**Database**

**Vault 101**

A Database Administration Project for a Fallout Shelter vault.

**PEDRO MAGALHÃES**

**DATABASE VAULT 101**

Database project introduced in my portfolio as a demonstration of my SQL skills, focusing on my future DBA internship.

**BRASÍLIA - DF**

**2023**

Summary

[1. INTRODUCTION 4](#_Toc136449342)

[2. ENTITY–RELATIONSHIP MODEL 4](#_Toc136449343)

[Entity list: 4](#_Toc136449344)

[Attributes list: 5](#_Toc136449345)

[Relationships list: 6](#_Toc136449346)

[3. STATEMENTS 7](#_Toc136449347)

[Functions 7](#_Toc136449348)

[Stored Procedures (for strong entities) 12](#_Toc136449349)

[Stored Procedures (for associative entities) 20](#_Toc136449350)

[4. CONCEPTUAL SCHEMA DESING 21](#_Toc136449351)

[5. LOGICAL SCHEMA DESIGN 24](#_Toc136449352)

# INTRODUCTION

Fallout Shelter is a strategy game where you play as a vault manager. Your mission is to keep dwellers alive, explore the post-apocalyptic world and manage resources. In this game a lot happens, a lot of data is generated, and that’s where the project is involved.

The propose of db\_vault\_101 (‘101’ has no special meaning) is to store events and resources, using stored procedures, functions and analyzing this data into charts and graphs (using Power BI).

# ENTITY–RELATIONSHIP MODEL

## Entity list:

* Decease.
* Decease\_type.
* Diary.
* Dweller.
* Dweller\_quest.
* Exploration.
* Incident.
* Incident\_type.
* Junk.
* Kinship.
* Outfit.
* Pet.
* Quest.
* Room.
* Room\_type.
* Sale.
* Scrap.
* Special.
* Storage.
* Storage\_exploration.
* Storage\_scrap.
* Storage\_quest.
* Weapon.

## Attributes list:

* Decease: id\_decease, date.
* Decease\_type: id\_decease\_type, description.
* Diary: id\_diary, date, dweller\_count, money, room\_count, quest\_count, incident\_count, outfit\_count, weapon\_count, pet\_count, energy, water, food.
* Dweller: id\_dweller, name, gender, level, status, id\_special, id\_outfit, id\_weapon, id\_pet, id\_kinship, id\_decease, id\_room.
* Dweller\_quest: id\_dweller, id\_quest.
* Exploration: id\_exploration, duration, id\_dweller.
* Incident: id\_incident, date, id\_incident\_type, id\_room.
* Incident\_type: id\_incident\_type, description.
* Junk: id\_junk, name, quantity, id\_storage.
* Kinship: id\_kinship, id\_mother, id\_father.
* Outfit: id\_outfit, name, quantity, id\_storage.
* Pet: id\_pet, name, description, quantity, id\_storage.
* Quest: id\_quest, description, date\_begin, date\_end.
* Room: id\_room, size, floor, id\_room\_type.
* Room\_type: id\_room\_type, name, job.
* Sale: id\_sale, id\_product, quantity, profit, date.
* Scrap: id\_scrap, id\_product, scrap\_quantity, date.
* Special: id\_special, strength, perception, endurance, charisma, intelligence, agility, luck, category.
* Storage: id\_storage, category.
* Storage\_exploration: id\_storage, id\_exploration, quantity.
* Storage\_scrap: id\_scrap, id\_storage, junk\_quantity.
* Storage\_quest: id\_storage, id\_quest, quantity.
* Weapon: id\_weapon, name, damage, quantity, id\_storage.

