TOUCHPAD
Computer Textbook Series

12

Informatics Practices

with

PYTHON

Beta*
*Updated Copy coming soon

ORANGE

Class XII

Answer Key

Informatics Practices

1. Data Handling Using Pandas

Assessment

A.

1. b	2. a	3. d	4. a	5. d

B.

1. False	2. True	3. True	4. False	5. False

C.

1. tai 1	2. in	3. loc	4. iloc	5. pd.Series

D. 1. Differences Between Attributes in Pandas Series:

- head and tail:
- head(): Returns the first n elements of the Series (default is 5).
  - tail(): Returns the last n elements of the Series (default is 5).
- values and index:
- values: Returns the data of the Series as a NumPy array.
  - index: Returns the index (labels) of the Series.

2. 1200 (300 + 400 + 500)

3.
```
I1    81.0
I2    NaN
I3    NaN
I4    63.0
I5    54.0
dtype: float64
```

4. 30

5.
```
# (i) Find the highest and lowest marks.
highest_marks = marks.max()
lowest_marks = marks.min()
```


IMPRINT 1

```
# (ii) Determine the subject name(s) with the highest marks obtained
in the previous part.
highest_subjects = subjects[marks == highest_marks].unique()

# (iii) Determine the average marks scored in Math.
math_average = marks[subjects == 'Math'].mean()

# (iv) Find the number of subjects in which the student scored above
90.
subjects_above_90 = (marks > 90).sum()

# (v) Find the subjects in which the student scored below 90.
subjects_below_90 = subjects[marks < 90].unique()

# (vi) Find the average marks scored by student with roll number
101.
average_marks_101 = marks[marks.index == 101].mean()
```

Case-based Questions

```
import pandas as pd

# Create the Series with scores
artScores = pd.Series([85, 92, 78, 89, 95, 87, 93, 91, 84, 88])

# Sort the scores in descending order
sortedScores = artScores.sort_values(ascending=False)

# Extract the top 5 scores
top5Scores = sortedScores.iloc[:5]

print(top5Scores)
```

2. Data Handling Using Pandas Dataframe

Assessment

A. 1. b 2. c 3. c 4. a 5. a
6. b 7. b

B. 1. True 2. True 3. False 4. False 5. True
6. False 7. True 8. True

C. 1. describe 2. unique 3. 1 4. rename 5. include
6. 1 7. set_index 8. drop 9. groupby 10. to_csv

D. 1.
```
import pandas as pd

student1 = pd.Series(['Seema', 10, 'A', 99])
student2 = pd.Series(['Supriya', 11, 'B', 82])
student3 = pd.Series(['Mehak', 12, 'A', 95])
student4 = pd.Series(['Madhu', 13, 'B', 80])

df = pd.DataFrame([student1, student2, student3, student4],
columns=['Name', 'RollNumber', 'Grade', 'Marks'])
print(df)
```

2.
```
data = {
"Gender": ["Male", "Male", "Male", "Male", "Female", "Female",
"Female", "Female", "Female", "Female", "Female", "Male", "Male",
"Female", "Female", "Female", "Female", "Female", "Male", "Male",
"Male", "Female", "Female", "Female", "Male", "Male", "Male",
"Male", "Male", "Male", "Male", "Male", "Male", "Female", "Female",
"Female", "Female", "Female", "Female", "Male", "Male", "Male",
"Male", "Female", "Female", "Female", "Female"],
"Age group": ["25-29", "30-34", "55-59", "80-84", "0-4", "30-34",
"45-49", "60-64", "100+", "15-19", "95-99", "75-79", "95-99", "10-
14", "35-39", "50-54", "55-59", "65-69", "50-54", "70-74", "85-89",
"40-44", "75-79", "90-94", "5-9", "20-24", "45-49", "60-64", "0-4",
"10-14", "15-19", "35-39", "40-44", "65-69", "90-94", "100+", "5-9",
"20-24", "25-29", "70-74", "80-84", "85-89"],
```


IMPRINT 1

```
"Population": [63129, 59247, 30768, 3944, 55340, 54832, 39938, 25038,
23, 61074, 176, 7166, 119, 59621, 50436, 34835, 29953, 19773, 36249,
12639, 1718, 44528, 8407, 812, 63226, 66257, 42078, 25049, 59967,
65685, 67336, 54325, 47639, 19370, 568, 15, 57728, 59916, 57608,
13496, 5115, 2401]
}

df = pd.DataFrame(data)
print(f"Number of rows: {df.shape[0]}")
print(f"Number of columns: {df.shape[1]}")
print(f"Column labels: {df.columns.tolist()}")
```

3.
```
data = {
   'Name': ['Hetansh', 'Supriya', 'Mehak', 'Madhu', 'Rama'],
   'RollNumber': [10, 11, 12, 13, 14],
   'Grade': ['A', 'B', 'A', 'B', 'C'],
   'Marks': [99, 82, 95, 80, 60]
}

studentDF = pd.DataFrame(data)
print(studentDF)
```

4.
```
import pandas as pd

# Given DataFrame
df = pd.DataFrame({
     'Name': ['Hetansh', 'Supriya', 'Mehak', 'Madhu', 'Rama'],
     'RollNumber': [10, 11, 12, 13, 14],
     'Grade': ['A', 'B', 'A', 'B', 'C'],
     'Marks': [99, 82, 95, 80, 60]
})
```

Output of Each Statement:

(i) Output: `2`

Explanation: df.ndim returns the number of dimensions of the DataFrame, which is 2 (rows and columns).

