



B.TECH. (CSE)

V SEMESTER

**UE19CS301 - DATABASE MANAGEMENT
SYSTEM**

ASSIGNMENT-2

SUBMITTED BY

TEAM ID-3

NAME	SRN
T. LOHITH SRINIVAS	PES2UG19CS203
LALITHA SRAVANTI DASU	PES2UG19CS201
MEENAKSHI SURESH	PES2UG19CS228

AUGUST - DECEMBER 2021

**DEPARTMENT OF COMPUTER SCIENCE &
ENGINEERING**

**ELECTRONIC CITY CAMPUS,
BENGALURU - 560100,
KARNATAKA, INDIA**

REASONS FOR CHOOSING RDBMS OVER OTHER DATABASE SYSTEMS:

1. Simple Model

A Relational Database system is the most simple model, as it does not require any complex structuring or querying processes. It doesn't involve tedious architectural processes like hierarchical database structuring or definition. As the structure is simple, it is sufficient to be handled with simple SQL queries and does not require complex queries to be designed.

2. Data Accuracy

In the relational database system, there can be multiple tables related to one another with the use of a primary key and foreign key concepts. This makes the data to be non-repetitive. There is no chance for duplication of data. Hence the accuracy of data in the relational database is more than any other database system.

3. Easy Access to Data

In the Relational Database System, there is no pattern or pathway for accessing the data, as to another type of databases can be accessed only by navigating through a tree or a hierarchical model. Anyone who accesses the data can query any table in the relational database. Using join queries and conditional statements one can combine all or any number of related tables in order to fetch the required data. Resulting data can be modified based on the values from any

column, on any number of columns, which permits the user to effortlessly recover the relevant data as the result. It allows one to pick on the desired columns to be incorporated in the outcome so that only appropriate data will be displayed.

4. Data Integrity

Data integrity is a crucial characteristic of the Relational Database system.

Sturdy Data entries and legitimacy validations ensure that all the Data in the database confines within suitable arrangements and the data necessary for creating the relationships are present. This relational reliability amongst the tables in the database helps in avoiding the records from being imperfect, isolated or unrelated. Data integrity aids in making sure of the relational database's other significant characteristics like Ease of use, precision, and stability of the data.

5. Flexibility

A Relational Database system by itself possesses qualities for leveling up, expanding for bigger lengths, as it is endowed with a bendable structure to accommodate the constantly shifting requirements. This facilitates the increasing incoming amount of data, as well as the update and deletes wherever required. This model consents to the changes made to a database configuration as well, which can be applied without difficulty devoid of crashing the data or the other parts of the database.

A Data Analyst can insert, update or delete tables, columns or individual data in the given database system promptly and easily, in order to meet the business needs. There is supposedly no boundary on the number of rows, columns or tables a relational database can hold. In any practical application, development and transformation are restricted by the Relational Database Management System and the hardware contained by the servers. So these changes can create an alteration in other peripheral functional devices connected to the particular relational database system.

6. Normalization

The methodical style is maintained for making sure of a relational database structure is liberated of any variances that can make a difference in the integrity and accuracy of the tables in the database. A normalization process provides a set of regulations, characteristics, and purposes for the database structure and evaluation of a relational database model.

Normalization aims at illustrating multiple levels of breaking down the data. Any level of normalization is expected to be accomplished on the same level, that is, before moving ahead to the next levels. A relational database model is usually confirmed to be normalized, only when it satisfies the necessary conditions of the third normalization form. Normalization offers an impression of reassurance on the database plan, to be extra strong and reliable.

7. High Security

As the data is divided amongst the tables of the relational database system, it is possible to make a few tables to be tagged as confidential and others not. This segregation is easily implemented with a relational database management system, unlike other databases. When a data analyst tries to login with a username and password, the database can set boundaries for their level of access, by providing admission only to the tables that they are allowed to work on, depending on their access level.

8. Feasible for Future Modifications

As the relational database system holds records in separate tables based on their categories, it is straightforward to insert, delete or update records that are subjected to the latest requirements. This feature of the relational database model tolerates the newest requirements that are presented by the business. Any number of new or existing tables or columns of data can be inserted or modified depending on the conditions provided, by keeping up with the basic qualities of the relational database management system.

