

Тема „База от данни за служителите в една фирма“

Изготвил: Радослава Христова ф.н. 71677

1. Обхват на модела. Дефиниране на задачата.

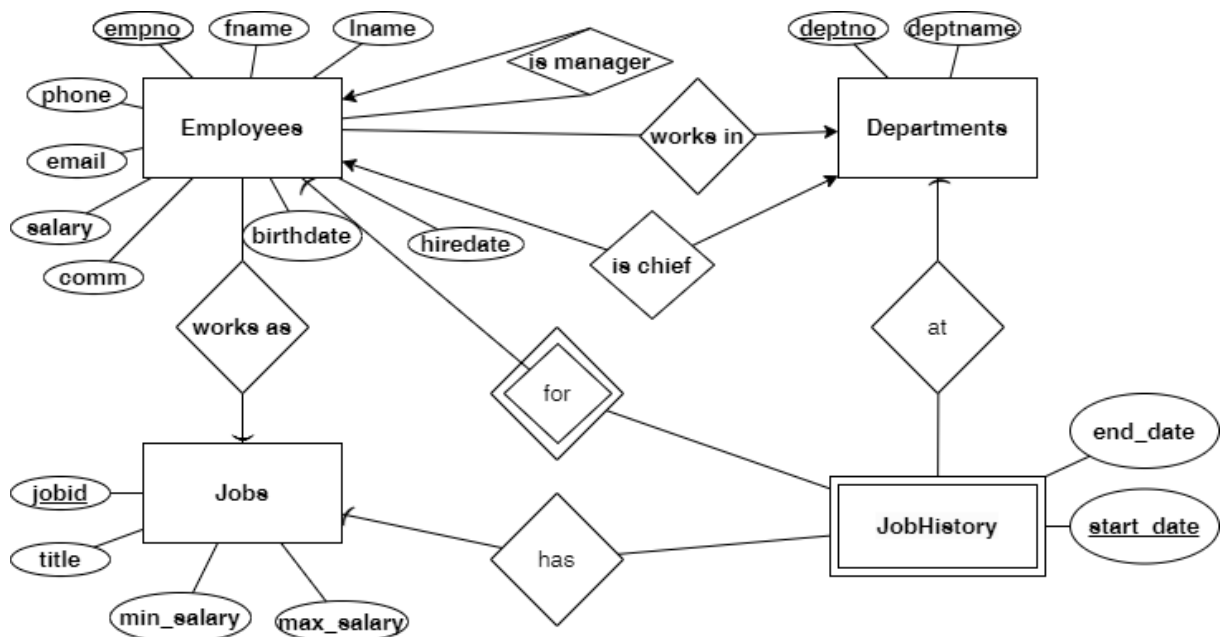
Информационна система съхранява информация за служителите в една фирма. Пазят се данни за служителите – номер на служител (уникален идентификатор), име, фамилия, email, телефонен номер, дата на наемане, рождена дата, заплата, комисионна. Всеки служител си има ръководител, с изключение на президента на компанията, за който ръководителя е NULL. Служителите работят в отдели. Като в един отдел могат да работят много служители, а един служител работи само в един отдел. Възможно е да има служител, за който да не е известен отдела, в който работи. За всеки отдел се пази информация за име на отдела, номер на отдела – уникален идентификатор. Всички отдел си имат ръководител, който е служител. Служител може да е ръководител само на един отдел. Възможно е да има отдел, за който да няма информация за ръководителя. Всеки служител заема определена длъжност. За всяка длъжност се пази информация за номер на длъжност – уникален идентификатор, име на длъжност, минимална заплата за тази длъжност и максимална заплата за тази длъжност. Една длъжност може да бъде заемана от много служители, но един служител в даден момент може да бъде само на една длъжност. За всяка заемана от служител длъжност във времето се пази история – кой служител е заемал тази длъжност, от коя дата до коя дата, каква длъжност е заемал и към кой отдел.

