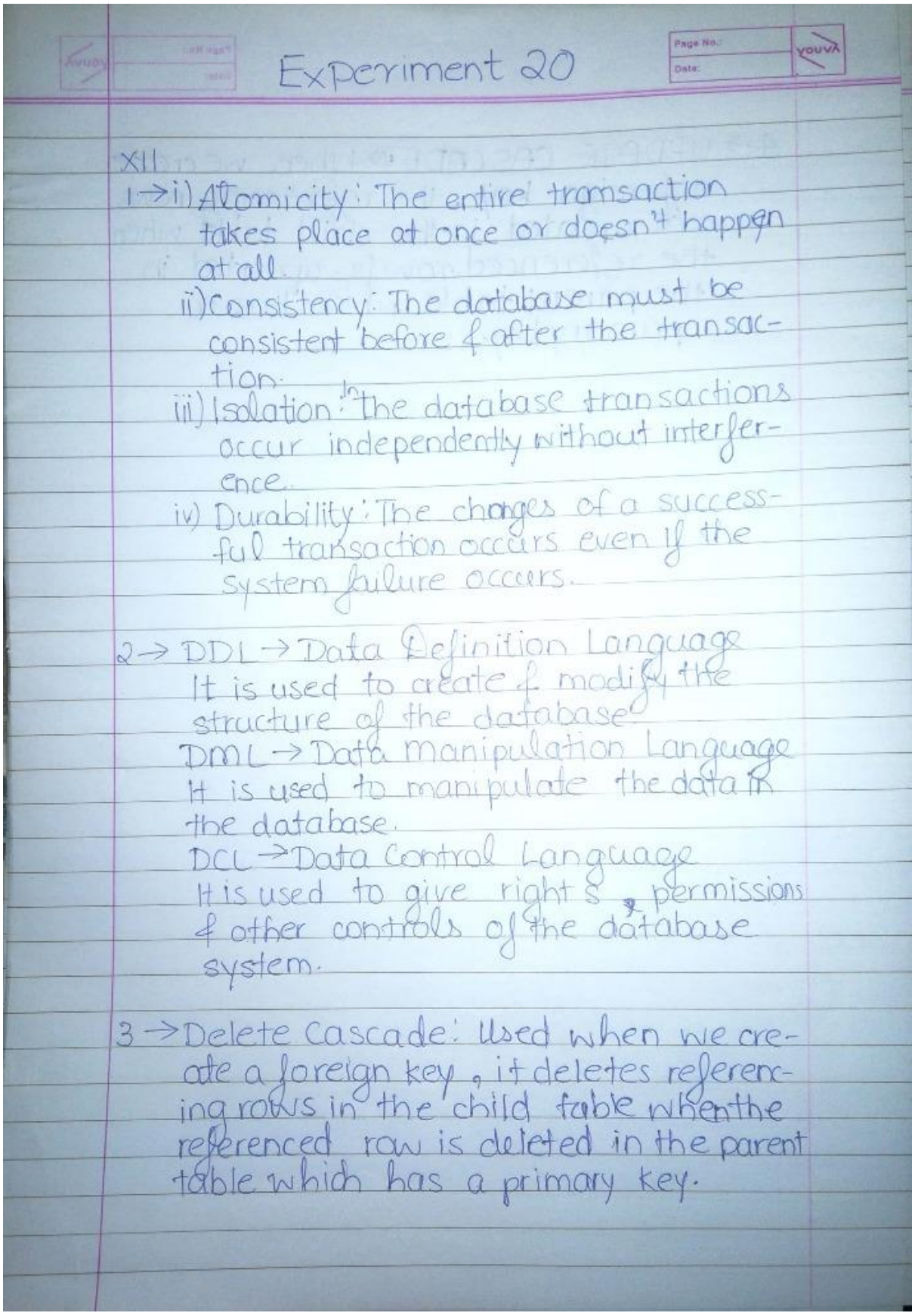
Experiment 20



Experiment 20

XII

1→ i) Atomicity: The entire transaction takes place at once or doesn't happen at all.

ii) Consistency: The database must be consistent before & after the transaction.

iii) Isolation: The database transactions occur independently without interference.

iv) Durability: The changes of a successful transaction occurs even if the system failure occurs.

2→ DDL → Data Definition Language
It is used to create & modify the structure of the database.
DML → Data manipulation Language
It is used to manipulate the data in the database.
DCL → Data Control Language
It is used to give right's, permissions & other controls of the database system.

3 → Delete Cascade: Used when we create a foreign key, it deletes referencing rows in the child table when the referenced row is deleted in the parent table which has a primary key.

