2023 Lancaster University

SCC 201 Lab Week 6

Instructions:

TASK1: Schema refinement and normal forms

FDs = {I->ITSAW, T->W}

The maximum Normalisation is 2NF.

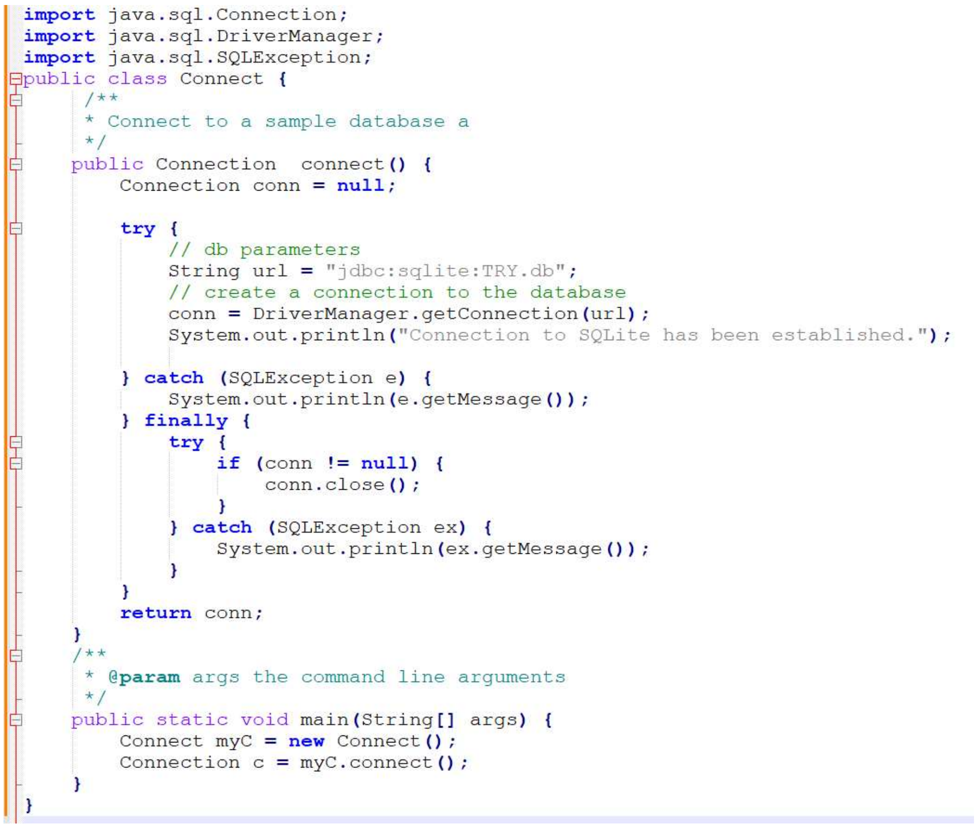
Due to T->W, if we decompose ITSAW to ITSA and TW, both will become 3NF and BCNF.

TASK2: Schema refinement and normal forms

Since A->B, and B->C, A->C.

Since C->DEF, A->DEF, then by augmentaiton we have A**G**->DEF**G**. Moreover, by augmentation, we can write AG->DEFG as A**ABC**G->**ABC**DEFG, which implies ABCG->ABCDEFG. But since A->B and A->C, we have AAAG->ABCDEFG. So AG->ABCDEFG.

TASK3: JDBC



```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class Connect {
    /**
     * Connect to a sample database a
     */
    public Connection  connect() {
        Connection conn = null;

        try {
            // db parameters
            String url = "jdbc:sqlite:TRY.db";
            // create a connection to the database
            conn = DriverManager.getConnection(url);
            System.out.println("Connection to SQLite has been established.");

        } catch (SQLException e) {
            System.out.println(e.getMessage());
        } finally {
            try {
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException ex) {
                System.out.println(ex.getMessage());
            }
        }
        return conn;
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Connect myC = new Connect();
        Connection c = myC.connect();
    }
}
```

TASK4: JDBC

Please see the attached Database.java

TASK5:SQL

A1:

```
      SELECT * FROM EMPLOYEE;
```

A2:

```
      SELECT SALARY FROM EMPLOYEE WHERE ADDRESS=”Lancaster”;
```

A3:

```
      SELECT EMPLOYEE.NAME FROM EMPLOYEE, DEPT_LOCATIONS WHERE
EMPLOYEE.DNO= DEPT_LOCATIONS.DNUMBER AND
DEPT_LOCATIONS.DLOCATION=”London”;
```

A4:

```
      SELECT  E.NAME
      FROM  EMPLOYEES E
      WHERE  NOT EXISTS
                    ((SELECT  P.PNUMBER
                       FROM  PROJECTS P)
                     EXCEPT
                      (SELECT  W.PNO
                       FROM  WORKSON W
                       WHERE  W.ESSN =  E.NAME))
```

A5:

```
      SELECT PNAME FROM PROJECTS WHERE PLOCATION=”Kahramanmaras”;
```

A6:

```
      SELECT DEPARTMENTS.DNAME FROM DEPARTMENTS,EMPLOYEE WHERE
((EMPLOYEE.NAME =”Sofiia” OR EMPLOYEE.NAME =”Ryan”) AND
EMPLOYEE.DNO=DEPARTMENTS.DNUMBER);
```