RDBMS chosen for execution: PostgreSQL

CREATE STATEMENTS

```
DROP DATABASE employeemanagement;
```

```
CREATE DATABASE employeemanagement;  
\c employeemanagement;
```

```
CREATE TABLE Building (  
  "id" Int,  
  "name" varchar(100),  
  "capacity" Int,  
  PRIMARY KEY ("id")  
);
```

```
CREATE TABLE Employee (  
  "SSN" Int,  
  "b_date" Date,  
  "Status" varchar(100),  
  "sex" varchar(10),  
  "s-name" varchar(100),  
  "l-name" varchar(100),  
  "Address" varchar(500),  
  "super_ssn" Int,  
  PRIMARY KEY ("SSN"),  
  CONSTRAINT "FK_Employee.super_ssn"  
  FOREIGN KEY ("super_ssn")  
  REFERENCES Employee("SSN")  
);
```

```
CREATE TABLE Department (  
  "name" varchar(100),  
  "Mssn" Int,  
  PRIMARY KEY ("name"),  
  CONSTRAINT "FK_Department.Mssn"  
  FOREIGN KEY ("Mssn")  
  REFERENCES Employee("SSN")  
);
```

```
CREATE TABLE houses (  
  "id" Int,  
  "dname" varchar(100),  
  "bid" Int,  
  PRIMARY KEY ("id"),  
  CONSTRAINT "FK_houses.dname"  
  FOREIGN KEY("dname")  
  REFERENCES Department("name"),
```

```
  CONSTRAINT "FK_houses.bid"  
  FOREIGN KEY ("bid")
```

```
REFERENCES Building("id")
);
```

```
CREATE TABLE Project (
"name" varchar(100),
"budget" Int,
PRIMARY KEY ("name")
);
```

```
CREATE TABLE assigned_to(
"id" Int,
"pname" varchar(100),
"dname" varchar(100),
PRIMARY KEY("id"),
CONSTRAINT "FK_assigned_to.pname"
FOREIGN KEY ("pname")
REFERENCES Project("name"),
CONSTRAINT "FK_assigned_to.dname"
FOREIGN KEY ("dname")
REFERENCES Department("name")
);
```

```
CREATE TABLE works_in(
"id" Int,
"essn" Int,
"dname" varchar(100),
PRIMARY KEY ("id"),
CONSTRAINT "FK_works_in.essn"
FOREIGN KEY ("essn")
REFERENCES Employee("SSN"),
CONSTRAINT "FK_works_in.dname"
FOREIGN KEY ("dname")
REFERENCES Department("name")
);
```

```
CREATE TABLE Dependents (
"Name" varchar(100),
"Essn" Int,
"Sex" varchar(10),
"b_date" Date,
"relationship" varchar(100),
PRIMARY KEY ("Name"),
CONSTRAINT "FK_Dependents.Essn"
FOREIGN KEY ("Essn")
REFERENCES Employee("SSN")
);
```

```
CREATE TABLE Salary_Range (
```

```
"id" Int,  
"grade" Int,  
"start_sal" Int,  
"end_sal" Int,  
PRIMARY KEY ("id")  
);
```

```
CREATE TABLE Earns(  
"id" Int,  
"sid" Int,  
"essn" Int,  
PRIMARY KEY ("id"),  
CONSTRAINT "FK_Earns.essn"  
FOREIGN KEY ("essn")  
REFERENCES Employee("SSN"),  
CONSTRAINT "FK_Earns.sid"  
FOREIGN KEY ("sid")  
REFERENCES Salary_Range("id")  
);
```

```
CREATE TABLE Leave_type (  
"id" Int,  
"Type" varchar(50),  
"NoOfDays" Int,  
"Essn" Int,  
PRIMARY KEY ("id")  
);
```

```
CREATE TABLE Avails_leave(  
"id" Int,  
"essn" Int,  
"lid" Int,  
PRIMARY KEY ("id"),  
CONSTRAINT "FK_Avails_leave.essn"  
FOREIGN KEY ("essn")  
REFERENCES Employee("SSN"),  
CONSTRAINT "FK_Avails_leave.lid"  
FOREIGN KEY ("lid")  
REFERENCES Leave_type("id")  
);
```

```
CREATE TABLE Skill (  
"id" Int,  
"name" varchar(100),  
"tech" varchar(100),  
PRIMARY KEY ("id")  
);
```



```
CREATE TABLE Has (  
  "id" Int,  
  "essn" Int,  
  "sid" Int,  
  PRIMARY KEY("id"),  
  CONSTRAINT "FK_Skill.essn"  
  FOREIGN KEY ("essn")  
  REFERENCES Employee("SSN"),  
  CONSTRAINT "FK_Skill.sid"  
  FOREIGN KEY ("sid")  
  REFERENCES Skill("id")  
);
```

```
CREATE TABLE Works_On (  
  "Start_date" Date,  
  "Hours" Int,  
  "Essn" Int,  
  "pname" varchar(100),  
  PRIMARY KEY ("Essn", "pname"),  
  CONSTRAINT "FK_Works On.Essn"  
  FOREIGN KEY ("Essn")  
  REFERENCES Employee("SSN"),  
  CONSTRAINT "FK_Works On.pname"  
  FOREIGN KEY ("pname")  
  REFERENCES Project("name")  
);
```

