Санкт-Петербургский национально исследовательский университет

информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники



**Этап №3 курсовой работы по предмету**

**“Информационные системы и базы данных”**

Выполнил: Балтабаев Дамир

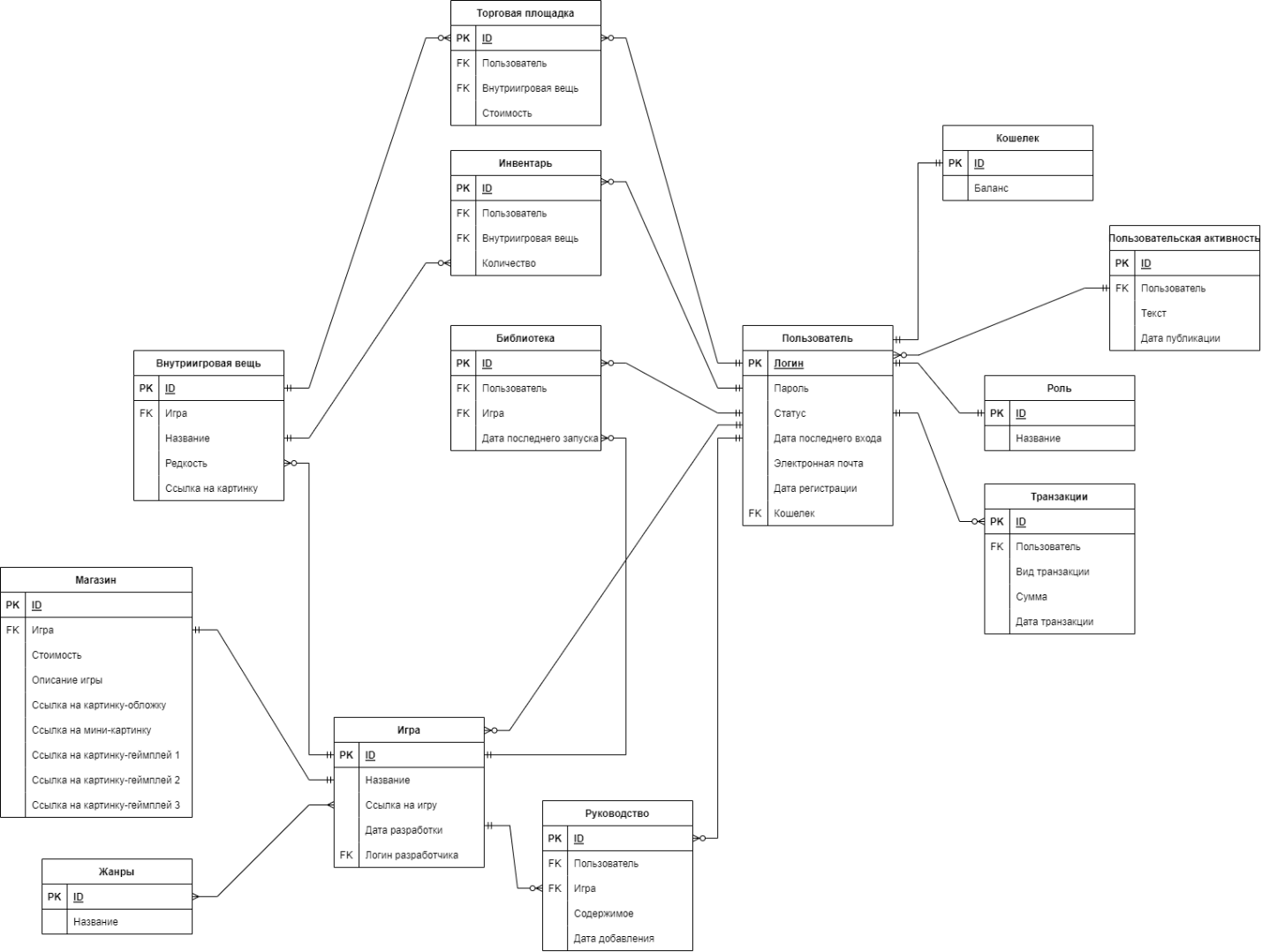
Группа: P33121

Преподаватель: Шешуков Дмитрий Михайлович

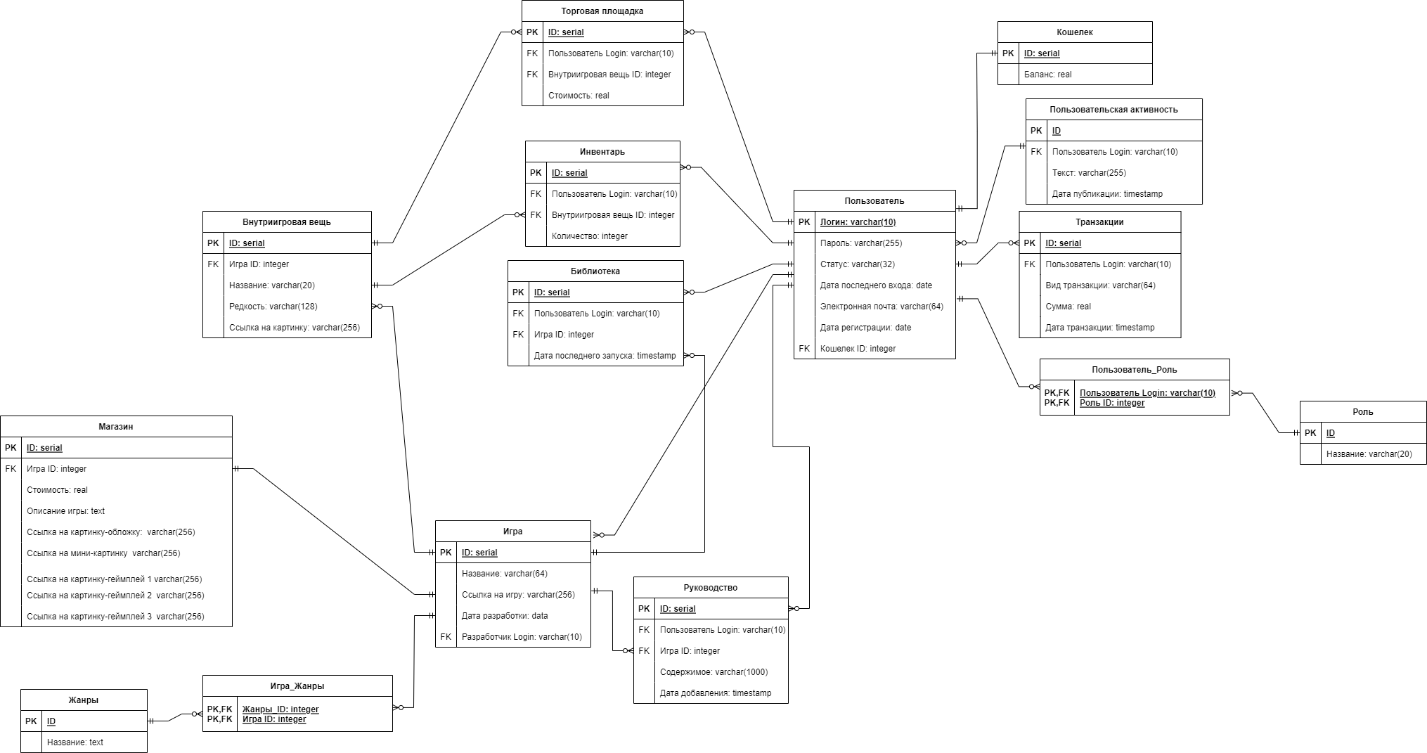
г. Санкт-Петербург

2022

**Инфологическая модель**

****

**Даталогическая модель**

****

**Скрипты**

https://github.com/damir2407/db\_c3

**Триггеры:**

"checkGameOnUnique" – В таблице игра должна быть уникальная запись с игрой

"checkShopOnUnique" – В таблице магазин должна быть уникальная запись с игрой

"checkLibraryOnUnique" – В таблице библиотека должна быть уникальная запись с игрой

"checkInventoryOnUnique" – В таблице инвентарь должна быть уникальная запись с игрой

"checkMarketOnSell" – Пользователь может продавать только те вещи, которые есть у него в инвентаре

**Функции и процедуры:**

"loginAsUser"(arg\_login varchar(10)) - Процедура для входа в систему за пользователя

"logoutFromUser "(arg\_login varchar(10)) – Процедура для выхода пользователя из системы

"checkGameOnUniqueFunction"() - Триггерная-функция для проверки на уникальность записи в игре

"checkShopOnUniqueFunction"() - Триггерная-функция для проверки на уникальность записи в магазине

"getGameNameById "(id\_arg integer) - Функция для получения названия игры по ее ID

"replenishbalance "(arg\_login varchar(10), arg\_balance double precision) - Функция для пополнения баланса кошелька

