Build a Base

Group 7

Andreas Buch, Aleksandr Sorokin, Frederik Christiansen, Frederik Jensen.

Business Vision:

Creating a relational database that can CRUD tables and ensures uniqueness of a primary key. It must include a search tool which can combine and select information from multiple tables.

Requirements:

- Create a new table or a new line in a table.
- Read specific lines from table.
- Update existing lines with new values.
- Delete an existing table or delete an existing line.
- Assure the uniqueness of the Primary Key(s).
- Must be able to take appropriate action if a primary key is not unique.

Functional requirements:

REQ01	Tool should be able to return specific data that the user requests.
REQ02	The tool should be able to update existing values in the tables.
REQ03	The tool should be able to delete existing values and entire tables.
REQ04	The tool should be able to create new values and entire tables.
REQ05	The tool should be able to join data from multiple tables and return the combined result to the user.

Non-functional requirements:

Efficiency	The searching tool should have a fast responding time.
Availability	Should be able to use the searching tool anytime throughout the day.
Usability	The searching tool should be easy to interact in.
Maintainability	The searching tool should be easy to fix when errors or bug are discovered.
Simplicity	No same data stored twice.

Actors:

User - The user of the system.
System - The CRUD tool.

Use Cases:

- **Usecase 1** - CREATE Table
  - Create table with
  - Primary key
  - Column names and of different data types in each column

- **Usecase 2** - INSERT
  - Add row to table
  - Automatically increment Primary key of row (for each new row)
  - Make sure there isn't duplicate Primary Keys (That they're unique)

- **Usecase 3** - SELECT
  - Choose what columns to get from a specified table
  - Return data in a table

- **Usecase 4** - WHERE
  - Limit the rows returned in the statement by conditions such as equals, less than, more than etc.

- **Usecase 5** - JOIN
  - Extend the SELECT statement to return data from several tables joined together by columns with matching data

- **Usecase 6** - UPDATE
  - Update data in column(s) where the new data is specified by the SET keyword & the row is limited by the WHERE usecase
  - Update data without deleting & creating a new row so as to keep the same Primary Key

- **Usecase 7** - DELETE
  - Remove row from table specified by the WHERE usecase.

Brief Use Case:

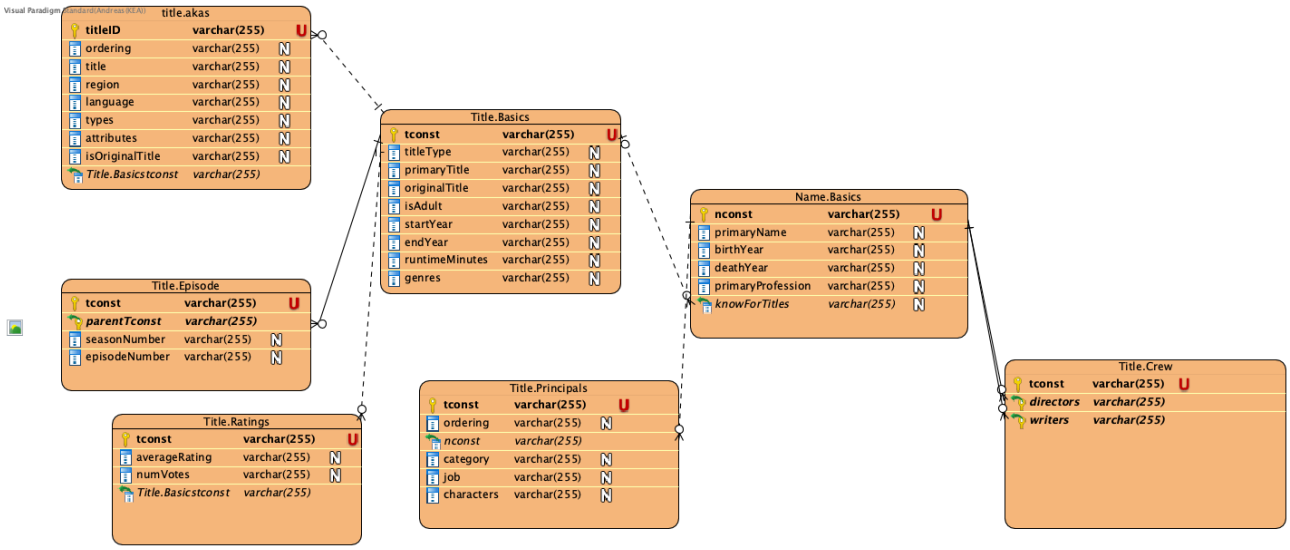
Property	Description
Use case name	Use case 9 - DELETE.
Scope	Database manager.
Level	User goal.
Primary actor	User.
Stakeholder	Developers, programmers, users.
Preconditions	Knowledge of the syntax, know the specific ID of the data you want to delete.
Success guarantee	Filter the data you want to delete with a WHERE clause.
Main Success Scenario	1. User wants to delete an actor. 2. User types in the actors name. 3. The program now tells you how many rows have been affected.