INSERT STATEMENTS

INSERT INTO Employee VALUES

```
(1, '2000-11-03', 'permanent', 'female', 'Lalitha Sravanti', 'Dasu',
'HSR Layout'),
(2, '2001-04-10', 'permanent', 'female', 'Meenakshi', 'Suresh',
'sector-5 HSR Layout, Bangalore'),
(3, '2001-12-26', 'permanent', 'male', 'Lohith', 'Srinivas',
'Electronic City, Bangalore'),
(4, '1983-03-20', 'permanent', 'male', 'Deep', 'Mehta', 'Bellandur,
Bangalore'),
(5, '2001-09-03', 'temporary', 'female', 'Satya', 'Rajan',
'Koramangala, Bangalore'),
(6, '1989-11-05', 'permanent', 'female', 'Sunaina', 'Agrawal',
'Koramangala 5th block, Bangalore'),
(7, '1991-03-10', 'permanent', 'male', 'Rahul', 'Mittal',
'Banashankari, Bangalore'),
(8, '1997-07-12', 'permanent', 'male', 'Sanjay', 'Dutt', 'Whitefield,
Bangalore'),
(9, '2000-09-07', 'temporary', 'male', 'Vinod', 'Narayan',
'Bommanahalli, Bangalore'),
(10, '1986-12-02', 'permanent', 'male', 'Neel', 'Roy', 'Bellandur,
Bangalore');
```

INSERT INTO Dependents VALUES

```
('Sunitha Mehta', 4, 'female', '1985-07-21', 'wife'),
('Riya Mehta', 4, 'female', '2012-03-20', 'daughter'),
('Anika Sen', 4, 'female', '1965-04-05', 'mother'),
('Raj Agrawal', 6, 'male', '1984-03-02', 'husband'),
('Baldev Das', 6, 'male', '1959-09-12', 'father'),
('Chitra Suresh', 2, 'female', '1974-04-20', 'mother'),
('Abhimanyu Roy', 10, 'male', '1960-12-20', 'father'),
('Neelam Roy', 10, 'female', '1969-02-21', 'mother');
```

INSERT INTO Leave_type VALUES

```
(1, 'Annual Leave', 20),
(2, 'Loss of Pay Leave', 90),
(3, 'Sick Leave', 15),
(4, 'Maternity Leave', 180),
(5, 'Paternity Leave', 160);
```

INSERT INTO Building VALUES

```
(1, 'Verizon Corporation Limited', 1000),
(2, 'Verizon Research Lab', 300);
```