(ii) Output: `(5, 4)`

Explanation: df.shape returns a tuple representing the dimensionality of the DataFrame, which is 5 rows and 4 columns.

(iii) Output: `RangeIndex(start=0, stop=5, step=1)`

Explanation: df.index returns the index (row labels) of the DataFrame, which is a range index from 0 to 4.

(iv) Output: `Index(['Name', 'RollNumber', 'Grade', 'Marks'], dtype='object')`

Explanation: df.columns returns the column labels of the DataFrame.

(v) Output:

	Name	RollNumber	Grade	Marks
0	Hetansh	10	A	99
1	Supriya	11	B	82
2	Mehak	12	A	95
3	Madhu	13	B	80
4	Rama	14	C	60

Explanation: df.head(5) returns the first 5 rows of the DataFrame.

(vi) Output:

	Name	RollNumber	Grade	Marks
3	Madhu	13	B	80
4	Rama	14	C	60

Explanation: df.tail(2) returns the last 2 rows of the DataFrame.

(vii) Output:

```
Name          Mehak
RollNumber       12
Grade             A
Marks            95
Name: 2, dtype: object
```

Explanation: df.iloc[2] returns the 3rd row of the DataFrame as a Series (indexing starts from 0).

(viii) Output:

Explanation: df.loc[1, 'Name'] returns the value in the 'Name' column for the row with index 1.

(ix) Output:

```
Name       Grade     Marks   RollNumber
10       Hetansh         A           99
11       Supriya         B           82
12         Mehak         A           95
13         Madhu         B           80
14          Rama         C           60
```

Explanation: df.set_index('RollNumber') returns a new DataFrame with 'RollNumber' set as the index.

(x) Output:

```
        Name    RollNumber    Grade    Marks
0    Hetansh            10        A       99
```

Explanation: df[df['Marks'] > 98] returns the rows where the 'Marks' column is greater than 98.

5.
```
import pandas as pd

studentDF = pd.DataFrame({
    'Name': ['Hetansh', 'Supriya', 'Mehak', 'Madhu', 'Rama'],
    'RollNumber': [10, 11, 12, 13, 14],
    'Grade': ['A', 'B', 'A', 'B', 'C'],
    'Marks': [99, 82, 95, 80, 60]
```

(i)
```
print(studentDF.head(2))
```

(ii) Retrieve the columns 'Name' and 'RollNumber'.
```
print(studentDF[['Name', 'RollNumber']])
```

(iii) Retrieve details of students with 'Marks' greater than 85.
```
print(studentDF[studentDF['Marks'] > 85])
```

(iv) Retrieve rows where 'Grade' is A.
```
print(studentDF[studentDF['Grade'] == 'A'])
```

(v) Use the iloc function to select the rows from index 2 to index 4.
```
print(studentDF.iloc[2:5])
```

(vi) Set column ID (earlier RollNumber) as the row indexes.
```
studentDF.set_index('RollNumber', inplace=True)
```

```
studentDF.index.name = 'ID'
print(studentDF)
```

(vii) Set the column labels of the DataFrame df to be ['Name', 'ID', 'Grade', 'MarksScored'] and display the modified DataFrame.

```
studentDF.columns = ['Name', 'Grade', 'MarksScored']
print(studentDF)
```

6. (i) head and tail

head: Returns the first n rows of the DataFrame (default is 5).

```
print(studentDF.head())
```

tail: Returns the last n rows of the DataFrame (default is 5).

```
print(studentDF.tail())
```

(ii) index and columns

index: Represents the row labels of the DataFrame.

```
print(studentDF.index)
```

columns: Represents the column labels of the DataFrame.

```
print(studentDF.columns)
```

7. (i) Output:

IMPRINT 1

8. (i) Output:

```
Name  RollNumber    Marks  Grade
A          2           2      2
B          2           2      2
C          1           1      1
```

(ii) Output: `83.2`

(iii) Output: `14.600684991371999`

(iv) Output: `99`

(v) Output: `Mehak`

(vi) Output:

```
         Name    RollNumber    Grade
0    Hetansh            10        A
1    Supriya            11        B
2      Mehak            12        A
```

(vii) Output:

```
A    2
B    2
C    1
Name: Grade, dtype: int64
```

(viii) Output: `array(['A', 'B', 'C'], dtype=object)`

(ix) Output:

```
Grade
A    97.0
B    81.0
C    60.0
Name: Marks, dtype: float64
```

(x) Output:

```
          Name    Grade    MarksScored
ID
10    Hetansh        A             99
11    Supriya        B             82
12      Mehak        A             95
13      Madhu        B             80
14       Rama        C             60
```

```
10     Hetansh        A            99
11     Supriya        B            82
12       Mehak        A            95
13       Madhu        B            80
14        Rama        C            60
```

(xi) Output:

```
    Full Name   RollNumber    Grade   Total Marks
0     Hetansh           10        A            99
1     Supriya           11        B            82
2       Mehak           12        A            95
3       Madhu           13        B            80
4        Rama           14        C            60
```

9. (i) Retrieve the summary statistics of the DataFrame

```
print(studentDF.describe())
```

(ii) Determine the minimum marks.

```
print(studentDF['MarksScored'].min())
```

(iii) Determine the average marks.

```
print(studentDF['MarksScored'].mean())
```

(iv) Retrieve the number of occurrences for each unique value in the 'Grade' column.

```
print(studentDF['Grade'].value_counts())
```

(v) Add a new column ModeratedMarks to the DataFrame to be calculated as Marks + 1

```
studentDF['ModeratedMarks'] = studentDF['MarksScored'] + 1
print(studentDF)
```

