

Задание 3.2: Анализ БД на избыточность

Проанализировать информацию по избыточности баз данных и привести примеры неверного логического проектирования. Предложить алгоритм решения (по устранению) недостатков логической структуры.

Избыточность в базах данных может проявляться в различных формах. Одной из распространенных форм является дублирование данных, когда одна и та же информация хранится в нескольких таблицах или записях. Другой формой является функциональная зависимость, когда один атрибут таблицы может определяться другим атрибутом.

Причины избыточности:

- Отсутствие нормализации: Неспособность должным образом нормализовать базу данных может привести к избыточности.

Рассмотреть случай, когда в отношении ПЗ (Позиция заказа) будут так же указаны атрибуты «Мин игроков» и «Макс игроков». Данное отношение не будет соответствовать 2НФ, так как РК является комбинация атрибутов «Номер заказа», «Название игры» и «Производитель», однако атрибуты «Мин игроков» и «Макс игроков» будут зависеть только от части первичного ключа, а именно от атрибутов «Название игры» и «Производитель».

- Неполные обновления: Когда обновления не применяются последовательно ко всем экземплярам дублирующихся данных, это может привести к несогласованности и избыточности.

Можно предположить, что в сущности ПЗ (Позиция заказа) атрибуты «Название игры» и «Производитель» будут не внешними ключами, а просто вводимыми данными. В таком случае если придётся изменять название какой-либо игры или какого-либо производителя, то менять данные нужно будет в обоих отношениях. При ошибочном (неполном) обновлении из одной игры появятся несколько: старая (не обновленная) и новая (обновленная). Таким образом появится недействительная игра, которая станет избыточностью.

- Плохое проектирование базы данных: Плохо спроектированная схема базы данных может привести к избыточности. Например, если схема содержит ненужные повторяющиеся столбцы или таблицы, может возникнуть избыточность.

В случае если телефон клиента будет указан в отношении З (Заказ) и в отношении К (Клиент), то атрибут «Телефон» будет повторяться ровно столько раз, сколько заказов сделает клиент (в тех же строках), то есть личная информация, которая зависит напрямую от клиента добавлена ещё и в заказ. Это создает излишние данные.

Алгоритм решения (по устранению) недостатков логической структуры:

- 1) Проверить базу данных на избыточность
- 2) Провести тщательную нормализацию таблиц (может помочь уменьшить избыточность за счет организации данных в таблицы и устранения дублирующих данных)

- 3) При нормализации использовать внешние ключи (могут использоваться для установления связей между таблицами, уменьшая потребность в дублирующихся данных)
- 4) Внедрить строгие правила проверки данных (может помочь предотвратить избыточность, гарантируя, что в базу данных вводятся только достоверные и непротиворечивые данные)
- 5) Реализовать ограничения базы данных, таких как ограничения уникальности и ссылочной целостности (может помочь предотвратить избыточность и поддерживать согласованность данных)
- 6) Регулярно обновлять и обслуживать (может помочь обеспечить идентификацию и удаление или консолидацию избыточных данных)