## Relationships list:

* Decease - has - Decease\_type.
* Decease - has - Incident\_type.
* Diary - counts - Dweller.
* Diary - counts - Incident.
* Diary - counts - Outfit.
* Diary - counts - Pet.
* Diary - counts - Quest.
* Diary - counts - Room.
* Diary - counts - Weapon.
* Dweller - has - Decease.
* Dweller - goes - Dweller\_quest.
* Dweller - goes - Exploration.
* Dweller - has - Kinship.
* Dweller - has - Outfit.
* Dweller - has - Pet.
* Dweller - belongs - Room.
* Dweller - has - Special.
* Dweller - has - Weapon.
* Exploration - gets - Storage\_exploration.
* Incident - has - Incident\_type.
* Incident - has - Room.
* Junk - belongs - Storage.
* Kinship - Involves - Dweller.
* Outfit - has - Special.
* Outfit - belongs - Storage.
* Pet - belongs - Storage.
* Quest - has - Dweller\_quest.
* Quest - gets - Storage\_quest.
* Room - has - Room\_type.
* Sale - has - Storage.
* Scrap - has - Storage.
* Scrap - make - Storage\_scrap.
* Storage - belongs - Dweller\_quest.
* Storage - belongs - Storage\_exploration.
* Storage - belongs - Storage\_scrap.
* Storage - belongs - Storage\_quest.
* Weapon - belongs - Storage.

# STATEMENTS

## Functions

|  |  |
| --- | --- |
| **Name** | fn\_calculate\_junk |
| **Description** | Calculates how much junk is available in storage for sale and scrap |
| **Parameters** | |
| **Name** | **Description** |
| ct\_junk | Represents id\_junk |
| accept\_zero | If True, will allow to return 0 in case there is no junk available instead to return a Error Message |
| **Returns** | |
| **Value** | **Description** |
| 0 | When ct\_junk is NULL |
| Error Message | When junk doesn’t exist or when there isn’t junk available for usage (in case accept\_zero is False) |
| Junk quantity | The number of junk available at storage |

|  |  |
| --- | --- |
| **Name** | fn\_calculate\_outfit |
| **Description** | Calculates how much outfit is available in storage for use, sale, and scrap |
| **Parameters** | |
| **Name** | **Description** |
| ct\_outfit | Represents id\_outfit |
| accept\_zero | If True, will allow to return 0 in case there is no outfit available instead to return a Error Message |
| **Returns** | |
| **Value** | **Description** |
| 0 | When ct\_outfit is NULL |
| Error Message | When outfit doesn’t exist or when there isn’t outfit available for usage (in case accept\_zero is False) |
| Outfit quantity | The number of outfit available at storage |

|  |  |
| --- | --- |
| **Name** | fn\_calculate\_pet |
| **Description** | Calculates how much pet is available in storage for use, sale, and scrap |
| **Parameters** | |
| **Name** | **Description** |
| ct\_pet | Represents id\_pet |
| accept\_zero | If True, will allow to return 0 in case there is no outfit available instead to return a Error Message |
| **Returns** | |
| **Value** | **Description** |
| 0 | When ct\_pet is NULL |
| Error Message | When pet doesn’t exist or when there isn’t pet available for usage (in case accept\_zero is False) |
| Pet quantity | The number of pet available at storage |

|  |  |
| --- | --- |
| **Name** | fn\_calculate\_weapon |
| **Description** | Calculates how much weapon is available in storage for use, sale, and scrap |
| **Parameters** | |
| **Name** | **Description** |
| ct\_weapon | Represents id\_weapon |
| accept\_zero | If True, will allow to return 0 in case there is no outfit available instead to return a Error Message |
| **Returns** | |
| **Value** | **Description** |
| 0 | When ct\_weapon is NULL |
| Error Message | When weapon doesn’t exist or when there isn’t weapon available for usage (in case accept\_zero is False) |
| Weapon quantity | The number of weapon available at storage |

|  |  |
| --- | --- |
| **Name** | fn\_count\_floor\_size |
| **Description** | Counts the sizes of the floor, to make sure there isn’t too much room in it |
| **Parameters** | |
| **Name** | **Description** |
| in\_floor | Represents id\_room |
| **Returns** | |
| **Value** | **Description** |
| Floor sizes | Returns how much space was taken up on the floor |
| Floor sizes + 1 | Returns how much space was taken up on the floor (if it’s on the first floor) |

|  |  |
| --- | --- |
| **Name** | fn\_insert\_kinship |
| **Description** | Inserts a new kinship or returns the id\_kinship for the existing kinship |
| **Parameters** | |
| **Name** | **Description** |
| in\_id\_mom | Represents the id\_dweller from the mother |
| in\_id\_dad | Represents the id\_dweller from the father |
| **Returns** | |
| **Value** | **Description** |
| id\_kinship | If the kinship already exists, returns the id\_kinship |
| Last Insert Id | Inserts the new kinship and then returns its id |
| Error Message | In case there is any mistakes at the parameters |

