

Database Report – Wan Shi Tong’s Library

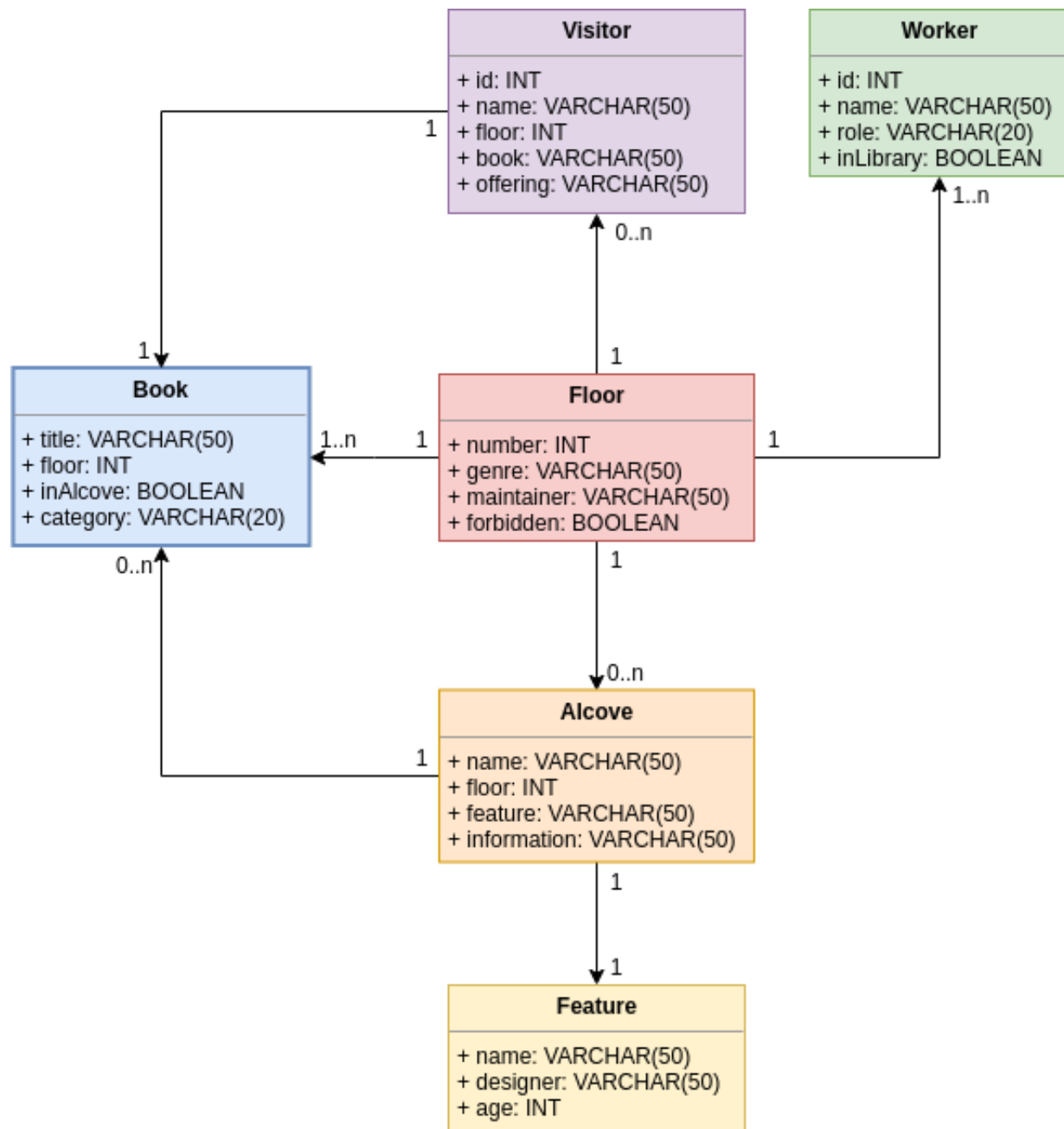
I created this database based on a fictional library. The library houses a wealth of knowledge, including books of fiction, non-fiction, and myth. There are many floors on the library, each floor with a main section for a genre of book, and several alcoves for more specific collections of information. The database describes mainly the structure of the library – a sample of several floors (the size of the library is too vast to model completely) and what is on the floors. The database also tracks which outsiders are inside the library, including information on them, and what they offered to gain entry (the library is private and requires an acceptable offering of knowledge to gain entry).

