

```
1
2
3 --Step 1 Creating database and importing the csv file
4 create database SalesDB;
5
6 use SalesDB;
7
8 select * from sales;
9
10 --step 2 Data cleaning
11
12 --1 checking for how many rows imported
13
14 SELECT COUNT(*) as total_rows FROM sales;
15
16 --2 checking for duplicates
17
18 SELECT transaction_id,COUNT(*) as duplicates_count
19 FROM sales
20 GROUP BY transaction_id
21 HAVING COUNT(transaction_id) >1
22
23 -- delete the duplicate rows
24
25 WITH CTE AS (
26   SELECT *,
27     ROW_NUMBER() OVER (PARTITION BY transaction_id ORDER BY
28       transaction_id) AS Row_Num
29   FROM sales
30 )
31 DELETE FROM CTE
32 WHERE Row_Num=2
33 SELECT * FROM CTE
34 WHERE transaction_id IN
35   ('TXN240646', 'TXN342128', 'TXN855235', 'TXN981773')
36
37 --3 change the column names if needed or if any spel mistakes
38
39 EXEC sp_rename'sales.quantiy', 'quantity', 'COLUMN'
40
41 EXEC sp_rename'sales.prce', 'price', 'COLUMN'
42
43 select * from sales;
44
45 --4 check for datatypes
46 SELECT
47   COLUMN_NAME,
48   DATA_TYPE,
49   CHARACTER_MAXIMUM_LENGTH
50   FROM INFORMATION_SCHEMA.COLUMNS
51 WHERE TABLE_NAME = 'sales';
52
53 --5 Check for nullvalues
```

```
52
53 SELECT
54     SUM(CASE WHEN transaction_id IS NULL THEN 1 ELSE 0 END) AS      ↵
55         null_transaction_id,
56     SUM(CASE WHEN customer_id IS NULL THEN 1 ELSE 0 END) AS      ↵
57         null_customer_id,
58     SUM(CASE WHEN customer_name IS NULL THEN 1 ELSE 0 END) AS      ↵
59         null_customer_name,
60     SUM(CASE WHEN customer_age IS NULL THEN 1 ELSE 0 END) AS      ↵
61         null_customer_age,
62     SUM(CASE WHEN gender IS NULL THEN 1 ELSE 0 END) AS null_gender,    ↵
63     SUM(CASE WHEN product_id IS NULL THEN 1 ELSE 0 END) AS      ↵
64         null_product_id,
65     SUM(CASE WHEN product_name IS NULL THEN 1 ELSE 0 END) AS      ↵
66         null_product_name,
67     SUM(CASE WHEN product_category IS NULL THEN 1 ELSE 0 END) AS      ↵
68         null_product_category,
69     SUM(CASE WHEN quantity IS NULL THEN 1 ELSE 0 END) AS null_quantity,   ↵
70     SUM(CASE WHEN price IS NULL THEN 1 ELSE 0 END) AS null_price,       ↵
71     SUM(CASE WHEN payment_mode IS NULL THEN 1 ELSE 0 END) AS      ↵
72         null_payment_mode,
73     SUM(CASE WHEN purchase_date IS NULL THEN 1 ELSE 0 END) AS      ↵
74         null_purchase_date,
75     SUM(CASE WHEN time_of_purchase IS NULL THEN 1 ELSE 0 END) AS      ↵
76         null_time_of_purchase,
77     SUM(CASE WHEN status IS NULL THEN 1 ELSE 0 END) AS null_status
78 FROM sales;
79
80 -- Handling null values
81
82 SELECT *
83 FROM sales
84 WHERE transaction_id IS NULL
85 OR
86 customer_id IS NULL
87 OR
88 customer_name IS NULL
89 OR
90 customer_age IS NULL
91 OR
92 gender IS NULL
93 OR
94 product_id IS NULL
95 OR
96 product_name IS NULL
97 OR
98 product_category IS NULL
99 OR
100 quantity IS NULL
101 or
102 payment_mode is null
103 or
104 purchase_date is null
```

```
95 or
96 status is null
97 or
98 price is null
99
100 DELETE FROM sales
101 WHERE transaction_id IS NULL
102
103
104 SELECT * FROM sales
105 Where Customer_name='Ehsaan Ram'
106
107 UPDATE sales
