Database Systems (CS687)

SQL Project
Mar 24, 2020

Prepared By : Pooja Khanal(pk0050)

Table of contents

Project Description Document

1.System Overview

a. The Company Database Schema given in 5.5. and tables given in 5.6 Of *"Elmasri and Navathe, Fundamentals of Database Systems, Addison-Wesley, Seventh Edition "* is implemented in postgresQL. First, the corresponding tables are created and corresponding tuples are inserted.

i. The tables created are (taken from textbook)

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

ii. The Schema are as (taken from textbook)

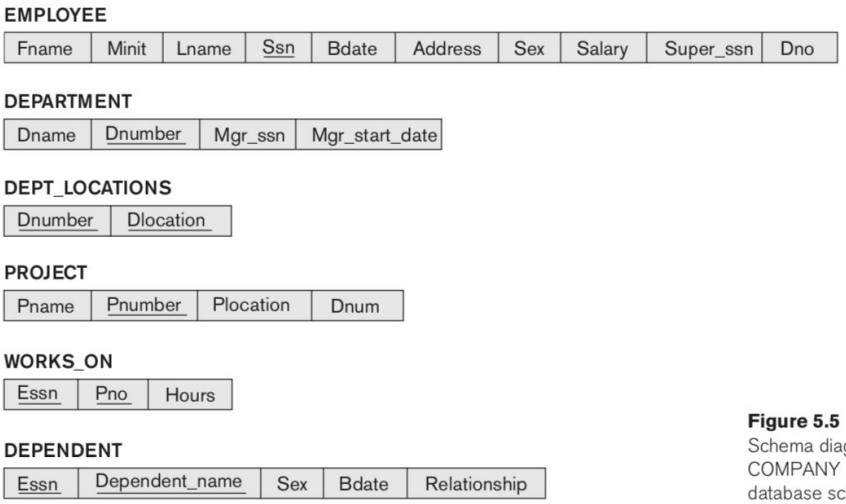
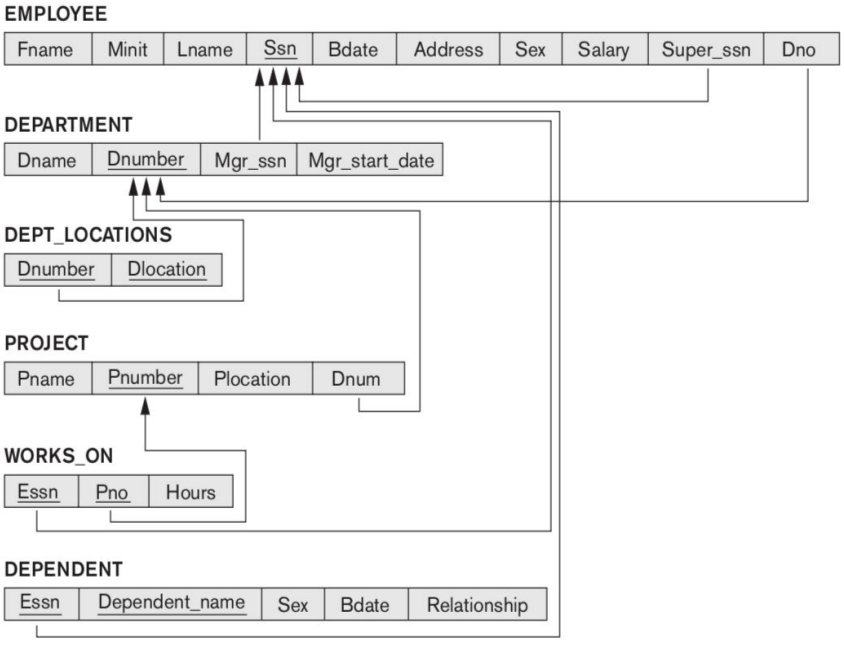


Figure 5.5
Schema diagram for the COMPANY relational database schema.

iii. The referential constraints are (taken from textbook)



The domains, key entity and referential integrity are preserved.

b. An ecpg program has been created to perform -add or -subtract the number of hours worked by an employee in a project. [1]
   i. –add will be followed by ssn, pno and hours (e.g., -add –ssn 123456789 –pno 1 –hours 5)
      - The program provides the total hours worked by the employee on the project.

- The number of hours worked on project **ProductX** by **John Smith** is **XX.X** hours (if the employee is working on the project).
- **John Smith** does not work on project **ProductX** (if the employing is not working on the project).

