***Name:Bijapure Naheen***

***R.No:08***

***Code:***

*import java.util.\*;*

*import java.sql.\*;*

*class Tuple {*

*Set<Integer> itemset;*

*int support;*

*Tuple() {*

*itemset = new HashSet<>();*

*support = -1;*

*}*

*Tuple(Set<Integer> s) {*

*itemset = s;*

*support = -1;*

*}*

*Tuple(Set<Integer> s, int i) {*

*itemset = s;*

*support = i;*

*}*

*}*

*class Apriori {*

*static Set<Tuple> c;*

*static Set<Tuple> l;*

*static int d[][];*

*static float min\_support;*

*public static void main(String args[]) throws Exception {*

*getDatabase();*

*c = new HashSet<>();*

*l = new HashSet<>();*

*Scanner scan = new Scanner(System.in);*

*int i, j, m, n;*

*System.out.println("Enter the minimum support (as an integer value):");*

*min\_support = scan.nextFloat();*

*Set<Integer> candidate\_set = new HashSet<>();*

*for(i=0 ; i < d.length ; i++) {*

*System.out.println("Transaction Number: " + (i+1) + ":");*

*for(j=0 ; j < d[i].length ; j++) {*

*System.out.print("Item number " + (j+1) + " = ");*

*System.out.println(d[i][j]);*

*candidate\_set.add(d[i][j]);*

*}*

*}*

*Iterator<Integer> iterator = candidate\_set.iterator();*

*while(iterator.hasNext()) {*

*Set<Integer> s = new HashSet<>();*

*s.add(iterator.next());*

*Tuple t = new Tuple(s, count(s));*

*c.add(t);*

*}*

*prune();*

*generateFrequentItemsets();*

*}*

*static int count(Set<Integer> s) {*

*int i, j, k;*

*int support = 0;*

*int count;*

*boolean containsElement;*

*for(i=0 ; i < d.length ; i++) {*

*count = 0;*

*Iterator<Integer> iterator = s.iterator();*

*while(iterator.hasNext()) {*

*int element = iterator.next();*

*containsElement = false;*

*for(k=0 ; k < d[i].length ; k++) {*

*if(element == d[i][k]) {*

*containsElement = true;*

*count++;*

*break;*

*}*

*}*

*if(!containsElement) {*

*break;*

*}*

*}*

*if(count == s.size()) {*

*support++;*

*}*

*}*

*return support;*

*}*

*static void prune() {*

*l.clear();*

*Iterator<Tuple> iterator = c.iterator();*

*while(iterator.hasNext()) {*

*Tuple t = iterator.next();*

*if(t.support >= min\_support) {*

*l.add(t);*

*}*

*}*

*System.out.println("-+- L -+-");*

*/\*for(int t=0 ; t < l.size() ; t++) {*

*System.out.print(itemset[t]+" : "+support[t]);*

*}\*/*

*for(Tuple t : l) {*

*System.out.println(t.itemset + " : " + t.support);*

*}*

*}*

*static void generateFrequentItemsets() {*

*boolean toBeContinued = true;*

*int element = 0;*

*int size = 1;*

*Set<Set> candidate\_set = new HashSet<>();*

*while(toBeContinued) {*

*candidate\_set.clear();*

*c.clear();*

*Iterator<Tuple> iterator = l.iterator();*

*while(iterator.hasNext()) {*

*Tuple t1 = iterator.next();*

*Set<Integer> temp = t1.itemset;*

*Iterator<Tuple> it2 = l.iterator();*

*while(it2.hasNext()) {*

*Tuple t2 = it2.next();*

*Iterator<Integer> it3 = t2.itemset.iterator();*

*while(it3.hasNext()) {*

*try {*

*element = it3.next();*

*} catch(ConcurrentModificationException e) {*

*// Sometimes this Exception gets thrown, so simply break in that case.*

*break;*

*}*

*temp.add(element);*

*if(temp.size() != size) {*

*Integer[] int\_arr = temp.toArray(new Integer[0]);*

*Set<Integer> temp2 = new HashSet<>();*

*for(Integer x : int\_arr) {*

*temp2.add(x);*

*}*

*candidate\_set.add(temp2);*

*temp.remove(element);*

*}*

*}*

*}*

*}*

*Iterator<Set> candidate\_set\_iterator = candidate\_set.iterator();*

*while(candidate\_set\_iterator.hasNext()) {*

*Set<Integer> s = candidate\_set\_iterator.next();*

*// These lines cause warnings, as the candidate\_set Set stores a raw set.*

*c.add(new Tuple(s, count(s)));*

*}*

*prune();*

*if(l.size() <= 1) {*

*toBeContinued = false;*

*}*

*size++;*

*}*

*System.out.println("\n=+= FINAL LIST =+=");*

*for(Tuple t:l) {*

*System.out.println(t.itemset + " : " + t.support);*

*}*

*}*

*static void getDatabase() throws Exception {*

*Class.forName("oracle.jdbc.driver.OracleDriver");*

*Connection con = DriverManager.getConnection("jdbc:oracle:thin:@DESKTOP-8KR21VC:1521:sqlplus","scott","finalbtech8");*