108 SET customer_id='CUST9494'
109 WHERE transaction_id='TXN977900'
110
111 SELECT * FROM sales
112 Where Customer_name='Damini Raju'
113
114 UPDATE sales
115 SET customer_id='CUST1401'
116 WHERE transaction_id='TXN985663'
117
118 SELECT * FROM sales
119 Where Customer_id='CUST1003'
120
121 UPDATE sales
122 SET customer_name='Mahika Saini',customer_age=35,gender='Male'
123 WHERE transaction_id='TXN432798'
124
125 select * from sales;
126
127 --step 6 Formating handling
128
129 select distinct gender from sales; --F, Male, Female, M
130
131 UPDATE sales
132 SET gender='M'
133 WHERE gender='Male'
134
135 UPDATE sales
136 SET gender='F'
137 WHERE gender='Female'
138
139 SELECT DISTINCT payment_mode
140 FROM sales
141
142 UPDATE sales
143 SET payment_mode='Credit Card'
144 WHERE payment_mode='CC'
145
146 ----- →
```

```
147 --Data Analysis
148 -- solving business problems
149
150 --1 total num of transctions
151
152 SELECT COUNT(*) AS total_transactions
153 FROM sales;
154
155 -- count unique products
156
157 SELECT COUNT(DISTINCT product_id) AS unique_products
158 FROM sales;
159
160 -- total quantity sold
161 SELECT SUM(quantity) AS total_quantity_sold
162 FROM sales;
163
164 -- avaerage customer age
165
166 SELECT AVG(customer_age) AS avg_age
167 FROM sales;
168
169 -- customer count b gender
170 SELECT gender, COUNT(*) AS gender_count
171 FROM sales
172 GROUP BY gender;
173
174 -- orders by payment mode
175
176 SELECT payment_mode, COUNT(*) AS order_count
177 FROM sales
178 GROUP BY payment_mode;
179
180 -- most frequently ordered products
181 SELECT product_name, COUNT(*) AS order_count
182 FROM sales
183 GROUP BY product_name
184 ORDER BY order_count DESC;
185
186 -- total quantity sold per product
187
188 SELECT product_name, SUM(quantity) AS total_quantity
189 FROM sales
190 GROUP BY product_name
191 ORDER BY total_quantity DESC;
192
193 -- Daily transctions
194
195 SELECT purchase_date, COUNT(*) AS total_orders
196 FROM sales
197 GROUP BY purchase_date
198 ORDER BY purchase_date;
199
```

```
200 --top 10 questions for analysis
201
202 -- 1. What are the top 5 most selling products by quantity?
203
204 SELECT TOP 5 product_name, SUM(quantity) AS total_quantity_sold
205 FROM sales
206 WHERE status='delivered'
207 GROUP BY product_name
208 ORDER BY total_quantity_sold DESC
209
210 -- Business Impact / Solution:
211 -- Helps identify top 5 high-demand products.
212 -- Supports better inventory and stock management.
213 -- Avoids stockouts and improves customer satisfaction.
214 -- Helps focus promotions on best-selling items.
215 -- Improves sales forecasting and demand planning.
216
217 -- 2. Which products are most frequently cancelled?
218
219 SELECT TOP 5 product_name, COUNT(*) AS total_cancelled
220 FROM sales
221 WHERE status='cancelled'
222 GROUP BY product_name
223 ORDER BY total_cancelled DESC
224
225 --Business Problem: Frequent cancellations affect revenue and customer trust.
226
227 -- Business Impact / Solution:
228 -- Helps identify products with high cancellation rates.
229 -- Business can check if the issue is quality, delivery, or pricing.