I will represent the library using tables for floors, rooms, workers, worker roles, visitors, and books. The attributes of each table will be as follows:

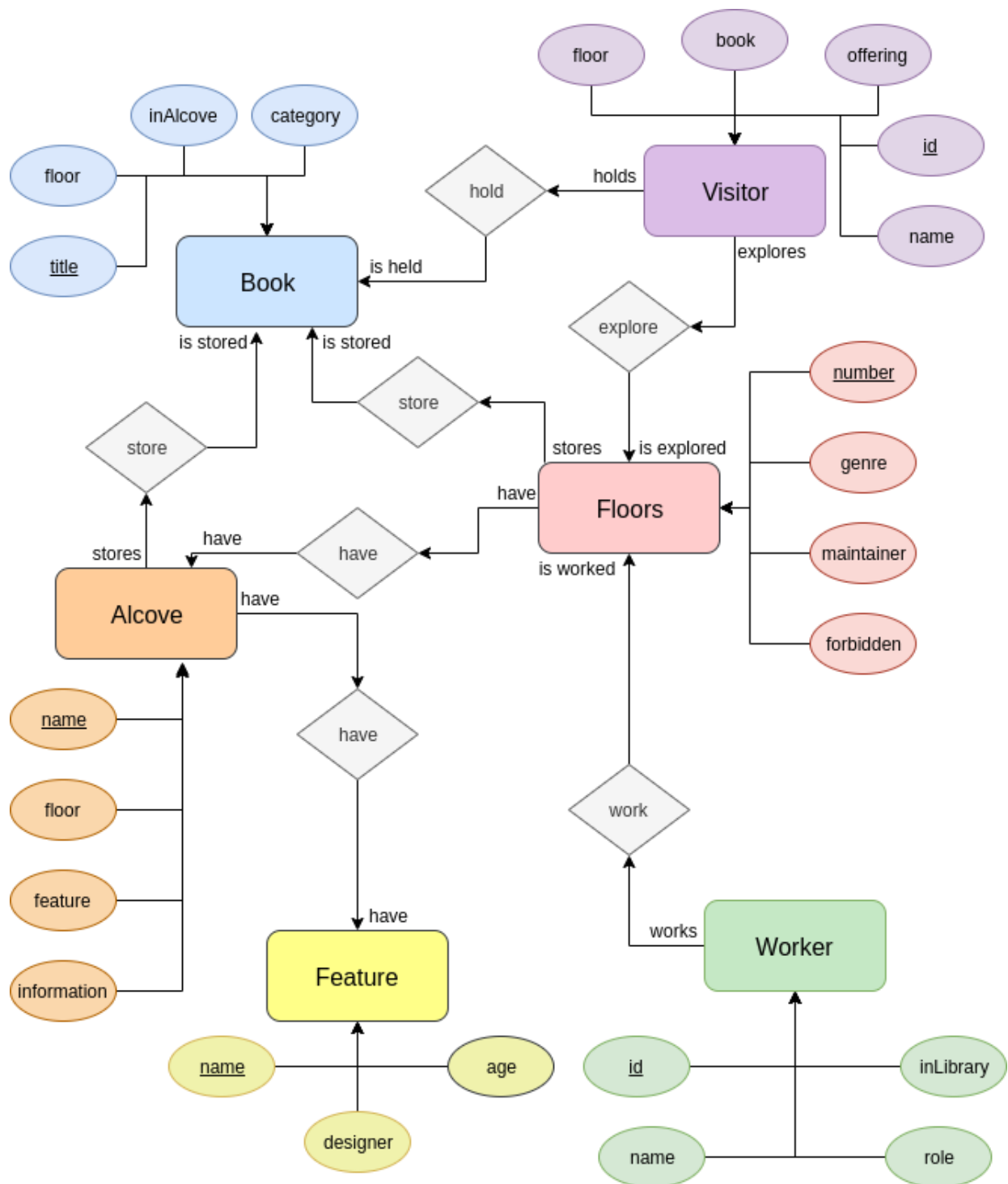
- Floors: A floor has a number (ascending from 1), genre, employee maintaining it, and a forbidden option (if the floor is off-limits to visitors).
- Alcoves: An alcove has a name, a certain floor, a certain unique feature, and a specific category of information.
- Workers: Workers have a name, a floor they work for or are on, a role, and they may be in the library or away from it, gathering knowledge. They also have an id number.
- Visitors: They have a name, they may carry a single book with them, the time they entered the library, where they are, and their admission offering (an offering of knowledge to gain entry to the library, that must be unique). Visitors also have an id number.
- Books: They have a title, floor (which is a genre), they may be in an alcove, and may be fiction, non-fiction, or myth. There are no multiple copies, so the title is unique. We do not store authors in the Book table as it would be information bloat – the author is already contained in the book, and the books are organised by title and genre on the shelves.
- Features: These have a name, a designer, and an age.