- If the employee is already working on a project, the number of hours will be incremented by 5 (as an example). Information as follows is provided.
  - The number of hours worked on project **ProductX** by employee **John Smith** increased from **XX.X** hours to **YY.Y** hours.
- If the employee is not working on the project, a new tuple for this employee will be added to Works_On relation. Information as follows is provided.
  - Employee **John Smith** started to work on project **ProductX** for **5** hours**.**

ii. –subtract will be followed by ssn, pno and hours (e.g., -subtract –ssn 123456789 –pno 1 –hours 5) [1]

- The program provides the total hours worked by the employee on the project.
  - The number of hours worked on project **ProductX** by **John Smith** is **XX.X** hours (if the employee is working on the project).
  - **John Smith** does not work on project **ProductX** (if the employee is not working on the project).
- If the employee is working more than 5 (as an example) hours on the project, the number of hours will be decremented by 5. Information as follows will be provided.
  - The number of hours worked on project **ProductX** by employee **John Smith** is reduced from **XX.X** hours to **YY.Y** hours.
  - If the employee is working less than or equal to 5 (as an example) hours on the project, the number of hours will be decremented by 5.
- Employees cannot work for 0 or negative hours, so the corresponding tuple will be deleted. Information as follows is provided.
  - Employee **John Smith** used to work on project **ProductX** for **XX.X** hours**.** The employee stopped working on this project.

iii. Both the arguments also prints, [1]

- information about the project numbers and hours the employee is working on.
- the name of his/her department
- total hours worked by the employee
- the number of his/her dependents
- his/her salary, and
- the difference between his/her salary and average salary in his/her department.

pk0050Company –add –ssn 123456789 –pno 1 –hours 5

pk0050Company –subtract –ssn 123456789 –pno 1 –hours 5

2.Referenced Documents

[1] Project SQL , Ramazan Aygun, CS687 Spring 2020

[2] https://www.postgresql.org/docs/

[3] Elmasri and Navathe, *Fundamentals of Database Systems*, Addison-Wesley, Seventh Edition

3. Concepts of Execution

3.1. User input and Prompts

add/subtract , ssn and hours are given by the user in terms of user provided arguments

Eg:

3.1.1.pk0050Company –add –ssn 123456789 –pno 1 –hours 5
3.1.2.pk0050Company –subtract –ssn 123456789 –pno 1 –hours 5

4. CREATE TABLE Statements

4.1. EMPLOYEE

4.1.1.

```
CREATE TABLE employee (
    fname character varying(15) NOT NULL,
    minit character(1),
    lname character varying(15) NOT NULL,
    ssn character(9) NOT NULL,
    bdate date,
    address character varying(30),
    sex character(1),
    salary numeric(10,2),
    superssn character(9),
    dno integer NOT NULL
);
```

4.1.2.

```
ALTER TABLE ONLY employee
      ADD CONSTRAINT employee_pkey PRIMARY KEY
(ssn);
```

4.1.3.

```
ALTER TABLE ONLY employee
      ADD CONSTRAINT employee_dno_fkey FOREIGN
KEY (dno) REFERENCES department(dnumber);
```

4.1.4.

```
ALTER TABLE ONLY employee
        ADD CONSTRAINT employee_superssn_fkey
FOREIGN     KEY     (superssn)     REFERENCES
employee(ssn);
```

4.2. DEPARTMENT

4.2.1.

```
CREATE TABLE department (
       dname character varying(15) NOT NULL,
       dnumber integer NOT NULL,
       mgrssn character(9) NOT NULL,
       mgrstartdate date
);
```

4.2.2.

```
ALTER TABLE ONLY department
            ADD  CONSTRAINT  department_dname_key  UNIQUE
    (dname);
```

4.2.3.

```
ALTER TABLE ONLY department
    ADD CONSTRAINT department_pkey PRIMARY KEY (dnumber);
```

4.1.2.

```
ALTER TABLE ONLY department
      ADD CONSTRAINT for_dept FOREIGN KEY (mgrssn)
REFERENCES employee(ssn);
```

4.3. DEPT_LOCATIONS

4.3.1.

```
CREATE TABLE dept_locations (
    dnumber integer NOT NULL,
    dlocation character varying(15) NOT NULL
);
```

4.3.2.

```
ALTER TABLE ONLY dept_locations
       ADD  CONSTRAINT  dept_locations_pkey  PRIMARY  KEY
(dnumber, dlocation);
```