(vi) Drop the record of the student with the roll number 14 from the DataFrame.

```
studentDF.drop(14, inplace=True)
print(studentDF)
```

(vii) Rename the column 'RollNumber' to 'RNo'

```
studentDF.rename(columns={'RollNumber': 'RNo'}, inplace=True)
print(studentDF)
```

(viii) Write the contents of the DataFrame df to a CSV file named 'details.csv'.

```
studentDF.to_csv('details.csv', index=False)
```

(ix) Group the DataFrame by 'Grade' and calculate the average marks for each group.

```
print(studentDF.groupby('Grade')['MarksScored'].mean())
```

3. Data Visualization

Assessment

A. 1. b 2. a 3. d 4. c 5. d

B. 1. True 2. True 3. True 4. False 5. False

C. 1. plt.title() 2. plt.hist() 3. plt.savefig() 4. scatter

D. 1.

```
import matplotlib.pyplot as plt

temperatures = [22, 24, 26, 25, 27, 29, 28, 30, 28, 26, 25, 24, 23,
22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12]
times = range(len(temperatures))

plt.plot(times, temperatures, marker='o')
plt.xlabel('Time of Day')
plt.ylabel('Temperature (°C)')
plt.title('Temperature Differences Throughout the Day')
plt.show()
```

2.

```
import matplotlib.pyplot as plt

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']
channel1 = [1000, 1200, 1300, 1400, 1500, 1600]
channel2 = [900, 1000, 1100, 1200, 1300, 1400]

plt.plot(months, channel1, marker='o', label='Channel 1')
plt.plot(months, channel2, marker='x', label='Channel 2')
plt.xlabel('Months')
```

```
plt.ylabel('Subscribers')
plt.title('YouTube Channel Growth Over Six Months')
plt.legend()
plt.show()
```

3.
```
import matplotlib.pyplot as plt

weights = [50, 55, 52, 48, 60, 62, 59, 57, 53, 58, 56, 51, 54, 61, 55,
50, 52, 58, 53, 56, 60, 57, 49, 63, 61, 54, 51, 59, 55, 57, 53, 52, 55,
56, 50, 57, 60, 58, 53, 52, 59, 61, 56, 54, 51, 57, 58, 53, 59, 55]

plt.hist(weights, bins=10, edgecolor='black')
plt.xlabel('Weight (kg)')
plt.ylabel('Number of Students')
plt.title('Weight Distribution of Students')
plt.show()
```

4.
```
import matplotlib.pyplot as plt

study_hours = [2, 3, 4, 5, 6, 7, 8, 9, 6, 5, 4, 3, 5, 6, 7]
students = range(len(study_hours))

plt.bar(students, study_hours)
plt.xlabel('Student')
plt.ylabel('Hours Spent Studying')
plt.title('Study Hours of Students')
plt.show()
```

5.
```
import matplotlib.pyplot as plt

ages = [25, 30, 32, 28, 27, 35, 40, 29, 31, 37, 36, 26, 33, 42, 39,
34, 28, 26, 29, 38, 41, 30, 31, 35, 37, 28, 36, 39, 31, 33, 29, 32,
34, 37, 30, 28, 31, 39, 40, 33, 35, 36, 32, 29, 30]

plt.hist(ages, bins=10, edgecolor='black')
plt.xlabel('Age')
plt.ylabel('Number of Participants')
plt.title('Age Distribution of Marathon Race Participants')
plt.show()
```

4. Database Query Using SQL

Assessment

A. 1. b 2. a 3. b 4. b 5. b

B. 1. False 2. False 3. True 4. True
5. True 6. False

C. 1. DISTINCT 2. primary 3. constraint 4. UNIQUE 5. HAVING
6. outer, inner 7. ORDER BY

D. 1. (i), (ii), (iii), (iv), (vi), (vii), (x), (xi) are true.

2.
```
ItemCode: VARCHAR(10)
ItemName: VARCHAR(50)
Price: DECIMAL(10, 2)
```

3. No, a relation cannot have two identical tuples because each tuple must be unique to maintain the integrity of the relational model and to allow for the proper functioning of primary keys.

4. (i) The set of values that an attribute can take defines its domain.
(ii) A constraint is a rule that restricts the values that can be stored in a database.
(iii) A set of attributes that uniquely identify each tuple in a relation is called its candidate key.
(iv) A candidate key that is not chosen as the primary key is called an alternate key.

5. (i) Suitable primary key for Employee: Emp_id
Suitable primary key for Department: Dept_no

(ii) a.
```
INSERT INTO Employee (Emp_id, Name, Dept_no) VALUES ('E005', 'Anil Kumar', 20);
```
b.
```
INSERT INTO Department (Dept_no, Dept_name) VALUES (40, 'HR');
```
c.
```
INSERT INTO Employee (Emp_id, Name, Dept_no) VALUES ('E001', 'Anil Kumar', 20);
```
d.
```
INSERT INTO Department (Dept_no, Dept_name) VALUES (10, 'HR');
```
e.
```
INSERT INTO Employee (Emp_id, Name, Dept_no) VALUES ('E005', 'Anil Kumar', 50);
```
f.
```
INSERT INTO Department (Dept_no, Dept_name) VALUES (10, NULL);
```
g. Deleting the tuple with Dept_no 20 would be inconsistent because there are employees (Jatin Chawla) assigned to department 20.