|  |  |
| --- | --- |
| **Name** | fn\_string\_filter |
| **Description** | Filters a string, removing invalid characters |
| **Parameters** | |
| **Name** | **Description** |
| in\_string | Represents the string to be filtered |
| **Returns** | |
| **Value** | **Description** |
| TRIM(in\_string) | The filtered string |

|  |  |
| --- | --- |
| **Name** | fn\_string\_length |
| **Description** | Filters a string, calculating its real length |
| **Parameters** | |
| **Name** | **Description** |
| in\_string | Represents the string to be filtered |
| **Returns** | |
| **Value** | **Description** |
| String Length | The string sizes, including the filter from fn\_string\_filter() |

|  |  |
| --- | --- |
| **Name** | fn\_verify\_special |
| **Description** | Verify if SPECIAL is valid |
| **Parameters** | |
| **Name** | **Description** |
| in\_str | Represents *strength* and verify if it is between 0 and 10 |
| in\_per | Represents *perception* and verify if it is between 0 and 10 |
| in\_end | Represents *endurance* and verify if it is between 0 and 10 |
| in\_cha | Represents *charisma* and verify if it is between 0 and 10 |
| in\_int | Represents *intelligence* and verify if it is between 0 and 10 |
| in\_agi | Represents *agility* and verify if it is between 0 and 10 |
| in\_luc | Represents *luck* and verify if it is between 0 and 10 |
| in\_category | Verify if SPECIAL’s category is ‘dweller’ or ‘outfit’ |
| **Returns** | |
| **Value** | **Description** |
| Error Message | When any parameter is invalid |
| TRUE | When all parameters are valid |

|  |  |
| --- | --- |
| **Name** | fn\_verify\_storage |
| **Description** | Calculates how much product (it can be outfit, weapon, pet, and junk) has at storage |
| **Parameters** | |
| **Name** | **Description** |
| in\_product | The name of the product |
| return\_type | The type of value to return |
| accept\_zero | If True, will allow to return 0 in case there is no outfit available instead to return a Error Message |
| **Returns** | |
| **Value** | **Description** |
| Error Message | When product doesn’t exist or when return\_type not between 1-3 |
| id\_storage | The id\_storage of the in\_product |
| Product category | Describes if the category is outfit, weapon, pet, or junk |
| Storage quantity | Product quantity at storage |

