# Инфологическое проектирование базы данных «Видеоигр»

Общая постановка задачи: база данных создается для информационного обслуживания руководства организации, руководителей и участников разработки видеоигр.

Основная деятельность – создание видеоигр по договору или в индивидуальном порядке (группы разработчиков объединяются для создания проекта в игровой индустрии без участия издателя, издатель нередко выступает в роли корпорации, а разработчики – дочерних студий).

Так как создание видеоигр один их самых сложным видов разработки и создания какого-либо видеоконтента, то, очевидно, процесс делиться на различные этапы. Сроки в игровой индустрии играют посредственную роль, из-за постоянных проблем, с которыми на этапе производства сталкиваются компании и, непосредственно, разработчики, поэтому этапы разработки служат лишь для упрощения процесса и более наглядной видимости для команды.

БД должна содержать данные об играх, издателях и разработчиках, участвующие в создании видеоигры. Разработчик может выступать в роли руководителя или консультанта.

В соответствии с предметной областью система строится с учётом следующих особенностей:

– Каждая игра может иметь несколько жанров, существует несколько игр одного жанра.

– Каждая игра должна быть создана в заданные сроки.

– Разработка каждой игры делится на несколько этапов, конкретный этап разработки относится к конкретной игре.

– Каждый разработчик может участвовать в нескольких этапах создания игры, над каждым этапом разработки может трудиться несколько разработчиков.

– Выплата разработчикам зависит от количества рабочих часов (почасовая оплата) и участия в этапах разработки (какой этап и количество этапов, в которых участвует).

– Каждую команду разработчиков возглавляет руководитель (GameDirector) – человек из этой же команды (назначается издателем, если он существует)

– Разработчик может сотрудничать (подписать договор) только с одним издателем, с издателем может работать (сотрудничать) несколько разработчиков, объединенных в команды или работающих в одиночку.

– Разработка игры может вестись без участия издателя - полное финансирование идет от самих разработчиков, либо от личных инвестиций потребителей (краудфандинговые кампании, например kickstarter) и самостоятельная продажа игры.

– Издать может издавать (выпускать) в продажу несколько игр, видеоигра издается одним издателем.

Изображение выглядит как текст, карта

Автоматически созданное описание

**Игры**: полное название игры, аббревиатура, год выпуска, дата начала разработки, платформа, бюджет

**Разработчики**: ФИО, паспортные данные, дата рождения, пол, данные об образовании, опыт работы, должность, оклад, почта, соцсети телефоны

**Жанры**: название, аббревиатура

**Издатель**: название, аббревиатура, местоположение (город) главного офиса, год основания компании, почта, контактный телефон

**Этапы разработки**: название, аббревиатуры (например ОБТ и ЗБТ – открытые и закрыты бета-тестирования), даты начала и завершения этапа, итоговая дата закрытия этапа

Руководитель разработки (GameDirector):

1. Назначить сотрудника
2. Распределить обязанности
3. Получение/изменение полной информации о игре
4. Изменение окладов
5. Полный контроль над разработкой
6. Заключение/расторжения договоров с издателем (если таковые имеются)
7. Распределение этапов разработки

Издатель:

1. Полный/частичный контроль над разработчиками
2. Заключение/расторжения договоров со студиями (командой разработчиков)
3. Назначение руководителя(ей) разработки

Сотрудники (workers):

1. Прием/увольнение сотрудников
2. Внесение/изменение данных о сотрудниках

Бухгалтеры:

1. Получение ведомостей на выплату зарплаты

Сотрудники - разработчики:

1. Просмотр данных о других участниках разработки
2. Просмотр текущего этапа разработки

# Преобразование ER–диаграммы в схему базы данных

Изображение выглядит как снимок экрана

Автоматически созданное описание

# Составление реляционных отношений

Таблица 1. Схема отношения *Игры* - Games

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Аббревиатура игры | G\_ID | С(10) | **Первичный ключ** |
| Полное название | G\_NAME | V(100) | Обязательное поле |
| Дата начала разработки | G\_DBEGIN | D | Обязательное поле |
| Дата окончания разработки | G\_DEND | D | Больше даты начала разработки |
| Дата релиза | G\_RELEASE | D | Больше даты окончания разработки |
| Платформы | G\_ PLATFORM | V(30) | Обязательное многозначное поле |
| Издатель | G\_PUBLISH | C(10) | Внешний ключ (к Publisher) |
| Бюджет | G\_BUDGET | N(10) | Обязательное поле, ≥ 0 |
| Краудфандинг | G\_CFUNDING | N(10) | ≥ 0 |

**Краудфандинг** – дополнительный вклад в бюджет игры за счет пожертвований через краудфандинговые платформы (Kickstarter).

