

Самостоятельная работа №3

Задание 3.2 Анализ БД на избыточность

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

Примеры неверного логического проектирования:

1. Дублирование данных в БД (например, имена сотрудников хранятся в двух таблицах одновременно).
2. Отсутствие нормализации (данные повторяются к разных строках таблицы).
3. Неправильная реализация связей (например, связь многие-ко-многим реализована без промежуточной таблицы)
4. Отсутствие целостности данных.

Рассмотрим примеры:

1. Допустим, что в таблицу ЗАКАЗЫ из задание 3.1 для заказов с доставкой добавили поле «Адрес»:

Номер	Получение	Создан	Таб номер	ИД клиента	Адрес
1	Самовывоз	12.09.2017 16:12	C1	K1	
2	Самовывоз	12.09.2017 16:14	C1	K2	
3	Доставка	13.09.2017 10:09	C2	K2	ул. Пионерская, 26-58

Но мы и так можем определить адрес клиента, воспользовавшись таблицей КЛИЕНТЫ, воспользовавшись **ИД клиента!**

2. Например, в системе что-то сломалось, и, когда клиент с номером K2 сделал свой второй заказ, в таблицу КЛИЕНТЫ добавилась еще одна строка с его информацией. Абсолютно идентичная.
3. По заданию 3.1 помним, что связь ЗАКАЗ-ИГРА – это многое-ко-многим. Поэтому если не реализовывать промежуточную таблицу, то получится таблица-монстр, содержащая много лишних данных:

Номер	Получение	Создан	Таб номер	Ид клиента	Название	Производитель
1	Самовывоз	12.09.2017 16:12				
2	Самовывоз	12.09.2017 16:14				
3	Доставка	13.09.2017 10:09				

В итоге получится не три строки таблицы, а сколько, сколько игр в сумме было заказано.

4. Если же в этой же таблице ЗАКАЗЫ не будет внешнего ключа, связывающего ее с таблицей КЛИЕНТЫ или же СОТРУДНИК, то это приведет к потере целостности данных.

Какой алгоритм можно предложить для того, чтобы не допускать подобных ошибок:

1. Для начала нужно изучить уже имеющуюся БД на предмет избыточности. Определить, если она есть, то где именно.
2. Определить связи между сущностями
3. Нормализовать таблицы, опираясь на связи.

4. Для сохранения целостности установить первичные и внешние ключи, а также определить, где значения не должны быть нулями или должны быть уникальными.
5. Проверить еще раз получившуюся БД на предмет оставшейся и незамеченной избыточности. Если таковая нашлась, повторить процесс.