There will be a “book_location” view for this database, which will be used to hold locations of books that are being carried by visitors, using relevant information from the Book and Visitors tables. This can be used by employees to find books that are not in their place. As the library is quite unique, the database will always have the relevant location of a visitor. Additionally, there will be a “offering_location” view. This will show where current visitor’s offering is being stored.

Class Diagram

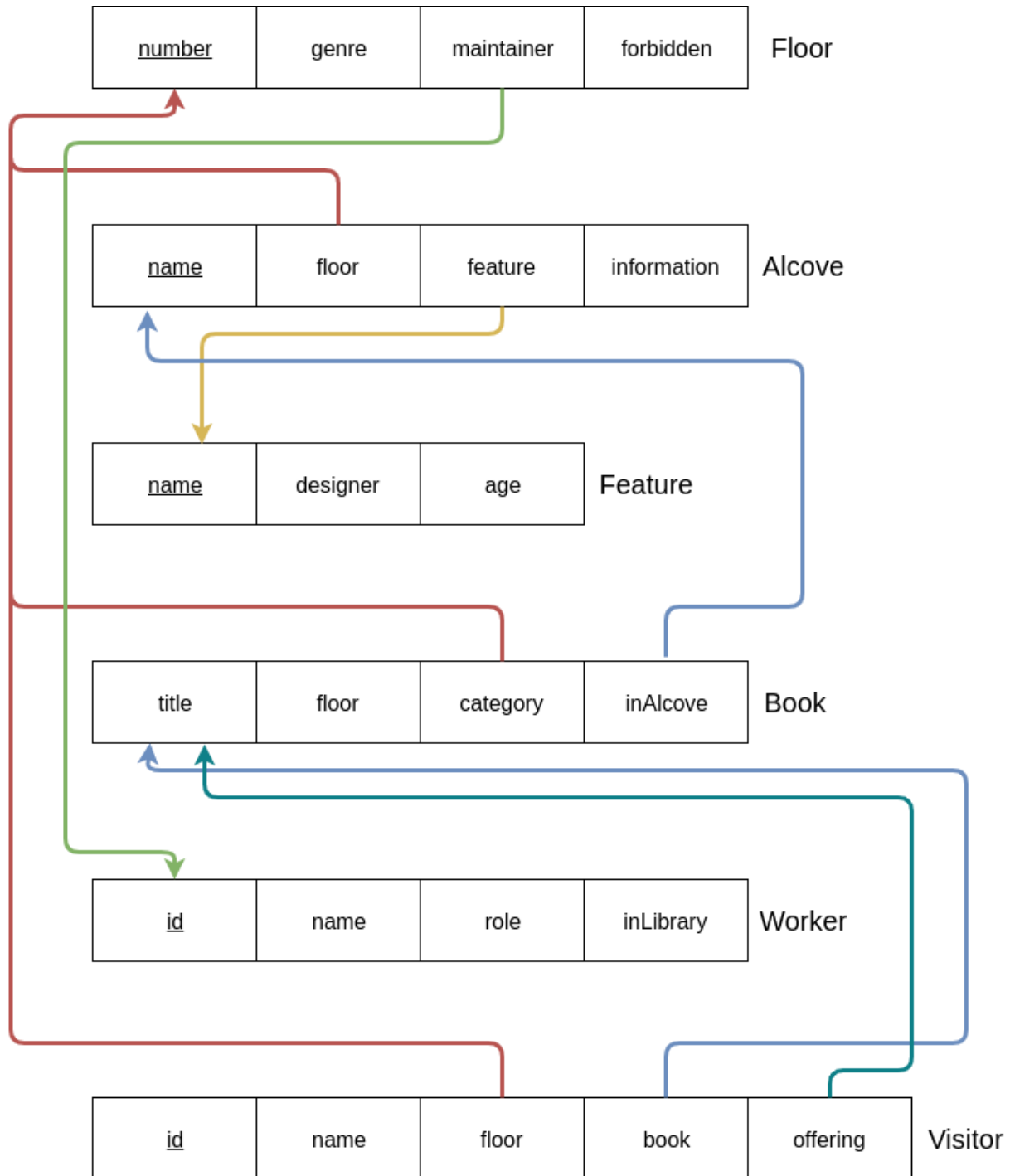


Entity Relationship Diagram



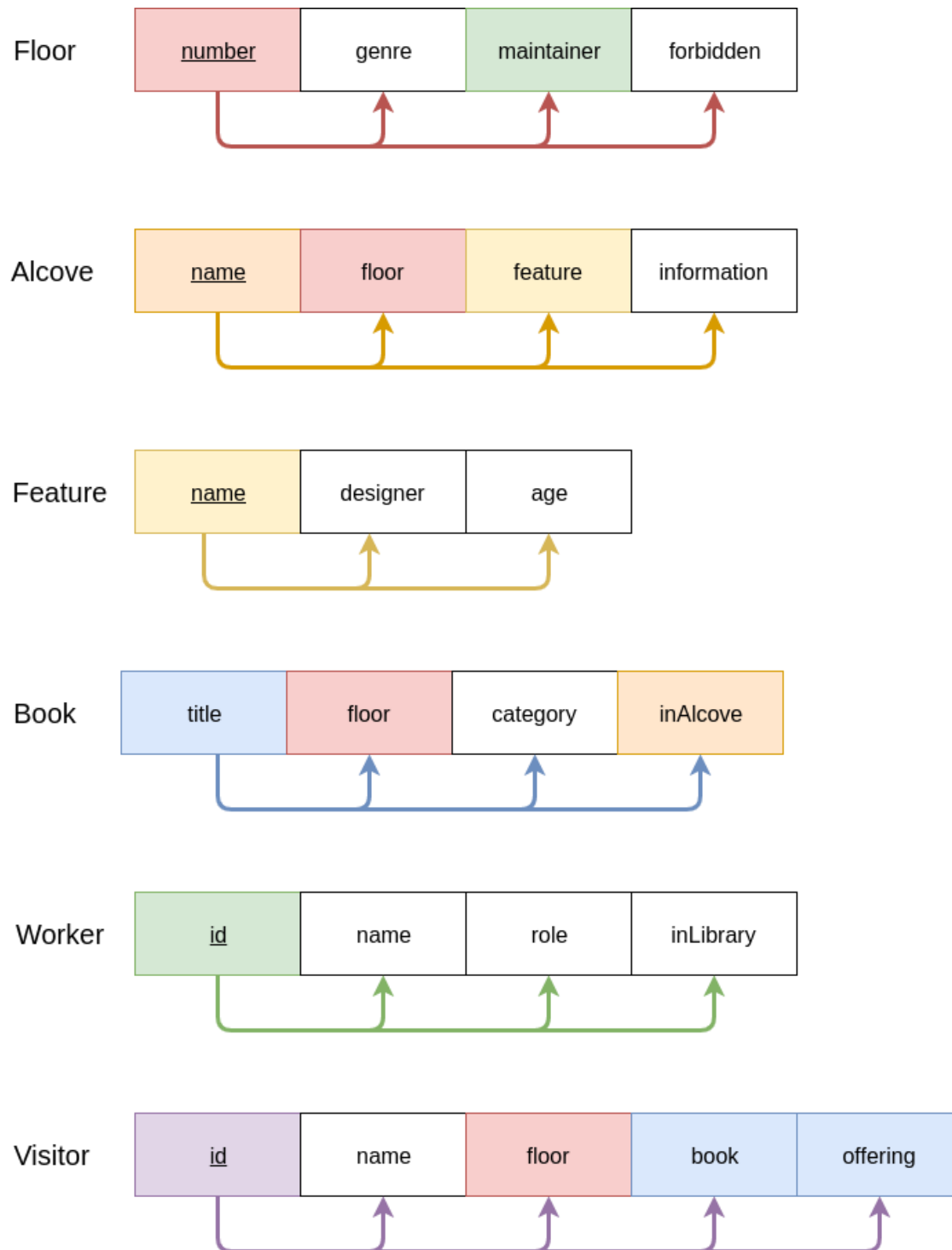
Note: Underlined attributes are Primary Keys.