A7:

```
      SELECT EMPLOYEE.NAME FROM EMPLOYEE, DEPENDENT WHERE
((DEPENDENT.DEPENDENT_NAME =”ZSofia” OR DEPENTEND.DEPENDENT_NAME
=”Denver”) AND EMPLOYEE.SSN=DEPENDENT.ESSN);
```

TASK5:SQL (ADVANCED)

A1:

```
SELECT     EMPLOYEE.NAME
FROM        EMPLOYEE, DEPARTMENT
WHERE     DEPARTMENT.MGRSSN = EMPLOYEE.SSN   AND
                   EMPLOYEE.SSN IN
                             (SELECT DISTINCT ESSN
                               FROM DEPENDENT)
```

A2 :

```
SELECT NAME
FROM EMPLOYEE
WHERE SSN NOT IN (SELECT WORKSON.ESSN
                       FROM WORKSON, PROJECT
                       WHERE WORKSON.PNO = PROJECT.PNUMBER
                 AND PROJECT.DNUM = 5)
```

A3:

```
SELECT SUM(SALARY) AS SALARY_SUM, MAX(SALARY) AS MAX_SALARY,
       MIN(SALARY) AS MIN_SALARY, AVG(SALARY) AS AVERAGE_SALARY
FROM EMPLOYEE
```

A4:

```
SELECT SUM(SALARY) AS SALARY_SUM, MAX(SALARY) AS MAX_SALARY,
       MIN(SALARY) AS MIN_SALARY, AVG(SALARY) AS AVERAGE_SALARY
FROM EMPLOYEE, DEPARTMENT
WHERE EMPLOYEE.DNO = DEPARTMENT.DNUMBER
      AND DEPARTMENT.DNAME = 'RESEARCH'
```

A5:

```
SELECT COUNT(*) AS EMPLOYEE_COUNT
FROM EMPLOYEE, DEPARTMENT
WHERE EMPLOYEE.DNO = DEPARTMENT.DNUMBER
     AND DEPARTMENT.DNAME = 'RESEARCH'
```

A6:

```
SELECT COUNT(DISTINCT SALARY)
FROM EMPLOYEE
```

A7:

```
SELECT E.NAME
FROM WORKSON W, EMPLOYEE E
WHERE E.SSN = W.ESSN
AND EXISTS ((SELECT P.PNUMBER
               FROM PROJECT P
               WHERE P.DNUM = 5)
              EXCEPT (SELECT W1.PNO
                      FROM WORKSON W1
                      WHERE W1.ESSN = W.ESSN))
```

A8:

```
    SELECT E.NAME
    FROM WORKSON W, EMPLOYEE E
    WHERE E.SSN = W.ESSN
          AND EXISTS (SELECT P.PNUMBER
                      FROM PROJECT P
                      WHERE P.DNUM = 5
                      AND NOT EXISTS (SELECT W1.ESSN
                                      FROM WORKSON W1
                                       WHERE W1.ESSN = W.ESSN
                                 AND W1.PNO = P.PNUMBER))
```

A9:

```
    SELECT NAME
    FROM EMPLOYEE
    WHERE SUPERSSN IS NULL
```

A10:

```
    SELECT NAME
    FROM EMPLOYEE
    WHERE SSN IN (SELECT ESSN
                  FROM DEPENDENT
                  WHERE RELATIONSHIP = 'Child'
                  GROUP BY ESSN
                  HAVING COUNT(*) > 5)
```

A11:

```
SELECT DNO, COUNT(*) AS EMPLOYEE_COUNT, AVG(SALARY) AS
AVERAGE_SALARY

FROM EMPLOYEE

GROUP BY DNO
```

A12:

```
SELECT PROJECT.PNAME, PROJECT.PNUMBER, COUNT(*) AS EMPLOYEE_COUNT

FROM PROJECT, WORKSON

WHERE WORKSON.PNO = PROJECT.PNUMBER

GROUP BY PROJECT.PNUMBER
```

If YOU SEE ANYTHING WRONG WITH THIS QUERY- E-mail to u.turker@lancaster.ac.uk for a bonus !

A13:

```
SELECT PROJECT.PNAME, PROJECT.PNUMBER, COUNT(*) AS EMPLOYEE_COUNT

FROM PROJECT, WORKSON

WHERE WORKSON.PNO = PROJECT.PNUMBER

GROUP BY PROJECT.PNUMBER, PROJECT.PNAME
```

A14:

```
SELECT PROJECT.PNAME, PROJECT.PNUMBER, COUNT(*) AS EMPLOYEE_COUNT

FROM PROJECT, WORKSON

WHERE WORKSON.PNO = PROJECT.PNUMBER

GROUP BY PROJECT.PNUMBER, PROJECT.PNAME

HAVING COUNT(*) > 2
```

A15:

```
 SELECT DNO, COUNT(*) AS EMPLOYEE_COUNT
 FROM EMPLOYEE
 WHERE  SALARY * 12 > 40000
      AND DNO IN (SELECT DNO
                    FROM EMPLOYEE
                    GROUP BY DNO
                    HAVING COUNT(*) > 5)
 GROUP BY DNO
```