230 -- Can improve or replace the problematic products.
231 -- Helps reduce future cancellations and improve customer experience.
232
233
234 -- 3. What time of the day has the highest number of purchases?
235
236 select * from sales
237
238     SELECT
239         CASE
240             WHEN DATEPART(HOUR,time_of_purchase) BETWEEN 0 AND 5 THEN 'NIGHT'
241             WHEN DATEPART(HOUR,time_of_purchase) BETWEEN 6 AND 11 THEN 'MORNING'
242             WHEN DATEPART(HOUR,time_of_purchase) BETWEEN 12 AND 17 THEN 'AFTERNOON'
243             WHEN DATEPART(HOUR,time_of_purchase) BETWEEN 18 AND 23 THEN 'EVENING'
244         END AS time_of_day,
245         COUNT(*) AS total_orders
246     FROM sales
247     GROUP BY
```

```
248     CASE
249         WHEN DATEPART(HOUR,time_of_purchase) BETWEEN 0 AND 5 THEN ↵
250             'NIGHT'
251         WHEN DATEPART(HOUR,time_of_purchase) BETWEEN 6 AND 11 THEN ↵
252             'MORNING'
253         WHEN DATEPART(HOUR,time_of_purchase) BETWEEN 12 AND 17 THEN ↵
254             'AFTERNOON'
255         WHEN DATEPART(HOUR,time_of_purchase) BETWEEN 18 AND 23 THEN ↵
256             'EVENING'
257     END
258 ORDER BY total_orders DESC
259 -----
260 -----
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
```

248 CASE  
249 WHEN DATEPART(HOUR,time\_of\_purchase) BETWEEN 0 AND 5 THEN ↵  
250 'NIGHT'  
251 WHEN DATEPART(HOUR,time\_of\_purchase) BETWEEN 6 AND 11 THEN ↵  
252 'MORNING'  
253 WHEN DATEPART(HOUR,time\_of\_purchase) BETWEEN 12 AND 17 THEN ↵  
254 'AFTERNOON'  
255 WHEN DATEPART(HOUR,time\_of\_purchase) BETWEEN 18 AND 23 THEN ↵  
256 'EVENING'  
257 END  
258 ORDER BY total\_orders DESC  
259 -----  
260 -----  
261  
262  
263 --Business Problem : Find peak sales times.  
264  
265 -- Business Solution: Knowing the peak purchase time helps the business ↵  
plan better by increasing staff during busy hours, preparing enough ↵  
inventory, and running targeted promotions when customers are most ↵  
active. This improves customer service, reduces wait times, avoids ↵  
stockouts, and can increase overall sales and operational efficiency.  
266  
267 --👤 4. Who are the top 5 highest spending customers?  
268  
269 SELECT \* FROM sales  
270  
271 SELECT TOP 5  
272 customer\_name,  
273 FORMAT(SUM(CAST(price AS DECIMAL(18,2)) \* CAST(quantity AS DECIMAL(18,2))),  
274 'C0', 'en-IN') AS total\_spend  
275 FROM sales  
276 GROUP BY customer\_name  
277 ORDER BY SUM(CAST(price AS DECIMAL(18,2)) \* CAST(quantity AS DECIMAL(18,2))) DESC;  
278  
279  
280 --Business Problem Solved: Identify VIP customers.  
281  
282 --Business Impact: Personalized offers, loyalty rewards, and retention.  
283 -- Identifying the top spending customers helps the business ↵  
understand who its VIP customers are. This allows the company to ↵  
give them special offers, loyalty rewards, and personalized services ↵  
to keep them engaged and retain them for the long term, which ↵  
increases repeat sales and overall revenue.  
284  
285 --⌚ 5. Which product categories generate the highest revenue?