4.3.3.

```
ALTER TABLE ONLY dept_locations
     ADD CONSTRAINT dept_locations_dnumber_fkey FOREIGN
KEY (dnumber) REFERENCES department(dnumber);
```

4.4. PROJECT

4.4.1.

```
CREATE TABLE project (
    pname character varying(15) NOT NULL,
    pnumber integer NOT NULL,
    plocation character varying(15),
    dnum integer NOT NULL
);
```

4.4.2.

```
ALTER TABLE ONLY project
    ADD CONSTRAINT project_pkey PRIMARY KEY (pnumber);

ALTER TABLE ONLY project
    ADD CONSTRAINT project_pname_key UNIQUE (pname);
```

4.4.3.

```
ALTER TABLE ONLY project
     ADD CONSTRAINT project_dnum_fkey FOREIGN KEY (dnum)
REFERENCES department(dnumber);
```

4.5. DEPENDENT

4.5.1.

```
CREATE TABLE dependent (
    essn character(9) NOT NULL,
    dependent_name character varying(15) NOT NULL,
    sex character(1),
    bdate date,
    relationship character varying(8)
);
```

4.5.2.

```
ALTER TABLE ONLY dependent
```

```
      ADD CONSTRAINT dependent_pkey PRIMARY KEY (essn,
dependent_name);
```

4.5.3.

```
ALTER TABLE ONLY dependent
       ADD CONSTRAINT dependent_essn_fkey FOREIGN KEY
(essn) REFERENCES employee(ssn);
```

4.6. WORKS_ON

4.6.1.

```
  CREATE TABLE works_on (
        essn character(9) NOT NULL,
        pno integer NOT NULL,
        hours numeric(3,1) NOT NULL
    );
```

4.6.2.

```
    ALTER TABLE pk0050.works_on OWNER TO pk0050;
    ALTER TABLE ONLY works_on
         ADD CONSTRAINT works_on_pkey PRIMARY KEY (essn,
    pno);
```

4.6.3.

```
    ALTER TABLE ONLY works_on
          ADD CONSTRAINT works_on_essn_fkey FOREIGN KEY
    (essn) REFERENCES employee(ssn);
```

4.6.4.

```
    ALTER TABLE ONLY works_on
          ADD CONSTRAINT works_on_pno_fkey FOREIGN KEY
    (pno) REFERENCES project(pnumber);
```

5. Database State and relational schema

5.1. EMPLOYEE

5.1.1.Table for EMPLOYEE relation

```
cs687=> select * from employee;
  fname   | minit |  lname  |    ssn    |   bdate    |         address          | sex |  salary  | superssn  | dno
----------+-------+---------+-----------+------------+--------------------------+-----+----------+-----------+-----
 James    | E     | Borg    | 888665555 | 1937-11-10 | 450 Stone, Houston, TX   | M   | 55000.00 |           |   1
 Jennifer | S     | Wallace | 987654321 | 1941-06-20 | 291 Berry, Bellaire, TX  | F   | 43000.00 | 888665555 |   4
 John     | B     | Smith   | 123456789 | 1965-01-09 | 731 Fondren, Houston, TX | M   | 30000.00 | 333445555 |   5
 Ramesh   | K     | Narayan | 666884444 | 1962-09-15 | 975 Fire Oak, Humble, TX | M   | 38000.00 | 333445555 |   5
 Joyce    | A     | English | 453453453 | 1972-07-31 | 5631 Rice, Houston, TX   | F   | 25000.00 | 333445555 |   5
 Alicia   | J     | Zelaya  | 999887777 | 1968-01-19 | 3321 Castle, Spring, TX  | F   | 25000.00 | 987654321 |   4
 Ahmad    | V     | Jabbar  | 987987987 | 1969-03-27 | 980 Dallas, Spring, TX   | M   | 25000.00 | 987654321 |   4
 Franklin | T     | Wong    | 333445555 | 1955-12-08 | 638 Voss, Houston, TX    | M   | 40000.00 | 888665555 |   5
(8 rows)
```

