

2024

Database Modelling

ASSIGNMENT 1 OF 2

STUDENT: TOMAS ATANASOV

TUTOR: MICHAEL THORNHILL

Contents

Introduction	2
Task one – Design Documentation	2
Entity Definitions	2
Data Dictionary.....	3
ERD.....	11
Relationship Definitions.....	11
Business Rules	12
Assumptions.....	13
Normalisation Report.....	14
First Normal Form	14
Second Normal Form	14
Third Normal Form.....	15
Task Two - Security Concerns (notes)	16
Data Protection Act	16
General Data Protection Regulation.....	16
Hashing and Salting	18
References.....	20
Appendices	21
Appendix A.....	21
Appendix B.....	22

Introduction

The purpose of this document is to provide a full database design documentation for 'Game Knight' development team, which aims to build an online tabletop role-playing game (TTRPG) game with detailed world-building and interactive storytelling called 'Caves and Creatures'. Well-built data base is essential for this task as it will manage quests, inventory, and character data. The 'Game Knight' team goal is to provide an immersive gaming experience while adhering to the data protection regulations and protect users' personal information. The game will target a free-to-play audience exclusive of microtransactions or fees, solely focused on gameplay, skill, farm and hard work in order to progress.

Task one – Design Documentation

Entity Definitions

Entity	Definition
Player	Real life individual who controls a character and plays the game with their own objectives and goals in mind.
Character	A unique virtual persona within a game, named, controlled, and designed according to the player's needs.
Species	Various forms of creatures in the game world which can be assigned to a character, they usually possess unique physical features and traits that set them apart from each other.
Class	A pre-defined identity set of a character within the game, which has its own traits, abilities, playstyle, and visual effects which differentiate from the others.
Alignment	Different predetermined foundations of morals, ethics, values and behaviours which are applied to a character by the player who controls it.
Inventory	The individual character's personal storage area which is usually linked with the equipment interface.
Item	Anything that a character can loot and interact within the game world, like equipment, quest items, consumables, or materials.
Quest	A mission assigned to the character which the player chooses to complete in order to progress in the game or to achieve a specific goal and gain profit.
Region	Certain area within the game world which has unique location specified with its theme, difficulty, etc.

Character type	Categorizes characters as either controlled by human players or those controlled by the game system.
Item rarity	Unique categorisation for the difficulty of finding a specific item, usually linked by its drop chance.
Item type	A unique categorisation for specific group of items within the game, based on their type for example – quest items, warrior items, mage items, consumable items, materials, cloth items, leather items, plate items, wings, boots, rings, etc.
Difficulty	Outlines the complexity of a certain content within the game, for example quest, raid, zone etc.

Data Dictionary

Player		
Attribute	Data Type	Definition
player_id	integer	A unique identifier and a primary key for a specific individual account in the game.
player_forename	varchar	Forename set by the account's original creator, which can prove original player's identity and grant access to their account in case its details are lost or stolen.
player_surname	varchar	Surname set by the account's original creator, which can prove original player's identity and grant access to their account in case its details are lost or stolen.
player_username	varchar	Unique syntax chosen by the player during account creation, which is used by the player to log in to their account

player_password	varchar	Security measure to prove player's identity and grant access to their account.
player_email	varchar	Security measure to prove player's identity and retrieve account details.
player_dob	varchar	Date of birth set by the account's original creator, which can prove original player's identity and grant access to their account in case its details are lost or stolen.

Character		
Attribute	Data Type	Definition
character_id	integer	Unique identifier and a primary key for a specific character in the game.
player_id	integer	A unique identifier for specific individual account in the game.
character_name	varchar	Unique name of a character, chosen by the player on character creation, which also serves as a unique identifier which separates it from every other character in the game.
character_description	varchar	Provides a detailed information about the characteristics of a specific character within the game.
character_type_id	varchar	Unique identifier which categorizes character types and separates them from each other.

class_id	integer	Unique identifier for a class within the game.
species_id	integer	Unique identifier for a race within the game.
alignment_id	integer	Unique identifier for the character's personality within the game.
inventory_id	integer	A Unique identifier for every single player or NPC inventory in the game, which can serve as a tracking tool to grant or restore items.