Таблица 2. Схема отношения *Жанры* - Genres

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название жанра | GEN\_NAME | V(30) | **Первичный ключ** |

Таблица 3. Схема отношения (Игры – Жанры) *Принадлежность* - Attachment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания | |
| Жанр | A\_GEN | V(30) | Внешний ключ (к Genres) | **Составной первичный ключ** |
| Игра | A\_GAME | С(10) | Внешний ключ (к Games) |

Таблица 4. Схема отношения *Издатель* (Publisher)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Аббревиатура издателя | P\_ID | С(10) | **Первичный ключ** |
| Название | P\_NAME | V(100) | Обязательное поле |
| Год основания | P\_YEAR | N(4) | Обязательное поле |
| Местоположение главного офиса | P\_LOCA TION | V(100) | Обязательное поле |
| Почта | P\_EMAIL | V(50) | Обязательное поле |
| Контактный телефон | P\_PHONE | V(30) | Обязательное поле |

Таблица 5. Схема отношения *Разработчики* (Developers)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификационный номер | D\_ID | N(5) | **Первичный ключ** |
| ФИО | D\_NAME | V(50) | Обязательное поле |
| Дата рождения | D\_BERN | D | Обязательное поле |
| Пол | D\_GENDER | C(1) | Обязательное поле, «M» или «F» |
| Паспортные данные | D\_PASSPORT | V(50) | Обязательное поле |
| Образование | D\_EDU | V(250) | Многозначное поле |
| Опыт работы | D\_EXPER | N(2) | Обязательное поле, > 0 |
| Издатель | D\_PUBLISH | C(10) | Внешний ключ (к Publisher) |
| Должность | D\_POST | V(50) | Обязательное поле |
| Оклад | D\_SALARY | N(10) | ≥ 0 |
| Почта | D\_EMAIL | V(50) | Обязательное поле |
| Соцсети | D\_SOCIAL | V(150) | Многозначное поле |
| Телефоны | D\_PHONE | V(30) | Многозначное поле |

**Образование** – необязательное поле, потому что могут же брать хороших программистов без высшего.

Таблица 6. Схема отношения (Игры – Разработчики)

*Руководство* – Guide

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания | |
| Игра | GD\_GAME | С(10) | Внешний ключ (к Games) | **Составной первичный ключ** |
| Руководитель разработки | GD\_SUPERVISOR | N(5) | Внешний ключ (к Developers) |

Таблица 7. Схема отношения *Этапы разработки* - DevStages

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Номер этапа | DS\_NUM | N(5) | **Первичный ключ** |
| Игра | DS\_GAME | С(10) | Внешний ключ (к Games) |
| Название этапа | DS\_NAME | V(100) | Обязательное поле |
| Аббревиатура | DS\_ABR | С(3) | Обязательное поле, «ОБТ» или «ЗБТ» |
| Дата начала этапа | DS\_DBEGIN | D | Обязательное поле |
| Дата завершения этапа | DS\_DEND | D | Обязательное поле, больше даты начала этапа |
| Итоговая дата закрытия этапа | DS\_ FINISH | D | Больше даты начала этапа |

Таблица 8. Схема отношения (Этапы разработки – Разработчики)

*Участие* – Participation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания | |
| Этап разработки | PT\_STAGE | N(5) | Внешний ключ (к DevStages) | **Составной первичный ключ** |
| Разработчик | PT\_DEVELOP | N(5) | Внешний ключ (к Developers) |
| Деятельность | PT\_JOB | V(100) | Обязательно поле |
| Почасовая оплата | PT\_SALARY | N(10) | Обязательное поле, ≥ 1000 (₽) | |

