

Cybernetic Sabotage

Cybernetic Sabotage is an interesting thriller game that can be used to teach and test SQL concepts to students new to this topic.

In this game, the player plays the role of a leading cybersecurity analyst who has been called in after a major tech corporation, RoboTech Global, has experienced a series of strange occurrences within their AI-driven robotics division. There are fears of an inside job, corporate espionage, or even an AI gone rogue. Their mission is to use SQL to dive into the company's extensive databases and uncover the truth.

In classic fashion, this game comprises suspense and intrigue with an adventure role-play theme to introduce students to SQL concepts that grow in complexity gradually as the game progresses. It ends when the player has successfully used all the SQL concepts and written the relevant queries that eventually leads them to find the problem.

Concepts Covered

- Basic Query Structure of SQL Queries
- Additional Basic Operations
- Null Values
- Aggregate Functions
- Nested Subqueries
- Modification of the Database
- Join Expressions
- Views
- Integrity Constraints
- SQL Data Types and Schemas
- Given the schemas of a relation, create SQL queries using: SELECT, FROM, WHERE, EXISTS, NOT EXISTS, UNIQUE, NOT UNIQUE, ANY, ALL, DISTINCT, GROUP BY and HAVING.
- Set Operations
- Aggregate Functions
- Null Values Nested Subqueries

Database Schema

Employees: Employee ID, Name, Job Role, Division, Last Login Time

Robots: Robot ID, Model, Manufacturing Date, Status (functional, under repair, decommissioned), Last Software Update, Employee ID of the person who last updated the software

Logs: Log ID, Employee ID, Action Description, Timestamp, and Robot ID

Incidents: Incident ID, Description, Timestamp, Robot ID, Employee ID of the person who reported the incident

Access_Codes: Access Code ID, Employee ID, Level of Access, Timestamp of the last use

Plot and Gameplay

1. The game begins with an urgent video call from the CEO of RoboTech Global. There's a situation - malfunctions with their AI-driven robots. You, a top cybersecurity analyst, have been hired to investigate. The player is then introduced to the database and the schema of each table.
2. The player is asked to list all incidents from the 'Incidents' table. The results reveal a series of strange malfunctions, all seem to involve the XJ-5 and QZ-4 robot models. [SELECT, FROM]
3. The player is asked to find the most recent incident involving these models. The data reveals a recent incident with an XJ-5 robot. [WHERE, ORDER, LIMIT]
4. The player needs to find out how many XJ-5 and QZ-4 model robots exist in the company. This requires a `JOIN` operation and the `COUNT` function. The results show a surprisingly high number of both models. [JOIN, COUNT, GROUP BY]
5. The player is asked to find out how many of these robots have been updated in the past one week, pointing towards possible tampering. A significant number of both models have been updated recently, indicating potential sabotage. [Set Operations, NULL]
6. The CEO wants to know which employees have updated these robots recently. The result reveals a list of employees. [Nested Subqueries]
7. The player is asked to mark all XJ-5 and QZ-4 robots as "under repair" in the database. [Modification of Database]
8. The player needs to identify the employee who reported the highest number of incidents. [Constraints]
9. The complexity of queries increases, and the player is encouraged to create a view to simplify their task. For instance, they may create a view that joins the 'Incidents' and 'Robots' tables to easily see all incidents associated with each robot model. [Views]
10. The player is tasked with identifying the source of the malfunctions. The player uses `GROUP BY` and `HAVING` to find models of robots that have more than a certain number of incidents, and `DISTINCT` to remove duplicate entries. The results show that the XJ-5 and QZ-4 models are indeed the common thread in these incidents. [EXISTS, HAVING, DISTINCT]
11. The player is asked to create a new table that records the repair status of all robots. They'll need to define the schema for this table, including appropriate data types for each column. [Data Types, Schema]

12. The player is then asked to insert/update repair records for all XJ-5 and QZ-4 robots into the new table. [Database Modification]

13. The player is tasked with finding the last employee who updated the software of the malfunctioning robots. [Subqueries, Sets]

14. The player finds out that there's a common thread - an employee who had last updated the malfunctioning robots. The results point to an employee named "Alex Mercer". [Join, Subqueries/Nested Queries]

15. The CEO is alarmed by this revelation and asks the player to gather all evidence against Alex Mercer. This would involve joining multiple tables and gathering data related to Alex's access logs, incidents reported, and robots he had access to. The player would use all SQL concepts learned from the game to gather this evidence. []

16. The game ends with a dramatic video call with Alex Mercer, where the player presents the evidence. Alex, cornered, confesses his actions, revealing his motives as a disgruntled employee. The player is congratulated for their diligent analysis and receives an offer for a co-op position at RoboTech Global's Vancouver office.