INSERT INTO Department VALUES

```
('Product Development', 6),
('Human Resources', 4),
```

```
('Technical Support',7),  
( 'Sales Support', 8),  
( 'Product Research', 10);
```

```
INSERT INTO Project VALUES  
( 'Fingerprint Voting System', 40000),  
( 'Inventory Management', 50000),  
( 'Project Management Tool', 45000),  
( 'PDF Reader 2.1', 30000),  
( 'PDF Reader 2.1 Lite', 32000);
```

```
INSERT INTO Skill VALUES  
(1, 'Big Data Analysis', 'Hadoop, Spark'),  
(2, 'Project Planning', 'Trello'),  
(3, 'Quality Testing', 'Selenium'),  
(4, 'Database Systems', 'Oracle'),  
(5, 'Market Intelligence', 'Crunchbase'),  
(6, 'Integrated Circuit Design', 'Cadence'),  
(7, 'Object Oriented Design', 'C++'),  
(8, 'Programming', 'C++/Java'),  
(9, 'Written and Verbal Communication', 'None'),  
(10, 'Mentoring', 'None');
```

```
INSERT INTO Salary_Range VALUES  
(1, 4, 30000, 35000),  
(2, 5, 40000, 55000),  
(3, 6, 60000, 70000),  
(4, 7, 72000, 80000),  
(5, 8, 85000, 95000);
```

```
INSERT INTO houses VALUES  
(1, 'Product Development', 1),  
(2, 'Human Resources', 1),  
(3, 'Technical Support', 1),  
(4, 'Sales Support', 1),  
(5, 'Product Development', 2),  
(6, 'Technical Support', 2),  
(7, 'Product Research', 2);
```

```
INSERT INTO works_in VALUES  
(1, 1, 'Product Research'),  
(2, 2, 'Product Development'),  
(3, 3, 'Human Resources'),  
(4, 4, 'Human Resources'),  
(5, 5, 'Sales Support'),  
(6, 6, 'Product Development'),  
(7, 7, 'Technical Support'),  
(8, 8, 'Sales Support'),
```

```
(9, 9, 'Technical Support'),  
(10, 10, 'Product Research');
```

```
INSERT INTO works_on VALUES  
( '2020-08-10', 6, 1, 'Fingerprint Voting System'),  
( '2020-09-21', 6, 2, 'Fingerprint Voting System'),  
( '2019-04-22', 6, 6, 'Fingerprint Voting System'),  
( '2019-02-12', 7, 3, 'Fingerprint Voting System'),  
( '2020-11-03', 7, 5, 'Fingerprint Voting System'),  
( '2010-08-03', 7, 4, 'Inventory Management'),  
( '2015-03-17', 10, 7, 'Inventory Management'),  
( '2018-04-17', 5, 2, 'Inventory Management'),  
( '2017-11-21', 7, 3, 'Project Management Tool'),  
( '2018-08-03', 8, 4, 'Project Management Tool'),  
( '2017-02-02', 6, 6, 'Project Management Tool'),  
( '2018-09-09', 8, 9, 'Project Management Tool'),  
( '2021-01-21', 8, 5, 'PDF Reader 2.1'),  
( '2020-09-02', 6, 9, 'PDF Reader 2.1'),  
( '2017-08-21', 7, 10, 'PDF Reader 2.1'),  
( '2018-03-03', 8, 8, 'PDF Reader 2.1 Lite'),  
( '2018-06-01', 7, 1, 'PDF Reader 2.1 Lite'),  
( '2019-06-21', 8, 10, 'PDF Reader 2.1 Lite');
```

```
INSERT INTO Avails_leave VALUES  
(1, 1, 1),  
(2, 2, 1),  
(3, 3, 1),  
(4, 4, 1),  
(5, 5, 1),  
(6, 6, 1),  
(7, 7, 1),  
(8, 8, 1),  
(9, 9, 1),  
(10, 10, 1),  
(11, 6, 4),  
(12, 8, 2),  
(13, 4, 5),  
(14, 10, 5),  
(15, 4, 3),  
(16, 5, 3),  
(17, 8, 2);
```

```
INSERT INTO Has VALUES  
(1, 1, 7),  
(2, 1, 8),  
(3, 2, 4),  
(4, 2, 8),  
(5, 3, 2),
```

```
(6, 3, 9),
(7, 4, 9),
(8, 4, 10),
(9, 5, 9),
(10, 5, 3),
(11, 5, 5),
(12, 6, 10),
(13, 6, 1),
(14, 6, 6),
(15, 7, 8),
(16, 7, 9),
(18, 8, 5),
(19, 8, 9),
(20, 8, 10),
(21, 9, 8),
(22, 10, 10),
(23, 10, 6),
(24, 10, 8);
```

INSERT INTO assigned_to VALUES

```
(1, 'Fingerprint Voting System', 'Product Research'),
(2, 'Fingerprint Voting System', 'Product Development'),
(3, 'Fingerprint Voting System', 'Human Resources'),
(4, 'Fingerprint Voting System', 'Sales Support'),
(5, 'Inventory Management', 'Sales Support'),
(6, 'Inventory Management', 'Product Development'),
(7, 'Project Management Tool', 'Human Resources'),
(8, 'Project Management Tool', 'Product Development'),
(9, 'Project Management Tool', 'Technical Support'),
(10, 'PDF Reader 2.1', 'Sales Support'),
(11, 'PDF Reader 2.1', 'Technical Support'),
(12, 'PDF Reader 2.1', 'Product Research'),
(13, 'PDF Reader 2.1 Lite', 'Sales Support'),
(14, 'PDF Reader 2.1 Lite', 'Product Research');
```

INSERT INTO Earns VALUES

```
(1,3,1),
(2,3,2),
(3,2,3),
(4,5,4),
(5,3,5),
(6,5,6),
(7,4,7),
(8,4,8),
(9,1,9),
(10,5,10);
```

RUN THE *.sql* FILES TO CREATE RELATIONS AND INSERT VALUES

```
postgres=# \i /home/lalitha/Documents/misc/createTables.sql
DROP DATABASE
CREATE DATABASE
You are now connected to database "employeemanagement" as user "postgres".
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
employeemanagement=# \i /home/lalitha/Documents/misc/insert.sql
INSERT 0 10
INSERT 0 8
INSERT 0 5
INSERT 0 2
INSERT 0 5
INSERT 0 5
INSERT 0 10
INSERT 0 5
INSERT 0 7
INSERT 0 10
INSERT 0 18
INSERT 0 17
INSERT 0 23
INSERT 0 14
INSERT 0 10
```