# Нормализация полученных отношений

1. В отношении *Игры* многозначное поле «Платформы» вынесем в отдельное отношение *Компьютерные платформы* - Platforms:

Таблица 9. Схема отношения *Компьютерные платформы* – Platforms

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название комп. платформы | PL\_ NAME | V(30) | **Первичный ключ** |

Так как теперь между отношениями *Игры* (Games) и *Компьютерные платформы* (Platforms) связь многие ко многим (под каждую платформу выпускается несколько игр, игра может выпускаться под несколько копм. платформ), то введем отношение связи *Выпуск* (Release), где поля «Игра» и «Платформа» будут соответственно внешними ключами к отношениям *Игры* (Games) и *Компьютерные Платформы* (Platforms).

Таблица 10. Схема отношения (Компьютерные платформы – Игры)

*Выпуск* – Release

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания | |
| Игра | R\_GAME | С(10) | Внешний ключ (к Games) | **Составной первичный ключ** |
| Платформа | R\_PLATFORM | V(30) | Внешний ключ (к Platforms) |

1. В отношении *Разработчики* составные поля «ФИО», «Паспортные данные» и «Образование» разобьем на простые атрибуты:

* «ФИО» - атрибуты «Имя», «Фамилия», «Отчество»
* «Паспортные данные» - атрибуты «Серия, номер паспорта», «Когда выдан» и «Кем выдан»
* «Образование» - атрибуты «Университет», «Номер диплома», «Специальность», «Год окончания»

1. Так как поле «Образование» также является многозначным, то вынесем его в отдельное отношение *Образование разработчика* – EduDevelop

Таблица 11. Схема отношения *Образование разработчика* – EduDevelop

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Разработчик | ED\_ DEVELOP | С(10) | Внешний ключ (к Developers) |
| Университет | ED\_ UNI | V(50) | Обязательное поле |
| Номер диплома | ED\_DIPNUM | V(20) |  |
| Специальность | ED\_SPEC | V(40) | Обязательное поле |
| Год окончания | ED\_DEND | N(4) |  |

В таблице нет первичного ключа, так как на нее никто не ссылается – отношение *Образование разработчика* является справочным.

Атрибуты «Номер диплома» и «Год окончания» необязательные, потому что разработчик на момент работы может еще учиться.

1. Поля «Соцсети» и «Телефоны» являются многозначным – в отдельные справочные (без ПК) отношения *Соцсети разработчика* (SocialDevelop) и *Телефоны разработчика* (PhoneDevelop) с внешними ключами на разработчика.

Таблица 12. Схема отношения *Соцсети разработчика* – SocialDevelop

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Разработчик | SD\_ DEVELOP | С(10) | Внешний ключ (к Developers) |
| Ссылка на соцсеть | SD\_SOCIAL | V(50) |  |

Таблица 13. Схема отношения *Телефоны разработчика* – PhoneDevelop

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Разработчик | PD\_ DEVELOP | С(10) | Внешний ключ (к Developers) |
| Телефон | PD\_PHONE | V(20) |  |

1. Атрибут «Оклад» в отношении *Разработчики* функционально зависит от атрибута «Должность». Поэтому вынесем их в отдельное отношение *Должности* (PostsSalary), где поле «Название должности» будет первичным ключом, и на него будет ссылаться атрибут «Должность» в таблице *Разработчики*.

Таблица 14. Схема отношения *Должности* – PostsSalary

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название должности | PS\_ PNAME | V(50) | **Первичный ключ** |
| Оклад | PS\_SALARY | N(10) | > 0 |

1. Атрибут «Почасовая оплата» в отношении *Участие* функционально зависит от атрибута «Деятельность». Поэтому вынесем их в отдельное отношение *Деятельности* (JobSalary), где поле «Название деятельности» будет первичным ключом, и на него будет ссылаться атрибут «Деятельность» в таблице *Участие*.

