1. SQL Server

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| id | int | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор пользователя |
| name | nvarchar(50) | NOT NULL | Имя пользователя |
| last\_name | nvarchar(50) | NOT NULL | Фамилия пользователя |
| email | nvarchar(100) | UNIQUE, NOT NULL | Email пользователя |
| hash | nvarchar(255) | NOT NULL | Хеш пароля |
| role | nvarchar(50) | NOT NULL, CHECK (role IN ('admin', 'user', 'employee')) | Роль пользователя |

Таблица Users

Таблица Genres

|  |  |  |  |
| --- | --- | --- | --- |
| id | INT | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор жанра. |
| name | NVARCHAR(50) | UNIQUE, NOT NULL | Название жанра. |

Таблица Book

|  |  |  |  |
| --- | --- | --- | --- |
| id\_book | INT | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор книги |
| title | NVARCHAR(255) | NOT NULL | Название книги |
| price | DECIMAL(10,2) | NOT NULL | Цена книги |
| stock | INT | NOT NULL, DEFAULT 0 | Количество в наличии |
| genre\_id | INT | FOREIGN KEY REFERENCES Genres(id) ON DELETE SET NULL | Идентификатор жанра. |
| description | NVARCHAR(MAX) | NOT NULL | Описание книги |
| publication\_date | DATE | NOT NULL | Дата публикации. |
| author\_id | INT | FOREIGN KEY REFERENCES Authors(id) ON DELETE SET NULL | Идентификатор автора. |

Таблица Authors

|  |  |  |  |
| --- | --- | --- | --- |
| id | INT | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор жанра. |
| name | NVARCHAR(50) | NOT NULL | Название жанра. |
| biography | (NVARCHAR(MAX | NOT NULL | Биография автора |

Таблица Employee

|  |  |  |  |
| --- | --- | --- | --- |
| id | INT | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор сотрудника. |
| name | NVARCHAR(50) | NOT NULL | Имя сотрудника. |
| last\_name | NVARCHAR(50) | NOT NULL | Фамилия сотрудника. |
| job | NVARCHAR(100) | NOT NULL | Должность |

Таблица Orders

|  |  |  |  |
| --- | --- | --- | --- |
| id\_order | INT | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор заказа |
| user\_id | INT | FOREIGN KEY REFERENCES Users(id) ON DELETE CASCAD | Идентификатор пользователя |
| price | DECIMAL(10,2) | NOT NULL | Итоговая цена заказ |
| count | INT | NOT NULL | Количество товаров в заказе |
| employee\_id | INT | FOREIGN KEY REFERENCES Employee(id) ON DELETE SET NULL | Идентификатор сотрудник |

Таблица OrderDetails

|  |  |  |  |
| --- | --- | --- | --- |
| id | INT | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор записи |
| order\_id | INT | FOREIGN KEY REFERENCES Orders(id\_order) ON DELETE CASCADE | Идентификатор заказа |
| product\_id | INT | FOREIGN KEY REFERENCES Book(id\_book) ON DELETE CASCADE | Идентификатор книги |
| count | INT | NOT NULL | Количество заказанных единиц |
| date\_order | DATETIME | DEFAULT GETDATE() | Дата заказа |

Таблица Payments

|  |  |  |  |
| --- | --- | --- | --- |
| id | INT | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор платежа. |
| order\_id | INT | FOREIGN KEY REFERENCES Orders(id\_order) ON DELETE CASCADE | Идентификатор заказа |
| sum | DECIMAL(10,2) | NOT NULL | Сумма платеж |

1. Индексы

-- Индексы для таблицы genres

CREATE INDEX idx\_genres\_name ON genres (name);

-- Индексы для таблицы authors

CREATE INDEX idx\_authors\_name ON authors (name);

-- Индексы для таблицы book

CREATE INDEX idx\_book\_title ON book (title);

CREATE INDEX idx\_book\_genre ON book (genre\_id);

