##### **Вариативная часть**

**Вариант 1**.

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

Избыточность возникает, когда одни и те же данные хранятся в нескольких местах без необходимости, что приводит к Аномалиям обновления (необходимость синхронизации данных в нескольких таблицах), увеличению размера БД, снижению производительности (из-за лишних операций вставки, обновления и удаления).

Примеры неверного логического проектирования

1: Нарушение нормальных форм (1НФ, 2НФ, 3НФ, НФБК)

Orders

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OrderID | CustomerName | CustomerPhone | ProductID | ProductName | Quantity |

В таблице Orders хранятся данные о заказе и клиенте, но есть некоторые нарушения:

1. Нарушение 2НФ - ProductName зависит только от ProductID, а не от всего ключа
2. Нарушение 3НФ - CustomerPhone зависит от CustomerName, а не от OrderID

2: Избыточная связь

Employees

|  |  |  |  |
| --- | --- | --- | --- |
| EmployeeID | Name | DepartmentID | DepartmentName |

В таблице Employees хранится и ID отдела, и название отдела, но название отдела дублируется для каждого сотрудника

3: Хранение вычисляемых данных

Invoices

|  |  |
| --- | --- |
| InvoiceID | TotalAmount |

|  |  |  |  |
| --- | --- | --- | --- |
| InvoiceID | ItemID | Price | Quantity |

InvoiceItems

В таблице Invoices хранится общая сумма, которая может быть вычислена из позиций. Проблема в том, что TotalAmount можно получить агрегацией SUM(Price \* Quantity)

Алгоритм устранения недостатков логической структуры

1. Анализ текущей структуры:

Выявить дублирование данных.

Проверить соблюдение нормальных форм (1НФ → 2НФ → 3НФ → НФБК).

Найти избыточные связи и вычисляемые поля.

1. Нормализация:

1НФ: Убрать повторяющиеся группы (разделить строки).

2НФ: Удалить частичные зависимости (вынести атрибуты, зависящие от части ключа).

3НФ: Удалить транзитивные зависимости (вынести атрибуты, зависящие от неключевых полей).

НФБК: Устранить зависимости от составных ключей.

1. Оптимизация связей:

Заменить избыточные связи на промежуточные таблицы.

Проверить необходимость денормализации (если нужна оптимизация для чтения).

1. Удаление вычисляемых полей:

Заменить их на представления (VIEW) или вычисления в запросе.

1. Проверка целостности:

Внедрить внешние ключи (FOREIGN KEY).

Использовать триггеры или ограничения (CHECK) для сложных правил.

1. Тестирование:

Проверить, что изменения не нарушают бизнес-логику.

Оценить влияние на производительность.