Таблица 15. Схема отношения *Деятельности* – JobSalary

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название деятельности | JS\_JNAME | V(100) | **Первичный ключ** |
| Почасовая оплата | JS\_SALARY | N(10) | Обязательное поле, ≥ 1000 (₽) |

Отношения, полученные после нормализации

Таблица 16. Схема отношения *Игры* - Games

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Аббревиатура игры | G\_ID | С(10) | **Первичный ключ** |
| Полное название | G\_NAME | V(100) | Обязательное поле |
| Дата начала разработки | G\_DBEGIN | D | Обязательное поле |
| Дата окончания разработки | G\_DEND | D | Больше даты начала разработки |
| Дата релиза | G\_RELEASE | D | Больше даты окончания разработки |
| Издатель | G\_PUBLISH | C(10) | Внешний ключ (к Publisher) |
| Бюджет | G\_BUDGET | N(10) | Обязательное поле, ≥ 0 |
| Краудфандинг | G\_CFUNDING | N(10) | ≥ 0 |

Таблица 17. Схема отношения *Жанры* - Genres

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название жанра | GEN\_NAME | V(30) | **Первичный ключ** |

Таблица 18. Схема отношения (Игры – Жанры)

*Принадлежность* - Attachment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания | |
| Жанр | A\_GEN | V(30) | Внешний ключ (к Genres) | **Составной первичный ключ** |
| Игра | A\_GAME | C(10) | Внешний ключ (к Games) |

Таблица 19. Схема отношения *Компьютерные платформы* – Platforms

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название комп. платформы | PL\_ NAME | V(30) | **Первичный ключ** |

Таблица 20. Схема отношения (Компьютерные платформы – Игры)

*Выпуск* – Release

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания | |
| Игра | R\_GAME | С(10) | Внешний ключ (к Games) | **Составной первичный ключ** |
| Платформа | R\_PLATFORM | V(30) | Внешний ключ (к Platforms) |

Таблица 21. Схема отношения *Издатель* (Publisher)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Аббревиатура издателя | P\_ID | С(10) | **Первичный ключ** |
| Название | P\_NAME | V(100) | Обязательное поле |
| Год основания | P\_YEAR | N(4) | Обязательное поле |
| Местоположение главного офиса | P\_LOCA TION | V(100) | Обязательное поле |
| Почта | P\_EMAIL | V(50) | Обязательное поле |
| Контактный телефон | P\_PHONE | V(30) | Обязательное поле |

Таблица 22. Схема отношения *Разработчики* (Developers)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификационный номер | D\_ID | N(5) | **Первичный ключ** |
| Фамилия | D\_ LNAME | V(25) | Обязательное поле |
| Имя | D\_ FNAME | V(20) | Обязательное поле |
| Отчество | D\_ PNAME | V(25) |  |
| Дата рождения | D\_BERN | D | Обязательное поле |
| Пол | D\_GENDER | C(1) | Обязательное поле, «M» или «F» |
| Серия, номер паспорта | D\_PASSNUM | C(10) | Обязательное уникальное поле |
| Кем выдан | D\_PASSGIVE | V(50) | Обязательное поле |
| Когда выдан | D\_PASSDATE | D | Обязательное поле |
| Опыт работы | D\_EXPER | N(2) | Обязательное поле, > 0 |
| Издатель | D\_PUBLISH | C(10) | Внешний ключ (к Publisher) |
| Должность | D\_POST | V(50) | Внешний ключ (к PostsSalary) |
| Почта | D\_EMAIL | V(50) | Обязательное поле |

Таблица 22. Схема отношения *Образование разработчика* – EduDevelop

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Разработчик | ED\_ DEVELOP | С(10) | Внешний ключ (к Developers) |
| Университет | ED\_ UNI | V(50) | Обязательное поле |
| Номер диплома | ED\_DIPNUM | V(20) |  |
| Специальность | ED\_SPEC | V(40) | Обязательное поле |
| Год окончания | ED\_DEND | N(4) |  |

Таблица 23. Схема отношения *Должности*– PostsSalary

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название должности | PS\_ PNAME | V(50) | **Первичный ключ** |
| Оклад | PS\_SALARY | N(10) | > 0 |