4→ UPDATE CASCADE: When we create a foreign key, the referencing rows are updated in the child table when the referenced row is updated in the parent table which has a primary key.

```java
import java.sql.*;

public class Exp20a {
    public String database = "C:\\Users\\deong\\College\\Java\\Manual-Programs\\Experiment20\\SampleDatabase.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }

    public void closeConnection() {
        try {
            conn.close();
        } catch (SQLException e) {
            System.out.println("Close Connection Failed ?");
        }
    }

    public void updateQuery(String query) {
        try {
            Statement statement = conn.createStatement();
            statement.executeUpdate(query);
        } catch (SQLException e) {
            System.out.println("Error in updateQuery()");
        }
    }

    public static void main(String[] args) {
        Exp20a dbconn = new Exp20a();
        try {
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
        } catch (Exception e) {
            System.out.println("Error in Loading Driver");
        }
        dbconn.createConnection();

        dbconn.updateQuery("DELETE FROM Students WHERE id = 4;");
    }

}
```

Students

ID	Name	Percentage
1	abc	70
2	def	50
3	ghi	80
5	mno	60
6	John	80

Students

ID	Name	Percentage
1	abc	70
2	def	50
3	ghi	80
4	jkl	90
5	mno	60
6	John	80

```java
import java.sql.*;

public class Exp20b {
    public String database = "C:\\Users\\deong\\College\\Java\\Manual-Programs\\Experiment20\\SampleDatabase.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }

    public void query() {
        try {
            PreparedStatement st = conn.prepareStatement("update student set roll_no = 3 where name = 'Abhishek'");
            st.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("Error in Query");
        }
    }

    public static void main(String[] args) {
        Exp20b dbconn = new Exp20b();
        try {
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
        } catch (Exception e) {
            System.out.println("Error in Loading Driver");
        }
        dbconn.createConnection();
        dbconn.query();

    }
}
```

roll_no	name
1	Jay
2	Abhishek

roll_no	name
1	Jay
3	Abhishek

```java
import java.sql.*;

public class Exp20c {
    public String database = "C:\\Users\\deong\\College\\Java\\Manual-Programs\\Experiment19\\SampleDatabase.accdb";

    private Connection conn;

    // Create Connection
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }

    public void closeConnection() {
        try {
            conn.close();
        } catch (SQLException e) {
            System.out.println("Close Connection Failed ?");
        }
    }

    public void updateQuery(String query) {
        try {
            Statement statement = conn.createStatement();
            statement.executeUpdate(query);
        } catch (SQLException e) {
            System.out.println("Error in updateQuery()");
        }
    }

    public static void main(String[] args) {
        Exp20c dbconn = new Exp20c();
        try {
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
        } catch (Exception e) {
            System.out.println("Error in Loading Driver");
        }
        dbconn.createConnection();

        dbconn.updateQuery("UPDATE Students SET name = 'Jack' WHERE name = 'John';");
    }

}
```

Students

ID	Name	Percentage
1	abc	70
2	def	50
3	ghi	80
5	mno	60
6	Jack	80

Students

ID	Name	Percentage
1	abc	70
2	def	50
3	ghi	80
5	mno	60
6	John	80

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Exp20d {
   public String database = "C:\\Users\\deong\\College\\Java\\Manual-
Programs\\Experiment20\\SampleDatabase.accdb";

   private Connection conn;

   // Create Connection
```

```
    public void createConnection() {
        try {
            conn = DriverManager.getConnection("jdbc:ucanaccess://" + database);
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            System.exit(1);
        }
    }

    public void updateQuery(String query) {
        try {
            Statement statement = conn.createStatement();
            statement.executeUpdate(query);
        } catch (SQLException e) {
            System.out.println("Error in updateQuery()");
        }
    }

    public void insertQuery(String id, String name, float price) {
        try {
            PreparedStatement statement = conn.prepareStatement("INSERT INTO Product (ID, Name,
Price) VALUES(?,?,?);");
            statement.setString(1, id);
            statement.setString(2, name);
            statement.setFloat(3, price);
            statement.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Error in insertQuery()");
        }
    }

    public void selectQuery(String query) {
        try {
            Statement statement = conn.createStatement();
            ResultSet resultSet = statement.executeQuery(query);
            while (resultSet.next()) {
                String product = "Product " + resultSet.getString("ID") + ":" + "\n\tName : "
                        + resultSet.getString("name") + "\n\tPrice : " + resultSet.getFloat("price");
                System.out.println(product);
            }
        } catch (SQLException e) {
            System.out.println("Error in Printing With WHERE Condition");
        }
    }

    public static void main(String[] args) {
        Exp20d dbconn = new Exp20d();
        try {
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
        } catch (Exception e) {
            System.out.println("Error in Loading Driver");
```

```
        }
        dbconn.createConnection();

        dbconn.updateQuery("DROP TABLE Product;");
        dbconn.updateQuery("CREATE TABLE Product (ID VARCHAR(10), name VARCHAR(50), price
FLOAT);");
        dbconn.insertQuery("P1234", "abc", 300);
        dbconn.insertQuery("P1234", "abc", 900);
        dbconn.insertQuery("P1234", "abc", 700);
        dbconn.selectQuery("SELECT ID , name , price FROM Product WHERE price > 500 AND ID =
'P1234';");
    }
}
```

ID	name	price
P1234	abc	300
P1234	abc	900
P1234	abc	700

```
Product P1234:
        Name : abc
        Price : 900.0
Product P1234:
        Name : abc
        Price : 700.0
```