PRINT LIST OF RELATIONSHIPS:

```
employee management=# \d
List of relations
Schema |      Name      | Type  | Owner
-----+-----+-----+-----
public | assigned_to    | table | postgres
public | avails_leave   | table | postgres
public | building       | table | postgres
public | department     | table | postgres
public | dependents     | table | postgres
public | earns          | table | postgres
public | employee       | table | postgres
public | has            | table | postgres
public | houses         | table | postgres
public | leave_type     | table | postgres
public | project        | table | postgres
public | salary_range   | table | postgres
public | skill          | table | postgres
public | works_in       | table | postgres
public | works_on       | table | postgres
(15 rows)
```

SCREENSHOT OF TABLE SCHEMA

Assigned_to ,Avails_leave, Building:

```
employeemanagement=# \d assigned_to;
                                Table "public.assigned_to"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
  id     | integer                |           | not null |
  pname  | character varying(100) |           |          |
  dname  | character varying(100) |           |          |
Indexes:
    "assigned_to_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "FK_assigned_to.dname" FOREIGN KEY (dname) REFERENCES department(name)
    "FK_assigned_to.pname" FOREIGN KEY (pname) REFERENCES project(name)

employeemanagement=# \d avails_leave;
                                Table "public.avails_leave"
  Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
  id     | integer |           | not null |
  essn   | integer |           |          |
  lid    | integer |           |          |
Indexes:
    "avails_leave_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "FK_Avails_leave.essn" FOREIGN KEY (essn) REFERENCES employee("SSN")
    "FK_Avails_leave.lid" FOREIGN KEY (lid) REFERENCES leave_type(id)

employeemanagement=# \d building;
                                Table "public.building"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
  id     | integer                |           | not null |
  name   | character varying(100) |           |          |
  capacity | integer                |           |          |
Indexes:
    "building_pkey" PRIMARY KEY, btree (id)
Referenced by:
    TABLE "houses" CONSTRAINT "FK_houses.bid" FOREIGN KEY (bid) REFERENCES building(id)
```


Dependents, Departments:

```
employeemanagement=# \d dependents;
               Table "public.dependents"
  Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----
 Name         | character varying(100) |           | not null |
 Essn         | integer          |           |          |
 Sex          | character varying(10) |           |          |
 b_date       | date             |           |          |
 relationship  | character varying(100) |           |          |
Indexes:
    "dependents_pkey" PRIMARY KEY, btree ("Name")
Foreign-key constraints:
    "FK_Dependents.Essn" FOREIGN KEY ("Essn") REFERENCES employee("SSN")

employeemanagement=# \d department;
               Table "public.department"
  Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----
 name         | character varying(100) |           | not null |
 Mssn         | integer          |           |          |
Indexes:
    "department_pkey" PRIMARY KEY, btree (name)
Foreign-key constraints:
    "FK_Department.Mssn" FOREIGN KEY ("Mssn") REFERENCES employee("SSN")
Referenced by:
    TABLE "assigned_to" CONSTRAINT "FK_assigned_to.dname" FOREIGN KEY (dname) REFERENCES department(name)
    TABLE "houses" CONSTRAINT "FK_houses.dname" FOREIGN KEY (dname) REFERENCES department(name)
    TABLE "works_in" CONSTRAINT "FK_works_in.dname" FOREIGN KEY (dname) REFERENCES department(name)
```

Earns:

```
employeemanagement=# \d earns;
               Table "public.earns"
  Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----
 id           | integer          |           | not null |
 sid          | integer          |           |          |
 essn         | integer          |           |          |
Indexes:
    "earns_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "FK_Earns.essn" FOREIGN KEY (essn) REFERENCES employee("SSN")
    "FK_Earns.sid" FOREIGN KEY (sid) REFERENCES salary_range(id)
```