Таблица 24. Схема отношения *Соцсети разработчика* – SocialDevelop

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Разработчик | SD\_ DEVELOP | С(10) | Внешний ключ (к Developers) |
| Ссылка на соцсеть | SD\_SOCIAL | V(50) |  |

Таблица 25. Схема отношения *Телефоны разработчика* – PhoneDevelop

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Разработчик | PD\_ DEVELOP | С(10) | Внешний ключ (к Developers) |
| Телефон | PD\_PHONE | V(20) |  |

Таблица 26. Схема отношения (Игры – Разработчики)

*Руководство* – Guide

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания | |
| Игра | GD\_GAME | С(10) | Внешний ключ (к Games) | **Составной первичный ключ** |
| Руководитель разработки | GD\_SUPERVISOR | N(5) | Внешний ключ (к Developers) |

Таблица 27. Схема отношения *Этапы разработки* - DevStages

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Номер этапа | DS\_NUM | N(5) | **Первичный ключ** |
| Игра | DS\_GAME | С(10) | Внешний ключ (к Games) |
| Название этапа | DS\_NAME | V(100) | Обязательное поле |
| Аббревиатура | DS\_ABR | С(3) | Обязательное поле, «ОБТ» или «ЗБТ» |
| Дата начала этапа | DS\_DBEGIN | D | Обязательное поле |
| Дата завершения этапа | DS\_DEND | D | Обязательное поле, больше даты начала этапа |
| Итоговая дата закрытия этапа | DS\_ FINISH | D | Больше даты начала этапа |

Таблица 28. Схема отношения (Этапы разработки – Разработчики)

*Участие* – Participation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания | |
| Этап разработки | PT\_STAGE | N(5) | Внешний ключ (к DevStages) | **Составной первичный ключ** |
| Разработчик | PT\_DEVELOP | N(5) | Внешний ключ (к Developers) |
| Деятельность | PT\_JOB | V(100) | Внешний ключ (к JobSalary) |

Таблица 29. Схема отношения *Деятельность* – JobSalary

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Название деятельности | JS\_JNAME | V(100) | **Первичный ключ** |
| Почасовая оплата | JS\_SALARY | N(10) | Обязательное поле, ≥ 1000 (₽) |

Схема базы данных после нормализации:

Изображение выглядит как снимок экрана

Автоматически созданное описание

# Права доступа пользователей базы данных:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблицы | Группы пользователей | | | | |
| Руководитель разработки | Издатель | Сотрудники | Бухгалтеры | Сотрудники - разработчики |
| Комп. Платформы | SI | SI |  |  |  |
| Игры | SI | SI |  |  |  |
| Жанры | SIUD | S |  |  |  |
| Этапы разработки | SIU | SIUD |  |  | S |
| Деятельность | S | SIUD | SIUD | S |  |
| Разработчики | SIUD | SIUD | SIUD | S |  |
| Издатель | S | SIUD |  |  |  |
| Соцсети | S | S | SIUD | S |  |
| Телефоны | S | S | SIUD | S |  |
| Должности | SIUD | S | S |  |  |
| Образование | S | S | SIUD |  |  |

# Создание таблиц:

CREATE TABLE Publisher

(

P\_ID CHAR(10) PRIMARY KEY,

P\_NAME VARCHAR(100) NOT NULL,

P\_YEAR NUMERIC(4) NOT NULL,

P\_LOCATION VARCHAR(100) NOT NULL,

P\_EMAIL VARCHAR(50) NOT NULL,

P\_PHONE VARCHAR(30) NOT NULL

);

CREATE TABLE Games

(

G\_ID CHAR(10) PRIMARY KEY,

G\_NAME VARCHAR(100) NOT NULL UNIQUE,

G\_DBEGIN DATE NOT NULL,

G\_DEND DATE CHECK(G\_DEND > G\_DBEGIN),

G\_RELEASE DATE CHECK(G\_YEAR > G\_DEND),

G\_PUBLISH CHAR(10) REFERENCES Publisher(P\_ID),

G\_BUDGET NUMERIC(10) NOT NULL CHECK(G\_BUDGET >= 0),

G\_CFUNDING NUMERIC(10) CHECK(G\_CFUNDING >= 0)

