



eduresourceza.com

QUESTION 1 SQL**40 MARKS**

- 1.1 The secretary needs to list the pupils to create class lists. Sort all the pupil according to grade, class and then surname. (4)

**select * from pupils
order by PGrade√, PClass√, PName; √**

- 1.2 The ID field is no longer suitable. Change the ID to be the year of the learners date of birth the second and third letter of their name. (4)

**UPDATE learners√
SET ID = concat√ (year(PDOB) √, MID(PNAME,2,2)) √**

- 1.3 Insert a learner using your name and date of birth in Grade 11 R house M. Do not insert data for the ID field. (4)

**INSERT INTO excelhigh.learners (`PName`, `PGrade`, `PClass`, `PHouse`, `PDOB`, `PGender`) √√
VALUES ('Smith, Fred', 11, 'R', 'M', '1999-09-13 00:00:00', 'M'); √√**

- 1.4 Count the number of learners in each grade. Display the grade and the number of learners. (4)

**SELECT PGrade, COUNT(*)√√AS NUMINGRADE
FROM Learners
GROUP BY PGrade√√**

- 1.5 List the names of the boys who are in Grade 10 or Grade 11. (5)

**select Pname √
from learners√
where PGender = 'M'√ AND (PGrade=10√ OR PGrade =11); √**

- 1.6 Display the average age of all the female learners in each house. Rename this field AveAge. Display the house and the field AveAge. (7)

**select PHouse, √ AVG(year(now())√- year(PDOB)) √AS AveAge√
from learners
where PGender = 'F'√
GROUP BY PHouse; √√**

- 1.7 Display the classes who have more than 25 learners in each class. Display the grade, the class and the number in the class. (7)

**select PGrade, Pclass, √ Count(*)√ as TOTAL
from learners
GROUP√BY Pgrade√, PClass√
HAVING√ Total > 25; √**

- 1.8 The database will be rolled over for next year. Write two SQL statements to remove all the Grade 12 learners and then increase the Grade of each learner to the next grade. (5)

**DELETE FROM Learners√
WHERE PGrade = 12√**

**UPDATE learners√
SET PGrade = PGrade + 1√√**

QUESTION 2

25 MARKS

```

//Question 2.1 (4)
public class Source✓ {

    private ✓String✓ author, year, title, book, city; ✓

    //Question 2.2 (4)
    public Source✓ (String author, String year, String title, String
book, String city) ✓ {
        this.author = author; ✓✓
        this.year = year;
        removeBrackets();
        this.title = title;
        this.book = book;
        this.city = city;
    }

    //Question 2.6 (4)
    public void removeBrackets() ✓ {
        System.out.println(year);
        year✓ = year.substring(1, 5); ✓✓
    }

    //Question 2.7 (5)
    public String getCitation() ✓✓{
        String temp = "(" + author.substring(0, author.indexOf("
)); ✓✓
        temp = temp + ", " + year + ")"; ✓
        return temp; ✓
    }
//Question 2.3 (3) ✓
    public String getAuthor() {
        return author;
    }

    public String getYear() ✓ {
        return year;
    }

    public String getCity() ✓
    {
        return city;
    }
    //Question 2.4 (1)
    public void setYear(String year) ✓ {
        this.year = year;
    }

    //Question 2.5 (4)
    @Override
    public String✓ toString()✓ {

```

```

        return "" + author + "\t" + year + "\t" + title + "\t" + book
+ "\t" + city;
    }

}

```

QUESTION 3**48 MARKS**

//Question 3.1 (5)

```

public class SrcArray {

```

```

    private Source[] refArr = new Source[20];

```

```

    private int size = 0;

```

//Question 3.2 (9)

```

    public SrcArray() {

```

```

        try {

```

```

            Scanner scFile = new Scanner(new File("Ref.txt"));

```

```

            String line, a, y, t, b, c;

```

```

            while (scFile.hasNext()) {

```

```

                line = scFile.nextLine();

```

```

                // System.out.println(line);

```

```

                Scanner sc = new Scanner(line).useDelimiter(",");

```

```

                a = sc.next();

```

```

                y = sc.next();

```

```

                t = sc.next();

```

```

                b = sc.next();

```

```

                c = sc.next();

```

```

                //System.out.println(a + "\tIn the Scanner reading
file" + y+ "\t" + t+ "\t" + b+ "\t" + c);

```

```

                sc.close();

```

```

                refArr[size] = new Source(a, y, t, b, c);

```

```

                size++;

```

```

            }

```

```

            scFile.close();

```

```

        } catch (FileNotFoundException f) {

```

```

            System.out.println("File Not Found - check file name or
path");

```

```

        }

```

```

    }

```

//Question 3.4 (8)

```

    public void sort() {

```

```

        Source temp;

```

```

        for (int i = 0; i < size - 1; i++) {

```

```

            for (int j = i + 1; j < size; j++) {

```

```

                if

```

```

                (refArr[i].getAuthor().equals(refArr[j].getAuthor())) {

```

```

                    if

```

```

                    (refArr[i].getYear().compareTo(refArr[j].getYear()) > 0) {

```

3.7 (3)

```

        temp = refArr[i];
        refArr[i] = refArr[j]; ✓
        refArr[j] = temp;
    }
} else if
(refArr[i].getAuthor().compareTo(refArr[j].getAuthor()) > 0) {✓✓

    temp = refArr[i];
    refArr[i] = refArr[j]; ✓✓
    refArr[j] = temp;
}
}
}
}
//Question 3.5 (5)
public int countAll() {✓
    int i = 0;
    for (int j = 0; j < size; j++) {✓
        if (refArr[j].getCity().equalsIgnoreCase("London")) {✓
            i++;✓
        }
    }
    return i; ✓
}

//Question 3.6 (4)
public String listCitations() {✓
    String temp = "";
    for (int i = 0; i < size; i++) {✓
        temp = temp + refArr[i].getCitation() + "\n";✓
    }
    return temp; ✓
}

//Question 3.7
public void checkCitations() {✓
    char let = 'a';
    for (int i = 0; i < size - 1; i++) {✓
        if (refArr[i].getAuthor().equals(refArr[i +
1].getAuthor()) && refArr[i].getYear().substring(0, 4).equals(refArr[i
+ 1].getYear())) {✓✓
            if (refArr[i].getYear().length() == 4) {✓
                refArr[i].setYear(refArr[i].getYear() + let); ✓
            }
            let++;✓
            if (refArr[i + 1].getYear().length() == 4) {✓
                refArr[i + 1].setYear(refArr[i + 1].getYear() +
let); ✓
            }
        } else {

```

```

        let = 'a'; ✓
    }
}

//Question 3.3
public String toString() {
    String temp = "";
    for (int i = 0; i < size; i++) {
        temp = temp + refArr[i] + "\n";
    }
    return temp;
}
}

```

QUESTION 4**7 MARKS**

```

//Question 4.1
public class Grade11Prac2014 ✓{

    /**
     * @param args the command line arguments
     */

    public static void main(String[] args) {
        SrcArray ra = new SrcArray(); ✓
        //Question 4.2
        System.out.println("\nOriginal Array");
        System.out.println(ra); ✓
        //Question 4.3
        ra.sort();
        System.out.println("\nSorted Array");
        System.out.println(ra); ✓
        //Question 4.4
        System.out.println("\nList Citations");
        System.out.println(ra.listCitations()); ✓
        //Question 4.6
        System.out.println("\nCitations fixed Array");
        ra.checkCitations();
        System.out.println(ra); ✓
        //Question 4.5
        System.out.println("The number of source in London " +
ra.countAll()); ✓
    }
}

```