PostgreSQL Assignment-3

BY Manish Yadav

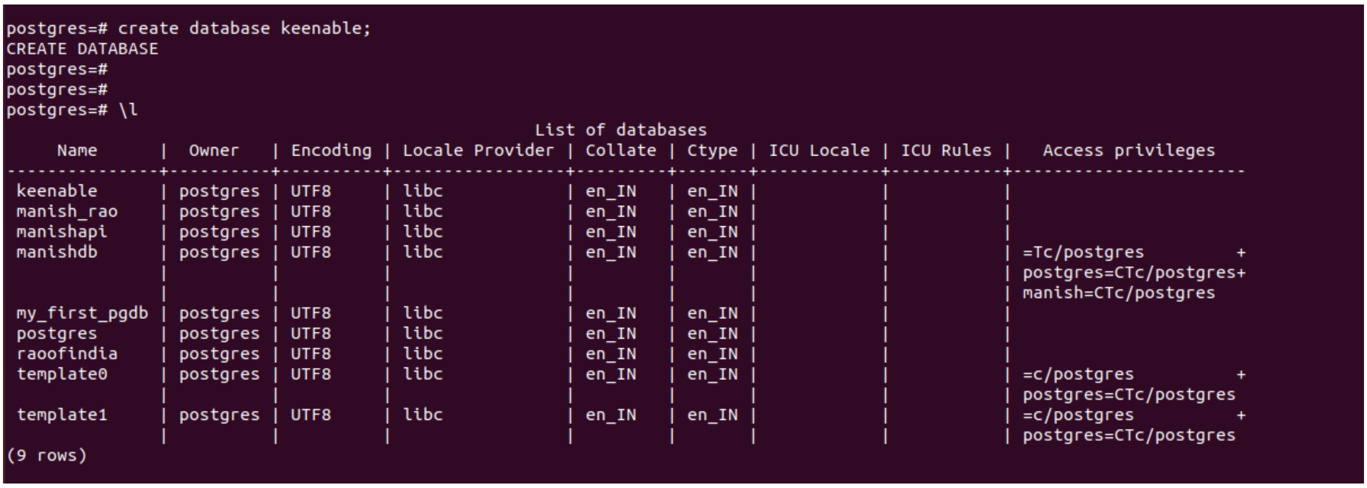
1. Create a new PostgreSQL database named "keenable".

postgres=# create database keenable;

Here's a short explanation of the command:

```
create database: This is the SQL command for creating a new database.
keenable: This is the name of the database being created. You can replace
"keenable" with the desired name for your database.
```

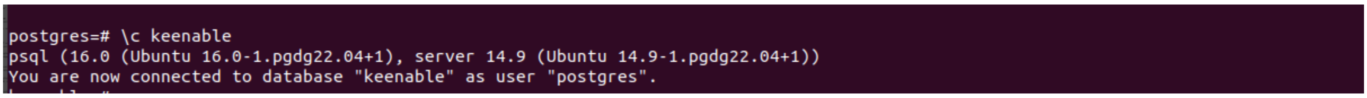
This command is essential for initializing a new database within your PostgreSQL server.



```
postgres=# create database keenable;
CREATE DATABASE
postgres=#
postgres=#
postgres=# \l
                                                       List of databases
     Name      |  Owner   | Encoding | Locale Provider | Collate | Ctype | ICU Locale | ICU Rules |   Access privileges
---------------+----------+----------+-----------------+---------+-------+------------+-----------+-----------------------
 keenable      | postgres | UTF8     | libc            | en_IN   | en_IN |            |           |
 manish_rao    | postgres | UTF8     | libc            | en_IN   | en_IN |            |           |
 manishapi     | postgres | UTF8     | libc            | en_IN   | en_IN |            |           |
 manishdb      | postgres | UTF8     | libc            | en_IN   | en_IN |            |           | =Tc/postgres         +
               |          |          |                 |         |       |            |           | postgres=CTc/postgres+
               |          |          |                 |         |       |            |           | manish=CTc/postgres
 my_first_pgdb | postgres | UTF8     | libc            | en_IN   | en_IN |            |           |
 postgres      | postgres | UTF8     | libc            | en_IN   | en_IN |            |           |
 raoofindia    | postgres | UTF8     | libc            | en_IN   | en_IN |            |           |
 template0     | postgres | UTF8     | libc            | en_IN   | en_IN |            |           | =c/postgres          +
               |          |          |                 |         |       |            |           | postgres=CTc/postgres
 template1     | postgres | UTF8     | libc            | en_IN   | en_IN |            |           | =c/postgres          +
               |          |          |                 |         |       |            |           | postgres=CTc/postgres
(9 rows)
```

2. Define three schemas within the database: "hr", "technical", and "management".

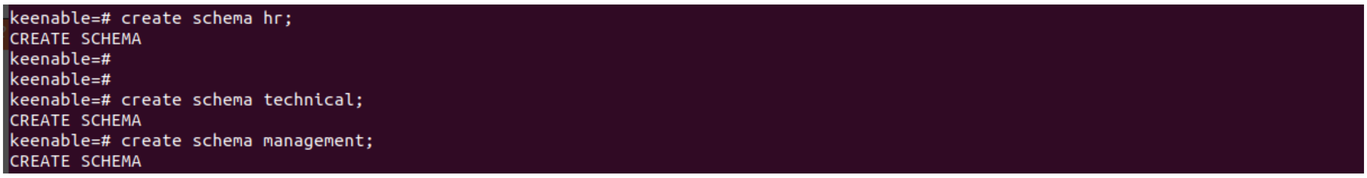
postgres=# \c keenable



```
postgres=# \c keenable
psql (16.0 (Ubuntu 16.0-1.pgdg22.04+1), server 14.9 (Ubuntu 14.9-1.pgdg22.04+1))
You are now connected to database "keenable" as user "postgres".
```

It's a way to change current database context to "keenable" for running SQL queries and operations within that database.