2. Правила и проверки

За базата от данни са в сила следните правила и проверки:

- Заплатата на служителя е положително число
- Комисионата на служителя е число между 1 и 100
- Минималната заплата за длъжност е по-малка от максималната заплата за същата длъжност
- Заплатата на служителя трябва да е в диапазона от минималната и максималната за съответната длъжност

3. E/R модел на данни



4. Релационен модел на данни

След преобразуване на модела „Същност-връзки“ към релационен модел и оптимизиране на връзките много-един се получава следния релационен модел на базата от данни:

```
Employees (empno, fname, lname, birthdate, hiredate, phone,  
           email, salary, comm, jobid, deptno, mgrno);  
Departments (deptno, deptname, empno_mgr);  
Jobs (jobid, title, min_salary, max_salary);  
JobHistory (empno, start_date, end_date, jobid, deptno);
```

Ограничения:

- Първични ключове (Primary key)

Employees: PK (empno); Departments: PK (deptno); Jobs: PK(jobid);

JobHistory (empno, start_date)

- Външни ключове (Foreign keys)

Employees: FK(jobid) -> Jobs (jobid); Employees: FK(mgrno) -> Employees(empno);

Employees: FK(deptno) -> Departments (deptno); JobHistory: FK(jobid) -> Jobs (jobid);

Departments: FK(empno_mgr) -> Employees(empno)

JobHistory: FK(empno) -> Employees(empno);

JobHistory: FK(deptno) -> Departments (deptno)

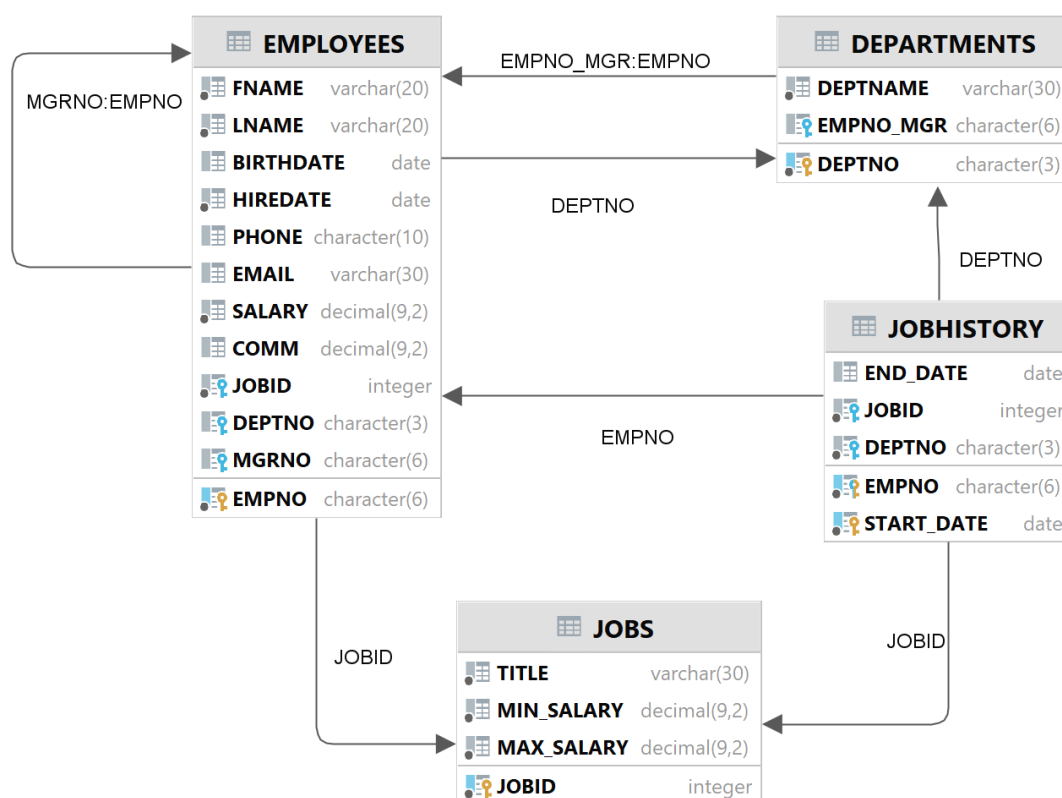
- Други ограничения (Check constraint)

Employees: CK (email - @ .); Employees: CK (salary > 0);

Jobs: CK (min_salary < max_salary and min_salary > 0);

Employees: CK (comm between 1 and 100);

