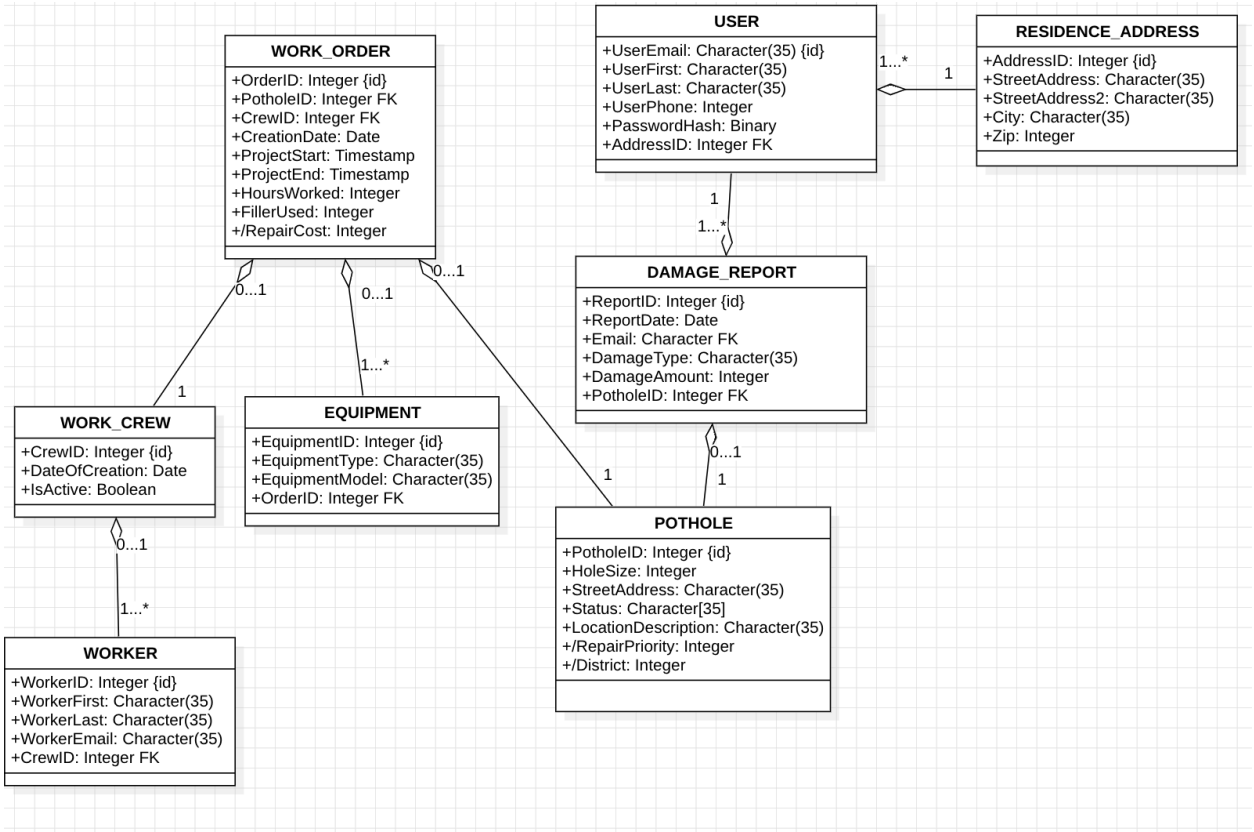
Public Works Department Database Documentation

Our database provides administrative staff at the Public Works Department with a structure for tracking information about reported potholes, users submitting damage reports, equipment and employees. To do this we defined eight entities as described and modeled in our UML diagram.

