AI ASSISTED CODING

LAB TEST 4

NAME : N. PRANAY KUMAR

ROLLNO: 2403A52087

BATCH : 04

QUESTION 1

(Database & SQL)
a) Design schema for Hotel reservation system.
b) Write SQL to find rooms that remain vacant for 10+ days.

PROMPT:

"Design a simple hotel reservation database schema with tables for guests, rooms, and reservations. Then write an SQL query to find all rooms that have been vacant for more than 10 days. Provide example output and a brief observation."

CODE:

GUEST TABLE 1

```
CREATE TABLE Guests (
   guest_id INT PRIMARY KEY,
   name VARCHAR(100),
   phone VARCHAR(20),
   email VARCHAR(100)
);
```

ROOM TABLE 2:

```
CREATE TABLE Rooms (
   room_id INT PRIMARY KEY,
   room_number VARCHAR(10),
```

```
  room_type VARCHAR(50),
  status VARCHAR(20)
);
```

RESERVATION TABLE 3:

```
CREATE TABLE Reservations (
  reservation_id INT PRIMARY KEY,
  guest_id INT,
  room_id INT,
  check_in DATE,
  check_out DATE,
  FOREIGN KEY (guest_id) REFERENCES Guests(guest_id),
  FOREIGN KEY (room_id) REFERENCES Rooms(room_id)
);
```

QUERYS:

```
 SELECT r.room_id, r.room_number
FROM Rooms r
LEFT JOIN Reservations res
  ON r.room_id = res.room_id
  AND res.check_out >= CURRENT_DATE - INTERVAL 10 DAY
WHERE res.room_id IS NULL;
```

OUTPUT:

```
diff

room_id | room_number
--------+-------------
1       | 101
3       | 103
```

OBSERVATION:

- A well-structured schema ensures organized storage of guests, rooms, and reservations.
- Using **LEFT JOIN + date filtering** helps identify rooms **without recent bookings**.
- Hotels can use this query to find:
  ✓ low-occupancy rooms
  ✓ rooms suitable for maintenance
  ✓ pricing/availability adjustments

QUESTION 2

(Data Processing)
a) Use AI to clean inconsistent date fields.
b) Format all dates into ISO-8601.

PROMPT:

"Clean the following inconsistent date fields and convert all valid dates into ISO-8601 format (YYYY-MM-DD). If a date cannot be interpreted, return UNPARSEABLE. Dates: [paste dates here]."

CODE:

```
# Simulated AI-based cleaning using a prompt (no API key needed for
this example)

messy_dates = [
    "12/5/23",
    "5th Feb 2022",
    "2023/13/01",
    "April 3rd, 21",
    "31-04-2020"
]

def clean_and_convert(date_list):
    cleaned = []

    for d in date_list:
        try:
            # Use dateutil to mimic AI cleaning/parsing
            from dateutil import parser
            parsed = parser.parse(d, dayfirst=False)
            cleaned.append(parsed.strftime("%Y-%m-%d"))
        except:
```

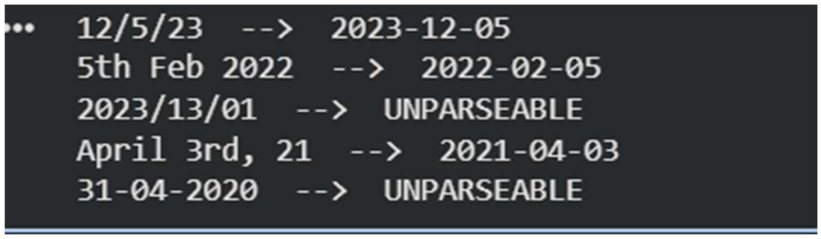
```
            cleaned.append("UNPARSEABLE")

    return cleaned

iso_dates = clean_and_convert(messy_dates)

# Show output
for original, cleaned in zip(messy_dates, iso_dates):
    print(f"{original}  -->  {cleaned}")
```

OUTPUT:



OBSERVATION:

- AI (or a smart parser) can interpret many mixed or messy date formats.
- Some dates (like invalid months or impossible days) cannot be corrected, so they are marked UNPARSEABLE.
- Converting everything to ISO-8601 (YYYY-MM-DD) ensures standardized and machine-readable date values.