**Отчёт по проекту базы данных: создание схемы и процедур**

**1. Описание предметной области**

Предметная область охватывает платформу для пользователей, где:

* Пользователи могут регистрироваться, оставлять отзывы и оценивать игры.
* Информация о разработчиках, жанрах и платформах для игр централизована.
* Игры связаны с определёнными платформами, разработчиками и жанрами, а также имеют средний рейтинг, обновляемый динамически на основе отзывов.

Целью является создание базы данных, которая обеспечит удобное хранение, модификацию и обработку информации для:

* Управления пользователями и их правами.
* Каталогизации игр, их жанров, разработчиков и платформ.
* Управления отзывами и оценками, которые влияют на общий рейтинг игр.

**2. Требования к базе данных**

1. **Схемы:**
   * Логическая организация объектов базы данных в три схемы:
     + auth — хранение информации о пользователях и их ролях.
     + games — каталогизация игр, платформ, жанров и разработчиков.
     + users — хранение отзывов пользователей.
2. **Функциональность:**
   * Валидация данных (например, уникальность имени платформы, ограничение рейтинга в пределах от 0 до 10).
   * Обеспечение каскадного удаления связанных записей для поддержания целостности.
   * Поддержка производительности через индексы (поиск игр по названию).
   * Автоматическое обновление среднего рейтинга игры при добавлении, обновлении или удалении отзывов.
3. **Безопасность:**
   * Разделение привилегий по схемам с помощью процедур grant\_privileges\_for\_schema.
4. **Оптимизация:**
   * Минимизация избыточности данных через нормализацию.
   * Сохранение уникальности и референциальной целостности.

**3. Схема базы данных**

**Схема auth**

* Таблица Users:
  + Хранит информацию о пользователях (уникальное имя, хэш пароля, роль).

**Схема games**

* Таблицы:
  + Developers: разработчики с именами и странами.
  + Genres: жанры игр.
  + Platforms: платформы (уникальное имя платформы).
  + Games: игры с ссылками на разработчиков, жанры, а также средней оценкой.
  + GamePlatforms: связывает игры с платформами (многие-ко-многим).

**Схема users**

* Таблица Reviews:
  + Отзывы пользователей, включающие рейтинг и комментарий.

**4. Нормальные формы**

1. **Первая нормальная форма (1НФ):**
   * Все атрибуты атомарны (неделимы).
   * Пример: поле Title в Games содержит только название игры, а не дополнительные метаданные.
2. **Вторая нормальная форма (2НФ):**
   * Все атрибуты зависят от первичного ключа.
   * Пример: поле DeveloperID в Games зависит только от первичного ключа GameID.
3. **Третья нормальная форма (3НФ):**
   * Все неключевые атрибуты независимы друг от друга.
   * Пример: поле AverageRating зависит только от GameID и не связано напрямую с другими таблицами.

**5. Обоснование нормализации**

Нормализация позволила:

* Избежать избыточности данных (например, хранение имени платформы в отдельной таблице).
* Упростить обновление и удаление данных.
* Поддерживать референциальную целостность через внешние ключи.

**6. Описание объектов и процедур**

1. **Процедура create\_tables**:
   * Создаёт схемы, таблицы и привилегии, если они ещё не существуют.
   * Гибкость: процедура выполняет проверку на наличие объектов перед их созданием.
2. **Индекс на поле Title**:
   * Повышает производительность при поиске игр.
3. **Триггер для обновления среднего рейтинга**:
   * Динамически обновляет AverageRating при изменении отзывов, что устраняет необходимость дополнительных запросов.

**7. Вывод**

Проект удовлетворяет требованиям предметной области, включая модульность, производительность и безопасность. Благодаря нормализации достигается минимизация избыточности, а автоматизация через процедуры и триггеры упрощает управление данными.