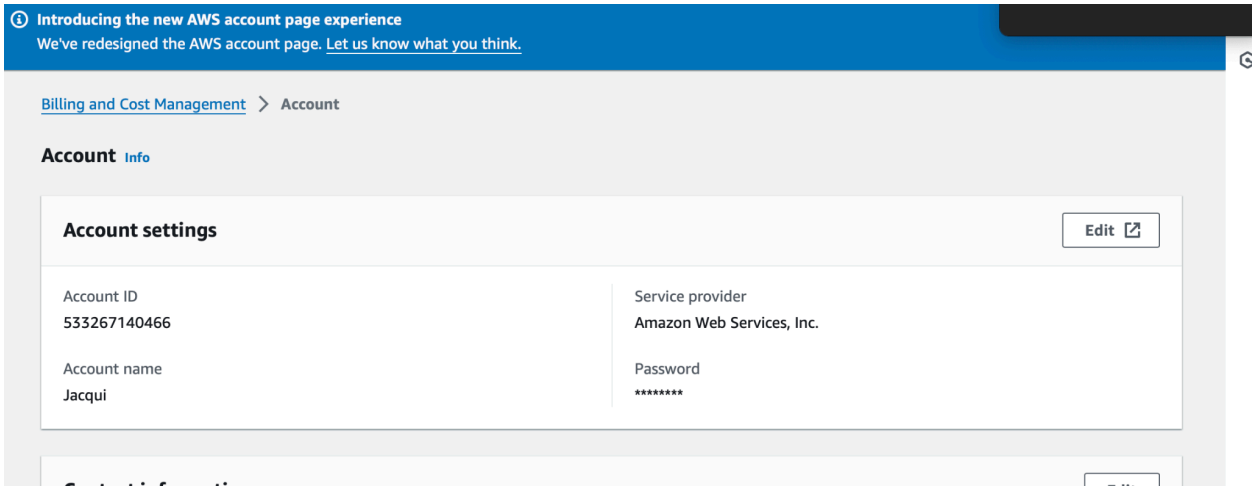
- WORK_ORDER
    - Purpose: To provide information about how a pothole has been chosen to be handled. So users can file a damage report or a pothole can just exist, but until an administrator files a work order on the pothole it won't be fixed. Work orders can be inactive for various reasons (hasn't started, holidays, repaired, administrative issues, etc…). Also provides a start and end date for the repair.
    - Children:
        - POTHOLE: A work order must be associated with one pothole. However, a pothole can be associated with 0 or 1 work orders.
        - WORK CREW: A work order must be associated with one work crew. But a work crew can be associated with 0 or 1 work orders. (Constraint: A work order cannot be inactive with an active work crew assigned to it) (Constraint: A work order can be created prior to work crew but the project can't start before work crew is created).
- WORK_CREW
    - Purpose: To provide information about the group of workers assigned to repair a certain pothole through a work order. A work crew can exist without having been assigned a work order but can be assigned to at most one work order.
- EQUIPMENT
    - Purpose: To provide information about the equipment used to repair a certain pothole through an issued work order.
    - Children:
        - WORK_ORDER: A work order has 1 or many equipment assigned to it. 1 or many equipment can be associated with 0 or 1 work orders. This is to highlight that equipment exists (and possibly free to assign), without a work order existing for it.
- WORKER
    - Purpose: To provide information about a worker assigned to repair a certain pothole as part of a work crew.
    - Children:
        - WORK_CREW: A work crew is made up of 1 to many workers. The number of workers per work crew can vary. A worker can only be assigned to 0 or 1 work crews. A worker can exist without being assigned to a work crew.
- USER
    - Purpose: To provide information about a user who's filed a damage report about a certain pothole. A user's first name, last name and phone number can be used