5.1.2.Schema for EMPLOYEE Relation

```
                    Table "pk0050.employee"
  Column  |         Type          | Modifiers | Storage  | Description
----------+-----------------------+-----------+----------+-------------
 fname    | character varying(15) | not null  | extended |
 minit    | character(1)          |           | extended |
 lname    | character varying(15) | not null  | extended |
 ssn      | character(9)          | not null  | extended |
 bdate    | date                  |           | plain    |
 address  | character varying(30) |           | extended |
 sex      | character(1)          |           | extended |
 salary   | numeric(10,2)         |           | main     |
 superssn | character(9)          |           | extended |
 dno      | integer               | not null  | plain    |
Indexes:
    "employee_pkey" PRIMARY KEY, btree (ssn)
Foreign-key constraints:
    "employee_dno_fkey" FOREIGN KEY (dno) REFERENCES department(dnumber)
    "employee_superssn_fkey" FOREIGN KEY (superssn) REFERENCES employee(ssn)
Referenced by:
    TABLE "dependent" CONSTRAINT "dependent_essn_fkey" FOREIGN KEY (essn) REFERENCES employee(ssn)
    TABLE "employee" CONSTRAINT "employee_superssn_fkey" FOREIGN KEY (superssn) REFERENCES employee(ssn)
    TABLE "department" CONSTRAINT "for_dept" FOREIGN KEY (mgrssn) REFERENCES employee(ssn)
    TABLE "works_on" CONSTRAINT "works_on_essn_fkey" FOREIGN KEY (essn) REFERENCES employee(ssn)
Has OIDs: no
```

5.2. DEPARTMENT

5.2.1.Table for DEPARTMENT relation

```
cs687=> select * from DEPARTMENT;
     dname      | dnumber |  mgrssn   | mgrstartdate
----------------+---------+-----------+--------------
 Research       |       5 | 333445555 | 1988-05-22
 Administration |       4 | 987654321 | 1995-01-01
 Headquarters   |       1 | 888665555 | 1981-06-19
(3 rows)
```

5.2.2.Schema for DEPARTMENT relation

```
                     Table "pk0050.department"
   Column    |         Type          | Modifiers | Storage  | Description
-------------+-----------------------+-----------+----------+-------------
 dname       | character varying(15) | not null  | extended |
 dnumber     | integer               | not null  | plain    |
 mgrssn      | character(9)          | not null  | extended |
 mgrstartdate | date                 |           | plain    |
Indexes:
    "department_pkey" PRIMARY KEY, btree (dnumber)
    "department_dname_key" UNIQUE CONSTRAINT, btree (dname)
Foreign-key constraints:
    "for_dept" FOREIGN KEY (mgrssn) REFERENCES employee(ssn)
Referenced by:
    TABLE "dept_locations" CONSTRAINT "dept_locations_dnumber_fkey" FOREIGN KEY (dnumber) REFERENCES department(dnumbe
r)
    TABLE "employee" CONSTRAINT "employee_dno_fkey" FOREIGN KEY (dno) REFERENCES department(dnumber)
    TABLE "project" CONSTRAINT "project_dnum_fkey" FOREIGN KEY (dnum) REFERENCES department(dnumber)
Has OIDs: no
```

5.3. DEPT_LOCATIONS

5.3.1.Table for DEPT_LOCATIONS relation

```
cs687=> select * from dept_locations;
 dnumber | dlocation
---------+-----------
       1 | Houston
       4 | Stafford
       5 | Bellaire
       5 | Sugarland
       5 | Houston
(5 rows)
```

5.3.2.Schema for DEPT_LOCATIONS relation

```
                Table "pk0050.dept_locations"
  Column   |        Type         | Modifiers | Storage  | Description
-----------+---------------------+-----------+----------+-------------
 dnumber   | integer             | not null  | plain    |
 dlocation | character varying(15) | not null  | extended |
Indexes:
    "dept_locations_pkey" PRIMARY KEY, btree (dnumber, dlocation)
Foreign-key constraints:
    "dept_locations_dnumber_fkey" FOREIGN KEY (dnumber) REFERENCES department(dnumber)
Has OIDs: no
```

5.4. PROJECT

5.4.1.Table for PROJECT relation

```
cs687=> select * from project;
      pname       | pnumber | plocation | dnum
------------------+---------+-----------+------
 ProductX         |       1 | Bellaire  |    5
 ProductY         |       2 | Sugarland |    5
 ProductZ         |       3 | Houston   |    5
 Reorganization   |      20 | Houston   |    1
 Newbenefits      |      30 | Stafford  |    4
 Computerization  |      10 | Stafford  |    4
(6 rows)
```