6. (i) Employee (E_id, Ename, City, Salary, Department, YearofJoining)
Project (P_no, PName, City, DeptName, StartYear)
WorksOn (P_no, E_id)

IMPRINT 1

- Primary keys:
  - Employee: E_id
  - Project: P_no
  - WorksOn: (P_no, E_id) (composite key)
- Foreign keys:
  - WorksOn: P_no (references Project.P_no), E_id (references Employee.E_id)
- Example operations:

(i) `INSERT INTO Employee (E_id, Ename, City, Salary, Department, YearofJoining) VALUES (1, 'John Doe', 'New York', 60000, 'HR', 2021);`

(ii) `INSERT INTO Project (P_no, PName, City, DeptName, StartYear) VALUES (1, 'Project X', 'New York', 'HR', 2021);`

(iii) `INSERT INTO WorksOn (P_no, E_id) VALUES (1, 1);`

(iv) `INSERT INTO Employee (E_id, Ename, City, Salary, Department, YearofJoining) VALUES (1, 'John Doe', 'New York', 60000, 'HR', 2021);`

(v) `INSERT INTO Project (P_no, PName, City, DeptName, StartYear) VALUES (1, 'Project X', 'New York', 'HR', 2021);`

(vi) `INSERT INTO WorksOn (P_no, E_id) VALUES (1, 1);`

7. Primary keys:
   - `Suppliers: SNo`
   - `Parts: PNo`
   - `Project: JNo`
   - `Shipment: (SNo, PNo, JNo) (composite key)`

   Foreign keys:
   - `Shipment: SNo (references Suppliers.SNo), PNo (references Parts.PNo), JNo (references Project.JNo)`

8. (i) `INSERT INTO Suppliers (SNo, SName, Status, SCity) VALUES (1, NULL, 'Active', 'New York');`

(ii) `INSERT INTO Parts (PNo, PName, Colour, Weight, City) VALUES (1, NULL, 'Red', 10, 'New York');`

(iii) `INSERT INTO Suppliers (SNo, SName, Status, SCity) VALUES (1, 'Supplier1', 'Active', 'New York');`

(iv) `INSERT INTO Parts (PNo, PName, Colour, Weight, City) VALUES (1, 'Part1', 'Red', 10, 'New York');`

IMPRINT 1

(v) `INSERT INTO Project (JNo, Jname, JCity) VALUES (1, 'Project1', 'New York');`

(vi) `INSERT INTO Shipment (SNo, PNo, JNo, Quantity) VALUES (1, 1, 1, 100);`

9. `Employees:`

   `Primary Key: employee_id`

   `Foreign Key: department_id (references department_id in Department table)`

   `Department:`

   `Primary Key: department_id`

   `Resources:`

   `Primary Key: resource_id`

   `Foreign Key: department_id (references department_id in Department table)`

   `Table Creation Order`

   `To ensure the foreign keys can be correctly referenced, the Department table should be created first, followed by the Employees table, and finally the Resources table.`

10. Entity Integrity: It ensures that each table has a primary key and that the columns defined as the primary key are unique and not null. Example: In a table named Students, the student_id must be unique and cannot be null.

    Referential Integrity: It ensures that a foreign key value always points to an existing row in another table. Example: In the table Orders, the customer_id must exist in the Customers table.

11. (i) <18, "Mukesh Agrawal", 11, "C", 88>: Successful. No constraint violation.

    (ii) <21, "Sanjay", 11, "A", 76>: Unsuccessful. Violates the unique constraint on S_ID (21 already exists).

    (iii) <NULL, "Phule Bai", 11, "A", 88>: Unsuccessful. Violates the primary key constraint (S_ID cannot be NULL).

    (iv) <20, NULL, 11, "A", 88>: Unsuccessful. Violates the NOT NULL constraint on SName.

    (v) <20, NULL, 11, NULL, NULL>: Unsuccessful. Violates the NOT NULL constraint on SName.

    (vi) <20, NULL, NULL, NULL, NULL>: Unsuccessful. Violates the NOT NULL constraint on SName.

12. (i) <799575933699, NULL, "Pooja", "1990-10-01", 9776626565>: Unsuccessful. Violates the NOT NULL constraint on last_name.

(ii) <712349049911,"Singh","Gagan","1990-11-11",9812476543>: Unsuccessful. Violates the uniqueness constraint on AadharNo (712349049911 already exists).

13. 1. Insert the new manager as a regular employee first:

```
INSERT INTO Employee (Emp_id, Name, Dept_no) VALUES ('E0011',
'Rashmi Singhania', NULL);
```

2. Insert the new department:

```
INSERT INTO Department (Dept_no, Dept_name) VALUES (6, 'Food and
Beverage');
```

3. Update the new employee to be the manager of the new department:

```
UPDATE Employee SET Dept_no = 6 WHERE Emp_id = 'E0011';
```

Assertion Reasoning Based Questions

1. d 2. c. 3. b

Case-based Questions

1. (i) Display the structure of the table:

```
DESCRIBE Stationary;
```

(ii) Display records whose quantity is less than 5:

```
SELECT * FROM Stationary WHERE QTY < 5;
```

(iii) Display the name and price of stationary items whose names end with 'Marker':

```
SELECT SNAME, PRICE FROM Stationary WHERE SNAME LIKE '%Marker';
```

(iv) Display the price of the most expensive item:

```
SELECT MAX(PRICE) FROM Stationary;
```

(v) Display the average price of all items:

```
SELECT AVG(PRICE) FROM Stationary;
```

(vi) Display the brand name and count of items of each brand:

```
SELECT BRAND, COUNT(*) FROM Stationary GROUP BY BRAND;
```

(vii) Display the total number of records in the table:

```
SELECT COUNT(*) FROM Stationary;
```

(viii) Display the records in descending order of quantity:

```
SELECT * FR OM Stationary ORDER BY QTY DESC;
```