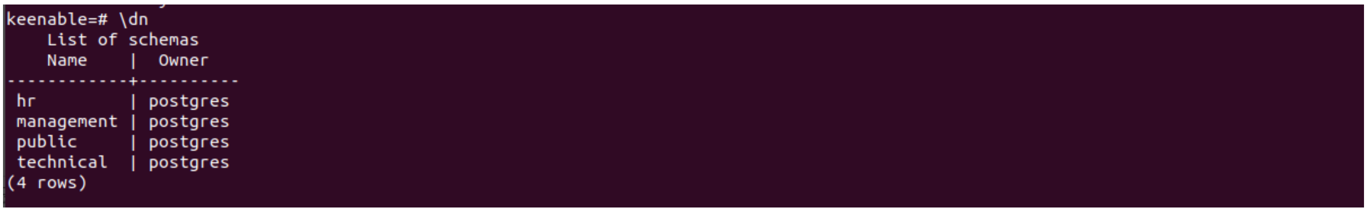
A. keenable=# create schema hr;
B. keenable=# create schema technical;
C. keenable=# create schema management;



```
keenable=# create schema hr;
CREATE SCHEMA
keenable=#
keenable=#
keenable=# create schema technical;
CREATE SCHEMA
keenable=# create schema management;
CREATE SCHEMA
```

This query creates a schema named "hr","technical","management" in the "keenable" database.

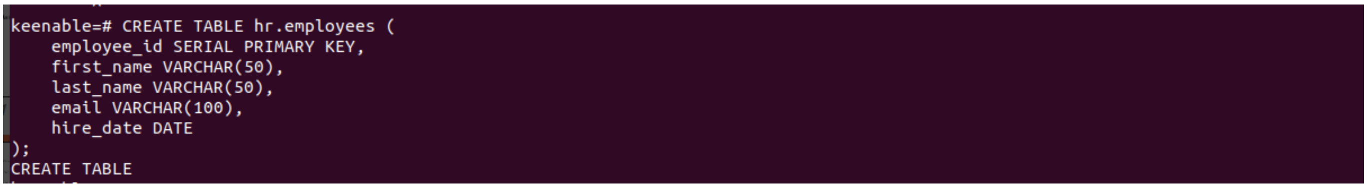
keenable=# \dn



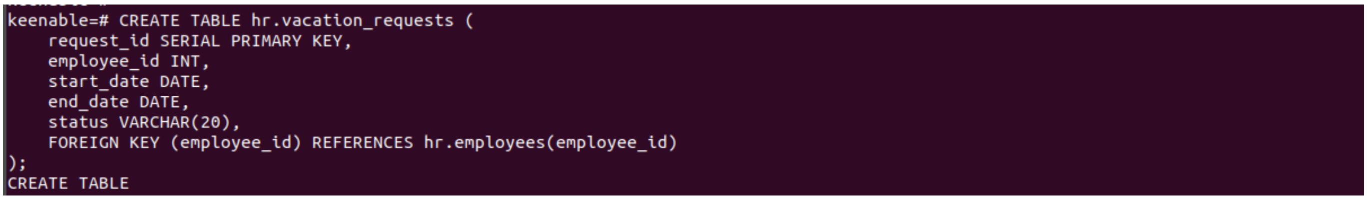
```
keenable=# \dn
    List of schemas
    Name    |  Owner
------------+----------
 hr         | postgres
 management | postgres
 public     | postgres
 technical  | postgres
(4 rows)
```

The command \dn in PostgreSQL is used to list all the schemas in the current database.

2.1 HR Schema:



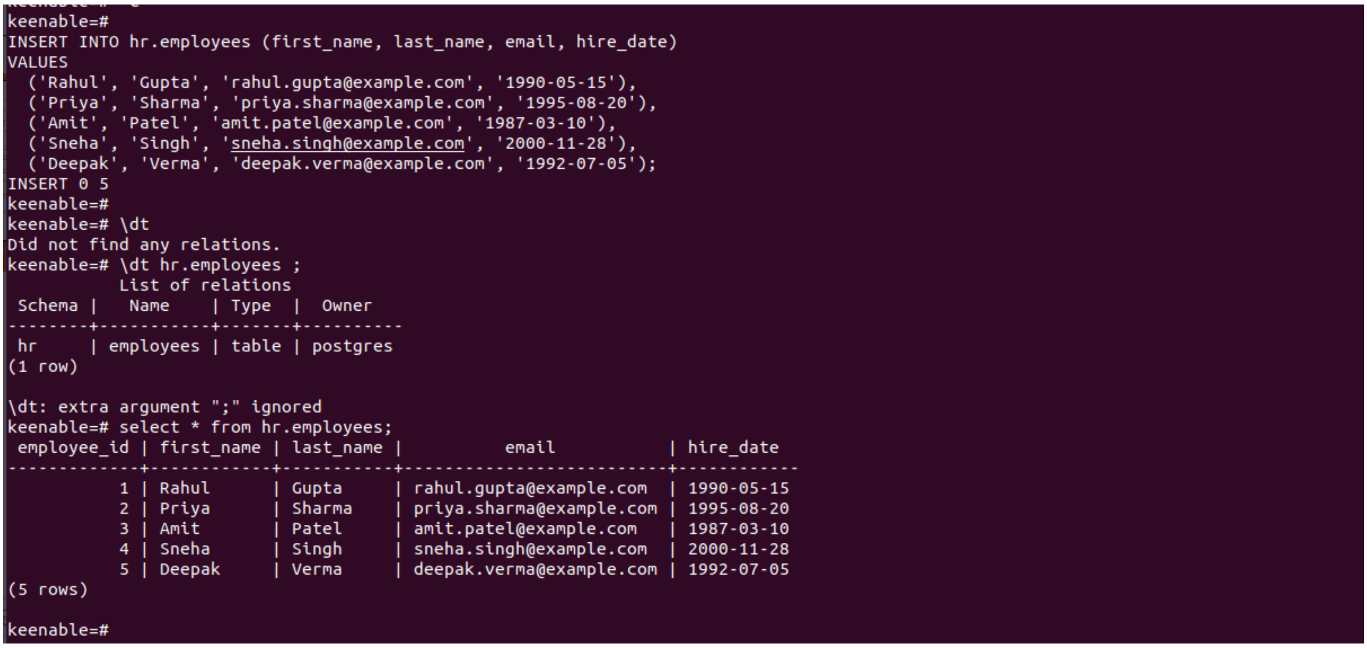
```
keenable=# CREATE TABLE hr.employees (
    employee_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    hire_date DATE
);
CREATE TABLE
```



```
keenable=# CREATE TABLE hr.vacation_requests (
    request_id SERIAL PRIMARY KEY,
    employee_id INT,
    start_date DATE,
    end_date DATE,
    status VARCHAR(20),
    FOREIGN KEY (employee_id) REFERENCES hr.employees(employee_id)
);
CREATE TABLE
```