Character type

Attribute	Data Type	Definition
character_type_id	integer	Unique identifier which categorizes character types and separates them from each other.
character_type_name	varchar	The name of a specific character type within the game.
character_type_description	varchar	Detailed information about a specific character type within the game.
character_type_playability	varchar	Identifier that categorizes characters as either controlled by players or those controlled by the game system.

Species

Attribute	Data Type	Definition
species_id	integer	A unique identifier and a primary key for each different race in the game.
species_name	varchar	Title of a specific specie within the game.

species_description	varchar	Provides a detailed description about the characteristics of a specific specie within the game.
Class		
Attribute	Data Type	Definition
class_id	integer	A unique identifier and a primary key for each different class in the game.
class_name	varchar	Title of a specific class. within the game.
class_description	varchar	Provides a detailed description about the characteristics of a specific class within the game.
Alignment		
Attribute	Data Type	Definition
alignment_id	integer	A unique identifier for each different personality in the game.
alignment_name	varchar	Title of a specific alignment within the game.
alignment_description	varchar	Provides a detailed description about the characteristics of a specific alignment within the game.
Inventory		
Attribute	Data Type	Definition
Inventory_id	integer	Unique identifier for each individual character's personal storage area.

inventory_capacity	varchar	Unique identifier for a character in the game.
item_id	integer	A unique identifier for any single item that a character can loot and interact within the game world, like equipment, quest items, consumables, or materials.
item_count	integer	A numerical indicator which represents the quantity of a particular item.
Item		
Attribute	Data Type	Definition
item_id	integer	A unique identifier and primary key for any single item that a character can loot and interact within the game world, like equipment, quest items, consumables, or materials.
item_name	varchar	Title of a specific item within the game.
item_description	varchar	Provides a detailed description about the characteristics of a specific item within the game.
item_type_id	integer	A unique identifier for specific group of items within the game, based on their type for example – quest items, warrior items, mage items, consumable items, materials, cloth items, leather items, plate items, wings, boots, rings, etc.

item_rarity_id	integer	Unique identifier for every specific group of items, based on their quality.
character_id	integer	Unique identifier for a specific character in the game.
Item type		
Attribute	Data Type	Definition
item_type_id	integer	A unique identifier for specific group of items within the game, based on their type for example – quest items, warrior items, mage items, consumable items, materials, cloth items, leather items, plate items, wings, boots, rings, etc.
item_type_name	varchar	A title for specific group of items within the game, based on their type for example – quest items, warrior items, mage items, consumable items, materials, cloth items, leather items, plate items, wings, boots, rings, etc.
item_type_description	varchar	A detailed description for specific group of items within the game, based on their type for example – quest items, warrior items, mage items, consumable items, materials, cloth items, leather items, plate items, wings, boots, rings, etc.
Item rarity		
Attribute	Data Type	Definition
item_rarity_id	integer	Unique identifier for every specific group of items, based on their quality.

item_rarity_name	varchar	Title for a specific group of items, based on their quality.
item_rarity_description	varchar	A detailed description for a specific group of items, based on their quality.

Quest		
Attribute	Data Type	Definition
quest_id	integer	A unique identifier and primary key for a mission assigned to the character which the player chooses to complete in order to progress in the game or to achieve a specific goal and gain profit.
quest_name	varchar	Title for a mission assigned to the character which the player chooses to complete in order to progress in the game or to achieve a specific goal and gain profit.
quest_description	varchar	Detailed description for a mission assigned to the character which the player chooses to complete in order to progress in the game or to achieve a specific goal and gain profit.
region_id	integer	Unique identifier for certain area within the game world which has unique location specified with its theme, difficulty, etc.
difficulty_id	integer	A unique identifier for the difficulty of a mission assigned to the character which the player chooses to complete to progress in the game or to achieve a