Employee,Has:

```
employeemanagement=# \d employee;
               Table "public.employee"
  Column      |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
SSN            | integer                |           | not null |
b_date         | date                   |           |          |
Status         | character varying(100) |           |          |
sex            | character varying(10)  |           |          |
s-name         | character varying(100) |           |          |
l-name         | character varying(100) |           |          |
Address        | character varying(500) |           |          |
super_ssn      | integer                |           |          |
Indexes:
    "employee_pkey" PRIMARY KEY, btree ("SSN")
Foreign-key constraints:
    "FK_Employee.super_ssn" FOREIGN KEY (super_ssn) REFERENCES employee("SSN")
Referenced by:
    TABLE "avails_leave" CONSTRAINT "FK_Avails_leave.essn" FOREIGN KEY (essn) REFERENCES employee("SSN")
    TABLE "department" CONSTRAINT "FK_Department.Mssn" FOREIGN KEY ("Mssn") REFERENCES employee("SSN")
    TABLE "dependents" CONSTRAINT "FK_Dependents.Essn" FOREIGN KEY ("Essn") REFERENCES employee("SSN")
    TABLE "earns" CONSTRAINT "FK_Earns.essn" FOREIGN KEY (essn) REFERENCES employee("SSN")
    TABLE "employee" CONSTRAINT "FK_Employee.super_ssn" FOREIGN KEY (super_ssn) REFERENCES employee("SSN")
    TABLE "has" CONSTRAINT "FK_Skill.essn" FOREIGN KEY (essn) REFERENCES employee("SSN")
    TABLE "works_on" CONSTRAINT "FK_Works On.Essn" FOREIGN KEY ("Essn") REFERENCES employee("SSN")
    TABLE "works_in" CONSTRAINT "FK_works_in.essn" FOREIGN KEY (essn) REFERENCES employee("SSN")

employeemanagement=# \d has;
               Table "public.has"
  Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
id       | integer |           | not null |
essn     | integer |           |          |
sid      | integer |           |          |
Indexes:
    "has_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "FK_Skill.essn" FOREIGN KEY (essn) REFERENCES employee("SSN")
    "FK_Skill.sid" FOREIGN KEY (sid) REFERENCES skill(id)
```

Houses,Leave_type,Project:

```
employeemanagement=# \d houses;
               Table "public.houses"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
id       | integer                |           | not null |
dname    | character varying(100) |           |          |
bid      | integer                |           |          |
Indexes:
    "houses_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "FK_houses.bid" FOREIGN KEY (bid) REFERENCES building(id)
    "FK_houses.dname" FOREIGN KEY (dname) REFERENCES department(name)

employeemanagement=# \d leave_type;
               Table "public.leave_type"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
id       | integer                |           | not null |
Type     | character varying(50)  |           |          |
NoOfDays | integer                |           |          |
Essn     | integer                |           |          |
Indexes:
    "leave_type_pkey" PRIMARY KEY, btree (id)
Referenced by:
    TABLE "avails_leave" CONSTRAINT "FK_Avails_leave.lid" FOREIGN KEY (lid) REFERENCES leave_type(id)

employeemanagement=# \d project;
               Table "public.project"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
name     | character varying(100) |           | not null |
budget   | integer                |           |          |
Indexes:
    "project_pkey" PRIMARY KEY, btree (name)
Referenced by:
    TABLE "works_on" CONSTRAINT "FK_Works On.pname" FOREIGN KEY (pname) REFERENCES project(name)
    TABLE "assigned_to" CONSTRAINT "FK_assigned_to.pname" FOREIGN KEY (pname) REFERENCES project(name)
```

Salary_range, Skill, Works_in:

```
employee management=# \d salary_range;
          Table "public.salary_range"
  Column |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
id       | integer        |           | not null |
grade    | integer        |           |          |
start_sal | integer        |           |          |
end_sal   | integer        |           |          |
Indexes:
    "salary_range_pkey" PRIMARY KEY, btree (id)
Referenced by:
    TABLE "earns" CONSTRAINT "FK_Earns.sid" FOREIGN KEY (sid) REFERENCES salary_range(id)

employee management=# \d skill;
          Table "public.skill"
  Column |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
id       | integer        |           | not null |
name     | character varying(100) |           |          |
tech     | character varying(100) |           |          |
Indexes:
    "skill_pkey" PRIMARY KEY, btree (id)
Referenced by:
    TABLE "has" CONSTRAINT "FK_Skill.sid" FOREIGN KEY (sid) REFERENCES skill(id)

employee management=# \d works_in;
          Table "public.works_in"
  Column |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
id       | integer        |           | not null |
essn     | integer        |           |          |
dname    | character varying(100) |           |          |
Indexes:
    "works_in_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "FK_works_in.dname" FOREIGN KEY (dname) REFERENCES department(name)
    "FK_works_in.essn" FOREIGN KEY (essn) REFERENCES employee("SSN")
```

Works_on:

```
employee management=# \d works_on;
          Table "public.works_on"
  Column |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
Start_date | date          |           |          |
Hours      | integer       |           |          |
Essn       | integer       |           | not null |
pname      | character varying(100) |           | not null |
Indexes:
    "works_on_pkey" PRIMARY KEY, btree ("Essn", pname)
Foreign-key constraints:
    "FK_Works On.Essn" FOREIGN KEY ("Essn") REFERENCES employee("SSN")
    "FK_Works On.pname" FOREIGN KEY (pname) REFERENCES project(name)
```

