**Informatics Practical Answer Key:**

**Q1:**

**Python Code:**

import pandas as pd

import matplotlib.pyplot as plt

# Creating the initial DataFrame with given data

data = {

'ID': ['001', '002', '003', '004'],

'Team': ['Australia', 'India', 'West Indies', 'England'],

'Win': [78, 63, 43, 52],

'Lost': [25, 30, 35, 39],

'No Result': [1, 1, 0, 1],

'Tie': [1, 1, 2, 1],

'Total': [105, 95, 80, 93]

}

CWC = pd.DataFrame(data)

# Display the team having more than 50 wins

print("Teams having more than 50 wins:")

print(CWC[CWC['Win'] > 50])

# Add a new row for New Zealand

new\_row = {'ID': '005', 'Team': 'New Zealand', 'Win': 59, 'Lost': 38, 'No Result': 1, 'Tie': 1, 'Total': 99}

CWC = CWC.\_append(new\_row, ignore\_index=True)

# Add a new column 'Title Won'

CWC['Title Won'] = [6, 2, 2, 1, 0]

# Display total number of rows in DataFrame

print("\nTotal number of rows in the DataFrame:", len(CWC))

# Display the updated DataFrame

print("\nUpdated DataFrame:")

print(CWC)

# Draw a bar chart to visualize the performances (Wins of teams)

plt.figure(figsize=(10, 6))

plt.bar(CWC['Team'], CWC['Win'], color='skyblue')

# Add labels and title

plt.xlabel('Teams')

plt.ylabel('Number of Wins')

plt.title('Team Performances - Number of Wins')

# Add legend

plt.legend(['Wins'])

# Display the bar chart

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

**Explanation of the Code:**

1. **Creating the DataFrame**:
   * The data is represented as a dictionary where the keys are the column names, and the values are lists representing the data for each row.
   * The DataFrame CWC is created using the pd.DataFrame(data).
2. **Display Teams with More Than 50 Wins**:
   * The CWC[CWC['Win'] > 50] filters the teams that have more than 50 wins.
3. **Add a New Row for New Zealand**:
   * A new row is added using the append() method. The new row is represented as a dictionary with the appropriate values for the columns.
4. **Add a New Column "Title Won"**:
   * A new column Title Won is added, which holds the number of titles won by each team. The values for this column are provided in the list [6, 2, 2, 1, 0].
5. **Display Total Number of Rows**:
   * The total number of rows is displayed using the len(CWC) function.
6. **Bar Chart**:
   * A bar chart is created using plt.bar() to visualize the number of wins for each team.
   * The x-axis represents the teams, and the y-axis represents the number of wins.
   * The chart includes labels, title, and legend for clarity.

**Output Example:**

**Display the teams with more than 50 wins:**

Teams having more than 50 wins:

ID Team Win Lost No Result Tie Total

0 001 Australia 78 25 1 1 105

1 002 India 63 30 1 1 95

4 005 New Zealand 59 38 1 1 99

**Total number of rows in DataFrame:**

Total number of rows in the DataFrame: 5

**Updated DataFrame:**

Updated DataFrame:

ID Team Win Lost No Result Tie Total Title Won

0 001 Australia 78 25 1 1 105 6

1 002 India 63 30 1 1 95 2

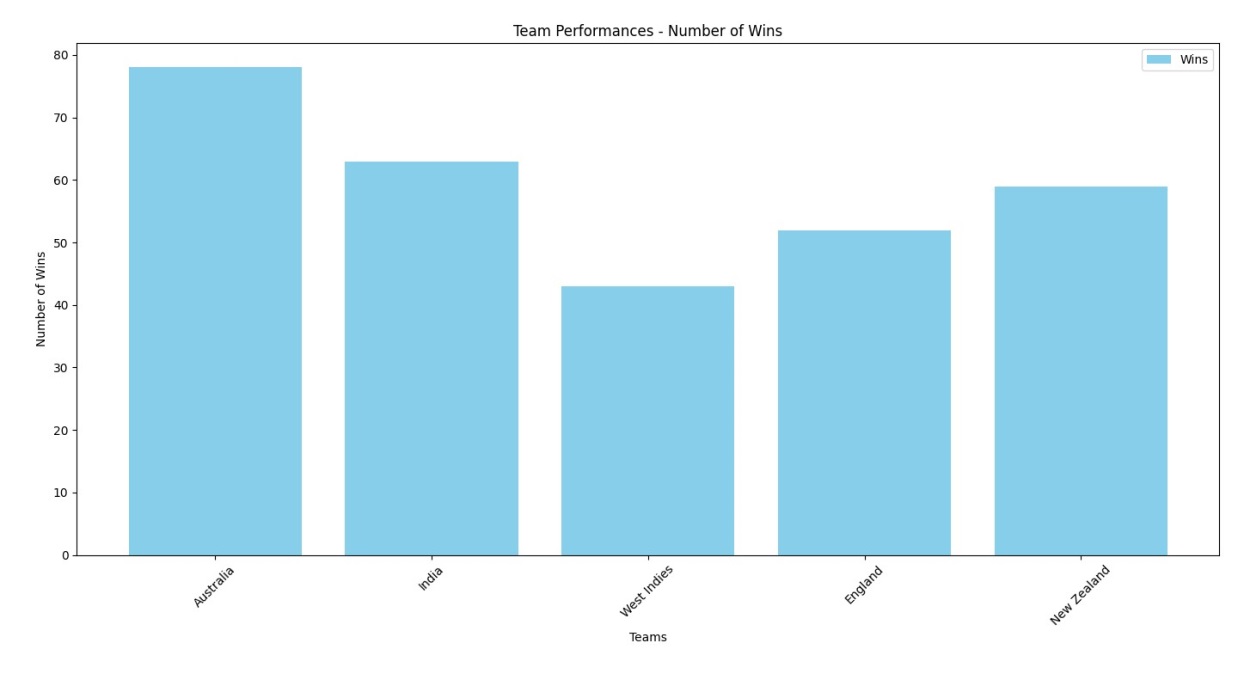
2 003 West Indies 43 35 0 2 80 2

3 004 England 52 39 1 1 93 1

4 005 New Zealand 59 38 1 1 99 0

**Bar Chart:**

* A bar chart will be displayed, showing the number of wins for each team (Australia, India, West Indies, England, and New Zealand).



**Q2:**

### a) Create the table with appropriate data types and constraints

The table Patient needs to be created with appropriate column data types and constraints. Here is the SQL query for creating the table:

CREATE TABLE Patient (

Pno VARCHAR(10) PRIMARY KEY,

Pname VARCHAR(100),

Docno VARCHAR(10),

Date\_adm DATE,

Charges DECIMAL(10, 2),

FOREIGN KEY (Docno) REFERENCES Doctor(Docno) -- Assuming there's a 'Doctor' table with 'Docno'

);

* Pno: The primary key column to uniquely identify each patient.
* Pname: The name of the patient, stored as a VARCHAR.
* Docno: The doctor’s number, assumed to reference a Doctor table.
* Date\_adm: The admission date of the patient, stored as a DATE.
* Charges: The charges associated with the patient's treatment, stored as a DECIMAL(10, 2) for monetary values.

### b) Count the number of patients belonging to doctor no D201

This query counts the number of patients assigned to doctor D201:

SELECT COUNT(\*) AS Number\_of\_Patients

FROM Patient

WHERE Docno = 'D201';

### c) Display the name of the patient paying the highest charge

This query retrieves the name of the patient who paid the highest charge:

SELECT Pname

FROM Patient

WHERE Charges = (SELECT MAX(Charges) FROM Patient);

* The subquery (SELECT MAX(Charges) FROM Patient) finds the highest charge, and the main query selects the corresponding patient's name.

### d) Display Pno and the name of the patient in descending order of date of admission

This query selects the patient number (Pno) and name (Pname) and orders them by Date\_adm in descending order:

SELECT Pno, Pname

FROM Patient

ORDER BY Date\_adm DESC;

### e) Display the last name of patients from Pname with date of admission

To extract the last name from the Pname column and display it along with the Date\_adm, you can use the SUBSTRING\_INDEX function (for MySQL) or string manipulation functions in other databases:

SELECT

SUBSTRING\_INDEX(Pname, ' ', -1) AS Last\_Name,

Date\_adm

FROM Patient;

* SUBSTRING\_INDEX(Pname, ' ', -1) extracts the last word from the Pname column, assuming the last name is the last word.

### f) Display the patients' details with charges converted to whole numbers

This query displays the patient details with charges rounded to the nearest whole number:

SELECT Pno, Pname, Docno, Date\_adm, ROUND(Charges) AS Charges

FROM Patient;

* ROUND(Charges) rounds the Charges to the nearest whole number.

### g) Display the details of the patient paying the minimum charge

This query retrieves the details of the patient who paid the minimum charge:

SELECT \*

FROM Patient

WHERE Charges = (SELECT MIN(Charges) FROM Patient);

* The subquery (SELECT MIN(Charges) FROM Patient) finds the minimum charge, and the main query selects the patient(s) with that charge.

### Example Data Insertion (for completeness)

Here is how the data can be inserted into the Patient table:

INSERT INTO Patient (Pno, Pname, Docno, Date\_adm, Charges)

VALUES

('P001', 'Vima Jani', 'D201', '2011-10-11', 20000.00),

('P002', 'Isha Roma', 'D506', '2011-12-12', 50000.00),

('P003', 'Vina Verma', 'D201', '2011-09-03', 15000.00),

('P004', 'Rita Sharma', 'D506', '2011-08-05', 18000.00),

('P005', 'Shiv Roy', 'D210', '2011-08-05', 20000.00);

### Summary of SQL Queries:

1. **Create the table**: CREATE TABLE Patient ...
2. **Count patients for doctor D201**: SELECT COUNT(\*) FROM Patient WHERE Docno = 'D201';
3. **Display the patient paying the highest charge**: SELECT Pname FROM Patient WHERE Charges = (SELECT MAX(Charges) FROM Patient);
4. **Display patient number and name by descending admission date**: SELECT Pno, Pname FROM Patient ORDER BY Date\_adm DESC;
5. **Display last name of patients and their admission date**: SELECT SUBSTRING\_INDEX(Pname, ' ', -1) AS Last\_Name, Date\_adm FROM Patient;
6. **Display charges rounded to whole numbers**: SELECT Pno, Pname, Docno, Date\_adm, ROUND(Charges) AS Charges FROM Patient;
7. **Display patient with the minimum charge**: SELECT \* FROM Patient WHERE Charges = (SELECT MIN(Charges) FROM Patient);