Adventurer's Guild Database System

Agenda

01. Project Overview
02. Views
03. Subqueries
04. Stored Procedures
05. Triggers
06. Documentation

01

Project Overview

Business Case: Adventurers Guild

- Manages adventurers, quests, inventory, and rewards
- Ensures seamless operations using SQL automation

02.0

Views

What are SQL Views?

A view is a virtual table that represents a stored query result.

Views are used to:

Simplify complex queries

Improve security by restricting access to specific columns

Enhance readability and data presentation

02.1

Views

ActiveQuests - Displays only available quests.

- Business Use:

  Helps members see available quests quickly.

```
CREATE VIEW ActiveQuests
AS SELECT quest_id, title,
difficulty_rating FROM quest
WHERE completed = 0;
```

02.2

Views

TopAdventurers - Ranks adventurers by completion rate.

Business Use:

Encourages competition among adventurers.

```
CREATE VIEW
TopAdventurers AS SELECT
member_id, name,
quest_completion_rate
FROM member ORDER BY
quest_completion_rate DESC;
```

02.3

Views

High-Value Items in the Vault

Business Use:

Acts as a prize pool for guild tournaments or achievement milestones

```
CREATE VIEW
High_Value_Items AS SELECT
i.item_id, i.name, i.rarity,
i.value, g.capacity FROM
item i LEFT JOIN guild_vault
g ON i.vault_id = g.vault_id
WHERE i.value > 1000;
```

02.4 Views

Member Rankings and Completion Rates

Business Use:

Encourages adventurers to have milestones to work toward.

```
CREATE VIEW Member_Stats
AS SELECT m.member_id,
m.name, r.rank_name,
m.quest_completion_rate
FROM member m LEFT JOIN
rank r ON m.rank_id =
r.rank_id;
```

03.0

Subqueries

What is a Subquery?
A subquery (also called a nested query) is a query inside another SQL query. It is
used to retrieve intermediate results that help filter or refine the final output.

03.1

Subqueries

Finding the most profitable adventurer:

```
SELECT member_id, name, SUM(base_amount) AS earnings FROM member JOIN quest_reward USING(member_id) GROUP BY member_id ORDER BY earnings DESC LIMIT 1;
```

highlight elite adventurers for recruitment campaigns, sponsorship, leadership roles

03.2 Subqueries

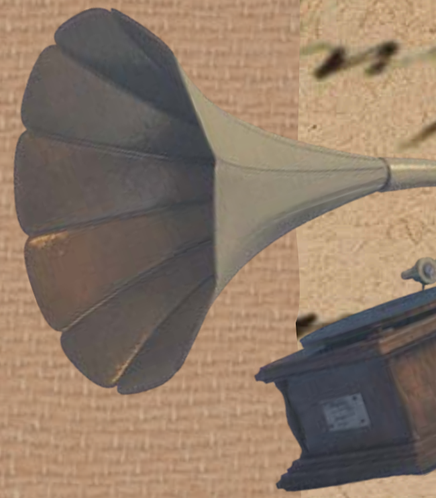
Find the most valuable item stored in the vault.

Apply greater security/restrictions, auction & trade opportunities

```
SELECT name, value FROM item
WHERE value = (SELECT
MAX(value) FROM item);
```

04.0

Stored Procedures & Stored Functions



What are Stored Procedure & Stored Functions?

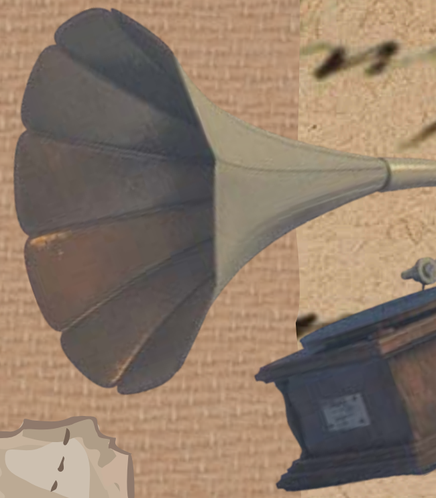
A stored procedure is a reusable SQL script that performs a specific task without returning a value (unless using OUT parameters).

It reduces code duplication, improves performance, and automates repetitive tasks.