(ix) Display the records whose price is not known:

```
SELECT * FROM Stationary WHERE PRICE IS NULL;
```

(x) Display the name of each brand only once:

```
SELECT DISTINCT BRAND FROM Stationary;
```

2. (i) Display the name and price of each watch:

```
SELECT WName, Price FROM Watches;
```

(ii) Change the price of Apple iWatch to 59350:

```
UPDATE Watches SET Price = 59350 WHERE WName = 'Apple iWatch';
```

(iii) Display the records of all Sports watches:

```
SELECT * FROM Watches WHERE Type = 'Sports Watch';
```

(iv) Display the names of all brands, without repetition:

```
SELECT DISTINCT Brand FROM Watches;
```

(v) Display the count of designer watches:

```
SELECT COUNT(*) FROM Watches WHERE Type = 'Designer';
```

(vi) Display the count of watches of each brand:

```
SELECT Brand, COUNT(*) FROM Watches GROUP BY Brand;
```

(vii) Display the name and brand of watches that have been purchased in the year 2020:

```
SELECT WName, Brand FROM Watches WHERE YEAR(YOP) = 2020;
```

5. SQL: Working with Two Tables

Assessment

A. 1. c 2. c 3. b 4. a

B. 1. True 2. True 3. False 4. False 5. False

C. 1. attributes 2. equality 3. foreign 4. primary

D. 1. Referential Integrity Constraints: Referential integrity constraints ensure that a foreign key value always refers to an existing primary key value in another table.

Example: An EMPLOYEE works in a DEPARTMENT, these two entities are related via Dept_No of the table EMPLOYEE, which refers to the primary key Dept_No of DEPARTMENT.

DBMS uses foreign keys to enforce the integrity of the database. For example, DBMS will disallow an attempt to insert a tuple in the EMPLOYEE table having a Dept_No that is not present in the DEPARTMENT table. Using SQL, we can specify Dept_No as a foreign key in the table EMPLOYEE and indicate that it references the primary key Dept_No of the table DEPARTMENT as follows:

```
FOREIGN KEY (Dept_No) REFERENCES DEPARTMENT (Dept_No)
```

```
CREATE TABLE EMPLOYEE
(
ID INT PRIMARY KEY,
FName VARCHAR(20) NOT NULL,
LName VARCHAR(20) NOT NULL,
Gender CHAR(1) NOT NULL,
Address VARCHAR(30),
City VARCHAR(20),
Pin_Code CHAR(6),
DOB DATE,
Salary INT NOT NULL,
Dept_No SMALLINT,
     FOREIGN KEY(Dept_No) REFERENCES DEPARTMENT(Dept_No)
);
```

2. SQL supports a NATURAL JOIN operator that removes the duplicate column common to both the tables and positions the common attribute as the first column in the result. A NATURAL JOIN query to get the details of all the managers of all the departments may be formulated as follows:

```
SELECT *
FROM EMPLOYEE
NATURAL JOIN DEPARTMENT;
```

3. Suppose we want to get complete details of all the departments' managers. The following SQL query joins the tables EMPLOYEE and DEPARTMENT to achieve this:

```
SELECT *
FROM DEPARTMENT, EMPLOYEE
WHERE EMPLOYEE.Dept_No = DEPARTMENT.Dept_No;
```

Note that the result of executing the above query yields the column Dept_No twice, once for each table. As the above query joins only those tuples which have identical values of Dept_No, such a join is called equijoin.

D. 1. (i)
```
SELECT STUDENT.Roll_Num, SUBJECT.Subject_Name
FROM STUDENT
JOIN SUBJECT ON STUDENT.Subject_No = SUBJECT.Subject_No;
```

(ii)
```
SELECT STUDENT.Roll_Num, STUDENT.Subject_Name, SUBJECT.Dept_No
FROM STUDENT
JOIN SUBJECT ON STUDENT.Subject_No = SUBJECT.Subject_No;
```

(iii)
```
SELECT STUDENT.Roll_Num, STUDENT.Subject_Name
FROM STUDENT
JOIN SUBJECT ON STUDENT.Subject_No = SUBJECT.Subject_No
WHERE SUBJECT.Subject_Name = 'Chemistry';
```

(iv)
```
SELECT STUDENT.Subject_Name
FROM STUDENT
JOIN SUBJECT ON STUDENT.Subject_No = SUBJECT.Subject_No
WHERE SUBJECT.Subject_Name = 'Physics';
```

(v)
```
SELECT SUBJECT.Subject_Name, COUNT(STUDENT.Roll_Num) AS
StudentCount
FROM SUBJECT
LEFT JOIN STUDENT ON SUBJECT.Subject_No = STUDENT.Subject_No
GROUP BY SUBJECT.Subject_Name;
```

(vi)
```
SELECT SUBJECT.Subject_Name, STUDENT.Roll_Num, STUDENT.Subject_
Name
FROM SUBJECT
LEFT JOIN STUDENT ON SUBJECT.Subject_No = STUDENT.Subject_No;
```

2. (i)
```
SELECT DEPARTMENT.Dept_Name, SUBJECT.Subject_Name
FROM DEPARTMENT
JOIN SUBJECT ON DEPARTMENT.Dept_No = SUBJECT.Dept_No
ORDER BY DEPARTMENT.Dept_Name, SUBJECT.Subject_Name;
```

(ii)
```
SELECT DEPARTMENT.Dept_Name, COUNT(SUBJECT.Subject_No) AS
NumberOfSubjects
FROM DEPARTMENT
LEFT JOIN SUBJECT ON DEPARTMENT.Dept_No = SUBJECT.Dept_No
GROUP BY DEPARTMENT.Dept_Name;
```

