DB with Roles

Seungin Lyu, Danny Kang

26.0k

Accidentally destroyed production database on first day of a job, and was told to leave, on top of this i was told by the CTO that they need to get legal involved, how screwed am i?

Today was my first day on the job as a Junior Software Developer and was my first non-internship position after university. Unfortunately i screwed up badly.

I was basically given a document detailing how to setup my local development environment. Which involves run a small script to create my own personal DB instance from some test data. After running the command i was supposed to copy the database url/password/username outputted by the command and configure my dev environment to point to that database. Unfortunately instead of copying the values outputted by the tool, i instead for whatever reason used the values the document had.

Unfortunately apparently those values were actually for the production database (why they are documented in the dev setup guide i have no idea). Then from my understanding that the tests add fake data, and clear existing data between test runs which basically cleared all the data from the production database. Honestly i had no idea what i did and it wasn't about 30 or so minutes after did someone actually figure out/realize what i did.

While what i had done was sinking in. The CTO told me to leave and never come back. He also informed me that apparently legal would need to get involved due to severity of the data loss. I basically offered and pleaded to let me help in someway to redeem my self and i was told that i "completely fucked everything up".

So i left. I kept an eye on slack, and from what i can tell the backups were not restoring and it seemed like the entire dev team was on full on panic mode. I sent a slack message to our CTO explaining my screw up. Only to have my slack account immediately disabled not long after sending the message.

I haven't heard from HR, or anything and i am panicking to high heavens. I just moved across the country for this job, is there anything i can even remotely do to redeem my self in this situation? Can i possibly be sued for this? Should i contact HR directly? I am really confused, and terrified.

EDIT Just to make it even more embarrassing, i just realized that i took the laptop i was issued home with me (i have no idea why i did this at all).

EDIT 2 I just woke up, after deciding to drown my sorrows and i am shocked by the number of responses, well wishes and other things. Will do my best to sort through everything.

Tldr;

An intern at a startup company **destroyed** the production database on **Day 1** (after following a tutorial on how to setup the local dev environment)

https://www.reddit.com/r/cscareerquestions/comments/6ez8ag/accidentally_destroyed_production_database_on/

https://aws.amazon.com/message/680587

Summary of the December 24, 2012 Amazon ELB Service Event in the US-East Region

We would like to share more details with our customers about the event that occurred with the Amazon Elastic Load Balancing Service ("ELB") earlier this week in the US-East Region. While the service disruption only affected applications using the ELB service (and only a fraction of the ELB load balancers were affected), the impacted load balancers saw significant impact for a prolonged period of time.

The service disruption began at 12:24 PM PST on December 24th when a portion of the ELB state data was logically deleted. This data is used and maintained by the ELB control plane to manage the configuration of the ELB load balancers in the region (for example tracking all the backend hosts to which traffic should be routed by each load balancer). The data was deleted by a maintenance process that was inadvertently run against the production ELB state data. This process was run by one of a very small number of developers who have access to this production environment. Unfortunately, the developer did not realize the mistake at the time. After this data was deleted, the ELB control plane began experiencing high latency and error rates for API calls to manage ELB load balancers. In this initial part of the service disruption, there was no impact to the request handling functionality of running ELB load balancers because the missing ELB state data was not integral to the basic operation of running load balancers.

GitLab

https://www.reddit.com/r/cscareerquestions/comments/6ez8ag/accidentally_destroyed_production_database_on/

yorickpeterse **GitLab** 15.3k points · 1 year ago · edited 1 year ago 🏅9

Hi, guy here who accidentally nuked GitLab.com's database earlier this year. Fortunately we did have a backup, though it was 6 hours old at that point.

This is not your fault. Yes, you *did* use the wrong credentials and ended up removing the database but there are so many red flags from the company side of things such as:

A few solutions..

- Backup
- Database Roles and Privileges
- Row-Level Security
- Stay away from
  DROP TABLE table_name;
  DROP DATABASE database_name;

Database Roles and Privileges

Role: a database user, or a group of database users (depends on the setup)

"Roles can own database objects (tables) and can assign privileges on those objects to other roles to control who has access to which objects" (PostgreSQL)

Roles

CREATE ROLE name;

DROP ROLE name;

CREATE ROLE name LOGIN (CREATE USER name)

GRANT group_role TO role1, ... ;

REVOKE group_role FROM role1, ... ;

Grant Privileges

GRANT { { SELECT | INSERT | UPDATE | DELETE | RULE | REFERENCES | TRIGGER }

    [,...] | ALL [PRIVILEGES] }

    ON [TABLE] tablename [, ...]

    TO { username | GROUP groupname | PUBLIC } [, ...] [WITH GRANT OPTION]

Row-Level Security (RLS)

RLS supports two types of security predicates.

- "Filter predicates silently filter the rows available to read operations (SELECT, UPDATE, and DELETE)"

- "Block predicates explicitly block write operations (AFTER INSERT, AFTER UPDATE, BEFORE UPDATE, BEFORE DELETE) that violate the predicate"

Row-Level Security (RLS)

- A hospital can create a security policy that allows nurses to view data rows for their patients only.

- A bank can create a policy to restrict access to financial data rows based on an employee's business division or role in the company.

What We Made

A Car Sales DB with three roles and two tables (Cars, Users)
(not entirely query based, but you get the idea)

	Manager	Employee	Customer
Cars	Create, Read, Update Delete,	Read, Update	Read
Users	Create, Read, Update Delete,	Read, Update	

Demo

Thank you!