);

CREATE TABLE Genres

(

GEN\_NAME VARCHAR(30) PRIMARY KEY

);

CREATE TABLE Attachment

(

A\_GEN VARCHAR(30) REFERENCES Genres(GEN\_NAME),

A\_GAME CHAR(10) REFERENCES Games(G\_ID)

);

CREATE TABLE PostsSalary

(

PS\_PNAME VARCHAR(50) PRIMARY KEY,

PS\_SALARY NUMERIC(10) CHECK(PS\_SALARY > 0)

);

CREATE TABLE Developers

(

D\_ID NUMERIC(5) PRIMARY KEY,

D\_LNAME VARCHAR(25) NOT NULL,

D\_FNAME VARCHAR(20) NOT NULL,

D\_PNAME VARCHAR(25),

D\_BERN DATE NOT NULL,

D\_GENDER CHAR(1) NOT NULL CHECK(D\_GENDER = 'M' OR D\_GENDER = 'F'),

D\_PASSNUM CHAR(10) NOT NULL UNIQUE,

D\_PASSGIVE VARCHAR(50) NOT NULL,

D\_PASSDATE DATE NOT NULL,

D\_EXPER NUMERIC(2) NOT NULL CHECK(D\_EXPER > 0),

D\_PUBLISH CHAR(10) REFERENCES Publisher(P\_ID),

D\_POST VARCHAR(50) REFERENCES PostsSalary(PS\_PNAME),

D\_EMAIL VARCHAR(50) NOT NULL UNIQUE

);

CREATE TABLE Guide

(

GD\_GAME CHAR(10) REFERENCES Games(G\_ID),

GD\_SUPERVISOR NUMERIC(5) REFERENCES Developers(D\_ID)

);

CREATE TABLE DevStages

(

DS\_NUM NUMERIC(5) PRIMARY KEY,

DS\_GAME CHAR(10) REFERENCES Games(G\_ID),

DS\_NAME VARCHAR(100) NOT NULL,

DS\_ABR CHAR(3) NOT NULL CHECK(DS\_ABR = 'OBT' OR DS\_ABR = 'CBT'),

DS\_DBEGIN DATE NOT NULL,

DS\_DEND DATE NOT NULL CHECK(DS\_DEND > DS\_DBEGIN),

DS\_FINISH DATE CHECK(DS\_FINISH > DS\_DBEGIN)

);

CREATE TABLE JobSalary

(

JS\_JNAME VARCHAR(100) PRIMARY KEY,

JS\_SALARY NUMERIC(10) NOT NULL CHECK(JS\_SALARY >= 1000)

);

CREATE TABLE Participation

(

PT\_STAGE NUMERIC(5) REFERENCES DevStages(DS\_NUM),

PT\_DEVELOP NUMERIC(5) REFERENCES Developers(D\_ID),

PT\_JOB VARCHAR(100) REFERENCES JobSalary(JS\_JNAME)

);

CREATE TABLE Platforms

(

PL\_NAME VARCHAR(30) PRIMARY KEY

);

CREATE TABLE Releasee

(

R\_GAME CHAR(10) REFERENCES Games(G\_ID),

R\_PLATFORM VARCHAR(30) REFERENCES Platforms(PL\_NAME)

);

CREATE TABLE EduDevelop

(

ED\_DEVELOP NUMERIC(5) REFERENCES Developers(D\_ID),

ED\_UNI VARCHAR(50) NOT NULL,

ED\_DIPNUM VARCHAR(20),

ED\_SPEC VARCHAR(40) NOT NULL,

ED\_DEND NUMERIC(4)

);

CREATE TABLE SocialDevelop

(

SD\_DEVELOP NUMERIC(5) REFERENCES Developers(D\_ID),

SD\_SOCIAL VARCHAR(50)

);

CREATE TABLE PhoneDevelop

(

PD\_DEVELOP NUMERIC(5) REFERENCES Developers(D\_ID),

PD\_PHONE VARCHAR(20)

);