		specific goal and gain profit.
character_id	integer	Unique identifier for a specific character in the game.
Region		
Attribute	Data Type	Definition
region_id	integer	Unique identifier for certain area within the game world which has unique location specified with its theme, difficulty, etc.
region_name	varchar	Title for certain area within the game world which has unique location specified with its theme, difficulty, etc.
region_description	varchar	A detailed description about certain area within the game world which has unique location specified with its theme, difficulty, etc.
Difficulty		
Attribute	Data type	Definition
difficulty_id	integer	Unique identifier and primary key for the difficulty level and complexity of a certain content within the game, for example quest, raid, zone etc.
difficulty_name	varchar	Title representing the difficulty level and complexity of a certain content within the game, for example quest, raid, zone etc.
difficulty_description	varchar	Detailed description about the difficulty level and complexity of a certain

		content within the game, for example quest, raid, zone etc.
--	--	---

ERD

(Appendix A).

https://lucid.app/lucidchart/e1452c3e-7278-42a6-8193-21d4378052cd/edit?viewport_loc=-2116%2C-960%2C3328%2C1630%2C0_0&invitationId=inv_3f3b8f46-7589-479f-a8d0-2cfeb22b5e5d

Relationship Definitions

Entities	Relationships
Player and character	A Player can have zero or many characters. Character must be linked to zero or one player.
Character and character type	A character must be linked to one and only one character type. A character type can be linked to one or many characters.
Character and class	A character must be linked to one and only one class. A class can be linked to one or many characters.
Character and species	A character must be linked to one and only one specie. A specie can be linked to one or many characters.
Character and alignment	A character must be linked to one and only one alignment. An alignment can be linked to one or many characters.
Character and quest	A character can have zero or many quests. One or many quests can be linked to one or many characters.

Character and item	A character can have zero or many items. An item can be linked to one or many characters.
Character and inventory	A character must have one and only one inventory. An inventory is given to one or many characters.
Item and inventory	An item can be in zero or many inventories. One or many items can be linked to zero or many inventories.
Item and item type	An item must have one and only one item type. An Item type can be assigned to one or many items.
Item and item rarity	An item must have one and only one item rarity. A rarity can be assigned to one or many items.
Quest and difficulty	A quest must have one and only one difficulty. A difficulty can be assigned to one or many quests.
Quest and region	A quest can have requirements from one or many regions. A region can have one or many quests in it.

Business Rules

- Player accounts should be unique and linked to an email address, a player may create multiple characters within the game.
- Characters within the game are categorised as player and non-player, a player may only link to their own characters, and categorised by Class, Species, Alignment.

- Items should be linked to a character through an inventory system, items must have an associated type and rarity.
- A Quest must be linked to a region, categorised by difficulty rating and can be completed by multiple Characters.

Assumptions

- Players might be able to change their individual character's species within the game, upon meeting a certain condition or paying for it.
- Players might be able to change their individual character's class within the game, upon meeting a certain condition or paying for it.
- Players might be able to change their individual character's name within the game, upon meeting a certain condition or paying for it.
- Players might be able to change their individual character's alignment within the game, upon meeting a certain condition or paying for it.
- Players might be able to transfer a specific character into a different account upon meeting a certain condition or paying for it.
- Players may have multiple accounts.
- Players may have to pay a subscription to access the game monthly.
- Players might have to buy the game to play.
- Players might need to buy certain microtransactions to progress in the game after a certain level.
- Some items might be bound to a character and not transferable/tradeable.

- Some items might only be transferable to other characters within the account they populate.
- Items under a certain rarity might drop from the character on death.
- Players might be able to increase items drop rate upon meeting a certain condition or paying for it.
- Players might be able to upgrade items rarity upon meeting a certain condition or paying for it.
- A quest might be exclusive only to a single species, class, or alignment.
- A quest might be available only after a certain level.

Normalisation Report

According to geeksforfeeks (2023), normalisation is the process of using Normal Forms to terminate or decrease any redundancy in relations across database tables. Rouse and Vaughan, (2019) also describe the process of normalisation further by explaining its various levels as 1st, 2nd, 3rd, 4th, 5th, etc. Normal Forms which are used to refine, sort, and present the rationalized data in the most efficient way possible.