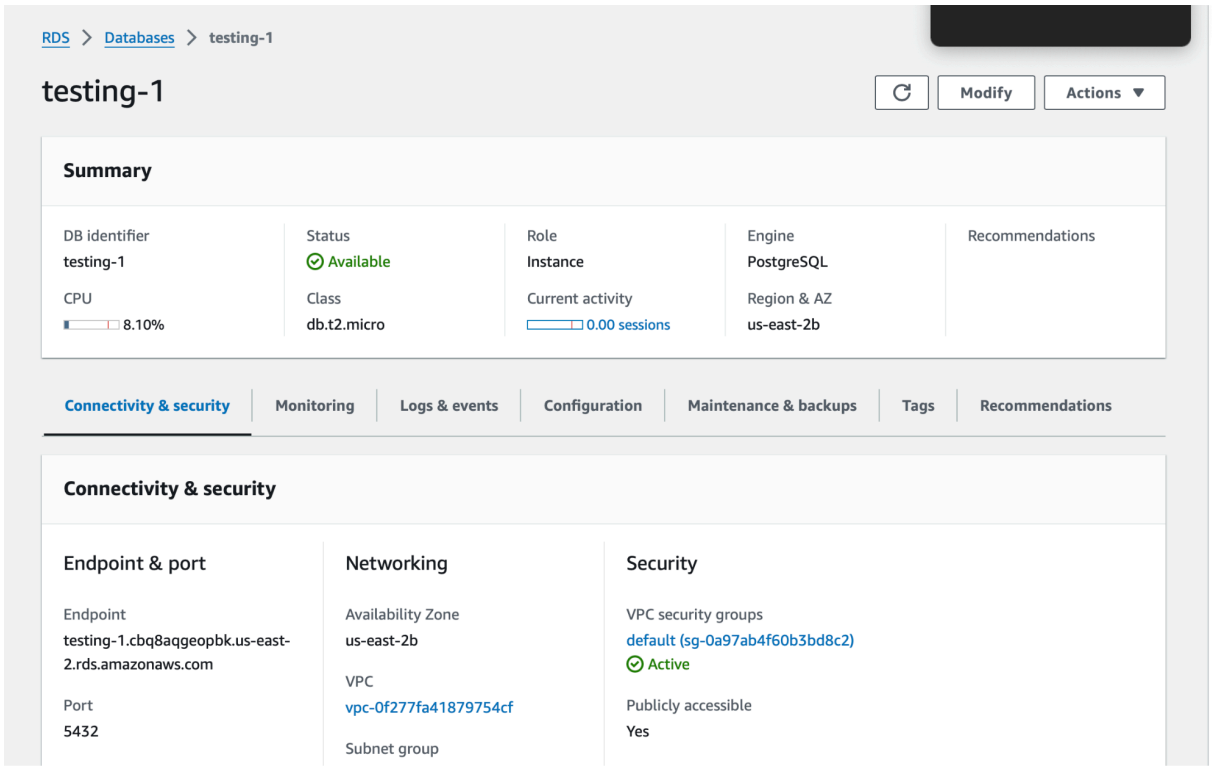
as an alternative key. Additionally, the phone number itself can be an alternative key.
    - Children:
        - RESIDENCE_ADDRESS: Each user must be associated with one address. One address can be associated with 1 or more users. This is to highlight that many people living in the same place can file reports on potholes.
- DAMAGE_REPORT
    - Purpose: To provide information about the report a user has made on a pothole and the damage associated with it.
    - Children:
        - POTHOLE: A damage report must be associated with one pothole. However a pothole can be associated with 0 or 1 damage reports. In our design we chose to highlight that a pothole can exist without someone having filed a damage report on it.
        - USER: A user can have made 1 or more damage reports. A damage report can only be associated with a single user.
- POTHOLE
    - Purpose: To provide information about existing potholes including size, location, and whether it's be repaired. Repair priority and district are derived attributes. Repair priority will be calculated by the size of pothole and district by street address.
- RESIDENCE_ADDRESS
    - Purpose: To provide information about where a user who's filed a damage report lives.