Relational Schema



Note: Underlined attributes are Primary Keys. Coloured arrows show primary-foreign key links.

Functional Dependency Diagram



Note: Underlined attributes are Primary Keys. Coloured squares in the first column indicate Primary keys, other coloured squares indicate Foreign keys.

Constraints

We employ several simple semantic constraints on this database, mainly in relation to the ID-named Primary Keys. These along with the Floor's 'number' attribute should always be greater than 0. Due to the library being quite large, the positive upper INT range is an acceptable maximum for all other IDs.

The 'category' attribute of the Book table isn't synonymous with genre; the three listed categories for a Book are 'fiction', 'non-fiction', and 'myth'. Myth is the category for books that may be not be fiction but cannot be proved to be non-fiction. We add a constraint to this attribute to accept only one of these three possibilities. This is like the 'role' attribute of Worker.

We also specify a maximum VARCHAR length on insertion of any VARCHAR within the database. Similarly, the rest of the data in the database will be appropriately constrained upon creation by its data type, so there is no need for further constraints here.

We also have NOT NULL options for each Primary key, to ensure that they exist.

Relevant Commands

```
ALTER TABLE Floor ADD CHECK (number > 0);
ALTER TABLE Worker ADD CHECK (id > 0);
ALTER TABLE Visitor ADD CHECK (id > 0);
ALTER TABLE Book ADD CHECK (id > 0);
ALTER TABLE Book ADD CHECK (category IN('fiction','non-
fiction','myth'));
ALTER TABLE Worker ADD CHECK (category
IN('maintainer','gatherer','owner'));
```

Security

Regarding security, the set-up for this database is relevantly simple. The library is private and owned and operated by Wan Shi Tong along with workers that maintain the library and acquire new books. Wan Shi Tong will therefore have all privileges associated with the table, while workers will have only the ability to INSERT new items into the tables to update the library database.

Relevant Commands

```
CREATE ROLE 'owner';
CREATE ROLE 'worker';
GRANT ALL ON 'library.' TO 'owner';
GRANT INSERT ON 'library.' TO 'worker';
GRANT SELECT ON 'library.' TO 'worker';
```

Views

The relevant view we will implement will allow workers to see where a specific book is, given it is with a visitor whose location is stored within the database. While not the most useful view function in reality, there is an underlying assumption that this library is always aware where its visitors are.

```
CREATE OR REPLACE VIEW book_location(book, location) AS SELECT book, floor FROM Visitor WHERE book IS NOT NULL;
```

We can also write a view to show the locations of books that were offered by current visitors. This view works on the principle that each offering must not exist in the library for it to be accepted. An offering is then added to the Book table and joined to the relevant visitor as a Foreign Key.

```
CREATE OR REPLACE VIEW offering_location( book, location) AS SELECT Visitor.offering, Book.floor FROM Visitor, Book WHERE Visitor.offering = Book.title;
```

Updates

I have also included a sample update, to showcase how the database may be updated to reflect the status of certain books, e.g. when one is removed from a shelf.

```
UPDATE Visitor SET Visitor.book = `A History of Nice Pottery` WHERE Visitor.name = `Leo`;
```

Appendix

Create

```
CREATE TABLE `Floor` (  
  `number` int NOT NULL UNIQUE,  
  `genre` varchar(50) UNIQUE,  
  `maintainer` varchar(50) NOT NULL UNIQUE,  
  `forbidden` BOOLEAN NOT NULL DEFAULT 'FALSE',  
  PRIMARY KEY (`number`)  
);  
  
CREATE TABLE `Alcove` (  
  `name` varchar(255) NOT NULL UNIQUE,  
  `floor` int NOT NULL,  
  `Feature` varchar(255) NOT NULL,  
  `information` varchar(255) NOT NULL,  
  PRIMARY KEY (`name`)  
);  
  
CREATE TABLE `Feature` (  
  `name` varchar(255) NOT NULL UNIQUE,  
  `designer` varchar(50),  
  `age` int,  
  PRIMARY KEY (`name`)  
);  
  
CREATE TABLE `Book` (  
  `title` varchar(255) NOT NULL UNIQUE,  
  `floor` int NOT NULL,  
  `category` varchar(16) NOT NULL,  
  `inAlcove` varchar(255) DEFAULT 'NULL',  
  PRIMARY KEY (`title`)  
);  
  
CREATE TABLE `Worker` (  
  `id` int NOT NULL UNIQUE,  
  `name` varchar(50) NOT NULL,  
  `role` varchar(16),  
  `inLibrary` BOOLEAN NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
CREATE TABLE `Visitor` (  
  `id` int NOT NULL UNIQUE,  
  `name` varchar(50) NOT NULL,  
  `floor` int NOT NULL,  
  `book` varchar(255),  
  `offering` varchar(255) NOT NULL UNIQUE,  
  PRIMARY KEY (`id`)  
);
```


Alter

```
ALTER TABLE `Floor` ADD FOREIGN KEY (`maintainer`) REFERENCES `Worker`(`id`);

ALTER TABLE `Alcove` ADD FOREIGN KEY (`floor`) REFERENCES `Floor`(`number`);

ALTER TABLE `Alcove` ADD FOREIGN KEY (`feature`) REFERENCES `Feature`(`name`);

ALTER TABLE `Book` ADD FOREIGN KEY (`floor`) REFERENCES `Floor`(`number`);

ALTER TABLE `Book` ADD FOREIGN KEY (`inAlcove`) REFERENCES `Alcove`(`name`);

ALTER TABLE `Visitor` ADD FOREIGN KEY (`floor`) REFERENCES `Floor`(`number`);

ALTER TABLE `Visitor` ADD FOREIGN KEY (`book`) REFERENCES `Book`(`title`);

ALTER TABLE `Visitor` ADD FOREIGN KEY (`offering`) REFERENCES `Book`(`title`);
```

Insert

```
INSERT INTO Worker VALUES(1,`Wan Shi Tong`, `owner`, TRUE);
INSERT INTO Worker VALUES(2,`Alex`, `gatherer`, FALSE);
INSERT INTO Worker VALUES(3,`Samantha`, `maintainer`, TRUE);
INSERT INTO Worker VALUES(4,`Peter`, `gatherer`, TRUE);
INSERT INTO Worker VALUES(5,`Gary`, `maintainer`, TRUE);
INSERT INTO Worker VALUES(6,`Andrew`, `maintainer`, TRUE);
INSERT INTO Worker VALUES(7,`Joshua`, `gatherer`, FALSE);
INSERT INTO Worker VALUES(8,`Joey`, `maintainer`, TRUE);
INSERT INTO Worker VALUES(9,`Georgia`, `maintainer`, TRUE);

INSERT INTO Floor VALUES(1, `Ancient History of the Earth Kingdom`, `Gary`, FALSE);
INSERT INTO Floor VALUES(2, `Tribe of the North Pole`, `Andrew`, TRUE);
INSERT INTO Floor VALUES(3, `Southern Temples and Antiquities`, `Georgia`, FALSE);
INSERT INTO Floor VALUES(4, `Stories from the 2nd Century`, `Joey`, FALSE);
INSERT INTO Floor VALUES(5, `Legends of the Western Swamps`, `Samantha`, FALSE);

INSERT INTO Feature VALUES(`A Natural Phenomenon`, NULL, 131);
INSERT INTO Feature VALUES(`The First Tools`, NULL, NULL);
INSERT INTO Feature VALUES(`Ba Sing Se: A City Divided`, `Marcus Zhang`, 14);
INSERT INTO Feature VALUES(`Monk Statue #17`, NULL, 268);
INSERT INTO Feature VALUES(`The Spirit Escapes`, `Julian Haru`, 1);

INSERT INTO Alcove VALUES(`The Aurora Room`, 2, `A Natural Phenomenon`, `About the Northern Lights`);
INSERT INTO Alcove VALUES(`The Clay Tool Room`, 1, `The First Tools`, `On the oldest tools of the Earth Kingdom`);
INSERT INTO Alcove VALUES(`The Great Divide Room`, 1, `Ba Sing Se: A City Divided`, `On the Architecture of the Ba Sing Se Wall`);
INSERT INTO Alcove VALUES(`The Room of Southern Monks`, 3, `Monk Statue #17`, `The life of Southern Temple monks`);
INSERT INTO Alcove VALUES(`The Room of the Swamp Monster`, 5, NULL, `Is the swamp monster fact or fiction?`);
INSERT INTO Alcove VALUES(`The Room of the The Blue Spirit`, 4, `The Spirit Escapes`, `On the stories of the Blue Spirit`);
```

```
INSERT INTO Book VALUES(`Tales of the Full Moon`,2,`fiction`,NULL);
INSERT INTO Book VALUES(`People of the Vines`,5,`myth`,`The Room of the Swamp
Monster`);
INSERT INTO Book VALUES(`Architect of the Wall`,1,`non-fiction`,`The Great Divide
Room`);
INSERT INTO Book VALUES(`The Impact of the Caste System on the Ba Sing Se
Economy`,1,`non-fiction`, NULL);
INSERT INTO Book VALUES(`Mating Behaviour of Sky Bisons in the Southern
Temples`,3,`non-fiction`,NULL);
INSERT INTO Book VALUES(`The Use of High-Altitude Winds to create Music`,3,`non-
fiction`,NULL);
INSERT INTO Book VALUES(`The Life of Monk Kiatsu: A Biography`,3,`non-fiction`,`The
Room of Southern Monks`);

INSERT INTO Visitor VALUES(1, `Leo`, 1, NULL,`Mating Behaviour of Sky Bisons in the
Southern Temples`);
INSERT INTO Visitor VALUES(2, `Ryan`, 3, NULL,`People of the Vines`);
INSERT INTO Visitor VALUES(3, `Luke`, 3, `Mating Behaviour of Sky Bisons in the
Southern Temples`,`Architect of the Wall`);
INSERT INTO Visitor VALUES(4, `Sarah`, 4, `Architect of the Wall`,`The Impact of the
Caste System on the Ba Sing Se Economy`);
INSERT INTO Visitor VALUES(5, `Liza`, 5, NULL,`The Life of Monk Kiatsu: A Biography`);
```