"enterTheGame "(arg\_login varchar(10), game\_name varchar(64)) - Процедура для запуска игры из библиотеки пользователя

"checkInventoryOnUniqueFunction "() - Триггерная-функция для проверки на уникальность записи в инвентаре

"sellItemOnMarket"() - Триггерная функция для проверки может ли пользователь продать вещь, а также удаления/изменение записей в Инвентаре пользователя

"replenishbalanceForSoldItem "(arg\_login varchar(10), arg\_balance double precision) -Функция для пополнения баланса кошелька на сумму проданной вещи

"chargebalanceForSoldItem "(arg\_login varchar(10), arg\_balance double precision) – Функция для списания баланса кошелька на сумму купленной вещи

**Индексы:**

**Инвентарь\_index** – btree индекс созданный для объекта “Инвентарь” по атрибутам "Логин\_Пользователя", "Внутриигровая\_вещь ID". Необходим для ускорения поиска записей. Для реализации данного индекса лучше всего подходит hash index, т.к запросы, которые используются у меня в функциях и процедурах используют только операнд ‘=’, а hash index как известно лучший для данного вида операций. Но из-за ограничения hash индекса на создание индексации по нескольким атрибутам я решил использовать btree index. Конечно в таком случае я мог создать два hash индекса на каждый из столбцов, но мне кажется это проиграет по времени одному btree индексу, т.к потребуется еще память для хранения еще одного hash индекса, а также при обновлении/удалении индексируемых столбцов будут меняться значения в обоих индексах.