A stored function is similar to a procedure but always returns a single value.

It is often used for calculations, conversions, or data transformations

04.1 Stored Procedures

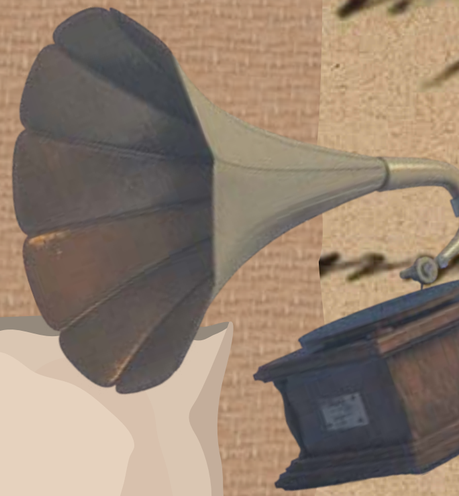


```
DELIMITER $$
CREATE PROCEDURE AssignQuest(IN questID
INT, IN memberID INT)
BEGIN UPDATE quest SET assigned_party_id =
(SELECT party_id FROM party_member
WHERE member_id = memberID)
WHERE quest_id = questID;
END $$
DELIMITER ;
```

Assigning a quest to an adventurer:

Automates quest assignment to adventurers.

04.2 Stored Procedures

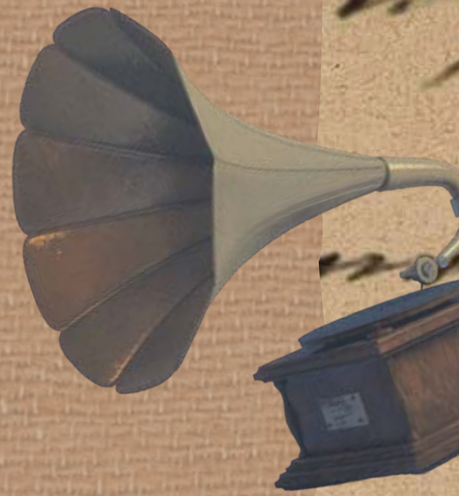


Promote a member based on completion rate

```
DELIMITER //
CREATE PROCEDURE PromoteMember(IN
memberId INT)
BEGIN
    UPDATE member
    SET rank_id = rank_id + 1
    WHERE member_id = memberId AND
quest_completion_rate > 90;
END //
DELIMITER ;
```

Encourage adventurers to complete assigned quests

4.3 Stored Procedures



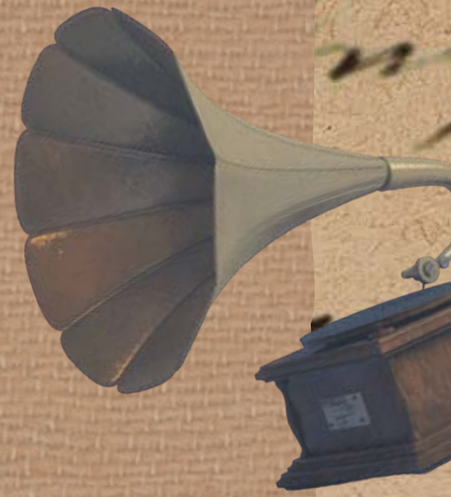
Calculate total value of items in the vault

```
DELIMITER //
CREATE FUNCTION TotalVaultValue()
RETURNS FLOAT
DETERMINISTIC BEGIN DECLARE total FLOAT;
SELECT SUM(value) INTO total FROM item;
RETURN total;
END //
DELIMITER ;
```

Financial management, proper budgeting, vault capacity management

5.0

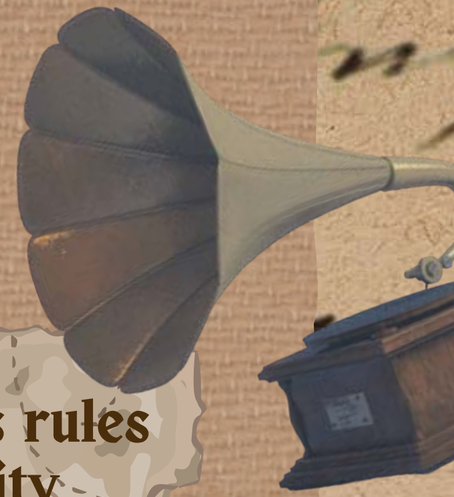
Triggers



What are SQL Triggers?
A trigger is an automatic database action executed before or after a specific event (INSERT, UPDATE, DELETE).
Triggers enforce rules, maintain consistency, and automate processes