CREATE INDEX idx\_book\_author ON book (author\_id);

CREATE INDEX idx\_book\_price ON book (price);

CREATE INDEX idx\_book\_stock ON book (stock);

-- Покрывающий индекс для book с дополнительными полями (без CLOB)

CREATE INDEX idx\_book\_title\_covering ON book (title, id\_book, price, stock, genre\_id, author\_id, publication\_date);

-- Индексы для таблицы employee

CREATE INDEX idx\_employee\_name ON employee (name);

CREATE INDEX idx\_employee\_last\_name ON employee (last\_name);

-- Индексы для таблицы Orders

CREATE INDEX idx\_orders\_user ON Orders (user\_id);

CREATE INDEX idx\_orders\_employee ON Orders (employee\_id);

CREATE INDEX idx\_orders\_price ON Orders (price);

CREATE INDEX idx\_orders\_count ON Orders (count);

-- Индексы для таблицы orderdetails

CREATE INDEX idx\_orderdetails\_order ON orderdetails (order\_id);

CREATE INDEX idx\_orderdetails\_product ON orderdetails (product\_id);

-- Покрывающий индекс для orderdetails с дополнительными полями

CREATE INDEX idx\_orderdetails\_order\_covering ON orderdetails (order\_id, product\_id, count, date\_order);

-- Индексы для таблицы payments

CREATE INDEX idx\_payments\_order ON payments (order\_id);

-- Если нужно индексировать колонку типа CLOB (например, description в таблице book), используйте полнотекстовый индекс

CREATE INDEX idx\_book\_description ON book (description)

INDEXTYPE IS CTXSYS.CONTEXT;

-- Дополнительные составные индексы (composite indexes) для повышения производительности запросов

-- Составной индекс для Orders с дополнительными полями

CREATE INDEX idx\_orders\_user\_covering ON Orders (user\_id, id\_order, price, count, employee\_id);

-- Составной индекс для book с дополнительными полями

CREATE INDEX idx\_book\_genre\_author\_covering ON book (genre\_id, author\_id, id\_book, price, stock);

2. Представления (Views)

vw\_books – Представление книг с жанрами и авторами.

vw\_orders – Представление заказов с информацией о пользователях и сотрудниках.

3. Последовательности (Sequences)

seq\_users – Последовательность для идентификаторов пользователей.

seq\_books – Последовательность для идентификаторов книг.

seq\_orders – Последовательность для идентификаторов заказов.

Процедуры:

1. add\_user - Добавляет нового пользователя в таблицу Users, с указанными именем, фамилией, электронной почтой, паролем и ролью.
2. create\_order - Создает новый заказ в таблице Orders для указанного пользователя и сотрудника с заданной стоимостью и количеством товаров.
3. delete\_order - Удаляет заказ по его ID, сначала удаляя детали заказа из orderdetails, а затем сам заказ из Orders.
4. get\_order\_info - Извлекает и выводит информацию о заказе по его ID, включая данные о пользователе, стоимости, количестве товаров и сотруднике, который обработал заказ.
5. add\_genre - Добавляет новый жанр в таблицу genres.

Функции:

1. get\_order\_total - Рассчитывает полную стоимость заказа, включая количество и цену всех товаров в заказе.
2. get\_books\_count\_by\_genre - Возвращает количество книг, относящихся к указанному жанру.
3. get\_full\_user\_name - Возвращает полное имя пользователя, объединяя его имя и фамилию.
4. get\_total\_items\_in\_order - Возвращает общее количество товаров, заказанных в конкретном заказе.
5. get\_full\_book\_name - Возвращает полное имя книги, состоящее из названия книги и имени автора.