5. Схема на базата от данни



6. Функции, тригери и изгледи

Функции

Описание на функция 1: По подаден номер на отдел, като входен параметър връща информация за служителите от този отдел – имена, email, телефон

SQL:

```
CREATE FUNCTION F_GET_EMP_INFO_BY_DEPTNO(P_DEPTNO CHAR(3))
RETURNS TABLE (EMPNAME VARCHAR(50), EMAIL VARCHAR(30), PHONE CHAR(10) )
RETURN
    SELECT FNAME || ' ' || LNAME, EMAIL, PHONE
    FROM EMPLOYEES
    WHERE DEPTNO = P_DEPTNO;
```

Извикване на функцията:

```
SELECT D.deptname, T.*
FROM TABLE(FN71677.F_GET_EMP_INFO_BY_DEPTNO('100')) T, DEPARTMENTS D
WHERE D.DEPTNO = '100';
```

Резултат:

	DEPTNAME	EMPNAME	EMAIL	PHONE
1	Finance	Nancy Greenberg	ngreenbe@micom.bg	515.124.45
2	Finance	Daniel Faviyet	dfaviyet@micom.bg	515.124.41
3	Finance	John Chen	jchen@micom.bg	515.124.42
4	Finance	Ismael Sciarra	isciarra@micom.bg	515.124.43
5	Finance	Jose Manuel Urman	jmurman@micom.bg	515.124.44
6	Finance	Luis Popp	lpopp@micom.bg	515.124.45

Описание на функция 2: По подадена ключова дума за длъжност, като входен параметър, връща информация за служителите на тази длъжност – име на отдел, име на служител, име на длъжност и заплата

SQL:

```
CREATE FUNCTION F_GET_EMP_INFO_BY_JOBID(P_TITLE VARCHAR(30))
RETURNS TABLE (DEPTNAME VARCHAR(30), EMPNAME VARCHAR(50), JOBTITLE
VARCHAR(30), SALARY DECIMAL(9,2) )
RETURN
    SELECT DEPTNAME, FNAME || ' ' || LNAME, TITLE, SALARY
    FROM EMPLOYEES E, DEPARTMENTS D, JOBS J
    WHERE E.DEPTNO = D.DEPTNO AND E.JOBID = J.JOBID
    AND E.JOBID IN (SELECT JOBID FROM JOBS
        WHERE INSTR(LOWER(TITLE), LOWER(P_TITLE)) > 0);
```

Извикване на функцията (тест 1):

```
SELECT *
FROM TABLE(FN71677.F_GET_EMP_INFO_BY_JOBID('PROG')) T
ORDER BY T.JOBTITLE;
```

Резултат:

	DEPTNAME	EMPNAME	JOBTITLE	SALARY
1	IT	Alexander Hunold	Programmer	9000.00
2	IT	Valli Pataballa	Programmer	4800.00
3	IT	Diana Lorentz	Programmer	4200.00
4	Purchasing	Sigal Tobias	Programmer	2800.00

Извикване на функцията (тест 2):

```
SELECT *  
FROM TABLE(FN71677.F_GET_EMP_INFO_BY_JOBID('SALES')) T  
ORDER BY T.JOBTITLE;
```

Резултат:

	DEPTNAME	EMPNAME	JOBTITLE	SALARY
1	Sales	John Russell	Sales Manager	14000.00
2	Sales	Karen Partners	Sales Manager	13500.00
3	Sales	Alberto Errazuriz	Sales Manager	12000.00
4	Sales	Gerald Cambrault	Sales Manager	11000.00
5	Sales	Eleni Zlotkey	Sales Manager	10500.00
6	Shipping	Curtis Davies	Sales Representative	3100.00

Тригери

Описание на тригер 1: При въвеждане на нов служител проверява въведената заплата и ако тя е не е в диапазона MIN_SALARY и MAX_SALARY, извежда съобщение за грешка.