Create the following tables in the "hr" schema:

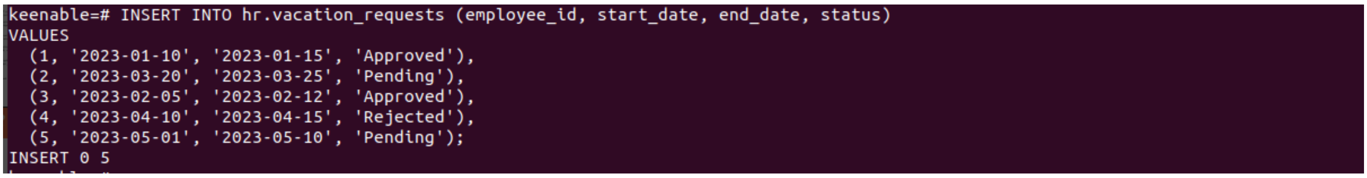
2.1.1.



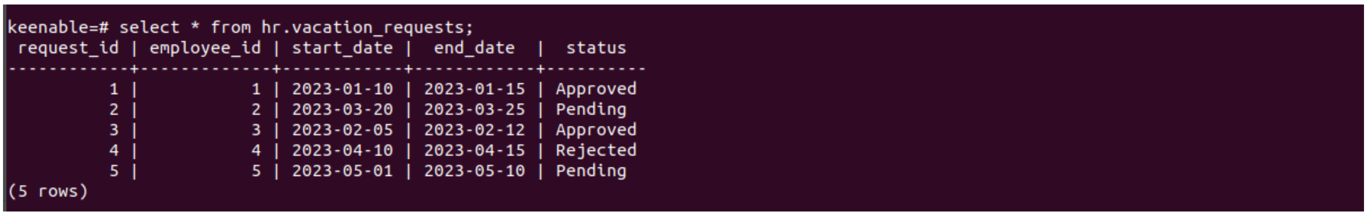
```
keenable=#
INSERT INTO hr.employees (first_name, last_name, email, hire_date)
VALUES
  ('Rahul', 'Gupta', 'rahul.gupta@example.com', '1990-05-15'),
  ('Priya', 'Sharma', 'priya.sharma@example.com', '1995-08-20'),
  ('Amit', 'Patel', 'amit.patel@example.com', '1987-03-10'),
  ('Sneha', 'Singh', 'sneha.singh@example.com', '2000-11-28'),
  ('Deepak', 'Verma', 'deepak.verma@example.com', '1992-07-05');
INSERT 0 5
keenable=#
keenable=# \dt
Did not find any relations.
keenable=# \dt hr.employees ;
           List of relations
 Schema |   Name    | Type  |  Owner
--------+-----------+-------+----------
 hr     | employees | table | postgres
(1 row)

\dt: extra argument ";" ignored
keenable=# select * from hr.employees;
 employee_id | first_name | last_name |          email           | hire_date
-------------+------------+-----------+--------------------------+------------
           1 | Rahul      | Gupta     | rahul.gupta@example.com  | 1990-05-15
           2 | Priya      | Sharma    | priya.sharma@example.com | 1995-08-20
           3 | Amit       | Patel     | amit.patel@example.com   | 1987-03-10
           4 | Sneha      | Singh     | sneha.singh@example.com  | 2000-11-28
           5 | Deepak     | Verma     | deepak.verma@example.com | 1992-07-05
(5 rows)

keenable=#
```

"employees" to store employee information (employee_id, first_name,last_name, email, hire_date).

2.1.2.



