SQL ASSIGNMENT

1.

```
SELECT education.id, education.name
  CASE
        WHEN (education.id>85 AND behaviour.id>85) OR (education.id>85 AND
sports.id>85) OR (sports.id>85 AND behaviour.id>85) THEN 'MERIT!'
        WHEN (education.id>65 AND behaviour.id>65) OR (education.id>65 AND
sports.id>65) OR (sports.id>65 AND behaviour.id>65) THEN 'PASS!'
        ELSE 'FAIL'
  END AS 'Outcome'
  FROM education
  INNER JOIN sports ON sports.id =  education.id
  INNER JOIN behaviour ON behaviour.id = sports.id;
```

2. ON APPLYING THE FOLLOWING COMMANDS - RESULTS -

a) INNER JOIN - 7 rows

```
SELECT COUNT(*)
FROM T1
INNER JOIN T2 ON T1.id = T2.id;
```

b) LEFT OUTER JOIN - 10 rows

```
SELECT COUNT(*)
FROM T1
LEFT JOIN T2
ON T1.id = T2.id;
```

c) RIGHT OUTER JOIN - 14 rows

```
SELECT COUNT(*)
FROM T1
RIGHT  JOIN T2
ON T1.id = T2.id;
```

d) FULL OUTER JOIN - 17 rows

```
SELECT COUNT(*)
FROM T1
FULL OUTER  JOIN T2
ON T1.id = T2.id;
```

e) CROSS JOIN - 72 rows

```
SELECT COUNT(*)
FROM T1
CROSS JOIN T2;
```

f) UNION - 6 rows

```
SELECT COUNT(*) FROM T1
UNION
SELECT COUNT(*) FROM T2;
```

g) UNION ALL - 6 rows

```
SELECT COUNT(*) FROM T1
```

```
UNION ALL
SELECT COUNT(*) FROM T2; // no duplicate rows
```

3.

```
SELECT * FROM (SELECT 1 a UNION ALL  2b) a
JOIN (SELECT 1a, 2b UNION ALL SELECT 1a,1b)b ON a.a=b.b
```

Here, the output will be

(SELECT 1 a UNION ALL 2b) a —>

a
1
2

(SELECT 1a, 2b UNION ALL SELECT 1a,1b)b —>

a	b
1	2
1	1

```
SELECT * FROM (SELECT 1 a UNION ALL  2b) a
JOIN (SELECT 1a, 2b UNION ALL SELECT 1a,1b)b ON a.a=b.b
```

a	a	b
1	1	1
2	1	2

4. UPDATE Emp SET Gender = CASE WHEN Gender = ‘M’ THEN ‘F’ ELSE ‘F’ END;

5. a) In query 1, the joining will be based on both TeamA.Inning = TeamB.Inning AND TeamA.Inning = 2, meaning even Inning = 1 and Inning = 2 from TeamA will have NULL values in TeamB table. And all rows from TeamA will be displayed.
In query 2, the joining will be based on only TeamA.Inning = TeamB.Inning, meaning even Inning = 1 and Inning = 2 from TeamA will have values in TeamB table. But, the final output will only display a single row which is filtered using WHERE clause.

b) Both will have the same output.

6.

```
SELECT table1.Item AS "Item Cart 1", table2.Item AS "Item Cart 2" FROM `table1` FULL
OUTER JOIN table2 ON table1.Item = table2.Item;
```

7. i)

```
SELECT * FROM (
(SELECT *  FROM FeedA
UNION ALL
SELECT *  FROM FeedB)
UNION ALL
SELECT *  FROM FeedC
) AS AllFeeds
GROUP BY AllFeeds.listing_number, AllFeeds.zip_code, AllFeeds.address, AllFeeds.city
HAVING COUNT(AllFeeds.listing_number) > 1;
```

ii)

```
INSERT into Listing (SELECT * From (
(SELECT *  FROM FeedA
UNION ALL
SELECT *  FROM FeedB)
UNION ALL
SELECT *  FROM FeedC
) AS AllFeeds);
```

8.

```
SELECT DISTINCT d1.`Customer ID` FROM deliveries AS d1
INNER JOIN deliveries d2
ON d1.`Customer ID` = d2.`Customer ID`
WHERE d1.`Delivery State` = 'New Delhi' AND d2.`Delivery State` = 'Mumbai';
```

9.

```
SELECT DISTINCT c1.`Candidate ID` FROM candidates c1
INNER JOIN candidates c2
INNER JOIN candidates c3
WHERE c1.`Description` = "Geologist" AND c2.`Description` = "Astrogator" AND
c3.`Description` = "Technician";
```

10.

```
SELECT Workflow, (`Case 1` + `Case 2` + `Case 3`) AS Passed FROM workflow_result;
```

11.