(iii)
```
SELECT SUBJECT.Subject_Name, SUBJECT.Dept_No
FROM SUBJECT
WHERE SUBJECT.Subject_No IN (SELECT DISTINCT Subject_No FROM
TEACHER);
```

3. (i)
```
SELECT *
FROM PASSENGERS
WHERE TNO = 12030;
```

Output:

PNR	TNO	PNAME	GENDER	AGE	TRAVELDATE
P004	12030	S K SAXENA	MALE	42	2018-10-12
P005	12030	S SAXENA	FEMALE	35	2018-10-12
P006	12030	P SAXENA	FEMALE	12	2018-10-12
P008	12030	J K SHARMA	MALE	65	2018-05-09
P009	12030	R SHARMA	FEMALE	58	2018-05-09

(ii)
```
SELECT COUNT(*) AS FemalePassengers
FROM PASSENGERS
WHERE TNO = 12030 AND GENDER = 'FEMALE';
```

Output:

FemalePassengers
3

(iii)
```
SELECT P.PNAME, T.TNAME
FROM PASSENGERS P
JOIN TRAINS T ON P.TNO = T.TNO;
```

Output

PNAME	TNAME
R N AGRAWAL	Amritsar Mail
P TIWARY	Ajmer Shatabdi
S TIWARY	Ajmer Shatabdi
S K SAXENA	Swarna Shatabdi
S SAXENA	Swarna Shatabdi
P SAXENA	Swarna Shatabdi
N S SINGH	Amritsar Mail
J K SHARMA	Swarna Shatabdi
R SHARMA	Swarna Shatabdi

(iv)
```
SELECT DISTINCT TRAVELDATE
FROM PASSENGERS
JOIN TRAINS ON PASSENGERS.TNO = TRAINS.TNO
WHERE TRAINS.TNAME = 'Ajmer Shatabdi';
```

Output:

TRAVELDATE
2018-11-10

(v) Part a.

TNAME	PNAME
Amritsar Mail	R N AGRAWAL
Swarna Shatabdi	S K SAXENA

Part b.

TNO	TNAME	NoOfBookings
12015	Ajmer Shatabdi	2
12030	Swarna Shatabdi	5
13005	Amritsar Mail	2

4. (i) a.
```
SELECT SHIPMENT.OrderNo, PRODUCT.Pname
FROM SHIPMENT
JOIN PRODUCT ON SHIPMENT.PNo = PRODUCT.PNo;
```
b.
```
SELECT SHIPMENT.OrderNo
FROM SHIPMENT
JOIN PRODUCT ON SHIPMENT.PNo = PRODUCT.PNo
WHERE PRODUCT.Brand = 'Whirlpool';
```
c.
```
SELECT DISTINCT PRODUCT.Pname
FROM SHIPMENT
JOIN PRODUCT ON SHIPMENT.PNo = PRODUCT.PNo
WHERE SHIPMENT.Price > 70000;
```
d.
```
SELECT PRODUCT.Pname, MIN(SHIPMENT.Price) AS MinPrice, SHIPMENT.SNo
FROM SHIPMENT
JOIN PRODUCT ON SHIPMENT.PNo = PRODUCT.PNo
GROUP BY PRODUCT.Pname, SHIPMENT.SNo;
```
e.
```
SELECT PRODUCT.City, SUM(SHIPMENT.Quantity) AS TotalQuantity
FROM SHIPMENT
JOIN PRODUCT ON SHIPMENT.PNo = PRODUCT.PNo
GROUP BY PRODUCT.City;
```
(ii) a.
```
SELECT SUPPLIER.Sname
FROM SUPPLIER
JOIN SHIPMENT ON SUPPLIER.SNo = SHIPMENT.SNo
WHERE SHIPMENT.PNo = 201;
```

b.
```
SELECT SUPPLIER.Sname
FROM SUPPLIER
JOIN SHIPMENT ON SUPPLIER.SNo = SHIPMENT.SNo
GROUP BY SUPPLIER.Sname
HAVING COUNT(SHIPMENT.OrderNo) > 2;
```

c.
```
SELECT SUPPLIER.Sname, COUNT(SHIPMENT.OrderNo) AS NumberOfOrders
FROM SUPPLIER
JOIN SHIPMENT ON SUPPLIER.SNo = SHIPMENT.SNo
GROUP BY SUPPLIER.Sname;
```

d.
```
SELECT DISTINCT SUPPLIER.Sname
FROM SUPPLIER
JOIN SHIPMENT ON SUPPLIER.SNo = SHIPMENT.SNo
WHERE SUPPLIER.SCity = 'Mumbai' AND SHIPMENT.Price > 50000;
```

e.
```
SELECT DISTINCT SUPPLIER.Sname
FROM SUPPLIER
JOIN SHIPMENT ON SUPPLIER.SNo = SHIPMENT.SNo
WHERE SUPPLIER.SCity = 'Mumbai' AND SHIPMENT.Quantity > 40;
```

f.
```
SELECT SHIPMENT.*
FROM SHIPMENT
JOIN SUPPLIER ON SHIPMENT.SNo = SUPPLIER.SNo
WHERE SUPPLIER.SCity = 'Bangalore';
```

Case-based Questions

1. (i)
```
SELECT Theatre.AUDI, Troupe.TNAME
FROM Theatre
JOIN Troupe ON Theatre.TROUPID = Troupe.TROUPID;
```

(ii)
```
SELECT Troupe.TNAME, Theatre.TITLE, Theatre.TICKET_PRICE
FROM Theatre
JOIN Troupe ON Theatre.TROUPID = Troupe.TROUPID
WHERE Theatre.AUDI = 1;
```