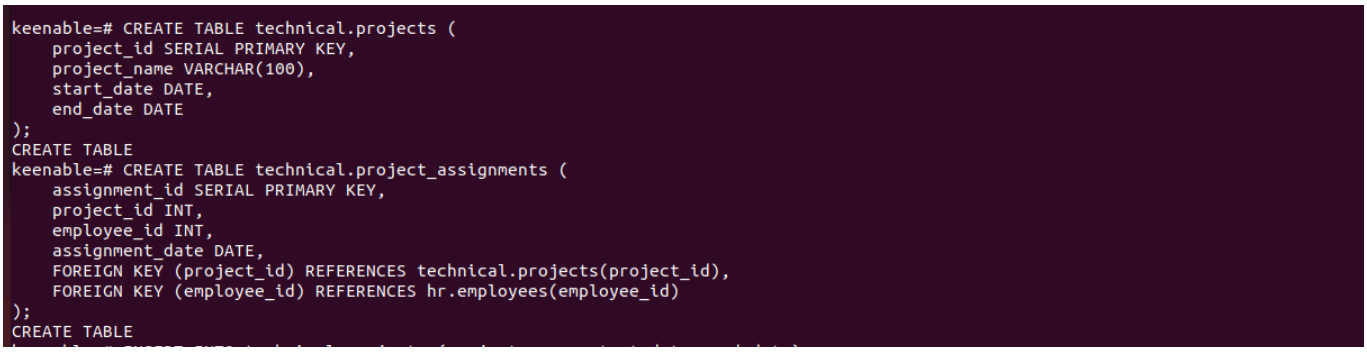
```
keenable=# INSERT INTO hr.vacation_requests (employee_id, start_date, end_date, status)
VALUES
  (1, '2023-01-10', '2023-01-15', 'Approved'),
  (2, '2023-03-20', '2023-03-25', 'Pending'),
  (3, '2023-02-05', '2023-02-12', 'Approved'),
  (4, '2023-04-10', '2023-04-15', 'Rejected'),
  (5, '2023-05-01', '2023-05-10', 'Pending');
INSERT 0 5
```



```
keenable=# select * from hr.vacation_requests;
 request_id | employee_id | start_date |  end_date  |  status
------------+-------------+------------+------------+----------
          1 |           1 | 2023-01-10 | 2023-01-15 | Approved
          2 |           2 | 2023-03-20 | 2023-03-25 | Pending
          3 |           3 | 2023-02-05 | 2023-02-12 | Approved
          4 |           4 | 2023-04-10 | 2023-04-15 | Rejected
          5 |           5 | 2023-05-01 | 2023-05-10 | Pending
(5 rows)
```

"vacation_requests" to track employee vacation requests (request_id,employee_id, start_date, end_date, status).

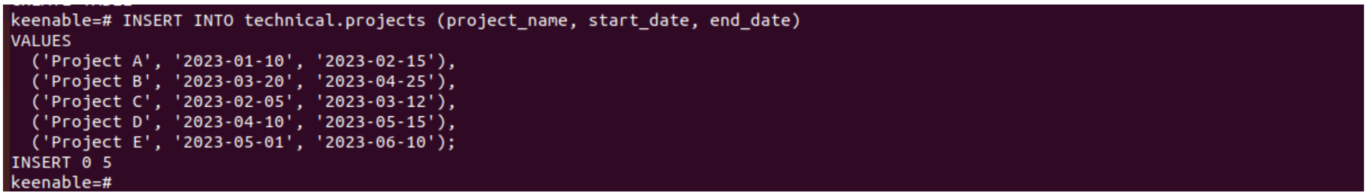
2.2. Technical Schema:



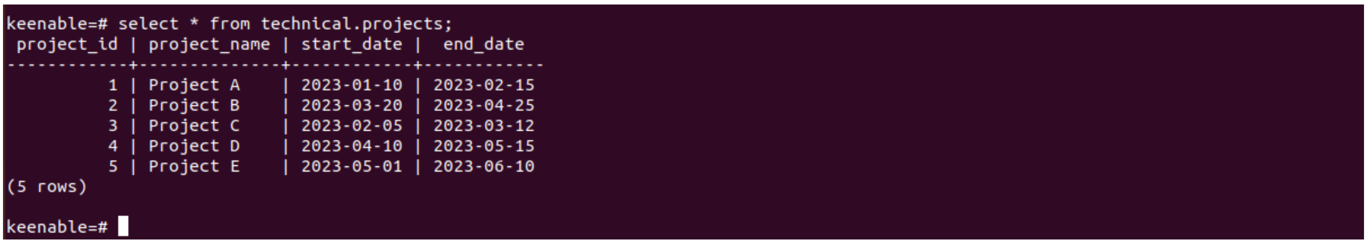
```
keenable=# CREATE TABLE technical.projects (
    project_id SERIAL PRIMARY KEY,
    project_name VARCHAR(100),
    start_date DATE,
    end_date DATE
);
CREATE TABLE
keenable=# CREATE TABLE technical.project_assignments (
    assignment_id SERIAL PRIMARY KEY,
    project_id INT,
    employee_id INT,
    assignment_date DATE,
    FOREIGN KEY (project_id) REFERENCES technical.projects(project_id),
    FOREIGN KEY (employee_id) REFERENCES hr.employees(employee_id)
);
CREATE TABLE
```

Create the following tables in the "technical" schema:

2.2.1.



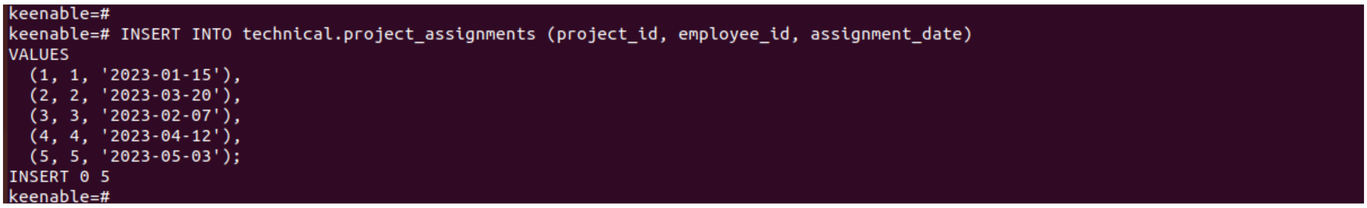
```
keenable=# INSERT INTO technical.projects (project_name, start_date, end_date)
VALUES
  ('Project A', '2023-01-10', '2023-02-15'),
  ('Project B', '2023-03-20', '2023-04-25'),
  ('Project C', '2023-02-05', '2023-03-12'),
  ('Project D', '2023-04-10', '2023-05-15'),
  ('Project E', '2023-05-01', '2023-06-10');
INSERT 0 5
keenable=#
```