5.1

Triggers



Enforces business rules & data integrity

Auto-Rank Up when an adventurer's quest completion rate is high:

Business Use:

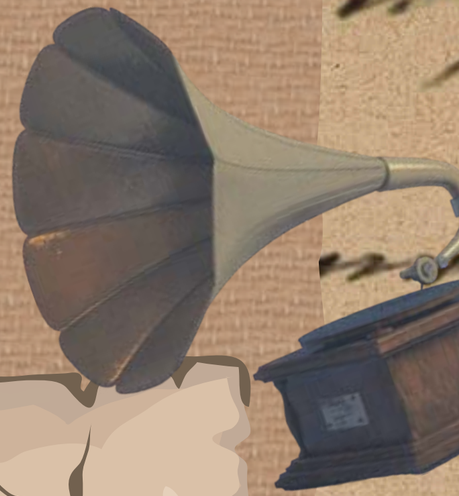
Automatically promotes adventurers based on performance.

```
CREATE TRIGGER AutoRankUp AFTER UPDATE
ON member FOR EACH ROW BEGIN IF
NEW.quest_completion_rate >= 0.9 AND
NEW.rank_id < 5 THEN UPDATE member SET
rank_id = rank_id + 1 WHERE member_id =
NEW.member_id; END IF; END;
```

5.2 Triggers

Prevent deletion of rare or legendary items

```
DELIMITER // CREATE TRIGGER Prevent_Item_Deletion BEFORE DELETE ON item FOR EACH ROW BEGIN IF OLD.rarity IN ('Rare', 'Legendary') THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot delete Rare or Legendary items.'; END IF; END // DELIMITER ;
```



It protects irreplaceable gear, maintains game fairness, and upholds the prestige of elite rewards.

5.3 Triggers

The auto-set quest completion date ensures accurate tracking of when a quest was finished

This is crucial for reward distribution, progress analytics, and historical record-keeping.

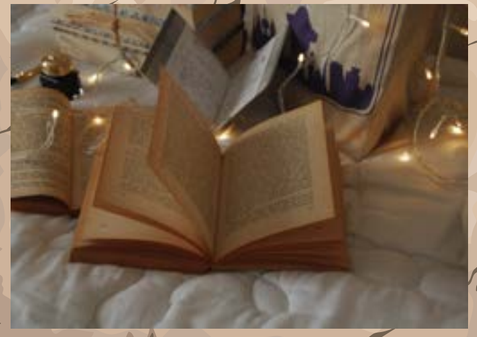
Auto-set quest completion date when marked as completed

```
DELIMITER //
CREATE TRIGGER
Set_Completion_Date BEFORE UPDATE ON quest
FOR EACH ROW
BEGIN
    IF NEW.completed = 1 AND OLD.completed = 0
    THEN SET NEW.deadline = CURDATE();
    END IF;
END //
DELIMITER ;
```

Documentation

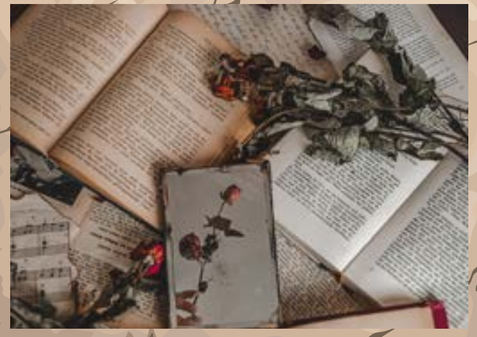
Event 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla accumsan nisl sit amet faucibus accumsan. Aliquam fringilla erat non est blandit.



Event 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla accumsan nisl sit amet faucibus accumsan. Aliquam fringilla erat non est blandit.



Event 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla accumsan nisl sit amet faucibus accumsan. Aliquam fringilla erat non est blandit.

Thank you