(iii)
```
SELECT LANGUAGE, COUNT(*) AS PlayCount
FROM Theatre
GROUP BY LANGUAGE;
```

(iv)
```
SELECT TNAME
FROM Troupe
ORDER BY TNAME;
```

(v)
```
SELECT Theatre.TITLE, Troupe.TNAME
FROM Theatre
JOIN Troupe ON Theatre.TROUPID = Troupe.TROUPID
WHERE Troupe.CITY = 'Delhi'
ORDER BY Theatre.SHOWDATE;
```

(vi)
```
ALTER TABLE Troupe
ADD COLUMN CONTACTNO VARCHAR(10);
```

6. Computer Networks

Assessment

A. 1. c 2. b 3. c 4. b

B. 1. False 2. True 3. True 4. False 5. True
6. True 7. False 8. True 9. True 10. True

C. 1. Wi-Fi 2. packet 3. data rate 4. light 5. circuit
6. star 7. Bit rate 8. Bandwidth

D. 1. Give full form of the following:

MAC: Media Access Control

NIC: Network Interface Card

ARPANET: Advanced Research Projects Agency Network

LAN: Local Area Network

MAN: Metropolitan Area Network

WAN: Wide Area Network

2. Name four types of networks.

LAN (Local Area Network)

MAN (Metropolitan Area Network)

WAN (Wide Area Network)

PAN (Personal Area Network)

IMPRINT 1

3. Give two examples of MAN.

   City-wide Wi-Fi network

   Cable TV network within a city

4. Define the following terms:
   - **Transmission medium:** The physical path between the transmitter and receiver in a communication system.
   - **Topology:** The arrangement of various elements (links, nodes, etc.) of a computer network.
   - **Communication channel:** A medium used to transport information from a sender to a receiver.
5. Differentiate between the followings:
   - **Star topology and Tree topology:**
     - **Star Topology:** All nodes are connected to a central hub.
     - **Tree Topology:** A hierarchy of nodes where each node is connected to a parent node, forming a tree-like structure.
   - **Bus topology and Ring topology:**
     - **Bus Topology:** All nodes share a common communication line (bus).
     - **Ring Topology:** Each node is connected to exactly two other nodes, forming a ring.
   - **LAN and PAN:**
     - **LAN:** Network covering a small geographical area like a single building.
     - **PAN:** Network covering a very small area, typically within a range of a few meters (e.g., Bluetooth devices).
   - **MAN and WAN:**
     - **MAN:** Network covering a city.
     - **WAN:** Network covering a large geographical area, possibly worldwide.
   - **Hub and Switch:**
     - **Hub:** A basic networking device that connects multiple devices in a LAN.
     - **Switch:** A more advanced device that connects multiple devices and can filter and forward data to the correct destination.
   - **Modem and Router:**
     - **Modem:** Device that modulates and demodulates signals for data transmission.
     - **Router:** Device that forwards data packets between computer networks.
   - **Guided and unguided transmission media:**
     - **Guided:** Uses physical wires or cables (e.g., twisted pair, coaxial cable).
     - **Unguided:** Uses wireless methods (e.g., radio waves, infrared).

IMPRINT 1

6. Network used by Ramya:
   - PAN (Personal Area Network)
7. Minimum required data rate:
   - 7200 bits per second (bps)
   - Calculation:
   - 12 pages in 15 seconds, each page 600 characters, each character 8 bits
   - Total bits per second = (12 pages * 600 characters/page * 8 bits/character) / 15 seconds = 38400 bits / 15 = 2560 bits per second
8. Ways to connect devices in a LAN:
   - Using Ethernet cables
   - Using Wi-Fi (wireless)
9. Example of wired PAN:
   - Connecting a smartphone to a laptop via USB cable
10. Two characteristics of a hub:
    - Broadcasts data to all devices in a network.
    - Operates at the physical layer of the OSI model.
11. What is a modem?
    - Device that modulates analog signals to digital signals and vice versa for data transmission.
    - Categories: Internal modem, external modem
12. Short note on modems:
    - Modems are essential for converting digital data from a computer into an analog signal for phone lines and vice versa. They enable internet connectivity and are available in various forms, including dial-up, DSL, and cable modems.
13. Device for wireless Internet access:
    - Wireless router
14. Differentiate between sender and receiver:
    - Sender: Device or person that initiates the communication and sends the data.
    - Receiver: Device or person that receives the data sent by the sender.
15. Use of MAC address:
    - Unique identifier for network interfaces, used to ensure that data is transmitted to the correct device on a network.

Assertion Reasoning Based Questions

1. a 2. d. 3. a

Case-based Questions

1. a. Mumbai HO ⟶ Gurugram Operational Centre ⟶ Kolkata Branch Office
   ↓ ↓
   Madurai Branch Office Mumbai HO
   b. Gurugram Operational Centre
   c. a
   d. a
2. a. Administrative Unit
   b. Administrative Unit - Oncology Unit - Nephrology Unit - Neurology Unit - Orthopaedic Unit
   c. Switch
   d. Topology: Star
   Network cable: Ethernet cables

7. Computer Networks: Internet, Website, and Web Browsers

Assessment

A. 1. a 2. b 3. a 4. d 5. a
6. c 7. d 8. d 9. d

B. 1. False 2. True 3. True

C. 1. domain name 2. World Wide Web 3. cache,
4. web browsers 5. Cookies

D. 1. The protocol used is PPP (Point-to-Point Protocol).
2. Examples include XMPP (Extensible Messaging and Presence Protocol) and IRC (Internet Relay Chat).
3. VoIP (Voice over Internet Protocol) allows users to make voice calls using a broadband Internet connection instead of a regular (or analog) phone line.
4. a. **Web Server:** A web server is a software or hardware that serves web pages to users in response to their requests, which are made via web browsers.
   b. **Web Browser:** A web browser is a software application used to access information on the World Wide Web.
   c. **URL (Uniform Resource Locator):** A URL is the address used to access web pages and resources on the Internet.