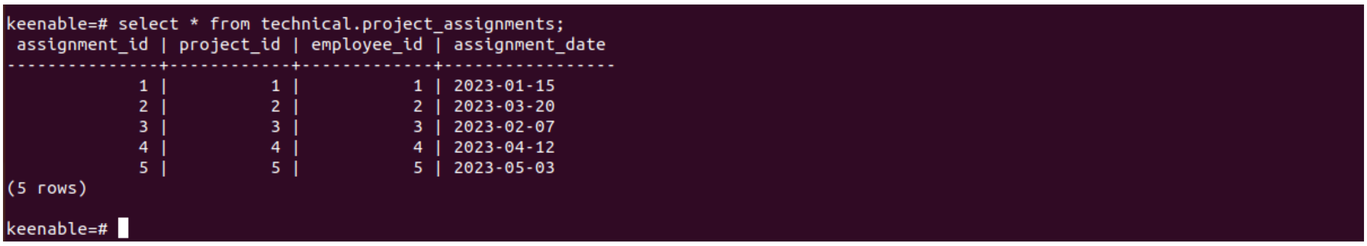
```
keenable=# select * from technical.projects;
 project_id | project_name | start_date |  end_date
------------+--------------+------------+------------
          1 | Project A    | 2023-01-10 | 2023-02-15
          2 | Project B    | 2023-03-20 | 2023-04-25
          3 | Project C    | 2023-02-05 | 2023-03-12
          4 | Project D    | 2023-04-10 | 2023-05-15
          5 | Project E    | 2023-05-01 | 2023-06-10
(5 rows)

keenable=#
```

"projects" to store project details (project_id, project_name, start_date,end_date).

2.2.2.



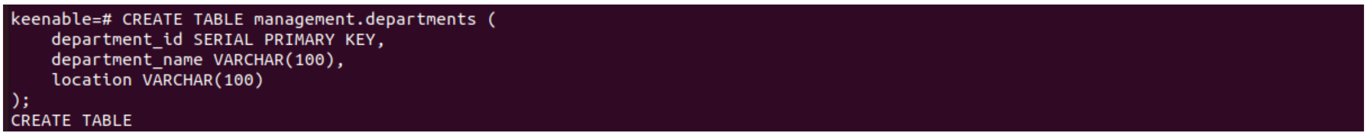
```
keenable=#
keenable=# INSERT INTO technical.project_assignments (project_id, employee_id, assignment_date)
VALUES
  (1, 1, '2023-01-15'),
  (2, 2, '2023-03-20'),
  (3, 3, '2023-02-07'),
  (4, 4, '2023-04-12'),
  (5, 5, '2023-05-03');
INSERT 0 5
keenable=#
```



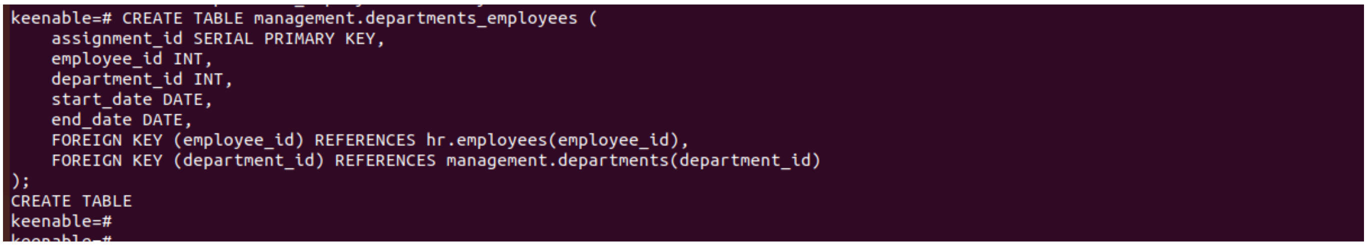
```
keenable=# select * from technical.project_assignments;
 assignment_id | project_id | employee_id | assignment_date
---------------+------------+-------------+-----------------
             1 |          1 |           1 | 2023-01-15
             2 |          2 |           2 | 2023-03-20
             3 |          3 |           3 | 2023-02-07
             4 |          4 |           4 | 2023-04-12
             5 |          5 |           5 | 2023-05-03
(5 rows)

keenable=# 
```

"project_assignments" to associate employees with projects (assignment_id, project_id, employee_id, assignment_date).

2.3. Management Schema:



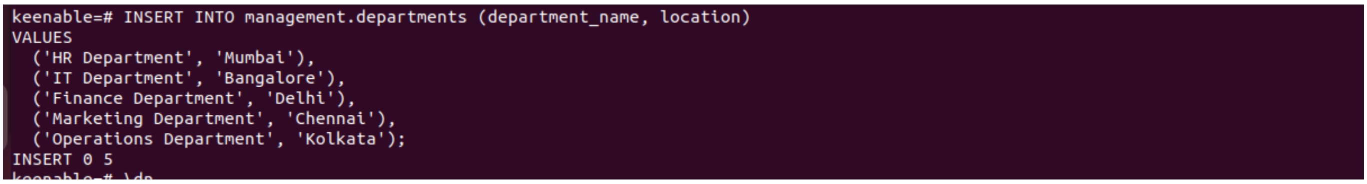
```
keenable=# CREATE TABLE management.departments (
    department_id SERIAL PRIMARY KEY,
    department_name VARCHAR(100),
    location VARCHAR(100)
);
CREATE TABLE
```