## Stored Procedures (for strong entities)

|  |  |
| --- | --- |
| **Name** | sp\_add\_product |
| **Description** | Adds more products to storage based on product id\_storage |
| **Parameters** | |
| **Name** | **Description** |
| in\_product\_category | Defines the product’s category |
| in\_id\_storage | Represents the product id\_storage |
| in\_qnt\_storage | Actual product’s quantity at storage |
| in\_quantity | The quantity of product to add at storage |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_baby |
| **Description** | Inserts a new dweller as a baby |
| **Parameters** | |
| **Name** | **Description** |
| in\_name | Defines the name of the baby |
| in\_gender | Defines the baby’s gander |
| in\_id\_mom | Defines the id\_dweller from the mother |
| in\_id\_dad | Defines the id\_dweller from the father |
| in\_str | Defines *strength* |
| in\_per | Defines *perception* |
| in\_end | Defines *endurance* |
| in\_cha | Defines *charisma* |
| in\_int | Defines *intelligence* |
| in\_agi | Defines *agility* |
| in\_luc | Defines *luck* |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_decease |
| **Description** | Inserts a new decease (used with sp\_insert\_dweller\_decease) |
| **Parameters** | |
| **Name** | **Description** |
| in\_decease\_type | Defines the local where the decease happened |
| in\_incident\_type | Defines the type of incident |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_diary |
| **Description** | Records the number of items, resources, events, and dwellers at the vault |
| **Parameters** | |
| **Name** | **Description** |
| in\_money | Defines the quantity of money at the moment |
| in\_energy | Defines the percents of energy at the moment |
| in\_water | Defines the percents of water at the moment |
| in\_food | Defines the percents of pet at the moment |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_dweller |
| **Description** | Inserts a new dweller to the vault |
| **Parameters** | |
| **Name** | **Description** |
| in\_name | Defines the name of the dweller |
| in\_gender | Defines the dweller’s gander |
| in\_level | Defines the dweller’s level |
| in\_room | Defines the id\_room where dweller is |
| in\_str | Defines *strength* |
| in\_per | Defines *perception* |
| in\_end | Defines *endurance* |
| in\_cha | Defines *charisma* |
| in\_int | Defines *intelligence* |
| in\_agi | Defines *agility* |
| in\_luc | Defines *luck* |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_dweller\_decease |
| **Description** | Inserts the last id\_decease into the deceased dweller |
| **Parameters** | |
| **Name** | **Description** |
| in\_dweller | Represents the dweller’s id/name |
| in\_decease\_type | Defines the local where the decease happened |
| in\_incident\_type | Defines the type of incident |
| in\_save\_product | If True, will save dweller’s items at storage, otherwise will exclude them |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_exploration |
| **Description** | Inserts a new exploration |
| **Parameters** | |
| **Name** | **Description** |
| in\_dweller | Represents the dweller id |
| in\_duration | Represents the time (in minutes) the dweller was exploring |
| in\_date | Represents the date dweller came back from exploration |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_incident |
| **Description** | Inserts a new incident |
| **Parameters** | |
| **Name** | **Description** |
| in\_date | Define the date when the incident happens, not accepting future dates |
| in\_incident\_type | Define the type of the incident |
| in\_room | Defines in which room this happened |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_junk |
| **Description** | Inserts a new junk at the storage |
| **Parameters** | |
| **Name** | **Description** |
| in\_name | Defines the name of the junk |
| in\_qnt | Define how many junks is being inserted |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_outfit |
| **Description** | Inserts a new outfit at the storage |
| **Parameters** | |
| **Name** | **Description** |
| in\_name | Defines the name of the outfit |
| in\_qnt | Define how many outfits is being inserted |
| in\_str | Defines *strength* |
| in\_per | Defines *perception* |
| in\_end | Defines *endurance* |
| in\_cha | Defines *charisma* |
| in\_int | Defines *intelligence* |
| in\_agi | Defines *agility* |
| in\_luc | Defines *luck* |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_pet |
| **Description** | Inserts a new pet at the storage |
| **Parameters** | |
| **Name** | **Description** |
| in\_name | Defines the name of the pet |
| in\_description | Describes the pet’s function |
| in\_qnt | Define how many pets are being inserted |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_quest |
| **Description** | Inserts a new quest at tb\_quest and the dwellers who went at tb\_dweller\_quest |
| **Parameters** | |
| **Name** | **Description** |
| in\_description | Represents a small description about the quest |
| in\_date\_begin | Defines the date that the quest began |
| in\_dweller\_1 | Represents the first dweller’s id who went to the quest |
| in\_dweller\_2 | Represents the second dweller’s id who went to the quest |
| in\_dweller\_3 | Represents the third dweller’s id who went to the quest |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_room |
| **Description** | Inserts a new room |
| **Parameters** | |
| **Name** | **Description** |
| in\_size | Define the size of the room, from 1 to 3 |
| in\_floor | Define the floor in which the room is, from 1 to 25 |
| in\_room\_type | Define the room type and what job is in it |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_sale |
| **Description** | Inserts a new sale |
| **Parameters** | |
| **Name** | **Description** |
| in\_product | Represents the product which was sold |
| in\_quantity | Defines the quantity of product sold |
| in\_profit | Defines the profit obtained from the sale |
| in\_date | Defines the sale’s date |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_scrap |
| **Description** | Inserts a new scrap |
| **Parameters** | |
| **Name** | **Description** |
| in\_product | Represents the product which was scraped |
| in\_quantity | Defines the quantity of product scraped |
| in\_date | Defines the scrap’s date |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_special |
| **Description** | Inserts a new special |
| **Parameters** | |
| **Name** | **Description** |
| in\_str | Defines *strength* to ‘dweller’ or ‘outfit’ |
| in\_per | Defines *perception* to ‘dweller’ or ‘outfit’ |
| in\_end | Defines *endurance* to ‘dweller’ or ‘outfit’ |
| in\_cha | Defines *charisma* to ‘dweller’ or ‘outfit’ |
| in\_int | Defines *intelligence* to ‘dweller’ or ‘outfit’ |
| in\_agi | Defines *agility* to ‘dweller’ or ‘outfit’ |
| in\_luc | Defines *luck* to ‘dweller’ or ‘outfit’ |
| in\_category | Define for which category the SPECIAL belongs |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_storage |
| **Description** | Inserts a new product at the storage |
| **Parameters** | |
| **Name** | **Description** |
| in\_category | Define which is the product category, accepting ‘outfit’, ‘weapon’, ‘pet’ and ‘junk’ |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_weapon |
| **Description** | Inserts a new weapon at the storage |
| **Parameters** | |
| **Name** | **Description** |
| in\_name | Defines the name of the weapon |
| in\_damage | Describes the weapon’s damage |
| in\_qnt | Define how many weapons are being inserted |