*Statement s = con.createStatement();*

*ResultSet rs = s.executeQuery("SELECT \* FROM apriori");*

*Map<Integer, List <Integer>> m = new HashMap<>();*

*List<Integer> temp;*

*while(rs.next()) {*

*int list\_no = Integer.parseInt(rs.getString(1));*

*int object = Integer.parseInt(rs.getString(2));*

*temp = m.get(list\_no);*

*if(temp == null) {*

*temp = new LinkedList<>();*

*}*

*temp.add(object);*

*m.put(list\_no, temp);*

*}*

*Set<Integer> keyset = m.keySet();*

*d = new int[keyset.size()][];*

*Iterator<Integer> iterator = keyset.iterator();*

*int count = 0;*

*while(iterator.hasNext()) {*

*temp = m.get(iterator.next());*

*Integer[] int\_arr = temp.toArray(new Integer[0]);*

*d[count] = new int[int\_arr.length];*

*for(int i=0 ; i < d[count].length ; i++) {*

*d[count][i] = int\_arr[i].intValue();*

*}*

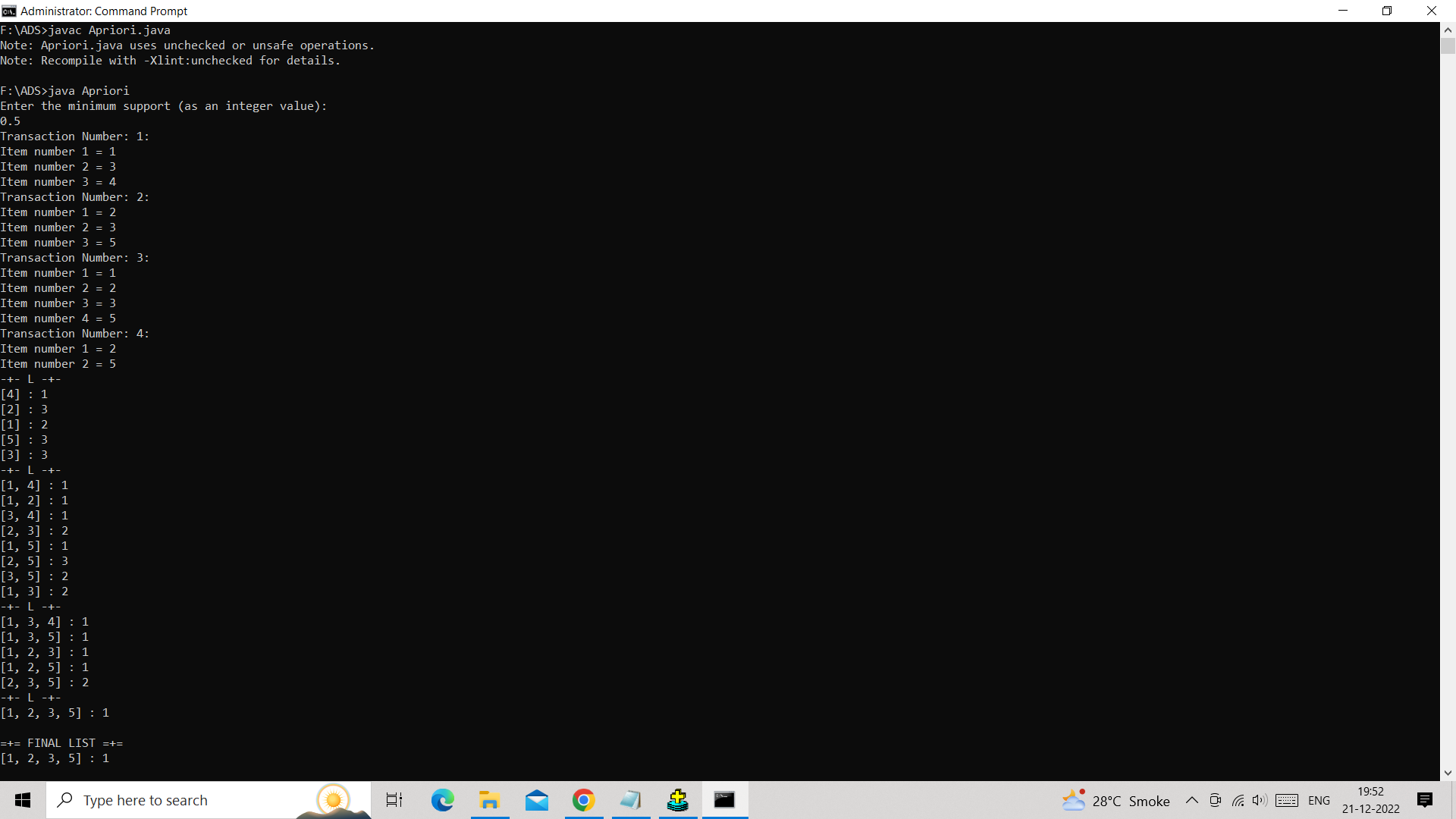
*count++;*

*}*

*}*

*}*

***Output:***

******