**Brief Summary of the Project**
This database models the elements of the game Minecraft, including blocks, players, achievements, worlds, and servers. It models how the quantity of different types of objects like blocks, recipes, and mobs may change over time as a player plays in a world. It also represents how a world itself may also be part of a larger server, hosted by certain players and enjoyed by other players.

**Timeline of the Project**
**Assignments:**
Alexi (Purple)
Livia (Blue)
Yoobeen (Pink)
All (White)

**Deadlines:**
EOD March 24 (Red) — SQL structure deadline
EOD March 31 (Orange) — Code and GUI implementation deadline
10PM April 2 (Yellow) — Documentation, deliverables, and Milestone 4 submission deadline

**Tasks:**

| **Task + Details** | **Member** |
|---|---|
| **SQL Structure Deadline – EOD March 24** | |
| Create 1 sql script file (DROP TABLE at beginning of file, contains enough data for testing and demo, no GUI button required to initialize script setup) | ALL |
| **Code and GUI implementation Deadline – EOD March 31** | |
| Implement INSERT on one relation that contains foreign key | Alexi |
| Specify INSERT input values in GUI | Alexi |
| Handle case where foreign key of inserted tuple does not exist in GUI (error message or insert tuple in other table) | Alexi |
| Implement UPDATE on one relation with 2 non-primary key attributes and 1 of non-primary key attribute with UNIQUE constraint or foreign key | Yoobeen |
| Display tuples in GUI that users can UPDATE | Yoobeen |
| Implement DELETE on one relation (should be ON DELETE CASCADE) | Livia |
| User chooses values to delete in GUI (inputting primary key of tuple, selecting from list of tuples, etc.) | Yoobeen |
| Implement SELECT on one relation (condition can be simple “=” equality) | Alexi |

| | |
|---|---|
| Support any number of AND/OR clauses/combinations of attributes in GUI; AND/OR clauses should be implemented as dropdowns (can implement “+” button to “add condition” and drop down that specifies whether to connect with AND or OR) | Alexi |
| Implement PROJECT on one relation | Livia |
| Support viewing any number of attributes in GUI | Livia |
| Implement JOIN between at least 2 relations | Yoobeen |
| User must provide at least one WHERE clause in GUI | Yoobeen |
| Implement AGGREGATION WITH GROUP BY (can be hardcoded) | Alexi |
| Provide interface (button, dropdown, etc.) to execute with GROUP BY in GUI | Alexi |
| Implement AGGREGATION WITH HAVING (can be hardcoded) | Livia |
| Provide interface (button, dropdown, etc.) to execute with HAVING in GUI | Livia |
| Implement NESTED AGGREGATION WITH GROUP BY (aggregation occurs in each nested query, GROUP BY only needs to occur in 1; can be hardcoded) | Yoobeen |
| Provide interface (button, dropdown, etc.) to execute NESTED in GUI | Yoobeen |
| Implement DIVISION (can be hardcoded) | Livia |
| Provide interface (button, dropdown, etc.) to execute DIVISION in GUI | Livia |
| Sanitization (encryption not required) | Yoobeen |
| Miscellaneous error handling notifications in GUI (e.g. inserting duplicate value, invalid input data type, etc.) | ALL |
| GUI notifications about success/failure of INSERT, DELETE, UPDATE and can verify success in GUI | ALL |
| List of needed SQL queries and where they can be found in code (file name and line number(s)) | ALL |
| For AGGREGATIONS and DIVISION, copy of SQL queries and 1-2 sentences describing what each does | ALL |
| **Documentation, deliverables, and Milestone 4 Deadline – EOD April 2** | |
| Submit Milestone 4 deliverables (cover page, sql script file, and pdf file with repository link) | Livia |
| Short description of final project and what it accomplished | Yoobeen |

| | |
|---|---|
| Description of how final schema differed from initial schema and why (IF APPLICABLE) | Alexi |
| Copy of schema and screenshots showing data in each relation after SQL initialization (can be modeled in pdf of excel spreadsheet) | Alexi |
| README file in repository that provides extra information or “No extra information” | Livia |
| Milestone 6 peer and self-evaluation (EVERYONE MUST DO) | ALL |