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| id | NUMBER | GENERATED BY DEFAULT AS IDENTITY, PRIMARY KEY | Уникальный идентификатор пользователя |
| name | nvarchar2(50) | NOT NULL | Имя пользователя |
| last\_name | nvarchar2(50) | NOT NULL | Фамилия пользователя |
| email | nvarchar2(100) | UNIQUE, NOT NULL | Email пользователя |
| hash | nvarchar2(255) | NOT NULL | Хеш пароля |
| role | nvarchar2(50) | NOT NULL, CHECK (role IN ('admin', 'user', 'employee')) | Роль пользователя |

Oracle

Таблица Users

Таблица Genres

|  |  |  |  |
| --- | --- | --- | --- |
| id | NUMBER | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор жанра. |
| name | NVARCHAR2(50) | UNIQUE, NOT NULL | Название жанра. |

Таблица Book

|  |  |  |  |
| --- | --- | --- | --- |
| id\_book | NUMBER | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор книги |
| title | NVARCHAR2(255) | NOT NULL | Название книги |
| price | DECIMAL(10,2) | NOT NULL | Цена книги |
| stock | NUMBER | NOT NULL, DEFAULT 0 | Количество в наличии |
| genre\_id | NUMBER | FOREIGN KEY REFERENCES Genres(id) ON DELETE SET NULL | Идентификатор жанра. |
| description | NVARCHAR2(MAX) | NOT NULL | Описание книги |
| publication\_date | DATE | NOT NULL | Дата публикации. |
| author\_id | NUMBER | FOREIGN KEY REFERENCES Authors(id) ON DELETE SET NULL | Идентификатор автора. |

Таблица Authors

|  |  |  |  |
| --- | --- | --- | --- |
| id | NUMBER | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор жанра. |
| name | NVARCHAR2(50) | NOT NULL | Название жанра. |
| biography | (NVARCHAR2(MAX | NOT NULL | Биография автора |

Таблица Authors

|  |  |  |  |
| --- | --- | --- | --- |
| id | NUMBER | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор жанра. |
| name | NVARCHAR2(50) | NOT NULL | Название жанра. |
| biography | (NVARCHAR2(MAX | NOT NULL | Биография автора |

Таблица Employee

|  |  |  |  |
| --- | --- | --- | --- |
| id | NUMBER | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор сотрудника. |
| name | NVARCHAR2(50) | NOT NULL | Имя сотрудника. |
| last\_name | NVARCHAR2(50) | NOT NULL | Фамилия сотрудника. |
| job | NVARCHAR2(100) | NOT NULL | Должность |

Таблица Orders

|  |  |  |  |
| --- | --- | --- | --- |
| id\_order | NUMBER | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор заказа |
| user\_id | NUMBER | FOREIGN KEY REFERENCES Users(id) ON DELETE CASCAD | Идентификатор пользователя |
| price | DECIMAL(10,2) | NOT NULL | Итоговая цена заказ |
| count | NUMBER | NOT NULL | Количество товаров в заказе |
| employee\_id | NUMBER | FOREIGN KEY REFERENCES Employee(id) ON DELETE SET NULL | Идентификатор сотрудник |

Таблица OrderDetails

|  |  |  |  |
| --- | --- | --- | --- |
| id | NUMBER | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор записи |
| order\_id | NUMBER | FOREIGN KEY REFERENCES Orders(id\_order) ON DELETE CASCADE | Идентификатор заказа |
| product\_id | NUMBER | FOREIGN KEY REFERENCES Book(id\_book) ON DELETE CASCADE | Идентификатор книги |
| count | NUMBER | NOT NULL | Количество заказанных единиц |
| date\_order | DATETIME | DEFAULT GETDATE() | Дата заказа |

Таблица Payments

|  |  |  |  |
| --- | --- | --- | --- |
| id | NUMBER | PRIMARY KEY, IDENTITY(1,1) | Уникальный идентификатор платежа. |
| order\_id | NUMBER | FOREIGN KEY REFERENCES Orders(id\_order) ON DELETE CASCADE | Идентификатор заказа |
| sum | DECIMAL(10,2) | NOT NULL | Сумма платеж |

1. Индексы

-- Индексы для таблицы Users

CREATE INDEX idx\_users\_email ON Users (email);