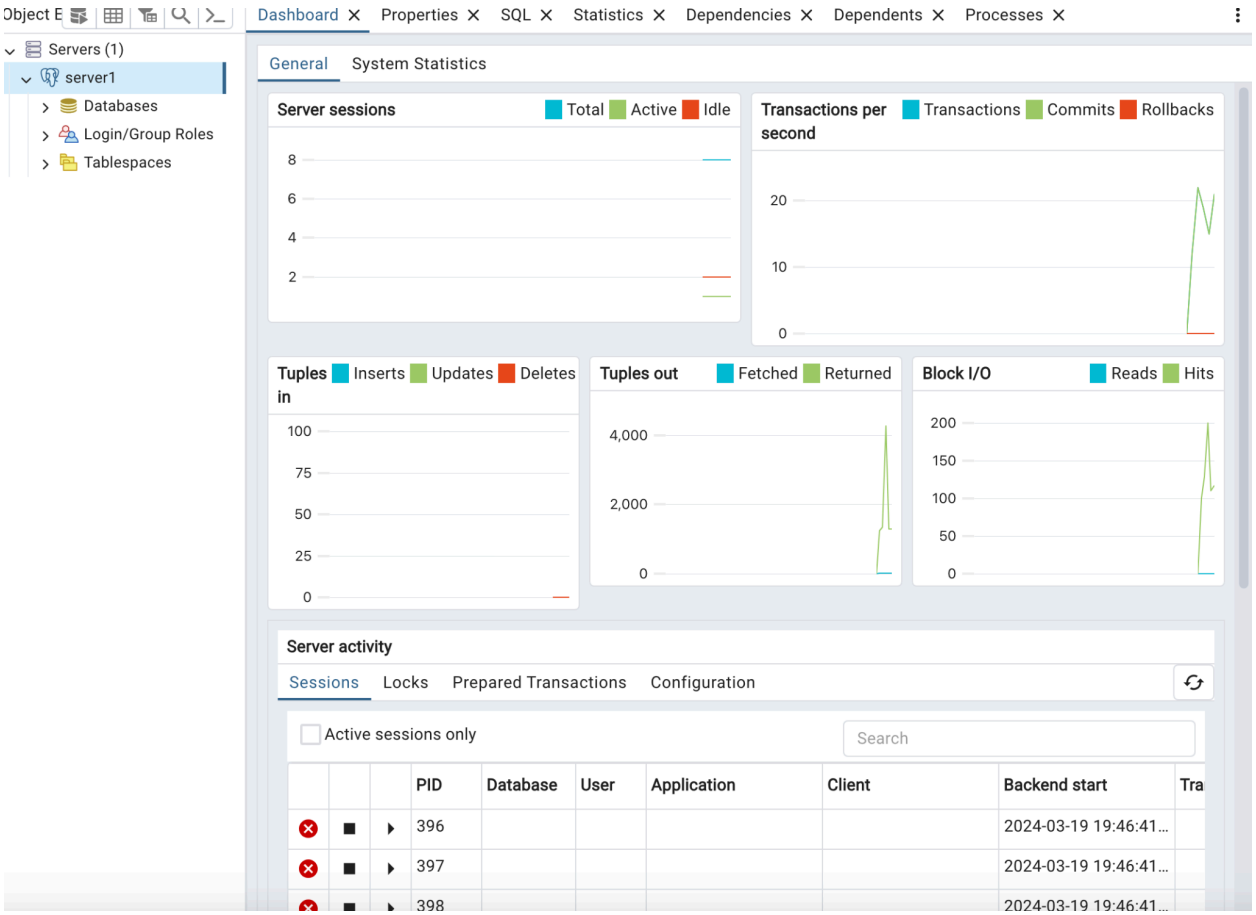


PostgreSQL instance on Amazon Web Services

Experience: I learned that AWS offers the Amazon RDS service which we use to host PostgreSQL databases which we will manage using pgAdmin. Using AWS allows us to easily scale database resources up or down depending on demand and PgAdmin allows us to manage and administer the PostgreSQL databases hosted on AWS. AWS makes it easier to deploy, configure and maintain our databases while pgAdmin gives us a “friendlier” way to manage database objects, run queries and monitor performance.


Introducing the new AWS account page experience
We've redesigned the AWS account page. Let us know what you think.
Billing and Cost Management > Account
Account Info
Account settings
Edit
Account ID
533267140466
Service provider
Amazon Web Services, Inc.
Account name
Jacqui
Password
****


RDS > Databases > testing-1
testing-1
Modify
Actions
Summary
DB identifier
testing-1
Status
Available
Role
Instance
Engine
PostgreSQL
Recommendations
CPU
8.10%
Class
db.t2.micro
Current activity
0.00 sessions
Region & AZ
us-east-2b
Connectivity & security
Monitoring
Logs & events
Configuration
Maintenance & backups
Tags
Recommendations
Connectivity & security
Endpoint & port
Endpoint
testing-1.cbq8aqgeopbk.us-east-2.rds.amazonaws.com
Port
5432
Networking
Availability Zone
us-east-2b
VPC
vpc-0f277fa41879754cf
Subnet group
Security
VPC security groups
default (sg-0a97ab4f60b3bd8c2)
Active
Publicly accessible
Yes


Object E
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Servers (1)
server1
Databases
Login/Group Roles
Tablespaces
General
System Statistics
Server sessions
Total
Active
Idle
8
6
4
2
Transactions per second
Transactions
Commits
Rollbacks
20
10
0
Tuples in
Inserts
Updates
Deletes
100
75
50
25
0
Tuples out
Fetched
Returned
4,000
2,000
0
Block I/O
Reads
Hits
200
150
100
50
0
Server activity
Sessions
Locks
Prepared Transactions
Configuration
Active sessions only
Search
PID
Database
User
Application
Client
Backend start
Tra
396
2024-03-19 19:46:41...
397
2024-03-19 19:46:41...
398
2024-03-19 19:46:41...