SQL:

```
CREATE OR REPLACE TRIGGER TRIG_BEFORE_INSERT_EMP_SAL  
BEFORE INSERT ON EMPLOYEES  
REFERENCING NEW AS N  
FOR EACH ROW  
WHEN ((SELECT COUNT(*) FROM JOBS WHERE JOBID = N.JOBID  
AND (N.SALARY > MAX_SALARY OR N.SALARY < MIN_SALARY)) >= 1)  
BEGIN ATOMIC  
    DECLARE V_MINSAL DECIMAL(9,2);  
    DECLARE V_MAXSAL DECIMAL(9,2);  
  
    SET V_MINSAL = (SELECT MIN_SALARY FROM JOBS WHERE JOBID = N.JOBID);  
    SET V_MAXSAL = (SELECT MAX_SALARY FROM JOBS WHERE JOBID = N.JOBID);  
  
    SIGNAL SQLSTATE '75999' ('Salary is out of range: [' || V_MINSAL || ',  
' || V_MAXSAL || ']');  
END;
```

Тестване на тригер 1:

```
SELECT * FROM JOBS WHERE JOBID = 34;
```

```
INSERT INTO EMPLOYEES(EMPNO, FNAME, LNAME, BIRTHDATE, HIREDATE, PHONE,  
EMAIL, SALARY, COMM, JOBID, DEPTNO, MGRNO)  
VALUES (302, 'Penka', 'Petrova', '2000-01-13', '2022-12-13', '1232193021',  
'ppetrova@micom.bg', 3000, 20, 34, 60, 103);
```

```
SELECT * FROM EMPLOYEES WHERE EMPNO = 302;
```

```
INSERT INTO EMPLOYEES(EMPNO, FNAME, LNAME, BIRTHDATE, HIREDATE, PHONE,  
EMAIL, SALARY, COMM, JOBID, DEPTNO, MGRNO)  
VALUES (302, 'Penka', 'Petrova', '2000-01-13', '2022-12-13', '1232193021',  
'ppetrova@micom.bg', 5000, 20, 34, 60, 103);
```

```
SELECT * FROM EMPLOYEES WHERE EMPNO = 302;
```

Описание на тригер 2: При промяна на длъжността на служител да въвежда ред в таблицата JOBHISTORY за позицията, на която този служител е бил.

SQL:

```
CREATE OR REPLACE TRIGGER TRIG_AFTER_UPD_EMP  
AFTER UPDATE OF JOBID ON EMPLOYEES  
REFERENCING NEW AS N OLD AS O  
FOR EACH ROW  
WHEN (N.JOBID <> O.JOBID)  
BEGIN  
    INSERT INTO JOBHISTORY(EMPNO, START_DATE, END_DATE, JOBID, DEPTNO)  
    VALUES(O.empno, O.hiredate, CURRENT_DATE, O.JOBID, O.deptno);  
end;
```

Тестване на тригер 2:

```
SELECT * FROM EMPLOYEES WHERE EMPNO = 302;  
SELECT * FROM JOBHISTORY WHERE EMPNO = 302;
```

```
UPDATE EMPLOYEES  
SET JOBID = 27  
WHERE EMPNO = 302;
```

```
SELECT * FROM JOBHISTORY WHERE EMPNO = 302;
```

Изгледи

Описание на изглед 1: Изглед, който връща резултатно множество с информация за името на отдела, имената на служителите и длъжността, която заемат в момента

SQL:

```
CREATE VIEW V_EMPINFO
AS
SELECT D.DEPTNAME, E.FNAME || ' ' || E.LNAME AS EMPNAME, J.title AS
POSITION
FROM DEPARTMENTS D, EMPLOYEES E, JOBS J
WHERE D.DEPTNO = E.DEPTNO AND E.JOBID = J.JOBID;
```

Извикване на изглед 1:

```
SELECT * FROM V_EMPINFO
WHERE DEPTNAME = 'IT';
```

Резултат:

	DEPTNAME	EMPNAME	POSITION
1	IT	Penka Petrova	Sales Manager
2	IT	Alexander Hunold	Programmer
3	IT	Penka Petrova	Programmer
4	IT	Diana Lorentz	Programmer
5	IT	Valli Pataballa	Programmer

Описание на изглед 2: Изглед, който връща резултатно множество с информация за името на отдела длъжностите, които са заемали служителите в този отдел