First Normal Form

Further In the work of Rouse and Vaughan, (2019) the First normal form is presented as the simplest level of normalisation which just apply the following rules, every column in the table must have unique name, and every cell in the table should contain only a single value, which ultimately leads to the removal of repeated information and makes queries easier.

- each cell in every table should contain only a single value.
- each column in every table should have a unique name.

Second Normal Form

Following the 1NF, Kidd, (2022) elaborates on the 2NF and presents it as a rule to apply a single main key to the table while following the criteria set by the 1NF, which means creating extra tables connected by foreign key tags for data subsets that need more rows. Additionally, the article of Kidd, (2022) expands on the 3NF data model and describes that it must adhere to the 2NF while each piece of data in a table is identified by the primary key in the table.

- Adhere to all the 1NF requirements.
- Have one primary key applied to every single table.

Third Normal Form

In her article, Pedamkar, (2023) highlights the advantages of the 3NF by claiming that it improves data quality by removing unnecessary information and fixing update issues. It ensures that tables don't have any problems with adding, updating, or deleting data, which keeps databases reliable. Even though it's helpful, Pedamkar, (2023) exposes some of the disadvantages of the 3NF by outlining that implementing it can make databases more complex and take longer to develop, but however, understanding how it works is crucial for creating databases that meet business needs.

Normalisation has been carefully followed in the model that is presented in this document, ensuring that it meets the standards of the Third Normal Form (3NF) and the process is shown in ([Appendix B](#)). Therefore, each of the columns in every table consist of unique names, all the cells in every table stores only a single value, a primary key is applied to each table, and every piece of data in each of the tables is identified by the primary key of the table.

- Adhere to all the 2NF requirements.
- Data in the tables should be identified only by the primary key.

Task Two - Security Concerns (notes)

Data Protection Act

Data protection laws in the UK were built on the 1998 Data Protection Act (DPA) which covers any stored data about people and was developed to protect and regulate the way their data can be used. Organisations adopted the eight principles outlined in DPA to develop their own data protection policies. Compliance with these eight principles was critical for companies to meet their commitments. Organisations should adhere to this act in order to protect individual's rights, preserve their trust and reputation, and avoid legal consequences.

If organisation was found to be breaching this act, it could result in prosecuting the organisation criminally or being subject to a compensation claim by the affected individuals.

However, this act was designed in times when data sharing and storage was still something new and unknown to many people, 20 years passed, technology advanced and this act became rather outdated and vulnerable, so it was replaced by the General Data Protection Regulation (GDPR)

General Data Protection Regulation

During 2018, a stricter and more punishing law called General Data Protection Regulation (GDPR) replaced the 1998 DPA, it imposed various extra duties and penalties on organisations and is designed to protect citizens and control the way their personal information is used, and any organisation must adhere to it to prevent stealing and misuse of their personal information and causing any harm.

Repercussions in case if inadequate security measures are not taken and an organisation breach this act include a fine, up to £17 million or 4% of their annual worldwide turnover, whichever is greater. As a result, the organisation could face major financial problems, including loss of reputation, serious financial stress, and even bankruptcy in the worst cases.

Regardless of GDPR, the eight data protection principles established in the 1998 Data Protection Act have been slightly adjusted, but remain relevant to this day.

The eight principles of data protection are the following:

1. Processing personal information fairly and lawfully - Personal data should be processed fairly and lawfully and, shall not be processed unless certain conditions, set out in the Act, are met.
2. Processing personal data for specified purposes only - Personal data shall be obtained only for one or more specified and lawful purposes and shall not be further processed in any manner incompatible with that purpose or those purposes”.
3. The amount of personal information - Personal data shall be adequate, relevant, and not excessive in relation to the purpose or purposes for which they are processed.
4. Keeping personal information accurate and up to date - Personal data shall be accurate, and where necessary, kept up to date.
5. Keeping personal information - Personal data processed for any purpose or purposes shall not be kept for longer than is necessary for that purpose or those purposes.
6. Ensuring that people’s rights are maintained - Personal data shall be processed in accordance with rights of the person. For example, a person has the right to request access to the information, prevent the processing of information if it is likely to cause damage or distress, correct mistakes in the information, claim compensation if the information is inaccurate or if the Data Protection Act has been breached.
7. Information Security - Appropriate technical and organisational measures shall be taken against unauthorised or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data.
8. Sending information outside the European Union - Personal data shall not be transferred to a country or territory outside the EEA unless that country or territory ensures an adequate level of protection for the rights of data for subjects in relation to the processing of personal data.

Conclusion

In today's fast-paced world of technology, keeping data safe is like navigating through a maze. It's not just about having strong locks on your digital doors, it's also about making sure everyone in the organization knows how to keep things safe. That means investing time and resources in training programs so that everyone understands the importance of data protection and knows what to do to keep data secure.

Plus, with data flowing across borders like never before, it's like trying to juggle different languages and customs. Companies must make sure they're following all the rules, not just in their own country, but wherever their data goes. It's a bit like being a global traveller, needing to understand the local laws and customs of every place you visit.

But data protection isn't just about following rules, it's also about building trust. It's like being a trustworthy friend who always keeps your secrets safe. By showing that they take data protection seriously and respect people's privacy, companies can build strong relationships with their customers and partners.

And just like having a strong roof over your head protects you from the rain, having robust data protection measures in place shields companies from the storms of data breaches. By investing in things like cybersecurity and good data management practices, companies can bounce back quickly if something goes wrong.

In the end, the shift from the old Data Protection Act to the new General Data Protection Regulation is like upgrading from an old bicycle to a shiny new car. It's all about keeping up with the times and making sure that everyone's data stays safe and sound. By sticking to the principles of data protection, companies can steer clear of trouble, protect people's rights, and build a world where everyone feels safe sharing their data.

Hashing and Salting

Encrypting data offers robust protection against unauthorized or unlawful access however, Hashing is one level above encryption, it is the golden standard of storing data in database modelling. Any type of data can be hashed, whether it's a text string, a file, or even binary data. Hashing is the process of applying an additional layer of security to sensitive information, by using an algorithm to generate unique hash value

which is transforming data into a fixed length alphanumeric string of random characters. This process is disfiguring the data so much, that its almost destroying it, because its making it completely unrecognisable and unreadable, pretty much impossible to decrypt so even if there is a data breach, there is nothing useful to steal. Hashing is also a one-way operation, meaning once the data is hashed, its irreversible. A good hashing algorithm should be quick to calculate but hard to reverse, making it tough to figure out the original input from its hash value. Strong hash function should also minimize the possibility of two different inputs producing the same hash value. This property is essential for security, as it prevents attackers from finding different inputs that produce the same hash.

Hashing is commonly used to store passwords, and if the system notifies you that the password you're trying to create is already in use, it implies that someone else has already chosen that password. In such a case, if you also have access to their username, it will grant you unauthorized access to their account. This highlights the critical importance of enforcing policies that require each user to have a unique password, ensuring that passwords are unique to each user.