5.4.2.Schema for PROJECT relation

```
                      Table "pk0050.project"
  Column   |         Type          | Modifiers | Storage  | Description
-----------+-----------------------+-----------+----------+-------------
 pname     | character varying(15) | not null  | extended |
 pnumber   | integer               | not null  | plain    |
 plocation | character varying(15) |           | extended |
 dnum      | integer               | not null  | plain    |
Indexes:
    "project_pkey" PRIMARY KEY, btree (pnumber)
    "project_pname_key" UNIQUE CONSTRAINT, btree (pname)
Foreign-key constraints:
    "project_dnum_fkey" FOREIGN KEY (dnum) REFERENCES department(dnumber)
Referenced by:
    TABLE "works_on" CONSTRAINT "works_on_pno_fkey" FOREIGN KEY (pno) REFERENCES project(pnumber)
Has OIDs: no
```

5.5. DEPENDENT

5.5.1.Table for DEPENDENT relation

```
cs687=> select * from DEPENDENT;
   essn    | dependent_name | sex |   bdate    | relationship
-----------+----------------+-----+------------+--------------
 333445555 | Alice          | F   | 1986-04-05 | DAUGHTER
 333445555 | Thodore        | M   | 1983-10-25 | SON
 333445555 | Joy            | F   | 1958-05-03 | SPOUSE
 987654321 | Abner          | M   | 1942-02-28 | SPOUSE
 123456789 | Michael        | M   | 1988-01-04 | SON
 123456789 | Alice          | F   | 1988-12-30 | DAUGHTER
 123456789 | Elizabeth      | F   | 1967-05-05 | SPOUSE
(7 rows)
```

5.5.2.Schema for DEPENDENT Relation

```
                      Table "pk0050.dependent"
    Column     |         Type          | Modifiers | Storage  | Description
---------------+-----------------------+-----------+----------+-------------
 essn          | character(9)          | not null  | extended |
 dependent_name | character varying(15) | not null  | extended |
 sex           | character(1)          |           | extended |
 bdate         | date                  |           | plain    |
 relationship  | character varying(8)  |           | extended |
Indexes:
    "dependent_pkey" PRIMARY KEY, btree (essn, dependent_name)
Foreign-key constraints:
    "dependent_essn_fkey" FOREIGN KEY (essn) REFERENCES employee(ssn)
Has OIDs: no
```

5.6. WORKS_ON

5.6.1.Table for WORKS_ON Relation

```
cs687=> SELECT * from WORKS_ON;
   essn     | pno | hours
-----------+-----+-------
 123456789 |   1 |  32.5
 123456789 |   2 |   7.5
 666884444 |   3 |  40.0
 453453453 |   1 |  20.0
 453453453 |   2 |  20.0
 333445555 |   2 |  10.0
 333445555 |   3 |  10.0
 333445555 |  10 |  10.0
 333445555 |  20 |  10.0
 999887777 |  30 |  30.0
 999887777 |  10 |  10.0
 987987987 |  10 |  35.0
 987987987 |  30 |   5.0
 987654321 |  30 |  20.0
 987654321 |  20 |  15.0
 888665555 |  20 |  40.0
(16 rows)
```

5.6.2.Schema for WORKS_ON relation

```
               Table "pk0050.works_on"
 Column |     Type     | Modifiers | Storage  | Description
--------+--------------+-----------+----------+-------------
 essn   | character(9) | not null  | extended |
 pno    | integer      | not null  | plain    |
 hours  | numeric(3,1) | not null  | main     |
Indexes:
    "works_on_pkey" PRIMARY KEY, btree (essn, pno)
Foreign-key constraints:
    "works_on_essn_fkey" FOREIGN KEY (essn) REFERENCES employee(ssn)
    "works_on_pno_fkey" FOREIGN KEY (pno) REFERENCES project(pnumber)
Has OIDs: no
```

6.Code Outline for ecpg

List of Functions used

1. **Void populate()**
   a. **Purpose:** Get the details of the employee from database and populate the corresponding variables
   b. **Argument:** None
   c. **Return Type:** void

2. **Void addition()**
   a. **Purpose:**
      i. If the employee is working on the project, add given hours to the employee working hours on that project.
      ii. If the employee is not working on the project, create a new tuple for the employee and provide the given hour as the employee working hours on that project.
   b. **Argument:** None
   c. **Return Type:** void