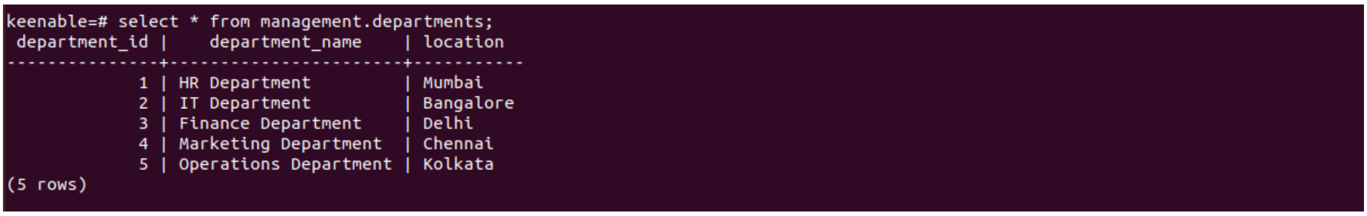
```
keenable=# CREATE TABLE management.departments_employees (
    assignment_id SERIAL PRIMARY KEY,
    employee_id INT,
    department_id INT,
    start_date DATE,
    end_date DATE,
    FOREIGN KEY (employee_id) REFERENCES hr.employees(employee_id),
    FOREIGN KEY (department_id) REFERENCES management.departments(department_id)
);
CREATE TABLE
keenable=#
keenable=#
```

Create the following tables in the "management" schema:

2.3.1.



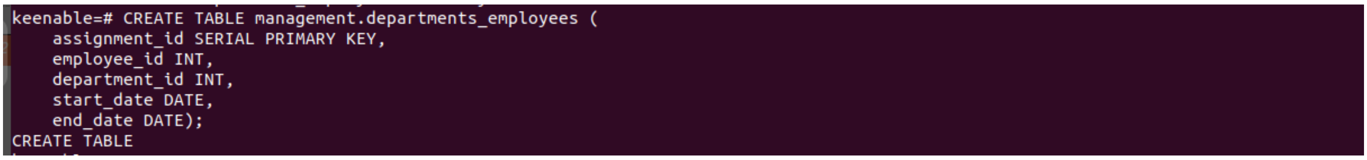
```
keenable=# INSERT INTO management.departments (department_name, location)
VALUES
  ('HR Department', 'Mumbai'),
  ('IT Department', 'Bangalore'),
  ('Finance Department', 'Delhi'),
  ('Marketing Department', 'Chennai'),
  ('Operations Department', 'Kolkata');
INSERT 0 5
keenable=# \dn
```



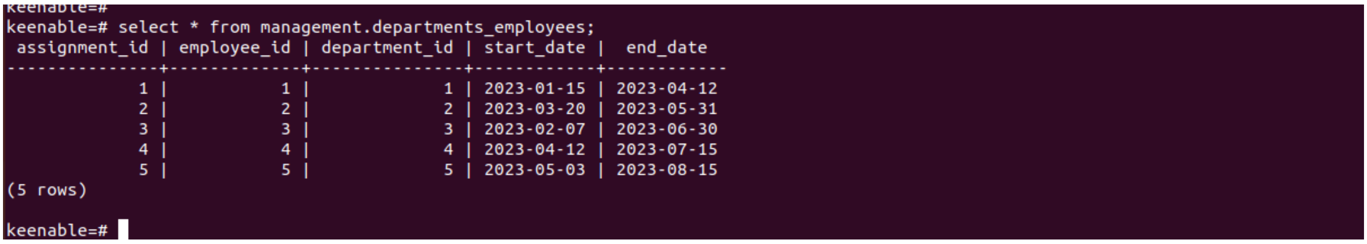
```
keenable=# select * from management.departments;
 department_id |    department_name    | location
---------------+-----------------------+-----------
             1 | HR Department         | Mumbai
             2 | IT Department         | Bangalore
             3 | Finance Department    | Delhi
             4 | Marketing Department  | Chennai
             5 | Operations Department | Kolkata
(5 rows)
```

"departments" to manage department information (department_id,department_name, location).

2.3.2.



```
keenable=# CREATE TABLE management.departments_employees (
    assignment_id SERIAL PRIMARY KEY,
    employee_id INT,
    department_id INT,
    start_date DATE,
    end_date DATE);
CREATE TABLE
```



```
keenable=#
keenable=# select * from management.departments_employees;
 assignment_id | employee_id | department_id | start_date |  end_date
---------------+-------------+---------------+------------+------------
             1 |           1 |             1 | 2023-01-15 | 2023-04-12
             2 |           2 |             2 | 2023-03-20 | 2023-05-31
             3 |           3 |             3 | 2023-02-07 | 2023-06-30
             4 |           4 |             4 | 2023-04-12 | 2023-07-15
             5 |           5 |             5 | 2023-05-03 | 2023-08-15
(5 rows)

keenable=#
```

"department_employees" to track employee department assignments (assignment_id, employee_id, department_id, start_date, end_date).