CREATE INDEX idx\_users\_role ON Users (role);

-- Индексы для таблицы genres

CREATE INDEX idx\_genres\_name ON genres (name);

-- Индексы для таблицы authors

CREATE INDEX idx\_authors\_name ON authors (name);

-- Индексы для таблицы book

CREATE INDEX idx\_book\_title ON book (title);

CREATE INDEX idx\_book\_genre ON book (genre\_id);

CREATE INDEX idx\_book\_author ON book (author\_id);

CREATE INDEX idx\_book\_price ON book (price);

CREATE INDEX idx\_book\_stock ON book (stock);

-- Покрывающий индекс для book с дополнительными полями (без CLOB)

CREATE INDEX idx\_book\_title\_covering ON book (title, id\_book, price, stock, genre\_id, author\_id, publication\_date);

-- Индексы для таблицы employee

CREATE INDEX idx\_employee\_name ON employee (name);

CREATE INDEX idx\_employee\_last\_name ON employee (last\_name);

-- Индексы для таблицы Orders

CREATE INDEX idx\_orders\_user ON Orders (user\_id);

CREATE INDEX idx\_orders\_employee ON Orders (employee\_id);

CREATE INDEX idx\_orders\_price ON Orders (price);

CREATE INDEX idx\_orders\_count ON Orders (count);

-- Индексы для таблицы orderdetails

CREATE INDEX idx\_orderdetails\_order ON orderdetails (order\_id);

CREATE INDEX idx\_orderdetails\_product ON orderdetails (product\_id);

-- Покрывающий индекс для orderdetails с дополнительными полями

CREATE INDEX idx\_orderdetails\_order\_covering ON orderdetails (order\_id, product\_id, count, date\_order);

-- Индексы для таблицы payments

CREATE INDEX idx\_payments\_order ON payments (order\_id);

-- Если нужно индексировать колонку типа CLOB (например, description в таблице book), используйте полнотекстовый индекс

CREATE INDEX idx\_book\_description ON book (description)

INDEXTYPE IS CTXSYS.CONTEXT;

-- Дополнительные составные индексы (composite indexes) для повышения производительности запросов

-- Составной индекс для Orders с дополнительными полями

CREATE INDEX idx\_orders\_user\_covering ON Orders (user\_id, id\_order, price, count, employee\_id);

-- Составной индекс для book с дополнительными полями

CREATE INDEX idx\_book\_genre\_author\_covering ON book (genre\_id, author\_id, id\_book, price, stock);

2. Представления (Views)

vw\_books – Представление книг с жанрами и авторами.

vw\_orders – Представление заказов с информацией о пользователях и сотрудниках.

3. Последовательности (Sequences)

seq\_users – Последовательность для идентификаторов пользователей.

seq\_books – Последовательность для идентификаторов книг.

seq\_orders – Последовательность для идентификаторов заказов.

Процедуры:

1. add\_user - Добавление нового пользователя в таблицу Users.
2. create\_order - Создание нового заказа в таблице Orders, с привязкой к пользователю и сотруднику.
3. delete\_order - Удаление заказа с его деталями из таблиц Orders и orderdetails по ID заказа.
4. get\_order\_info - Получение информации о заказе (пользователь, цена, количество, сотрудник), вывод через DBMS\_OUTPUT.
5. add\_genre - Добавление нового жанра в таблицу genres.

Функции:

1. get\_order\_total - Рассчитывает полную стоимость заказа, суммируя цену товаров в заказе (с учетом количества).
2. get\_books\_count\_by\_genre - Получает количество книг в определенном жанре.
3. is\_book\_in\_stock - Проверка наличия книги на складе (возвращает 1, если в наличии, и 0, если нет).
4. get\_full\_user\_name - Получение полного имени пользователя (имя и фамилия) по его ID.
5. get\_total\_items\_in\_order - Получение общего количества товаров в заказе по его ID.