IMPRINT 1

d. **Web Hosting:** Web hosting is a service that allows organizations and individuals to post a website or web page onto the Internet.

5. The **World Wide Web Consortium (W3C)** is an international community that develops open standards to ensure the long-term growth of the Web.
6. The **Domain Name System (DNS)** translates domain names into IP addresses, allowing browsers to load Internet resources.
7. A web browser works by sending a request to a web server, receiving the HTML document in response, and rendering it into a web page that users can interact with.

Assertion Reasoning Based Questions

1. b 2. d. 3. a

Case-based Questions

Protocol: https
Domain Name: www.incredibleindia.org
Path: /content/incredible-india-v2/en/experiences/
Name of the HTML Document: adventure.html

8. Societal Impacts

Assessment

A.

1. a	2. a	3. b	4. d	5. a
6. d	7. c	8. b	9. a	10. d

B.

1. False	2. True	3. True	4. True	5. False
6. False	7. True	8. True	9. True	10. True

C.

1. Trojan horse	2. Denial of Service (DoS)	3. copyright	4. 20
5. Digital Rights Management (DRM)	6. software license	7. cyber laws	8. e-waste
9. Braille			

D.

1. a. **Denial of Service (DoS):** An attack that aims to make a machine or network resource unavailable to its intended users by overwhelming it with a flood of internet traffic.
   b. **Plagiarism:** The act of using someone else's work or ideas without giving proper credit, presenting them as one's own.
   c. **Privacy Law:** Regulations that govern the collection, storage, and use of personal information about individuals, ensuring their privacy rights are protected.

IMPRINT 1

d. **Copyright Infringement:** The unauthorized use or reproduction of another person's work protected by copyright laws.

e. **E-waste:** Discarded electronic appliances such as mobile phones, computers, and televisions, which are no longer functional or desired.

2. An eavesdropping attack may involve listening directly to digital or analog voice transmissions, or intercepting text, images, or videos over a communication channel. An example of eavesdropping includes a tool installed on a phone that sends details of all calls to an intruder.

   Example: Two colleagues, Sonia and Madhu, are having lunch in a crowded cafe. They are discussing a new project with confidential details. At a nearby table, unbeknownst to them, sits their competitor, Dheeraj. Dheeraj leans in slightly, trying to catch snippets of their conversation to gain an unfair advantage on the project.

   This is a classic example of eavesdropping. Dheeraj is secretly listening in on a private conversation he is not a part of.

3. a. **File viruses and macro viruses:** File viruses attach themselves to executable files and activate when the infected program is run, whereas macro viruses infect documents and templates in programs that support macros, like Microsoft Word.

   b. **Adware and Spyware:** Adware is software that automatically displays or downloads advertising material when a user is online, while spyware secretly collects user information without their knowledge.

   c. **Copyright and Patent:** Copyright protects original works of authorship like books and music, whereas a patent protects inventions and grants exclusive rights to the inventor for a limited period.

   d. **Proprietary License and Free and Open Source License:** A proprietary license restricts users from modifying, sharing, or distributing the software, whereas a free and open source license allows users to freely use, modify, and distribute the software.

4. a. Boot Sector Virus: Michelangelo

   b. File Infector Virus: CIH (Chernobyl)

   c. Macro Virus: Melissa

   d. Worm: ILOVEYOU

   e. Online plagiarism checker: Turnitin

5. **IPR:** Intellectual Property Rights

   DRM: Digital Rights Management

   GPL: General Public License

6. Cookies store user preferences and login information, making the browsing experience more personalized and efficient. Cache stores frequently accessed web content locally, reducing loading times for websites.

IMPRINT 1

7. Freely accessible to everyone.

   Can be used, modified, and shared without restrictions

8. The software license contains information about the rights and restrictions for using the software, including how it can be installed, used, and distributed.
9. Because the source code is not made available to the users, preventing them from modifying or sharing the software.
10. Creative Commons is a non-profit organization that provides free licenses and tools to creators to enable them to share their work with the public. These licenses allow creators to specify which rights they reserve and which they waive, helping to facilitate the sharing and use of creative works while respecting the creator's rights.
11. MIT License

    Apache License 2.0

12. **Recycling:** Extracting valuable materials from e-waste for reuse.

    Incineration: Burning e-waste at high temperatures to reduce its volume and destroy harmful substances.

    Landfilling: Disposing of e-waste in landfills, though this method is discouraged due to environmental concerns.

    Reusing: Refurbishing and repairing electronic devices for extended use.

13. Prolonged use of technology can lead to issues such as eye strain, headaches, and poor posture. Overexposure to screens can disrupt sleep patterns, and constant use of mobile devices can contribute to repetitive strain injuries.

Assertion Reasoning Based Questions

1. a 2. c 3. a

Case-based Questions

1. Yes, Amarjit is a victim of a cybercrime known as ransomware, where attackers encrypt data or lock users out of their systems until a ransom is paid.
1. Yes, Kashvi can get legal protection under copyright law. The punishable offense committed by her competitor is copyright infringement, as they used Kashvi's original designs without permission and passed them off as their own.