**Библиотека\_index** – unique btree индекс созданный для объекта “Библиотека” по атрибутам "Логин\_Пользователя", "Игра ID". Необходим для ускорения поиска записей. Для реализации данного индекса лучше всего подходит hash index, т.к запросы, которые используются у меня в функциях и процедурах используют только операнд ‘=’, а hash index как известно лучший для данного вида операций. Но из-за ограничения hash индекса на создание индексации по нескольким атрибутам я решил использовать btree index. Конечно в таком случае я мог создать два hash индекса на каждый из столбцов, но мне кажется это проиграет по времени одному btree индексу, т.к потребуется еще память для хранения еще одного hash индекса, а также при обновлении/удалении индексируемых столбцов будут меняться значения в обоих индексах. Также использовал ключевое слово unique для того чтобы отметить, что индексируемые значения будут уникальными.

**Кошелек\_index** – hash индекс созданный для объекта “Кошелек” по атрибуту "Логин\_Пользователя". Необходим для ускорения поиска записей. Для реализации данного индекса лучше всего подходит hash index, т.к запросы, которые используются у меня в функциях и процедурах используют только операнд ‘=’, а hash index как известно лучший для данного вида операций.

**Игра\_index** – hash индекс созданный для объекта “Игра” по атрибуту "Название". Необходим для ускорения поиска записей. Для реализации данного индекса лучше всего подходит hash index, т.к запросы, которые используются у меня в функциях и процедурах используют только операнд ‘=’, а hash index как известно лучший для данного вида операций.

**Магазин\_index** – hash индекс созданный для объекта “Магазин” по атрибуту "Игра\_ID". Необходим для ускорения поиска записей. Для реализации данного индекса лучше всего подходит hash index, т.к запросы, которые используются у меня в функциях и процедурах используют только операнд ‘=’, а hash index как известно лучший для данного вида операций.

**Анализ**

Наиболее часто используемая информация хранится в таблицах “Пользователь”, “Игра”, “Библиотека”, “Инвентарь”. Дело в том, что в моей предметной области главными объектами являются Пользователь и Игра. Большое количество запросов по заданным таблицам можно объяснить следующим:

1. Пользователь – основной объект, который производит большинство действий. Он может купить игру в магазине, продать вещь на Торговой площадке, купить вещь на торговой площадке, вступать в группы и т.д. Соответственно поэтому большое количество запросов связано с таблицей “Пользователь”.
2. Игра – не менее важный объект, на котором завязана реализация БД. Почти все что есть в БД – связано с играми. Их можно покупать и хранить в библиотеке, с них могут выпасть вещи, их можно запускать, на них можно писать руководства и создавать моды. Именно поэтому такое большое количество запросов связанных с этой таблицей
3. Библиотека – объект, часто подвергающийся проверкам и изменениям, то-есть запросам. В библиотеке хранятся купленные игры, а также эти игры запускаются именно с библиотеки пользователя.

* Пользователи не могут покупать игры, которые и так есть у них в библиотеке,
* Пользователи не могут опубликовывать обзоры на игры, которыми не владеют
* Пользователи не могут создавать моды на игры, которыми не владеют
* После запуска игры, в записи должна изменится дата последнего запуска
* У пользователя не может быть игра в библиотеке, которой нет в Магазине

Именно из-за этих ограничений и обновлений информации происходит большое количество запросов к данной таблице.

1. Инвентарь – объект, часто подвергающийся проверкам и изменениям, то-есть запросам. В инвентаре хранятся вещи, выпавшие во время геймплея в ту или иную игру, либо же купленные на Тороговой площадке

* Пользователи не могут продавать вещи на Торговой площадке, которыми не владеют в Инвентаре
* Если пользователь продает вещь, то нужно снизить количество этой вещи в Инвентаре
* Если пользователь покупает/выбивает вещь, которая и так есть у него в инвентаре, то необходимо увеличить количество этой вещи
* Если пользователь снимает лот с Торговой площадки, то необходимо вернуть вещь в инвентарь

Именно из-за этих ограничений и обновлений информации происходит большое количество запросов к данной таблице.

Именно поэтому мне пришлось создать необходимые индексы для быстрого доступа к часто используемым таблицам.