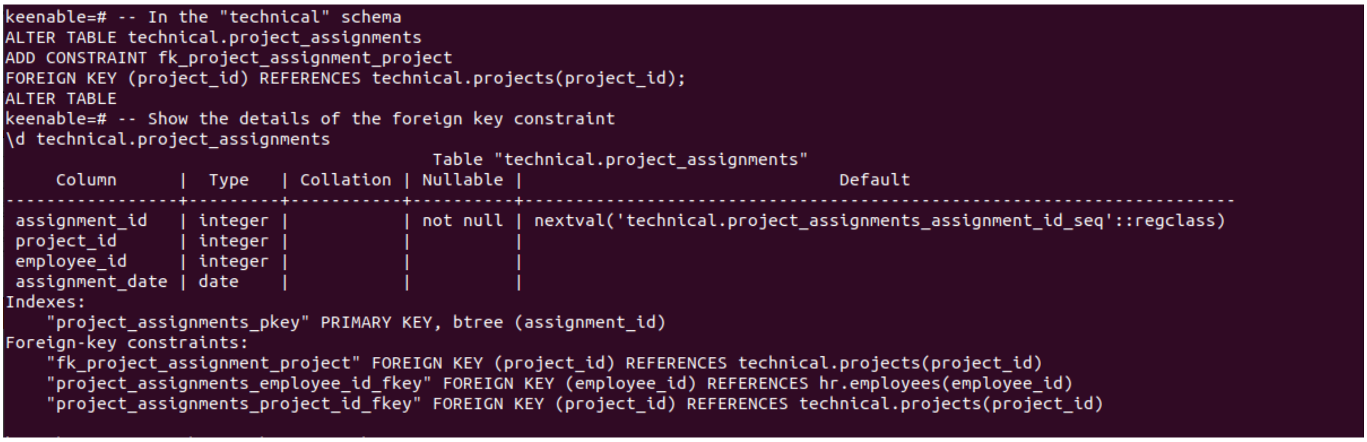
3. Relationships:

Establish appropriate foreign key relationships between tables to maintain referential integrity. For example, link "project_assignments" to "projects" and "employees."

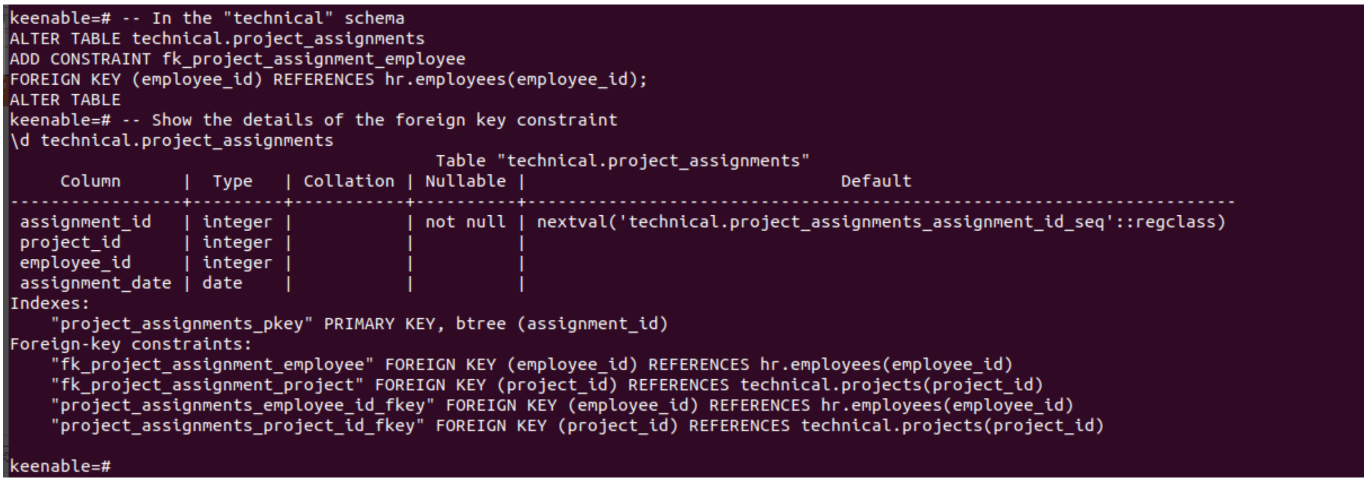
For that I had used the following command:

1. keenable=#-- In the "technical" schema ALTER TABLE technical.project_assignments ADD CONSTRAINT fk_project_assignment_project FOREIGN KEY (project_id) REFERENCES technical.projects(project_id); ALTER TABLE

2. keenable=# -- In the "technical" schema ALTER TABLE technical.project_assignments ADD CONSTRAINT fk_project_assignment_employee FOREIGN KEY (employee_id) REFERENCES hr.employees(employee_id); ALTER TABLE



```
keenable=# -- In the "technical" schema
ALTER TABLE technical.project_assignments
ADD CONSTRAINT fk_project_assignment_project
FOREIGN KEY (project_id) REFERENCES technical.projects(project_id);
ALTER TABLE
keenable=# -- Show the details of the foreign key constraint
\d technical.project_assignments
                                        Table "technical.project_assignments"
     Column      |  Type   | Collation | Nullable |                               Default
-----------------+---------+-----------+----------+------------------------------------------------------------------------
 assignment_id   | integer |           | not null | nextval('technical.project_assignments_assignment_id_seq'::regclass)
 project_id      | integer |           |          |
 employee_id     | integer |           |          |
 assignment_date | date    |           |          |
Indexes:
    "project_assignments_pkey" PRIMARY KEY, btree (assignment_id)
Foreign-key constraints:
    "fk_project_assignment_project" FOREIGN KEY (project_id) REFERENCES technical.projects(project_id)
    "project_assignments_employee_id_fkey" FOREIGN KEY (employee_id) REFERENCES hr.employees(employee_id)
    "project_assignments_project_id_fkey" FOREIGN KEY (project_id) REFERENCES technical.projects(project_id)
```



```
keenable=# -- In the "technical" schema
ALTER TABLE technical.project_assignments
ADD CONSTRAINT fk_project_assignment_employee
FOREIGN KEY (employee_id) REFERENCES hr.employees(employee_id);
ALTER TABLE
keenable=# -- Show the details of the foreign key constraint
\d technical.project_assignments
                                        Table "technical.project_assignments"
     Column      |  Type   | Collation | Nullable |                               Default
-----------------+---------+-----------+----------+------------------------------------------------------------------------
 assignment_id   | integer |           | not null | nextval('technical.project_assignments_assignment_id_seq'::regclass)
 project_id      | integer |           |          |
 employee_id     | integer |           |          |
 assignment_date | date    |           |          |
Indexes:
    "project_assignments_pkey" PRIMARY KEY, btree (assignment_id)
Foreign-key constraints:
    "fk_project_assignment_employee" FOREIGN KEY (employee_id) REFERENCES hr.employees(employee_id)
    "fk_project_assignment_project" FOREIGN KEY (project_id) REFERENCES technical.projects(project_id)
    "project_assignments_employee_id_fkey" FOREIGN KEY (employee_id) REFERENCES hr.employees(employee_id)
    "project_assignments_project_id_fkey" FOREIGN KEY (project_id) REFERENCES technical.projects(project_id)

keenable=#
```