TABLES WITH VALUES INSERTED

Assigned_to:

```
employeemanagement=# SELECT * FROM assigned_to;
 id |                pname                |      dname
-----+-----+-----
  1 | Fingerprint Voting System | Product Research
  2 | Fingerprint Voting System | Product Development
  3 | Fingerprint Voting System | Human Resources
  4 | Fingerprint Voting System | Sales Support
  5 | Inventory Management      | Sales Support
  6 | Inventory Management      | Product Development
  7 | Project Management Tool    | Human Resources
  8 | Project Management Tool    | Product Development
  9 | Project Management Tool    | Technical Support
 10 | PDF Reader 2.1            | Sales Support
 11 | PDF Reader 2.1            | Technical Support
 12 | PDF Reader 2.1            | Product Research
 13 | PDF Reader 2.1 Lite       | Sales Support
 14 | PDF Reader 2.1 Lite       | Product Research
(14 rows)
```

Avails_leave,Building:

```
employeemanagement=# SELECT * FROM avails_leave;
 id | essn | lid
-----+-----+-----
  1 |    1 |    1
  2 |    2 |    1
  3 |    3 |    1
  4 |    4 |    1
  5 |    5 |    1
  6 |    6 |    1
  7 |    7 |    1
  8 |    8 |    1
  9 |    9 |    1
 10 |   10 |    1
 11 |    6 |    4
 12 |    8 |    2
 13 |    4 |    5
 14 |   10 |    5
 15 |    4 |    3
 16 |    5 |    3
 17 |    8 |    2
(17 rows)

employeemanagement=# SELECT * FROM building;
 id |                name                | capacity
-----+-----+-----
  1 | Verizon Corporation Limited |    1000
  2 | Verizon Research Lab       |     300
(2 rows)
```

Department:

```
employee management=# SELECT * FROM department;
      name      | Mssn
-----+-----
Product Development | 6
Human Resources   | 4
Technical Support | 7
Sales Support     | 8
Product Research  | 10
(5 rows)
```

Dependents, Earns:

```
employee management=# SELECT * FROM dependents;
      Name      | Essn | Sex | b_date | relationship
-----+-----+-----+-----+-----
Sunitha Mehta | 4 | female | 1985-07-21 | wife
Riya Mehta    | 4 | female | 2012-03-20 | daughter
Anika Sen     | 4 | female | 1965-04-05 | mother
Raj Agrawal   | 6 | male   | 1984-03-02 | husband
Baldev Das    | 6 | male   | 1959-09-12 | father
Chitra Suresh | 2 | female | 1974-04-20 | mother
Abhimanyu Roy | 10 | male   | 1960-12-20 | father
Neelam Roy    | 10 | female | 1969-02-21 | mother
(8 rows)
```

```
employee management=# SELECT * FROM earns;
 id | sid | essn
----+----+-----
  1 |  3 | 1
  2 |  3 | 2
  3 |  2 | 3
  4 |  5 | 4
  5 |  3 | 5
  6 |  5 | 6
  7 |  4 | 7
  8 |  4 | 8
  9 |  1 | 9
 10 |  5 | 10
(10 rows)
```

Employee:

```
employeemanagement=# SELECT * FROM employee;
SSN | b_date | Status | sex | s-name | l-name | Address | super_ssn
-----+-----+-----+-----+-----+-----+-----+-----
 1 | 2000-11-03 | permanent | female | Lalitha Sravanti | Dasu | HSR Layout | 
 2 | 2001-04-10 | permanent | female | Meenakshi | Suresh | sector-5 HSR Layout, Bangalore | 
 3 | 2001-12-26 | permanent | male | Lohith | Srinivas | Electronic City, Bangalore | 
 4 | 1983-03-20 | permanent | male | Deep | Mehta | Bellandur, Bangalore | 
 5 | 2001-09-03 | temporary | female | Satya | Rajan | Koramangala, Bangalore | 
 6 | 1989-11-05 | permanent | female | Sunaina | Agrawal | Koramangala 5th block, Bangalore | 
 7 | 1991-03-10 | permanent | male | Rahul | Mittal | Banashankari, Bangalore | 
 8 | 1997-07-12 | permanent | male | Sanjay | Dutt | Whitefield, Bangalore | 
 9 | 2000-09-07 | temporary | male | Vinod | Narayan | Bommanahalli, Bangalore | 
10 | 1986-12-02 | permanent | male | Neel | Roy | Bellandur, Bangalore | 
(10 rows)
```