Property	Description
Use case name	Use case 10 - Update row with values
Scope	Database manager
Level	User goal
Primary actor	User goal
Stakeholder	Developers, programmers, users.
Preconditions	Knowledge of the syntax
Success guarantee	Data sheet to see the current tables & their column names
Main Success Scenario	1. User types a valid update statement. 2. The update statement contains the new data for the specified columns chosen. 3. Program returns the amount of rows affected.
Extensions	Filters the update statement with a WHERE classes to limit the rows being affected

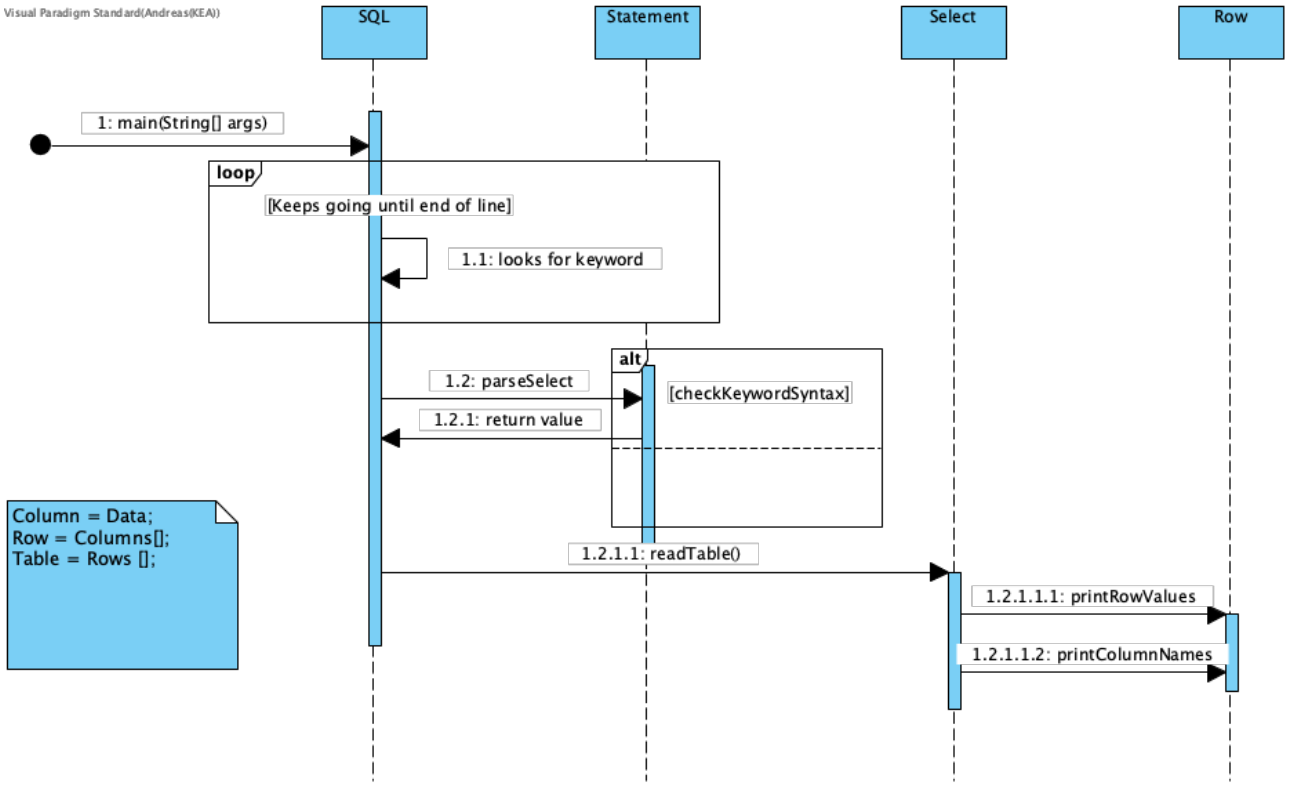
Fully Dressed Use Case:

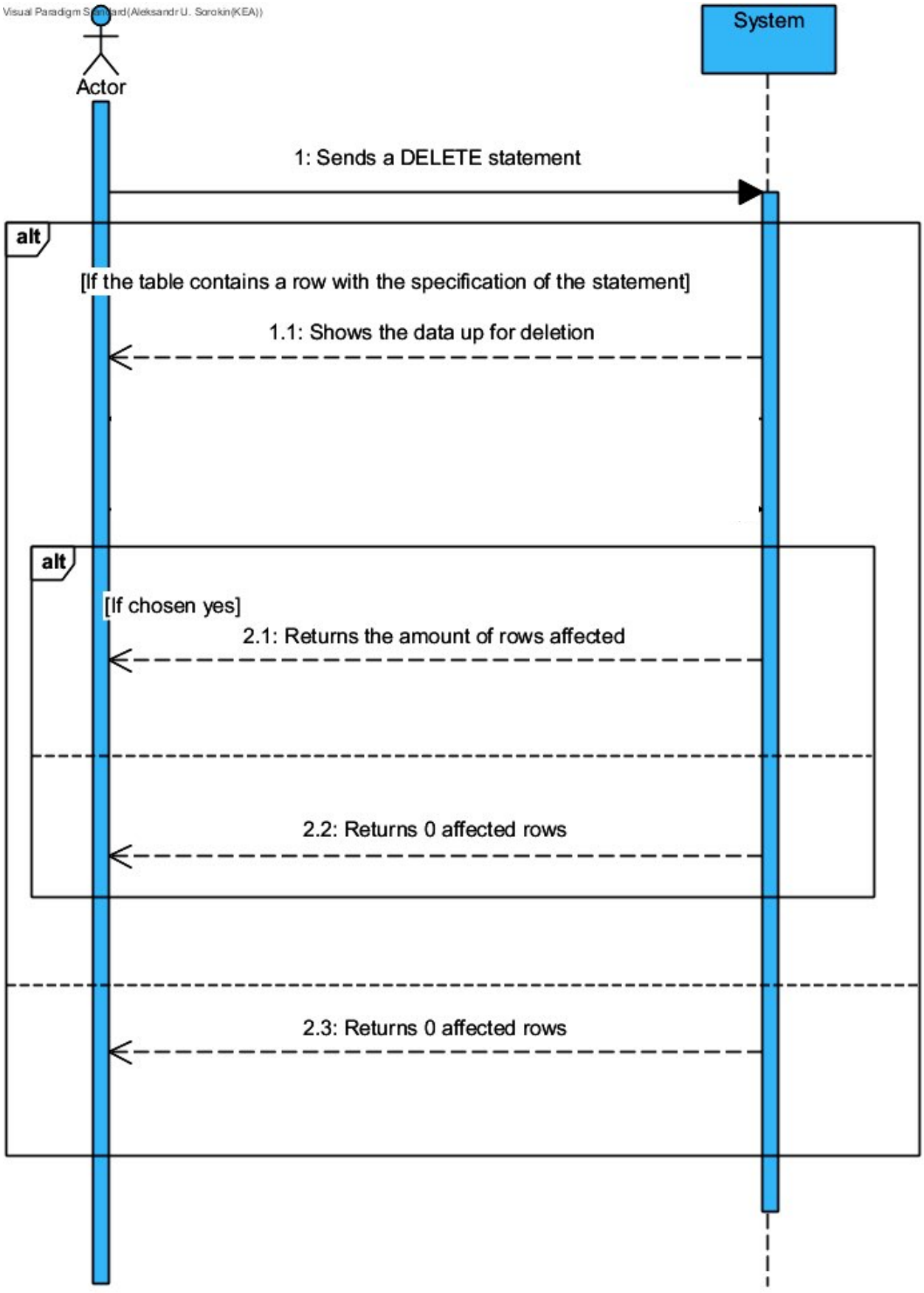
Property	Description
Use case name	Usecase 8 - SELECT
Scope	Database manager
Level	User goal
Primary actor	User
Stakeholders	Developer, user
Preconditions	Datasheets, working search engine
Success guarantee	Accurate output bases on query
Main success scenario	1. User wants to know what year an actor is born 2. User opens the CRUD tool 3. User types in “SELECT birthYear FROM name.basics WHERE primaryName = ‘Leonardo DiCaprio’” 4. The tool returns a table with the name of the actor and what year he is born
Extensions	1. User misspells actor name: a. User receives an error telling him that the system could not find an actor with specified name b. User types in the correct name and receives the wanted output 2. At any time, system fails: a. User restarts the tool and types in the command 3. Actor is not located in the database: a. User receives an error telling him that the system could not find an actor with the specified name b. User can now add actor to the database
Special requirements	1. The user should be able to access the database while offline 2. The user should receive the searched information within 30 seconds
Technology and data Variations list	Data identifier entered by keyboard in the search tool

Entity relation diagram:



Sequence Diagram:




Visual Paradigm Standard(Aleksandr U. Sorokin(KEA))
Actor
System
1: Sends a DELETE statement
alt
[If the table contains a row with the specification of the statement]
1.1: Shows the data up for deletion
alt
[If chosen yes]
2.1: Returns the amount of rows affected
2.2: Returns 0 affected rows
2.3: Returns 0 affected rows

Class diagram: