Expand

Unknown integration DataFrame as

```
-- Checking rating column for missing, NULL, or inconsistent values
```

```
SELECT DISTINCT rating
FROM survey
ORDER BY rating;
```

...	↑↓	...	↑↓
0		2	
1		3	
2		4	
3		5	
4		6	
5		7	

Rows: 6

↗ Expand

 Unknown integration DataFrame as

```
-- Checking timestamp column for missing, NULL, or inconsistent values
```

```
SELECT DISTINCT timestamp
FROM survey
ORDER BY timestamp;
```

...	↑↓	timestamp	...	↑↓
0		2023-12-01T00:00:00.000		
1		2023-12-02T00:00:00.000		
2		2023-12-03T00:00:00.000		
3		2023-12-04T00:00:00.000		
4		2023-12-05T00:00:00.000		
5		2023-12-06T00:00:00.000		
6		2023-12-07T00:00:00.000		
7		2023-12-08T00:00:00.000		
8		2023-12-09T00:00:00.000		
9		2023-12-10T00:00:00.000		

Rows: 10

 Expand

After thoroughly exploring each column of the support table, we observed the following:

1. survey_id: PERFECT! 1987 ids. No NULLS, inconsistencies, or missing values. The PRIMARY KEY of the table.
2. customer_id: 192 distinct customer ids.
3. rating: 6 distinct ratings
4. timestamp: 10 distinct timestamps.

Task 1

Before you can start any analysis, you need to confirm that the data is accurate and reflects what you expect to see.

It is known that there are some issues with the `support` table, and the data team have provided the following data description.

Write a query to return data matching this description. You must match all column names and description criteria.

Column Name	Criteria
id	Discrete. The unique identifier of the support ticket. Missing values are not possible due to the database structure.
customer_id	Discrete. The unique identifier of the customer. Missing values should be replaced with 0.
category	Nominal. The category of the support request, can be one of Feedback, Billing Enquiry, Bug, Installation Problem, Other. Missing values should be replaced with Other.
status	Nominal. The current status of the support ticket, one of Open, In Progress or Resolved. Missing values should be replaced with 'Resolved'.
creation_date	Discrete. The date the ticket was created. Can be any date in 2023. Missing values should be replaced with 2023-01-01.
response_time	Discrete. The number of days taken to respond to the support ticket. Missing values should be replaced with 0.
resolution_time	Continuos. The number of hours taken to resolve the support ticket, rounded to 2 decimal places. Missing values should be replaced with 0.

Unknown integration DataFrame as

```

SELECT
    -- Required, never NULL per specs
    id,
    -- Replacing NULL in columns customer_id
    COALESCE(customer_id, 0) AS customer_id, -- NULL → 0

    -- Replacing NULL with 'Other' in category column
    CASE
        WHEN category IS NULL
            THEN 'Other'
        WHEN category NOT IN ('Feedback', 'Billing Enquiry', 'Bug', 'Installation Problem', 'Other')
            THEN 'Other'
        ELSE category
    END AS category, -- NULL/invalid → 'Other'

    -- Replacing NULL with 'Resolved' in status column
    CASE
        WHEN status IS NULL
            THEN 'Resolved'
        WHEN status NOT IN ('Open', 'In Progress', 'Resolved')
            THEN 'Resolved'
        ELSE status
    END AS status, -- NULL/invalid → 'Resolved'

    -- Replacing NULL in creation_date
    COALESCE(creation_date::date, '2023-01-01'::date)
        AS creation_date, -- NULL → 2023-01-01 (explicit cast)

    -- Replacing NULL in response_time
    COALESCE(response_time, 0) AS response_time, -- NULL → 0

    -- Replacing NULL with '0' in resolution_time column and round the values to 2 decimal places
    ROUND(
        COALESCE(
            NULLIF(REGEXP_REPLACE(TRIM(resolution_time), '[^0-9.]', '', 'g'), '')
        )::numeric,
        0
    ), 2
) AS resolution_time -- NULL/non-numeric → 0, rounded to 2 decimals

```

FROM support;

...	↑↓	...	↑↓	cus...	...	↑↓	category	...	↑↓	sta...	...	↑↓	creation_date	...	↑↓	respon...	...	↑↓	resolution...	...
0		1		1062			Installation Problem			In Progress			2023-01-26T00:00:00.000				6			
1		2		892			Other			Open			2023-06-18T00:00:00.000				3			
2		3		433			Feedback			Open			2023-08-17T00:00:00.000				1			
3		6		764			Other			Open			2023-01-16T00:00:00.000				3			
4		7		1144			Other			Open			2023-06-01T00:00:00.000				2			
5		8		288			Feedback			Open			2023-01-22T00:00:00.000				2			
6		9		1495			Bug			In Progress			2023-02-05T00:00:00.000				1			
7		10		1090			Bug			In Progress			2023-05-09T00:00:00.000				3			
8		11		1397			Feedback			In Progress			2023-09-17T00:00:00.000				2			
9		12		54			Feedback			Open			2023-09-13T00:00:00.000				2			
10		14		1207			Bug			Resolved			2023-09-04T00:00:00.000				2			
11		15		1256			Bug			In Progress			2023-11-20T00:00:00.000				3			
12		16		146			Feedback			Open			2023-09-08T00:00:00.000				1			
13		17		455			Installation Problem			In Progress			2023-11-07T00:00:00.000				7			
14		19		950			Bug			In Progress			2023-11-21T00:00:00.000				2			

15	21	215	Feedback	Open	2023-05-29T00:00:00Z	2	
----	----	-----	----------	------	----------------------	---	--

Rows: 1,987

[Expand](#)

Task 2

It is suspected that the response time to tickets is a big factor in unhappiness.

Calculate the minimum and maximum response time for each category of support ticket.

Your output should include the columns `category`, `min_response` and `max_response`.

Values should be rounded to two decimal places where appropriate.

Unknown integration DataFrame as

```
-- Getting the min and max response time and rounding the values to 2 decimal places. Grouping by category to get min and max for each group.
```

```
SELECT
    category,
    ROUND(MIN(response_time), 2) AS min_response,
    ROUND(MAX(response_time), 2) AS max_response
FROM support
GROUP BY category;
```

...	↑↓	category	...	↑↓	min...	↑↓	max...	↑↓
0	Other				1			5		
1	Bug				1			13		
2	Feedback				1			2		
3	Billing enquiry				2			8		
4	Installation Problem				5			17		

Rows: 5

[Expand](#)

Task 3

The support team want to know more about the `rating` provided by customers who reported `Bugs` or `Installation Problem`s.

Write a query to return the `rating` from the survey, the `customer_id`, `category` and `response_time` of the support ticket, for the customers & categories of interest.

Use the original support table, not the output of task 1.

Unknown integration DataFrame as

```
-- Write your query for task 3 in this cell
SELECT
    su.rating,
    s.customer_id,
    s.category,
    s.response_time
FROM support AS s
INNER JOIN survey su
ON s.customer_id = su.customer_id
WHERE s.category = 'Bug'
OR s.category = 'Installation Problem';
```

...	↑↓	...	↑↓	cus...	...	↑↓	category	...	↑↓	respon...	...	↑↓
0		6		621			Installation Problem			7		
1		5		1703			Installation Problem			6		
2		5		766			Installation Problem			7		
3		5		1824			Bug			3		
4		4		931			Installation Problem			9		
5		6		1795			Installation Problem			7		
6		5		1703			Bug			2		
7		2		747			Bug			2		
8		5		1836			Bug			1		
9		7		1882			Installation Problem			6		
10		6		1882			Installation Problem			6		
11		7		1772			Installation Problem			5		
12		3		879			Bug			3		
13		3		902			Bug			2		
14		5		1569			Installation Problem			6		
15		3		1035			Bug			4		

Rows: 123

-expand