Has:

```
employeemanagement=# SELECT * FROM has;
id | essn | sid
-----+-----+-----
 1 | 1 | 7
 2 | 1 | 8
 3 | 2 | 4
 4 | 2 | 8
 5 | 3 | 2
 6 | 3 | 9
 7 | 4 | 9
 8 | 4 | 10
 9 | 5 | 9
10 | 5 | 3
11 | 5 | 5
12 | 6 | 10
13 | 6 | 1
14 | 6 | 6
15 | 7 | 8
16 | 7 | 9
18 | 8 | 5
19 | 8 | 9
20 | 8 | 10
21 | 9 | 8
22 | 10 | 10
23 | 10 | 6
24 | 10 | 8
(23 rows)
```

Houses,Leave_type,Project:

```
employeemanagement=# SELECT * FROM houses;
id |          dname          | bid
----+-----+-----
  1 | Product Development    |  1
  2 | Human Resources        |  1
  3 | Technical Support      |  1
  4 | Sales Support          |  1
  5 | Product Development    |  2
  6 | Technical Support      |  2
  7 | Product Research       |  2
(7 rows)

employeemanagement=# SELECT * FROM leave_type;
id |      Type      | NoOfDays | Essn
----+-----+-----+-----
  1 | Annual Leave   |      20 |
  2 | Loss of Pay Leave |      90 |
  3 | Sick Leave     |      15 |
  4 | Maternity Leave |     180 |
  5 | Paternity Leave |     160 |
(5 rows)

employeemanagement=# SELECT * FROM project;
          name          | budget
-----+-----
Fingerprint Voting System | 40000
Inventory Management      | 50000
Project Management Tool   | 45000
PDF Reader 2.1            | 30000
PDF Reader 2.1 Lite       | 32000
(5 rows)
```

Salary_range, Skill:

```
employeemanagement=# SELECT * FROM salary_range;
id | grade | start_sal | end_sal
----+-----+-----+-----
  1 |    4  |   30000  |   35000
  2 |    5  |   40000  |   55000
  3 |    6  |   60000  |   70000
  4 |    7  |   72000  |   80000
  5 |    8  |   85000  |   95000
(5 rows)

employeemanagement=# SELECT * FROM skill;
id |          name          | tech
----+-----+-----
  1 | Big Data Analysis     | Hadoop, Spark
  2 | Project Planning       | Trello
  3 | Quality Testing        | Selenium
  4 | Database Systems      | Oracle
  5 | Market Intelligence    | Crunchbase
  6 | Integrated Circuit Design | Cadence
  7 | Object Oriented Design | C++
  8 | Programming            | C++/Java
  9 | Written and Verbal Communication | None
 10 | Mentoring              | None
(10 rows)
```

Works_in, Works_on:

```
employeemanagement=# SELECT * FROM works_in;
```

id	essn	dname
1	1	Product Research
2	2	Product Development
3	3	Human Resources
4	4	Human Resources
5	5	Sales Support
6	6	Product Development
7	7	Technical Support
8	8	Sales Support
9	9	Technical Support
10	10	Product Research

(10 rows)

```
employeemanagement=# SELECT * FROM works_on;
```

Start_date	Hours	Essn	pname
2020-08-10	6	1	Fingerprint Voting System
2020-09-21	6	2	Fingerprint Voting System
2019-04-22	6	6	Fingerprint Voting System
2019-02-12	7	3	Fingerprint Voting System
2020-11-03	7	5	Fingerprint Voting System
2010-08-03	7	4	Inventory Management
2015-03-17	10	7	Inventory Management
2018-04-17	5	2	Inventory Management
2017-11-21	7	3	Project Management Tool
2018-08-03	8	4	Project Management Tool
2017-02-02	6	6	Project Management Tool
2018-09-09	8	9	Project Management Tool
2021-01-21	8	5	PDF Reader 2.1
2020-09-02	6	9	PDF Reader 2.1
2017-08-21	7	10	PDF Reader 2.1
2018-03-03	8	8	PDF Reader 2.1 Lite
2018-06-01	7	1	PDF Reader 2.1 Lite
2019-06-21	8	10	PDF Reader 2.1 Lite

(18 rows)

CONTRIBUTIONS:

NAME	SRN	CONTRIBUTION
LOHITH SRINIVAS T	PES2UG19CS203	Er diagram to relational schema conversion. Preparation of report. (6 HOURS)
LALITHA SRAVANTI DASU	PES2UG19CS201	Create statements in a SQL file. Insert statements in SQL file. (8 HOURS)
MEENAKSHI SURESH	PES2UG19CS228	Create statements in a SQL file. Insert statements in SQL file. (8 HOURS)

Note: *createTables.sql* and *insert.sql* files attached on teams.