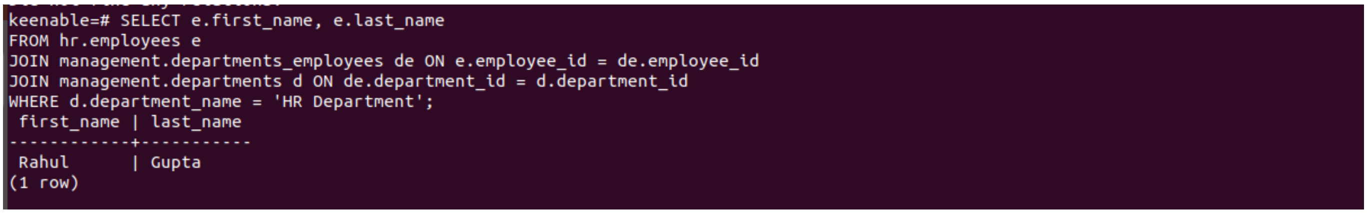
4. Queries and Reporting:

Write SQL queries to retrieve information such as: 4.1. Employees in a specific department 4.2. Projects assigned to an employee. 4.3. Vacation requests for an employee.

4.1.

For that I had used the following command:

keenable=# SELECT e.first_name, e.last_name FROM hr.employees e JOIN management.departments_employees de ON e.employee_id = de.employee_id JOIN management.departments d ON de.department_id = d.department_id WHERE d.department_name = 'HR Department';



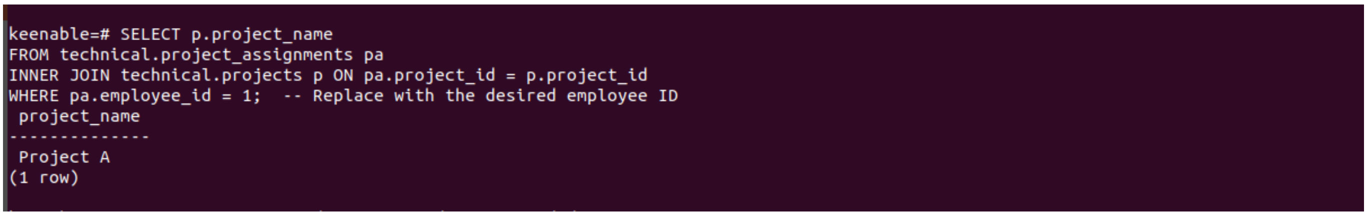
```
keenable=# SELECT e.first_name, e.last_name
FROM hr.employees e
JOIN management.departments_employees de ON e.employee_id = de.employee_id
JOIN management.departments d ON de.department_id = d.department_id
WHERE d.department_name = 'HR Department';
 first_name | last_name
------------+-----------
 Rahul      | Gupta
(1 row)
```

Employees in a specific department

4.2.

For that I had used the following command:

keenable=# SELECT p.project_name FROM technical.project_assignments pa INNER JOIN technical.projects p ON pa.project_id = p.project_id WHERE pa.employee_id = 1; -- Replace with the desired employee ID



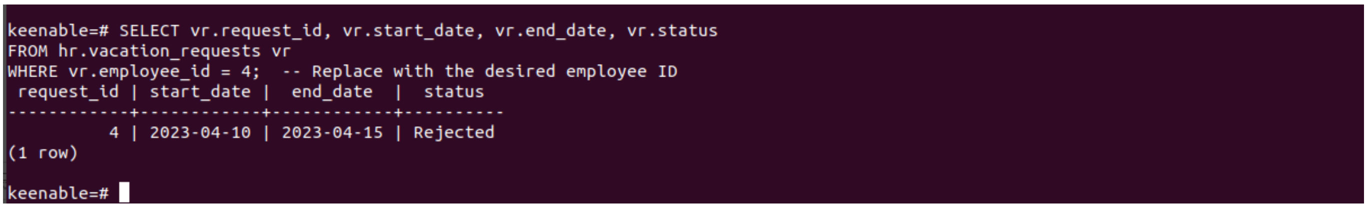
```
keenable=# SELECT p.project_name
FROM technical.project_assignments pa
INNER JOIN technical.projects p ON pa.project_id = p.project_id
WHERE pa.employee_id = 1;  -- Replace with the desired employee ID
 project_name
--------------
 Project A
(1 row)
```

Projects assigned to an employee.

4.3.

For that I had used the following command:

keenable=# SELECT vr.request_id, vr.start_date, vr.end_date, vr.status FROM hr.vacation_requests vr WHERE vr.employee_id = 4; -- Replace with the desired employee ID



```
keenable=# SELECT vr.request_id, vr.start_date, vr.end_date, vr.status
FROM hr.vacation_requests vr
WHERE vr.employee_id = 4;  -- Replace with the desired employee ID
 request_id | start_date |  end_date  |  status
------------+------------+------------+----------
          4 | 2023-04-10 | 2023-04-15 | Rejected
(1 row)

keenable=# 
```

Vacation requests for an employee.