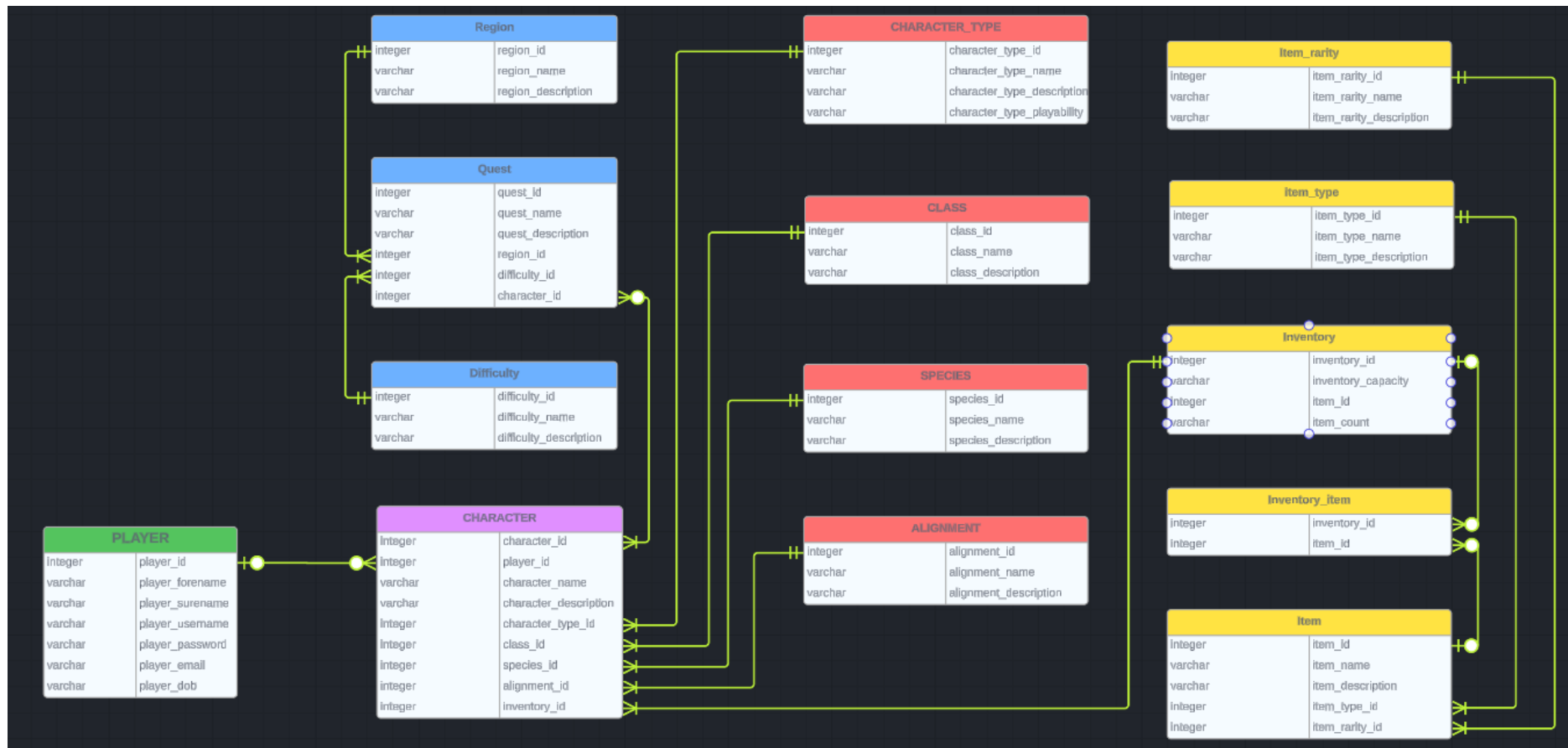
Requiring each user to have a unique password is security measure that mitigates potential impact on password security, and it is crucial to effectively safeguard sensitive information. However, some would say that it creates a scenario where if one user's password is compromised, it potentially puts all other users who share that same password at risk of unauthorized access. However, this is where Salting comes in play, it is one more additional layer of protection which can support Hashing and make it even harder to penetrate. In cryptography, salting means adding a random data to the input of a hash function to ensure a random output, even when the inputs are completely identical. A simple example is, two users having the same password, if there is not salting applied prior to hashing the data, attackers could know the password of both users by knowing the password of one of them, but with salting applied, the hash of both passwords will look different, even if they are identical. This will make it way harder for attackers to penetrate and will definitely enhance security.

References

- geeksforgeeks (2023) *Normal Forms in DBMS*. Available at: <https://www.geeksforgeeks.org/normal-forms-in-dbms/> . (Accessed on 1st of March 2024).
- Kidd, C., (2022) *Data Normalization Explained: How To Normalize Data*. Available at: [Data Normalization Explained: How To Normalize Data | Splunk](#) . (Accessed on 1st of March 2024).
- Rouse M., and Vaughan J., (2019) *DEFINITION database normalization*. Available at: <https://www.techtarget.com/searchdatamanagement/definition/normalization> . (Accessed on 1st of March 2024).
- Pedamkar, P., (2023) *Third Normal Form*. Available at: <https://www.educba.com/third-normal-form/> . (Accessed on 7th of March 2024).

Appendices

Appendix A



Appendix B

