

AI ASSISTED CODING LAB TEST – 04

MOHAMMED ANAS SIDDIQUI

2403A52088

Batch – 04

SET-05)

Q1)

a)

Task: Design schema for Hotel reservation system.

Prompt: Design a schema for a Hotel Reservation system including room, customer, reservation and payment details

Code:

```
1  CREATE TABLE Room (
2    room_id INT PRIMARY KEY,
3    room_number VARCHAR(10),
4    room_type VARCHAR(20)
5  );
6
7  CREATE TABLE Guest (
8    guest_id INT PRIMARY KEY,
9    name VARCHAR(50),
10   email VARCHAR(50)
11 );
12
13 CREATE TABLE Reservation (
14   reservation_id INT PRIMARY KEY,
15   guest_id INT,
16   room_id INT,
17   check_in DATE,
18   check_out DATE,
19   FOREIGN KEY (guest_id) REFERENCES Guest(guest_id),
20   FOREIGN KEY (room_id) REFERENCES Room(room_id)
21 );
22
23 CREATE TABLE Payment (
24   payment_id INT PRIMARY KEY,
25   reservation_id INT,
26   amount DECIMAL(10, 2),
27   payment_date DATE,
28   payment_method VARCHAR(20),
29   FOREIGN KEY (reservation_id) REFERENCES Reservation(reservation_id)
30 );
```

Output:

```
Query OK, 0 rows affected
```

Observation: When tasked AI to generate a schema for the task, it quickly generated queries for SQL

b)

Task: Write SQL to find rooms that remain vacant for 10+ days

Prompt: Write SQL to find rooms that remain vacant for more than 10 days

Code:

```
SELECT r.room_id, r.room_number
FROM Room r
LEFT JOIN Reservation res ON r.room_id = res.room_id
    AND res.check_out >= CURRENT_DATE - INTERVAL '10 days'
WHERE res.reservation_id IS NULL;
```

Output:

room_id	room_number
-----+-----	

Observation: After inserting the details of the database reservation system and asked AI to generate query to find the rooms that were vacant for 10+ days, it quickly gave the query and gave the output.

Q2)

a) **Task:** Use AI to clean inconsistent date fields

Prompt: Generate a code to clean inconsistent date fields

Code:

```
import pandas as pd
import dateparser

def clean_dates(date_list):
    return [dateparser.parse(str(dt)).strftime('%Y-%m-%d') if
dateparser.parse(str(dt)) else None for dt in date_list]

dates = ["2025-12-01", "01/12/2025", "Dec 1, 2025", "2025/12/01"]
cleaned_dates = clean_dates(dates)
print(cleaned_dates)
```

Output:

```
['2025-12-01', '2025-12-01', '2025-12-01', '2025-12-01']
```

Observation: When tasked AI to clean the inconsistent date fields, it quickly generated a code to clean the date fields, and match the above format.

b)

Task: Format all dates into ISO-8601.

Prompt: Format all dates into ISO-8601

Code:

```
import pandas as pd

df = pd.DataFrame({'date': ["2025-12-01", "01/12/2025", "Dec 1, 2025",
"2025/12/01"]})
df['iso_date'] = pd.to_datetime(df['date'], dayfirst=True,
errors='coerce').dt.strftime('%Y-%m-%d')
print(df)
```

Output:

	date	iso_date
0	2025-12-01	2025-12-01
1	01/12/2025	2025-12-01
2	Dec 1, 2025	2025-12-01
3	2025/12/01	2025-12-01

Observation: When tasked AI to format the given dates into ISO-8601, it quickly generated a code to format the dates