3. **Void subtraction()**
   a. **Purpose:**
      i. If the employee is working on the project, subtract the given hour from the employee's working hours on that project.
      ii. If the employee is not working on the project , print that the given employee is not working on that project.
      iii. If the employee after subtraction results in a negative or zero working hour, remove that employee's tuples with that project from WORKS_ON database.
   b. **Argument:** None
   c. **Return Type:** void

4. **Void getDetails()**

a. **Purpose:**
   i. Get the name of the employee's department.
   ii. Get the project number and hours worked by the employee in that project.
   iii. Get the total hours worked by the employee.
   iv. Get the number of dependents of the employee.
   v. Get the salary of the employee.
   vi. Get the average salary of the department the employee belongs to.
   vii. Get the difference between the employee's salary and the average salary of his department.
b. **Argument:** None
c. **Return Type:** void

5. **Void handleCommandLineArguments(int argc, char* argv[])**
   a. **Purpose:** Take the command line arguments from the user
   b. **Argument:**
      i. The number of arguments given by the user.
      ii. The value of each argument given by the user.
   c. **Return type:**void

6. **Void main (int argc, char* argv[])**
   a. **Purpose:** The entry point of the program
   b. **Argument:**
      i. The number of arguments given by the user.
      ii. The value of each argument given by the user.
   c. **Return Type:** void

7. **Void print sqlca()**
   a. **Purpose:** To print any errors in the SQL commands
   b. **Argument:** none
   c. **Return type:** void

7. SQL QUERIES implemented

7.1. Query to get the employee's first name, middle initial, last name, ssn, salary, department name, department number and project name.

```
SELECT

     EMPLOYEE.fname, EMPLOYEE.minit, EMPLOYEE.lname,
     EMPLOYEE.ssn,EMPLOYEE.salary,

     DEPARTMENT.dname, DEPARTMENT.dnumber, PROJECT.pname

FROM

     pk0050.EMPLOYEE, pk0050.DEPARTMENT, pk0050.PROJECT

WHERE

     EMPLOYEE.dno = DEPARTMENT.dnumber AND

     EMPLOYEE.ssn = :input_ssn AND

     PROJECT.pnumber =:input_pno;
```

7.2. Query to get the hours the employee works on the given project.

```
SELECT

     WORKS_ON.hours

FROM

     pk0050.WORKS_ON

WHERE

     WORKS_ON.essn = :input_ssn AND WORKS_ON.pno =
:input_pno;
```

7.3. Query to insert a new tuple into the WORKS_ON table

```
INSERT
INTO
      pk0050.WORKS_ON("essn", "pno", "hours")
VALUES
      (:input_ssn, :input_pno, :input_hours);
```

7.4. Query to update the hours worked by the employee in the WORKS_ON table by the employee's ssn and employee's project number.

```
UPDATE
     pk0050.WORKS_ON SET hours = :input_hours
WHERE
     essn = :input_ssn and pno= :input_pno;
```

7.5. Query to delete the employee's tuple for employee's having working hours less than or equal to Zero.

```
DELETE FROM
     pk0050.works_on
WHERE
     hours <= 0 ;
```

7.6. Query to get project number, project name and the hours the employee with given ssn is working on.

```
SELECT
      WORKS_ON.pno, PROJECT.pname, WORKS_ON.hours
FROM
      pk0050.WORKS_ON, pk0050.PROJECT
WHERE
```

```
    WORKS_ON.essn = :input_ssn AND
    PROJECT.pnumber = WORKS_ON.pno;
```

7.7. Query to count the number of dependents of an employee by his/her ssn.

```
SELECT
    count(DISTINCT DEPENDENT.dependent_name)
FROM
    pk0050.WORKS_ON, pk0050.DEPENDENT
WHERE
    WORKS_ON.essn = DEPENDENT.essn AND
    WORKS_ON.essn = :input_ssn;
```

7.8. Query to get the average salary of the department the employee belongs to.

```
SELECT
    AVG(EMPLOYEE.salary)
FROM
    pk0050.EMPLOYEE
WHERE
    EMPLOYEE.dno = :dnumber;
```

Project Test Plan Document

1. Test Procedures

a. **Purpose:** To test the validity of software and to test if the software is acceptable to the market or not. For this program, various test cases are created.

b. **Procedures followed for Testing and Testing Strategies**

i. **Black box testing**

1. The input file was processed and seen if the output was written correctly to the output file or not.
2. Below mentioned test cases were run.

ii. **White box testing**

1. Modular approach was taken and each function was tested to see if it works or not. For such, the return type and return values were also compared.
2. After all the modules (functions) were tested, all the modules were integrated into one program and tested.