SQL:

```
CREATE VIEW V_DEPT_JOBHISTORY
AS
SELECT D.DEPTNAME, E.FNAME || ' ' || E.LNAME AS EMPNAME, H.start_date,
H.end_date, J.TITLE AS POSITION
FROM DEPARTMENTS D, JOBHISTORY H, JOBS J, EMPLOYEES E
WHERE D.DEPTNO = H.DEPTNO AND H.JOBID = J.JOBID AND H.EMPNO = E.EMPNO;
```

Извикване на изглед 2:

```
SELECT * FROM V_DEPT_JOBHISTORY
WHERE DEPTNAME = 'IT';
```

Резултат:

	DEPTNAME	EMPNAME	START_DATE	END_DATE	POSITION
1	IT	Lex De Haan	1993-01-13	1998-07-24	Programmer
2	IT	Penka Petrova	2022-12-13	2024-05-02	Programmer

7. Приложение за достъп до базата

Java код:

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

public class DB2Test {

    private Connection connection;
    private Statement statement;
    private ResultSet resultSet;

    public void openConnection() {

        // Step 1: Load IBM DB2 JDBC driver

        try {

            DriverManager.registerDriver(new com.ibm.db2.jcc.DB2Driver());

        }

        catch(Exception cnfex) {

            System.out.println("Problem in loading or registering IBM DB2
JDBC driver");

            cnfex.printStackTrace();

        }

        // Step 2: Opening database connection

        try {

            connection =
DriverManager.getConnection("jdbc:db2://62.44.108.24:50000/SAMPLE",
"db2admin", "db2admin");

            statement = connection.createStatement();

        }

        catch(SQLException s){

            s.printStackTrace();

        }

    }

    public void closeConnection() {

        try {

            if(null != connection) {
```



```

        // cleanup resources, once after processing

        resultSet.close();
        statement.close();

        // and then finally close connection

        connection.close();

    }
}

catch (SQLException s) {

    s.printStackTrace();

}

}

= 2 public void select(String stmtnt, int column) {
    // Example query : select title, year from db2movie.movie - column

    try{
        resultSet = statement.executeQuery(stmtnt);
        String result = "";

        while(resultSet.next()) {
            for (int i = 1; i <= column; i++) {

                result += resultSet.getString(i);

                if (i == column) result += " \n";
                else result += ", ";

            }

            System.out.println("Executing query: " + stmtnt + "\n");
            System.out.println("Result output \n");
            System.out.println("----- \n");
            System.out.println(result);

        }
        catch (SQLException s)
        {
            s.printStackTrace();
        }

    }

    public void insert(String stmtnt) {

        try{
            statement.executeUpdate(stmtnt);
            System.out.println("Successfully inserted!");
        }

        catch (SQLException s){
            System.out.println("NOT inserted!");
            s.printStackTrace();
        }

    }
}

```

```

    }

    public void delete(String stmtnt) {

        try{

            statement.executeUpdate(stmtnt);

        }

        catch (SQLException s){

            s.printStackTrace();

        }

        System.out.println("Successfully deleted!");

    }

    public static void main(String[] args) {

        DB2Test db2Obj = new DB2Test();
        String stmtnt = "";

        db2Obj.openConnection();

        stmtnt = "SELECT EMPNAME, POSITION FROM FN71677.V_EMPINFO WHERE
DEPTNAME = 'Shipping'";

        db2Obj.select(stmtnt, 2);

        db2Obj.closeConnection();

    }

}

```

Результат:

Executing query: SELECT EMPNAME, POSITION FROM FN71677.V_EMPINFO WHERE DEPTNAME = 'Shipping'

Result output

```

-----
Curtis Davies, Sales Representative
Matthew Weiss, Stock Manager
Kevin Mourgos, Stock Manager
Shanta Vollman, Stock Manager
Payam Kaufling, Stock Manager
Adam Fripp, Stock Manager
Irene Mikkilineni, Stock Clerk
Douglas Grant, Stock Clerk
Alana Walsh, Stock Clerk
Vance Jones, Stock Clerk
Sarah Bell, Stock Clerk
Randall Perkins, Stock Clerk
Kelly Chung, Stock Clerk
Anthony Cabrio, Stock Clerk
Julia Dellinger, Stock Clerk
Alexis Bull, Stock Clerk
Nandita Sarchand, Stock Clerk

```