|  |  |
| --- | --- |
| **Name** | sp\_subtract\_product |
| **Description** | Subtracts some products from storage based on product id\_storage |
| **Parameters** | |
| **Name** | **Description** |
| in\_product\_category | Defines the product’s category |
| in\_id\_storage | Represents the product id\_storage |
| in\_qnt\_storage | Actual product’s quantity at storage |
| in\_quantity | The quantity of product to subtract at storage |

|  |  |
| --- | --- |
| **Name** | sp\_update\_dweller\_bonus |
| **Description** | Inserts new items in dweller |
| **Parameters** | |
| **Name** | **Description** |
| up\_dweller | Represents the dweller’s id/name |
| up\_outfit | Represents the outfit id |
| up\_weapon | Represents the weapon id |
| up\_pet | Represents the pet id |

|  |  |
| --- | --- |
| **Name** | sp\_update\_dweller\_special |
| **Description** | Updates dwellers’s SPECIAL |
| **Parameters** | |
| **Name** | **Description** |
| up\_dweller | Represents the dweller’s id/name |
| up\_str | Defines *strength* |
| up\_per | Defines *perception* |
| up\_end | Defines *endurance* |
| up\_cha | Defines *charisma* |
| up\_int | Defines *intelligence* |
| up\_agi | Defines *agility* |
| up\_luc | Defines *luck* |

|  |  |
| --- | --- |
| **Name** | sp\_dweller\_status |
| **Description** | Updates dweller’s status |
| **Parameters** | |
| **Name** | **Description** |
| up\_dweller | Represents the dweller’s id/name |
| up\_level | Defines the dweller’s level |
| up\_room | Defines the id\_room where dweller is |

|  |  |
| --- | --- |
| **Name** | sp\_update\_room |
| **Description** | Updates room’s size |
| **Parameters** | |
| **Name** | **Description** |
| up\_room | Represents room’s id |
| up\_size | Defines room’s new size |

## Stored Procedures (for associative entities)

|  |  |
| --- | --- |
| **Name** | sp\_insert\_storage\_exploration |
| **Description** | Insert earnings in exploration |
| **Parameters** | |
| **Name** | **Description** |
| in\_exploration | References the exploration’s id |
| in\_product | References the product’s id |
| in\_quantity | Defines how many products earns |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_storage\_quest |
| **Description** | Insert earnings in quest |
| **Parameters** | |
| **Name** | **Description** |
| in\_quest | References the quest’s id |
| in\_product | References the product’s id |
| in\_quantity | Defines how many products earns |

|  |  |
| --- | --- |
| **Name** | sp\_insert\_storage\_scrap |
| **Description** | Insert earnings in scrap |
| **Parameters** | |
| **Name** | **Description** |
| in\_scrap | References the scrap’s id |
| in\_product | References the product’s id |
| in\_quantity | Defines how many products earns |

# CONCEPTUAL SCHEMA DESING

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

# LOGICAL SCHEMA DESIGN

Diagrama, Esquemático

Descrição gerada automaticamente

Diagrama, Esquemático

Descrição gerada automaticamente