**Java Developer Assignment**

5. Create query for result

Table – routes

|  |  |  |
| --- | --- | --- |
| id | name | number |
| 1 | 200-D | SBV-BHJ |
| 2 | 300-A | ASD-WE |

Table - route\_points

|  |  |  |  |
| --- | --- | --- | --- |
| id | route\_id | order | distance |
| 1 | 1 | 1 | 0 |
| 2 | 1 | 2 | 100 |
| 3 | 2 | 1 | 0 |
| 4 | 2 | 2 | 50 |
| 5 | 2 | 3 | 100 |

Expected Result

|  |  |  |
| --- | --- | --- |
| route\_id | name | total\_distance |
| 2 | 300-A | 150 |
| 1 | 200-D | 100 |

**Query:**

SELECT route\_id, name, SUM(distance) As total\_distance FROM route\_points

inner join routes where route\_points.route\_id = routes.id

group by route\_id

order by route\_id DESC;

6. Make a query for the result by station\_id and slot. Find all stations where slot 1 and time less than 7.45 and order by time.

Table – station

|  |  |
| --- | --- |
| id | station\_name |
| 1 | STA 1 |
| 2 | STB 2 |

Table – times

|  |  |  |  |
| --- | --- | --- | --- |
| id | station\_id | slot | time |
| 1 | 1 | 1 | 6:30 |
| 2 | 1 | 2 | 6:45 |
| 3 | 1 | 1 | 7:40 |
| 4 | 1 | 2 | 7:25 |
| 5 | 2 | 1 | 6:40 |
| 6 | 2 | 2 | 6:50 |
| 7 | 2 | 1 | 7:50 |
| 8 | 2 | 2 | 8:15 |

Require columns in output

|  |  |  |  |
| --- | --- | --- | --- |
| station\_id | station\_name | slot | time |

**Query:**

SELECT station\_id, station\_name, slot, times.time FROM times INNER JOIN station

where station.id = times.station\_id and CAST(times.time as time) < '07:45';

7. Make a query for get result for same route by stop id for source and destination

Table – routes

|  |  |  |
| --- | --- | --- |
| id | name | number |
| 1 | 200-D | SBV-BHJ |
| 2 | 300-D | ASD-WER |

Table - route\_points

|  |  |  |  |
| --- | --- | --- | --- |
| id | station\_id | order | stop\_id |
| 1 | 1 | 1 | 1 |
| 2 | 1 | 2 | 2 |
| 3 | 2 | 1 | 1 |
| 4 | 2 | 2 | 2 |
| 5 | 2 | 3 | 3 |

Expected Result

|  |  |  |
| --- | --- | --- |
| route\_id | source\_stop\_id | dest\_stop\_id |
| 1 | 1 | 2 |
| 2 | 1 | 3 |

**Query:**

SELECT routes.id, route\_points.stop\_id AS source\_stop\_id, max(route\_points.stop\_id)

AS dest\_stop\_id FROM route\_points

inner join routes where route\_points.station\_id = routes.id

group by routes.id;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Optional: If the above output is achieved. Please try the following

Table – station

|  |  |
| --- | --- |
| id | station\_name |
| 1 | STA 1 |
| 2 | STB 2 |
| 3 | STC 3 |

Expected Result

|  |  |  |
| --- | --- | --- |
| route\_id | source\_stop\_name | dest\_stop\_name |
| 1 | STA 1 | STB 2 |
| 2 | STA 1 | STC 3 |

10. Give the output for the following and explain.

class Dog {

int leg;

}

class Main {

public static void main(String[] args) {

Dog d1 = new Dog();

d1.leg = 4;

Dog d2 = new Dog();

d2.leg = 3;

swap(d1, d2);

System.out.println("d1.leg = "+d1.leg);

System.out.println("d2.leg = "+d2.leg);

modify(d1);

System.out.println("d1.leg = "+d1.leg);

}

static void swap(Dog d1, Dog d2) {

Dog temp = d2;

d2 = d1;

d1 = temp;

}

static void modify(Dog d1) {

d1.leg = 0;

}

}

**OUTPUT & EXPLANATION:**

d1.leg = 4

d2.leg = 3

d1.leg = 0

d1 d2 are local references for that object created, hence changing the local reference doesn’t affect the original.

Modify is used to change the value of leg through the reference variable hence the leg of d1 is changed to 0.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_