2. Test cases

1. Ssn of employee not working on the given pno
2. Ssn of the employee already working on the given pno
3. When the employee working hour reduces to equal or less than 0.
4. Employee working in his own department but project of different department.

3. Assumptions

1. Valid ssn, valid pno are provided.

2. An employee can be affiliated to one department as shown in the EMPLOYEE table. However, the employee can work on projects from other departments.
3. The hours column of the WORKS_ON table is added as NOT NULL so the existing **NULL** value for a tuple as shown in the textbook is replaced by **40** hours.

4. Tested Environments:

1. Unix server at **pk0050@voyager.uah.edu**

5. Sample Runs

Command to compile and build:

- ecpg pk0050Company.pgc
- cc pk0050Company.c -o pk0050Company -I/usr/include/postgresql -L/usr/lib -lecpg

For the below state of WORKS_ON table.

```
   essn     | pno | hours
-----------+-----+-------
 123456789 |   1 |  32.5
 123456789 |   2 |   7.5
 666884444 |   3 |  40.0
 453453453 |   1 |  20.0
 453453453 |   2 |  20.0
 333445555 |   2 |  10.0
 333445555 |   3 |  10.0
 333445555 |  10 |  10.0
 333445555 |  20 |  10.0
 999887777 |  30 |  30.0
 999887777 |  10 |  10.0
 987987987 |  10 |  35.0
 987987987 |  30 |   5.0
```

```
987654321 |  30 |  20.0
987654321 |  20 |  15.0
888665555 |  20 | 40
```
(16 rows)

a. Test Run I :

Command:

```
./pk0050Company -add -ssn 888665555 -pno 20 -hours 5
```

Output in console:

```
Submitted by Pooja Khanal(pk0050)

The number of hours worked on project Reorganization by James Borg is 40.0.

The number of hours worked on project Reorganization by employee James Borg increased from 40.0 hours to 45.0 hours.

Printing the details after updating

Employee James Borg works on project Reorganization having pnumber 20 for 45.0 hours.

James Borg is associated with  Headquarters department.

James Borg works for total of 45.0 hours.

The number of Dependents of James Borg is 0.

James Borg's salary is 55000.000.
```

The average salary for the Headquarters department is 55000.000.

The difference between the average salary of the Headquarters department and James Borg's salary is 0.000.

Change in WORKS_ON table

```
   essn    | pno | hours
-----------+-----+-------
 888665555 |  20 |  45.0
 123456789 |   1 |  32.5
 123456789 |   2 |   7.5
 666884444 |   3 |  40.0
 453453453 |   1 |  20.0
 453453453 |   2 |  20.0
 333445555 |   2 |  10.0
 333445555 |   3 |  10.0
 333445555 |  10 |  10.0
 333445555 |  20 |  10.0
 999887777 |  30 |  30.0
 999887777 |  10 |  10.0
 987987987 |  10 |  35.0
 987987987 |  30 |   5.0
 987654321 |  30 |  20.0
 987654321 |  20 |  15.0
(16 rows)
```

b. Test Run II :

Command:

```
./pk0050Company -sub -ssn 123456789 -pno 2 -hours 10
```

Output in console:

```
Submitted by Pooja Khanal(pk0050)


Employee John Smith used to work on project ProductY for
7.5 hours. The employee stopped working on this project.


Printing the details after updating


Employee John Smith works on project ProductX having
pnumber 1 for 32.5 hours.

John Smith is associated with  Research department.

John Smith works for total of 32.5 hours.

The number of Dependents of John Smith is 3.

John Smith's salary is 30000.000.

The average salary for the Research department is
33250.000.

The difference between the average salary of the Research
department and John Smith's salary is 3250.000.
```

Change in WORKS_ON table

```
  essn     | pno | hours
-----------+-----+-------
 888665555 |  20 |  45.0
 123456789 |   1 |  32.5
 666884444 |   3 |  40.0
 453453453 |   1 |  20.0
 453453453 |   2 |  20.0
 333445555 |   2 |  10.0
 333445555 |   3 |  10.0
 333445555 |  10 |  10.0
 333445555 |  20 |  10.0
 999887777 |  30 |  30.0
 999887777 |  10 |  10.0
 987987987 |  10 |  35.0
 987987987 |  30 |   5.0
 987654321 |  30 |  20.0
 987654321 |  20 |  15.0
(15 rows)
```