# Создание представлений:

1. Соединение 4-х таблиц

create or replace view relations as

select \*

from DevStages ds, Games g, Participation pt, Developers d

where ds.ds\_game = g.g\_id and pt.pt\_stage = ds.ds\_num and pt.pt\_develop = d.d\_id

2. Разработчики с несколькими завершенными образованиями

create or replace view education\_dev\_more1 (name\_dev, count\_edu) as

select d\_lname || ' ' || d\_fname || ' ' || d\_pname, count(ed\_develop)

from Developers, EduDevelop

where d\_id = ed\_develop and ed\_dipnum is not null and ed\_dend is not null

group by d\_lname, d\_fname, d\_pname

having count(ed\_develop) > 1

order by 1;

3. Список людей фрилансеров (работа не по контракту с издателем), которые занимались разработкой уже вышедших в релиз игр

create or replace view freelance (Dev\_name, Game\_name, Game\_release) as

select distinct r.d\_fname || ' ' || r.d\_lname || ' ' || r.d\_pname, r.g\_name, r.g\_release

from relations r

where r.d\_publish is null and r.d\_post is null and r.g\_release < current\_date

order by r.g\_release;

4. Количество этапов разработки каждой игры

create or replace view count\_stages (Game\_name, count\_devstage) as

select g.g\_name, count(ds.ds\_game)

from Games g, Participation pt, DevStages ds

where ds.ds\_game = g.g\_id and pt.pt\_stage = ds.ds\_num

group by g.g\_name

order by count(ds.ds\_game);

5. Список людей участвующих в разработке конкретной игры

create or replace view all\_develop\_game (Game, Name, Role) as

select g\_name, d\_fname || ' ' || d\_lname || ' ' || d\_pname, 'Разработчик'

from relations

where g\_name = 'Название игры'

union all

select g\_name, d\_fname || ' ' || d\_lname || ' ' || d\_pname, 'Руководитель'

from Developers, Guide, Games

where g\_name = 'Название игры' and gd\_game = g\_id and gd\_supervisor = d\_id

order by 2;

6. Посчитать почасовую оплату разработчика за участие в этапах разработки - разработчик-фрилансер получает почасовую оплату в зависимости от деятельности в этапе разработки, может участвовать в нескольких этапах (например, в одном этапе разработчик получает 1500/час, в другом этапе 1300/час и т.д, представление считает сумму 1500 + 1300 + … = общая сумма в час)

create or replace view sum\_jobsalary (First\_Name, Last\_Name, Salary) as

select d\_fname, d\_lname, sum(js\_salary)

from Participation, Developers, JobSalary

where pt\_develop = d\_id and pt\_job = js\_jname

group by d\_fname, d\_lname

order by sum(js\_salary);

7.

-- Количество жанров игры и поддерживаемых платформ

create or replace view count\_gen\_pl (game\_name, count\_genres, count\_platforms) as

select g.g\_name,

(select count(a.a\_game) from Attachment a, Genres gen where g.g\_id = a.a\_game and gen.gen\_name = a.a\_gen ) genres,

(select count(r.r\_game) from Releasee r, Platforms pl where g.g\_id = r.r\_game and pl.pl\_name = r.r\_platform ) platforms

from Games g

order by genres, platforms;

# Назначение прав доступа:

CREATE ROLE Game\_director WITH LOGIN PASSWORD 'director';

CREATE ROLE Publ\_user WITH LOGIN PASSWORD 'publuser';

CREATE ROLE Users WITH LOGIN PASSWORD 'users';

CREATE ROLE Booker WITH LOGIN PASSWORD 'booker';

CREATE ROLE Dev\_users WITH LOGIN PASSWORD 'dusers';

GRANT SELECT, INSERT ON Platforms TO Game\_director;

GRANT SELECT, INSERT ON Games TO Game\_director;

GRANT SELECT, INSERT, UPDATE, DELETE ON Genres TO Game\_director;

GRANT SELECT, INSERT, UPDATE ON Participation TO Game\_director;

GRANT SELECT ON JobSalary TO Game\_director;

GRANT SELECT, INSERT, UPDATE, DELETE ON Developers TO Game\_director;

GRANT SELECT ON Publisher TO Game\_director;