```
286
287 SELECT * FROM sales
288
289 SELECT
290     product_category,
291     FORMAT(SUM(CAST(price AS DECIMAL(18,2)) * CAST(quantity AS DECIMAL(18,2))), 'C0', 'en-IN') AS Revenue
292 FROM sales
293 GROUP BY product_category
294 ORDER BY SUM(CAST(price AS DECIMAL(18,2)) * CAST(quantity AS DECIMAL(18,2))) DESC;
295
296
297
298 --Business Problem Solved: Identify top-performing product categories.
299
300 --Business Impact: Refine product strategy, supply chain, and promotions.
301 --allowing the business to invest more in high-margin or high-demand categories.
302
303 -- 6. What is the return/cancellation rate per product category?
304
305 SELECT * FROM sales
306 --cancellation
307 SELECT product_category,
308     FORMAT(COUNT(CASE WHEN status='cancelled' THEN 1 END)*100.0/COUNT(*), 'N3')+' %' AS cancelled_percent
309 FROM sales
310 GROUP BY product_category
311 ORDER BY cancelled_percent DESC
312
313 --Return
314 SELECT product_category,
315     FORMAT(COUNT(CASE WHEN status='returned' THEN 1 END)*100.0/COUNT(*), 'N3')+' %' AS returned_percent
316 FROM sales
317 GROUP BY product_category
318 ORDER BY returned_percent DESC
319
320 --Business Problem Solved: Monitor dissatisfaction trends per category.
321
322
323 --Business Impact: Reduce returns, improve product descriptions/expectations.
324 --Helps identify and fix product or logistics issues.
325
326
327 -- 7. What is the most preferred payment mode?
328
329 SELECT * FROM sales
330
331 SELECT payment_mode, COUNT(payment_mode) AS total_count
```

```
332 FROM sales
333 GROUP BY payment_mode
334 ORDER BY total_count desc
335
336
337 --Business Problem Solved: Know which payment options customers prefer.
338
339 --Business Impact: Streamline payment processing, prioritize popular ↵
  modes.
340
341 --⌚ 8. How does age group affect purchasing behavior?
342
343 SELECT * FROM sales
344 --SELECT MIN(customer_age) ,MAX(customer_age)
345 --from sales
346
347 SELECT
348     CASE
349         WHEN customer_age BETWEEN 18 AND 25 THEN '18-25'
350         WHEN customer_age BETWEEN 26 AND 35 THEN '26-35'
351         WHEN customer_age BETWEEN 36 AND 50 THEN '36-50'
352         ELSE '51+'
353     END AS age_group,
354     FORMAT(
355         SUM(CAST(price AS DECIMAL(18,2)) * CAST(quantity AS DECIMAL
356             (18,2))),
357         'C0','en-IN'
358     ) AS total_purchase
359 FROM sales
360 GROUP BY
361     CASE
362         WHEN customer_age BETWEEN 18 AND 25 THEN '18-25'
363         WHEN customer_age BETWEEN 26 AND 35 THEN '26-35'
364         WHEN customer_age BETWEEN 36 AND 50 THEN '36-50'
365         ELSE '51+'
366     END
367 ORDER BY
368     SUM(CAST(price AS DECIMAL(18,2)) * CAST(quantity AS DECIMAL(18,
369             2))) DESC;
370
371
372 --Business Problem Solved: Understand customer demographics.
373
374 --Business Impact: Targeted marketing and product recommendations by ↵
  age group.
375
376
377 --⌚ 10. Are certain genders buying more specific product categories?
378
379 --Method 1
380 SELECT gender,product_category,COUNT(product_category) AS ↵
```

```
    total_purchase
381 FROM sales
382 GROUP BY gender,product_category
383 ORDER BY gender
384
385 --Method 2
386 SELECT *
387 FROM (
388     SELECT gender,product_category
389     FROM sales
390 ) AS source_table
391 PIVOT (
392     COUNT(gender)
393     FOR gender IN ([Male],[Female])
394 ) AS pivot_table
395 ORDER BY product_category
396
397 --Business Problem Solved: Gender-based product preferences.
398
399 --Business Impact: Personalized ads, gender-focused campaigns.
400
```