GRANT SELECT ON SocialDevelop TO Game\_director;

GRANT SELECT ON PhoneDevelop TO Game\_director;

GRANT SELECT, INSERT, UPDATE, DELETE ON PostsSalary TO Game\_director;

GRANT SELECT ON EduDevelop TO Game\_director;

GRANT SELECT, INSERT ON Platforms TO Publ\_user;

GRANT SELECT, INSERT ON Games TO Publ\_user;

GRANT SELECT ON Genres TO Publ\_user;

GRANT SELECT, INSERT, UPDATE, DELETE ON Participation TO Publ\_user;

GRANT SELECT, INSERT, UPDATE, DELETE ON JobSalary TO Publ\_user;

GRANT SELECT, INSERT, UPDATE, DELETE ON Developers TO Publ\_user;

GRANT SELECT, INSERT, UPDATE, DELETE ON Publisher TO Publ\_user;

GRANT SELECT ON SocialDevelop TO Publ\_user;

GRANT SELECT ON PhoneDevelop TO Publ\_user;

GRANT SELECT ON PostsSalary TO Publ\_user;

GRANT SELECT ON EduDevelop TO Publ\_user;

GRANT SELECT, INSERT, UPDATE, DELETE ON JobSalary TO Users;

GRANT SELECT, INSERT, UPDATE, DELETE ON Developers TO Users;

GRANT SELECT, INSERT, UPDATE, DELETE ON SocialDevelop TO Users;

GRANT SELECT, INSERT, UPDATE, DELETE ON PhoneDevelop TO Users;

GRANT SELECT ON PostsSalary TO Users;

GRANT SELECT, INSERT, UPDATE, DELETE ON EduDevelop TO Users;

GRANT SELECT ON JobSalary TO Booker;

GRANT SELECT ON Developers TO Booker;

GRANT SELECT ON SocialDevelop TO Booker;

GRANT SELECT ON PhoneDevelop TO Booker;

GRANT SELECT ON Participation TO Dev\_users;

# Создание индексов:

create index G\_PUBL\_INDX on Games(G\_PUBLISH);

create index A\_ATTACH\_INDX on Attachment(A\_GEN, A\_GAME);

create index R\_RELEASE\_INDX on Releasee(R\_GAME, R\_PLATFORM);

create index D\_PUBL\_INDX on Developers(D\_PUBLISH);

create index D\_POST\_INDX on Developers(D\_POST);

create index ED\_DEVEL\_INDX on EduDevelop(ED\_DEVELOP);

create index SD\_DEVEL\_INDX on SocialDevelop(SD\_DEVELOP);

create index PD\_DEVEL\_INDX on PhoneDevelop(PD\_DEVELOP);

create index GD\_GUIDE\_INDX on Guide(GD\_GAME, GD\_SUPERVISOR);

create index DS\_GAME\_INDX on DevStages(DS\_GAME);

create index PARTIC\_INDX on Participation(PT\_STAGE, PT\_DEVELOP, PT\_JOB);

# Определение требований к операционной обстановке:

Так как игровая индустрия еще не структурирована, процесс разработки игр в каждой компании проходит по-разному, то говорить о требованиях к опер. обстановке приходится сугубо предположительно.

Будем считать, что в среднем процесс разработки игры занимает 2 года, и для примера возьмем компанию – издателя с внутренними студиями. Тогда:

* Одновременно могут быть в разработке несколько игр (проектов), например 5, каждая игра будет занимать 2Кб (из-за жанров, платформ, масштаба игры)
* «Этапов» может быть много, но основных 5 (0.5 Кб на этап)
* Сотрудников (в общем и целом, в компании) может быть около 1000 (0.3 Кб на сотрудника)
* Наличие аутсорсеров, как сотрудников, - 10 (0.1 Кб на человека)
* Участвовать в проекте может 150 человек (0.2 кб на человека)

М(д) = 5\*2+5\*5\*0.5+1000\*0.3+10\*0.1+5\*5\*150\*0.2= 5570 Кб

# Выбор СУБД и других программных средств:

В рамках данной Базы Данных мы выбрали Postgree Sql как основную программу.